



Datenblatt CODESYS Professional Module Developer

Der CODESYS Application Composer ermöglicht für modulare Maschinen ein effizientes, No-Code-Erstellen und vollständiges Generieren von passenden Steuerungsapplikationen aller Größen durch Kombinieren und Parametrieren multi-modaler Softwaremodule.

Mit der einjährigen **Module Developer** Arbeitsplatzlizenz kann man Module zum Komponieren im CODESYS Application Composer entwickeln.

Weitere Informationen

Vertrieb:	sales@codesys.com
Einsatz und Vorteile: (CODESYS Website)	https://de.codesys.com/produkte/engineering/application-composer
Dokumentation: (CODESYS Onlinehilfe)	https://content.helpme-codesys.com/de/CODESYS%20Application%20Composer
Beispielprojekte:	https://content.helpme-codesys.com/de/CODESYS%20Examples/_ex_application_composer.html
Empfohlene Schulungen: (CODESYS Academy)	Objektorientiertes Programmieren, Application Composer: https://de.codesys.com/ecosystem/services/academy-training/training/

Technische Daten

Zum Betrieb der mit ‚#‘ markierten CODESYS Features sind entsprechende Lizenzen auf der Steuerung erforderlich.

Funktionsbereiche

Definition von Modulen	<ul style="list-style-type: none">• Befehl „Modul aus FB erzeugen“• Befehl „Makromodul erzeugen“ aus Modul-Unterbaum• Befehl „Erweiterungsmodul erzeugen“ für Submodul-Steckplatz• Moduldeklarationseditor• Ableiten von Basis-Moduldeklarationen per „IMPORTS“
Komposition von Modulinstanzen	<ul style="list-style-type: none">• Definition von Submodul-Slots mit Default-Submodulen• Einfügen in Modulbaum mit Auswahl passender Submodule• Sequenzeditor für Ausführungsreihenfolge der Submodule

Konfiguration der Modulinstanzen	<ul style="list-style-type: none"> • Setzen definierter Modulparameter (z.B. Größe eines variables Arrays) • Verknüpfen definierter Modul-IOs mit Gerätekanälen per Verbindungseditor • Auswahl einer Visualisierung für die Modulinstanz
Projektgenerierung	„Erzeugen“ Befehl und kombinierter Befehl „Erzeugen, Übersetzen und Einloggen“
Persistenzmanagement	IEC Variablen in Persistenz-Datengruppen einordnen und Persistenzkanäle sowie Speicherdateien konfigurieren
Remote Module	Austausch von Modul-IOs mit Modulinstanzen in anderen Applikation
IEC Zugriff auf Modulbaum	GetToplevelInstance, GetNextModule, GetFirstChild, NextSibling, GetParent, AncestorRange, ...

Inhalte des generierten Maschinenprojekts

Generierte IEC Programmierung

• IEC Objekte	POU, GVL, DUT, Task
• Modulinstanzbaustein maßgeschneidert für Modulkonfiguration	<ul style="list-style-type: none"> • Modul-Parameter: im Moduleditor gesetzte FB-Inputs • Modul-Steckplätze: im Modulbaum mit (Default) Submodulen belegte FB-Inputs, mit vordefinierter Parametrierung und Verknüpfung des Submodul-IOs • Modul-Referenzen: im Modulbaum gesetzte FB-Inputs zu anderen Modulinstanzen§ • Modul-IO: FB-Inputs und -Outputs, die auf Gerätekanäle mappbar sind (Kanäle generierter Geräte des Moduls oder per Verbindungseditor§) • VarArrays: Array mit Größe bestimmt durch Modul-Parameter oder durch Größe eines Multi-Slots oder einer Multi-Referenz • Proxies: lokale Proxies für Modulinstanzen in anderen Applikationen <p>(§) Kann sich auch in anderer Applikation des Projekts befinden.</p>
• Ausführung	<ul style="list-style-type: none"> • Automatische Applikations- und Task-Zuordnung per Moduldeklaration • Ausführungsreihenfolge bestimmt durch Baumstruktur oder Ablaufeditor • Proxy-Datenaustausch per Kommunikationstask, IRMPService, Netzwerkvariablen
Generierte I/O-Ebene	
• Geräte pro Modul	Generiert wie in Moduldeklaration festgelegt bzw. in Moduleditor konfiguriert

