

AtmelStudio einrichten

Eine Beschreibung zum Einrichten des AtmelStudios 6 findet sich z.B. auf: [Atmel Studio - Mikrokontroller.net](#).

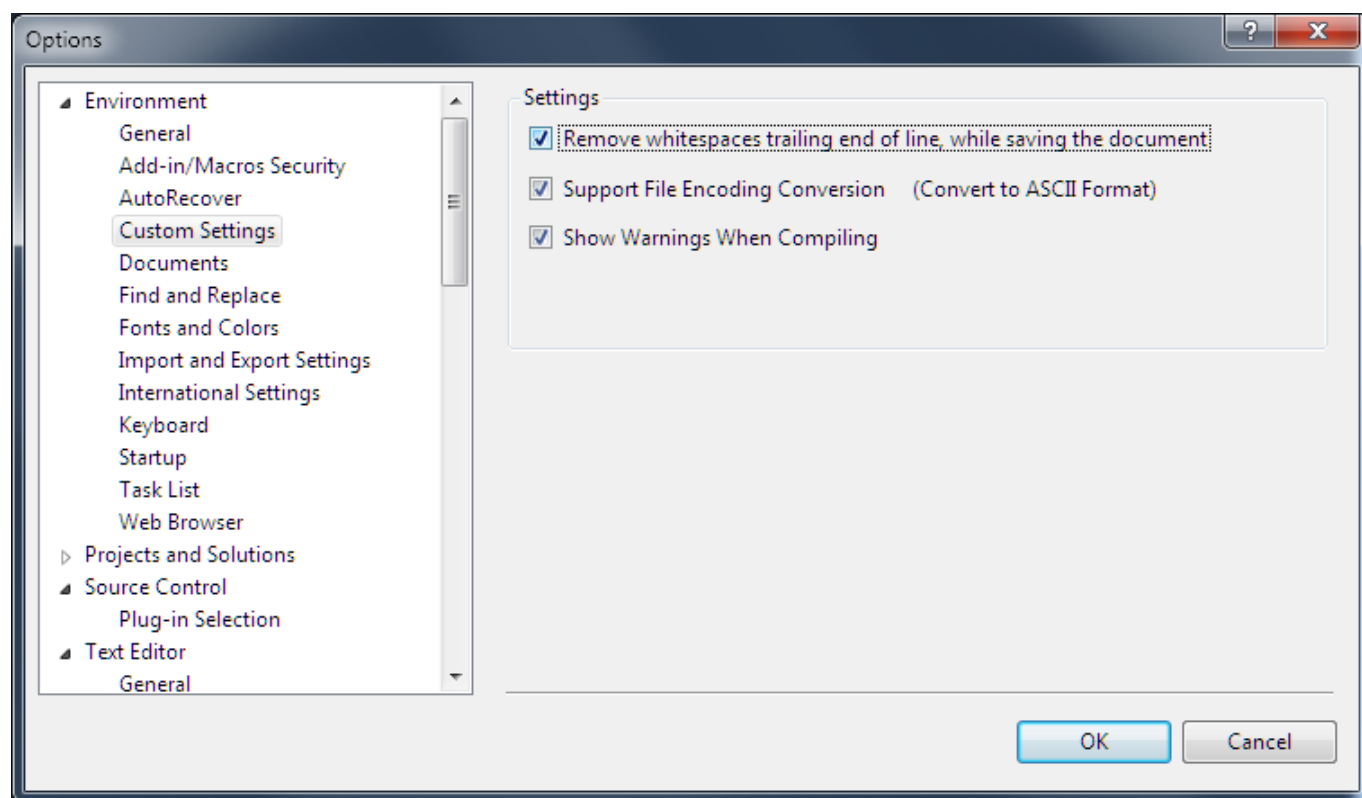
Das mittlerweile erschienene AtmelStudio 7 wird in Kürze (Stand 9.1.16) ebenfalls unterstützt.

