

Dokumentation zum Aktualisierungsprozess von fn2Web

Aktualisierungsprozess von fn2Web

Versionierung, Releases und Updates für fn2Web

fn2Web ist nach dem Schema <fn2Web-Version>.<Release>.<Update> versioniert. Die Versionsnummer

2.04.05.032

beschreibt zum Beispiel die fn2Web-Version 2.04, das Release 5 und das Update 32.

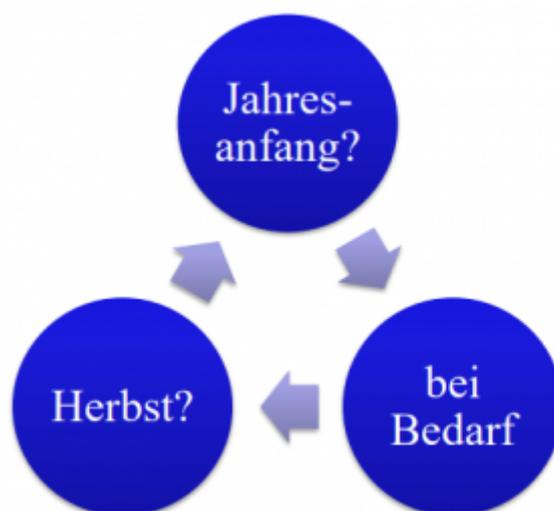
fn2Web-Version

Die fn2Web-Version wird nur bei grundlegenden, tiefgreifenden Änderungen im System fortgeführt, was nur sehr selten der Fall sein wird.

Releases

Releases werden von uns grob zwei Mal im Jahr geplant, zum Jahres-Anfang und im Herbst. Optional können bei Bedarf zusätzliche Releases veröffentlicht werden, z.B. wenn ein schwerer Fehler im System eine zeitnahe Aktualisierung bedingt. Releases beinhalten aufwändigere Erweiterungen und Anforderungen und verlangen häufig das Anpassen von hochschulspezifischen Konfigurationsdateien.

Obwohl ein neues Release so schnell wie möglich eingespielt werden sollte, ist uns bewusst, dass das aufgrund des hohen Test-Aufands nicht immer geht. Daher versorgen wir sowohl das aktuelle, als auch das vorhergehende Release mit Updates.



Die Termine für Releases und Updates ist absichtlich etwas vage gehalten. Die dadurch erlangte Freiheit ermöglicht es uns zeitnah und agil auf die Bedürfnisse unserer Kunden eingehen zu können.

Updates

Updates sind meist kleinere Erweiterungen und dringende Fehlerbehebungen, die von uns zeitnah veröffentlicht werden. Es ist nicht zwingend notwendig fn2Web immer auf dem neuesten Update-Stand zu halten. Dennoch sollten die [Releasenotes](#) aktiv gelesen werden, da mit Updates auch kritische allgemeine Fehler behoben werden. Eine viel genutzte best practice ist auch im [Ticketing-System](#) die neuesten erledigten Tickets im Auge zu behalten oder, bei Problemen, dort erst ein mal stichpunktartig zu suchen.

Releasezyklus für fn2Module

Der Releaseprozess für die fn2Module ändert sich nicht. Neue Versionen, die von Kunden angeforderte Features enthalten, werden nach Fertigstellung bereitgestellt. Die in den neuen Versionen enthaltenen Änderungen sind in den dazugehörigen Release Notes dokumentiert.

Gegenseitig abhängige Updates von fn2Module und fn2Web

Wenn es zwingend erforderlich ist, dass beide Bereiche (fn2Module und fn2Web) voneinander abhängig sind und folglich gleichzeitig aktualisiert werden müssen, gibt es dazu von uns vorab entsprechende Informationen. Ein simultanes Update von fn2Web und fn2Module ist bspw. erforderlich, wenn neue Modellierungsmöglichkeiten oder neue Druckkomponenten hinzugefügt worden sind. Bei kleineren Änderungen hingegen, z.B. Bedienverbesserungen in fn2Module, kann das Update von fn2Module unabhängig von fn2Web eingespielt werden. Das gleiche gilt für fn2Web.

