

SIEMENS

Ingenuity for life

24/7

Industry Online Support

Home

Anbindung einer SQL-Datenbank an einen S7-1500 Software Controller

SIMATIC ODK 1500S-SQL-Treiber V3.1

<https://support.industry.siemens.com/cs/ww/de/view/109479140>

Siemens
Industry
Online
Support



Gewährleistung und Haftung

Hinweis

Die Anwendungsbeispiele sind unverbindlich und erheben keinen Anspruch auf Vollständigkeit hinsichtlich Konfiguration und Ausstattung sowie jeglicher Eventualitäten. Die Anwendungsbeispiele stellen keine kundenspezifischen Lösungen dar, sondern sollen lediglich Hilfestellung bieten bei typischen Aufgabenstellungen. Sie sind für den sachgemäßen Betrieb der beschriebenen Produkte selbst verantwortlich. Diese Anwendungsbeispiele entheben Sie nicht der Verpflichtung zu sicherem Umgang bei Anwendung, Installation, Betrieb und Wartung. Durch Nutzung dieser Anwendungsbeispiele erkennen Sie an, dass wir über die beschriebene Haftungsregelung hinaus nicht für etwaige Schäden haftbar gemacht werden können. Wir behalten uns das Recht vor, Änderungen an diesen Anwendungsbeispiele jederzeit ohne Ankündigung durchzuführen. Bei Abweichungen zwischen den Vorschlägen in diesem Anwendungsbeispiel und anderen Siemens Publikationen, wie z. B. Katalogen, hat der Inhalt der anderen Dokumentation Vorrang.

Für die in diesem Dokument enthaltenen Informationen übernehmen wir keine Gewähr.

Unsere Haftung, gleich aus welchem Rechtsgrund, für durch die Verwendung der in diesem Applikationsbeispiel beschriebenen Beispiele, Hinweise, Programme, Projektierungs- und Leistungsdaten usw. verursachte Schäden ist ausgeschlossen, soweit nicht z. B. nach dem Produkthaftungsgesetz in Fällen des Vorsatzes, der groben Fahrlässigkeit, wegen der Verletzung des Lebens, des Körpers oder der Gesundheit, wegen einer Übernahme der Garantie für die Beschaffenheit einer Sache, wegen des arglistigen Verschweigens eines Mangels oder wegen Verletzung wesentlicher Vertragspflichten zwingend gehaftet wird. Der Schadensersatz wegen Verletzung wesentlicher Vertragspflichten ist jedoch auf den vertragstypischen, vorhersehbaren Schaden begrenzt, soweit nicht Vorsatz oder grobe Fahrlässigkeit vorliegt oder wegen der Verletzung des Lebens, des Körpers oder der Gesundheit zwingend gehaftet wird. Eine Änderung der Beweislast zu Ihrem Nachteil ist hiermit nicht verbunden.

Weitergabe oder Vervielfältigung dieser Anwendungsbeispiele oder Auszüge daraus sind nicht gestattet, soweit nicht ausdrücklich von der Siemens AG zugestanden.

Security-hinweise

Siemens bietet Produkte und Lösungen mit Industrial Security-Funktionen an, die den sicheren Betrieb von Anlagen, Lösungen, Maschinen, Geräten und/oder Netzwerken unterstützen. Sie sind wichtige Komponenten in einem ganzheitlichen Industrial Security-Konzept. Die Produkte und Lösungen von Siemens werden unter diesem Gesichtspunkt ständig weiterentwickelt. Siemens empfiehlt, sich unbedingt regelmäßig über Produkt-Updates zu informieren.

Für den sicheren Betrieb von Produkten und Lösungen von Siemens ist es erforderlich, geeignete Schutzmaßnahmen (z. B. Zellschutzkonzept) zu ergreifen und jede Komponente in ein ganzheitliches Industrial Security-Konzept zu integrieren, das dem aktuellen Stand der Technik entspricht. Dabei sind auch eingesetzte Produkte von anderen Herstellern zu berücksichtigen.

Weitergehende Informationen über Industrial Security finden Sie unter <http://www.siemens.com/industrialsecurity>.

Um stets über Produkt-Updates informiert zu sein, melden Sie sich für unseren produktspezifischen Newsletter an. Weitere Informationen hierzu finden Sie unter <http://support.industry.siemens.com>.

Inhaltsverzeichnis

Gewährleistung und Haftung	2
1 Einleitung	5
2 Aufgabe	6
2.1 Übersicht	6
3 Lösung	7
3.1 Übersicht	7
3.2 Beschreibung der Kernfunktionalität	7
3.3 Hard- und Software-Komponenten	7
3.3.1 Unterstützte SQL-Datenbanksysteme.....	8
3.3.2 Unterstützte SQL-Datentypen	9
3.3.3 Performance Angaben	11
3.3.4 Verwendete Komponenten.....	11
3.4 Versionierung und Update des Treibers	12
3.5 Neuerungen ODK 1500S SQL-Treiber V3	14
3.6 Lizenzmodell	14
4 Funktionsweise	16
4.1 Übersicht Funktionsprinzip	16
4.1.1 Schreiben von Prozessdaten	16
4.1.2 Universelle SQL-Schnittstelle.....	17
4.2 Übersicht Funktionsbausteine	18
4.3 Beschreibung der Funktionsbausteine.....	19
4.3.1 ODK1500S_SQL_V3_Load	19
4.3.2 ODK1500S_SQL_V3_Execute	19
4.3.3 ODK1500S_SQL_V3_ExecuteSimple	22
4.3.4 ODK1500S_SQL_V3_InsertUpdate.....	24
4.3.5 ODK1500S_SQL_V3_Unload	26
4.3.6 Informationen zum Verbindungsaufbau	26
4.4 Konfigurations-Datenbaustein erstellen	27
4.4.1 Beispiel-Datenbaustein	27
4.4.2 Konfigurations-Datenbaustein mit dem ConfigDBCcreator erstellen	29
4.4.3 Einschränkungen ODK1500S ConfigDBCcreator.....	33
5 Installation und Inbetriebnahme	34
5.1 Installation und Inbetriebnahme der MSSQL Demo-Datenbank.....	34
5.1.1 Installation MSSQL Server 2014 Express and Tools	34
5.1.2 Konfiguration des SQL-Servers	37
5.1.3 Erstellen der Demodatenbank mithilfe eines Skripts	38
5.2 Installation und Inbetriebnahme des SQL-Treibers auf dem Runtime Rechner	39
5.2.1 Installation der SQL-Treiber-DLL	39
5.2.2 Parametrierung des ODBC-Treibers.....	40
6 Bedienung der Applikation	41
6.1 Verwenden der Beispielapplikation	41
6.1.1 Aufbau des STEP 7 Beispielprojektes	42
6.1.2 Laden der SQL-Treiber-DLL.....	44
6.1.3 Ausführen von SQL-Anweisungen	45
6.1.4 Prozessdaten schreiben.....	45
6.1.5 SQL-Treiber-DLL entladen	46
6.1.6 Fehleranalyse.....	46
6.2 Anpassen der Beispielapplikation an eigene Anforderungen	46

6.2.1	Anpassen der Verbindungsinformationen	46
6.2.2	Anpassen der SQL-Anweisungen	47
7	Fehlermeldungen	48
7.1	Status und Error Codes ODK1500S	48
7.2	Error Codes ODK1500S SQL-Treiber	51
7.2.1	Function Error Codes	51
7.2.2	SQL Error Info	51
8	Ausgabe und Anzeige von Log-Informationen.....	52
9	Literaturhinweise	54
10	Ansprechpartner	55
11	Historie.....	56

1 Einleitung

Dieses Dokument beschreibt die Anbindung eines S7-1500 Software Controller an eine SQL-Datenbank mit dem ODK 1500S-SQL-Treiber.

Es wird unter anderem beschrieben, wie mit Hilfe des Treibers Daten von einer SQL-Datenbank gelesen werden oder Daten aus einem Datenbaustein in eine SQL-Datenbank geschrieben werden können.

2 Aufgabe

2.1 Übersicht

Einleitung

In der Automatisierungstechnik gibt es zahlreiche Anwendungen für eine Datenbank.

Datenbanken ermöglichen eine zentrale, schnelle, strukturierte und zuverlässige Ablage von Daten.

Speicherprogrammierbare Steuerungen, wie z.B. S7-1500 Controller werden für die Steuerung von Maschinen und Produktionsanlagen eingesetzt, sind aber kaum für die Ablage von großen Datenmengen geeignet die bei der Produktion anfallen oder benötigt werden.

Eine Anbindung einer Datenbank an eine speicherprogrammierbare Steuerung bietet daher zahlreiche Vorteile und Anwendungsfälle.

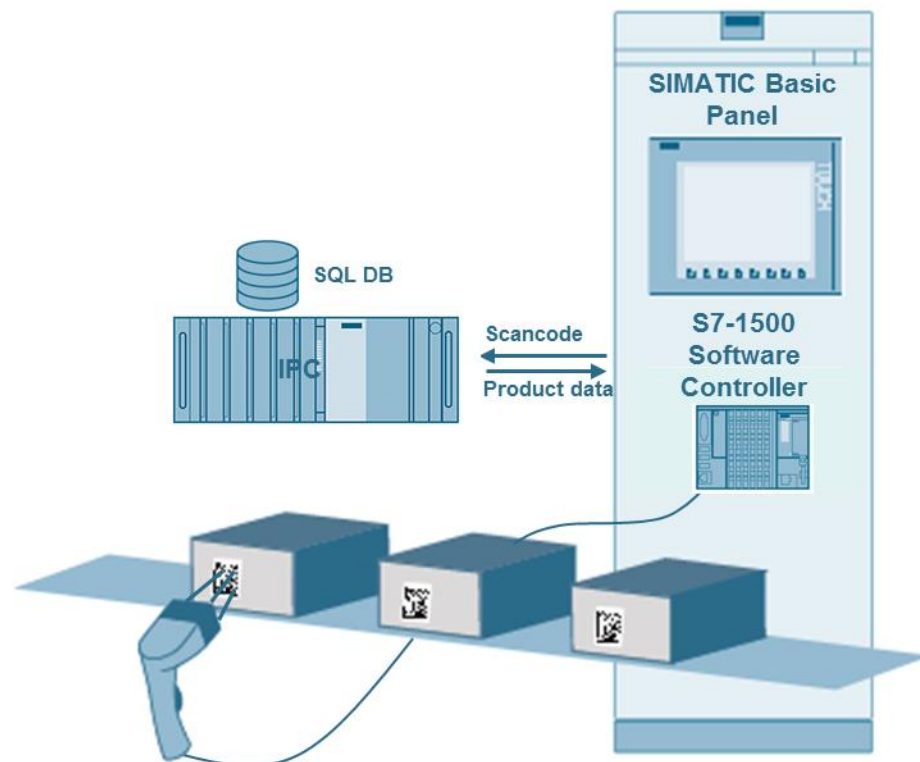
Zum Beispiel können Datenbanken genutzt werden, um Maschinendaten oder Meldungen während der Produktion zu loggen. In diesem Fall schreibt die Steuerung Daten in die Datenbank.

In einem anderen Anwendungsfall werden für die Produktion einer bestimmten Produktreihe benötigte Parameter aus einer zentralen Datenbank von der Steuerung ausgelesen.

Überblick über die Automatisierungsaufgabe

Folgende Abbildung gibt einen Überblick über eine mögliche Automatisierungsaufgabe.

Abbildung 2-1: Beispiel Automatisierungsaufgabe



3 Lösung

3.1 Übersicht

Eine Anbindung einer Datenbank an eine S7-1500 Steuerung ist mit Bordmitteln nicht möglich.

Zur Lösung des Problems wurde mit dem ODK 1500S (Open Development Kit) ein SQL-Treiber für den S7-1500 Software Controller entwickelt. Mit dem SQL-Treiber können Daten von einem S7-1500 Software Controller an eine oder mehrere SQL-Datenbanken übertragen oder Daten von Datenbanken gelesen werden.

Die Verbindung zur Datenbank (lokal oder remote) erfolgt i.d.R. über die ODBC-Schnittstelle (Open Database Connectivity).

Der ODK 1500S-SQL-Treiber besteht aus Funktionsbausteinen für das S7-Anwenderprogramm sowie einer Windows-DLL.

Vorausgesetzte Kenntnisse

Grundlegende Kenntnisse über SQL-Datenbanken, SQL-Anweisungen und TIA Portal werden vorausgesetzt.

3.2 Beschreibung der Kernfunktionalität

Folgende SQL-Datenbankfunktionen können mit Hilfe des Treibers aus dem STEP7 Anwenderprogramm ausgeführt werden:

- Übermitteln von SQL-Anweisungen mit flexiblen Argumenten.
 - SELECT
 - INSERT
 - UPDATE
 - DELETE
- Lesen von Daten aus einer SQL-Datenbank (über SELECT)
- Schreiben von Daten aus einem Datenbaustein in eine SQL-Datenbank

Hinweis

Die vier genannten SQL-Anweisungen sind getestet und somit freigegeben. Neben diesen Anweisungen können erfahrungsgemäß aber auch alle weiteren SQL-Anweisungen (z.B. Execute-Anweisungen für gespeicherte Prozeduren) genutzt werden, da der SQL-Treiber die vom Anwender übergebenen SQL-Strings und SQL-Arguments direkt an die SQL-Datenbank über die ODBC-Schnittstelle übergibt.

Aufgrund der Vielzahl von SQL-Anweisungen sowie möglichen Kombinationen können nicht alle Varianten getestet und somit auch nicht offiziell freigegeben werden. Der Einsatz nicht freigegebener SQL-Anweisungen erfolgt auf eigene Verantwortung.

3.3 Hard- und Software-Komponenten

Referenzanlage: Die Applikation wurde mit einer Laboranlage mit den nachfolgenden Komponenten erstellt und getestet. Die Verwendung in einem Produktivsystem erfolgt in der Verantwortung des Kunden. Evtl. notwendige Anpassungen sind individuell zu klären.

3.3.1 Unterstützte SQL-Datenbanksysteme

Der ODK1500S SQL-Treiber V3 wurde mit folgenden Datenbanken getestet:

- Microsoft SQL 2014 Server Express
- MySQL Server 5.7
- Oracle Database 11g Express Edition
- MariaDB 10.3
- PostgreSQL 12

Erfahrungsgemäß werden Datenbanksysteme, die die ODBC-Schnittstelle (Open Database Connectivity) unterstützen, vom ODK 1500S SQL-Treiber unterstützt. Einschränkungen kann es vor allem bei bestimmten Datentypen geben. Die Verwendung des ODK1500S SQL-Treibers mit nicht getesteten Datenbanken kann auf eigene Verantwortung erfolgen.

