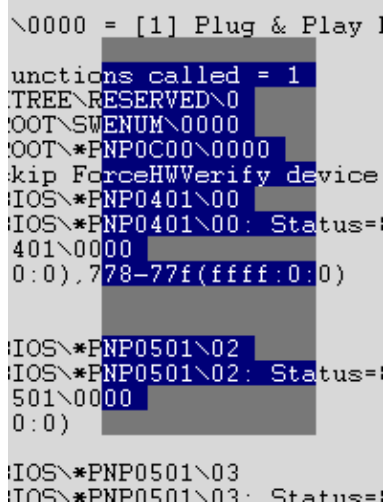


Jens' File Editor – Der ultimative Editor für jeden Softi

Da es oft bemängelt wurde das JFE ja ganz brauchbar, aber ohne jede Dokumentation , die nahezu unerschöpflichen Möglichkeiten gar nicht auszunutzen seien, soll jetzt endlich Abhilfe geschaffen werden. Hier werden aber nur die wichtigen Dinge beschrieben sein die sich von anderen Editoren unterscheiden.

- Blöcke bearbeiten
- Farbverwaltung
- Syntaxcolorierung
- Drucken
- Suchfunktion
- Lesezeichen
- Spezielle Einstellungen
- Toolmenü
- Autobrowser
- spezielle Operationen
- HTML-Projektfile
- Ein eigenes Template für ein HTML-Projekt
- Arbeitsbereiche
- Kontextmenü
- Makros
- Funktionstasten
- Anbindung von anderen Anwendungen über TCP-Ports
- Sonstiges

Wie können Blöcke markiert, gelöscht und eingefügt werden



```
\0000 = [1] Plug & Play ]
unctions called = 1
TREE\RESERVED\0
OOT\SWENUM\0000
OOT\*PNP0C00\0000
kip ForceHWVerify device
IOS\*PNP0401\00
IOS\*PNP0401\00: Status=
401\0000
0:0),778-77f(ffff:0:0)

IOS\*PNP0501\02
IOS\*PNP0501\02: Status=
501\0000
0:0)

IOS\*PNP0501\03
IOS\*PNP0501\03: Status=
```

Das funktioniert mit der Maus im Zusammenhang mit der ALT-Taste.

- Bei gedrückter ALT-Taste + mit linker Maustaste den Startpunkt setzen
- durch Ziehen bei mit gedrückter linker Maustaste den Bereich markieren.
- nach Loslassen von ALT und linker Maustaste ist der Bereich markiert und kann wie üblich kopiert (CTRL-C), gelöscht(DEL) oder ausgeschnitten(CTRL-X) werden.
- Markierung ist auch möglich mit ALTGr+Cursortasten

Bereiche die über die Zeile hinausgehen werden dabei mit Leerzeichen aufgefüllt und mit Zeilenumbruch abgeschlossen. Es entsteht also immer ein Textrechteck.

Die Farbe für solche beim Markieren tatsächlich nicht vorhandenen Leerzeichen

Jens File Editor Description

kann im Farbmenü eingestellt werden. (15. Texthintergrund Farbe).
 Blöcke (aber auch normal markierter Text) können entweder wie üblich an der aktuellen
 Cursorposition (CTRL-V) eingefügt oder aber in eine Spalte eingefügt werden.
 Letzteres geht folgendermaßen vor sich:

Es gibt drei Varianten Blöcke einzufügen

1. Der Block wird in den Text eingefügt da könnte so aussehen (B=eingefügter Block, T=Text)

```
TTTTTTTTTTTTTTTTTTTTTTT
TTTTTTTTBBBBBTT
TTTTTTTTBBBBB
TTTTTTTTBBBBBTTTTTTTTTT
TTTT  BBBB
TTTTTTTTBBBBBTT
TTTTTTTTTTTTTTTTTTTTTTT
```

Dazu:

- mit linker Maustaste Cursor auf den Einfügemarke setzen
- **ALT**-Taste drücken
- mit rechter Maustaste Kontextmenü öffnen.
- "Einfügen" wählen >> der Text wird an der angegebenen Position in der Spalte eingefügt.

2. Der Block wird in der Spalte eingefügt der Text nach unten geschoben

```
TTTTTTTTTTTTTTTTTTTTTTT
TTTTTTTTBBBBBTT
        BBBB
        BBBB
        BBBB
        BBBB

TTTTTTTTTTTTTTTTTTTTTTT
TTTT
TTTTTTTTT
TTTTTTTTTTTTTTTTTTTTTTT
```

Dazu:

- mit linker Maustaste Cursor auf den Einfügemarke setzen
- **SHIFT**-Taste drücken
- mit rechter Maustaste Kontextmenü öffnen.
- "Einfügen" wählen >> der Text wird an der angegebenen Position in der Spalte eingefügt.

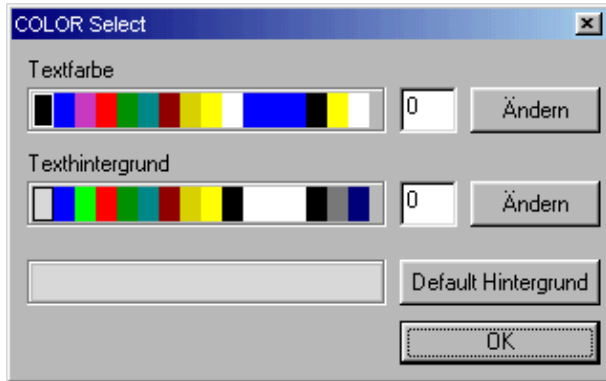
3. Der Block wird einfach mit Paste eingefügt

```
TTTTTTTTTTTTTTTTTTTTTTT
TTTTTTTTBBBBB
BBBBB
BBBBB
BBBBB
BBBBB
TT
TTTTTTTTTTTTTTTTTTTTTTT
TTTT
TTTTTTTTT
TTTTTTTTTTTTTTTTTTTTTTT
```

Natürlich können auch eigene Tastaturmakros definiert werden.

<<

Wie werden Farben verwaltet



Zu jedem Zeichen existiert ein Byte für die Farbinformation. Davon beschreibt das untere Nibble die Textfarbe, das obere Nibble den Texthintergrund. Die Farbnummer (0...15) ist ein Index auf eine Tabelle die dann den tatsächlichen 24Bit-**RGB**-Farbwert enthält.

Es sind also beliebige Farben auswählbar aber eben nur 16 verschiedene gleichzeitig.

Die Farbeinstellung erfolgt über den Menüpunkt "Einstellungen-Farben". Alle Farbeinstellungen werden in der INI-Datei bzw. einer Projektdatei gespeichert.

Geändert werden kann eine Farbe durch Markieren des jeweiligen Kästchens mit der Maus und Click auf "Ändern".

Aus dem Farbdialog kann dann eine andere Farbe gewählt werden.

Die Textvordergrundfarbe im Index 1(das erste Kästchen) ist die voreingestellte Schriftfarbe die immer für nicht selektierten und nicht syntaxcolorierten Text verwendet wird. (voreingestellt schwarz).

Gleichermaßen steht die voreingestellte Texthintergrundfarbe im Index 0 der darunterliegenden Tabelle für den Texthintergrund.

Der Texthintergrund bezieht sich tatsächlich nur auf die tatsächlich durch Text belegte Bereiche. Alle anderen Flächen werden mit der "Default Hintergrundfarbe" eingestellt. Zu Vereinfachung ist es so eingerichtet das eine Änderung dieser Farbe auch den Index 0 der Farbtabelle,d.h. die Texthintergrundfarbe mit ändert.

Wer tatsächlich einen andersfarbig hinterlegten Text wünscht muß erst den Default Hintergrund einstellen und danach den Texthintergrund.

Besondere Bedeutungen haben noch die Indizes 14 und 15.

