

# CANerator CANopen

## Generierung und Konfiguration von Source Code

Der CANerator CANopen ist ein Werkzeug zur Verwaltung von Source Code-Konfigurationseinstellungen und zur Erstellung bzw. automatischen Generierung des Objektverzeichnisses.

### Eigenschaften und Vorteile

Das Kommunikationsprofil CANopen zeichnet sich durch seine breite Skalierbarkeit für die Verwendung in Embedded-Netzwerken aus. Es bietet vielfältige Möglichkeiten, die benötigte Funktionalität von Geräten abzubilden und gleichzeitig den geringen Ressourcen eines Mikrocontrollers Rechnung zu tragen. Der CANopen Master und Slave Source Code von Vector Informatik bietet daher eine Vielzahl von Einstellmöglichkeiten.

Der Preis für diese Flexibilität ist eine höhere Komplexität des Entwicklungsprozesses. Zusätzlich besteht die Forderung nach Wiederverwendbarkeit und flexibler Adaptierbarkeit der Software für ganze Paletten verschiedener Geräte des Herstellers.

Neben der eigentlichen Konfiguration des Source Codes hat der Entwickler von CANopen-Geräten zusätzlich ein Objektverzeichnis zu erstellen. Dieses bildet die Schnittstelle zwischen Applikation und CANopen-Kommunikation und dient zur Beschreibung der Geräteeigenschaften.

Mit CANerator CANopen steht dem Software-Entwickler ein Werkzeug zur Verfügung, das ihn bei den genannten Aufgaben komfortabel unterstützt, und den Entwicklungsprozess wesentlich beschleunigt.

### Funktionen

CANerator CANopen stellt folgende Funktionen zur Verfügung:

- > Konfiguration des Source Codes mit Hilfe einer übersichtlichen Darstellung aller #define-Anweisungen
- > Erstellung und automatische Generierung des Objektverzeichnisses als Source Code
- > Automatische Erstellung von EDS-Dateien
- > Einfache Verwaltung von Projekten

Diese Funktionen werden nachfolgend im Detail vorgestellt.

### Anwendungsgebiete

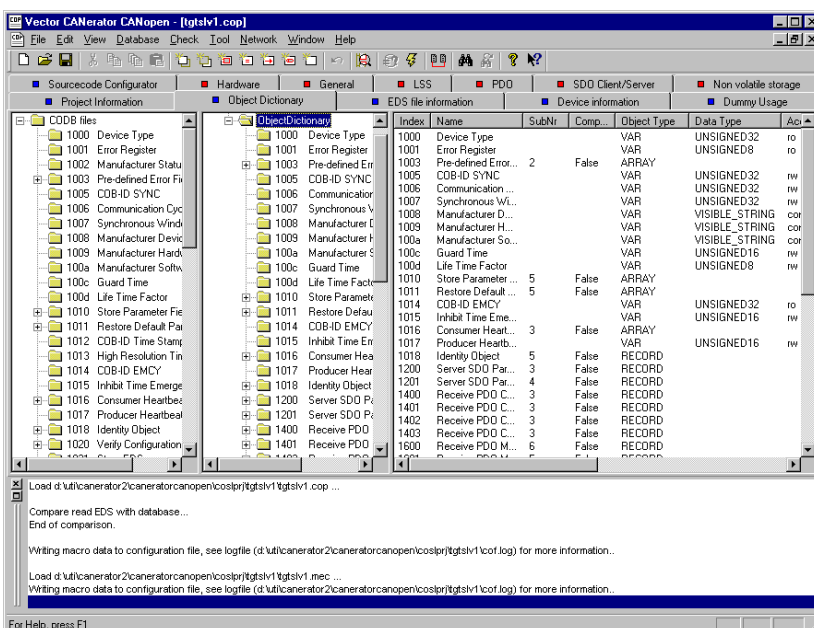
Der CANerator CANopen ist für den Einsatz zusammen mit dem CANopen Master und Slave Source Code von Vector vorgesehen.

### Source Code-Konfigurator

CANerator CANopen stellt alle relevanten Einstellungen des Source Codes (#defines) in einer Baumansicht dar.

Eine Instrumentierung des Source Codes erlaubt hierbei, nur die relevanten #define-Anweisungen darzustellen, und strukturiert sie in problembezogenen Gruppen.