3 Lösung

3.3 Hard- und Software-Komponenten

3.3.2 Unterstützte SQL-Datentypen

Der SIMATIC ODK 1500S SQL-Treiber V3 unterstützt folgende SQL-Datentypen:

Tabelle 3-1: Unterstützte SQL-Datentypen

MS SQL Datentyp	My SQL Datentyp	Oracle SQL Datentyp	PostgreSQL Datentyp	STEP 7 Datentyp
varchar(n)		varchar2(n)		String[n]
nvarchar(n)	varchar(n)	nvarchar2	character	WString[n]
	enum			WString
bigint	bigint(20)		bigint	LInt
int	int(11)	number(10,0)	integer	DInt
smallint	smallint(6)	number(5,0)	smallint	Int
tinyint	tinyint(4)	number(3,0)		Sint
float, double	double	binary_double	double precision	LReal
real	float	binary_float	real	Real
bit	bit(1)	number(1,0)	bit****	Bool
datetime2(6)	datetime(6)	timestamp*	timestamp without timezone	DTL
datetime**	datetime(3)**			DTL
binary(n)***	binary(n)***			Array[1..n] of Byte
varbinary(n)***	varbinary(n)***			Array[1..n] of Byte
	tinyblob***			Array[1..255] of Byte
	blob***			Array[1..65535] of Byte
	mediumblob***			Array[1..65535] of Byte
	longblob			Array[1..65535] of Byte

3 Lösung

3.3 Hard- und Software-Komponenten

* Oracle: Zeitformat muss dem ODBC Standard z.B. über Login Trigger in DB für die Session angepasst werden:
NLS_DATE_FORMAT:YYYY-MM-DD; NLS_TIMESTAMP_FORMAT: YYYY-MM-DD HH24:MI:SS.FF

** Der Eingangsparameter "EnableDatimeSupport" muss am Baustein ODK1500_SQLV3_ExecuteStatement oder ODK1500_SQLV3_InsertUpdate aktiviert sein

*** nur lesend

**** nur schreibend

ACHTUNG

Stringlänge

Die maximale Stringlänge n beträgt 254 Zeichen. Dies gilt auch für den Datentyp nvarchar bzw. WString.

Variable Stringlänge in Leserichtung wird aktuell nur für Microsoft SQL-Datenbanken unterstützt. In Schreibrichtung (PLC->Datenbank) wird die variable Stringlänge für allen Datenbanken unterstützt.

3.3.3 Performance Angaben

- Bis zu 1025 Datenpunkte (Variablen) bzw. 128kByte Daten können pro Funktionsaufruf in eine Datenbank geschrieben oder aus einer Datenbank gelesen werden.
- Die SQL-Treiber FBs sind multiinstanzfähig. Somit können mehrere SQL-Befehle aus dem Anwenderprogramm parallel ausgeführt werden oder z.B. Daten in mehrere Datenbanken geschrieben werden.
- SQL-Anweisungen mit bis zu 16382 Zeichen im STEP7-Programm
- Die Ausführzeit einer Funktion ist u.a. abhängig von der Prozessorleistung des eingesetzten IPCs
- Durchschnittliche Ausführzeit zum Schreiben von 1000 Real-Werten (4kByte) aus einem Datenbaustein in eine Datenbank beträgt ca. 40 ms gemessen mit CPU 1505SP FW 2.7 (Open Controller), SQL Express 2014 Datenbank auf IPC227E

3.3.4 Verwendete Komponenten

Die Applikation wurde mit den nachfolgenden Komponenten erstellt und getestet:

Hardware-Komponenten

Tabelle 3-2

Komponente	Anz.	Bestellnummer	Hinweis
CPU 1515SP PC	1	6ES7 677-2DBx2-0xB0	ET 200SP Open Controller G2
IPC 227E	1	6ES7 647-8BB64-4BA1	Nanobox

Software-Komponenten

Die Mindestsoftwareanforderungen für den ODK1500S SQL-Treiber V3 sind:

- S7-1500 Software Controller FW 2.5 (Runtime System)
- STEP7 Professional V15 (Engineering)

Neuere Versionen der oben genannten Komponenten (z.B. TIA V16) können ohne Einschränkungen eingesetzt werden.

Beim Testen der Software wurden folgende Komponenten verwendet:

Tabelle 3-3

Komponente	Anz.	Bestellnummer	Hinweis
STEP 7 Professional V15.1	1		
Microsoft SQL 2014 Server Express Edition	1		64 Bit - Version
Microsoft ODBC Driver 17 for SQL Server	1		64 Bit - Version
MySQL Server 5.7	1		64 Bit - Version
MySQL Connector/ODBC 5.8	1		64 Bit – Version MySQL ODBC 5.8 Unicode Driver

Komponente	Anz.	Bestellnummer	Hinweis
Oracle Database 19c Client	1		https://www.oracle.com/database/technologies/oracle19c-windows-downloads.html (WINDOWS.X64_193000_client.zip)
CPU 1507S FW 2.7	1	6ES7 672-7AC01-0YA0	Installiert auf IPC 227E (Windows 7)
CPU 1515SP PC FW 2.7	1	6ES7 677-2DBx2-0xB0	Open Controller Gen. 2 (Windows 10)
SQL-Treiber-DLL V3.0	1		Windows-DLL

Beispieldateien und Projekte

Die folgende Liste enthält alle Dateien und Projekte, die in diesem Beispiel verwendet werden.

Tabelle 3-4

Komponente	Hinweis
ODK1500S-SQL-V3_1_x.zip	Diese Datei enthält <ul style="list-style-type: none"> • STEP 7 V15 SP1 Projekt • SCL-Quellen • SQL-Treiber-DLL • Lizenzdatei • SQL-Skript "DEMO_MSSQL.sql" • ConfigDBCreator

3.4 Versionierung und Update des Treibers

Die Treiber DLL befindet sich nach der Installation in folgendem Ordner:

<\$ProgramData>\Siemens\Automation\ODK1500S\ODK1500S_SQL_V3.dll

Die Version des Treibers kann über die Datei-Eigenschaften herausgefunden werden (Rechtsklick auf die Treiber DLL -> Properties -> Details)

Bei der Entwicklung des ODK1500S SQ-Treibers wird das Konzept der semantischen Versionierung eingesetzt. Das bedeutet:

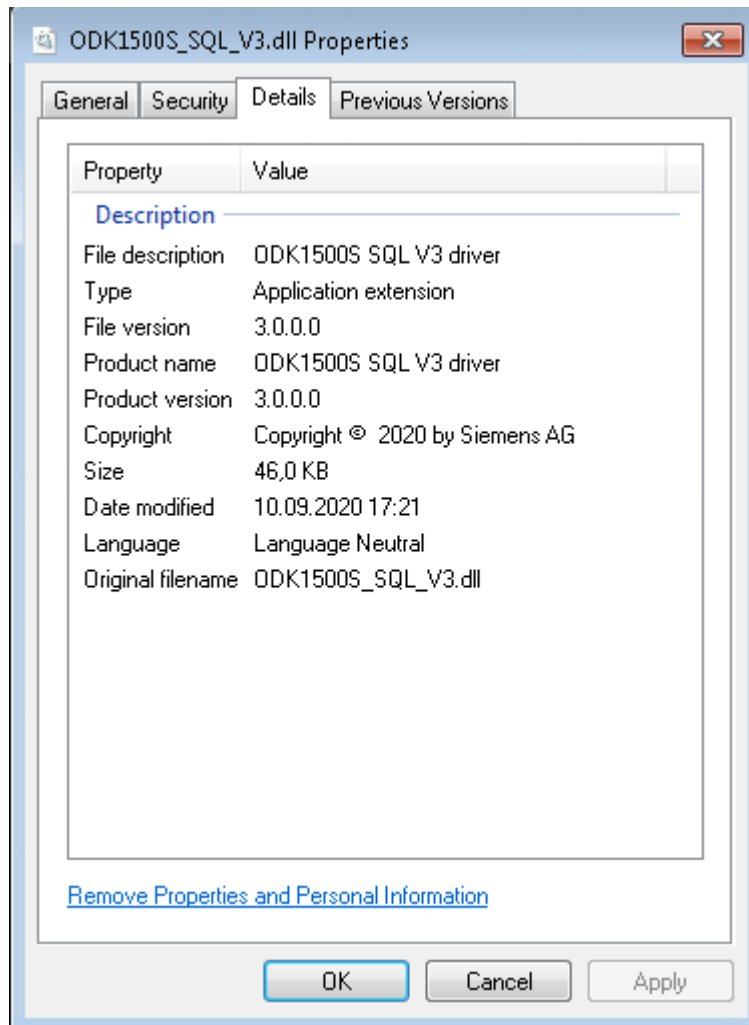
- Eine Versionsnummer besteht aus drei Teilen: MAJOR.MINOR.PATCH, zum Beispiel 2.2.3
- Eine Änderung der PATCH-Versionszahl bedeutet eine Fehlerkorrektur oder kleine Funktionserweiterung, die voll kompatibel zu früheren Versionen ist. Hier hat sich also nur intern etwas geändert. Zum Update des Treibers reicht es in diesem Fall die Treiber DLL auszutauschen
- Eine Änderung der MINOR-Versionszahl bedeutet eine Erweiterung der Funktionen. Es gibt neue Funktionen, die alten Funktionen funktionieren aber wie bisher. Oft wird in dem Zusammenhang die Schnittstelle der Funktionsbausteine angepasst oder erweitert. Zum Update des Treibers ist demnach ein Austausch der Treiber DLL notwendig, sowie die

3.4 Versionierung und Update des Treibers

Aktualisierung der Funktionsbausteine. Die FBs können z.B. über die mitgelieferte SCL-Quelle aktualisiert werden.

- Eine Änderung der MAJOR-Versionszahl bedeutet, dass es Umfangreiche Änderungen gibt. Die Kompatibilität zu früheren Versionen und Firmwareständen der CPU geht dabei i.d.R. verloren. Für das Hochrüsten ist oft ein Update der CPU-FW notwendig sowie die Aktualisierung der Treiber FBs und der Austausch der Treiber DLL.

Abbildung 3-1 Eigenschaften der Treiber DLL



ACHTUNG Wenn die Version der installierten Treiber DLL nicht zu der Version der STEP 7 Funktionsbausteine passt, wird folgende Fehlermeldung beim Laden der DLL ausgegeben:

0x8098 - ODK application could not be executed because the function is not supported.

3.5 Neuerungen ODK 1500S SQL-Treiber V3

Der ODK1500S SQL-Treiber V3 ist eine vollständige Neuentwicklung auf Basis ADO.NET und ODK1500S V2.5 und bietet einige Vorteile gegenüber den vorherigen Versionen. Aufgrund der Neuentwicklung hat sich auch die Anwenderschnittstelle geändert. Die wichtigsten Änderungen und Verbesserungen werden im Folgenden genannt:

- Jeder Treiber-FB ist nun multiinstanzfähig. Somit können verschiedene SQL-Befehle parallel ausgeführt werden oder z.B. Daten parallel in verschiedene Datenbanken geschrieben werden. (für die interne parallele Abarbeitung ist FW \geq 21.8 erforderlich)
- Die Verbindungsinformationen werden bei den Treiber FBs über einen „Connection String“ Parameter vom Anwender im SPS Programm vorgegeben. Eine separate Parametrierung des ODBC Treibers sowie das Handling der Datenbankverbindungen im SPS Programm (Connect / Disconnect) durch den Anwender entfällt somit.
- Es gibt nur noch einen Platzhalter für die Argumente. Anstatt der verschiedenen Platzhalter (%d, %s, %t...) für verschiedene Datentypen wird der universelle Platzhalter „?“ verwendet.
- Vereinfachtes Fehlerhandling: Fehlermeldungen werden zum Großteil auch als Klartext an den FBs ausgegeben. Außerdem wurde die Anzahl der Fehlercodes reduziert.
- Der ODK1500S-SQL-Treiber V3 ist eine nun 64-Bit-Applikation (erfordert nun auch einen 64 Bit ODBC Treiber)
- Ausgabe von Loginformationen FB-granular mit Hilfe des Tools „DebugView++“.
- Unterstützung der gängigen Datenbanken

3.6 Lizenzmodell

Der SQL-Treiber ist durch eine anwenderbezogene Lizenzdatei geschützt. Die Lizenzdatei wird bei der Installation des Treibers auf den Runtime Rechner (S7 1500 Software Controller) kopiert.

Bei jedem Ladevorgang des SQL-Treibers wird die Lizenzdatei auf Gültigkeit geprüft. Wenn die Lizenzdatei ungültig ist, wird der Ladevorgang beendet, es wird ein Fehlercode ausgegeben und die Funktionalität der SQL-Treiber FBs geht verloren.

Die Lizenzdatei ist eine digital signierte XML-Datei, die folgende Informationen enthält:

- Anschrift und Firmenname des Lizenzinhabers
- Ausstelldatum
- Ablaufdatum
- Trial-Lizenz (ja / nein)
- Version SQL-Treibers (Ausstellversion)
- Applikationsname
- Signatur

Die Datei kann z.B. über einen Webbrowser angezeigt werden, um Informationen über das Ablaufdatum bei einer Trial-Lizenz darzustellen:

Abbildung 3-2: Lizenzdatei ODK1500S-SQL-Treiber

```

▼<license>
  <name>Siemens AG</name>
  <address>Franz-Geuer-Str. 10, 50823 Köln, Deutschland</address>
  <issue>28.09.2020</issue>
  <expiry>11.11.2111</expiry>
  <trial>false</trial>
  <version>3</version>
  <application>ODK1500S SQL driver V3</application>
  ▼<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
    ▼<SignedInfo>
      <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
      <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
      ▼<Reference URI="">
        ▼<Transforms>
          <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
          </Transforms>
          <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
          <DigestValue>pIusp5MBZuiGKgMZ8z39PAJv8Sw=</DigestValue>
          </Reference>
        </SignedInfo>
        <SignatureValue>w/Hc2e9V26RppTvkMt+aRk78ccPIv34Yj0fa5a14ED0tfzvqgr3/jGG0uz67+Pheud84cQp+
      </Signature>
    </license>
  
```

Die Datei darf nicht geändert werden. Wenn die Datei manipuliert wird, verliert die Signatur die Gültigkeit und der SQL-Treiber kann nicht mehr genutzt werden.

Je nach Lizenzdatei verhält sich der Treiber folgendermaßen:

Tabelle 3-5 Verhalten der Lizenzierung

	Lizenz-Status	Beschreibung
1.	Kauf-Lizenz	Volle Funktionalität
2.	Testlizenz (nicht abgelaufen)	Volle Funktionalität
3.	Abgelaufene Testlizenz	Keine Funktionalität, FB ODK1500S_SQL_V3_Load gibt Fehlercode „0xF000“ aus.
4.	Manipulierte Lizenzdatei	Keine Funktionalität, FB ODK1500S_SQL_V3_Load gibt Fehlercode „0xF001“ aus.
5.	Keine Lizenzdatei vorhanden	Keine Funktionalität, FB ODK1500S_SQL_V3_Load gibt Fehlercode „0xFFFF“ aus.

4 Funktionsweise

In diesem Kapitel wird das Funktionsprinzip sowie die Funktionsbausteine vom ODK 1500S SQL-Treiber beschrieben. Außerdem wird auf den Aufbau der Verbindung zum SQL-Datenbankserver und die möglichen Funktionen eingegangen.