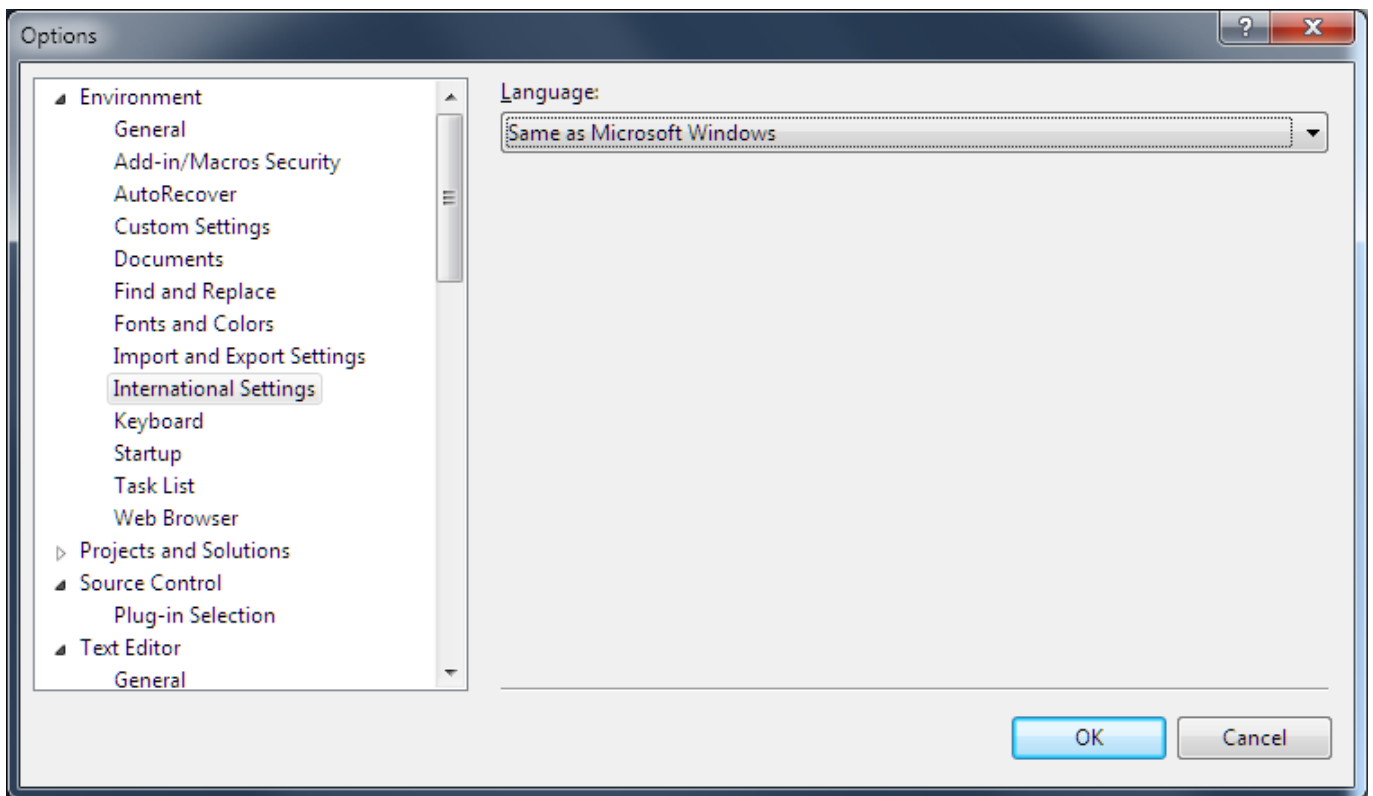
Weitere Features für das AtmelStudio können über den [AtmelStudio Extension Manager](#) eingerichtet werden.

Dieser Artikel enthält Hinweise - am Beispiel AtmelStudio 6, die das BiDiBOne-Projekt betreffen.

