**Building and Releasing**

**GenICam**

Install, Compile, and Test



Table of Contents

[1 Overview 3](#_Toc305439320)

[2 Building GenICam on Microsoft Windows 3](#_Toc305439321)

[2.1 Minimum Required Tools 3](#_Toc305439322)

[2.2 Tools Required for Building Setup 3](#_Toc305439323)

[2.3 Tools Required for Building Documentation 3](#_Toc305439324)

[2.4 Optional Tools 3](#_Toc305439325)

[2.5 Building and Testing Manually 3](#_Toc305439326)

[2.6 Supporting a New VisualStudio ServicePack 3](#_Toc305439327)

[2.7 Use the GenICam NSIS Installer 3](#_Toc305439329)

[3 Building GenICam on Linux and Mac OS X 3](#_Toc305439330)

[3.1 Minimum Required Tools 3](#_Toc305439331)

[3.2 Getting the sources 3](#_Toc305439332)

[3.3 Building and Testing 3](#_Toc305439333)

[3.4 Creating the packages for distribution 3](#_Toc305439334)

[4 Generating a Release Candidate 3](#_Toc305439338)

[4.1 Building a full GenICam or Software Module Release Candidate 3](#_Toc305439339)

[4.2 Generating a Release Candidate for Non Software Modules 3](#_Toc305439340)

[5 Collaboration Rules 3](#_Toc305439341)

[5.1 Roles and Modules 3](#_Toc305439342)

[5.2 Communication 3](#_Toc305439343)

[5.3 Reporting a Bug 3](#_Toc305439344)

[5.4 Fixing a Bug 3](#_Toc305439345)

[5.5 Maintenance Releases 3](#_Toc305439346)

[5.6 Adding a Feature 3](#_Toc305439347)

[5.7 Versioning Scheme 3](#_Toc305439348)

[5.8 Feature Releases 3](#_Toc305439349)

History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Changed by | Change |
| 1.0 | 09.01.2006 | Fritz Dierks, Basler | First Draft |
| 1.1 | 13.01.2006 | Fritz Dierks, Basler | Added the HTML Workshop |
| 1.2 | 16.01.2006 | Fritz Dierks, Basler | Added DOT and split between run-time and source |
| 1.3 | 28.04.2006 | Fritz Dierks, Basler | Added msxsl.exe as prerequisite |
| 1.4 | 07.05.2006 | Fritz Dierks, Basler | Local Environment; split solutions; xalan instead of msxsl; renamed |
| 1.5 | 15.06.2006 | Christoph Zierl, MVTec  Fritz Dierks, Basler | Added release instructions as decided on the GenICam meeting in Montreal |
| 1.6 | 30.06.2006 | Christoph Zierl, MVTec  Fritz Dierks, Basler | Review and some changes |
| 1.7 | 02.08.2006 | Christoph Zierl, MVTec | Updated required tools (Java SDK, Graphviz) and instruction in Section 3 (~SetGenICamRoot.cmd should be copied, not renamed); corrected cross references. |
| 1.8 | 18.07.2007 | Thomas Köller, Basler | Described Linux build |
| 1.9 | 31.08.2007 | Vincent Rowley, Pleora | Covered other GenICam modules. |
| 1.10 | 17.03.2008 | Natalia Weinstein, Basler | Updated installation instructions (DOT, HTML Help Workshop, PsInfo, Visual Studio additional components).  Updated the settings for running the tests from the GenApi solution.  Updated the names of solutions and files. |
| 1.11 | 24.03.2008 | Fritz Dierks, Basler | Added instructions how to run tests with different schemas |
| 2.0 | 22.09.2009 | Fritz Dierks, Basler | Changed the Build process to CMake |
| 2.1 | 15.01.2010 | Fritz Dierks, Basler | Added description how to switch to a new service pack |
| 2.2 | 09.02.2010 | Fritz Dierks, Basler | Added a comment to use the Release configuration for release builds. |
| 2.3 | 10.02.2010 | Fritz Dierks, Basler | Added VisualStudio 10.0 support |
| 2.4 | 08.03.2010 | Fritz Dierks, Basler | Added Ant |
| 2.5 | 11.02.2010 | Fritz Dierks, Basler | Made msm2msi part of the required toolchain |
| 2.6 | 26.03.2010 | Fritz Dierks, Basler | Made Cmake v2.8.1 mandatory because of OUTPUT\_DIRECTORY |
| 2.7 | 07.04.2010 | Fritz Dierks, Basler | Added security update to VS2005 requirements |
| 2.8 | 29.10.2010 | Vincent Rowley, Pleora | Updated Versioning Scheme section.  Specified that ballot should specify components that are voted on. |
| 2.9 | 01.06.2011 | Mark Jones, The MathWorks | Added Mac OS X |
| 2.10 | 10.06.2011 | Fritz Dierks, Basler | Explained how to build Xerces & Xalanc |
| 2.11 | 27.09.2011 | Thomas Hopfner, MVTec | Added Linux and revised Mac OS X |
| 2.12 | 28.10.2011 | Fritz Dierks, Basler | Added CppCheck as optional tool |
| 2.13 | 13.12.2011 | Stefan Battmer, MATRIX VISION | Added Linux ARM |
| 2.14 | 06.12.2012 | Fritz Dierks, Basler | Added VisualStudio 2012 support |

