

Text als Resource

Ziel, Inhalt

- Mit dieser Anleitung kann man Text in einer Resource speichern und mit einer Klasse TextResource verwenden

Text als Resource	1
Ziel, Inhalt	1
Text als Resource	2
Was sind Ressourcen	2
Ressourcen in Konsolenapplikationen	2
Anleitung	2
Einfügen der Resource	2
Speichern der Resource	3
Ändern der Ressourceneigenschaften	5
Editieren des Textes	5
Die Klasse ResourceText	5
Behandlung von Ressourcen	5
Einfügen aller nötigen Dateien	6

Text als Resource

Diese Datei findest du zum Download auf:

www.devmentor.ch/teaching/additional/TextResource/TextResource.pdf

Beispielcode mit der Klasse TextResource findest du auf:

www.devmentor.ch/teaching/additional/TextResource/TextResource.zip

Was sind Ressourcen

Programme bestehen im Normalfall aus Code. Ein Programm enthält aber auch weitere „Dinge“ wie Bitmaps oder Icons, die das Programm zur Anzeige braucht. Diese speziellen Daten sind keine Programmdateien befinden sich aber auch in der Exe-Datei. Solche Daten nennt man Ressourcen.

Ressourcen in Konsolenapplikationen

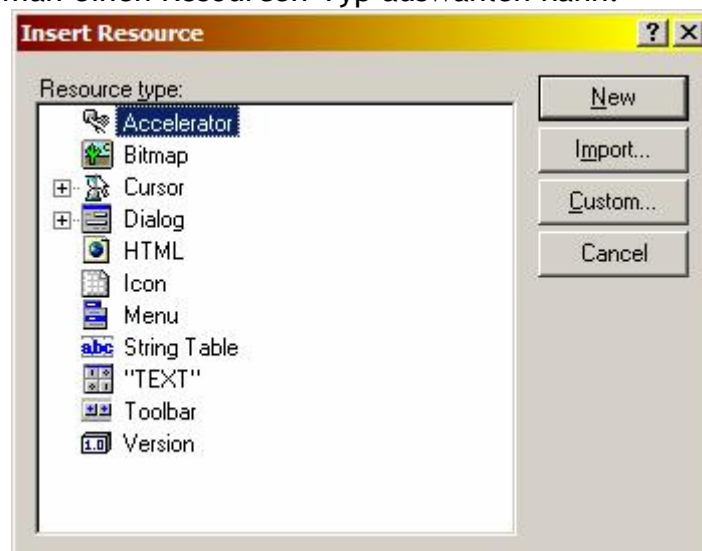
Ressourcen werden üblicherweise von Windows-Applikation verwendet, also Applikationen die Fenster, Menus etc. haben. Es ist jedoch auch möglich in einer Konsolenanwendung Ressourcen zu verwalten. In unserem Fall wollen wir einfach einen Text als Resource in der in der exe Datei „verstecken“.

Anleitung

Diese Anleitung gilt für das Visual Studio 6. Mit Visual Studio .NET kann man aber ähnlich vorgehen.

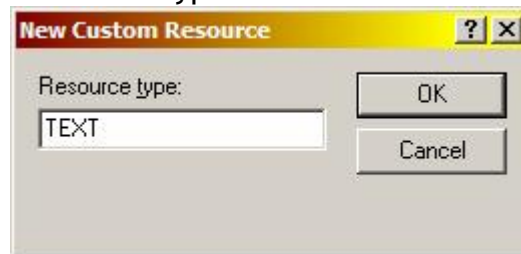
Einfügen der Resource

Wähle im Menu „Einfügen“ den Menüpunkt Resource (Ctrl + R). Ein Dialog erscheint wo man einen Ressourcen-Typ auswählen kann.

