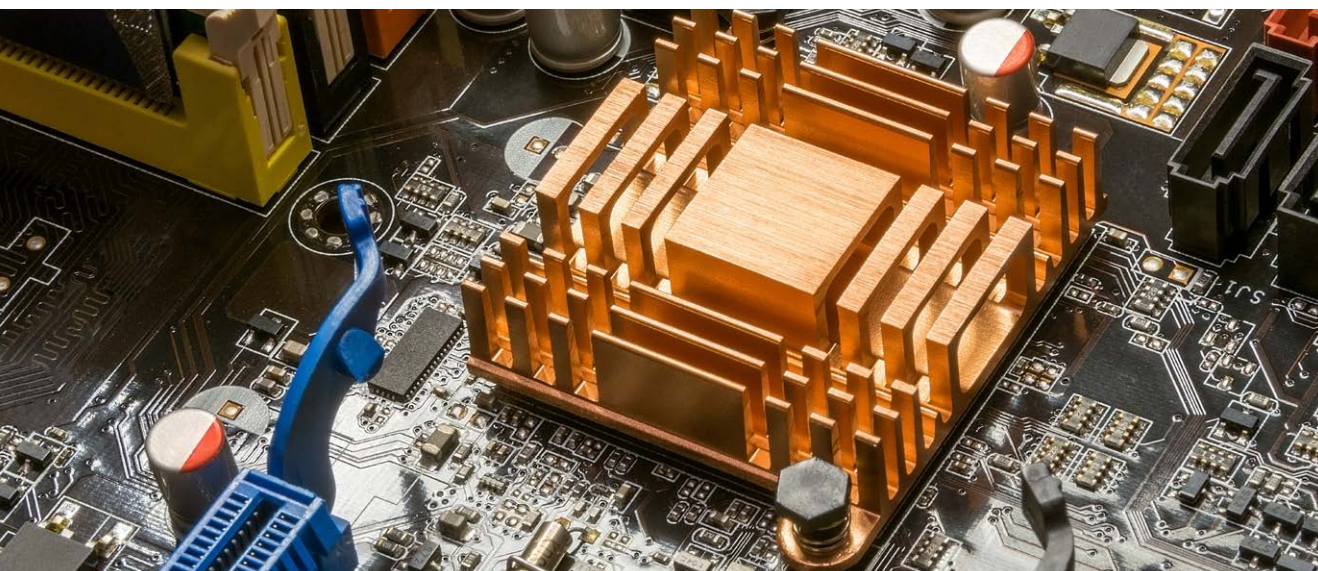


Middleware Gamma VI

Systemvoraussetzungen



www.proway.de



Systemvoraussetzung Gamma VI

Hardware

Die Mindestanforderungen an die Hardware, um Gamma unter Windows oder Ubuntu Linux zu betreiben, sind die gleichen, wie bei diesem Betriebssystem. Ansonsten kommt es auf den jeweiligen Anwendungsfall an. Zum Beispiel:

- ▶ Gamma kann ohne UI auf einem Ubuntu-Linux-System ohne Kopfhörer ausgeführt werden.
- ▶ Gamma kann auf einem vollwertigen Entwicklungssystem mit 32 GiB, 32 CPU-Kernen, etc. ausgeführt werden.
- ▶ Gamma kann auf einem Dual-Xeon-Backend-System für kritische Echtzeitanwendungen ausgeführt werden.

Software

Gamma basiert auf bewährten, industrietauglichen Komponenten, um ein qualitativ hochwertiges Softwarepaket bereitzustellen. Je nach Betriebssystem werden diese Abhängigkeiten automatisch (Ubuntu Linux Paketierungssystem) oder manuell (Windows) installiert.

Weitere Informationen zur Einrichtung des Systems, z. B. zur Erstellung benutzerdefinierte Anwendungen, finden Sie im Gamma Handbuch im Kapitel Toolchains.

Installation

Gamma wird mit Installationsprogrammen für Folgendes ausgeliefert:

- ▶ Ubuntu LTS Linux (x86_64)
- ▶ Windows 10 (x86_64 - MSVC 2022)

Diese Installationsprogramme/Installer werden passend zu den einzelnen Installationssystemen der Plattformen erstellt. Unter Linux werden deb-basierte Archive verwendet. Unter Windows werden exe-Installer bereitgestellt.

Alle Installer zielen auf die neueste Version der oben genannten Systeme ab. Sollte eine neue Version von z. B. Ubuntu LTS veröffentlicht werden, wird die alte Version noch bis zu sechs Monate unterstützt.

Dieser Zeitraum ermöglicht eine rechtzeitige Migration auf das neueste System. Nach dem angegebenen Migrationszeitraum wird nur noch die neueste Version offiziell unterstützt.

