

EAGLE - CAD Schulung

Datei-Namen und Suche mit Windows-Explorer und EAGLE

Seite 001: Inhaltsverzeichnis

Seite 002: Windows-Dateinamen Problem

Seite 003: Datei suche mit Windows-Explorer und EAGLE

Seite 004: Verbotene Zeichen im Pfad/Dateinamen

Seite 005: Ende dieser Dokumentation

Für die Erstellung des INDEX in diesem Schaltplan unbedingt die folgende Option des ULP benutzen:
RUN e-seiten-index EAGLE-Schulung_AZ-CAD

Probleme mit Dateinamen größer 260 Zeichen

Unter Windows sind in der Regel nur 260 Zeichen für Pfad plus Dateinamen erlaubt.

Versucht man eine Datei zu öffnen, deren Pfad/Datei-Name länger als die erlaubten 260 Zeichen ist, wird eine Meldung wie in der Description dieser Seite ausgegeben.

Kürze in diesem Fall den Dateinamen oder speichere die Datei unter einem kürzeren Pfad.

Dateien deren Pfad/Datei-Name länger als die erlaubten 260 Zeichen sind, werden auch mit dem User-Language-Programm 'file-list-dir.ulp' gemeldet.

**** Unter Windows 10 längere Dateinamen unterstützen:**

Quelle :

<https://answers.microsoft.com/de-de/windows/forum/all/aufhebung-der-dateinamenl%C3%A4nge-unter-windows/70811d50-e98c-4825-af28-fad79d3eca63>

Aufhebung der Dateinamenlänge unter Windows 10 aktuelle Versionen

- Windows-Taste + R drücken, „gpedit.msc“ eingeben und starten
- Computerkonfiguration -> „Administrative Vorlagen“ -> System -> Dateisystem
- Im rechten Fenster nun „Lange Win32-Pfade aktivieren“ („Enable NTFS long paths“) doppelt anklicken
- Auf „Aktiviert“ stellen Übernehmen und OK drücken

Datei suche mit Explorer und EAGLE

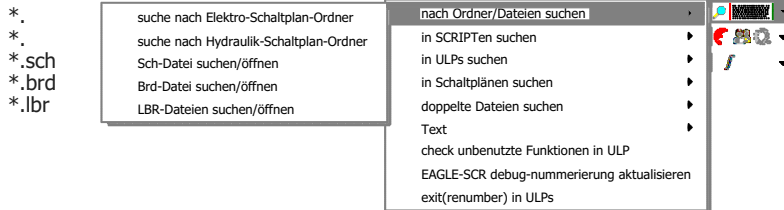
Ein weiteres Problem ist die Suche nach einer bestimmten Datei.

Benutzt man den Windows-Explorer, kann die Suche extrem lange dauern, da die Suchfunktion des System (Explorer) nach dem Suchbegriff auch innerhalb von Dateien sucht, je nach eingestellter Option im Explorer.

Je mehr Dateien unter dem Suchgrundpfad vorhanden sind und je größer die Dateien sind, dauert die Suche mehrere Minuten und evtl. auch das mehrfache davon.

Eine wesentlich schnellere Methode eine Datei zu finden ist das 'file-list-dir.ulp'.

In der Text-Botton-Leiste gibt es dazu schon vorgefertigte Sucheinträge für



Wie man das ULP benutzen kann, ist im Quelltext des ULP selbst beschrieben.

Um diese Beschreibung zu lesen kann man auch im EAGLE-Control-Panel der Pfad zu User-Language-Programm aufklappen, in der Liste der Dateien zu file-list-dir.ulp blättern und auf der rechten Hälfte des Fenster die Beschreibung (#usage) lesen.

Verbotene Zeichen in PFAD/DATEI-Namen

Warum bestimmte Zeichen wie @ und & und / verboten sind.

=====

Das AMPERSAND-Zeichen &

Loading D:/ENI3/Allgemeines/Elektro-Schaltplan/projects/Labortechnik/AMP Superseal M & F to open/AMP_Superseal_M&F_to_open.sch ...

Fehler:

Zeile 813, Spalte 88: Es wurde ';' erwartet, stattdessen erhalten ' '.

Das Problem liegt bei SVN, da der Dateiname in die Zeile und zwischen den Zeichen mit ">\$Id: \$<" eingefügt wird:

```
<text x="-81.28" y="-17.78" size="2.1844" layer="95">$Id: Fehler&Probleme.sch 23802 2021-11-03 13:17:49Z username1 $</text>
```

EAGLE in der HTML-Datei jedoch die Zeichenkombination "&" erwartet um das Ampersand darzustellen.

Hier kommt dann der HTML-Parser von EAGLE aus dem Tritt, meldet die Fehlerhafte Zeile und bricht den Ladevorgang ab.

=====

Das AT-Zeichen @

Es macht Probleme in DOS-Kommandos, also wenn z.B. aus EAGLE heraus eine BATCH-Datei erzeugt wird die mit system() als DOS-Kommando ausgeführt werden soll.

@ Unterdrückt die Ausgabe des Befehls in einer Stapelverarbeitungsdatei

Auszug aus HELP CMD

1. Sind alle folgenden Bedingungen erfüllt, wird ein doppeltes Anführungszeichen auf der Befehlszeile beibehalten:
 - keine Option "/S"
 - genau zwei doppelte Anführungszeichen
 - keines der folgenden Zeichen zwischen den doppelten Anführungszeichen: <>()@^|
 - es ist mindestens ein Leerzeichen zwischen den doppelten Anführungszeichen
 - die Zeichenfolge zwischen den doppelten Anführungszeichen ist der Name einer ausführbaren Datei

Und es macht Probleme mit/bei SVN:

Siehe auch in SVN PEG-Revision <<http://blog.blackslash.de/archives/15-Subversion-PEG-Revisionen.html>>

Die sog. "PEG-Revision". Die Peg-Revision legt den Aufsetzpunkt des History-Tracings fest. In der Praxis benötigt man Peg-Revisionen meistens dann, wenn im Projekt Refactoring- oder Umstrukturierungen rückgängig gemacht werden sollen. Möchte man eine gelöschte Datei aus einer älteren Revision für zukünftige Revisionen wiederherstellen, sollte man die Datei nicht einfach im Dateisystem kopieren. Denn in diesem Fall legt das Subversion die komplette Datei neu im Repository ab und behandelt diese, als wäre es eine vollkommen neue Datei. Stattdessen sollte man die Datei mit dem "svn copy" Befehl wiederherstellen und behält dadurch die Historie der Datei bei:

svn copy MeineDatei.txt@5 MeineDatei.txt

(Wobei die 5 durch die Revision zu ersetzen ist, in der die Datei im Repository noch vorhanden war)

<<https://svnbook.red-bean.com/de/1.5/svn.advanced.pegrevs.html>>

Peg-Revisionen werden dem Subversion-Kommandozeilen-Client in At-Syntax mitgegeben, die so genannt wird, da diese Syntax das Anhängen eines „At-Zeichens“ (@) und die Peg-Revision an das Ende des mit der Revision verbundenen Pfades vorsieht.

=====

Das Zeichen /

Ist in der DOS-BOX (Windows-Eingabeaufforderung) teilweise gleichbedeutend mit dem \, zumindest in Pfadangeben.

	1	2	3	4	5	6	7	8	9	10	11
A											
B											
C											
D											
E											
F											
G											

Vorläufiges Ende der Dokumentation 2021-08-10
EAGLE-Schulung_Datei-Namen_Probleme
A. Zaffran