Einstellungen

Tools | Options

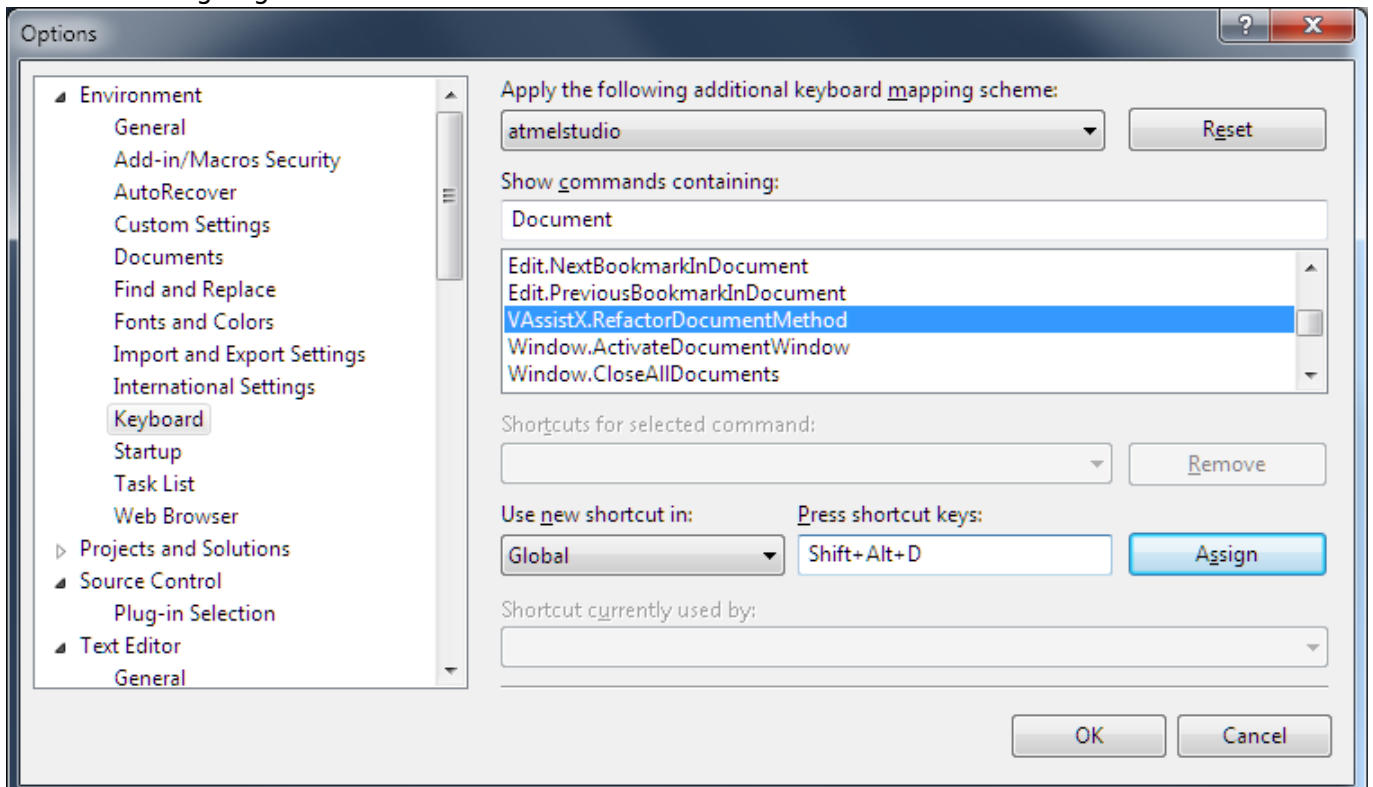