Ubuntu LTS Linux

Auf Ubuntu LTS Linux wird Gamma als reguläres .deb-Paket ausgeliefert:

- ▶ `gamma_<Version>_amd64.deb`: Basispaket (Laufzeit).
- ▶ `python3-gamma_<Version>_amd64.deb`: Basis-/Entwicklungspaket für Python.
- ▶ `gamma-dev_<Version>_amd64.deb`: Entwicklungsdateien (ohne Dokumentation).
- ▶ `gamma-dev-doc_<Version>_amd64.deb`: Entwicklungsdokumentation (API-Dokumentation, Demos, Handbücher, etc.).
- ▶ `gamma-gui_<Version>_amd64.deb`: Grafische Benutzeroberflächen (gaExplorer, etc.)
- ▶ `gamma-qt6-*<Version>_amd64.deb`: Viele qt6-spezifischer Pakete.
- ▶ `gamma-testframework-dev_<Version>_amd64.deb`, `gamma-util-cpp-dev_<Version>_amd64.deb`, `gamma-xercestools-dev_<Version>_amd64.deb`: Vorausgesetzte Pakete für die Entwicklung.

Bitte installieren Sie die Pakete als root, z. B.: `sudo apt install gamma_<Version>_amd64.deb`

Der Standardpfad unter Linux ist: `/usr/share/gamma`

Windows 10

Um Gamma unter Windows zu installieren, führen Sie die Installer-Binärdateien aus, die wie folgt benannt werden sollten:

- ▶ `gamma-<Version>-msvc_2022-x64.exe`: Basispaket (Laufzeit).
- ▶ `gamma-devel-<Version>-msvc_2022-x64.exe`: Entwicklungspaket (APIs, Dokumentation, Dämonen, etc.).

Wenn Sie nur bestehende Software ausführen, reicht es aus, nur das Basispaket zu installieren.

Für die Entwicklung von Software ist es erforderlich, das `devel`-Paket / Entwicklungspaket zu installieren. Um das System richtig einzurichten, ist es erforderlich, Gamma in einem Pfad ohne Leerzeichen zu installieren (ähnlich den Installationsanforderungen von Qt!).

Wenn Sie dies nicht tun, führt dies zu seltsamen Fehlern, z. B. wird CMake nicht in der Lage sein, einen `CMAKE_MODULE_PATH` mit einem Leerzeichen im Pfad zu verarbeiten, wenn es von der `MSYS2-Bash` aus gestartet wird. Die einzig bekannte Workaround ist die Verwendung von `cmd.exe`.

Das Installationsprogramm unter Windows verwendet den folgenden Standardpfad: `C:\Gamma`

Falls auf dem System keine Entwicklung stattfindet, ist es in Ordnung, das Paket an einem beliebigen Ort zu installieren.

Diese Änderung betrifft alle Windows-Gamma-Installationsprogramme > 6.2.2.

Warnung

Es ist erforderlich, nach der Installation einen Neustart durchzuführen. Andernfalls meldet CMake Fehler beim Erstellen der Beispiele, da die neu installierten DLLs nicht gefunden werden können.

Hinweis

Aus praktischen Gründen enthält das Gamma-Installationsprogramm bereits die MSVC 2022 Laufzeit-DLLs. Daher ist es nicht erforderlich, das entsprechende "Microsoft Visual C++ Redistributable (x64)" SW-Paket zu installieren.

Voraussetzung ist hingegen, dass "Microsoft Visual C++ 2010 x64 Redistributable" installiert ist. Dieses Paket wird jedoch höchstwahrscheinlich ohnehin auf dem Windows 10-System installiert, da es von verschiedenen anderen SW-Paketen benötigt wird.

Toolchains / Werkzeugketten¶

Ubuntu LTS Linux¶

Die Linux-SW wird mit der vom Betriebssystem bereitgestellten Standard-Toolchain (GCC) erstellt. Daher ist es möglich, cmake sofort / out-of-the-Box auszuführen. Außerdem werden Systempakete beim Linken gegen Upstream-Pakete wie Qt, Boost usw. verwendet.

Windows 10¶

Unter Windows wird die SW mit einer Kombination aus verschiedenen Toolchains wie folgt erstellt:

- ▶ MSVC 2022
- ▶ Qt 6.2.4 Entwicklungsdateien (msvc2022_64 - LTS) (benötigt für gamma-devel-qt6-<Version>-msvc_2022-x64.exe)
- ▶ CMake 3.24.2
- ▶ Boost 1.79.0 MSVC 14.2 x64
- ▶ Qt Jom
 - Python 2 (retrofit/legacy) und 3 erfordert die folgenden Upstream-Bibliotheken:
 - future
 - pywin32

Installieren Sie sie einfach mit pip, wie im folgenden Beispiel, das Python 3 verwendet, angegeben:

```
py -3 -m pip install future pywin32
```

Für Python 2 ersetzen Sie einfach '-3' durch '-2'. Für weitere Informationen lesen Sie bitte die pip-Dokumentation.

