

# How-To-Manual

SQL4automation-Bibliothek

Dieses Dokument wurde von der Jetter AG mit der gebotenen Sorgfalt und basierend auf dem ihr bekannten Stand der Technik erstellt. Änderungen und technische Weiterentwicklungen an unseren Produkten werden nicht automatisch in einem überarbeiteten Dokument zur Verfügung gestellt. Die Jetter AG übernimmt keine Haftung und Verantwortung für inhaltliche oder formale Fehler, fehlende Aktualisierungen sowie daraus eventuell entstehende Schäden oder Nachteile.



Jetter AG  
Gräterstraße 2  
71642 Ludwigsburg  
Germany

**Telefon:**

|                    |                   |
|--------------------|-------------------|
| Zentrale           | +49 7141 2550-0   |
| Vertrieb           | +49 7141 2550-531 |
| Technische Hotline | +49 7141 2550-444 |

**E-Mail:**

|                    |                   |
|--------------------|-------------------|
| Technische Hotline | hotline@jetter.de |
| Vertrieb           | sales@jetter.de   |

Originaldokument

Dokumentenversion 1.00

Ausgabedatum 19.03.2021

# Inhaltsverzeichnis

|   |           |
|---|-----------|
| <b>1 Einleitung</b> .....                                   | <b>4</b>  |
| 1.1 Darstellungskonventionen .....                          | 4         |
| 1.2 Produktbeschreibung .....                               | 4         |
| <b>2 Einbindung der Bibliothek in ein STX-Projekt</b> ..... | <b>5</b>  |
| 2.1 Bibliothek zu JetSym hinzufügen .....                   | 5         |
| 2.2 Bibliothek in STX-Projekt einbinden .....               | 5         |
| 2.3 Bibliothek in STX aufrufen .....                        | 6         |
| <b>3 Arbeiten mit dem Beispielprojekt</b> .....             | <b>8</b>  |
| 3.1 Aufbau des Beispielprojekts .....                       | 8         |
| 3.2 Kommunikationsprinzip .....                             | 9         |
| 3.2.1 Anfrage stellen (Request).....                        | 9         |
| 3.2.2 Parsen der Antwort (Response).....                    | 9         |
| <b>4 Fehlercodes (Return Values)</b> .....                  | <b>11</b> |

# 1 Einleitung

## 1.1 Darstellungskonventionen

Unterschiedliche Formatierungen erleichtern es, Informationen zu finden und einzuordnen. Im Folgenden das Beispiel einer Schritt-für-Schritt-Anweisung:

- ✓ Dieses Zeichen weist auf eine Voraussetzung hin, die vor dem Ausführen der nachfolgenden Handlung erfüllt sein muss.
- ▶ Dieses Zeichen oder eine Nummerierung zu Beginn eines Absatzes markiert eine Handlungsanweisung, die vom Benutzer ausgeführt werden muss. Arbeiten Sie Handlungsanweisungen der Reihe nach ab.
- ⇒ Der Pfeil nach Handlungsanweisungen zeigt Reaktionen oder Ergebnisse dieser Handlungen auf.

### **i** INFO

#### **Weiterführende Informationen und praktische Tipps**

In der Info-Box finden Sie weiterführende Informationen und praktische Tipps zu Ihrem Produkt.

## 1.2 Produktbeschreibung

Dieses Dokument beschreibt die Einbindung und Verwendung der SQL4automation-Bibliothek in ein JetSym-STX-Projekt sowie die Nutzung und Anpassung des beigefügten Beispielprojekts für eigene Applikationen.

Weitere Informationen zu SQL4automation finden Sie auf deren Homepage:

<https://www.SQL4automation.com>

## 2 Einbindung der Bibliothek in ein STX-Projekt

Die SQL4automation-Bibliothek muss zunächst über den Bibliotheksmanager zu JetSym hinzugefügt werden und anschließend in ein Projekt eingebunden werden. Die Vorgehensweise wird im Folgenden beschrieben.