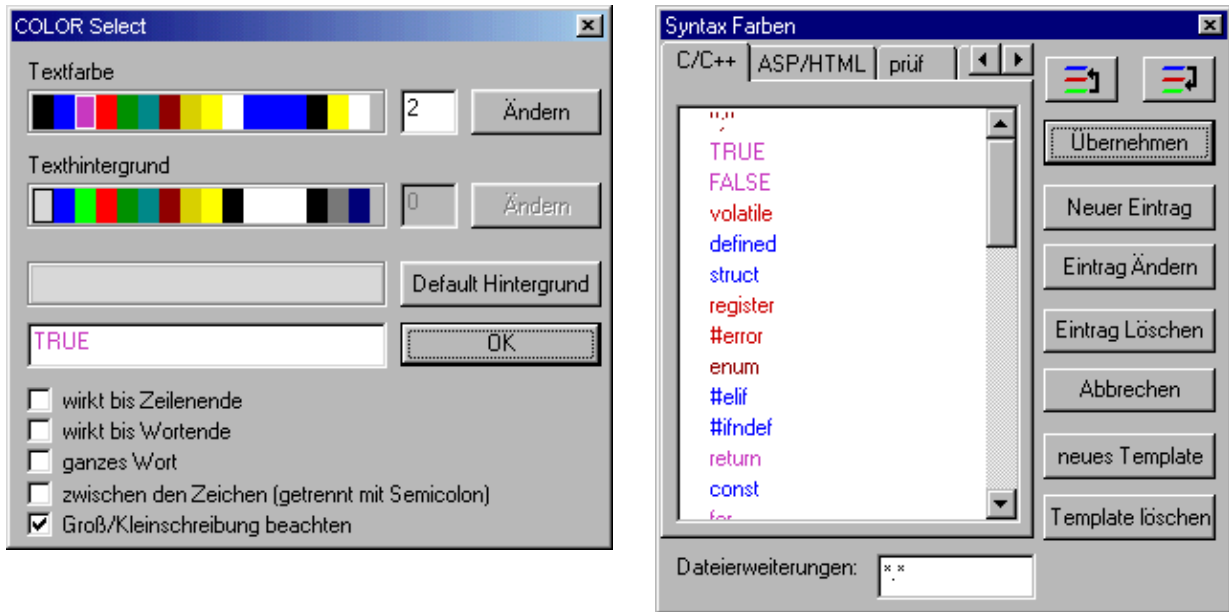
Der Farbindex 15 legt fest wie selektierte Bereiche in Vorder und Hintergrund dargestellt werden sollen. Damit kann es möglich das die Selektierung anders als durch die Windows Defaulteinstellung vorgegeben aussieht.

Der Index 14 hat eine doppelte Funktion. Zum einen bestimmt er das Aussehen der Zeilennummerierungsleiste zum Anderen gibt die Texthintergrundfarbe an wie bei Blockmarkierung nichtvorhandene Leerzeichen markiert werden.

<<

Wie definiert man sich ein eigenes Syntaxcoloring

Jens File Editor Description



Für C und C++ ist das Syntaxcoloring schon grundsätzlich vordefiniert. Man aber kann für verschiedene Programmiersprachen noch zusätzlich unterschiedliche Farbtemplates erstellen. Dazu öffnet man das Menü: "Einstellungen-Syntaxfarben.." und legt ein neues Template an. Der abgefragte Name ist beliebig und dient dazu die folgenden Einstellungen im INI-File zu unterscheiden. Das neue Template ist initial leer und muß erst mit den hervorzuhebenden Schlüsselworten gefüllt werden. Ein neuer Eintrag wird angelegt mit dem Button "neuer Eintrag". Im darauf geöffneten Dialog kann das Schlüsselwort, die Textfarbe und spezielle Flags angegeben werden. Die Schlüsselworte werden mit fallender Priorität abgearbeitet. Das bedeutet die weiter oben stehenden Schlüsselworte werden zuletzt bearbeitet und können u.u andere davor liegende überschreiben.

Beispiel:

1. `else` würde in so coloriert werden: `#else`
2. `#else`
anders herum wird das gewünschte Ergebnis erreicht:

1. `#else` würde in so coloriert werden: `#else`
2. `else`

Die Flags haben folgende Bedeutung:

- "wirkt bis Zeilenende"
(wenn das Schlüsselwort erkannt wird,wird dieses und der Rest der Zeile entsprechend gefärbt. Anwendung findet diese Option z.B. für Kommentare.)
Beispiel Basic: `F=SIN(X)` `'berechne den Sinus`
- "wirkt bis Wortende"
(wenn das Schlüsselwort erkannt wird,wird dieses und der Rest des Ausdrucks bis zum nächsten nicht alphanumerischen Zeichen entsprechend gefärbt.
Beispiel: zu markierendes Wort ist "F_COL", `F_COLORWHOLEWORD` und `F_COLUMN` werden selektiert nicht aber `F_CODE`.

Jens File Editor Description

- "ganzes Wort"
(das Schlüsselwort muss allein stehen und nicht innerhalb eines anderen Wortes)
Beispiel "if": #endif .. wird nicht gehilghted wenn die Option markiert ist sonst #endif
- "zwischen den Zeichen (getrennt mit Semikolon)"
Markiert den ganzen Bereich zwischen den beiden Strings. Dabei wird der erste String immer vom jeweiligen Zeilenanfang gesucht. Kann zusammen mit der Option "ganzes Wort" verwendet werden.
Beispiel HTML: "<,>" hilghted </p> oder auch </body>
- "Groß- Kleinschreibung beachten"
(damit sind casesensitive Unterschiede möglich zwischen gleichen Schlüsselworten)
Beispiel "int": int INT; wenn die Option markiert ist sonst int INT;

Die Hintergrundfarben sind nicht änderbar.

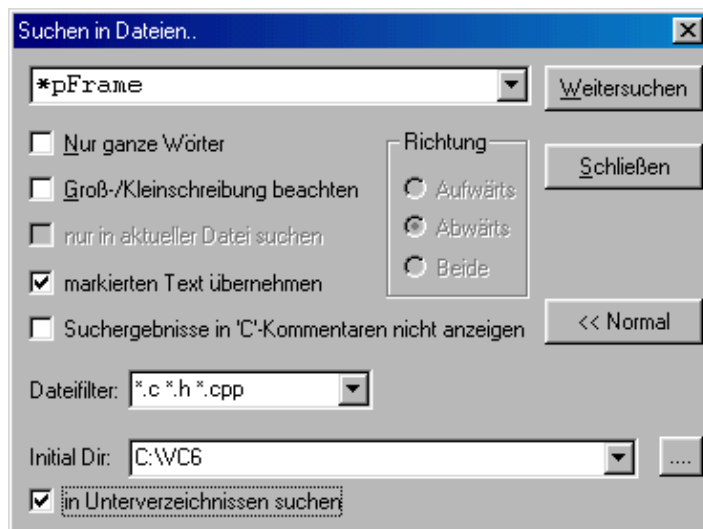
Eine Besonderheit gibt es noch: man kann auch einen Zeilenumbruch in ein Schlüsselwort aufnehmen mit "\n"

Dafür funktioniert aber auch die genauso geschriebene Zeichenkette nicht mehr!

Durch "OK" wird das neue Keyword übernommen. Das Ändern erfolgt analog.
Farben und Schlüsselworte sind im Klartext in der INI-Datei abgelegt und können auch direkt editiert werden.

<<

Die Suchfunktionen



Es gibt vier Stufen der Suchfunktion:

1. Suchen in dem gerade aktiven Dokument
2. Suchen in allen geöffneten Dateien
3. Suchen in der aktuellen Datei und Ausgabe der Suchergebnisse in das Ausgabefenster
4. Suchen in allen geöffneten Dateien und im aktuellen Pfad und optional in allen Unterverzeichnissen

Zur einfachen Bedienung läßt sich durch setzen des Checkmarks "**markierten Text übernehmen**" jede Selektion im Dokument direkt in das Suchfenster übernehmen. Ein wie in anderen Konkurrenzprodukten erforderliches Kopieren-Einfügen ist nicht erforderlich. Ansonsten entspricht die Suchfunktion unter 1. der sonst üblichen Funktionalität mit der Erweiterung das auch nach Sonderzeichen wie TAB "\t" oder Zeilenumbruch "\n" in Verbindung mit Text gesucht werden kann. Das Suchen nach Backslash muß

Jens File Editor Description

dadurch anders formuliert werden.

Wird die Option "nur in aktueller Datei suchen" aktiviert so werden die Suchergebnisse nicht einzeln angefahren sondern als Liste im

Ausgabefenster dargestellt. Durch Doppelklick kommt man dann an die entsprechende Stelle.

Beispiel:	c:\tmp	sucht nach "c:TABmp"
	c:\tmp	"c:\tmp"

Das Suchen nach den Varianten 2., 3. und 4. stellt das Suchergebnis im Ausgabefenster dar. Durch Doppelklick auf die jeweilige Zeile

wird das entsprechende Dokument aktiviert bzw. geladen.

Zusätzlich gibt es die Option sich Suchergebnisse innerhalb von C-Kommentaren nicht anzeigen zu lassen. Das ist zwar eine sehr sprachspezifische Einstellung, (*auf Wunsch eines einzelnen nicht genannt werden wollenden Nutzers*) aber bei der vorgesehenen

Anwendung als Programmiereditor sicher ganz nützlich.

