



Eclipse Webinar – 4diac



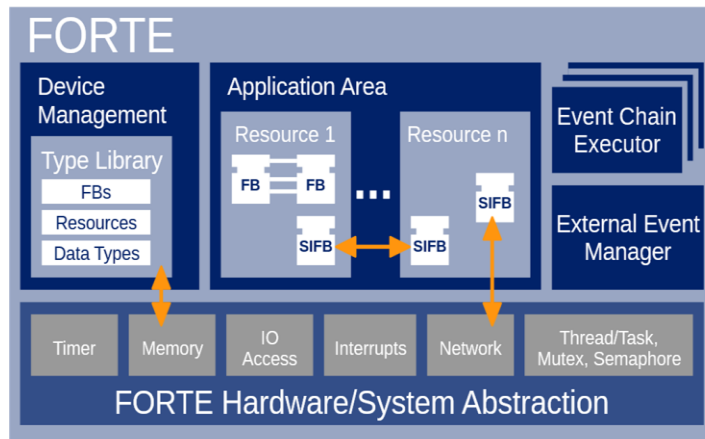
Dr. Michael Oberlehner

LIT | Cyber-Physical Systems Lab
Johannes Kepler University Linz



Introduction

- <https://eclipse.dev/4diac/download/>
- <https://download.eclipse.org/4diac/updates/rc/>
- Download Release Candidate
- 4diac 3.2.x will be release June 2026



Download Eclipse 4diac™

Latest Release: Eclipse 4diac 3.1

4diac IDE 

Development environment without 4diac FORTE executables:

- Linux 64-bit
- macOS Apple Silicon
- macOS Intel
- Windows 64-bit

4diac FORTE 

Packaged source code and selected binaries of 4diac FORTE:

- Source
- Prebuilt versions coming soon!

4diac FBE 

A versatile and easy to use build environment for 4diac FORTE:

- 4diac FBE Package

The components of Eclipse 4diac are provided under Eclipse Public License, Version 2.0!

First Step – Fork Repo

<https://gitlab.com/mjger/eclipsewebinar>

The screenshot shows the GitLab interface for the repository 'EclipseWebinar'. At the top, the repository name is displayed with a globe icon. Below it, the current branch is 'main' and the repository path is 'eclipsewebinar'. A recent commit is shown: 'updated to 2.0.0 [DEPLOY]' by Michael Oberlehner, 42 minutes ago. A table lists the repository files: 'Robot', '.gitlab-ci.yml', and 'README.md'. The 'README.md' file is selected, showing its content: '4diac Library Packaging with GitLab CI'. On the right side, the 'Project information' section is visible, including '4 Commits', '1 Branch', '0 Tags', and '116 KiB Project Storage'. The 'Fork' button is highlighted with a green box.

EclipseWebinar

main eclipsewebinar

updated to 2.0.0 [DEPLOY]
Michael Oberlehner authored 42 minutes ago

Name	Last commit	Last update
Robot	updated to 2.0.0 [DEPLOY]	42 minutes ago
.gitlab-ci.yml	Added .gitlab-ci.yml file	1 hour ago
README.md	test [DEPLOY]	1 hour ago

README.md

4diac Library Packaging with GitLab CI

This repository contains a 4diac source project.

The GitLab pipeline automatically creates a downloadable 4diac library package from this project and uploads it to the GitLab Package Registry.

Project information

- 4 Commits
- 1 Branch
- 0 Tags
- 116 KiB Project Storage

README

CI/CD configuration

- + Add LICENSE
- + Add CHANGELOG
- + Add CONTRIBUTING
- + Add Kubernetes cluster
- + Add Wiki
- + Configure Integrations

Clone your Fork Repo in 4diac

The screenshot shows the Visual Studio Code interface with the 'Add Git Repositories' dialog box open. The dialog is titled 'Add Git Repositories' and contains the following elements:

- Search and select Git repositories on your local file system**
- Found one repository scanning one folder in 8ms
- Search criteria**
 - Directory: C:\Users\Michael\Documents\4diacWebinar\eclipsewebinar
 - Buttons: Browse..., Search
 - Look for nested repositories
 - Skip hidden directories
- Search results**
 - type filter text
 - Search results list: C:\Users\Michael\Documents\4diacWebinar\eclipsewebinar\.git
- Buttons: Add, Cancel

In the background, the file explorer shows a Git repository structure with a green box highlighting the 'Clone Repo' button. A text box on the right side of the image contains the text: 'Don't forget to use GIT view'.