Daher ist es erforderlich, das System für die Kompilierung ordnungsgemäß einzurichten. Dies umfasst:

- ▶ Die PATH-Variable muss enthalten:
- ▶ den Pfad zu CMake (z.B. C:\Qt\Tools\CMake_64\bin)
- ▶ die MSVC-Toolchain (verwenden Sie vcvars64.bat - siehe MS-Hinweis zur Konsolenkompilierung)

Darüber hinaus wird empfohlen, die gleichen Versionen der Upstream-Pakete zu verwenden, wie z. B. Qt/Boost, um Verknüpfungsfehler zu vermeiden.

Hinweis

Das Windows-Installationsprogramm trägt den Namen der verwendeten Toolchain im Namen (z. B. gamma-x.y.z-msvc_2022-x64.exe), um dies explizit zu verdeutlichen und spätere, schwer nachvollziehbare Konflikte zu vermeiden.

Hinweis

Die API-Pakete verwenden den folgenden Mindeststandard:

- ▶ C API: verwendet/erfordert mindestens C11
- ▶ C++ API: verwendet/erfordert mindestens C++17
- ▶ Python API: verwendet/erfordert Python 2 und 3

Hinweis

Das Boost-Paket wird für bestimmte C++-Bibliotheken (z. B. gaWs) benötigt. In diesem Fall wird ein vorgefertigtes Paket verwendet, das von boost.org selbst bereitgestellt wird.

Zur Installation laden Sie einfach die Datei [boost_1_79_0-msvc-14.2-64.exe](#) herunter.

Für weitere Informationen besuchen Sie bitte <https://www.boost.org/users/download/>

Hinweis

Das Werkzeug jom wird als Ersatz für GNU make verwendet. Zur Installation laden Sie die Datei [jom.zip](#) herunter und legen Sie sie in einem Ordner ab, der Teil der Umgebungsvariablen PATH ist! Weitere Informationen finden Sie unter: https://wiki.qt.io/Jom#Download_the_binary_package.

Wichtig

Derzeit werden die Gamma Low-Level/High-Performance-APIs über die Sprachen C und C++ bereitgestellt, wobei der Schwerpunkt auf letzterer liegt. Neue APIs (z.B. gaServiceIo2) werden nur über C++ bereitgestellt. Darüber hinaus können neuere Funktionen in den bestehenden APIs nur in der C++-Version hinzugefügt werden.

Weitere Informationen finden Sie in unserem Gamma-Handbuch.

Kontaktieren Sie uns

Gerne beraten wir Sie bei einem persönlichen Termin und erarbeiten mit Ihnen Lösungen für Ihre spezifischen Anforderungen. Sie liefern die Herausforderung – wir die Lösung!



Bolzstraße 3
70173 Stuttgart



+49 (0) 711 25 36 05 - 0



Ingeborg-Bachmann-Straße 71
89134 Ulm-Blaustein



+49 (0) 731 55 01 98 - 0



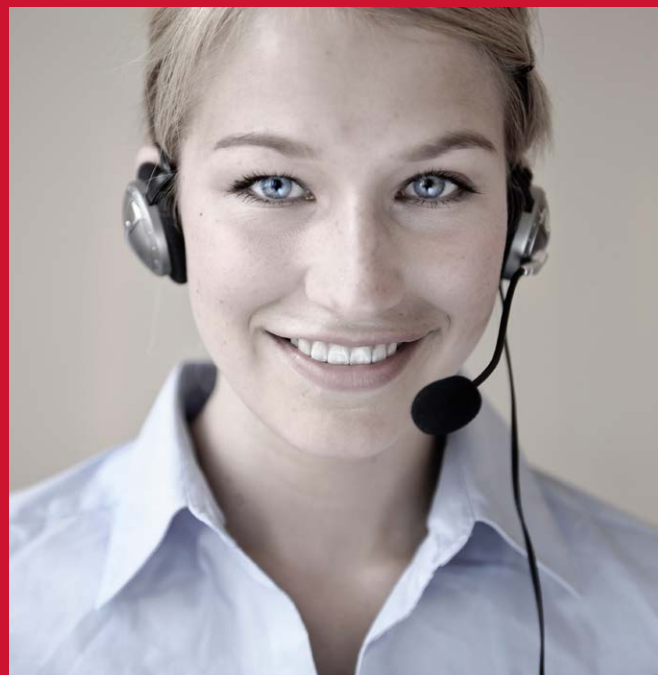
Carl-Zeiss-Straße 51
85521 Hohenbrunn



+49 (0) 89 961 601 8-00



sales@proway.de



www.proway.de

© Proway GmbH

PROWAY