Für die Verwendung einer SQL-Datenbank mit einem S7-1500 Software Controller stehen insgesamt fünf Funktionsbausteine zur Verfügung. Mit diesen Funktionsbausteinen wird:

- Die SQL-Treiber DLL geladen
- Die Datenbankverbindung aufgebaut
- SQL-Anweisungen ausführen (z.B. Insert, Update, Execute, Select...)
- Daten in eine SQL-Datenbank geschrieben
- Daten aus einer SQL-Datenbank gelesen
- Die Datenbankverbindung geschlossen
- Die SQL-Treiber-DLL entladen

4.1 Übersicht Funktionsprinzip

Der ODK1500S SQL-Treiber V3 bietet zwei Anwenderschnittstellen, um Datenbank-Operationen aus dem STEP7 Anwenderprogramm durchzuführen:

Abbildung 4-1: Übersicht Anwenderschnittstellen

Schreiben von Prozessdaten 	Universelle SQL Schnittstelle
Einfaches, performantes Schreiben von Prozessdaten	Arbeit mit SQL Anweisungen → SQL-Wissen notwendig, dafür volle Flexibilität
Vereinfachtes Handling durch Config Tool → Definition der zu speichernden Daten	Parameter für Datenaustausch mit Datenbank im SPS Programm bereitstellen und versorgen
Kein SQL-Wissen notwendig	SQL-Befehle mit Platzhaltern gespeichert → kein Verarbeiten von SQL Strings nötig
	Vereinfachtes Handling durch Config Tool → Definition der zu verarbeiteten Daten

Diese Schnittstellen werden über entsprechende Funktionsbausteine bereitgestellt.

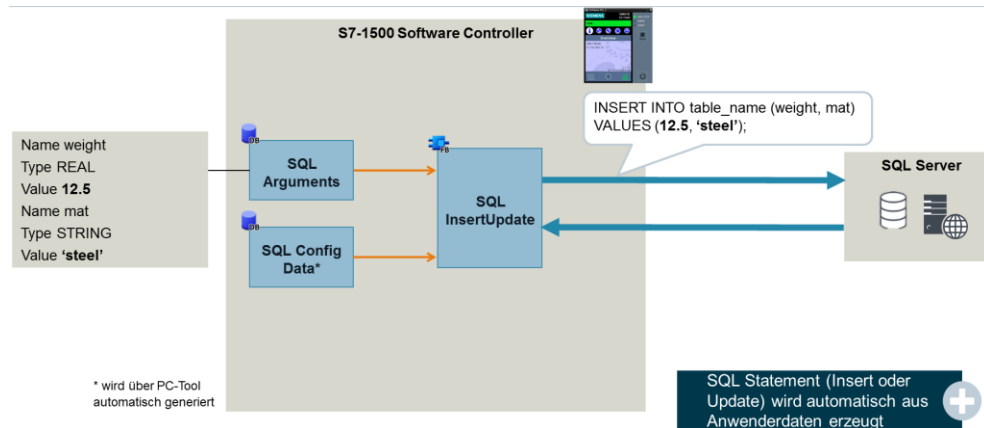
4.1.1 Schreiben von Prozessdaten

Prozessdaten aus einem Datenbaustein können einfach und performant über den Baustein ODK1500S_SQL_V3_InsertUpdate in eine Datenbank geschrieben werden. Das Funktionsprinzip dieser Schnittstelle wird in der folgenden Abbildung dargestellt:

4 Funktionsweise

4.1 Übersicht Funktionsprinzip

Abbildung 4-2: Funktionsprinzip FB ODK1500S_SQL_V3_InsertUpdate



Die Daten, die in die Datenbank geschrieben werden sollen, werden dem FB als Struktur übergeben. Über Eingangsparameter kann gewählt werden, ob Datensätze überschrieben werden sollen (Update) oder neu in die Datenbank hinzugefügt werden sollen (Insert). Außerdem kann der Name der Ziel-Tabelle in der Datenbank parametrisiert werden.

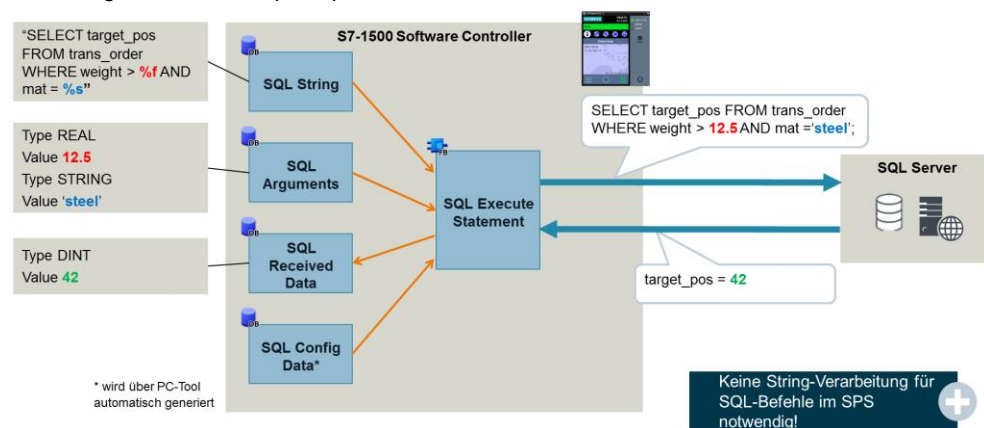
4.1.2 Universelle SQL-Schnittstelle

Über die Funktionsbausteine ODK1500S_SQL_V3_Execute und ODK1500S_SQL_V3_ExecuteSimple wird eine universelle SQL-Schnittstelle bereitgestellt. Über diese Schnittstelle können beliebige SQL-Anweisungen ausgeführt werden, um z.B. Daten in eine Datenbank zu schreiben oder Daten aus einer Datenbank, anzufordern.

Die eigentlichen Daten können separat zu den SQL-Anweisungen (SQL String) als Datenbaustein übergeben werden.

Das Funktionsprinzip wird anhand einer Select-Anweisung in der folgenden Abbildung dargestellt.

Abbildung 4-3: Funktionsprinzip FB ODK1500S_SQL_V3_ExecuteStatement

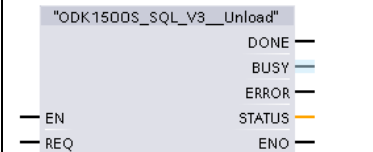


4.2 Übersicht Funktionsbausteine

Die folgende Tabelle gibt eine Übersicht über die zum SQL-Treiber gehörenden Funktionsbausteine sowie deren Funktion.

Tabelle 4-1: Übersicht Funktionsbausteine SQL-Treiber

Bild	Bausteinname	Funktion
<p>"ODK1500S_SQL_V3_Load"</p> <ul style="list-style-type: none"> EN REQ DONE BUSY ERROR STATUS ENO 	ODK1500S_V3_Load	SQL-Treiber DLL laden
<p>"ODK1500S_SQL_V3_ExecuteStatementSimple"</p> <ul style="list-style-type: none"> EN REQ sqlConnectionS sqlString commandTime out connectionTim eout enableLog ERROR DONE BUSY STATUS affectedRows fctError fctErrorcode sqlErrorInfo ENO 	ODK1500S_SQL_V3_ExecuteStatementSimple	SQL-Anweisung ausführen (nur für Anweisungen, die keine Daten zurückgeben)
<p>"ODK1500S_SQL_V3_Execute"</p> <ul style="list-style-type: none"> EN REQ sqlConnectionS sqlString ConnectionTim eout CommandTime out enableDatetim eSupport enableLog sqlArguments sqlArgumentsC onfigData sqlReceivedDat a ERROR DONE BUSY STATUS affectedRows fctError fctErrorcode sqlErrorInfo ENO 	ODK1500S_SQL_V3_Execute	SQL-Anweisung mit variablen Argumenten ausführen (ermöglicht das Schreiben und Lesen von Daten über SQL-Anweisungen)
<p>"ODK1500S_SQL_V3_InsertUpdate"</p> <ul style="list-style-type: none"> EN REQ sqlConnectionS tableName updateCondition writeMode ConnectionTim eout CommandTime out enableDatetim eSupport enableLog sqlArguments sqlArgumentsC onfigData ERROR DONE BUSY STATUS affectedRows fctError fctErrorcode sqlErrorInfo ENO 	ODK1500S_SQL_V3_InsertUpdate	Daten aus einem Datenbaustein in eine SQL-Datenbank schreiben (ohne SQL-Anweisung)

Bild	Bausteinname	Funktion
	ODK1500S_SQL_V3_Unload	Manuelles Entladen der DLL für z.B. Austausch der DLL

4.3 Beschreibung der Funktionsbausteine

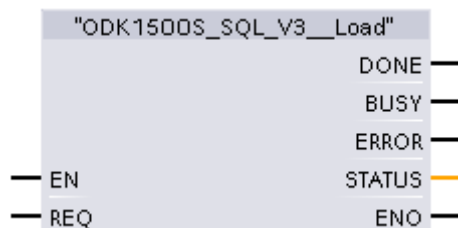
Im Folgenden werden die Treiber-FBs detailliert beschrieben.

4.3.1 ODK1500S_SQL_V3_Load

Bevor der SQL-Treiber genutzt werden kann muss die mitgelieferte Treiber-DLL mit dem ODK1500_SQL_V3_Load-Baustein geladen werden. In der mitgelieferten Beispiellapplikation wird dies automatisch ausgeführt.

Durch einen Flankenwechsel (0 nach 1) am Eingangsparameter "REQ" wird die Funktion gestartet.

Abbildung 4-4: ODK1500S_SQL_V3_Load Baustein



Bausteinparameter

Tabelle 4-2: Parameter ODK1500S_SQL_V3_Load

Typ	Name	Datentyp	Beschreibung
Input	REQ	BOOL	Eine steigende Flanke aktiviert das Laden der SQL-Treiber-DLL.
Output	DONE	BOOL	Zeigt an, dass die Anweisung das Laden der SQL-Treiber-DLL erfolgreich beendet hat.
Output	BUSY	BOOL	Zeigt an, dass die Anweisung die SQL-Treiber-DLL noch lädt.
Output	ERROR	BOOL	Zeigt an, dass ein Fehler während des Ladens der SQL-Treiber-DLL aufgetreten ist. STATUS gibt Ihnen weitere Informationen über die mögliche Fehlerursache.
Output	STATUS	INT	Gibt Auskunft über mögliche Fehlerquellen, wenn ein Fehler während des Ladens der SQL-Treiber-DLL aufgetreten ist. Dies ist zum Beispiel der Fall, wenn keine gültige Lizenz vorhanden ist.

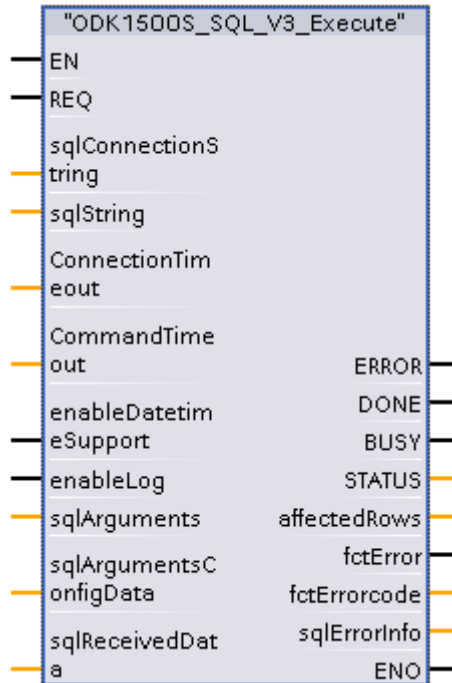
4.3.2 ODK1500S_SQL_V3_Execute

Mit dem Baustein ODK1500S_SQL_V3_Execute können SQL-Anweisungen an eine Datenbank gesendet werden. Der Verbindungsaufbau erfolgt über die am Parameter „sqlConnectionString“ hinterlegten Verbindungsinformationen. Dieser Baustein kann beliebig oft hintereinander oder als Multi Instanz parallel aufgerufen werden, um u.a. SELECT-, UPDATE-, INSERT- oder DELETE-Anweisungen auszuführen. Eine Anweisung wird asynchron bearbeitet. Es werden mehrere PLC-

4.3 Beschreibung der Funktionsbausteine

Zyklen durchlaufen, um den Baustein abzuarbeiten und zu beenden. Die Zykluszeit des PLC-Programms wird dadurch nicht nennenswert beeinflusst. Das Funktionsprinzip des Bausteins wird in Kapitel Universelle beschrieben.

Abbildung 4-5: ODK1500S_SQL_V3_Execute Baustein



Damit im Steuerungsprogramm die eigentlichen Daten, die z.B. in eine Datenbank geschrieben werden sollen nicht in Strings konvertiert werden müssen, können die Daten und die SQL-Anweisungen mit Hilfe von Platzhaltern getrennt an den Baustein übergeben werden:

SQL-Anweisungen bestehen dann in der Applikation aus zwei Teilen:

- SQL-String: enthält die eigentliche SQL-Anweisung mit Platzhaltern („?“) für die SQL-Argumente
- SQL-Argument: hier werden die eigentlichen Daten definiert

Der Baustein fügt den SQL-String und die SQL-Argumente wieder zusammen und sendet die vollständige SQL-Anweisung an die Datenbank.

Beispiel SQL-String:

```
SELECT year, test FROM ? WHERE titleid < ?
```

wobei „?“ jeweils als Platzhalter für die Argumente dient:

```
? => "titles"
? => 5
```

Die vollständige Anweisung für die Datenbank würde dann folgendermaßen interpretiert werden:

```
SELECT year, test FROM titles WHERE titleid < 5
```

Für die Platzhalter werden die entsprechenden Argumente der Reihe nach eingesetzt.

Die Anzahl der Argumente muss mit der Anzahl der Platzhalter übereinstimmen.

Hinweis

Die Daten und die SQL-Anweisungen müssen nicht unbedingt getrennt an den Baustein übergeben werden. Alternativ können auch vollständige SQL-Anweisungen am Eingangsparameter "sqlString" übergeben werden. In diesem Fall ist am Eingangsparameter "sqlArgument" ein beliebiger unterstützter Datentyp mit dem Namen "NoArgument" zu verwenden.

Alternativ kann auch der Baustein ODK1500S_SQL_V3_ExecuteStamtemtSimple verwendet werden.

Eine SQL-Anweisung wird mit einer steigenden Flanke am Eingang "REQ" ausgeführt. Bei einer SELECT-Anweisung werden zusätzlich die von der Datenbank angeforderten Daten eingelesen und über den InOut Parameter "sqlReceivedData" bereitgestellt. Die Daten im verschalteten Empfangs-Datenbaustein müssen in der gleichen Reihenfolge vorliegen, wie die Daten, die vom Funktionsbaustein bzw. von der Datenbank zur Verfügung gestellt werden. Werden zu wenig Datentypen bereitgestellt, wird eine Fehlermeldung ausgegeben.