(SQL-)Skripte zur Aktualisierung der Datenbankstrukturen

Sowohl für fn2Module als auch für die Komponenten von fn2Web gibt es SQL-Skripte. Diese **müssen** eingespielt werden.

Dateien mit den Skripten für die drei verschiedenen Datenbanken finden sich auf dem ftp-Server. Bitte suchen Sie selbst die passende Variante heraus. Achten Sie darauf, dass Sie Skripte lückenlos einspielen. Dabei hilft die eindeutige Skript-Nummer. Achten Sie auch darauf, jedes Skript mit „commit;“ abzuschließen, sofern kein Autocommit eingestellt ist. Sollte ein Skript eine längere Laufzeit haben, brechen Sie es **nicht** ab, auch wenn damit „nur“ ein Default-Wert gesetzt wird.

Es kann vorkommen, dass Sie selbst Werte im Skript eintragen oder ergänzen müssen. Beachten Sie dazu die Hinweise in den Kommentaren.

Bei Rückfragen wenden Sie sich zunächst an die im Skript angegebene Ansprechperson. Sollte ein Problem auftreten, das mehrere User betreffen könnte, eröffnen Sie bitte ein Ticket in dem entsprechenden Bereich SQL-Skripte unseres Ticketing-Systems

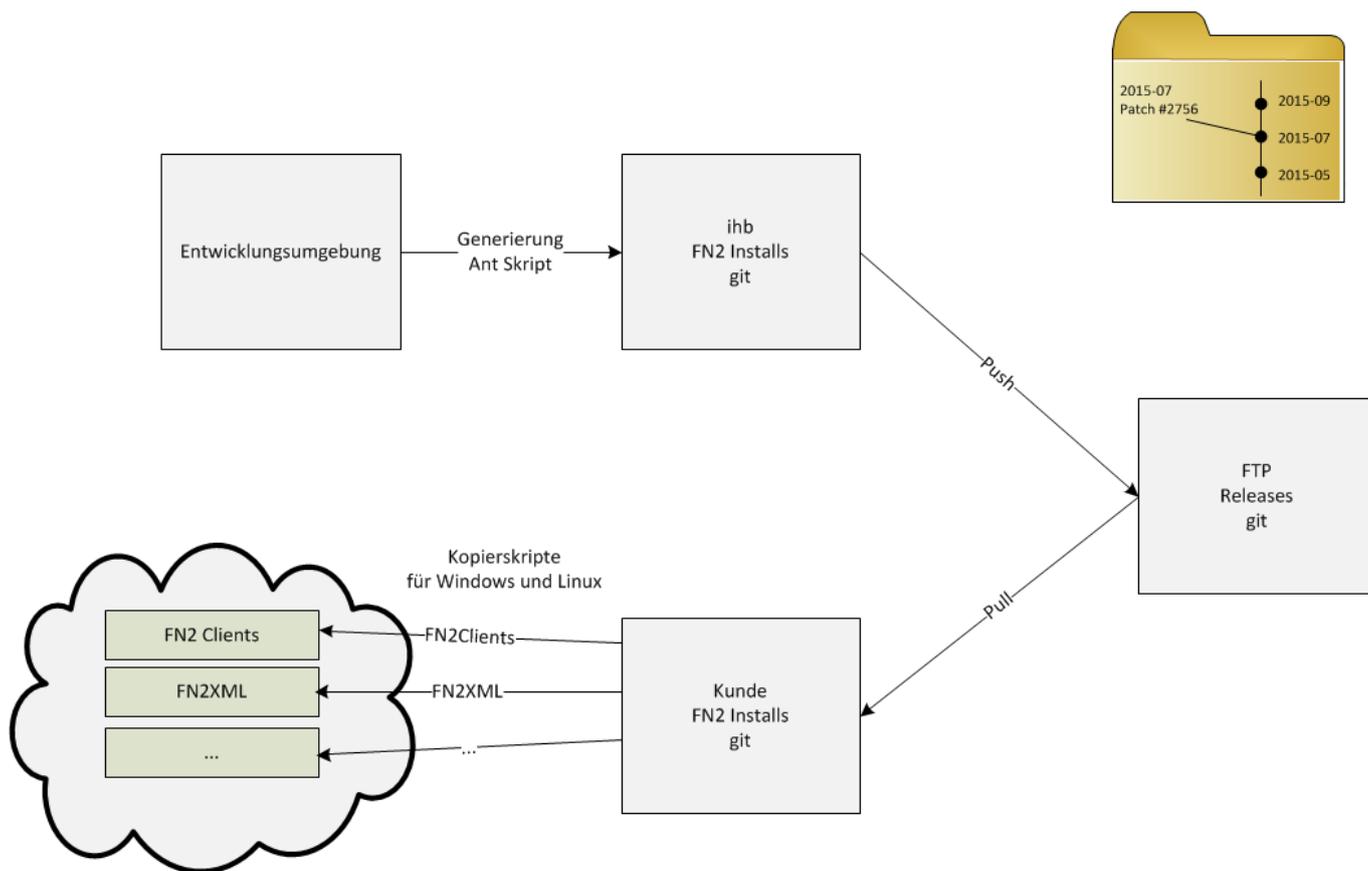
GIT zum Download der Releases

GIT ist eine freie Software zur verteilten Versionsverwaltung und kann [hier](#) für alle gängigen Plattformen heruntergeladen.

Eine Übersicht zahlreicher grafischer Oberflächen für GIT gibt es unter [hier](#). Die Verwendung einer grafischen Oberfläche ist aber nicht zwingend erforderlich, da alle benötigten Befehle in diesem Wiki beschrieben werden. Wir selbst nutzen neben der Kommandozeile [SourceTree von Atlassian](#). Das Programm ist sowohl für Windows als auch für Mac verfügbar.

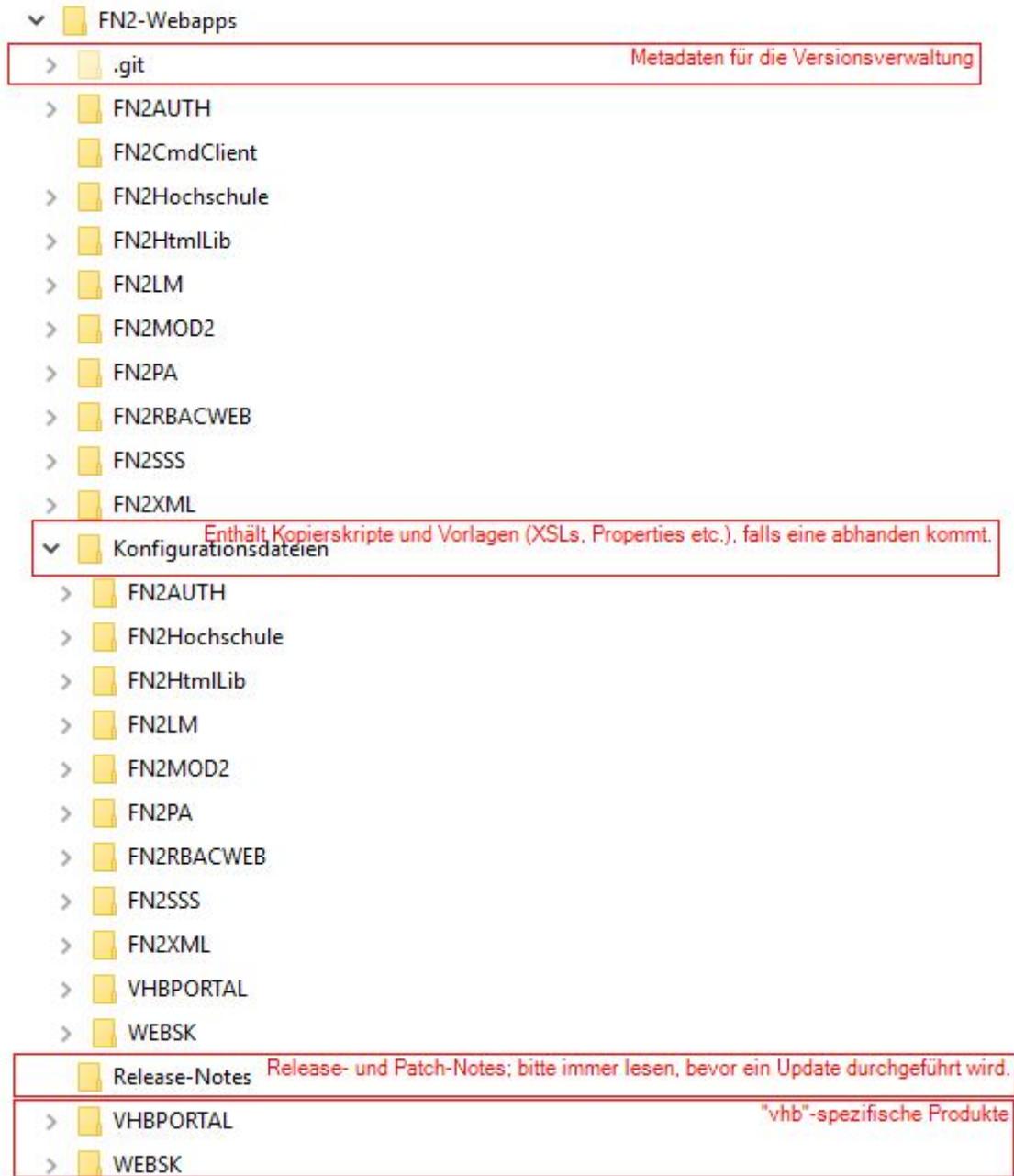
Ein Grundlagenbuch ist [hier](#) frei verfügbar. Einen tieferen Einblick in die Syntax von GIT-Befehlen erhalten Sie in der offiziellen [Dokumentation](#).