Die Suchvariante 4. wird durch Click auf "Erweitert" aus Suchvariante 2. eingestellt. Damit ist es möglich auch in nicht aktuell

geöffneten Dateien nach einem Text zu suchen. Hier ist es natürlich sinnvoll die Zahl der zu durchsuchenden Files einzuschränken.

Das ist möglich über die Angabe eines Dateifilters. Mehrere Filter werden durch Leerzeichen getrennt. Wildcards sind möglich.

Beispiele: *.c *.h sucht nach allen Dateien mit der Erweiterung "h" und "c"
sucht nach allen Dateien die mit "O" beginnen,
dann ein beliebiges Zeichen gefolgt von "t" und weiter beliebigen Zeichen
O?t*.* bis zum Punkt und eine beliebige Erweiterung besitzen.

findet: [Otto.txt](#), [Ottokar.cpp](#)

Die Suche wird in dem Verzeichnis durchgeführt das als Initial-Verzeichnis eingetragen ist. Das ist beim Öffnen des Dialogs zunächst der

aktuelle Arbeitspfad. Unterverzeichnisse werden mit durchsucht wenn die entsprechende Option selektiert ist.

<<

Lesezeichen

Lesezeichen (oder Bookmarks) dienen dazu eine Stelle in einem Dokument schnell wiederzufinden. Das macht sich besonders gut wenn man in einem File ständig hin- und her scrollt oder aber ständig zwischen mehreren Dokumenten

umschaltet. Ein Bookmark wird dargestellt als eine Taste in einem Toolbar der (default) an der rechten Seite des Arbeitsbereiches angezeigt wird. Die Bookmarks des gerade aktiven Fensters sind gelbe Büroklammern alle anderen schwarze. Der zu jedem Button verfügbare Tooltip zeigt Filename und Zeilennummer des Lesezeichens an. Einem Lesezeichen

kann auch ein beliebiger Toolliptext zugewiesen werden. Dazu klickt man mit der rechten Maustaste auf den zu ändernden Bookmark und gibt in das erscheinende Eingabefeld den gewünschten Namen ein.

Bookmarks werden gesetzt durch:


- Toolbarbutton "Lesezeichen"
- Menü "Bearbeiten-Lesezeichen hinzufügen"
- F12 (bzw. anders definierter Hotkey)
- rechte Maustaste – Kontextmenü

Jens File Editor Description

Gelöscht wird auf die gleiche Weise. Man geht zu dem zu löschenden Lesezeichen und löscht es mit den gleichen Kommandos.

<<

Spezielle Einstellungen

OpenWindows Mehrfach Click	Damit werden auch Mehrfach-Clicks möglich <ol style="list-style-type: none">1. Click –setzt Cursor2. (Doppelclick) markiert das Wort3. markiert die Zeile4. markiert den gesamten Text
Markierung nach Paste halten	Normalerweise wird nach dem Einfügen aus der Zwischenablage der Cursor an das Ende des eingefügten Textes gesetzt und die Markierung entfernt. Durch die Einstellung "Markierung nach Paste halten" bleibt der Text selektiert.
Focus erst beim 2.Click	Beim Wechsel zwischen unterschiedlichen Dokumenten wird das aktuelle Dokument erst editierbar nach dem zweiten Click.
nur eine Instanz öffnen	normalerweise würde für jedes File das aus dem Explorer geöffnet würde (Drag,Doppelclick,SendTo) eine neue Instanz von JFE gestartet. Das kann man mit dieser Option verhindern. So wird bei schon existierender Applikation das File von dieser geöffnet und aktiviert.
Tabulatoren	 <p>Tabulatoren können entweder als Hard-TABS (Zeichencode 0x09) oder als Folge von Leerzeichen (SoftTABS) eingefügt werden. Eine Umwandlung von Hard- in Softtabs ist möglich kann aber nicht umgekehrt erfolgen. Hard-TABS können auch als Zeichen dargestellt werden.(intern wird ein Hard-TAB durch die Zeichen 0x09+ variable Anzahl von 0x01 dargestellt). Hardtabs können auch angezeigt werden. Das Tabulatorzeichen ist ein Kästchen, die (virtuellen)Leerzeichen Unterstriche.</p>
Backup	Die Möglichkeit von Backups eines Files können verschieden konfiguriert werden. Zunächst kann beim Öffnen einer Datei eine Kopie angelegt werden die auch während der gesamten Session nicht mehr geändert wird. Diese bekommt die Erweiterung ".org". Backups (Erweiterung ".bak", aber auch beliebig wählbar) können sonst bei jedem Speichern des Dokumentes angelegt werden. Um das Risiko des Datenverlustes in Folge von (nie auszuschließenden) zu minimieren kann ein Dokument automatisch in einstellbaren Intervallen gespeichert werden.
Autoindent	Bei Zeilenumbruch wird der Cursor unter den Anfang der vorhergehenden Zeile gesetzt. Zeilenumbruch nach einer öffnenden geschweiften Klammer '{' rückt die nächste Zeile zusätzlich um eine Tabweite ein. Eine schließende geschweifte Klammer '}' wird wenn möglich wieder unter der zugehörigen Spalte der öffnenden Klammer plziert. Macros werden auch wenn sie mehrzeilig sind in allen Zeilen eingerückt.

Jens File Editor Description

Keine Suchergebnisse in C-Kommentaren	Beim Suchen werden gefundene Strings die innerhalb von C-Kommentaren liegen nicht angezeigt. Diese Einstellung wirkt sowohl für das Suchen in Dateien wie auch für das Suchen nach Klammern. Die normale Suchfunktion wird dadurch nicht beeinflusst.
Speichern im UNIX-Format	Speichert nicht wie unter DOS den Zeilenumbruch mit CR/LF (0x0D/0x0A) sondern nur mit LF (0x0A) Diese Funktion kann auch durch Click auf das entsprechende Symbol in der Statusleiste ausgeführt werden Das Format einer Datei wird beim Öffnen erkannt und, wenn nicht geändert, auch wieder so gespeichert. Der Menüpunkt zeigt immer das Format der gerade aktiven Datei an!
Auto Kopieren	Markierter Text wird dabei automatisch in die Zwischenablage übernommen. Ist die Funktion eingeschaltet, wird die Markierung mit einem Rahmen um jedes markierte Zeichen versehen.(Farbe = Hintergrundfarbe14)
Zeilenumbruch	Damit wird eingestellt bei welcher Spalte die Zeile selbständig umgebrochen wird. Man kann dann wählen ob der automatische Umbruch mit abgespeichert werden soll. Manchmal möchte man zwar einen automatischen Zeilenumbruch, den aber nicht mitten in einem Wort. Dann markiert man "Zeilenumbruch nur an Wortgrenzen". Der Umbruch erfolgt dann an der nächsten Wortgrenze (Leerzeichen oder Tab) vor dem Eingestellten Limit. Findet sich in der Zeile kein Trennzeichen wird trotzdem im Wort getrennt! ACHTUNG, der automatische Zeilenumbruch verlangsamt den Editor enorm, sollte also besser ausgeschaltet bleiben.
Auto Suche	Schaltet den <u>Autobrowser</u> ein/aus
Autosuch Konfiguration	Öffnet den Konfigurationsdialog für den <u>Autobrowser</u>
Defaultverzeichnis	Wenn dieser Eintrag gesetzt ist, wird für alle Öffnen /Speicher -operationen das gleiche Defaultverzeichnis angeboten. Das Defaultverzeichnis wird sowohl in Arbeitsbereichen als auch in der INI-Datei gespeichert.