# Overview

This document describes how to build, test and release GenICam software modules. It also covers the maintenance and release of all GenICam modules.

# Building GenICam on Microsoft Windows

## Minimum Required Tools

The following tools need to be installed on your computer if you want to get the code from SVN, build it and run the tests:

* **Microsoft** **Visual C++ VS80 SP1** (= Visual Studio 2005 SP1)

Make sure to include the service packs, the x64 support and the merge module components when installing the compiler. Make sure to install the [Microsoft Visual C++ 2005 Service Pack 1 Redistributable Package ATL Security Update](http://www.microsoft.com/downloads/details.aspx?familyid=766a6af7-ec73-40ff-b072-9112bab119c2&displaylang=en). This should normally be installed silently if you have enabled the auto-update function of Windows.

Note that we support also the following compiler versions:

* **VS71 (=**Visual Studio .NET 2003)
* **VS90 SP1 (=**Visual Studio 2008 SP1).
* **VS100 SP1** (= Visual Studio 2010 SP1)
* **VS110 SP1** (= Visual Studio 2012 SP1)

However the binaries are not distributed.

* **TortoiseSVN** version 1.6.5. or higher. This program is free and can be downloaded from [tortoisesvn.tigris.org](http://tortoisesvn.tigris.org/).
* **CMake** version 2.8.10 or higher. This program is free and can be downloaded from [www.cmake.org](http://www.cmake.org). The installer is asking to add CMake to the PATH variable. You need to do this.
* **Msm2msi** version 1.0 or higher. This program is free and can be downloaded from [www.ethalone.com/download/msm2msi.zip](http://www.ethalone.com/download/msm2msi.zip). You need to put the tool into a directory which is referenced in your PATH variable.

## Tools Required for Language Bindings

* **SWIG** version 2.0.11 a tool making the C++ code of GenICam available to Python. It can be downloaded from <http://www.swig.org>.
* **Python** version 2.7 or later. In order to build the bridge to version 2 of this programming language download from <http://www.python.org/download/>

## Tools Required for Building Setup

* **NSIS** version 2.44 or higher. This program is free and can be downloaded from [nsis.sourceforge.net/Download](http://nsis.sourceforge.net/Download) .   
  **Patch for large strings** has to be applied. Download from the NSIS Special build section  [large strings build](http://prdownloads.sourceforge.net/nsis/nsis-2.46-strlen_8192.zip?download) .

## Tools Required for Building Documentation

* **Doxygen** version 1.6.1 or higher. This program is free and can be downloaded from [www.stack.nl/~dimitri/doxygen](http://www.stack.nl/~dimitri/doxygen). You need to add the *bin* folder of the installation directory (e.g. “C:\Programme\doxygen\bin”) to your PATH variable.
* **Microsoft HTML Help Workshop**. This program is free and can be downloaded from [download.microsoft.com](http://download.microsoft.com/) (search for “Microsoft HTML Help Workshop”). You need to add the installation directory (e.g. “C:\Programme\HTML Help Workshop”) to your PATH variable.
* **GraphViz** v2.2 or higher. This program is free and can be downloaded from [[www.graphviz.org](http://www.graphviz.org/).](http://www.boostpro.com/download).
* **Highlight** v3.1 or higher. This program is free and can be downloaded from [www.andre-simon.de/](http://www.andre-simon.de/)