Bausteinparameter

Tabelle 4-3: Parameter ODK1500S_SQL_V3_Execute

Typ	Name	Datentyp	Beschreibung
Input	REQ	BOOL	Eine ansteigende Flanke dieses Input-Wertes aktiviert die Ausführung der ODK-Funktion.
Input	sqlConnectionString	STRING	Enthält die notwendigen Informationen für den Verbindungsaufbau. Im Connection String muss i.d.R. der Name des ODBC Treibers, die IP-Adresse des SQL Servers, der Name der Datenbank, der Benutzername und das Passwort hinterlegt werden. Beispiele für gängige Datenbanken sind im Beispielprojekt hinterlegt.
Input	sqlString	WSTRING	SQL-String, enthält die SQL-Anweisung mit den definierten Platzhaltern für die Argumente. Die maximale Länge des Strings beträgt ca. 16000 Zeichen.
Input	connectionTimeout	UINT	Timeout in Sekunden für den Verbindungsaufbau. Wenn innerhalb der angegebenen Zeit die Verbindung zur Datenbank nicht aufgebaut werden kann, wird die Ausführung des FBs beendet.
Input	commandTimeout	UINT	Timeout in Sekunden für das Ausführen von SQL-Befehlen.
Input	enableDatetimeSupport	BOOL	Setzt die Sekundenbruchteilgenauigkeit auf 3 Stellen, um die Unterstützung für den SQL "datetime" Datentyp zu ermöglichen. Bei nicht gesetztem Bit beträgt die Sekundenbruchteilgenauigkeit 6 Stellen (datetime2(6)).
Input	enableLog	BOOL	Aktiviert die Ausgabe von Log-Informationen. Diese können z.B. bei einer Inbetriebnahme mit Hilfe des Tools DebugView++ ausgelesen werden.
InOut	sqlArguments	VARIANT	SQL-Argument: An dieser Stelle muss der Datenbaustein mit der Struktur für die entsprechende SQL-Anweisung verschaltet werden. Diese Struktur enthält die Daten für die Platzhalter aus dem SQL-String.
InOut	sqlArgumentsConfigData	Array[0..1024] of "ODK1500S_SQL_V3_typeConfigDat	An dieser Stelle muss der Datenbaustein mit der Struktur mit den Konfigurationsdaten verschaltet werden. Die Konfigurationsdaten können mit dem mitgelieferten Tool "ConfigDBCcreator" anhand der sqlArguments automatisch erstellt werden.

4 Funktionsweise

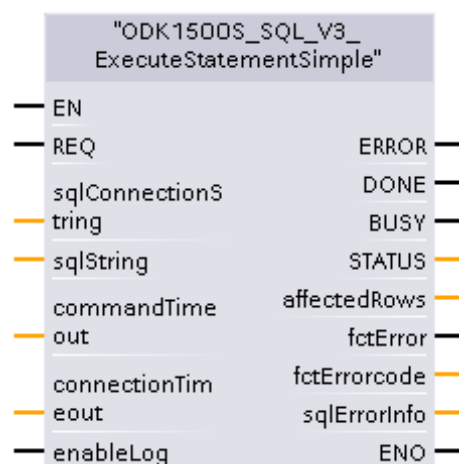
4.3 Beschreibung der Funktionsbausteine

Typ	Name	Datentyp	Beschreibung
		a	
InOut	sqlReceivedData	VARIANT	An dieser Stelle muss ein Datenbaustein mit der Struktur für die von der Datenbank zurückgegebenen Daten verschaltet werden.
Output	DONE	BOOL	Dieser Output-Wert zeigt an, dass die Anweisung die Ausführung der ODK-Funktion erfolgreich beendet hat.
Output	BUSY	BOOL	Dieser Output-Wert zeigt an, dass die Anweisung die ODK-Funktion noch ausführt.
Output	ERROR	BOOL	Dieser Output-Wert zeigt an, dass ein Fehler während der Ausführung der ODK-Funktion aufgetreten ist. Weitere Informationen darüber gibt der Output-Wert STATUS aus.
Output	STATUS	INT	Dieser Output-Wert gibt Auskunft über mögliche Fehlerquellen, wenn ein Fehler während der Ausführung der ODK-Funktion aufgetreten ist.
Output	affectedRows	DINT	Anzahl der Zeilen die durch die Ausführung des SQL-Befehls in der Datenbank geändert oder hinzugefügt wurden.
Output	fctError	BOOL	Dieser Output-Wert zeigt an, dass ein Fehler während der Ausführung einer SQL-Treiber-Funktion aufgetreten ist. Weitere Informationen darüber geben die Statuswerte an den Ausgängen fctErrorCode und sqlErrorInfo.
Output	fctErrorCode	DWORD	Fehlercode
Output	sqlErrorInfo	STRING	Enthält Informationen zu dem aufgetretenen Fehler von der SQL-Datenbank. Die Informationen sind abhängig vom eingesetzten ODBC Treiber bzw. der Datenbank.

4.3.3 ODK1500S_SQL_V3_ExecuteSimple

Alternativ zu dem Baustein ODK1500S_SQL_V3_Execute können mit diesem Baustein beliebige SQL-Anweisungen ausgeführt werden, die keine Daten aus der Datenbank anfordern.

Abbildung 4-6: ODK1500S_SQL_V3_ExecutedStatementSimple Baustein



Eine SQL-Anweisung wird mit einer steigenden Flanke am Eingang "REQ" ausgeführt.

Bausteinparameter

Tabelle 4-4: Parameter ODK1500S_SQL_V3_ExecuteSimple

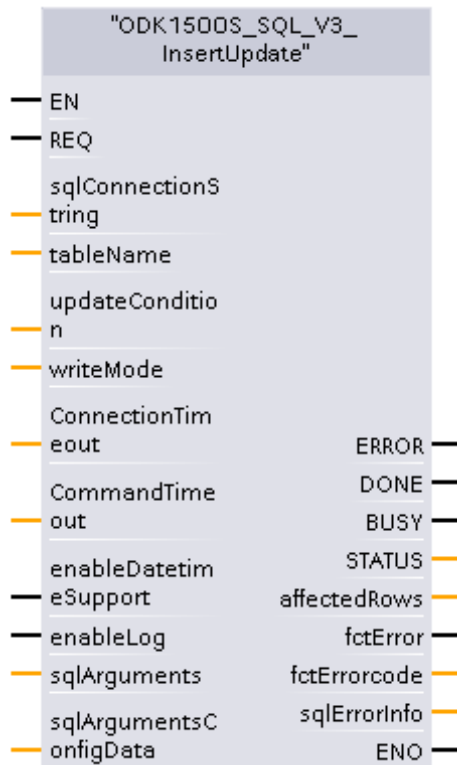
Typ	Name	Datentyp	Beschreibung
Input	REQ	BOOL	Eine ansteigende Flanke dieses Input-Wertes aktiviert die Ausführung der ODK-Funktion.
Input	sqlConnectionString	STRING	Enthält die notwendigen Informationen für den Verbindungsaufbau. Im Connection String muss i.d.R. der Name des ODBC Treibers, die IP-Adresse des SQL Servers, der Name der Datenbank, der Benutzername und das Passwort hinterlegt werden. Beispiele für gängige Datenbanken sind im Beispielprojekt hinterlegt.
Input	sqlString	WSTRING	SQL-String, enthält die SQL-Anweisung mit den definierten Platzhaltern für die Argumente. Die maximale Länge des Strings beträgt ca. 16000 Zeichen.
Input	connectionTimeout	UINT	Timeout in Sekunden für den Verbindungsaufbau. Wenn innerhalb der angegebenen Zeit die Verbindung zur Datenbank nicht aufgebaut werden kann, wird die Ausführung des FBs beendet.
Input	commandTimeout	UINT	Timeout in Sekunden für das Ausführen von SQL-Befehlen.
Input	enableLog	BOOL	Aktiviert die Ausgabe von Log-Informationen. Diese können z.B. bei einer Inbetriebnahme mit Hilfe des Tools DebugView++ ausgelesen werden.
Output	DONE	BOOL	Dieser Output-Wert zeigt an, dass die Anweisung die Ausführung der ODK-Funktion erfolgreich beendet hat.
Output	BUSY	BOOL	Dieser Output-Wert zeigt an, dass die Anweisung die ODK-Funktion noch ausführt.
Output	ERROR	BOOL	Dieser Output-Wert zeigt an, dass ein Fehler während der Ausführung der ODK-Funktion aufgetreten ist. Weitere Informationen darüber gibt der Output-Wert STATUS aus.
Output	STATUS	INT	Dieser Output-Wert gibt Auskunft über mögliche Fehlerquellen, wenn ein Fehler während der Ausführung der ODK-Funktion aufgetreten ist.
Output	affectedRows	DINT	Anzahl der Zeilen die durch die Ausführung des SQL-Befehls geändert oder hinzugefügt wurden.
Output	fctError	BOOL	Dieser Output-Wert zeigt an, dass ein Fehler während der Ausführung einer SQL-Treiber-Funktion aufgetreten ist. Weitere Informationen darüber geben die Statuswerte an den Ausgängen fctErrorCode und sqlErrorInfo.
Output	fctErrorCode	DWORD	Fehlercode
Output	sqlErrorInfo	STRING	Enthält Informationen zu dem aufgetretenen Fehler von der SQL-Datenbank. Die Informationen sind abhängig vom eingesetzten ODBC Treiber bzw. der Datenbank.

Hinweis String-Datentypen (z.B. varchar) müssen im SQL-String mit einfachen Anführungszeichen formatiert werden. Dies erfolgt im SQL-String mit '\$'. Zum Beispiel: '\$String\$'

4.3.4 ODK1500S_SQL_V3_InsertUpdate

Daten aus einem Datenbaustein können einfach und performant über den Baustein ODK1500S_SQL_V3_InsertUpdate in eine Datenbank geschrieben werden. Über die Baustein-Eingangsparameter kann die Tabelle in der Datenbank angegeben werden, in die die Daten geschrieben werden sollen.

Abbildung 4-7: ODK1500S_SQL_V3_InsertUpdate Baustein



Das Schreiben der Daten in die Datenbank erfolgt mit einer steigenden Flanke am Eingang "REQ".

ACHTUNG Da die Adressierung der Daten in der Datenbank über die Namen erfolgt, müssen die Namen der Variablen aus dem am Eingangsparameter sqlArguments verschalteten Datenbaustein, mit den Namen der Daten in der Datenbank übereinstimmen.
Falls dies nicht erfüllt ist, wird eine Fehlermeldung ausgegeben.

Bausteinparameter

Tabelle 4-5: Parameter ODK1500S_SQL_V3_InsertUpdate

Typ	Name	Datentyp	Beschreibung
Input	REQ	BOOL	Eine ansteigende Flanke dieses Input-Wertes aktiviert die Ausführung der ODK-Funktion.
Input	sqlConnectionString	STRING	Enthält die notwendigen Informationen für den Verbindungsaufbau. Im Connection String muss i.d.R. der Name des ODBC Treibers, die IP-Adresse des SQL Servers,

4 Funktionsweise

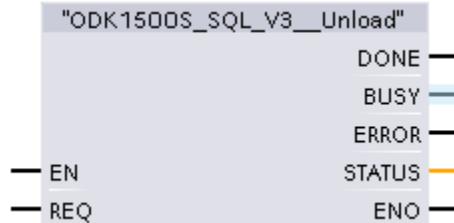
4.3 Beschreibung der Funktionsbausteine

Typ	Name	Datentyp	Beschreibung
			der Name der Datenbank, der Benutzername und das Passwort hinterlegt werden. Beispiele für gängige Datenbanken sind im Beispielprojekt hinterlegt.
Input	writeMode	BYTE	0: Insert – Daten werden in der Zieltabelle hinzugefügt, 1: Update - Daten werden in der Zieltabelle überschrieben bzw. aktualisiert
Input	tableName	STRING	Name der Tabelle in der Datenbank in die die Daten geschrieben werden sollen
Input	updateCondition	STRING	Ermöglicht beim writeMode 1 (Update) die Angabe der Datensätze die aktualisiert werden sollen. Zum Beispiel: "WHERE ID = 500")
Input	enableDatetime Support	Bool	Setzt die Sekundenbruchteilgenauigkeit auf 3 Stellen, um die Unterstützung für den SQL "datetime" Datentyp zu ermöglichen. Bei nicht gesetztem Bit beträgt die Sekundenbruchteilgenauigkeit 6 Stellen (datetime2(6)).
Input	connectionTime out	UINT	Timeout in Sekunden für den Verbindungsaufbau. Wenn innerhalb der angegebenen Zeit die Verbindung zur Datenbank nicht aufgebaut werden kann, wird die Ausführung des FBs beendet.
Input	commandTimeo ut	UINT	Timeout in Sekunden für das Ausführen von SQL-Befehlen.
Input	enableLog	BOOL	Aktiviert die Ausgabe von Log-Informationen. Diese können z.B. bei einer Inbetriebnahme mit Hilfe des Tools DebugView++ ausgelesen werden.
InOut	sqlArguments	VARIANT	SQL-Argument: An dieser Stelle muss ein Datenbaustein mit der Struktur für die in die Datenbank zu schreibenden Daten verschaltet werden. Die Namen der Variablen müssen mit den Namen der Daten in der Datenbank übereinstimmen.
InOut	sqlArgumentsC onfigData	Array[0..102 4] of "ODK1500S _SQL_V3_ty peConfigDat a	An dieser Stelle muss der Datenbaustein mit der Struktur mit den Konfigurationsdaten verschaltet werden. Die Konfigurationsdaten können mit dem mitgelieferten Tool "ConfigDBCcreator" anhand der sqlArguments automatisch erstellt werden.
Output	DONE	BOOL	Dieser Output-Wert zeigt an, dass die Anweisung die Ausführung der ODK-Funktion erfolgreich beendet hat.
Output	BUSY	BOOL	Dieser Output-Wert zeigt an, dass die Anweisung die ODK-Funktion noch ausführt.
Output	ERROR	BOOL	Dieser Output-Wert zeigt an, dass ein Fehler während der Ausführung der ODK-Funktion aufgetreten ist. Weitere Informationen darüber gibt der Output-Wert STATUS aus.
Output	STATUS	INT	Dieser Output-Wert gibt Auskunft über mögliche Fehlerquellen, wenn ein Fehler während der Ausführung der ODK-Funktion aufgetreten ist.
Output	affectedRows	DINT	Anzahl der Zeilen die durch die Ausführung des FBs geändert oder hinzugefügt wurden.
Output	fctError	BOOL	Dieser Output-Wert zeigt an, dass ein Fehler während der Ausführung einer SQL-Treiber-Funktion aufgetreten ist. Weitere Informationen darüber geben die Statuswerte an den Ausgängen fctErrorCode und sqlErrorInfo.
Output	fctErrorCode	DWORD	Fehlercode
Output	sqlErrorInfo	STRING	Enthält Informationen zu dem aufgetretenen Fehler von der SQL-Datenbank. Die Informationen sind abhängig vom eingesetzten ODBC Treiber bzw. der Datenbank.

