Publishing XFS4IoT specifications

Initial setup DRAFT

Copyright

All information contained herein is the property of KAL ATM Software GmbH (KAL) and all unauthorized use and reproduction is prohibited. Copyright © 2021 KAL. All rights reserved. Kalignite is a registered trademark of KAL.

Contact

Tel: +49 89 2019 0155  
Email: Use the email address of your designated support engineer.  
www: http://www.kal.com

Table of Contents

[Copyright i](#_Toc67562730)

[Contact i](#_Toc67562731)

[1. Overview 1](#_Toc67562732)

[2. Prerequisites 1](#_Toc67562733)

[3. Agreed conventions 2](#_Toc67562734)

[4. Generating the content 3](#_Toc67562735)

[4.1. Setting up the environment 3](#_Toc67562736)

[4.2. Generating documents 3](#_Toc67562737)

[5. Preparing files for publication 4](#_Toc67562738)

[5.1. HTML pages 4](#_Toc67562739)

[5.2. Converting HTML to DOCX 4](#_Toc67562740)

[5.3. Converting DOCX to PDF 4](#_Toc67562741)

[6. Publishing the documents 5](#_Toc67562742)

[Support 6](#_Toc67562743)

# Overview

This document details publishing XFS4IoT specifications to GitHub Pages.

# Prerequisites

Prepare a dedicated VM.

Download the following to that VM:

* All programs listed in the XFS4IoT VS Code extension instructions.
* Pandoc.
* A reference Word document (e.g. reference.docx).

# Agreed conventions

* Subfolder names should start with an uppercase character, and any references to related files should be modified to be consistent with this convention.
* Get rid of the old XFS3 naming convention.  
  For example, replace WFS\_CMD\_CIM\_CREATE\_P6\_SIGNATURE.Description.md with CreateP6Signature.Description.md.
* Any large blocks of text within YAML files should be moved into MD files.
* We should use a short naming file name format, dropping redundant preceding information (for example, XFS4IOT- and the Command/Event and Device Name) as in the following example:  
  Specifications/CardReader/Events/XFS4IoT-CardReader-Event-InsertCardEvent.yaml becomes Specifications/CardReader/Events/InsertCardEvent.yaml.
* Make sure local setup for creating YAML uses LF instead of CRLF for end of lines.
* All classes need to use the same formatting for heading levels. We have a clash between the heading at the YAML level and the MD level if they do not match. This breaks the table of contents and the HTML output appearance.   
  All headings will be handled at the YAML level and MD files shouldn’t have any heading tags in them.
* The text from all files should be run through a spelling and grammar check before committing to GitHub. Each committee member should be responsible for this on their end. Spell-checker extensions are available for VSCode.
* Use > for quotes/block specifiers for any text.
* List of recommended VSCode extensions:
  + Prettier
  + Markdownlint
  + YAML by RedHat (requires Java)
* Hyperlinks should be used to represent references to properties in text.
* Italics should be used to represent properties.
* Code style should be used to represent enumerators.

# Generating the content

## Setting up the environment

Complete the following steps to prepare the environment:

* Install VS Code.
* Follow the XFS4IoT VS Code extension installation instructions.
* Install pandoc.
* Copy the latest XFS4IoT specifications to the VM, either by using Git on the VM or by downloading the zip file from the XFS4IoT specifications GitHub webpage.

## Generating documents

Document generation is done with the VS Code extension Generate Documents command. See the extension readme for more information.

# Preparing files for publication

## HTML pages

The document generation command produces the index.html file in the location defined in the extension settings.

## Converting HTML to DOCX

Conversion uses the index.html file and a reference Word document. Both must be placed in the same folder.

To convert the HTML to DOCX:

1. Open the command prompt in Administrator mode.
2. Navigate to the directory where the index.html and reference files are saved with the cd command (or fully qualify the following command with the file path).
3. Enter the following:

pandoc index.html -s -o new\_chapters\_preview.docx --reference-doc=reference.docx

1. The Word document appears in the working directory.

## Converting DOCX to PDF

The Adobe Acrobat PDFMaker extension for Microsoft Word allows you to tweak the output PDF for size and quality as required.

To convert DOCX to PDF:

1. Copy the Word document to a machine with Microsoft Word with the Adobe Acrobat PDF extension installed.
2. Open the document in Word.
3. Select the Acrobat tab from the Office ribbon.
4. Click Preferences > adjust settings as required.
5. Click Create PDF.
6. Choose the file name and save location.
7. The document is processed and converted.  
   This process may take some time, as the documents are large files.

# Publishing the documents

To publish the documents:

1. Upload the documents to the correct folder in the gh-pages branch of the Specifications-Preview.github.io repository.
   1. Upload the HTML file to the pages folder.
   2. Upload the DOCX and PDF files to the assets folder.
2. Update the index.md file with the names and relative locations of the files.
3. Navigate to https://xfs4iot.github.io/Specifications-Preview.github.io/ to verify your changes. It may take a few moments for your changes to sync up with the server.

Support

KAL provides training, technical support, and professional services relating to all aspects of product development. Please contact us, using the contact information above, or contact your allocated support engineer directly.

If problems arise with your software, you may be asked to supply a Trace file.