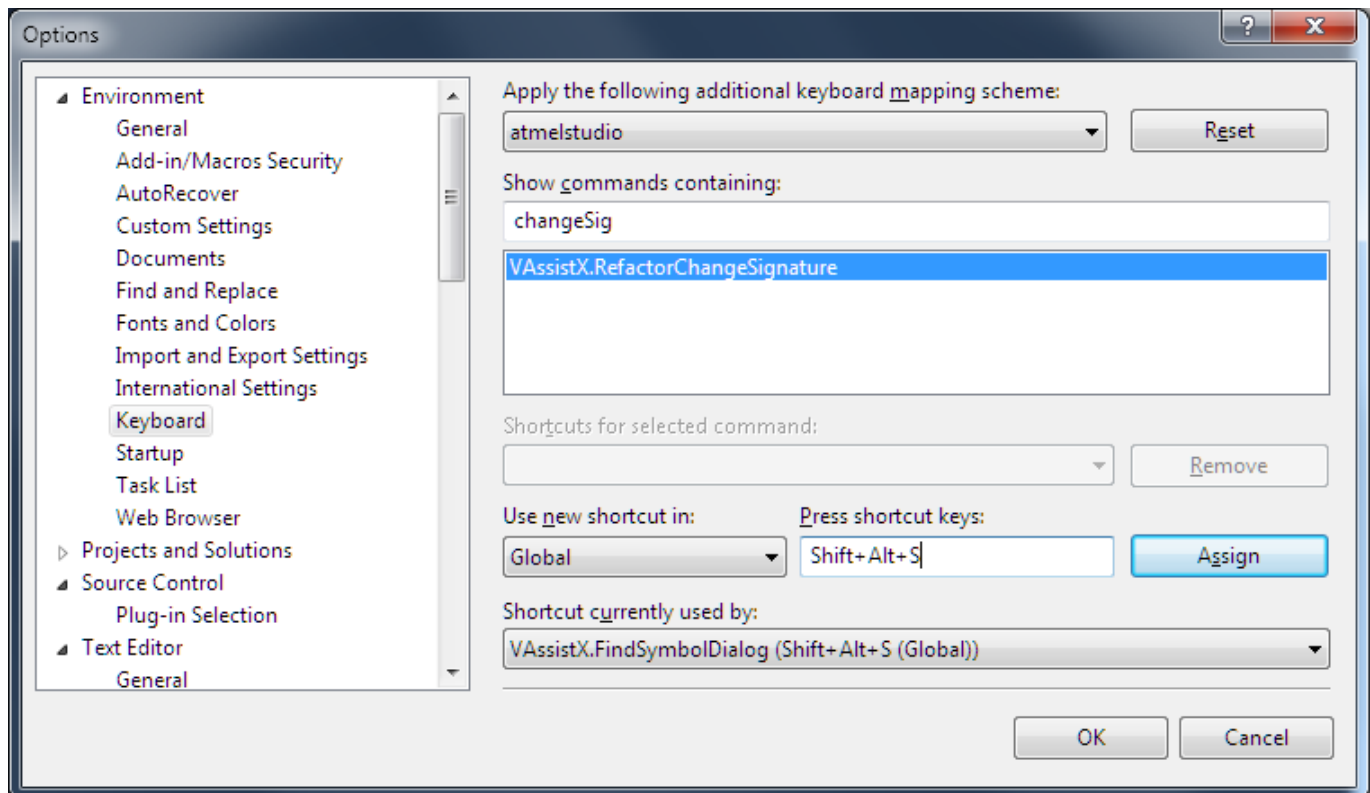
Tastaturkürzel

(VAssistX.RefactorRename: Shift+Alt+R ⇒ schon definiert)