## Optional Tools

The following tools need to be installed on your computer if you want use all provided command line utilities like, e.g. ZipThisDriectory.cmd:

* **Cygwin** version 1.5 or higher. This program is free and can be downloaded from [www.cygwin.com](http://www.cygwin.com). You need to add the *bin* folder of the installation directory (e.g. “C:\Programme\cygwin\bin”) to your PATH variable. You need to add it to the front of the list of directories because otherwise Cygwin’s find tool is masked by Window’s find tool.
* **7-Zip** version 1.5 or higher. This program is free and can be downloaded from [www.7-zip.org.](http://www.7-zip.org/) You need to add the *bin* folder of the installation directory (e.g. “C:\Programme\7-Zip”) to your PATH variable.
* **SlikSVN** version 1.6.5 or higher. This program is free and can be downloaded from [www.sliksvn.com](http://www.sliksvn.com).
* **Ant** version 1.7 or higher. This program is free and can be downloaded from [ant.apache.org](http://ant.apache.org/).
* **CppCheck** version 1.5.1 or higher. This program is free and can be downloaded from [cppcheck.sourceforge.net](http://cppcheck.sourceforge.net)

## Building and Testing Manually

Take the following steps to build and test GenICam manually.

1. Extract the whole GenICam file tree from the SNV repository to some arbitrary location
   1. Use the explorer to create a new folder, e.g. **C:\Projects\genicam.** Avoid locations with a blank in the path name, e.g. “c:\my home”.
   2. In the context menu of that folder choose .



* 1. Fill in the URL of the trunk and the directory just created



* 1. After clicking the ok button you are asked for authentication. Use your login data for the Trac system.



Note : you may need to add your proxy settings to TortoiseSVN’s settings.

1. In order to build the Win64 version you MUST first build the Win32 version because the Win64 cross tool chain uses the Win32 version of the GenApiPreProcessor. Start VisualStudio and load the GenICam solution by invoking StartStudio.cmd. This batch file takes three parameters:

**StartStudio** *Studio* [*Module*] [*Platform*]

with

**Studio =** *VS110, VS100, VS90, VS80, VC80express, VS71* **Module =** *GenICam, GenApi, CLParams* **Platform** *= Win32, Win64*

Examples:  
~ StartStudio VS90 starts Visual Studio 2009  
~ StartStudio VS90 GenICam starts Visual Studio 2009 and loads   
 GenICam.sln for Win32  
~ StartStudio VS90 GenICam Win64 starts Visual Studio 2009 and loads   
 GenICam.sln for Win64

1. Build the solution  
     
   In order to build the Debug version you MUST first build the Release version because the Debug build uses the Release version of the GenApiPreprocessor.

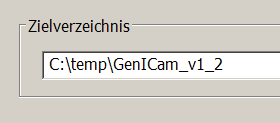
**BEWARE**: For release builds you MUST use the *Release* configuration (and not *MinSizeRel* or *RelWithDebInfo*).

1. Configure the **Debugging Properties** of the GenApiTest project

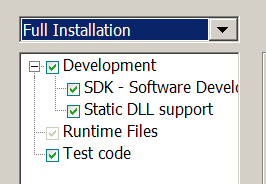
Command: DllPluginTester\_MDd\_VC90\_v1\_2.exe   
Command Arguments : -w $(TargetPath)

Note: this example assumes you are using VisualStudio 2008 and are in Debug mode.

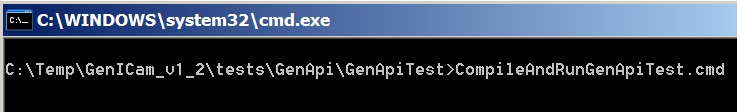
1. Press F5 to run the tests.
2. Building the installer required the following additional steps:
   1. Build the Release configuration as described above
   2. Build the project PACKAGE manually
   3. The installer is found in the subdirectory described by the configuration you are building, e.g. Win32\_i86\_VS90. Its name looks like GenICam\_VC90\_Win32\_i86\_v1\_2.exe
3. Testing the installer
   1. Start the installer
   2. Choose an arbitrary location for installation, e.g.