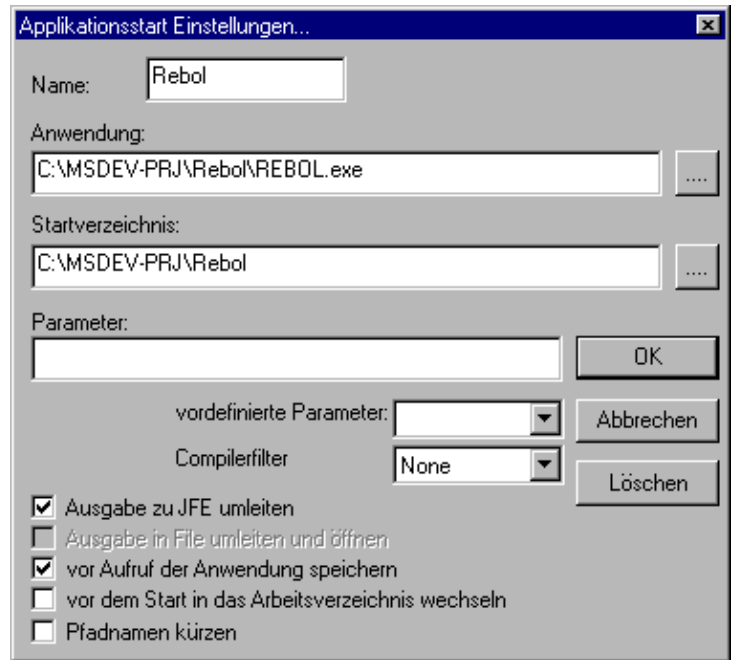
<<

Das Tool Menü

Es ist eine sehr angenehme Einrichtung editierte Quellen gleich aus dem Editor heraus zu Assemblieren, Compilieren oder zu Linken.
JFE liefert auch diese Möglichkeit. Unter dem Menüpunkt:
"Einstellungen-Tools hinzufügen" öffnet sich rechts dargestellter Dialog.(Bei einem neuen Tool ist er leer)Da unterschiedliche Windows Versionen und Applikationen doch sehr verschieden auf den Start aus einer anderen Anwendung heraus reagieren gibt es eine Reihe verschiedener Einstellungen um das gewünschte Verhalten zu erzielen.
Prinzipiell erfolgt kann die Ausgabe einer DOS-Anwendung d.h. die Ausgabe auf STDOUT und STDERR auf drei verschiedene Arten gehandhabt werden:

Jens File Editor Description

- Die Ausgabe geschieht ganz normal über die Konsole des gestarteten Programms.
- Die Ausgabe wird in ein Datei geschrieben und nach Beendigung des Programms wird diese Datei in das Ausgabefenster gelesen.
(**Das Problem, man kann nichts sehen solange das Programm läuft, also auch nicht auf Eingabeanforderungen reagieren!**)
- Die Ausgabe wird direkt in das Ausgabefenster von JFE umgeleitet. Eingabeanforderungen können direkt vom JFE-Ausgabefenster bedient werden. (Das ist sicher die eleganteste Methode. Leider gibt es eben Programme die damit nicht zurechtkommen. Dann muss auf eine der anderen Varianten übergegangen werden. **Die Umleitung funktioniert nur für Ausgaben auf die Standardausgabestreams. Ausgaben direkt auf die Konsole können nicht umgeleitet werden!**)



Für den Start einer Anwendung gibt es drei mögliche (aber nicht immernotwendige) Parameter:

- Das ausführbare Programm (*.exe, *.bat)
- Der Pfad zum Arbeitsverzeichnis (muß nicht notwendigerweise mit dem Pfad des Programms übereinstimmen)
- Parameter zum Aufruf des Programms.

Für die Parameter gibt es einige Macros die das Leben leichter machen:

- \$FILE ...der Name des aktiven (im Vordergrund) Dokumentes **mit** Dateierweiterung
- \$Name ...der Name des aktiven (im Vordergrund) Dokumentes **ohne** Dateierweiterung
- \$EXT ...die Dateinamenserweiterung des aktiven (im Vordergrund) Dokumentes
- \$PATH ... der komplette Pfad zum aktiven Dokument
- \$CURSEL ... der gerade selektierte Text im aktiven Dokument

Die Parameter können direkt aus der Liste ausgewählt, und an der Selektion im Editfeld eingetragen werden. Wenn ein File kompiliert wurde stehen die (zweifelloos zu erwartenden) Fehlermeldungen im Ausgabefenster. Nun wäre es ja schön wenn man diese nicht erst lesen und sich dann in den geöffneten Dateien mühselig zu den Fehlern hinscrollen müßte. Auch das Problem läßt sich mit JFE lösen. Doppelclick auf die Fehlermeldung und das Dokument wird gesucht und an die Stelle gescrollt. Das Problem, jeder Compiler produziert eine andere Ausgabe. Deshalb kann man aus einer Reihe von Formaten unter "Compilerfilter" einen passenden auswählen.

(**Wenn es keinen gibt kann ich auch noch weitere einbauen!**)

Folgende Filter sind eingebaut:

Keil C		ERROR IN LINE 50 OF ..\pearl\MOPORE.P Fehlerbeschreibung
Microtec PPC		"MOPORE.P" line 123, pos 15 Fehlerbeschreibung
EMACS		C:/Tmp/pearl/MOPORE.P:123: Fehlerbeschreibung
HPC		"k1762a_c.c", line 1195: Fehlerbeschreibung
MSC/C++		C:\MSDEV-Projects\JFE\buildhtm.cpp(539): error C2146: syntax error : missing ';' before identifier 'pv1'

Jens File Editor Description

GNU166		basic.c:755: warning: left shift count >= width of type
ViewLogic		**Error: LINE 166 *** Symbol >>MAY_TX_FROM_Q<< is not !!keine Erkennung des Dateinamens möglich
Borland		Error E2268 test.c 51: Call to undefined function 'reroll' in function main(int)
Motorola DSP		**** 3556 [test.asm 556]: ERROR --- Unrecognized mnemonic: je
CADUL		CC386-E- ERROR :PLCPAPI.C: 2003: expression syntax
TASKING		W 91: no prototype for function "istm1InvIfEntry_first" E 160: pointer mismatch at '=' 333: data = istm1InvIfEntry_next_idx(data)
PC-LINT		**** LINT : c:\work\dnu02\c16_tmp\b03a6.h(99) Error 46: field type should be int" **** LINT : Info 768: global struct member 'QIPMB1' (line 1105, file t03bc.h) not referenced"
Mitsubishi NC30		[Error(ccom):monitor.c,line 27] syntax error at near 'ng_out'
LCCWin32		Error srcinden.c: 96 redeclaration of `phase6' previously declared at c:\bcx\projects\srcinden\srcinden.c

In der rechten Spalte ist ein Beispiel für die erzeugte Ausgabe angegeben. So kann man einen passenden Filter auswählen.

Rot markiert sind die Suchkriterien in der Ausgabe an Hand derer die Zeilennummer erkannt wird.

Ein Tool erscheint wenn es angelegt wurde als Button in einer eigenen Toolbarleiste. Der dort angezeigte Name ist die Bezeichnung im Eingabefeld "Name". (max 15 Zeichen). Die Tools in der Tooleiste können nachträglich per Drag-Drop sortiert werden. Der Einstellungsdialog kann zu Einstellungsänderungen mit rechtem Mausklick auf den jeweiligen Toolbutton geöffnet werden.

Es hat sich gezeigt das die Ausgabeumleitung manchmal nicht funktioniert wenn im Hintergrund ein Virensuchprogramm läuft.

Das äußert sich dann dadurch das plötzlich unmotiviert auf das Diskettenlaufwerk zugegriffen wird oder make.exe hängen bleibt.

Da hilft bloß dieses auszuschalten oder auf die Ausgabeumleitung zu verzichten.

Manche alte Tools lassen sich nicht umleiten sondern produzieren immer eine Datei eines bestimmten Namens. Da hilft es manchmal in einer

Batchdatei, nach beendigung des Tools den Aufruf : type "ausgabedatei" anzugeben. Damit erfolgt dann die Ausgabe doch in das Ausgabefenster.

Die Checkboxen haben folgende Bedeutung:

- **"Ausgabe zu JFE umleiten"**
Die Ausgabe des Programms wird direkt in das Ausgabefenster von JFE umgeleitet.
- **"Ausgabe in File Umleiten und öffnen"**
Wenn die Ausgabe nicht direkt umgeleitet werden kann (manche Programme können offensichtlich nicht in Pipes schreiben) ist es möglich die Ausgabe zunächst in eine Datei zu schreiben und nach der Beendigung diese Datei einzulesen. Das klappt in den meisten Fällen.
Wenn auch das nicht hilft kann keine der beiden Optionen eingestellt werden. Dann wird das Programm ganz normal wie aus dem Explorer gestartet.
- **"vor Aufruf der Anwendung speichern"**
Speicher alle geänderten Dateien ehe das Tool (z.B Make) gestartet wird.
- **"vor dem Start in das Arbeitsverzeichnis wechseln"**
Es hat sich gezeigt das trotz Angabe eines Arbeitsverzeichnisses manche Programme damit ihr Problem haben.
Um die nicht ganz aufklärbaren Mysterien von Windows an dieser Stelle zu umgehen kann deshalb schon vor dem

Jens File Editor Description

Start der Anwendung in das Arbeitsverzeichnis gewechselt werden.

- **"Pfadnamen kürzen"**

Damit werden lange Pfadnamen beim Aufruf durch die Windows-Kurzform ersetzt.