Übersicht über die Zusammenhänge beim neuen Release-Mechanismus



fn2Web-Releases Verzeichnisstruktur

Mit dem nachfolgenden Screenshot wird die Verzeichnisstruktur, die man beim Download von fn2Web-Releases erhält, erklärt.



Wichtige GIT-Befehle

Zum erstmaligen Download der FN2-Webapps

```
git clone https://<BENUTZERNAME>:<PASSWORT>@gitlab.ihb-eg.de/fn2/fn2webapps.git
```

Der Befehl „git clone“ kopiert die Ablage in einen neu erstellten Ordner und erstellt einen remote-tracking Zweig für jeden Zweig in der originalen Ablage.

Während des Klonens kann mehrfach die folgende Fehlermeldung auftreten:

```
error: RETR response: 550 (curl_result = 78, http_code = 550, sha1 = 98120a75125204713db511227718579caea88aa6)
```

Diese bitte ignorieren und den Klon-Vorgang bis zum Schluss durchlaufen lassen.



In dem ausgecheckten Verzeichnis dürfen niemals vorhandene Dateien geändert werden!

Laden von Updates des aktuellen Zweiges

Um zu prüfen, ob neue Zweige oder Updates zur Verfügung stehen, ist

```
git fetch --all -v
```

einzugeben.

Mit

```
git pull
```

wird automatisch zum aktuellsten Stand des Zweiges gesprungen. Dazu werden alle Neuerungen geholt.



Um das problemlos zu ermöglichen, dürfen vorhandene Dateien niemals geändert werden!

Um anzuzeigen, **welche Zweige existieren**, gibt man den Befehl

```
git branch
```

ein. Mit * in der Ausgabe ist markiert, auf welchem Zweig man sich befindet.

Zu einem anderen Zweig kann gewechselt werden mittels des Befehls

```
git checkout ZWEIGNAME
```

Dabei beachten, wenn der Name des Branches mit einem # beginnt, diesen Namen in einfache Anführungszeichen setzen. Beispiel:

```
git checkout '#3563'
```

Um die komplette Historie anzeigen zu lassen, wird der Befehl

```
git log --graph --decorate --oneline --all --since="2015-10-01"
```

einggegeben. Um anschließend einen älteren Zustand wiederherzustellen, geben Sie folgenden Befehl

mit dem Label des commits ein der zu dem gewünschten älteren Zustand passt.

```
git checkout <commit label>
```

Die Markierung (**HEAD**) zeigt einem an, wo man sich gerade befindet.