* 1. Select the configuration “Full Installation”



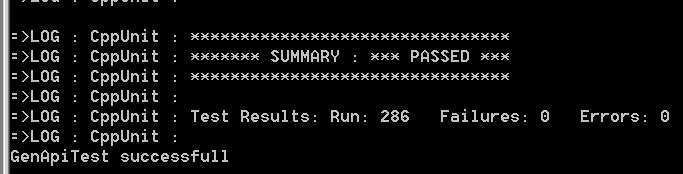
* 1. Install the software. Note that the installer will add some global environment variables to your system, such as GENICAM\_ROOT\_V2\_3 and add the binary directory to your PATH variable.
  2. Open a command shell in the GenApiTest subdirectory
  3. For Win32 run the batch file “CompileAndRunGenApiTest.cmd”.



This file requires CMake to be installed. It will compile the local GenApiTest code relying on the environment variables set and the run the test code. Note that IncrediBuild will be used if it is installed.

For Win64 run the batch file “RunGenApiTest.cmd”. This file only runs the tests and thus does not need a compiler of CMake to be present.

* 1. Make sure the test are run successfully



* 1. Run the Uninstaller which is located in the root of the installation folder



This will delete the environment variables set during installation and remove GenICam from the PATH.

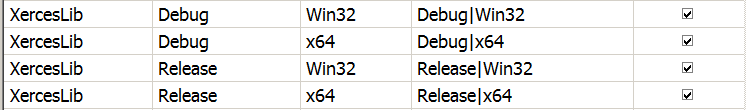
* 1. Delete the installation folder manually. This is required because the installer does only delete files it has installed himself. However compiling the test code creates more files which are removed in this step.

## Supporting a New VisualStudio ServicePack

There are several places in the SVN repository were binaries are checked in which depend on the version of the VisualStudio runtime. If a new service pack is installed these binaries have to be re-compiled and updated in the repository.

**Xalanc/Xerces**: These dlls are based on code which is located in $(GENICAM\_ROOT)/3rdparty in the files xerces-c-src\_2\_7\_0\_GenICam.zip and xml-xalan\_1\_10\_GenICam.zip. The following build steps assume VC100. For older compilers slight modifications might be necessary.

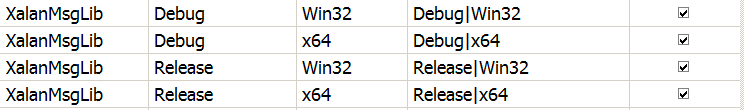
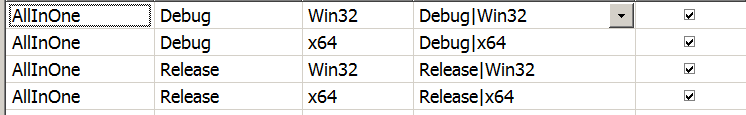
* Unzip these files to an arbitrary location, e.g.   
  *C:\projects\xerces-c-src\_2\_7\_0\_GenICam* and  
  *C:\projects\xml-xalan\_1\_10\_GenICam*
* First build Xerces. Open the solution   
  *C:\projects\xerces-c-src\_2\_7\_0\_GenICam\Projects\Win32\VC100\xerces-all\xerces-all.sln*
* Then build the following configurations



* Call the following batch file   
  *C:\projects\xerces-c-src\_2\_7\_0\_GenICam\Projects\Win32\VC100\CopyFiles.cmd*  
  which will copy all DLLs, LIBs, and PDBs created to the GenICam tree which is assumed to be located at *C:\projects\GenICam* (you man need to adapt the paths in the batch file)
* Next build Xalanc.
* First set the environment variable



* Open the solution  
  *C:\projects\xml-xalan\_1\_10\_GenICam\c\Projects\Win32\VC10\Xalan.sln*
* Build the following configurations



* Call the following batch file   
  *C:\projects\xml-xalan\_1\_10\_GenICam\c\Projects\Win32\VC10\CopyFiles.cmd*which will copy all DLLs, LIBs, and PDBs created to the GenICam tree which is assumed to be located at *C:\projects\GenICam* (you man need to adapt the paths in the batch file)
* If you have made any changes to the Xalan or Xerces sources zip clean the solutions (to restrict the size of the zip files), zip the root directories of Xalan and Xerces and copy the zip files to the 3rdparty directory.