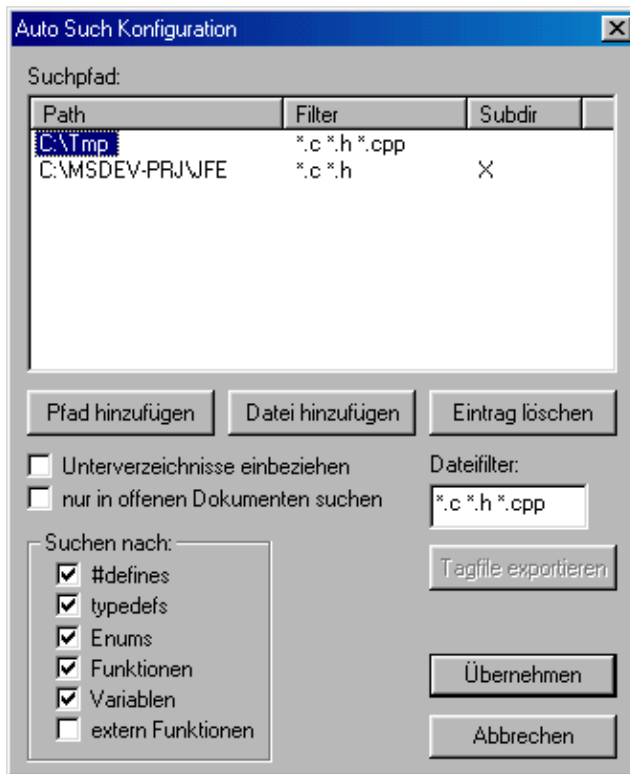
Aus : "C:\MeinToolverzeichnis\tool.bat" wird dann "C:\MeinTo~1\tool.bat"

≤

Der Autobrowser

Es ist sehr unangenehm wenn man beim Editieren und Bearbeiten größerer Projekte ständig zwischen verschiedenen Dateien hin und her springen, bzw. scrollen muß um sich Variablenamen, Typen, Definitionen Funktionsprototypen und Werte zusammen zusehen. Sofern diese Informationen innerhalb des Projektes eindeutig sind könnte diese Arbeit auch durch das Programm selbst erledigt werden. Genau das tut der Autobrowser. Wenn er gestartet wird durchsucht er alle konfigurierten Dateien oder Pfade nach diesen Informationen und baut sich eine interne Liste mit Verweisen auf die entsprechenden Stellen in den Dateien auf. Wird der Cursor in einem Dokument über einem Schlüsselwort das sich in der Liste befindet gehalten, wird ein Tooltip angezeigt. Je nach Listeninhalt hat der Tooltip nachstehenden Inhalt:

Tagtyp	Schlüsselwort	Tooltipinhalt
Definition	#define	Wert des Defines
Enum	enum	Enumname
Typdefinition	struct union	Name der Struktur bzw.Strukturmember
Funktion	(entsprechend Kontext)	Funktionsprototyp
Variable	(entsprechend Kontext)	Typ der Variable
extern Funktionen	extern	Funktionsprototyp



Die Suche kann entweder innerhalb der konfigurierten Pfade einschließlich der Unterverzeichnisse oder aber innerhalb der geöffneten Dateien durchgeführt werden.

Auch die Auswahl einzelner Dateien ist möglich.

Für jeden Pfad kann ein Filter für die einzubeziehenden Dateien angegeben werden. Bei mehreren Dateitypen werden diese mit Leerzeichen voneinander getrennt.

In dem Feld "Suchen nach:" kann ausgewählt werden welche C-Konstrukte mit in die Browserliste aufgenommen werden sollen.

Die interne Browserdatenbank kann über "Tagfile exportieren" als ASCII-Textdatei geschrieben werden. (Ev. wird man später ein solches File auch wieder importieren können)

Der Browser wird über den Button "B" in der Stauszeile oder über den Menüpunkt "Einstellungen-Auto Suche" gestartet.

Jens File Editor Description

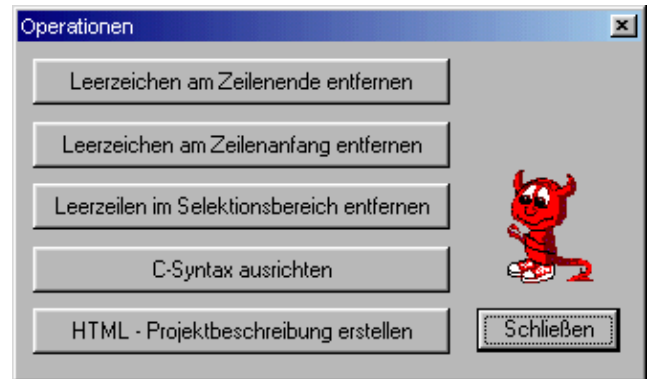
Der Browser kann auch vor Beendigung durch Click auf den rotierenden Pfeil in der Stauszeile angehalten werden.
Spätestens nach 5s sollte sich der Browserthread beenden.

Der Browserthread läuft im Hintergrund der Anwendung mit niedriger Priorität und stört deshalb nicht bei der normalen Arbeit. Wird während des Browserlaufes und Suche in den geöffneten Dateien eine Datei geschlossen, wird der Browserlauf abgebrochen.

<<

Das Operations Menü

Entfernt alle Leerzeichen am Ende einer Zeile.
Entfernt alle Leerzeichen am Anfang einer Zeile.
Der Gesamte Text wird linksbündig angeordnet.
Alle Leerzeilen werden im selektierten Bereich entfernt.
Wenn nichts selektiert ist werden alle Leerzeilen entfernt.
Richtet einen C oder C++ Quelltext so aus das eine einheitliche, gut lesbare Struktur entsteht.
Untersucht alle geöffneten Quellen und Header auf Referenzen
und erzeugt ein HTML-File in dem die gegenseitigen Aufrufe, Deklarationen und von Implementierungen der Funktionen durch Hyperlinks systematisch dargestellt werden.



Die Operationen dieses Dialogs können nicht rückgängig gemacht werden! Also besser vorher speichern.

<<

Nach welchen Regeln erfolgt die C-Syntaxausrichtung

Strukturiert C zu schreiben ist natürlich Geschmackssache. Demzufolge wird jede Form automatischer Ausrichtung von irgendwem auch wieder kritisiert werden. Deshalb folgendes Statement:

Die C-Syntaxstrukturierung realisiert meine persönliche Vorstellung davon wie C/C++ lesbar strukturiert werden sollte!

Wer's nicht mag muß es ja nicht benutzen. es bleiben aber trotzdem noch genügend Freiheiten.

Hier die Regeln nach denen ausgerichtet wird:

- nach Semikolon wird ein Zeilenumbruch eingefügt (Ausnahme danach kommt noch Kommentar)
- nach geschweiften Klammern wird Zeilenumbruch eingefügt es sei denn die schließende Klammer steht in der gleichen Zeile wie die öffnende Klammer
- jede Klammerebene wird um eine Tabweite eingerückt.
- **switch** und **case** rücken jeweils um eine Tabweite ein.
- einzeilige **if,while,else..** Anweisungen rücken die nächste Zeile um eine Tabweite ein.

Jens File Editor Description

- Marken werden linksbündig an den Seitenrand gesetzt.
- bei mehrzeilige Anweisungen in runden Klammern werden die Folgezeilen immer unter dem Beginn der ersten Anweisung ausgerichtet.
- Präprozessoranweisungen zwischen `#if...#else...#endif` werden eine Tabweite eingerückt.
- Leerzeichen am Zeilenende werden entfernt.
- Tabulatoren werden durch Leerzeichen ersetzt.
- Kommentare werden nicht verändert.

<<

Wozu ist die Operation HTML-Projekt erstellen verwendbar

Welcher Softi kennt das Problem nicht: Man bekommt einen Haufen Quellen und Headerfiles (oder man hats selbst geschrieben) und solls jetzt dokumentieren oder auch nur den Sinn des Ganzen verstehen. Und schon fängt man an nach einem Zipfel zu suchen von dem aus man das Knäuel aufwickeln kann. Üblicherweise ist der erste gegriffene Faden leider der falsche und man braucht unendliche Zeit um irgendwie Sinn in das Chaos hineinzudenken. Da wäre es doch schön wenn man eine Werkzeug hätte was einem die Arbeit abnehmen würde aufzuklären wer, wen, von wo aus und wie oft aufruft und wenigstens das mal aufschreibt. Alle Kommentare die als erstes Zeichen mit ' * ' beginnen werden auch als Funktionsbeschreibung mit in das HTML-Referenzfile übernommen. Und genau das soll dies Funktion tun. Und damit es besonders angenehm wird, werden Funktionsaufrufe gleich mit Links versehen. Damit kann man dann schön die Aufrufe per Link nachvollziehen. Darüberhinaus kann man das erstellte HTML-File sehr gut als Basis für eine Softwaredokumentation verwenden. (HTML läßt sich ja auch in Word importieren.) Soweit so gut. Allerdings gibts auch Grenzen wo die Maschinenlogik nicht mehr kann. Es lassen sich keine indirekten Aufrufe über Zeiger auflösen. Genauso unmöglich ist es Informationsflüsse über Messages zu dokumentieren. Hier muß man immer noch selbst im Code graben.