• mögliche Gerätetypen	alle (SoftMotion-Achsen#, Feldgeräte, Buskoppler, Gerätebäume, ...)
• mögliche Feldbusse#	alle (PROFINET, EtherCAT, Ethernet/IP, ...)
• Konfiguration des Geräts	Generiert wie in Moduldeklaration festgelegt bzw. in Moduleditor konfiguriert
• Mapping des Geräts	<ul style="list-style-type: none"> • Kanäle auf Modul-IOs gemappt gemäß Moduldeklaration oder Verbindungseditor • Geräte-FB auf Modul-Referenz gemappt gemäß Moduldeklaration

Generierte Visu-Funktionalität

• Visualisierung# pro Modul	ausgewähltes Visuobjekt für Modul-FB mit eingebetteten Submodul-Visus, Textlisten
• Alarmer pro Modul	Alarmkonfigurationsobjekt, Alarmspeicherungsobjekt, Alarmklassenobjekt, Alarmvisualisierung
• Trends pro Modul	TrendConfigObjekt, Trendvisualisierung

Vermischte Daten

Scripting	JA (alle oben genannten Funktionsbereiche)
Automation Platform SDK	<ul style="list-style-type: none"> • Neue Sektionen in Moduldefinition definieren und Generierung daraus • Generierung von zusätzlichen Projekthinhalten
Abhängigkeiten	<ul style="list-style-type: none"> • Installer 3.5.19.30 • Visualization 4.3.0.0 • Visualization Support 4.0.0.0 • Communication 4.0.0.0 • CFC 4.0.0.0 • Base Libraries 4.0.1.0 or 5.0.0.0 • Scripting 4.0.0.0 (optional) • License Provider 1.3.0.0 (optional) • Licensing Support 1.2.0.0 (optional)
Mitgelieferte Modul-Bibliotheken	AC_ModuleBase, AC_DeviceDiagnosis, AC_Alarming, AC_Trend, AC_DataLog, AC_Persistence

Allgemeine Informationen

Lieferant:

CODESYS GmbH
 Memminger Straße 151
 87439 Kempten
 Deutschland

Support:

Technischer Support ist bei diesem Produkt nicht enthalten. Um technischen Support zu erhalten, erwerben Sie bitte ein CODESYS Support Ticket.

<https://support.codesys.com>

Artikelname:

CODESYS Professional Module Developer

Artikelnummer:

2111000035

Vertrieb/Bezugsquelle:

CODESYS Store
<https://store.codesys.com>

Lieferumfang:

- Package „CODESYS Application Composer“
- Lizenzschlüssel

Systemvoraussetzungen und Einschränkungen

Programmiersystem	CODESYS Development System 3.5.19.30 oder höher
Laufzeitsystem	CODESYS Control Version 3.5.0.0
Unterstützte Plattformen/ Geräte	Steuerungen, auf denen das CODESYS Laufzeitsystem installiert ist.
Zusätzliche Anforderungen	-
Einschränkungen	<ul style="list-style-type: none"> • Ohne gültige Lizenz ist der CODESYS Application Composers funktional eingeschränkt auf den Persistence Manager und auf die Schritte nach dem Generieren der Applikation (bauen, herunterladen, debuggen). • Für die Lizenzen <i>Composer S</i> bis <i>XL</i> benötigen Sie CODESYS Application Composer Version 4.4.0.0 oder neuer.

- Für die Lizenz *Composer Module Developer* benötigen Sie CODESYS Application Composer Version 4.5.0.0 oder neuer.

Lizenzierung

Für die volle Funktionalität (Definieren und Komponieren eigener Module zur Generierung von Applikationen) ist eine gültige Lizenz erforderlich.

Die Lizenzierung erfolgt über den CODESYS License Provider Mechanismus und erfordert die folgenden Packages:

- CODESYS License Provider \geq 1.1.0.0
- CODESYS License Provider Enabler \geq 1.1.0.0
- CODESYS Licensing Support \geq 1.1.0.0

Diese werden beim Installieren über den CODESYS Installer automatisch per Referenz aufgelöst und mitinstalliert.

Die Aktivierung der Lizenzen erfolgt auf einem UFC-Lizenz-Container (Soft-Container oder USB-Dongle) und ist als Arbeitsplatzlizenz verfügbar. Durch Umstecken des CODESYS Keys kann die Lizenz an einem anderen Arbeitsplatz genutzt werden.



WORKSTATION

Erforderliches Zubehör

Optional: CODESYS Key

Bitte beachten Sie: Technische Änderungen, Druckfehler und Irrtümer vorbehalten. Es gilt der Inhalt der aktuellen Online-Version dieses Dokuments.

Erstellungsdatum: 21.05.2026