Weiter sind den Einstellungen Kommentare zugeordnet, die Zweck und Wertebereich erläutern. Verschiedenen Datentypen sind hierbei unterschiedliche Masken mit typsensitiven Validatoren zugeordnet.



Erstellen der EDS-Datei mit Hilfe von CANeds

### Generierung Objektverzeichnis

Durch vollständige Integration von CANeds stehen alle Möglichkeiten dieses Werkzeugs zur Verfügung. Im Einzelnen sind diese in der Produktinformation CANeds beschrieben. Mit Hilfe von CANeds werden alle Standardobjekte des Objektverzeichnisses per Drag & Drop erzeugt. Weiterhin ist es möglich, herstellerspezifische Objekte zu definieren. Die Objekte sind bereits mit sinnvollen Vorgabewerten belegt, die bei Bedarf modifiziert werden können. Nachdem der Entwickler die Objekte spezifiziert hat, zeigt CANerator CANopen den Aufbau des Objektverzeichnisses in einer hierarchischen Baumstruktur an. Anschließend werden den Objekten Variablen im Programm zugeordnet. Diese stellen letztlich die Schnittstelle zwischen dem protokollspezifischen Teil der Software und der Applikation dar. Der CANerator unterstützt hierbei neben der symbolischen Angabe von Variablenamen auch die Angabe physikalischer Adressen, um zum Beispiel direkt auf Hardware-Ports zugreifen zu können. Nachdem das Objektverzeichnis erstellt ist, werden die entsprechenden Teile des Source Codes automatisch erzeugt.

### Electronic Data Sheet (EDS)

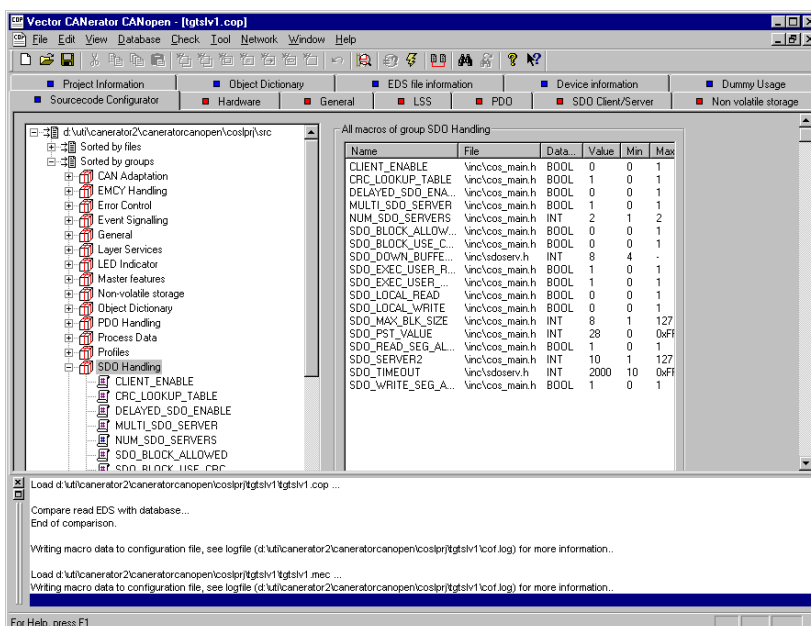
Quasi als Nebenprodukt fällt die Generierung eines Electronic Data Sheet (EDS) ab. Dieses kann mit den Funktionen von CANeds weiter bearbeitet und geprüft werden.

### Projektverwaltung

CANerator CANopen erlaubt die einfache und komfortable Verwaltung von Profilen. Ist ein Satz von Einstellungen erstellt worden, wird er unter eigenem Namen gespeichert. Verschiedene solcher Profile können auf ein und dasselbe Projekt zur Generierung angewendet werden. Hierdurch ist es zum Beispiel möglich, ein Projekt mit oder ohne Debugging-Informationen zu versehen, und mit einem Knopfdruck zwischen den Profilen zu wechseln.

Eine andere Anwendung ist die Erstellung von Profilen für verschiedene Gerätetypen auf Basis eines Source Codes. Dies erleichtert das Single-Source-Prinzip.

Als Projektdokumentation wird eine HTML-Seite generiert, die neben allgemeiner Projektinformation eine Liste aller Objekte mit ihren Attributwerten sowie eine Liste aller gesetzten Einstellungen enthält.



**CANerator CANopen im Konfigurationsmodus:  
alle #define-Einstellungen können  
übersichtlich bearbeitet werden**