Die Ausgabe ist wie folgt zu lesen: *Label des Commits (HEAD → dieser Zustand ist ausgecheckt, origin/... → Name und Zweig des entfernten Rechners (auf dem FTP-Server), origin/HEAD → dieser Zustand ist auf dem entfernten Rechner ausgecheckt, Name des Zweiges)*

Wartung: verwaiste Zweige löschen, um Speicherplatz frei zu geben

Dieser Befehl, löscht einen lokalen Zweig - üblicherweise, wenn eine Auslieferung zum Testen nicht mehr benötigt wird und Speicher freigegeben werden soll.

```
git branch -d <branchname>
```

Mit diesem Befehl werden die Zweige die auf dem Server bereits gelöscht wurden, bereinigt

```
git remote update --prune
```

Aktualisieren der FN2-Installation

Im Ordner Konfigurationsdateien gibt es Skripte für Windows (*CopyFN2-Webapps.cmd*) und Linux (*CopyFN2-Webapps.sh* und *fn2excl.txt*). Bitte diese als Kopiervorlage verwenden und in den Skripten die gewünschten Variablen setzen bzw. die auszuschließenden Ordner als solche markieren.

Wechseln eines Zweiges und anzeigen von Release-notes

Zu einem anderen Zweig kann gewechselt werden mittels des Befehls

```
git checkout ZWEIGNAME
```

Der Befehl

```
git checkout <commit hash>
```

wechselt zu einem früheren Zustand des Zweiges.

Die <commit hash> werden mit dem Befehl

```
git log
```

oder mittels

```
git log --pretty=format:"%ar - %cd - %H - %an - %s" --since="2015-10-01" --
```

```
graph --date=short
```

angezeigt. (Weitere Dokumentation zur Ausgabe gibt es [hier](#).).

Die Release-Notes können mit demselben Befehl

```
git log
```

angezeigt werden und mittels

```
git show
```

werden die Release-Notes und die durchgeführten Änderungen dargestellt.

Mit den folgenden Befehlen, können die release-notes in verschiedenen Formaten dargestellt werden.

```
git log --format=oneline
```

```
git log --format=short
```

```
git show --format=short
```

Der Befehl

```
git status
```

zeigt an, auf welchem Zweig man sich befindet.

Troubleshooting

Wenn der Hinweis ignoriert wurde und vorhandene Dateien geändert wurden, ist folgendes durchzuführen:

Ein Update wird dann mit dem Hinweis abbrechen, dass lokale Änderungen durch das Update überschrieben würden und man diese entweder committen (was von FlexNow-Seite aus nicht erwünscht ist) oder diese „Stashen“ soll. In diesem Fall ist ein Stash durchzuführen, d.h die veränderten Daten in einen Zwischenspeicher abzulegen.

```
git stash
```

Danach ist es nötig mit dem bereits erwähnten Befehl

```
git pull
```

ein Update zu holen und abschließend den Stash zu bereinigen.

```
git stash list
```

```
git stash drop stash@{0}
```