## Use the GenICam NSIS Installer

For normal installation just invoke the installer exe, e.g. GenICam\_VC80\_Win32\_i86\_v2\_3.exe and follow the instructions.

For silent install call: *GenICam\_VC80\_Win32\_i86\_v2\_3.exe /S*

For installation into a specific directory call: *GenICam\_VC80\_Win32\_i86\_v2\_3.exe /D c:\temp*

For installing with test code call: *GenICam\_VC80\_Win32\_i86\_v2\_3.exe /SELECT\_TESTS*

# Building GenICam on Linux and Mac OS X

The description how to build the Linux and Mac OS X for versions before v2.3 can be found in the [GenICam Wiki](https://genicam.mvtec.com/trac/genicam/wiki/GenApi/GettingStarted). Starting with version v2.3 it is documented here.

## Minimum Required Tools

The following tools need to be installed on your computer if you want to get the code from SVN, build it and run the tests:

* **GNU gcc-4.x** and necessary development tools like binutils**.**
* **Subversion** version 1.5 or higher.
* **CMake** version 2.6 or higher, 2.8.x is recommended.

These tools should be available in the software repositories of your system.

## Getting the sources

Note: You will need to use a text shell to input the following commands.

First you need to download the sources of the version you want to build. As the committee decided to use the operating systems native line endings for SVN checkout you can **not** simply **use the Windows version of the sources**! Using the Linux/Mac OS X sources on a Windows system is ok.

In the Subversion (SVN) repository the available versions are organized as usual, i.e. below [*https://genicam.mvtec.com/svn/genicam/*](https://genicam.mvtec.com/svn/genicam/) exist three directories: *branches*, *tags* and *trunk*. Usually only the *tags* directory is relevant as it contains released versions. You can see the available versions with a command like:

*svn list <https://genicam.mvtec.com/svn/genicam/tags/>*

After this you can check out a copy of the specific version to your local disk by using:

*svn co* [*https://genicam.mvtec.com/svn/genicam/tags/V2\_3\_0*](https://genicam.mvtec.com/svn/genicam/tags/V2_3_0)

This will create a new directory named V2\_3\_0 in your current working directory.

## Building and Testing

Note: You will need to use a text shell to input the following commands. In the following the 64 bit Linux version is used, please adapt this to your needs. Valid targets are ‘Linux32\_i86’, ‘Linux64\_x64’, ‘Linux32\_ARM’ and ‘Maci64\_x64’. The ‘Linux32\_ARM’ version uses a minimal subset of compiler flags and options that will allow usage of the resulting binaries on architectures starting from ARMv5(thus ARM9, ARM11 and Cortex-A are supported).

Change to the directory which contains the GenICam sources (see above) and then run a script to create a build directory and run CMake to generate the appropriate files and directories:

*./RunCMake.sh Linux64\_x64*

Now run the build script to generate the GenApi libraries:

*./BuildUnix.sh Linux64\_x64*

And finally run the tests:

*./run\_tests.sh Linux64\_x64*

Be prepared to get a very verbose output with lots of “check manually”. The output depends on the setting of the *$GENICAM\_LOG\_CONFIG\_Vx\_y* variable which usually points to *log/config-unix/TestLogging.properties.* Finally you should see something like:

*[…]*

*=>LOG : CppUnit : \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*=>LOG : CppUnit : \*\*\*\*\*\*\* SUMMARY : \*\*\* PASSED \*\*\**

*=>LOG : CppUnit : \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*=>LOG : CppUnit :*

*=>LOG : CppUnit : Test Results: Run: 362 Failures: 0 Errors: 0*

*=>LOG : CppUnit :*

## Creating the packages for distribution

Note: You will need to use a text shell to input the following commands. In the following the 64 bit Linux version is used, please adapt this to your needs.

Run the packaging script for your architecture:

*./pack\_unix.sh Linux64\_x64*

This will create three .tgz files which contain the runtime, the SDK and the tests.