Template:

Es gibt ein voreingestelltes Template das benutzt wird um das Pro ersetzt werden.

alphabetische Funktionsliste:

listet am Beginn alle gefundenen Funktionen auf.

mehrfache Aufrufe auflisten:

Normalerweise wird eine Funktion die mehrfach aufgerufen wird r einmal dokumentiert. Wenn diese Option markiert ist werden auch grauer Textfarbe

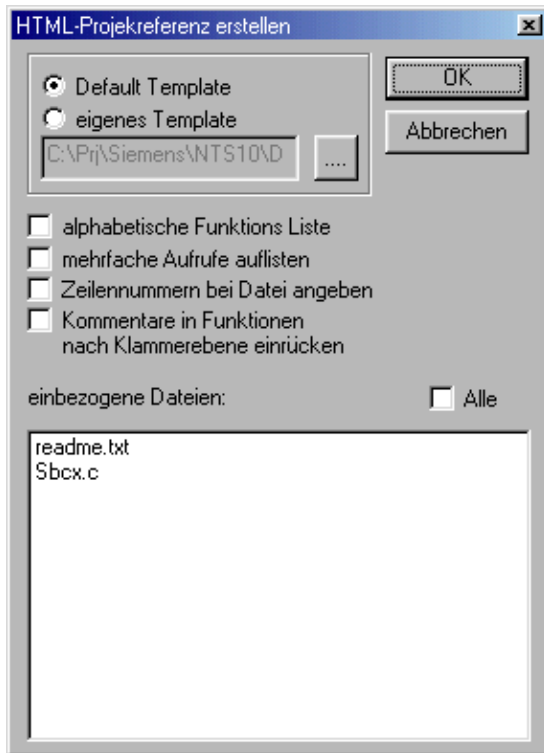
Zeilennummern in HTML-Datei angeben

Fügt nach dem Dateinamen in der die Funktion definiert ist die Ze Natürlich wäre es noch schöner wenn man den Link so gestalten angezeigt würde. Ich hab aber keine Möglichkeit dazu gefunden.

Kommentare in Funktionen nach Klammerebenen einrücken

Wenn Kommentare aus Funktionen direkt übernommen werden s Mit einiger Übung und Disziplin kann man diese Kommentare so s Text lesen läßt. Dieser Text kann nun noch zusätzlich strukturiert werden.

Jens File Editor Description



<<

Wie muss ein eigenes Template für ein HTML-Projekt aussehen.

Wie in diesem Beispiel das (fast) dem Defaulttemplate entspricht.

Wichtig ist das nach der Überschrift eine horizontale Linie eingebaut ist. Tabellen lassen sich nicht nutzen.

Es gibt einige Variablen die ersetzt werden:

\$f	der Funktionsname
\$s	der Name des Quellfiles
\$h	der Name des Headerfiles in dem die Funktion deklariert wurde
\$r	Funktionen die diese Funktion aufrufen. Wenn die aufrufende Funktion gefunden wurde wird hier ein Link eingefügt
\$c	Funktionen die durch diese Funktion aufgerufen werdn. Wenn die aufgerufene Funktion gefunden wurde wird hier ein Link eingefügt
\$p	Der Funktionprototyp der Funktion
\$dx	<p>Kommentare die in das HTML-file übernommen werden sollen. Dabei steht "x" für ein beliebiges Zeichen das als erstes nach demKommentar beginn steht (kein Leerzeichen). Damit ist es möglich auch verschiedene Informationen aus den Kommentaren in der HTML-Beschreibung zu separieren</p> <p>Beispiel: \$d# übernimmt alle Kommentare die mit # beginnen</p> <p>/*# dieser Kommentar wird genommen auch diese Zeile bis hier */</p> <p>///auch das kommt mit in das HTML-file</p>

Wenn das File erstellt wurde kann es natürlich weiter editiert werden, zum Beispiel durch Einfügen von Struktogrammen und verbaler Beschreibung. Ein erneutes Erstellen überschreibt solche Änderungen aber wieder.

Jens File Editor Description

Im Header des erstellten HTML-Files (also vor der ersten Funktionsbeschreibung) können auch noch Informationen aus die aus Kommentaren der jeweils einzelnen Dateien extrahiert wurden gesammelt werden. Das könnten zum Beispiel Versionskennungen usw. sein. Da es keine allgemein übliche oder notwendige Schreibweise dafür gibt, habe ich versucht es so universell wie möglich zu gestalten.

Was muß man tun:

Angenommen man hat in jeder Datei einen solchen Dateikopf:

```
/*
// CLASS: < ISDNPORT >
// CLASS ID:
// FILE NAME: isdnPort.h
// VERSION: 3.2.00
// FUNCTION: main function for anything
// REMARKS:
//-----*/
// Copyright © 1999 */
/*-----*/
// AUTHOR/S: Jens Altmann (TCD) */
//-----*/
// CREATION DATE: 02.11.99
//-----+-----+-----+-----*/
// VER | DATE | CHANGES | AUTHOR */
//-----+-----+-----+-----*/
//
/*
```

Nun hätte man gerne die Informationen über **Autor**, **Version** und **Erstellungsdatum** jeder der einbezogenen Dateien in den Kopf der zu erstellenden HTML-Datei übernommen.

Man sucht sich im Quellfile einen Kenner für die Information die man übernehmen möchte.
Also :

Autor AUTHOR/S:
Version DATE:
Erstellungsdatum VERSION:

Dieser Merker darf **keine** Trennzeichen (Leerzeichen, Tabs, Zeilenumbrüche) enthalten!
Für den Dateinamen und den Arbeitsbereich gibt es vordefinierte Merker:

Datei \$f
Arbeitsbereich \$prj

Man schreibt das Template. Dazu ist es unbedingt wichtig den Bereich in dem diese Informationen stehen sollen durch den Kommentar `<!--prjinfo-->` am Anfang und Ende einzugrenzen. Dieses Stück HTML-code wird dann für jede Datei kopiert und mit den entsprechenden Informationen gefüllt.

Die im Quellfile zu suchenden Merker werden im Template mit vorangestelltem \$ definiert.

Also:

\$VERSION:
\$AUTHOR/S
\$DATE:

Jens File Editor Description

Man speichert ab und erstellt das HTML-File. JFE sucht zuerst nach dem Kommentarbereich. Wenn dieser existiert werden all mit \$ markierten Worte gesucht. Wenn ein solches Wort gefunden wird, werden alle Dateien nach diesem Wort (ohne \$) durchsucht. Wenn es in einer Datei innerhalb eines Kommentars gefunden wird, wird der unmittelbar danachfolgende Text bis zu Zeilenende in das Template anstelle des Merkers eingetragen.

Klingt ziemlich verwirrend. Am besten man sieht sich mal das [Beispieltemplate](#) und seinen Quelltext an.

<<

Arbeitsbereiche

Arbeitsbereiche sind eine bequeme Einrichtung wenn man in mehreren Dateien arbeitet und nicht jedesmal beim Starten alle Files wieder einzeln öffnen will.

In einem Arbeitsbereich werden alle wichtigen Einstellungen abgespeichert wie:

- geöffnete Dateien einschließlich Cursorposition und Fensterzustand
- Syntaxcoloring –Einstellungen
- Tooleinstellungen
- Schriftart und spezielle Einstellungen

JFE verwaltet auch eine Liste der zuletzt geöffneten Arbeitsbereiche. Ein Arbeitsbereich wird automatisch beim Beenden gespeichert.

Der Name eines Arbeitsbereiches (Projektes) ist in einem [Macro](#) über die Variable \$PRJ verwendbar.

Man kann sich auch eine [Verknüpfung im Explorer](#) einrichten, so das Projekte auch direkt mit Doppelclick im Explorer geöffnet werden können.