### 2.1 Bibliothek zu JetSym hinzufügen

1. Kopieren Sie die .libpackage-Datei an einen beliebigen Ort auf Ihrem PC.
  2. Öffnen Sie JetSym und erstellen Sie ein STX-Projekt.
  3. Öffnen Sie den Bibliotheksmanager indem Sie auf „Extras“ und dann auf „Bibliotheksmanager“ klicken.
  4. Klicken Sie auf „Hinzufügen“ und wählen Sie die Datei SQL4automation\_Library\_1.0.00.00\_Beta\_00.libdesc auf Ihrem PC aus.
  5. Klicken Sie auf „Öffnen“.
- ⇒ Die Bibliothek wird zu Ihrem Projekt hinzugefügt.

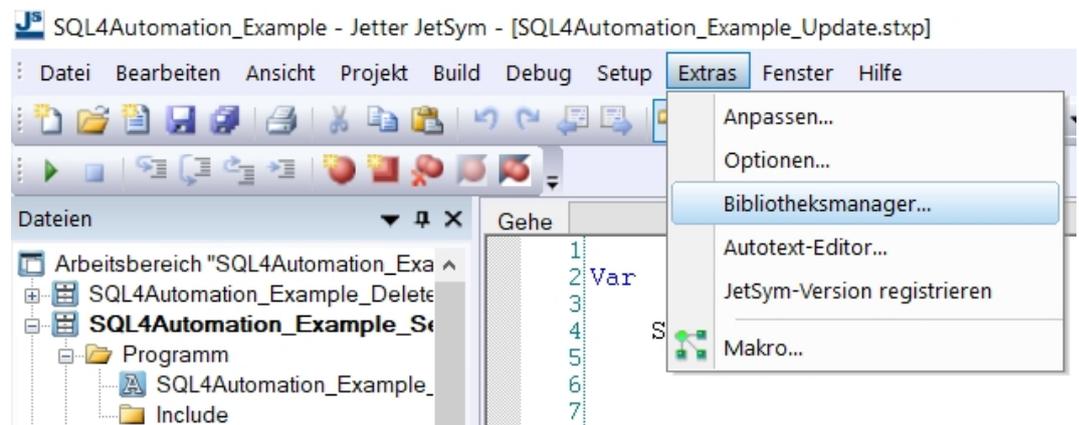


Abb. 1: Bibliotheksmanager

### 2.2 Bibliothek in STX-Projekt einbinden

1. Öffnen oder erstellen Sie ein STX-Projekt (z. B. das [Beispielprojekt](#) [► 8] „SQL4automation\_Example.wsw“)
2. Wählen Sie die Registerkarte „Dateien“ aus.
3. Rechtsklicken Sie auf den Eintrag „Bibliothek“ und wählen Sie „Bibliotheken hinzufügen...“ aus.  
⇒ Der Bibliotheksmanager öffnet sich.
4. Wählen Sie die neueste Version der installierten Bibliothek aus.  
⇒ Die Bibliothek wird eingebunden.