Project Overview – Library Provider Project

The screenshot displays the Eclipse IDE interface. On the left, the System Explorer shows a project named 'FourDays4diac' with a 'Type Library' containing several sub-libraries. Two 'MANIFEST.MF' files are highlighted with red boxes. The central Manifest Editor shows the content of a 'Project Manifest' (MANIFEST.MF) with a green border. It defines required libraries and exports. The right Manifest Editor shows the content of a 'Library Manifest' (MANIFEST.MF) with a red border, defining an exportable library.

Project Manifest:
Defines required libraries and exports

Library Manifest:
Defines exportable Library

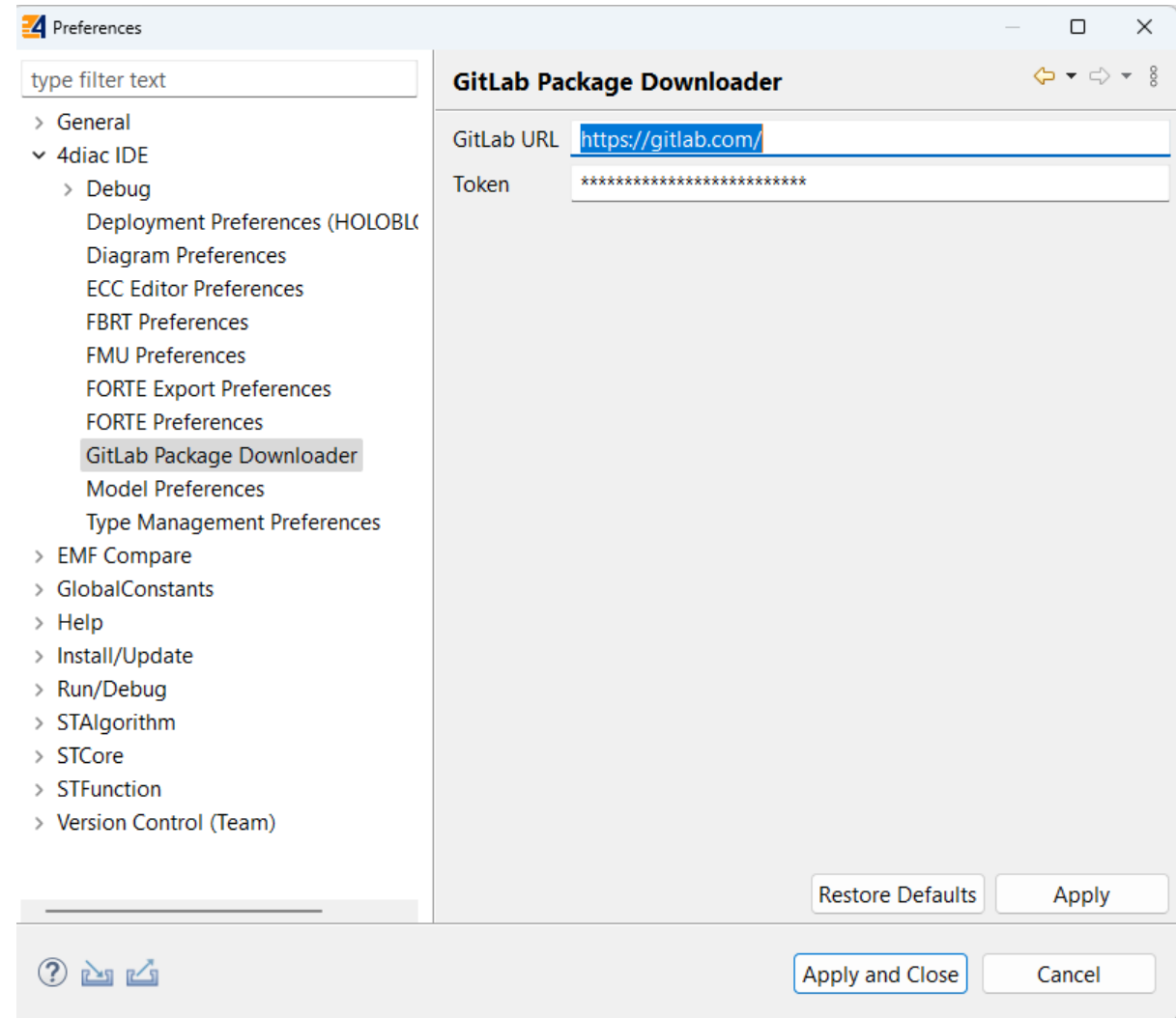
```
<?xml version="1.0" encoding="UTF-8"?>
<Manifest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.w3.org/2001/XMLSchema-instance" >
  <Dependencies>
    <Required SymbolicName="convert" Version="1.0.0"/>
    <Required SymbolicName="devices" Version="1.0.0"/>
    <Required SymbolicName="events" Version="1.0.0"/>
    <Required SymbolicName="iec61131-3" Version="1.0.0"/>
  </Dependencies>
  <Product>
    <VersionInfo Author="" Date="2025-02-06" Version="1.0.0"/>
  </Product>
  <Exports>
    <Library Comment="example for export" Name="Export Library 1" SymbolicName="lib01">
      <Includes>
        <LibraryElement>*</LibraryElement>
      </Includes>
      <Excludes>
        <LibraryElement>*:test:**</LibraryElement>
      </Excludes>
    </Library>
    <Library Comment="example for export" Name="Export Library 1" SymbolicName="lib02">
      <Includes>
        <LibraryElement>*</LibraryElement>
      </Includes>
      <Excludes>
        <LibraryElement>*:test:**</LibraryElement>
      </Excludes>
    </Library>
  </Exports>
</Manifest>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<Manifest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.w3.org/2001/XMLSchema-instance" >
  <Dependencies>
    <Required SymbolicName="lib01" Version="1.5.0"/>
  </Dependencies>
  <Product Name="Lib 02" SymbolicName="lib02" Comment="Test Library - Lib 02">
    <VersionInfo Version="1.5.2" Author="Eclipse 4diac" Date="2024-07-15"/>
  </Product>
</Manifest>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<Manifest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.w3.org/2001/XMLSchema-instance" >
  <Product Name="Lib 01" SymbolicName="lib01" Comment="Test Library - Lib 01">
    <VersionInfo Version="1.5.0" Author="Eclipse 4diac" Date="2024-07-15"/>
  </Product>
</Manifest>
```