# Generating a Release Candidate

## Building a full GenICam or Software Module Release Candidate

This section describes how to build a formal release candidate for a single or all GenICam software modules.

🡺 **Important**: This procedure is normally only accomplished by the responsible module maintainer (see section 5.1).

1. Add a symbolic tag regarding the RC, i.e. RC1 for the first release candidate.
2. Build GenApi for Windows (VS80) and Linux (gcc40) each for 32 and 64 bit and also for Linux ARM 32 bit and Mac OSX 64 bit. The result will be two installer files for Windows, nine .tgz files for linux(3 for each platform) and three .tgz files for Mac OSX. The Windows installers contain the Runtime, SDK and Tests in one file, for Linux and Mac OSX they are distributed in separate files. For an official release the Windows versions are built by Basler, the Linux32\_i86 and Linux64\_x64 versions are built by MVTec, the Linux32\_ARM version is built by MATRIX VISION and the Mac OSX version is built by MathWorks.
3. Publish the setup files to the GenICam working group for testing and acceptance (for details see section 5).
4. If the release candidate is accepted by the GenICam working group perform the following final steps:
   1. Add the final tag to the repository.
   2. Publish the result as described in section 6.
5. If the release candidate is not accepted, fix the problem and build a new release candidate repeating the whole procedure using the label and tag RCx where x is incremented for each release candidate by one.

## Generating a Release Candidate for Non Software Modules

Release candidates for non software GenICam modules, such as the SFNC module, are simply generated by tagging the appropriate files in the SVN repository.

🡺 **Important**: This procedure is normally only accomplished by the responsible module maintainer (see section 5.1).

For instance, the procedure to create a release candidate for the SFNC module is:

1. Add a symbolic tag regarding the RC, i.e. RC1 for the first release candidate.
2. Zip the SFNC Microsoft Word document to a file named GenICam Standard Features Naming Convention 1.5.1\_RC1.zip .
3. Publish the zip file to the GenICam working group for acceptance (for details see section 5).
4. If the release candidate is accepted by the GenICam working group perform the following final steps:
5. Add the final tag to the SVN
6. Publish the results as described in section 6.
7. If the release candidate is not accepted fix the problem and generate a new release candidate repeating the whole procedure using the label and tag RCx where x is incremented for each release candidate by one.
8. When the SFNC release candidate is accepted, follow the detailed procedure described in the “SFNC release checklist and documents location.docx” document that is located in the “\*SVN GenICam*\doc\Standard\_Feature\_List” repository.

# Collaboration Rules

## Roles and Modules

This document refers to several **roles** which are defined in this section

The **GenICam working group** consists of all associated and contributing members of the GenICam standard committee as listed on www.genicam.org.

The **maintainer** of a GenICam **module** is responsible for the integrity of this component. Normally only the maintainer may change the component. Any proposed changes (bug fixes, added features, other) must be sent to the maintainer who will include them in the component.

The following GenICam modules are currently set up:

* **GenApi Software Module** – reference implementation of code which is responsible for configuring a device
* **SFNC Module** – recommended names and types for common features
* **GenTL Software Module** – code responsible for transferring data between a device and a host
* **Validator Software Module** – code checking a camera configuration file for standard compliance
* **SVN Admin Module** – the data base holding the GenICam source code and documents, see https://genicam.mvtec.com/svn/genicam/
* **Trac Admin Module** – the wiki and ticket tracking system, see https://genicam.mvtec.com/
* **MailingList Admin Module** – the central communication tool for the GenICam working group
* **WebSite Admin Module** – the GenICam web site on http://[www.genicam.org](http://www.genicam.org)
* **FTP** **Admin Module** – the GenICam FTP server on ftp://[ftp.genicam.org](ftp://ftp.genicam.org) (tbd => create)
* **Releasing Admin Module –** preparing the official release version

A list of modules and their current maintainers can be found on XXX (tbd => SVN)

## Communication

Communication is accomplished via email. The following email addresses and aliases exist:

* [GenICam@imaging.de](mailto:GenICam@imaging.de) - the GenICam mailing list
* [GenApi@genicam.org](mailto:GenApi@genicam.org) - an email alias for the GenApi module’s maintainer and his/her deputies.
* [GenTL@genicam.org](mailto:GenTL@genicam.org) - an email alias for the GenTL module’s maintainer and his/her deputies.
* [Validator@genicam.org](mailto:Validator@genicam.org) - an email alias for the Validator module’s maintainer and his/her deputies.
* [CVS@genicam.org](mailto:CVS@genicam.org) - an email alias for the CVS maintainer and his/her deputies.
* [Mantis@genicam.org](mailto:Mantis@genicam.org) - an email alias for the Mantis maintainer and his/her deputies.
* [MailingList@genicam.org](mailto:MailingList@genicam.org) - an email alias for the mailing list maintainer and his/her deputies.
* [WebSite@genicam.org](mailto:WebSite@genicam.org) - an email alias for the web site maintainer and his/her deputies.
* [FTP@genicam.org](mailto:FTP@genicam.org) - an email alias for the FTP server maintainer and his/her deputies.
* [Releasing@genicam.org](mailto:Releasing@genicam.org) - an email alias for the FTP server maintainer and his/her deputies.
* [GenICam-contact@<Mycompany>.com](mailto:GenICam-contact@%3cMycompany%3e.com) - a local email alias of each company giving access to the GenICam related people at this company. All contributing members of the GenICam working group should provide this email alias; associated members might provide it, too.

The GenICam web site is maintained at http://[www.genicam.org](http://www.genicam.org)

A FTP site is available under ftp://[ftp.genicam.org](ftp://ftp.genicam.org)

During discussions on the mailing list the subject in the email header should not be changed in order to allow grouping the mails by thread.

## Reporting a Bug

Bugs are reported through the Trac ticket system and a copy is sent on the GenICam mailing list.

Any bug report must come with a description and a CPPUnit test case, when applicable, which reproduces the bug. The H/CPP files establishing the test case must be appended to the bug report.

The bug reporter is encouraged to provide a bug fix together with the bug report.

By choosing the correct component the bug report (=ticket) is automatically assigned to the module’s maintainer which should try to reproduce the problem within 1 week.

## Fixing a Bug

The following steps should be followed for fixing a bug in GenICam:

1. If the bug reporter submits a bug fix the maintainer of the affected module will review, possibly adapt and include the fix into the SVN and create an entry in the release notes. The maintainer may reject the implementation if it breaks the module integrity or architecture in major ways or does not follow the implementation rules such as the coding policy (see $(GENICAM\_ROOT)/doc/Coding\_Policy.doc.) in the case of a software module.
2. In case no bug fix is provided the maintainer will attempt to fix the bug in reasonable time himself.
3. The bug should be fixed in the trunk as well as all affected branches of the SVN.
4. After the fix is added to the SVN the maintainer sends out an email to the mailing list with subject “<ModuleName> BUG\_FIXED : < bug name>” including a copy of the release notes. ModuleName is the name of the GenICam module in which a bug was fixed.
5. The members of the working group then get a fresh copy from the affected SVN branch(es) and review the bug fix. They have normally 2 weeks to object against the changes.
6. If no one objects within 2 weeks the maintainer sends out an email with subject “<ModuleName> BUG\_FIX\_ACCEPTED : < bug name>” to indicate the end of the process.

## Maintenance Releases

1. A maintenance release can be requested by any member of the working group if he requires the current bug fixes to be published. The request is sent to one of the maintainers with a Cc on the mailing list and subject “<ProjectName> MAINTENANCE\_RELEASE\_REQUEST” where ProjectName is GenICam or the name of the GenICam module for which the maintenance release is requested.
2. The maintainer creates usually within 1 week a release candidate as described in section 4.1 if the maintenance involves a software module. Otherwise, a release candidate is generated as per Section 4.2.
3. He puts the release candidate on the FTP server and sends out an email with subject “<ProjectName> MAINTENANCE\_RELEASE\_CANDIDATE\_AVAILABLE”
4. The contributing members of the working group then review the release candidate. They have normally 2 weeks for accepting it.
5. If no one objects within 2 weeks the maintainer sends out an email with subject “<ProjectName> MAINTENANCE\_RELEASE\_CANDIDATE\_ACCEPTED” to indicate the end of the process.
6. The accepted release candidate is made a release and published on the GenICam web site.
7. In urgent cases the requester of the maintenance request can create a private build and provide it to customers affected by the bug. The private build must be clearly treated as such and made accessible with source code on the FTP server. The private build must only be used until the maintenance release is available. It is strongly recommended to replace the private build with the maintenance release at the affected customers.