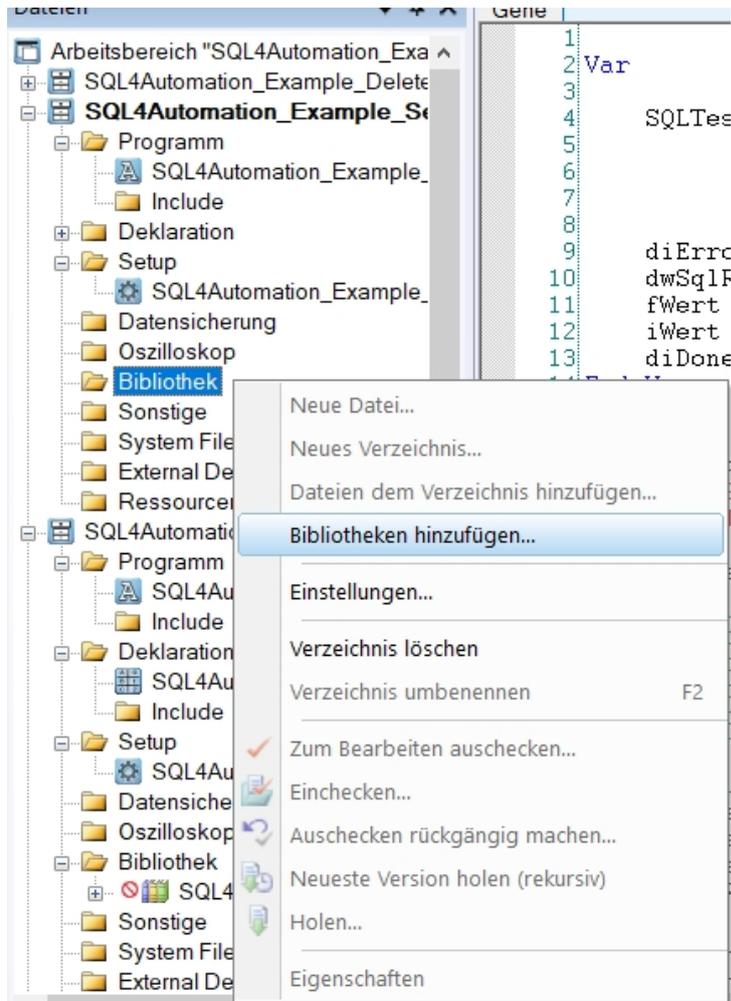


Abb. 2: Bibliothek hinzufügen

### 2.3 Bibliothek in STX aufrufen

Der Aufruf der Bibliothek über das STX-Programm erfolgt über folgenden Befehl:

```
SQLTest : CfbSQL4JetSym(
    enumTCPIPServerClient.Client,
    enumTCPIPProtocoll.TCP,
    '192.168.10.208',           //IP-Adresse
    1101                       //Portnummer
);
```

Parameter von CfbSQL4JetSym

| Parameter                  | Typ   | Bedeutung  |
|----------------------------|-------|--|
| tTimeout                   | Int8  | Timeout  |
| dwResponseMaxReturn-Params | DWORD | Maximale Anzahl an Return-Parametern                         |
| dwResponseMaxRows          | DWORD | Maximale Anzahl Zeilen in der Antwort der Datenbankabfrage   |
| dwResponseMaxColumns       | DWORD | Maximale Anzahl Spalten in der Antwort der Datenbankabfragen |

| Parameter                  | Typ                      | Bedeutung   |
|----------------------------|--------------------------|---|
| dwResponseMax-StringLen    | DWORD                    | Maximale String-Länge in der Antwort der Datenbankabfrage         |
| dwResponseCutStringLen     | DWORD                    | Strings in der Antwort werden auf die entsprechende Länge gekürzt |
| stResult.diResultState     | eSQL4JetSym_SQLErrorCode | Status der Datenbankabfrage/Fehlernummer                          |
| stResult.dwResultRows      | DWORD                    | Anzahl erhaltener Zeilen aus der Datenbankabfrage                 |
| stResult.dwResultColumns   | DWORD                    | Anzahl erhaltener Spalten aus der Datenbankabfrage                |
| stResult.sResultColumnName | Array of string          | Aus der Datenbankabfrage erhaltener Spaltenname                   |

**Tab. 1:** Parameter CfbSQL4JetSym

## 3 Arbeiten mit dem Beispielprojekt

Zusammen mit der Bibliothek wird ein Beispielprojekt geliefert, das als Basis für eigene Anpassungen an die eigene Applikation genutzt werden kann. Beachten Sie hierbei, dass dieses Programm fehlerfrei laufen sollte, bevor Sie Anpassungen vornehmen, um eventuell auftretende Fehler leichter finden und beheben zu können.

### INFO

Beachten Sie, dass zur Verwendung der SQL-Funktionen der Konnektor gestartet und die Verbindung zur Beispieldatenbank S4A\_Test\_DB.mdb konfiguriert sein muss. Mehr Informationen zu diesem Thema finden Sie im Benutzerhandbuch im Downloadbereich von SQL4automation.