<<

Das Kontextmenü

- Was macht "Kopieren++"
Es fügt zur Zwischenablage den zu kopierenden Text mit einem Zeilenumbruch an. Dadurch kann man sozusagen Text in der Zwischenanlage aufsammeln.
- Wechsel zum File Directory
Führt ein *chdir* in das Verzeichnis aus in dem das gerade aktive Dokument steht.
- Groß/Kleinschreibung, Kapitälchen
Großschreibung – wandelt den markierten Text in Großbuchstaben um
Kleinschreibung – wandelt den markierten Text in Kleinbuchstaben um
Kapitälchen – setzt den ersten Buchstaben jedes Wortes auf Groß-, den Rest auf Kleinbuchstaben
- Suche Klammer
Sucht von der angegebenen Cursorposition die zugehörige { } –Klammerebene und markiert diese ohne zu scrollen. Der Menüpunkt ist nur freigegeben wenn nichts markiert ist!
- Gehe zu Klammer
Springt mit dem Cursor zur zugehörigen Klammer. Das funktioniert nur wenn der Cursor neben einer öffnenden oder schließenden Klammer steht. Welche Art von Klammer, rund, eckig oder geschweift ist egal es wird die passende gesucht. Bei Bedarf wird die Sicht gescrollt. Der Menüpunkt ist nur freigegeben

Jens File Editor Description

wenn nichts markiert ist!

Wenn gleichzeitig die SHIFT gedrückt wird, so wird der Bereich zwischen den Klammern markiert!

- **Macro Einfügen**

Öffnet ein Submenü in dem die vorhandenen (max 20) Textbaustein-Makros ausgewählt werden können.

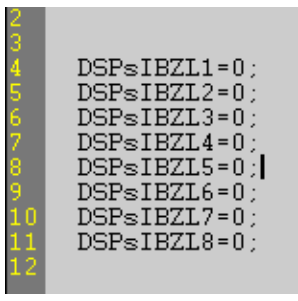
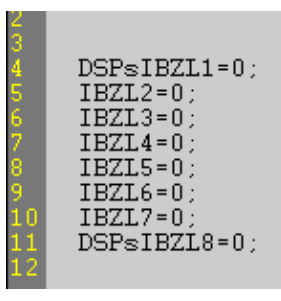
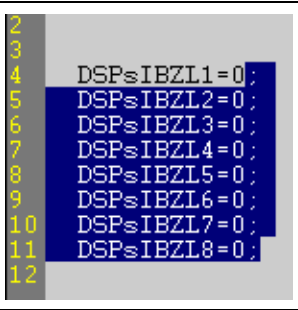
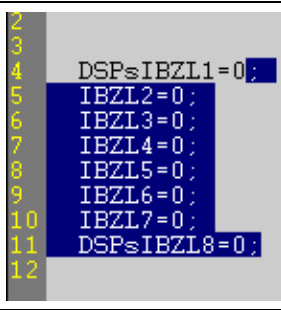
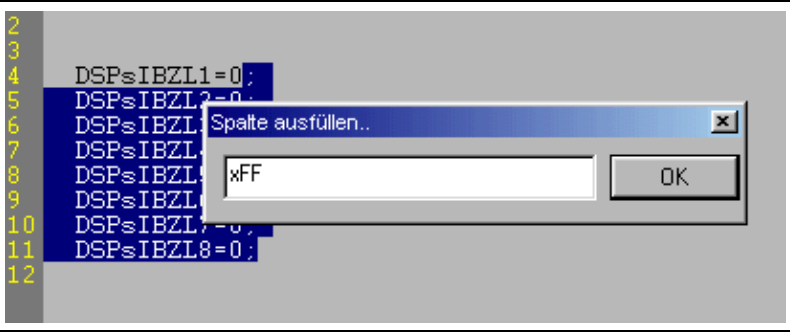
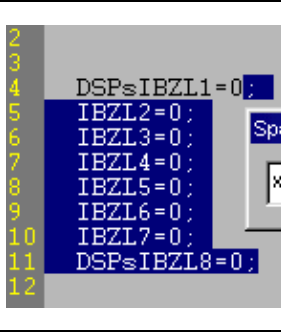
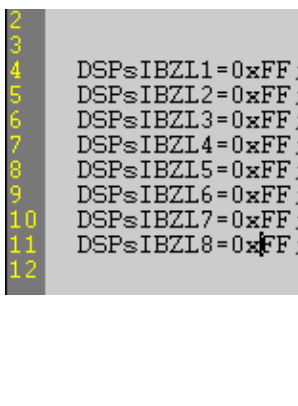
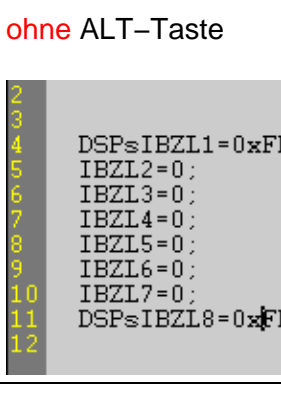
- **Macros aktualisieren**

Ersetzt sofort alle aktualisierbaren Makrovariablen. Normalerweise passiert das erst beim Speichern des Dokumentes

- **Spalte ausfüllen**

Diese Funktion ermöglicht es innerhalb eines selektierten Bereiches beginnend von der Spalte an der die Markierung beginnt die folgenden selektierten Zeilen in der Spalte mit einem String zu füllen:

Das geht so:

<p>1. Der Text soll so aussehen und man will vor dem Semicolon den String "xFF" einfügen</p>		
<p>2. Den gewünschten Bereich markieren</p>		
<p>3. Kontextmenü "Spalte ausfüllen" wählen und in dem Dialog "xFF" eingeben</p>		
<p>4. "OK" Abhängig vom Text und der ALT-Taste ergibt sich folgendes Ergebnis:</p>		<p>ohne ALT-Taste</p> 

			<div style="border: 1px solid black; padding: 5px;"> <p style="color: red;">mit gedrückter ALT-T</p> <pre> 2 3 4 DSP=IBZL1=0xFF 5 IBZL2=0;xFF 6 IBZL3=0;xFF 7 IBZL4=0;xFF 8 IBZL5=0;xFF 9 IBZL6=0;xFF 10 IBZL7=0;xFF 11 DSP=IBZL8=0;FF 12</pre> </div>
--	--	--	--

• Einrücken

Wie oft hat man doch das Problem einen Bereich einrücken zu müssen und fängt jetzt mühselig an jede Zeile einzeln mit Leerzeichen zu füllen. JFE hat dafür auch eine Lösung:

1.	<p>Markiere den zu verschiebenden Bereich. Der Anfang muß dabei auf dem ersten zu verschiebenden Zeichen stehen.</p>	<pre> 54 #define TSD6_A 59 55 56 #define TSD1_S 34 57 #define TSD2_S 35 58 #define TSD3_S 40 59 #define TSD4_S 41 60 #define TSD5_S 42 61 #define TSD6_S 43 62 63 #define TSU1_A 54 64 #define TSU2_A 55</pre>
2.	<p>Click auf rechte Maustaste und wähle Einrücken. Vor der einzurückenden Spalte wird ein roter Strich gezeichnet. Das Cursorsymbol ändert sich. (mit ESCAPE kann abgebrochen werden)</p>	<pre> 54 #define TSD6_A 59 55 56 #define TSD1_S 34 57 #define TSD2_S 35 58 #define TSD3_S 40 59 #define TSD4_S 41 60 #define TSD5_S 42 61 #define TSD6_S 43 62 63 #define TSU1_A 54</pre>
3.	<p>Verschiebe mit der Maus (gedrückte linke Taste) die rote Linie nach links bzw. recht an die gewünschte Position. Dann lasse die linke Maustaste los. Der Text steht nun, immer noch markiert in der neuen Spalte.</p>	<pre> 54 #define TSD6_A 59 55 56 #define TSD1_S 34 57 #define TSD2_S 35 58 #define TSD3_S 40 59 #define TSD4_S 41 60 #define TSD5_S 42 61 #define TSD6_S 43 62 63 #define TSU1_A 54</pre>

<<

Makros

Makros können das Leben leichter machen! Es gibt bei JFE zwei Möglichkeiten Makros zu definieren und zu benutzen. Einmal können Makros als Textbausteine eingefügt werden oder auch als Tastaturmakros die Tastaturanschläge aufzeichnen.

Textbaustein-Makros

Jens File Editor Description

JFE ermöglicht es bis zu **20** beliebige Textbaustein-Makros aus einer externen Datei zu laden. Diese Datei muß "macro.txt" heißen und im gleichen Verzeichnis wie "jfe.exe" stehen. Macros werden zugänglich über das Kontextmenü. Ein Macro kann ein beliebiges Stück Text sein und auch ersetzbare Variablen enthalten. Das Format um ein Macro zu definieren ist folgendes: (Beispiel)

```
$macro Switch  
switch( )  
{  
  case:      break;  
  
  case:      break;  
  
  default:   break;  
  
}  
$endmacro
```

Ein Macro beginnt immer mit der Anweisung **\$macro** und endet mit **\$endmacro**. Damit können die Macros in der Datei auseinander gehalten werden. Was dazwischen steht ist dann der Macrotext. Der Name der nach **\$macro** in der gleichen

Zeile folgt ist der Name unter dem das Macro im Kontextmenü erscheint.