4.3.5 ODK1500S_SQL_V3_Unload

Mit diesem Baustein kann die SQL-Treiber-DLL manuell entladen werden. Dies ist in der Regel nicht erforderlich. Ausnahme: Wenn eine neue Version der SQL-Treiber-DLL installiert werden soll, muss diese zuvor entladen werden.

Abbildung 4-8: ODK1500S_SQL_V3_Unload



Bausteinparameter

Tabelle 4-6: Parameter ODK1500S_SQL_V3_Unload

Typ	Name	Datentyp	Beschreibung
Input	REQ	BOOL	Eine steigende Flanke aktiviert das Laden der SQL-Treiber-DLL.
Output	DONE	BOOL	Zeigt an, dass die Anweisung das Laden der SQL-Treiber-DLL erfolgreich beendet hat.
Output	BUSY	BOOL	Zeigt an, dass die Anweisung die SQL-Treiber-DLL noch lädt.
Output	ERROR	BOOL	Zeigt an, dass ein Fehler während des Ladens der SQL-Treiber-DLL aufgetreten ist. STATUS gibt Ihnen weitere Informationen über die mögliche Fehlerursache.
Output	STATUS	INT	Gibt Auskunft über mögliche Fehlerquellen, wenn ein Fehler während des Ladens der SQL-Treiber-DLL aufgetreten ist.

4.3.6 Informationen zum Verbindungsaufbau

Die Treiber FBs Execute, ExecuteStatementSimple und InsertUpdate bauen beim ersten Aufruf eine Verbindung zur Datenbank über den im Connection String angegebenen ODBC Treiber auf. Wenn für den ODBC Treiber die Eigenschaft „Connection Pooling“ aktiviert ist wird die Verbindung bis zum Ablauf der parametrisierten Zeit aufrechterhalten, so dass die Ausführzeit des FBs bei erneutem Aufruf deutlich geringer ist.

Wie man das Connection Pooling einstellen kann wird im Kapitel: Parametrierung des ODBC-Treibers beschrieben.

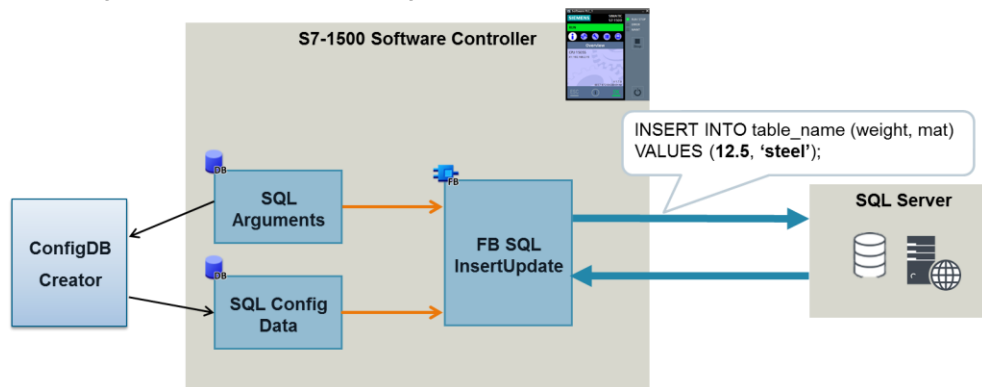
4.4 Konfigurations-Datenbaustein erstellen

Der SQL-Treiber benötigt Informationen (Datentyp und Name) von den Daten, die in die Datenbank geschrieben werden sollen.

Da die Anwenderdaten an den SQL-Treiber intern nur serialisiert als Byte-Array übergeben werden, müssen die Informationen bei Verwendung der Bausteine ODK1500S_SQL_V3_InsertUpdate oder ODK1500S_SQL_V3_Execute separat über den Eingangsparameter "sqlConfigData" übergeben werden.

Die Konfigurationsdaten können mit dem mitgelieferten Tool "ConfigDBCcreator" automatisch aus einem Datenbaustein (sqlArguments) erstellt werden:

Abbildung 4-9: Funktionsprinzip Konfigurations-Datenbaustein



Die Konfigurationsdaten werden jeweils als Array mit 1025 Elementen vom PLC-Datentyp "ODK1500S_SQL_V3_typeConfigData" angelegt.

4.4.1 Beispiel-Datenbaustein

Die folgende Abbildung enthält verschiedene Daten, die in eine Datenbank geschrieben werden sollen. Diese sind jeweils in einer Struktur zusammengefasst.

Abbildung 4-10: SQL-Arguments Datenbaustein

ODK1500S_SQL_V2_ARGUMENTS_DOKU			
Name	Data type	Start value	
Static			
SQL_Data_Insert	Struct		
id	Dint	500	
demo_integer	Dint	501	
demo_smallint	Int	502	
demo_tinyint	Sint	50	
demo_double	LReal	500.55555555...	
demo_bigint	LInt	50000000000...	
demo_char	String[50]	'Hallo Welt'	
demo_real	Real	500.5	
demo_bit	Bool	true	
demo_datetime	DTL	DTL#1970-01-01-4	
SQL_Data_Update	Struct		
SQL_Data_Select	Struct		

4 Funktionsweise

4.4 Konfigurations-Datenbaustein erstellen

Das Tool „ConfigDBCcreator“ generiert aus diesem Datenbaustein einen Datenbaustein mit den Konfigurationsdaten für die drei enthaltenen Datenstrukturen.

Abbildung 4-11: Konfigurations-Datenbaustein

ODK1500S_SQL_V2_ARGUMENTS_CFG		
Name	Data type	
Static		
sqlConfigData_SQL_Data_Insert	Array[0..1024] of *ODK1500S_SQL_V2_typeConfigData*	
sqlConfigData_SQL_Data_Update	Array[0..1024] of *ODK1500S_SQL_V2_typeConfigData*	
sqlConfigData_SQL_Data_Select	Array[0..1024] of *ODK1500S_SQL_V2_typeConfigData*	

Die Konfigurationsdaten enthalten die Datentyp-Identifizier sowie den Namen der Variablen:

Abbildung 4-12: Konfigurationsdaten für die Insert Struktur aus [Abbildung 4-10](#)

ODK1500S_SQL_V2_ARGUMENTS_CFG			
Name	Data type	Start value	
Static			
sqlConfigData_SQL_Data_Insert	Array[0..1024] of *ODK1500S_SQL_V2_ty...		
sqlConfigData_SQL_Data_Insert[0]	*ODK1500S_SQL_V2_typeConfigData*		
sqlDataType	Byte	BYTE#16#42	
sqlColumnName	String[25]	'id'	
sqlConfigData_SQL_Data_Insert[1]	*ODK1500S_SQL_V2_typeConfigData*		
sqlDataType	Byte	BYTE#16#42	
sqlColumnName	String[25]	'demo_integer'	
sqlConfigData_SQL_Data_Insert[2]	*ODK1500S_SQL_V2_typeConfigData*		
sqlDataType	Byte	BYTE#16#22	
sqlColumnName	String[25]	'demo_smallint'	
sqlConfigData_SQL_Data_Insert[3]	*ODK1500S_SQL_V2_typeConfigData*		
sqlDataType	Byte	BYTE#16#13	
sqlColumnName	String[25]	'demo_tinyint'	
sqlConfigData_SQL_Data_Insert[4]	*ODK1500S_SQL_V2_typeConfigData*		
sqlDataType	Byte	BYTE#16#85	
sqlColumnName	String[25]	'demo_double'	
sqlConfigData_SQL_Data_Insert[5]	*ODK1500S_SQL_V2_typeConfigData*		
sqlDataType	Byte	BYTE#16#83	
sqlColumnName	String[25]	'demo_bigint'	
sqlConfigData_SQL_Data_Insert[6]	*ODK1500S_SQL_V2_typeConfigData*		
sqlDataType	Byte	BYTE#16#50	
sqlColumnName	String[25]	'demo_char'	
sqlConfigData_SQL_Data_Insert[7]	*ODK1500S_SQL_V2_typeConfigData*		
sqlDataType	Byte	BYTE#16#51	
sqlColumnName	String[25]	'50'	
sqlConfigData_SQL_Data_Insert[8]	*ODK1500S_SQL_V2_typeConfigData*		
sqlDataType	Byte	BYTE#16#43	
sqlColumnName	String[25]	'demo_real'	
sqlConfigData_SQL_Data_Insert[9]	*ODK1500S_SQL_V2_typeConfigData*		
sqlDataType	Byte	BYTE#16#01	
sqlColumnName	String[25]	'demo_bit'	
sqlConfigData_SQL_Data_Insert[10]	*ODK1500S_SQL_V2_typeConfigData*		
sqlDataType	Byte	BYTE#16#C1	
sqlColumnName	String[25]	'demo_datetime'	
sqlConfigData_SQL_Data_Insert[11]	*ODK1500S_SQL_V2_typeConfigData*		
sqlDataType	Byte	BYTE#16#AA	
sqlColumnName	String[25]	'EndOfArgument'	

4.4.2 Konfigurations-Datenbaustein mit dem ConfigDBCcreator erstellen

Das Tool bietet zwei Wege aus den Anwenderdaten (sqlArguments) einen Konfigurationsdatenbaustein zu erstellen:

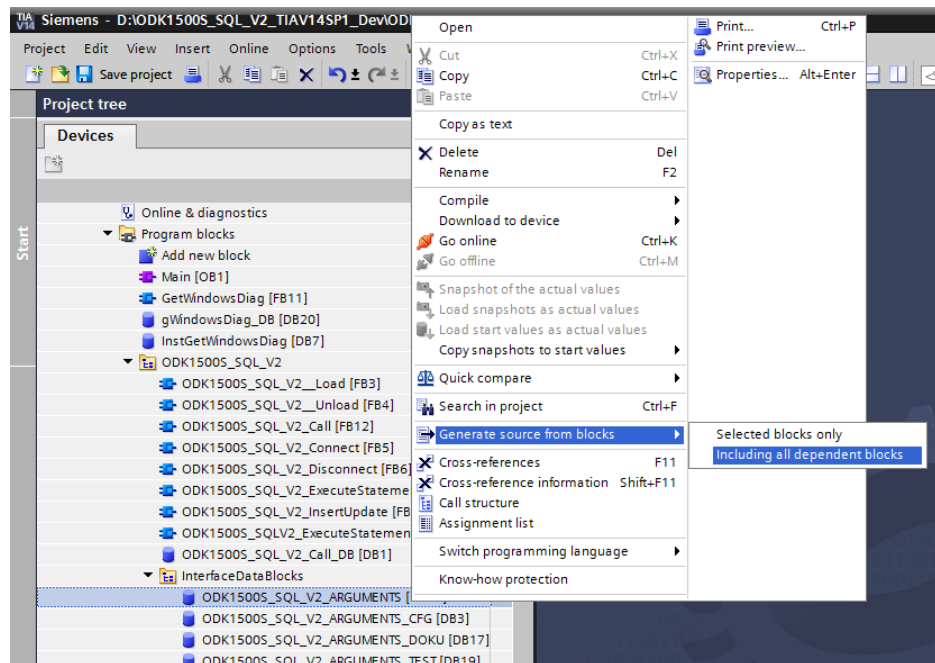
- Manueller Import und Export der Datenbausteine
- Direkter Zugriff mit dem ConfigDBCcreator auf das TIA-Portal Projekt

Manueller Import und Export der Datenbausteine

Bei dieser Variante müssen folgende Schritte ausgeführt werden:

- Datenbaustein mit den Daten, die in die Datenbank geschrieben werden sollen mit **allen Abhängigkeiten** als Quelle exportieren:

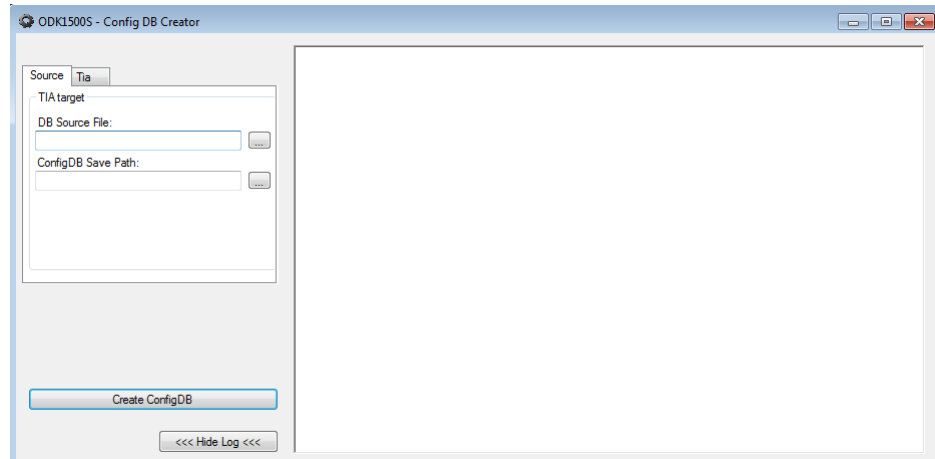
Abbildung 4-13: Datenbaustein exportieren



- Mit dem Tool ConfigDBCcreator.exe den Konfigurations-Datenbaustein generieren (exportierten Datenbaustein als DB Source File auswählen und Zielpfad angeben)

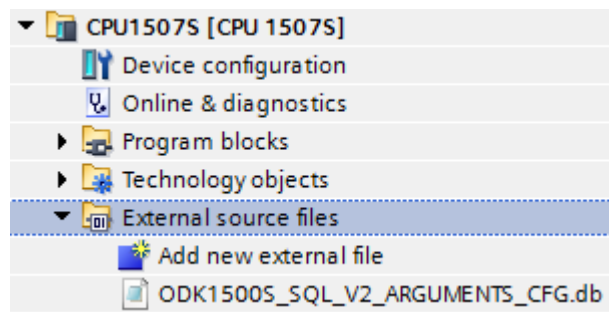
4.4 Konfigurations-Datenbaustein erstellen

Abbildung 4-14: GUI ConfigDBCcreator für manuellen Import und Export



- Generierten Konfigurations-Datenbaustein über externe Quellen im TIA-Portal einbinden und Datenbaustein generieren:

Abbildung 4-15: Konfigurations-Datenbaustein importieren und generieren



Nach dem der Konfigurations-Datenbaustein im TIA-Portal generiert wurde, liegt dieser im Ordner Program blocks. Der Baustein hat immer die Endung "CFG". Die Konfigurationsdaten können anschließend mit dem Eingangsparameter "sqlConfigData" vom Baustein ODK1500S_SQL_V3_Execute oder ODK1500S_SQL_V3_InsertUpdate verschaltet werden.

Direkter Zugriff mit dem ConfigDBCcreator auf das TIA-Portal Projekt

Das Tool kann über die TIA Portal Openness Schnittstelle direkt auf das TIA-Portal Projekt zugreifen und somit automatisch und komfortabel den Konfigurations-Datenbaustein generieren.

Voraussetzung hierfür ist, dass die kostenlose Software TIA Portal Openness auf dem Engineering Rechner installiert ist.