(<https://www.sql4automation.com/de/lizenzmodelle/download.php>)

Das Beispielprojekt beinhaltet vier Beispielprogramme, die die Grundbefehle von SQL darstellen:

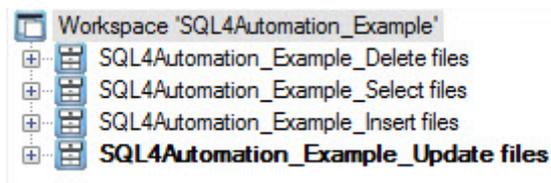


Abb. 3: Beispielprogramme

### 3.1 Aufbau des Beispielprojekts

Der Aufbau des Beispielprojekts besteht aus vier Schritten:

1. Aufruf der Funktion `Initialize()`.  
Damit wird die TCP/IP-Schnittstelle initialisiert und die Verbindung zum Konnektor aufgebaut. Wenn die Variable `DefaultInputParameter` den Wert `True` hat, werden zusätzlich Default-Parameter gesetzt. Andernfalls können manuell folgende Parameter definiert werden:
  - `InputFbSQL4.tTimeout := 30;`
  - `InputFbSQL4.dwResponseMaxReturnParams := 10;`
  - `InputFbSQL4.dwResponseMaxRows := 100;`
  - `InputFbSQL4.dwResponseMaxColumns := 15;`
  - `InputFbSQL4.dwResponseCutStringLength := 0;`
  - `InputFbSQL4.dwResponseMaxStringLength := 255;`
2. Aufruf der Funktion `Reset()`.  
In diesem Schritt werden sämtliche Variablen und Daten-Puffer gelöscht.
3. Erstellung der SQL-Query entsprechend der Kundenapplikation und Senden des Telegramms an die Datenbank durch das SQL4Automation-Framework. Anschließend wird auf eine Antwort vom Konnektor gewartet. Falls Daten zurückgeschickt werden, können diese bei erfolgreicher Ausführung ausgewertet werden. Im Fehlerfall oder wenn keine Antwort vom Konnektor eintrifft, wird eine entsprechende Fehlermeldung ausgegeben.
4. Aufruf der Funktion `Delete()`.  
Die Verbindung zum Konnektor wird hierdurch geschlossen.

### 3.2 Kommunikationsprinzip

Das Kommunikationsprinzip ist bei allen Beispielprojekten gleich:

Die Steuerung schickt eine Anfrage (Request) an die Datenbank und erhält eine Rückmeldung (Response). Der Datenfluss ist hierbei sekundär und es können Daten von der Datenbank sowohl gelesen (Select) als auch geschrieben werden (Insert, Update). Die exakte Syntax des SQL-Befehls wird einerseits vom Datenbank-Modell bestimmt, andererseits von der Aufgabe, die ausgeführt werden soll.

#### 3.2.1 Anfrage stellen (Request)

Eine Anfrage (Request) an die Datenbank kann sowohl über einen SELECT- als auch über einen INSERT-Befehl gestellt werden.

##### Beispiel: Select-Befehl

```
SQLTest.DefaultInputParameter := true;

//Initialisiere und starte die Anfrage
SQLTest.Initialize();
SQLTest.Reset();
SQLTest.AddRequest(
    'Select id,
    iParam,
    fParam2,
    sText1 FROM tTable WHERE id < 100 ORDER BY id ASC;'
);
```

##### Beispiel: Insert-Befehl

```
SQLTest.DefaultInputParameter := true;

//Initialisiere und starte die Anfrage
SQLTest.Initialize();
SQLTest.Reset();
SQLTest.AddRequest(
    StrFormat(
        'INSERT INTO tTable1(
        iParam1, fParam2, sText1)
    VALUES (
        %d, %g, 'This_is_a_string'); ',
        iWert,
        fWert)
    );
```

#### 3.2.2 Parsen der Antwort (Response)

Das Antwort-Telegramm der Datenbank wird beim Empfang auf Korrektheit geprüft. Wichtige Informationen werden intern gespeichert und stehen zum Auslesen der Daten zur Verfügung:

- OutputFbSQL4.stResult.diResultState
  - 0 = Kein Fehler
  - >0 = Fehlercode
- OutputFbSQL4.stResult.dwResultRows  
Anzahl der Datensätze im Response-Telegramm
- OutputFbSQL4.stResult.dwResultColumns  
Anzahl der Datenspalten im Response-Telegramm

**Auslesen der Daten** Zum Auslesen der Daten gibt es zwei Methoden:

1. Sobald die Abfrage erfolgreich durchgelaufen ist, stehen im Array `aResponseReturnParams[]` die empfangenen Datensätze. Der Zeilenindex und der Spaltenindex sind Null-basiert.
2. `OutputFbSQL4.stResult.sResultColumnName[]` gibt den Spaltennamen anhand des entsprechenden Spaltenindex zurück. Der Spaltenindex ist Null-basiert.