Nun gibt es vordefinierte Variablen die beim Einfügen eines Macros erst aktualisiert werden:

\$dd	Das aktuelle Datum im deutschen Format dd.mm.yy
\$de	Das aktuelle Datum im englischen Format mm/dd/yy
\$FILE	Der Filename ohne Pfad des aktuellen Dokumentes.
\$USER	Der im Windows eingeloggte Username.
\$PATH	Der komplette Pfadname des aktuellen Dokumentes.
\$we	Der abgekürzte Wochentag in Deutsch.
\$wd	Der abgekürzte Wochentag in Englisch.
\$PRJ	Der Name der Arbeitsbereichs (Projekt). Ist kein Arbeitsbereich vorhanden wird die Variable nicht ersetzt.
\$t	Die aktuelle Zeit im Format hh:mm.

Diese Variablennamen können in den Textmakros eingebaut sein und werden zum Zeitpunkt des Einfügens durch ihre Werte ersetzt.

Eine Spezialität sind aktualisierbare Makros. Was soll das sein? Normalerweise werden Variablen beim Einfügen des Makros ersetzt

und damit ist der Text festgelegt. Nun wäre es manchmal wünschenswert das ein Makro bei jedem Speichern einer Datei aktualisiert würde.

Beispielsweise um immer die letzte Änderung einer Datei zu dokumentieren. Nun ist ein Makro, einmal eingefügt ja nicht mehr als solches

zu erkennen. Man muss sich also ein Merkmal künstlich schaffen. Und das geht so: (Beispiel)

```
$macro $LETZTE AENDERUNG  
: $dd /$t $USER  
$endmacro
```

Dem Macronamen wird ein '\$' vorangestellt. Beim Speichern wird der Text nach allen Makronamen die mit \$ beginnen durchsucht

und der Text um den Macrotextersatz verlängert.

Im Beispiel würde das bedeuten das wenn der Text den String "**LETZTE AENDERUNG**" enthält, dieser (beispielsweise) durch:

Jens File Editor Description

LETZTE AENDERUNG: 05.05.99 /20:23 altj ersetzt würde. Und das bei jedem Speichern.

Das Verhalten der Variablen beim Expandieren ist unterschiedlich. Die Variablen die beim Expandieren eine unvorhersehbare Länge haben überschreiben die nachfolgenden Leerzeichen (! nur solche). Die Variablen die eine feste Länge haben(Zeit,Datum..) werden eingefügt und der danach liegender Text nach hinten geschoben.

Tastaturmakros

Tastaturmakros müssen vor der Benutzung aufgezeichnet werden. Dazu startet man über das Menü: "Bearbeiten-Macro aufzeichnen" den Macrorekorder. Die Bedienung ist leicht verständlich. Nach dem Click auf den roten Aufzeichnungsbutton werden alle Tastaturbetätigungen und ausgewählte Kommandos aufgezeichnet. Nach Beendigung der Aufzeichnung (Taste-dunkles Rechteck) kann das Makro als Datei mit der Endung ".mcr" gespeichert werden. (Diese Datei ist binär und kann nicht als ASCII -Text gelesen werden.) Gleichzeitig wird dem Benutzer angeboten das Makro als neues Tool in der Toolleiste anzulegen. Benutzt wird es dann auch wie ein solches. Ein Click auf den entsprechenden Button startet das Makro, d.h. alle aufgezeichneten Tasten und Kommandos werden **an der aktuellen Position !!** des Cursors eingefügt bzw. abgearbeitet. Man muß sich also sehr genau überlegen was man erreichen will.

Neben den Tastatureingaben werden auch noch folgende Kommandos aufgezeichnet:

Ausschneiden
Kopieren
Kopieren ++
Einfügen
Zeile Löschen
Suchen (einschließlich Suchtext und Suchflags)
Suche passende Klammer
Selektiere Klammerebene
Großschreibung
Kleinschreibung
An Port senden
Von Port empfangen
Kapitalchen

<<

Funktionstasten

Ich selbst benutze Hotkeys kaum aber es wurde immer wieder gewünscht. Da aber jeder eine andere Tastaturbelegung bevorzugt, kann sich nun jeder seine Tastaturbelegung selbst definieren. (Menü Tastaturbelegung..). Voreingestellt sind die folgenden Hotkeys.:

CTRL-C	Kopieren
CTRL-Einfg	
CTRL-X	Ausschneiden
CTRL-Entf	
CTRL-V	Einfügen
SHIFT-Einfg	

Jens File Editor Description

CTRL-Z	Rückgängig
CTRL-I	Markierten Bereich eine Spalte nach rechts einrücken
CTRL-U	Markierten Bereich eine Spalte nach links einrücken
CTRL-B	Markierten Block eine Spalte nach rechts einrücken
CTRL-M	Markierten Block eine Spalte nach links einrücken
CTRL-F	Such – Dialog öffnen
CTRL-R	Suchen/Ersetzen –Dialog öffnen
F3	Weitersuchen
CTRL-G	Gehe zur Zeile..
CTRL-Y	Lösche Zeile
CTRL-F9	Tool starten
F12	Lesezeichen setzen/löschen
CTRL-P	Drucken
CTRL-N	Neues Dokument
CTRL-S	Speichern
CTRL-O	Öffnen
F11	Speichern und Beenden ohne zu fragen
CTRL+F11	ohne Rückfrage Beenden ohne zu Speichern
F6	nächstes Dokument

<<

Druckeinstellungen

JFE verfügt natürlich auch über eine ausgebaute Druckfunktionalität. Im Gegensatz zu anderen Editoren kann auch ein farbiger Ausdruck erzeugt werden. Voraussetzung ist das das Syntaxhighlighting eingeschaltet wird. Über die Seitenansicht

kann das Layout kontrolliert werden. Der eingestellte Schriftfont wird auch für den Ausdruck verwendet.

Um den Druck individuellen Wünschen anzupassen ist dem Druckdialog ein spezieller Einstelldialog vorangestellt.

Jede Seite kann ein Banner erhalten. Dieses kann oben oder unten auf der Seite angeordnet sein. Das Banner kann enthalten:

- den Dateinamen
- das aktuelle Datum
- die Seitennummer

Es ist möglich auch nur eine Auswahl von Zeilen des Dokumentes zu drucken. Um Platz zu sparen kann mit der Option "enge Zeilen" der Zeilenzwischenraum auf das minimal möglich Maß verkleinert werden.

Es ist auch möglich nur einen Bereich von Zeilen zu drucken.

Zum Drucken kann unabhängig vom Bildschirmfont eine eigene Schriftart gewählt werden.





Anbindung von anderen Anwendungen über TCP-Ports

Der Toolbar zum Aufzeichnen von Makros enthält zwei zusätzliche Buttons zum Versenden und Empfangen von Text über

TCP-Ports. Damit ist es möglich externe Filter über Makros anzubinden.

Der Button "Sende zum Port" sendet den Inhalt der Zwischenablage an das gewählte Serverport.

Der Button "Lesen vom Port" löscht die Zwischenablage, wartet auf Daten vom eingestellten Port und kopiert in die Zwischenablage.



Sonstiges

- **Aufruf des Editors mit Arbeitsbereichen** (*.wsp)
Das ist möglich. Damit eine Datei mit der Erweiterung "wsp" aber selbst auch editierbar bleibt muß der Aufruf folgendermaßen erfolgen: **jfe.exe -wYourWorkspace.wsp**
- **Aufruf des Editors mit der Zeilennummer**
der Aufruf muß folgendermaßen aussehen: **jfe.exe Filename -Zeilennummer**
- **Pfad zu diesem Hilfefile lokal setzen**
Dazu findet man in jfe.ini den Eintrag :
HelpPath=http://home.t-online.de/home/Jens.Altmann/jfe.htm
Dieser kann beliebig umgesetzt werden.
- **Exportieren einer Datei als HTML-Dokument**
Manchmal möchte man ein Dokument mit der Farbinformation abspeichern.
Ein anwendungsbeispiel wäre z.B. ein großes Protokollfile indem man bestimmte Einträge farblich hervorgehoben hat. Dann eignet sich die HTML-Exportoption gut um ein solches Dokument anderen zugänglich zu machen. Sicher wäre auch RTF eine Variante gewesen aber HTML ist sicher noch verbreiteter.
Die Darstellung erfolgt wirklich so wie im Fenster des Editors. D.H. auch Zeilennummern, Hintergrundfarbe, Font und Syntaxhighlighting. Dabei ist zu bedenken das der Browser deutlich langsamer wird wenn in jeder Zeile mehrere Farbwechsel erfolgen!! Also besser die Zeilennummerierung ausschalten.