Sie können TIA Portal Openness über die Produkt-DVD von STEP 7 Professional V1x installieren:

Abbildung 4-16: Vorgehen Installation TIA Portal Openness

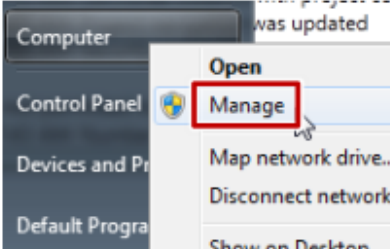
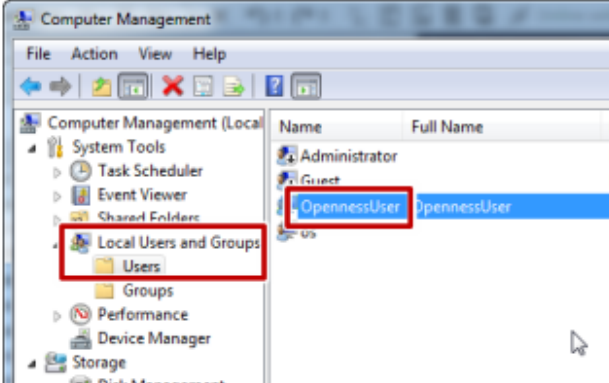
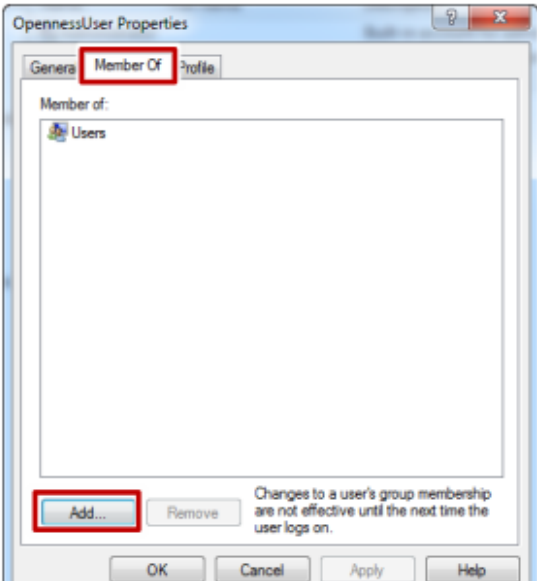
Nr.	Aktion
1.	Öffnen Sie die STEP 7 bzw. WinCC V14 SP1 Produkt-DVD. Starten Sie das Installations-Setup "Siemens_TIA_Openness_V14SP1.exe" im Ordner "Support".
2.	Folgen Sie den Anweisungen des Installations-Setup.

4 Funktionsweise

4.4 Konfigurations-Datenbaustein erstellen

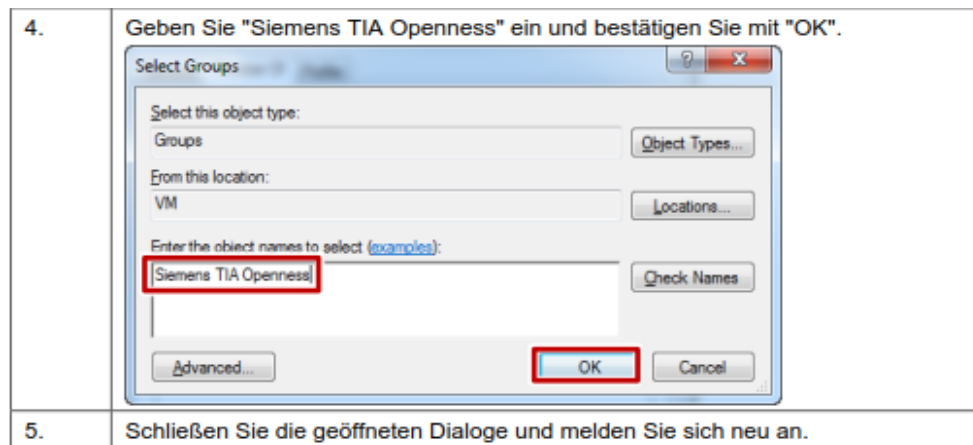
Um eine TIA Portal Openness Applikation zu nutzen muss der Benutzer einmalig zur Benutzergruppe "Siemens TIA Openness" hinzugefügt werden:

Abbildung 4-17: Benutzer verwalten

Nr.	Aktion
1.	<p>Klicken Sie in der Windows-Startleiste mit der rechten Maustaste auf "Computer" und wählen Sie "Verwalten" ("Manage").</p>  <p>The screenshot shows the Windows Start menu context menu for 'Computer'. The 'Manage' option is highlighted with a red box. Other options include 'Open', 'Map network drive..', 'Disconnect network', and 'Show on Desktop'.</p>
2.	<p>Öffnen Sie "Lokale Benutzer und Gruppen > Benutzer" und doppelklicken Sie auf den Benutzer.</p>  <p>The screenshot shows the Computer Management console. The 'Local Users and Groups > Users' folder is expanded and highlighted with a red box. The 'OpennessUser' entry is selected and highlighted in blue.</p>
3.	<p>Wechseln Sie in die Registerkarte "Mitglied von" ("Member Of") und klicken Sie die Schaltfläche "Hinzufügen" ("Add").</p>  <p>The screenshot shows the 'OpennessUser Properties' dialog box. The 'Member Of' tab is selected and highlighted with a red box. The 'Add...' button is also highlighted with a red box. The 'Member of:' list contains 'Users'. A note at the bottom states: 'Changes to a user's group membership are not effective until the next time the user logs on.'</p>

4 Funktionsweise

4.4 Konfigurations-Datenbaustein erstellen



Nach erfolgreicher Installation und Einrichtung von TIA Openness können Sie über das Tool ConfigDBCcreator.exe unter dem Reiter "Tia" über den Button "Select project/block" einen Datenbaustein auswählen aus dem die Konfigurationsdaten erstellt werden sollen. Nach anschließendem Klick auf den Button "Create ConfigDB" wird automatisch ein Datenbaustein mit den Konfigurationsdaten im geöffneten Projekt angelegt:

Abbildung 4-18: GUI ConfigDBCcreator mit direktem Zugriff auf das TIA-Projekt

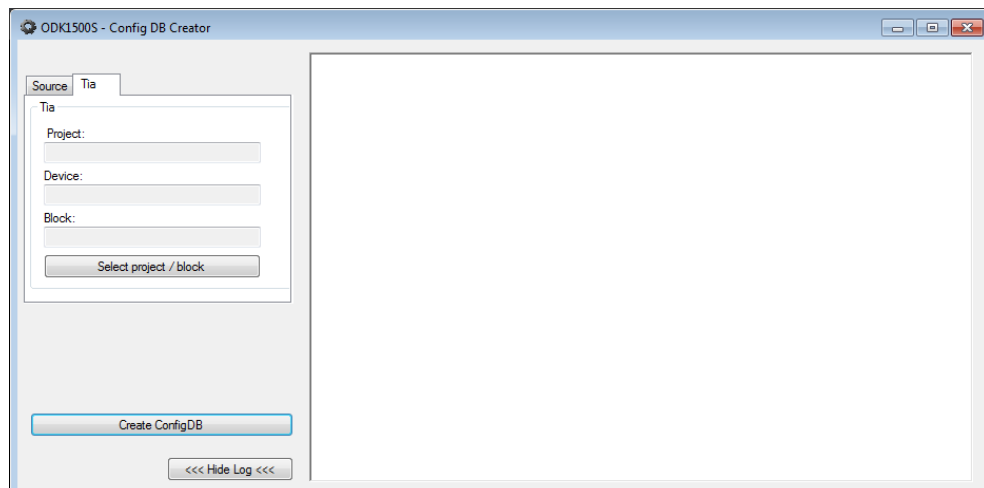
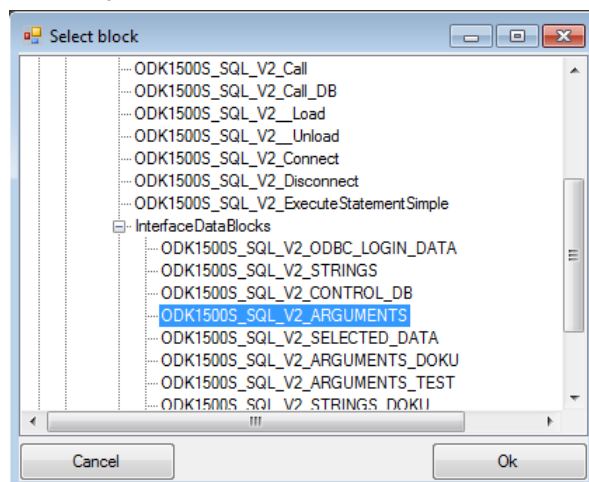


Abbildung 4-19: Auswahl Datenbaustein



4.4 Konfigurations-Datenbaustein erstellen

Der Konfigurations-Datenbaustein befindet sich im Ordner Program blocks. Der Baustein hat immer die Endung "CFG". Die Konfigurationsdaten können anschließend mit dem Eingangsparameter "sqlConfigData" vom Baustein ODK1500S_SQL_V3_Execute oder ODK1500S_SQL_V3_InsertUpdate verschaltet werden.

4.4.3 Einschränkungen ODK1500S ConfigDBCcreator

Das Tool unterstützt optimierte sowie nicht optimierte Datenbausteine. Innerhalb des Datenbausteins können alle vom SQL-Treiber unterstützten STEP7 Datentypen, Strukturen, PLC-Datentypen sowie Arrays von elementaren Datentypen genutzt werden.

Voraussetzung ist, dass die Daten zum Schreiben in die Datenbank jeweils in einer Struktur zusammengefasst werden.

Folgende Konstellationen werden nicht unterstützt, da diese in der Regel von Datenbanken nicht unterstützt werden:

- Array of PLC-Datentyp
- Verschachtelte Strukturen
- Globale Konstanten (wird bei manuellem Import/Export nicht unterstützt)
- Variablennamen mit mehr als 25 Zeichen (werden abgeschnitten)

5 Installation und Inbetriebnahme

Dieses Kapitel beschreibt die Installation und Inbetriebnahme des ODK 1500S-SQL-Treibers sowie die optionale Installation der zum S7-Beispielprojekt gehörenden MSSQL-Datenbank.

5.1 Installation und Inbetriebnahme der MSSQL Demo-Datenbank

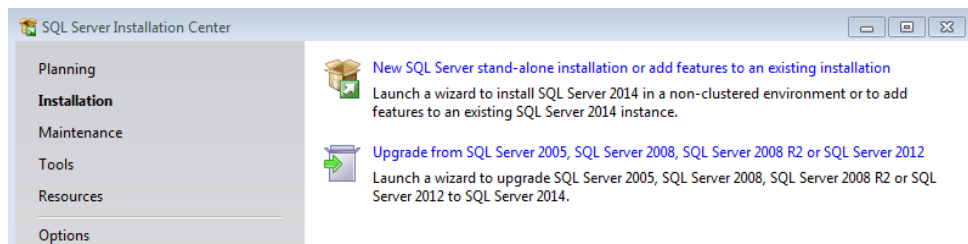
Die folgende Installationsanleitung dient zur Hilfe bei der Einrichtung der mit dem ODK 1500S-SQL-Treiber gelieferten Demodatenbank sowie zur Installation eines MS-SQL-Express-Servers.

5.1.1 Installation MSSQL Server 2014 Express and Tools

Auf dem Datenbankserver wird das Datenbanksystem "Microsoft SQL 2014 Server Express and Tools" installiert. Um die Datenbank zu installieren, wird die Datei "SQLEXPRT_x64_ENU" ausgeführt. Die Umgebungssprachen von dem Betriebssystem und von dem Datenbanksystem müssen übereinstimmen.

Unter dem Punkt Installation wird eine "New SQL Server stand-alone installation" ausgewählt:

Abbildung 5-1: Installationscenter



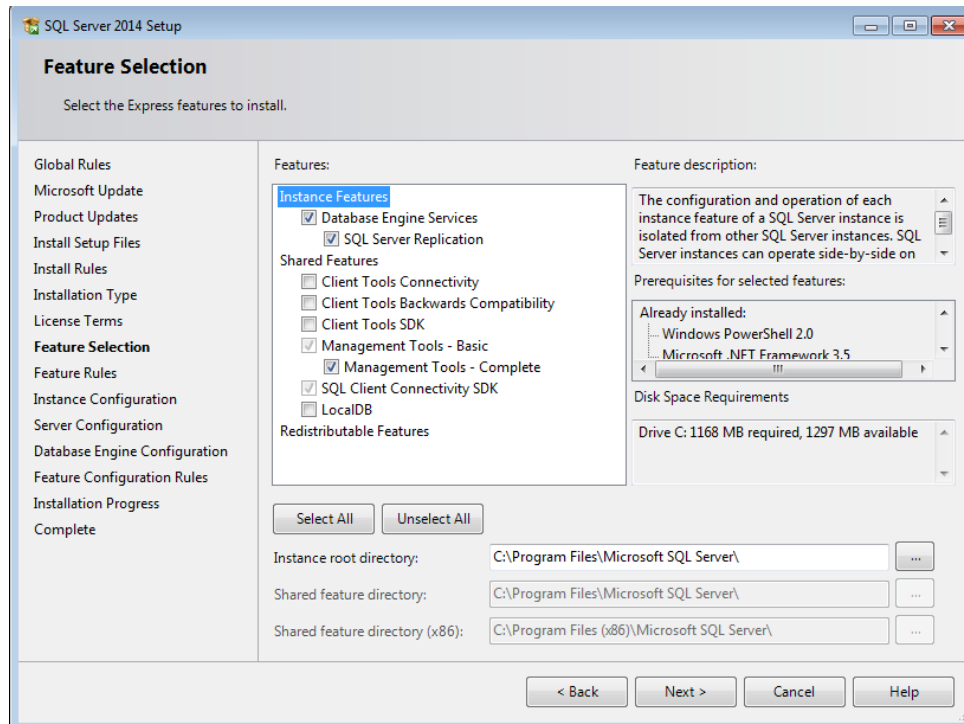
Die nächsten Schritte sind selbsterklärend. Hierbei müssen die Lizenzbedingungen akzeptiert werden und bis zum Punkt "Feature Selection" auf Weiter geklickt werden.

