



**Das „Redvers COBOL JSON Interface“ ermöglicht COBOL Anwendungen schnellen und effizienten Zugriff auf die Welt von JSON-Objekte. Es besteht aus dem JSON-Generator und dem JSON-Parser. Diese Standardsoftware wurde in reinem COBOL geschrieben. Alles, was Sie noch machen müssen, ist die Kodierung einer COBOL-Satzbeschreibung und eines CALL-Befehls, um JSON-Objekte zu generieren oder zu parsen.**

#### **Merkmale:**

- Läuft auf jedem Rechner, der COBOL ausführen kann
- Wird als COBOL Quellprogramm lizenziert (verschlüsselt)
- Generiert und parst mehrdimensionale Felder (verschachteltes occurs)
- Einfache Bedienung
- Schnell, effizient, professionell und skalierbar
- Benötigt keine Einarbeitungszeit für COBOL Programmierer
- Kostenloses Tool, um aus einem Schema ein Copybook einfach erstellen zu können
- Läuft im Batch- oder Online-Modus
- **Kostenlose 30-Tage-Demoversion**

Das „**Redvers COBOL JSON Interface**“ arbeitet auf Datensatzebene. Das hat zur Folge, daß alle JSON-Objektstrukturen vom Interface bei der Generierung automatisch erzeugt werden bzw. beim Parsen interpretiert werden. Durch diesen Ansatz vermeiden wir eine umständliche Anwendungslogik auf Objektebene und haben stattdessen nur einen einzigen CALL-Befehl, der den nächsten logischen Satz dem Interface übergibt bzw. von diesem erhält.

Unsere verschiedenen Interfaces werden derzeit von Kunden auf der ganzen Welt genutzt, auf den Plattformen **iSeries/AS400, UNIX, HP, Linux, Fujitsu BS2000, Micro Focus** und auf **IBM Mainframes**.

## Funktionsweise des Programms

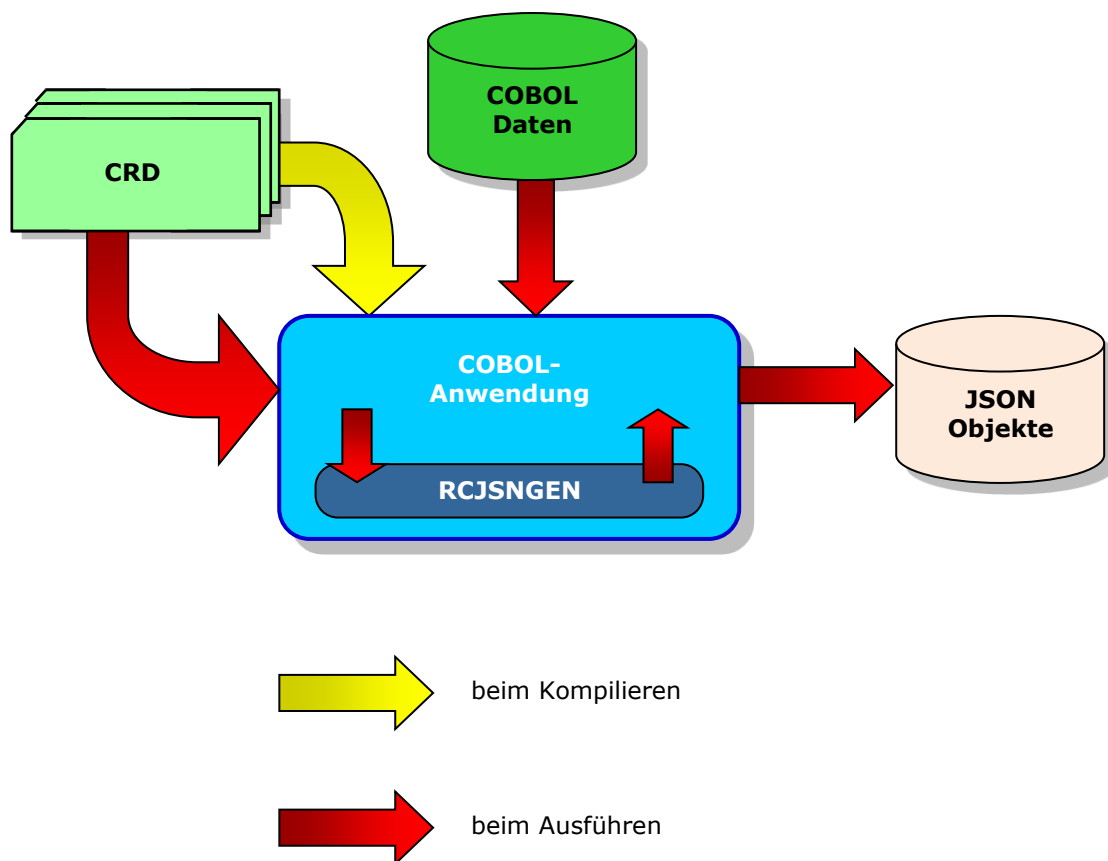
Bevor eine COBOL-Anwendung JSON generieren oder parsen kann, muß eine **COBOL Satzbeschreibung (CRD)** kodiert werden, in der es eine Felddefinition in der Working Storage für jedes JSON-Objekt gibt. Die CRD wird in die normale Copybibliothek eingefügt und ein **COPY/INCLUDE** Befehl wird der Anwendung hinzugefügt.