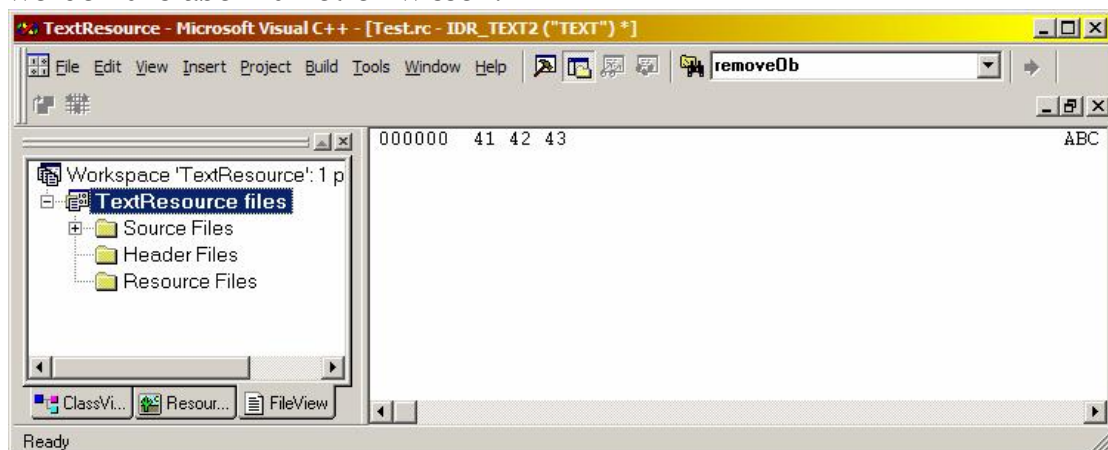


In der englischen Version musst du nun den Knopf „Custom...“ drücken. Das ist in der deutschen Version so etwas wie „Selbstdefiniert...“ oder

„Eigene...“. Jetzt hast du die Möglichkeit einen eigenen Resourcentyp zu definieren. Ich habe den TEXT typ definiert.



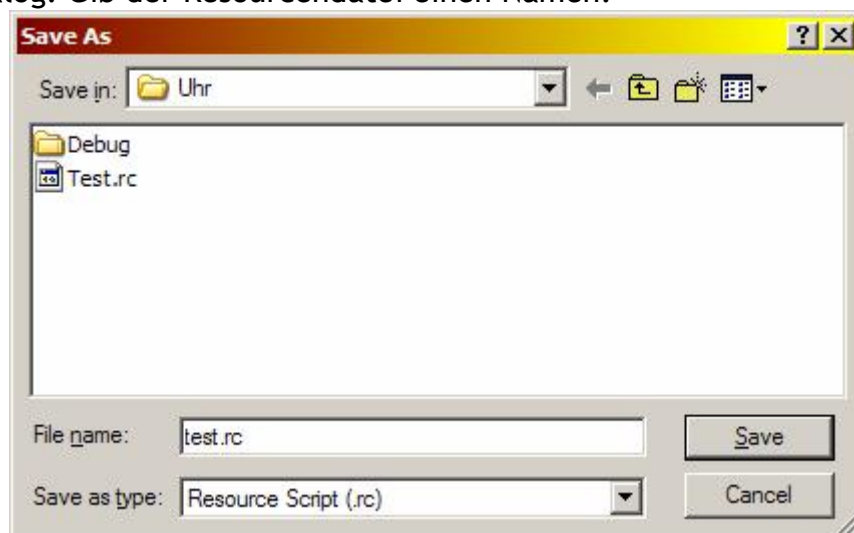
Nun erscheint ein Hex-Editor mit dem man die Resource editieren kann. Der Editor weiss nichts von Text, darum zu Beginn dieser komische Editor. Wir werden uns aber zu helfen wissen.