Wie in der folgenden Abbildung dargestellt, müssen mindestens folgende Komponenten ausgewählt werden:

5 Installation und Inbetriebnahme

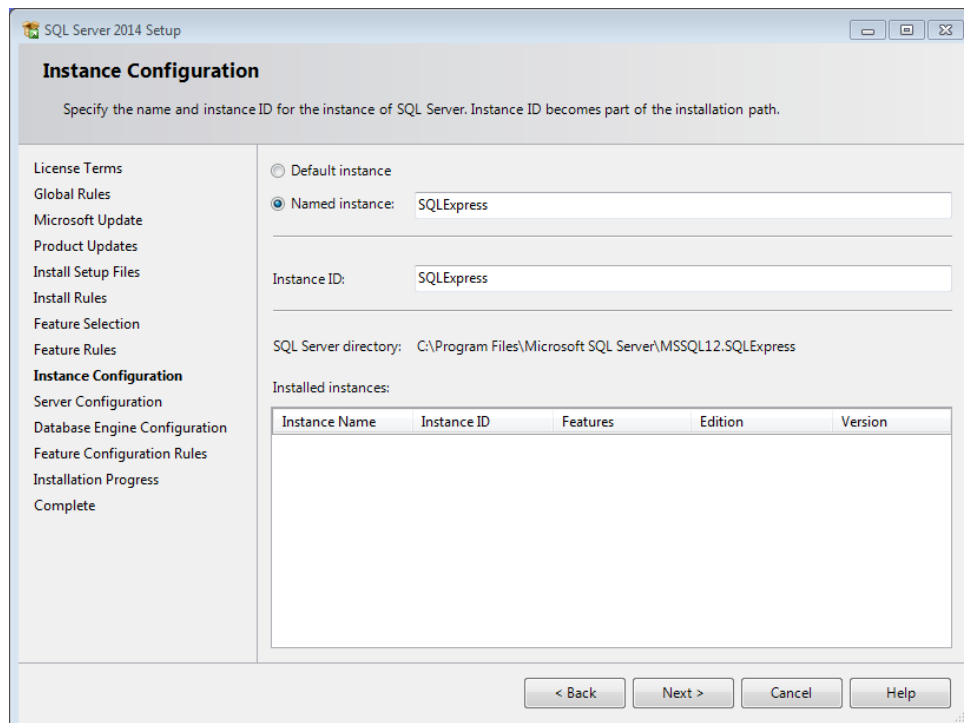
5.1 Installation und Inbetriebnahme der MSSQL Demo-Datenbank

Abbildung 5-2: Feature Selection



Der Instanzname wird für das Beispielprojekt beibehalten:

Abbildung 5-3: Instanzname

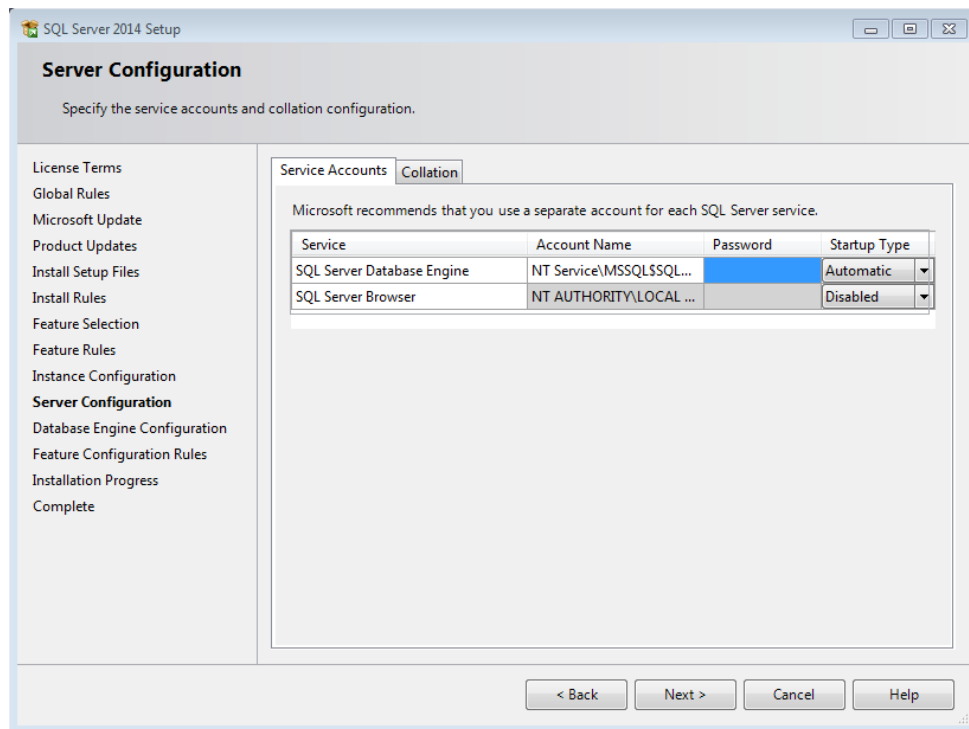


Bei der Serverkonfiguration wird der Netzwerkdienst ausgewählt. Für das Beispielprojekt reicht der SQL Server Dienst:

5 Installation und Inbetriebnahme

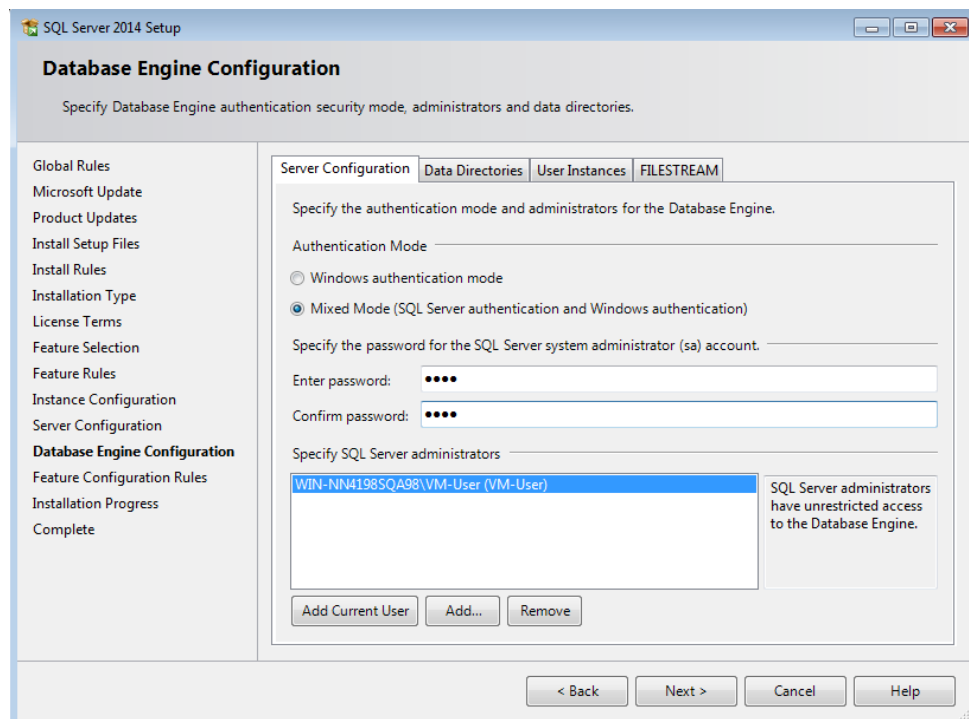
5.1 Installation und Inbetriebnahme der MSSQL Demo-Datenbank

Abbildung 5-4: Server Configuration



Damit eine Windows sowie eine Server Anmeldung möglich ist, muss bei der Datenmodulkonfiguration in der folgenden Abbildung der gemischte Dienst ausgewählt werden. Außerdem muss ein Serverpasswort sowie eine Server-Administration vergeben werden. Die Einstellungen für Fehler – und Verwaltungsberichte, die im nächsten Fenster zu Verfügung stehen werden für dieses Beispiel nicht verwendet. Nach der Zusammenfassung wird der SQL Server über die Taste "Install" installiert.

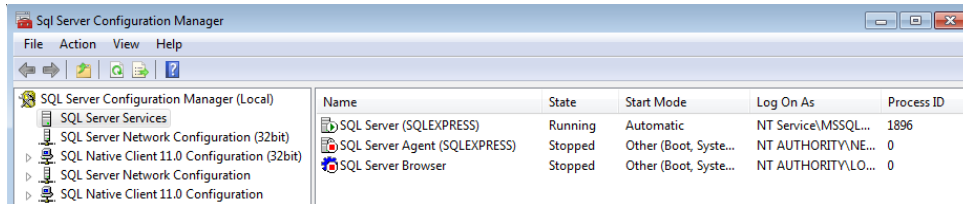
Abbildung 5-5: Database Engine Configuration



5.1.2 Konfiguration des SQL-Servers

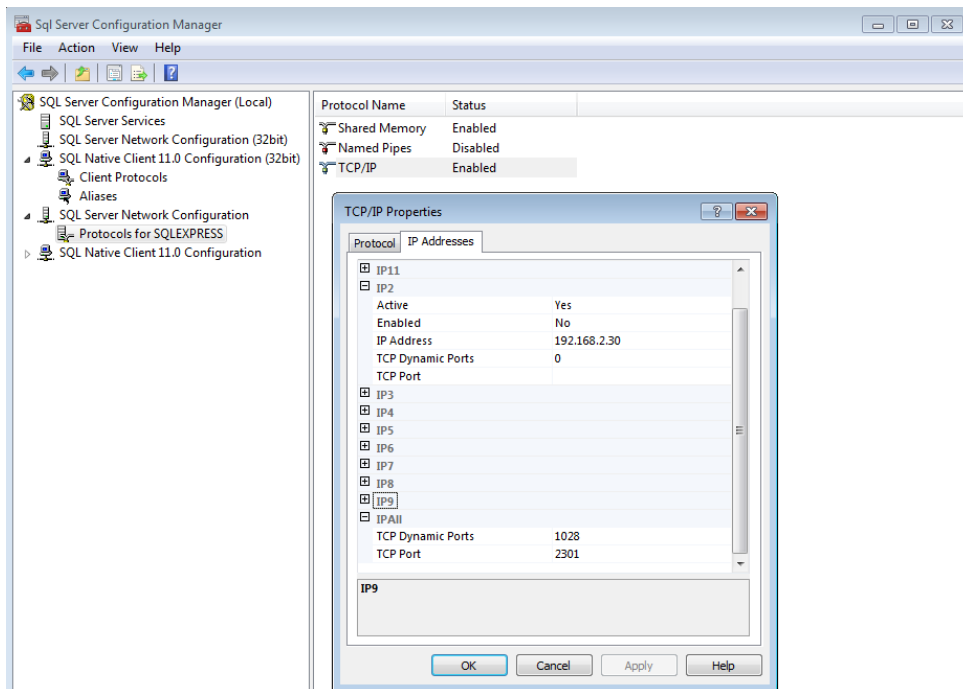
Nach dem der SQL Server installiert wurde, muss als nächstes die Kommunikation zu dem Runtime Rechner (S7-Software Controller) eingerichtet werden. Dazu wird zunächst der SQL Server-Konfigurations-Manager geöffnet. Dieser wird über *Start* → *Programme* → *Microsoft SQL Server 2014* → *Konfigurationstools* → *SQL Server Konfiguration* erreicht. Anschließend kann unter dem Menüpunkt *SQL Server-Dienste* eingesehen werden, welche Dienste ausgeführt werden. Hierbei ist darauf zu achten, dass der SQL Server Dienst aktiv ist:

Abbildung 5-6: SQL Server Dienste



Anschließend wird die TCP/IP Verbindung aktiviert und konfiguriert. Über *SQL Server Network Configuration* → *Protocols for SQLEXPRESS* → *TCP/IP* → rechte Maustaste → *Eigenschaften* kann dieser Schritt durchgeführt werden. Unter dem Reiter *Protokoll* kann die TCP/IP Kommunikation aktiviert werden. Unter *IP-Adressen* muss sowohl eine feste IP-Adresse sowie ein Port (z.B. 2301) eingetragen werden. Anschließend muss der Dienst neugestartet werden.

Abbildung 5-7: TCP/IP Einstellungen



Hinweis

Die Einstellungen werden erst wirksam, wenn der entsprechende Dienst neugestartet wurde.

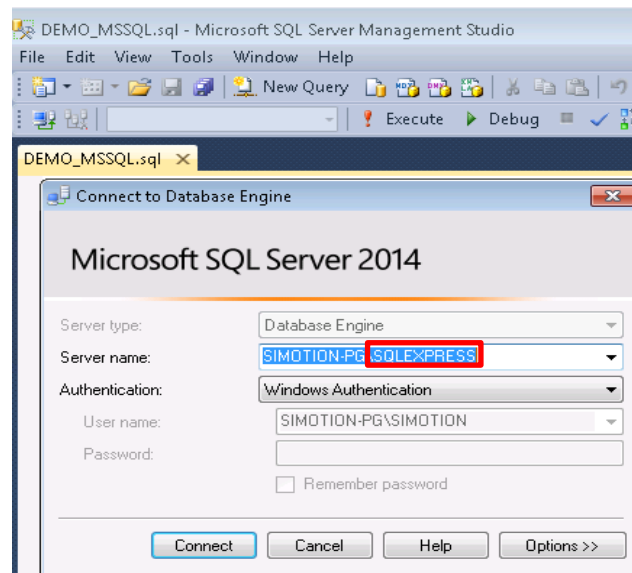
5.1.3 Erstellen der Demodatenbank mithilfe eines Skripts

Mithilfe des mitgelieferten Skripts "DEMO_MSSQL_V3.sql" kann eine Demo-Datenbank mit Tabelle, ein Benutzer Login mit Passwort und ein Benutzer für die Demo-Datenbank erzeugt werden. Das Skript kann sowohl von einem lokalen Verzeichnis als auch von einem USB-Stick ausgeführt werden.

Auf dem Datenbankserver, wo bereits SQL-Server Management Studio installiert ist, kann das Skript durch einen Doppelklick ausgeführt werden. Dafür ist eine Anmeldung als Administrator erforderlich. Danach wird das SQL Server Management Studio automatisch gestartet und eine Verbindung mit dem Server "SQLEXPRESS" mittels Windows Authentifizierung hergestellt (Abbildung 5-8).

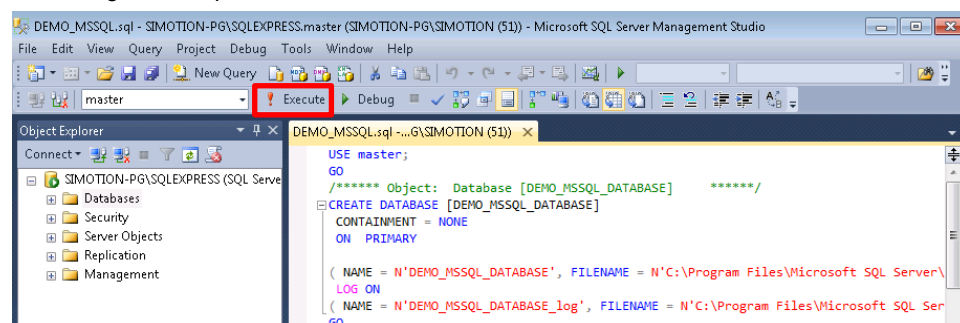
Hinweis Bitte beachten Sie, dass Sie eine Verbindung mit dem Server "SQLEXPRESS" herstellen.

Abbildung 5-8: Anmeldung mit Windows Authentifizierung



Nach der erfolgreichen Verbindung mit "SQLEXPRESS" ist das Skript in einem Reiter auf der rechten Seite zu sehen. Mit dem Befehl "Execute" wird das Skript ausgeführt.