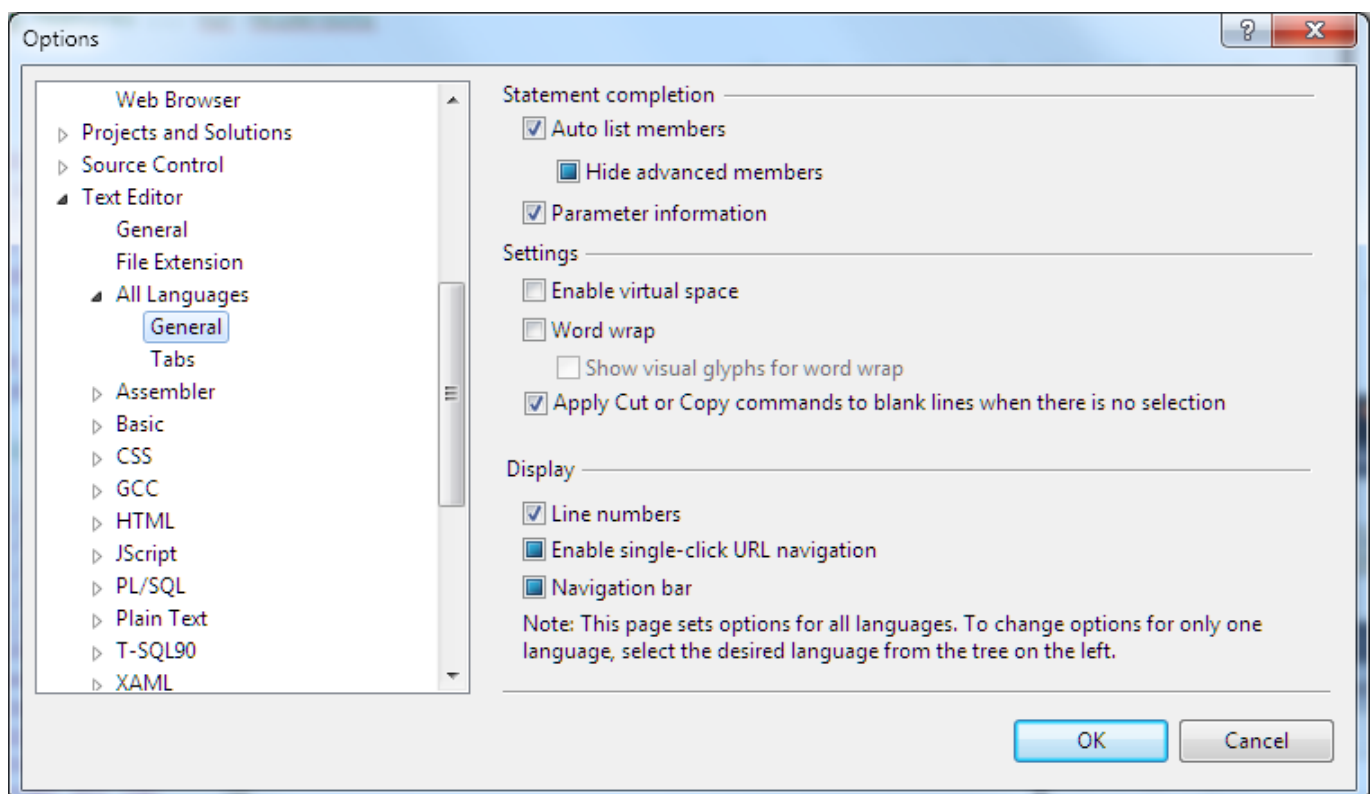
VAssistX.ChangeSignature: Shift+Alt+S