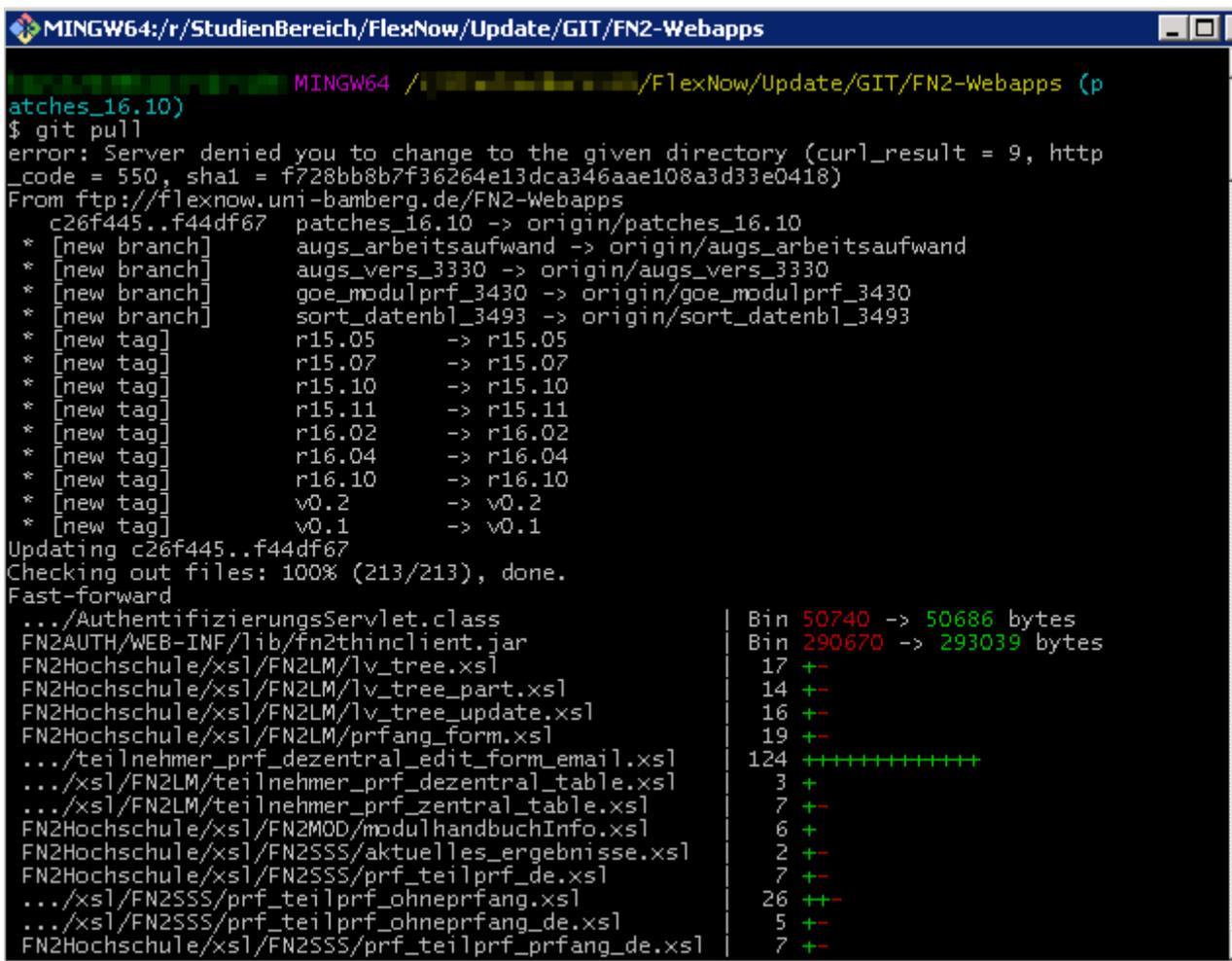
Hinweise gibt es im [GIT-Buch](#).

Probleme mit Sonderzeichen im Passwort

Falls ein Sonderzeichen im Passwort ist und der Zugang zum GIT-Server verweigert wird, muss das Passwort im lokalen Kunden-Repository in der Datei `.../.git/config` in Hochkommata gesetzt werden ([Quelle](#))

Fehlermeldungen beim Update

Fehlermeldungen, wie der beim nachfolgenden Screenshot (*error: Server denied you to change to the given directory*) oder wie dieser dokumentiert im [Ticket #2922](#) (*error: RETR respons: 550 ...*), können ignoriert werden. Das Update wird trotzdem ausgeführt. Überprüft werden kann das, indem entweder der Befehl `git pull` erneut eingegeben wird, diesmal mit dem Ergebnis *Already up-to-date*, oder, indem im Unterordner *Release-Notes* die aktuelle *Hinweis-Datei* geöffnet wird und geprüft wird, ob ein neuer Eintrag hinzugefügt wurde.



Link zum GitLab-Server

Der FTP-Server ist für Git nicht mehr verfügbar. Deshalb muss in der Datei /.git/config der Bereich remote = „origin“ folgende URL enthalten:

```
[remote "origin"]
  url = https://gitlab.ihb-eg.de/fn2/fn2webapps.git
```

From:

<https://wiki.ihb-eg.de/> - **FlexWiki**

Permanent link:

<https://wiki.ihb-eg.de/doku.php/fn2/aktualisierung?rev=1607081026>

Last update: **2020/12/04 12:23**