## Adding a Feature

Features are added to the trunk only.

The following steps should be followed for adding a new feature to any GenICam module:

1. The requester should send an email with subject “<ModuleName> FEATURE\_REQUEST : <feature name>” with a brief explanation of the feature he wants to add, how it should be done, why, and which modules would be affected. ModuleName is the name of the GenICam module for which a new feature is requested.
2. The feature request should be discussed on the mailing list and typically either accepted or rejected by the contributing members of the last meeting within 1 week. The maintainer of the module can decide to extend the discussion period to more than one week based on the nature of the feature to be added.
3. If the community agrees that the feature should be added the requester should implement the feature on a local copy of the latest version on the trunk of the GenICam SVN repository. The implementation must contain test cases achieving 100% function and condition/decision coverage, when applicable, as well as complete relevant documentation.
4. The requester should send the implemented feature to the maintainer(s) of the affected module(s) which will review, possibly adapt and commit the feature into the trunk of the SVN and create an entry in the release notes. The maintainer may reject the implementation if it breaks the module integrity or architecture in major ways or does not follow the implementation rules such as the coding policy (see $(GENICAM\_ROOT)/doc/Coding\_Policy.doc) for software modules.
5. After the new feature is added to the SVN the maintainer sends out an email to the mailing list with subject “<ModuleName> FEATURE\_ADDED : <feature name>” including a copy of the release notes.
6. The members of the working group then get a fresh copy from the SVN and review the new feature. They have normally 2 weeks for objecting against the changes.
7. If no one objects within 2 weeks the maintainer sends out an email with subject “<ModuleNane> FEATURE\_ACCEPTED : <feature name>” to indicate the end of the process.

## Versioning Scheme

The versioning of a GenICam module follows the following scheme:

<Major>.<Minor>.<Sub-Minor>

An example would be “1.1.2”.

The Major number is increased when there are significant changes in functionality, the Minor number is incremented when only minor features have been added and the Sub-Minor number is incremented when minor issues are fixed.

Therefore, a major release may introduce a break in backward compatibility but it is not necessarily the case. For instance, the Major number can be increased to indicate the achievement of a significant milestone that does not necessarily introduce a break in backward compatibility. On the other hand, a minor or maintenance release must maintain backward compatibility. If the module is a software module, or includes one, then binary compatibility must be maintained for a maintenance release. On the other hand, source compatibility must at least be maintained for the minor release of a module including a software one.

## Feature Releases

A feature release is run along the following steps:

1. A feature release can be requested by any member of the working group. The request is sent to one of the maintainers with a cc on the mailing list and subject “<ProjectName> FEATURE \_RELEASE\_REQUEST” which a brief explanation why a feature release would be necessary. ProjectName is GenICam or the name of the GenICam module for which the feature release is requested.
2. The feature release request should be discussed on the mailing list and typically either accepted or rejected within 2 weeks.
3. If the community agrees the maintainer creates a release candidate as described in section 4.1 if the maintenance involves a software module. Otherwise, a release candidate is generated as per Section 4.2..
4. He puts the release candidate on the FTP server and sends out an email with subject “<ProjectName> RELEASE\_CANDIDATE\_AVAILABLE”
5. The contributing members of the working group then review the release candidate. They have normally 2 weeks for accepting it.
6. If no one objects within 2 weeks the maintainer sends out an email with subject “<ProjectName> RELEASE\_CANDIDATE\_ACCEPTED” to indicate the end of the process.
7. If the release candidate is accepted technically it must be formally voted for by the contributing members. The ballot must clearly describe the module(s) and version number(s) that are voted on. When voting on a GenICam release, the version of the release must be specified as well as the related sub-modules and their versions. Normally this voting should be take place during a meeting where the release is done. However a phone conference is also acceptable which should be organized by the maintainer.
8. If the release is formally accepted it is made available on the GenICam web site.