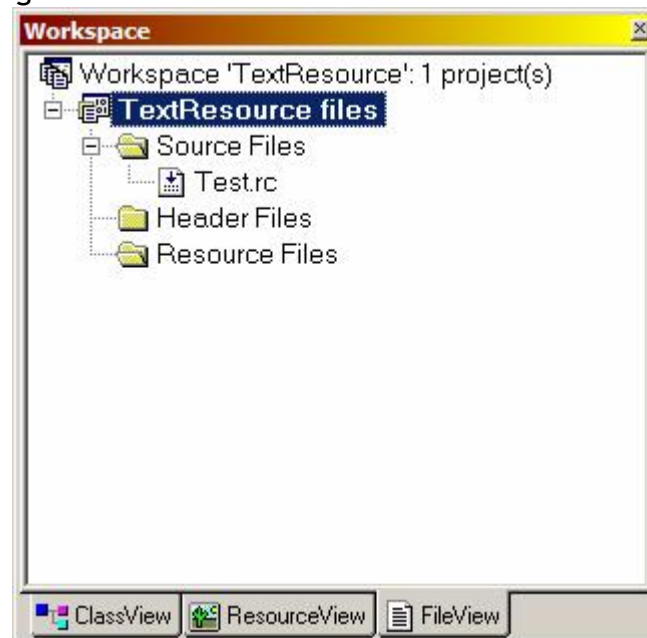
Durch Eingabe einiger Hexzahlen erscheint rechts Text. 41 entspricht 65 was der ASCII Code von A ist.

Speichern der Resource

Wenn du nun speichern wählst oder Ctrl + S drückst erscheint der Dateidialog. Gib der Ressurcendatei einen Namen.



Jetzt müssen wir noch dafür sorgen, dass die Ressourcen ganz in das Projekt kommen. Füge die .rc Datei in das Projekt ein, als würdest du eine .h oder .cpp Datei einfügen.

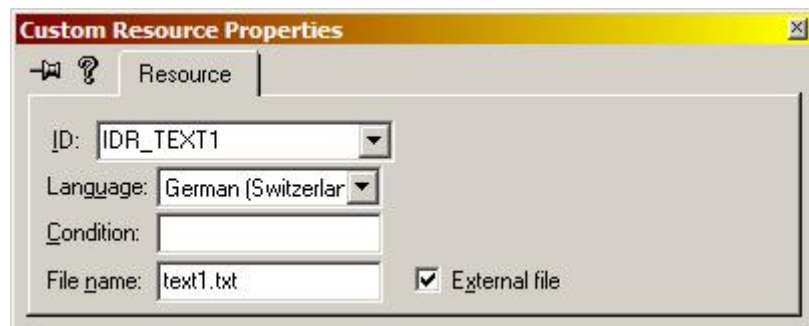


Du bemerkst nun sicher den neuen Tab für ResourceView!