### Beispiel: Auslesen über Array (Methode 1)

```
FOR dwRow := 0 TO SQLTest.OutputFbSQL4.stResult.dwResultRows DO
  astSQLExampleData[dwRow].diID := StrToInt(
    SQLTest.aResponseReturnParams[dwRow, 0]);
  astSQLExampleData[dwRow].iParam1 := StrToInt(
    SQLTest.aResponseReturnParams[dwRow, 1])
  astSQLExampleData[dwRow].iParam2 := StrToFloat(
    SQLTest.aResponseReturnParams[dwRow, 2])
  astSQLExampleData[dwRow].sText :=
    SQLTest.aResponseReturnParams[dwRow, 3]
END_FOR;

if SQLTest.OutputFbSQL4.stResult.diResultState <> 0 then
  diErrorCount += 1;
else
  diDoneCount += 1;
END_IF;
```

## 4 Fehlercodes (Return Values)

Auftretende Fehler werden in der Variable `diResultState` beschrieben:

| Fehlernummer | Beschreibung  |
|--------------|---|
| -1           | Fehler beim Verbindungsaufbau   |
| -2           | Interner Fehler beim Verbindungsaufbau  |
| -3           | Ungültiger Parameter beim Verbindungsaufbau   |
| -4           | Fehler beim Senden  |
| -5           | Ungültiges Handle beim Senden/Empfangen   |
| -6           | Fehler beim Empfang   |
| -8           | Timeout   |
| -9           | Fehler beim Löschen der Verbindung  |
| 0            | Keine Fehler  |
| 1            | Unbekannter SQL-Befehl  |
| 2            | Abfrage gibt mehr Datensätze zurück als mit <code>dwMaxRows</code> definiert.<br>Abfrage anpassen oder <code>dwMaxRows</code> vergrößern.                             |
| 3            | Abfrage gibt mehr Spalten als mit <code>dwMaxColumns</code> definiert.<br>Abfrage anpassen oder <code>dwMaxColumns</code> vergrößern.                                 |
| 4            | Abfrage gibt mehr Daten zurück als die definierte Puffergröße.<br>Abfrage oder Puffergröße anpassen. <code>Size</code> im Konnektor und in der Steuerung vergrößern.  |
| 5            | Abfrage gibt min. einen Wert zurück, der größer ist als mit <code>MaxStringLength</code> definiert.<br>Abfrage anpassen oder <code>MaxStringLength</code> vergrößern. |
| 10           | Interner Konnektorfehler  |
| 11           | Interner Konnektorfehler.<br>Datenbank kann nicht geöffnet werden.  |
| 21           | Keine IP-Adresse definiert  |
| 22           | Keine Portnummer definiert  |
| 23           | Request-String ist leer   |
| 24           | Anzahl <code>MaxRows</code> nicht definiert / <code>diMaxRows=0</code>  |
| 25           | Anzahl <code>MaxColumns</code> nicht definiert / <code>diMaxColumns=0</code>  |
| 29           | Größe <code>MaxStringLength</code> nicht definiert / <code>diStringLength=0</code>  |
| 51           | Request-String ist größer als der Sendepuffer.  |
| >100         | Fehlernummern der ODBC Datenbankanbindung   |
| 40002        | Allgemeiner Fehler bei der SQL Abfrage. Request-String ist ungültig.  |

**Tab. 2:** Fehlercodes

Jetter AG  
Gräterstraße 2  
71642 Ludwigsburg  
[www.jetter.de](http://www.jetter.de)

E-Mail [info@jetter.de](mailto:info@jetter.de)  
Telefon +49 7141 2550-0

We automate your success.