Der CRD-Quellcode wird auch in das rufende Programm eingefügt, damit die Daten im CALL an die Interface-Unterprogramme übergeben werden können. Das kann entweder dadurch geschehen, daß der Quelltext in die Working Storage fest kodiert wird, oder indem der Quelltext zum Zeitpunkt der Ausführung aus der Copybibliothek gelesen wird.

Um JSON zu **generieren**, wird die CRD von der COBOL-Anwendung befüllt, die sie anschließend dem Generierungs-Unterprogramm (**RCJSNGEN**) weiterreicht, zusammen mit dem CRD-Quelltext in einem CALL-Aufruf. **RCJSNGEN** konvertiert dann die COBOL-Daten zu JSON-Objekten, und gibt das oberste JSON-Objekt an das Anwendungsprogramm zurück.

Um JSON zu **parsen**, übergibt das Anwendungsprogramm das JSON-Objekt und den CRD-Quellcode an das Parser-Unterprogramm (**RCJSNPAR**), das die COBOL-Daten in den CRD-Feldern zurückliefert.

Der Generierungs Prozeß wird hier dargestellt:



*Parsing geschieht auf ähnliche Weise wie oben dargestellt, mit dem Unterschied, daß das Anwendungsprogramm das JSON-Objekt an **RCJSNPAR** übergibt, welches dann die COBOL-Daten in der CRD zurückliefert.*

## Technische Informationen

Die „**Redvers COBOL JSON Interface**“ 2.1 Unterprogramme werden als COBOL Quellcode verkauft. Sie werden installiert, indem sie in Ihre Quellcodebibliothek eingefügt und mit Ihrem normalen COBOL-Compiler kompiliert werden. Indem die Kompilierung beim Kunden passiert, kann die Installation auf unterschiedlichsten Plattformen erfolgen, und das stellt sicher, daß die Redvers-Software vollständig mit allen anderen COBOL-Anwendungsprogrammen, die dort vorhanden sind, kompatibel ist.

Die Verwendung von Quellcode erlaubt es dem Kunden auch, den Inhalt gewisser Werte zu verändern, die ansonsten fest kodiert wäre. Diese Felder werden „Benutzerdefinierte Variable“ oder im Original „**User Maintained Variables**“ (UMV's) genannt, und sie können verändert werden, um die Effizienz der Programmausführung zu steigern und/oder die Formatierung des ausgegebenen JSON zu verbessern. Details zu allen UMV's finden Sie in den Handbüchern zu unserer Software.

Um Kunden und Autoren vor unberechtigten Quellcodeänderungen zu schützen, wird der Quellcode der Unterprogramme mit dem „**Redvers Cloaking Device**“ verschlüsselt. Dadurch wird der Quellcode logisch nicht verändert, doch er wird für Menschen unverständlich.

Die **COBOL-Satzbeschreibung** (COBOL Record Definition oder CRD) kann so kodiert werden, daß jedes JSON-Objekt generiert oder geparst werden kann. Sehr große JSON-Objekte werden generiert oder geparst, indem das Interface mehrfach aufgerufen wird, so daß der Bedarf an Hauptspeicher minimiert wird.

Die generierten Objekte entsprechen dem [JSON Data Interchange Standard](https://www.json.org), wie er auf der Seite <https://www.json.org> dargestellt ist.