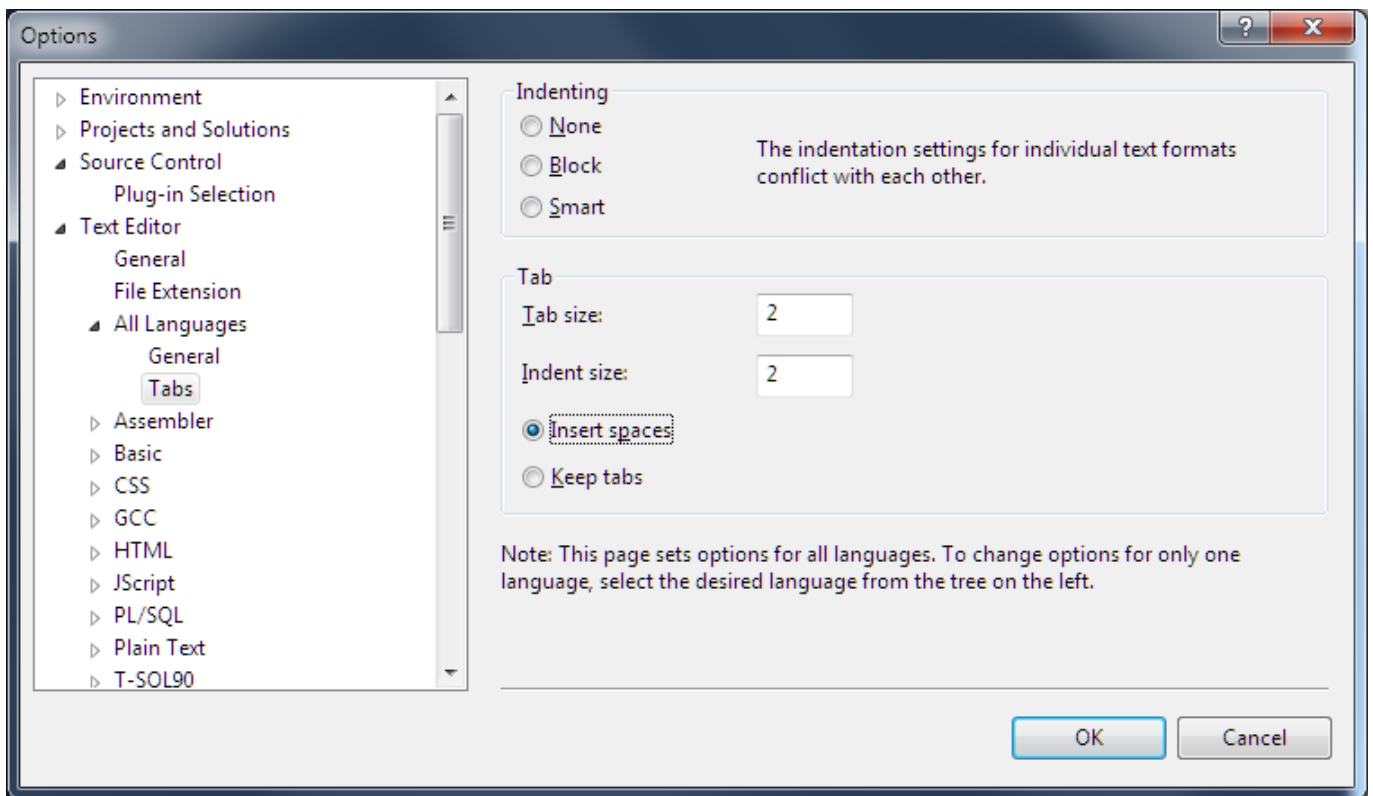
VAssistX.RefactorDocumentMethod: Shift+Alt+D



