

Features

CoDeSys V2.3

Letzte Aktualisierung: 05.08.02, V2.3.0

Änderungen, die sich im Rahmen von Service Packs ergeben, entnehmen Sie bitte dem jeweiligen Release Information Dokument, das auf der Installations-CD mit geliefert wird.

INHALTSVERZEICHNIS

1	ENI	2
2	INTELLISENSE	2
3	ERWEITERUNGEN IM KONTAKTPLAN-EDITOR	2
4	MAKROBIBLIOTHEKEN	2
5	SCHLÜSSELWÖRTER FÜR KOMMANDOPARAMETER (BATCH-BEFEHLE):	3
6	LIZENSIERUNG	3
7	ERWEITERUNGEN IN DER TASKKONFIGURATION	3
8	MIPS CODEGENERATOR	3
9	ERWEITERUNG BZGL. DER NETZWERKVARIABLEN	3
10	ERWEITERUNGEN IM OBJEKTVERZEICHNIS	3
11	CODESYS AUTOMATION INTERFACE	4
12	ZIELSYSTEM-VISUALISIERUNG	4
13	WEB-VISUALISIERUNG	4
14	CODESYS SOFTMOTION	4
15	TOOLS	4

1 ENI

Die **Schnittstelle 'Engineering Interface'** kann in CoDeSys optional aktiviert werden. Sie dient dazu, über einen eigenständigen **ENI Server** auf eine externe Datenbank zuzugreifen, in der CoDeSys Bausteine bzw. Übersetzungsdateien in verschiedenen Kategorien verwaltet werden. Dies eröffnet folgende Möglichkeiten:

- Multi-User-Betrieb bei der Erstellung von CoDeSys Projekten
- gemeinsamer Projektdaten-Pool für CoDeSys und andere Programme, die ebenfalls Clients des ENI Servers sind
- Versionsverwaltung für CoDeSys Projekte

In CoDeSys werden durch das ENI beispielsweise folgende Datenbankfunktionen zur Verfügung stehen: Abrufen, Auschecken, Einchecken, Auschecken rückgängig, Unterschiede anzeigen, Versionsgeschichte anzeigen, Version Labeln etc.

Standardmäßig stehen die (lizenzpflichtigen) Treiber für die Datenbank 'Visual SourceSafe 5.0', 'Visual SourceSafe 6.0', 'MKS Source Integrity' und (lizenzfrei) für ein lokales Dateisystem zur Verfügung. Für Anpassungen auf andere Datenbanksysteme gibt es das **ENI Driver Development Kit**.

Das eigenständige Programm **ENI Explorer** bietet die Möglichkeit, unabhängig von CoDeSys die Datenbank zu bedienen.

Die ENI-Funktionen in CoDeSys sind auch über entsprechende **Kommandozeilen-/datei-Befehle** steuerbar.

2 Intellisense

Die Intellisense-Funktion kann optional eingeschaltet werden. Sie dient der Erleichterung der Eingabe von Operanden in allen Editoren, im Watch- und Rezepturverwalter, in der Visualisierung und in der Tracekonfiguration.

Wird anstelle eines Bezeichners ein Punkt "." eingegeben, öffnet sich eine Auswahlliste aller lokalen und globalen Variablen. Wird als Bezeichner eine Funktionsblockinstanz oder eine als Struktur definierte Variable gefolgt von einem Punkt eingegeben, öffnet sich nach Eingabe des Punktes eine Auswahlliste der Ein- und Ausgangsvariablen des Funktionsblocks bzw. der Strukturkomponenten.

3 Erweiterungen im Kontaktplan-Editor

Achtung: bereits ab Service Pack 4 von V2.2 implementiert !

Neue Optionen für die Darstellung ('Extras' 'Optionen'):

- Jeder Kontakt kann mit einem Kommentar versehen werden, Angabe der Zeilenanzahl möglich
- Die Variablenname im Netzwerk können mehrzeilig sein, Angabe der Zeilenanzahl möglich
- Netzwerke können mit Umbrüchen versehen werden (ein-/ausschaltbar)

4 Makrobibliotheken

Die in CoDeSys erzeugten Makros können extern in Makrobibliotheken abgelegt werden, die dann bei Bedarf in Projekten eingebunden werden können, um die Makros zur Verfügung zu haben.

5 Schlüsselwörter für Kommandoparameter (Batch-Befehle):

Bei der Angabe der Kommandoparameter können folgende Schlüsselwörter verwendet werden:

\$PROJECT_NAME\$: Name des aktuellen CoDeSys-Projekts

\$PROJECT_PATH\$: Pfad des Verzeichnisses, in dem die aktuelle CoDeSys-Projektdatei liegt

\$PROJECT_DRIVE\$: Laufwerk auf dem das aktuelle CoDeSys-Projekt liegt

\$COMPILE_DIR\$: Übersetzungsverzeichnis des aktuellen CoDeSys-Projekts

\$EXE_DIR\$: Verzeichnis, in dem die codesys.exe-Datei liegt

6 Lizenzierung

Jeder Anwender oder OEM kann im Programmiersystem selbst lizensierbare Bibliotheken erstellen. Diese werden durch einen Software-Mechanismus geschützt. Der gleiche Schutz ist nun einheitlich für alle kostenpflichtigen Software-Module (ENI, CoDeSys HMI, CoDeSys SP RTE) von 3S vorgesehen.