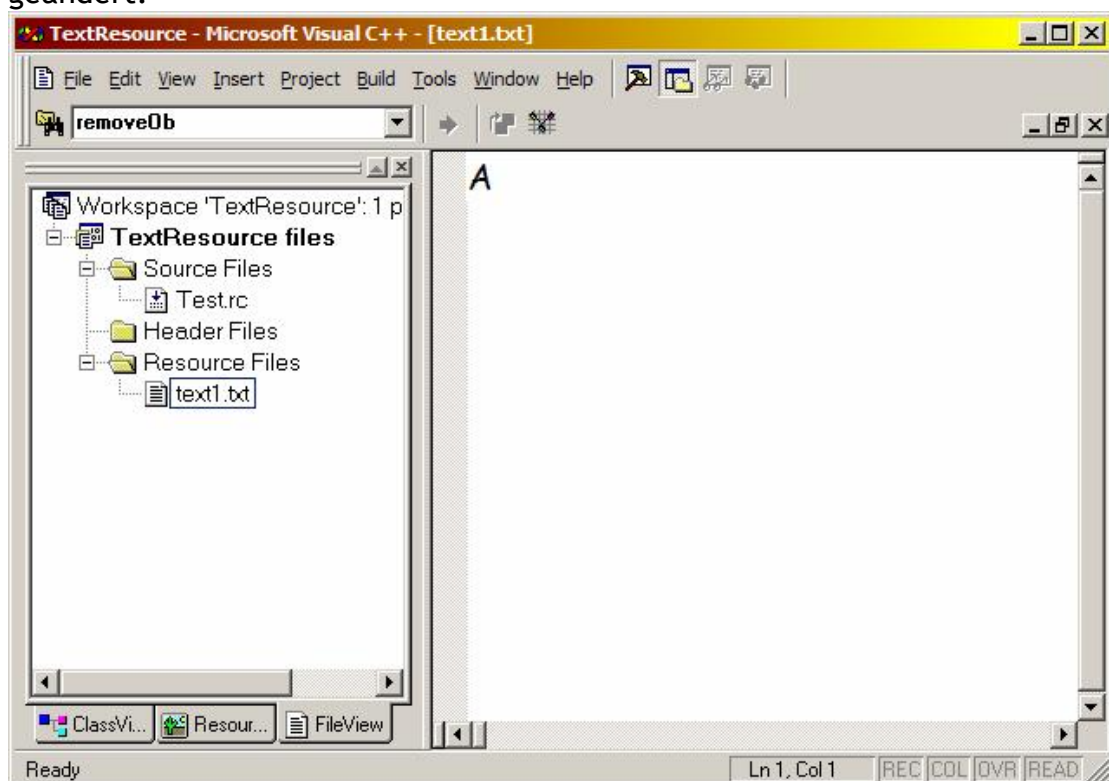


Hier ist die Resource, die du erzeugt hast. Jede Resource hat einen Typ-hier „TEXT“ und eine ID, hier IDR_TEXT1. Durch Rechtsklick auf diese Resource kannst du dir die Eigenschaften ansehen und ändern! Füge auch die Datei „resource.h“ in das Projekt ein. Diese enthält die Konstante IDR_TEXT1.

Ändern der Ressourceneigenschaften



Der Knopf External file gibt an, dass sich der Inhalt in einer besonderen Datei befinden soll. Ich habe einfach den Dateinamen auf text1.txt geändert!



Achte darauf, dass die Datei text1.txt sich im Verzeichnis der Resource files befindet.

Editieren des Textes

Editiere nun den Text.

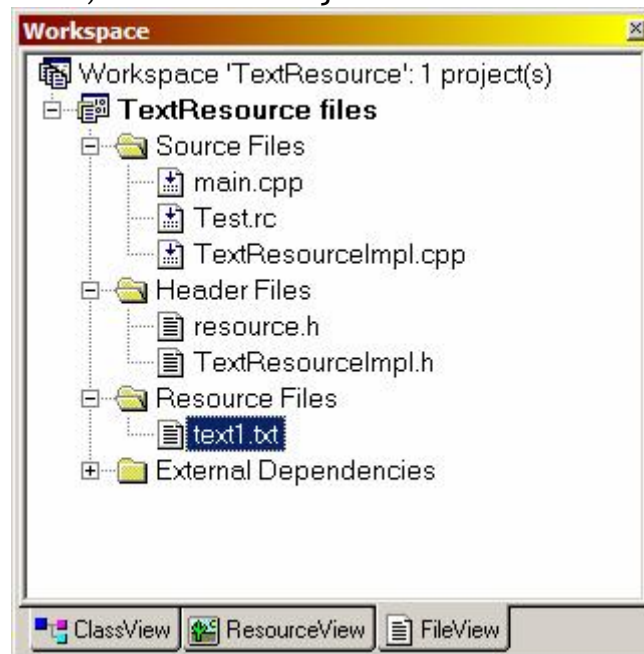
Die Klasse ResourceText

Behandlung von Ressourcen

Die Klasse ResourceText nimmt dir die Behandlung von Ressourcen ab und liefert einfach einen string, in dem sich der Text der Resource versteckt!

Einfügen aller nötigen Dateien

Füge alle Dateien ein, so dass das Projekt so aussieht:



Erzeuge nun ein main. Hier ein main mit einem kleinen Test:

```
#include "TextresourceImpl.h"
#include "resource.h"

#include <iostream>

using namespace std;

int main()
{
    TextResource r("TEXT", IDR_TEXT1);

    cout << r.str();

    return 0;
}
```

Die Klasse ResourceText hat nur einen Konstruktor und eine Methode str(). Der Konstruktor benötigt den Resourcentyp (hier „TEXT“) und die ID der Resource, die in der Datei Resource.h definiert ist. Die Methode str() liefert einfach den string!