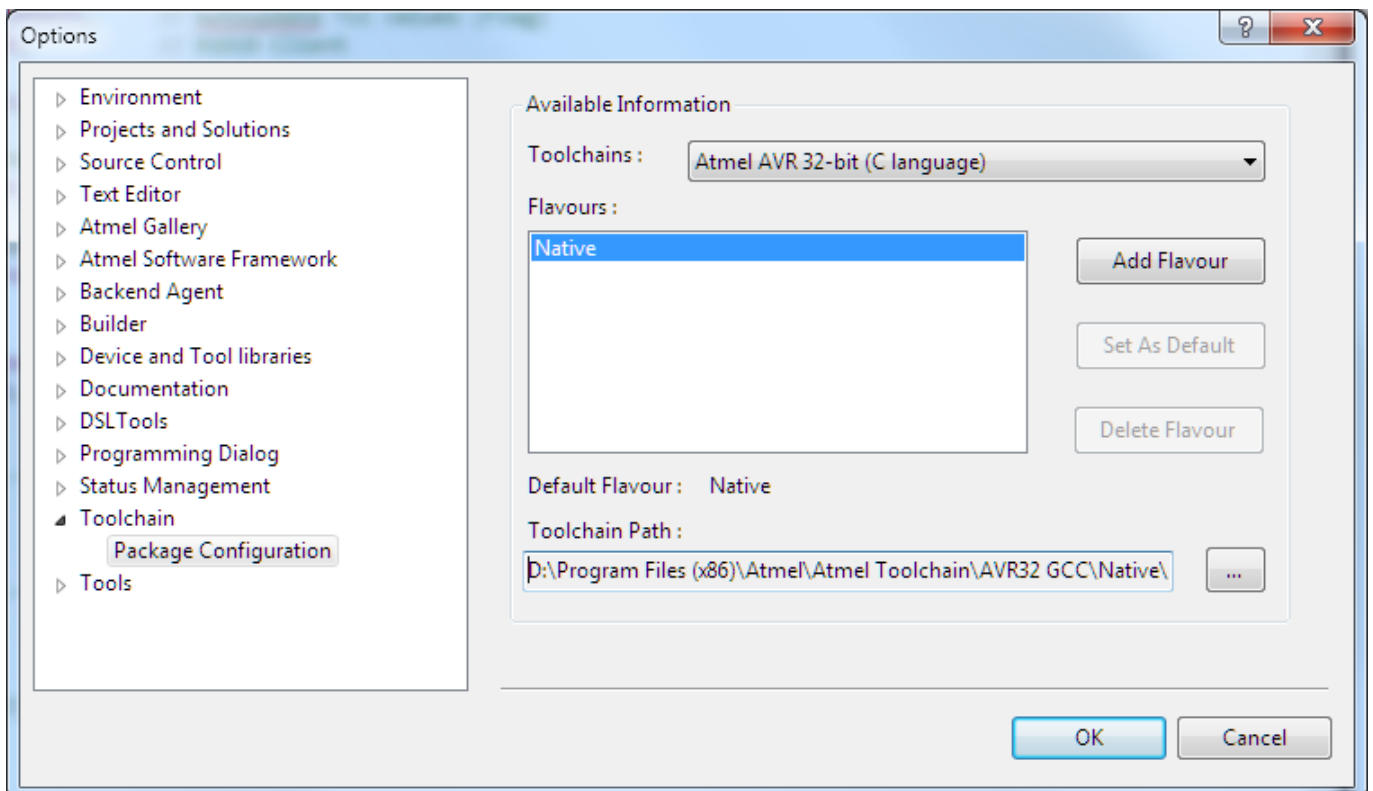
Zeilennummern



Tabulatoren und Einzüge



Pfad zu Compiler und Linker



(z.B. D:\Program Files (x86)\Atmel\Atmel Toolchain\AVR32 GCC\Native\3.4.2.1002\avr32-gnu-toolchain\bin)

Toolchain

Projekt anwählen, Kontext-Menü: PROJEKT-Properties – Toolchain

Abhängig vom Projekt die Eigenschaften einstellen.

Modulinformationen

VAssistX | Tools | VA Outline

Zeigt die „Innereien“ eines geladenen Moduls an, wie #defines und Funktionen.

FUSES

(siehe übergeordnetes Kapitel)

Item Templates

(siehe übergeordnetes Kapitel)

Extensions

Via Tools | Extension Manager ...

Beachte: Atmel-Konto notwendig.

Versionskontrolle

Leider unterstützt das AtmelStudio (hier Version 6 und 7) nicht mehr die Versionierungssystem Git. Deshalb muss auf ein externes Werkzeug zur Versionsverwaltung verwendet werden.