Setup Package Downloader

- Select **Window** → **Preferences**
- Copy Token from gitlab fork
- Copy token README from public gitlab repo (if no gitlab account is available)



First Step – Create Token for Package Download

The screenshot shows the GitLab web interface. On the left sidebar, the 'Settings' menu is expanded, and 'Access tokens' is highlighted with a green box. The main content area displays the 'Project access tokens' page for the project '4days4diac'. It includes a search bar, a description of project access tokens, and two tables: 'Active project access tokens' (containing one token named 'FourDaysOf4diac') and 'Inactive project access tokens' (which is empty).

Michael Oberlehner / 4days4diac / Access tokens

Q Search page

Project access tokens

Generate project access tokens scoped to this project for your applications that need access to the GitLab API. You can also use project access tokens with Git to authenticate over HTTP(S). [Learn more.](#)

Active project access tokens 1 Add new token

Token name	Description	Scopes	Created	Last Used	Expires	Role	Action
FourDaysOf4diac		api, read_api, create_runner, manage_runner, k8s_proxy, self_rotate, read_repository, write_repository, read_registry, write_registry, read_virtual_registry, write_virtual_registry	Feb 06, 2025	4 hours ago	in 2 days	Developer	

Inactive project access tokens 0

Token name	Description	Scopes	Created	Last Used	Expired	Role
This project has no inactive access tokens.						



Thank you!

Michael.Oberlehner@osoft.at

Commercial support, custom features, training
Osoft e.U.

Michael.Oberlehner@jku.at

Research topics
LIT | Cyber-Physical Systems Lab
Johannes Kepler University Linz