Ein Lizenzverwaltungs-Programm gibt dem Anwender Übersicht und Informationen zu den auf dem lokalen Rechner installierten lizenzpflichtigen 3S-Produkten. Es ermöglicht auch das Anfordern und Registrieren weiterer Lizenzen.

7 Erweiterungen in der Taskkonfiguration

- Zusätzliche Standard-Eigenschaften für Tasks (Überwachungszeit, Überwachung ein-/ausschaltbar, Zeitauflösung, optionale Verbindung mit einem Interrupt)
- Herstellerspezifische Task-Eigenschaften
- Online-Übersicht über Zeitverhalten der Tasks (minimale/maximale/durchschnittliche Zykluszeit, jitter)
- Verknüpfungstabelle von Events (Standard und herstellerspezifische) mit POUs; Beispiele für Standard-Events: SPS-Start, SPS-Stop, letzter Zyklus vor Online Change, SPS-BP....

8 MIPS Codegenerator

Für MIPS Prozessoren erhält das Programmiersystem zusätzlich den MIPS-Codegenerator.

9 Erweiterung bzgl. der Netzwerkvariablen

Zusätzliche Konfigurationsmöglichkeiten für Netzwerkvariablen:

- Bootup-Verhalten (z.B. beim Bootup erhält Knoten automatisch Update der Variablenwerte)
- Optional bestätigte Netzwerkkommunikation
- Optional Check-Summe in den Telegrammen

10 Erweiterungen im Objektverzeichnis

Zielsystemabhängig (!), d.h. wenn in der Target-Datei bzw. in den Targeteinstellungen so definiert: stand bereits bisher in den 'Ressourcen' im CoDeSys Programmiersystem das Feature 'Objektverzeichnis' zur Verfügung, das dem Austausch von Variablen zwischen zwei Steuerungen im Netzwerk dient. Der OD-Editor wird nun erweitert:

- Weitere Standardattribute für Objekteinträge: Minimal- /Maximalwert, Name, Einheit, Baustein der bei schreibendem/lesendem Zugriff aufgerufen wird
- Herstellerspezifische Attribute für Objekteinträge
- Tabelleneditor zur erleichterten Zuordnung von Index-Offsets zu Komponenten von Funktionsblockinstanzen

11 CoDeSys Automation Interface

Skriptfähige COM-Schnittstelle zur Erzeugung und Bearbeitung von Projekten

12 Zielsystem-Visualisierung

Zielsystemabhängig (!), d.h. wenn in der Target-Datei so definiert, kann CoDeSys für gewünschte Visualisierungsobjekte eines Projekts ST-Code (Strukturierter Text) erzeugen, der mit den anderen Programmbausteinen auf die Steuerung geladen wird. Wenn das Laufzeitsystem die Funktionalität unterstützt und entsprechende Monitorausstattung vorhanden ist, kann die Visualisierung direkt auf der Steuerung gestartet werden. Das Programmiersystem CoDeSys muß nicht mehr laufen, um die Visualisierung nutzen zu können, was erheblich verringerten Hauptspeicherbedarf bedeutet.

13 Web-Visualisierung

Zielsystemabhängig (!), d.h. wenn in der Target-Datei so definiert, kann CoDeSys die dafür ausgewählten Visualisierungsobjekte eines Projektes so verarbeiten, daß sie im XML-Format beim Download auf eine Steuerung geladen werden können. Dort verarbeitet ein Web-Server die aktuellen Steuerungsdaten ebenfalls im XML-Format und stellt die laufend aktualisierte Visualisierung zur Verfügung. Somit kann diese auf jedem anderen über Internet angebundenen Rechner plattformunabhängig in einem Web-Browser aufgerufen und dargestellt werden.

14 CoDeSys SoftMotion

Zielsystemabhängig (!), d.h. wenn in der Target-Datei so definiert, steht mit dem lizenzpflichtigen Feature 'SoftMotion' in den 'Ressourcen' im CoDeSys Programmiersystem ein 'Basis-Baukasten' zur Verfügung, mit dem Motion-Funktionalität in CoDeSys und CoDeSys SP integriert werden kann. Es dient der Programmierung anspruchsvoller Bewegungsaufgaben und bietet dazu graphische Editoren für die Sollwertvorgaben (NC-Editor (DIN66025), Kurvenscheibeneditor), eine Anzahl von Bausteinbibliotheken und ein standardisiertes (konfigurierbares) Antriebs-Interface für gängige Antriebssysteme (CAN, SERCOS).

15 Tools

Zielsystemabhängig (!), d.h. wenn in der Target-Datei so definiert, steht die Funktion 'Tools' im Register Ressourcen zur Verfügung. Es dient der Anknüpfung externer Tools, die dann aus CoDeSys heraus aufgerufen werden können. (Beispielsweise Öffnen eines externen Texteditors, eines pdf-Dokuments etc.)

Die Tools-Auswahl kann in der Target-Datei vordefiniert werden und ggfs. im CoDeSys-Projekt erweitert werden.