[Installation und Anmeldung bei GitLab sind in](#)

Installation Git

[und](#)

Git Anmeldung

beschrieben. **Beachte: Die Hinweise zum AtmelStudioX in diesen Dokumenten sind obsolet!**

Dokumentation

Zur Dokumentation oder zum Lesen dokumentierter Quellen bietet sich Doxygen an.

- laden, neu starten
- Konfiguration: Tools | Options | Documentation | Doxygen Settings:

- Doxygen Executable Path ==> z.B. D:\Dokumente\Michael\Anwendungsdaten\Doxygen
- Dokument erzeugen: Project | Doxygen ...
 - Config File Editor - Edit ==> z.B. OUTPUT_LANGUAGE = German
 - ...
 - Save Settings
 - Generate → als HTML- und TeX-Dokument nutzbar

Tipps

Spickeln

Atmel stellt in seinem Atmel Software Framework (ASF) Module für Teillösungen zur Verfügung, die in die eigene Entwicklung einbezogen werden können. Diese Module sind in der Regel sehr gut dokumentiert und beschreiben die verwendeten Techniken. Lädt man interessante Module z.B. in einem separaten Projekt und lässt die Dokumentation mit Doxygen erstellen, erhält man eine prima API-Beschreibung.

Zugriff auf Systemdaten gestatten

Bestimmte Security-Software blockiert standardmäßig den Zugriff auf Systemdateien etc. Dieser Zugriff muss für manche Applikationen aus dem Atmel Studio 6 explizit gewährt werden:

- arm-none-eabi-gcc.exe
- cc1.exe
- as.exe
- collect2.exe
- id.exe
- arm-none-eabi-objcopy.exe
- arm-none-eabi-objdump.exe
- arm-none-eabi-size.exe

Einige Funktionen werden erst im Laufe der Zeit bzw. beim Debugging aufgerufen.

Macken

Auch das beste Werkzeug hat in der Regel die eine oder andere Macke.

Start Debugging (F5) bei mehreren Projekten

(Stand: 9. Dez. 2014)

Legt man unter AtmelStudio 6 (Version: 6.2.993 - Beta) eine so genannte Solution mit mehreren gleichrangigen Projekten an, kann man beim Starten des Debug-Modus' gehörig gegen den Poller laufen. (Andere Versionen des Studios wurden nicht explizit getestet.) Das Studio führt zwar ein

Makefile für das aktuell angewählte Projekt aus, was auch im Output schön zu sehen ist. Allerdings wird immer das zuletzt durch Rebuild gebaute Projekt geladen und nicht das angezeigte.



Und was noch gemeiner ist: Selbst beim expliziten Laden über Tools | Device Programming ... werden die zuletzt durch Rebuild gebauten Dateien ausgeführt!

Workaround: Vor dem Debugging immer Rebuild zu-testendes-Projekt ausführen. Dann kann man mit F5 das Programm laden und starten. Alternativ alle zur Zeit unbenutzten Projekte „entladen“ und erst nach Fertigstellung wieder laden.

Unit Tests

Atmel bietet mit den ASF ein Testframework (z.B. Unit test framework (driver)) an.

Für Unit Test ist ein separates Projekt in der Solution anzulegen.

Die Tests werden über eine Test-Suite aus der main-Funktion dieses Projektes gestartet. Dieser Suite werden die gewünschten Testfälle bekannt gegeben.

Die Ausgabe der Testergebnisse erfolgt über Wrapper. Denkbar wäre z.B. dbg_info, das die pc_send-Funktionen nutzen könnte.

(Note: The test suite framework feeds its output to printf. It is left up to the test application to set up the stream.)

From:
<https://forum.opendcc.de/wiki/> - BiDiB Wiki

Permanent link:
https://forum.opendcc.de/wiki/doku.php?id=bidiboneentwicklungsumgebung:atmel_studio&rev=1452341775

Last update: 2016/07/05 10:52