Die geparsten Objekte werden auf Syntaxfehler untersucht. Falls ein Fehler gefunden wurde, erfolgt ein Hinweis, der die Stelle des fehlerhaften Zeichens im Eingabestring bezeichnet. Es wird keine Schemavalidierung durchgeführt.

Die Generierung von JSON geschieht mit einer Geschwindigkeit von **9,6 Megabytes pro Sekunde** und das Parsen mit **8,8 Megabytes pro Sekunde**, getestet auf einem IBM zSeries Mainframe unter z/OS 1.10. Die maximale Objektgröße bei Onlineverarbeitung beträgt **99 MByte**; es gibt keine maximale Objektgröße bei Batchanwendungen.

## Das Produktangebot

Eine Dauerlizenz für das „**Redvers COBOL JSON Interface**“ kann für eine einmalige Gebühr erworben werden. Alternativ kann die Software auch gemietet werden für eine jährliche Gebühr, die 20% des Betrags der Dauerlizenz beträgt.

### Für diesen Betrag erhalten Sie:

- den Quellcode (verschlüsselt)
- Beispiel für rufende Programme
- Handbücher
- eine unternehmensweit gültige Softwarelizenz
- Geld-zurück-Garantie
- Softwareupdates und Support per E-Mail\*

\* Kostenlos für das erste Jahr mit einer geringen jährlichen Folgegebühr.

### Zusätzliche Optionen:

- telefonischer Support rund um die Uhr
- Software Escrow / Quellcodehinterlegung bei Software Escrow Solutions

Die aufgeführte Software und Handbücher werden als Textdateien und PDF E-Mail-Anhänge geliefert, wenn keine abweichenden Vereinbarungen getroffen wurden. Sie werden installiert, indem Sie den Quelltext manuell in Ihre COBOL Quelltextbibliothek kopieren, und dann mit Ihrem üblichen Compiler kompilieren und linken.

Ausführliche Informationen zu den Preisen finden Sie auf: [https://www.cobol.de/cobol\\_json\\_pricing.php](https://www.cobol.de/cobol_json_pricing.php)

# Über Redvers Consulting

**Redvers Consulting bietet seit 1988 erstklassige Produkte und Dienstleistungen für COBOL-Anwendungen. Unser Ansatz, die Software als Quellcode auszuliefern, ermöglicht unseren Kunden die Erfüllung ihrer geschäftlichen Anforderungen mit einer zuverlässigen, effizienten und perfekt integrierten Lösung.**

Unsere Kunden sind überwiegend große Finanzdienstleister in Großbritannien und den USA. In zunehmendem Maße sind wir im deutschsprachigen Raum und auch in anderen Branchen tätig.

Da unsere Software als verschlüsselter Quellcode ausgeliefert wird, bieten wir Unterstützung für alle Hardwareplattformen und Betriebssysteme, für die ein COBOL-Compiler existiert - EBCDIC, ASCII, big endian und little endian.

## Einige unserer Kunden:

Agora (FR)  
ANZ (AUS)  
BAE Systems (USA)  
Canada Life Assurance (UK)  
Deutsche Bank (USA)  
Deutsche Rentenversicherung Bund (DE)  
FirstBank (USA)  
Fiserv (USA)  
GMAC Insurance (USA)  
Hanesbrands (USA)  
John Deere (USA)  
Landesbank Hessen Thüringen (DE)  
LBS / Finanz Informatik (DE)  
J P Morgan (USA)  
Oppenheimer (USA)  
Pacific Gas (USA)  
Network Rail (UK)  
R+V Allgemeine Versicherung (DE)  
Sasktel (CAN)  
SEB (DE)  
Standard Life Assurance (UK)  
Suncorp (AUS)  
SunGard / FIS (USA)  
WorkSafeBC (CAN)  
Zurich Insurance (UK & CHE)

**Kontakt:** <https://www.cobol.de/contact.php>

### Deutsches Büro:

Redvers Consulting Ltd  
Scharfeneckweg 2,  
50739 Köln,  
Deutschland

**Tel:** +49 (0)221 1704 9000

### Hauptbüro:

Redvers Consulting Ltd  
1st Floor, 48 Dangan Rd,  
London E11 2RF,  
UK

**Tel:** +44 (0)870 922 0633

### Entwicklungsbüro:

Redvers Consulting Ltd  
16-18 Woodford Road,  
London E7 0HA,  
UK

**Tel:** +44 (0)203 138 5788