Abbildung 5-9: Skript ausführen



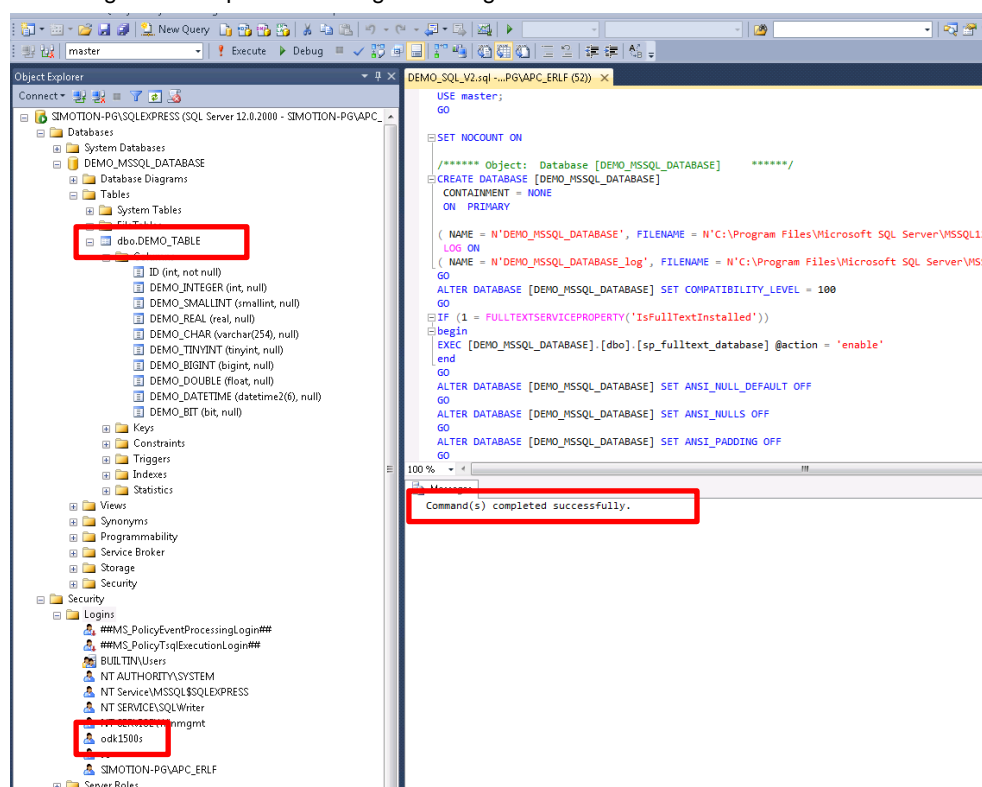
5 Installation und Inbetriebnahme

5.2 Installation und Inbetriebnahme des SQL-Treibers auf dem Runtime Rechner

Nach dem erfolgreichen Ausführen des Skriptes wurden folgenden Komponenten erzeugt:

- DEMO_MSSQL_DATABASE: Demo Datenbank mit Demo Tabelle
- Benutzer-Login: odk1500s mit Passwort 1234. Die Verbindung zum SQL-Server kann über die Windows-Authentifizierung oder über eine SQL-Server-Authentifizierung erfolgen. Für eine Remote-Verbindung über TCP/IP wird die SQL-Server-Authentifizierung benötigt. Dafür wird eine Benutzeranmeldung benötigt. Nach dem die neue Anmeldung erstellt wurde, kann beim nächsten Verbindungsaufbau zwischen der Windows- und SQL-Server-Authentifizierung ausgewählt werden.
- Benutzer odk1500s für die Demo-Datenbank

Abbildung 5-10: Skript wurde erfolgreich ausgeführt



5.2 Installation und Inbetriebnahme des SQL-Treibers auf dem Runtime Rechner

Im Folgenden wird die Installation des ODK 1500S-SQL-Treibers auf dem Runtime Rechner (S7-1500 Software Controller) erläutert.

5.2.1 Installation der SQL-Treiber-DLL

Die Installation des SQL-Treibers für die SQL-Datenbankverbindung beschränkt sich auf das Ausführen mit Administratorrechten der Install_Odk1500S-SQL.bat Datei. Die DLL muss sich bei der Installation im selben Verzeichnis wie die SQL_install.bat Datei befinden.

ACHTUNG Die Datei Install_Odk1500S-SQL.bat muss unter Windows als Administrator ausgeführt werden. Andernfalls werden die benötigten Dateien nicht in den Zielordner kopiert.

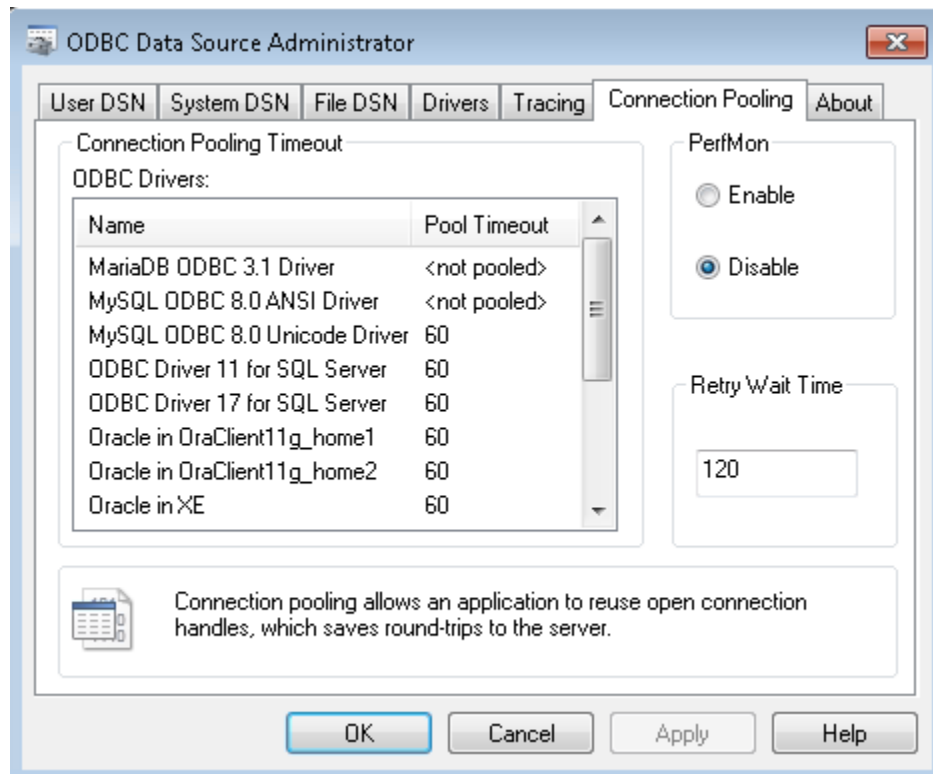
5.2.2 Parametrierung des ODBC-Treibers

Für die Verbindung zur SQL-Datenbank wird ein 64-Bit ODBC-Treiber benötigt. Dieser muss i.d.R. zuvor auf dem Runtime-PC installiert werden. Es wird empfohlen möglichst aktuelle ODBC-Treiber zu verwenden.

Eine Parametrierung des ODBC-Treibers ist ab Version 3.0 vom ODK1500S SQL-Treiber nicht mehr notwendig, da hier die Verbindung zur Datenbank über die Verbindungsinformationen im Connection String erfolgt.

Für eine optimale Performance wird empfohlen das Connection Pooling zu aktivieren. Dies kann über den Reiter „Connection Pooling“ im „ODBC Data Source Administrator“ für den jeweiligen ODBC Treiber aktiviert werden:

Abbildung 5-11: ODBC Treiber Connection Pooling



Die (64 bit) "ODBC Data Source Administrator" erreicht man, indem man z.B. unter Windows nach „ODBC“ sucht.

6 Bedienung der Applikation

6.1 Verwenden der Beispielapplikation

Mit dem ODK1500S-SQL-Treiber wird eine Beispielapplikation geliefert, die zeigt wie die SQL-Treiber FBs eingesetzt werden können, um Daten in die Demodatenbank zu schreiben, zu ändern oder Daten aus der Datenbank auszulesen.

Die Beispielapplikation besteht aus den folgenden Komponenten:

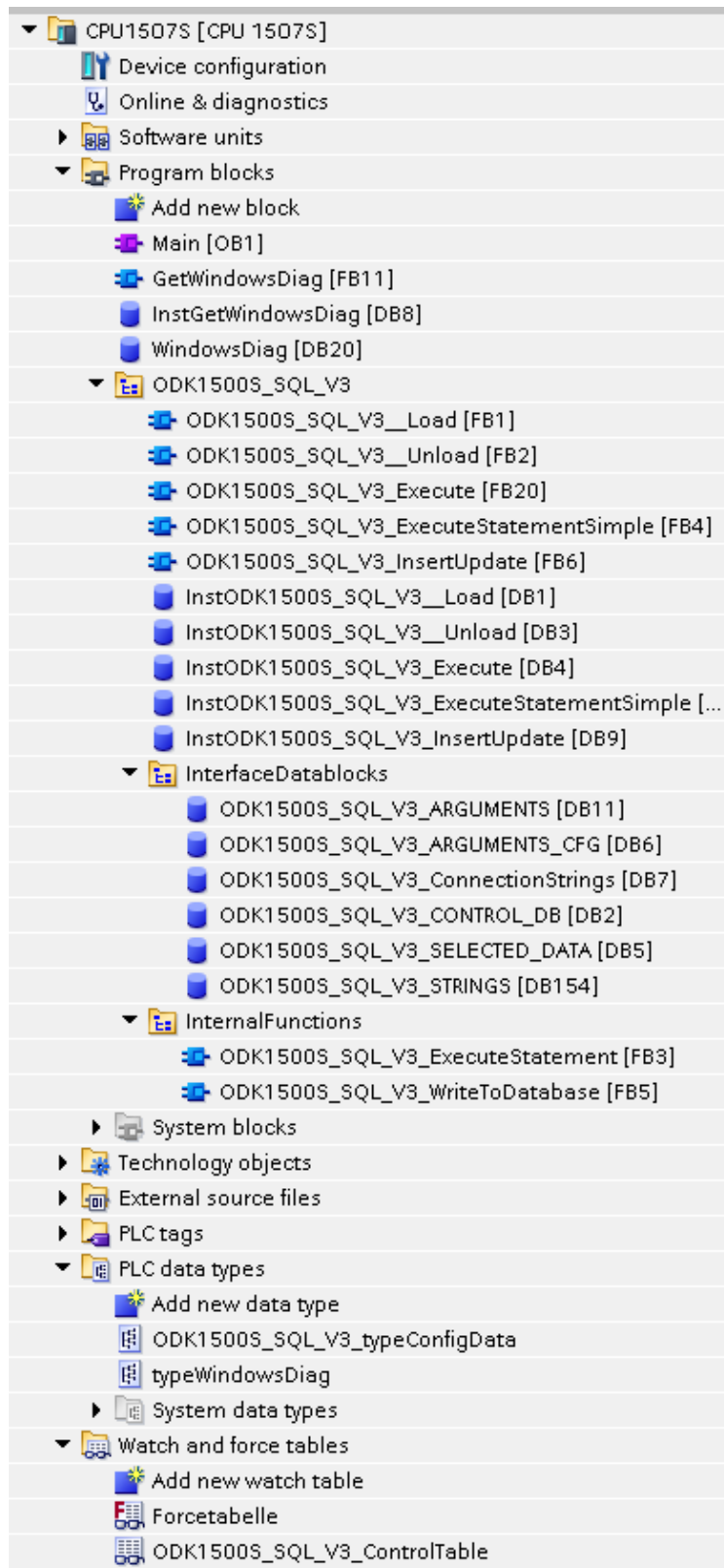
- STEP 7 V15 SP1 Professional Beispielprojekt
- MSSQL Demodatenbank "DEMO_MSSQL_DATABASE"

Installieren und konfigurieren Sie zunächst die Demodatenbank und den SQL-Treiber wie im Kapitel [Installation und Inbetriebnahme](#) beschrieben.

Deaktivieren Sie das mitgelieferte STEP 7 Beispielprojekt und passen Sie ggf. die Hardwarekonfiguration an Ihre verwendeten Komponenten und Einstellungen an. Anschließend laden Sie das Projekt in Ihre Steuerung.

6.1.1 Aufbau des STEP 7 Beispielprojektes

Abbildung 6-1: Aufbau Beispielprojekt



6.1 Verwenden der Beispielapplikation

Das Beispielprojekt besteht aus den ODK1500S-SQL-Funktionsbausteinen, PLC-Datentypen sowie Interface-Datenbausteinen in denen u.a. die SQL-Strings und SQL Argumente abgelegt werden können. Im Folgenden werden die einzelnen Komponenten erläutern:

Organisationsbausteine

- Main (OB1): Aufruf der SQL-Treiber FBs.

Funktionsbausteine

- ODK1500S_SQL_V3_Load
- ODK1500S_SQL_V3_Execute
- ODK1500S_SQL_V3_ExecuteStatementSimpel
- ODK1500S_SQL_V3_InsertUpdate
- ODK1500S_SQL_V3_Unload

Interface Datenbausteine

- ODK1500S_SQL_V3_ConnectionsStrings: Hier werden die Verbindungsinformationen für den Verbindungsaufbau zur Datenbank hinterlegt
- ODK1500S_SQL_V3_ARGUMENTS: Hier werden die SQL-Argumente angegeben (Daten die in die Datenbank geschrieben werden sollen)
- ODK1500S_SQL_V3_ARGUMENTS_CFG: Enthält die Konfigurationsdaten für die SQL-Argumente (mit dem Tool ConfigDBCcreator aus dem DB ODK1500S_SQL_V3_ARGUMENTS erstellt)
- ODK1500S_SQL_V3_STRINGS: Hier werden die SQL-Strings für die SQL-Anweisungen angegeben
- ODK1500S_SQL_V3_SELECTED_DATA: Hier werden die von z.B. einer SQL Select-Anweisung zurückgegebenen Daten der Datenbank abgelegt
- ODK1500S_SQL_V3_CONROL_DB: zentraler Steuer- und Überwachungs-Datenbaustein der Beispielapplikation

PLC-Datentypen

- ODK1500S_SQL_V3_typeConfigData: Struktur für die Konfigurationsdaten, die Anhand der SQL-Argumente mit dem Tool ConfigDBCcreator erstellt werden
- typeWindowsDiag: Optional: Wird im DB WindowsDiag verwendet, um Diagnoseinformationen zum Windows Betriebssystem zu liefern

Beobachtungstabellen

- ODK1500S_SQL_V3_ControlTable: Beobachtungstabelle zum Steuern der Beispielapplikation (enthält Variablen aus dem DB ODK1500S_SQL_V3_CONTROL_DB)

Abbildung 6-2: Beobachtungstabelle "ODK1500S_SQL_V3_ControlTable"

i	Name	...	Display format
// Load ODK1500S SQL driver V3 dll			
	"ODK1500S_SQL_V3_CONTROL_DB".ODK1500S_SQL_Load.REQ		Bool
// Select one SQL Statement to be executed (some sql driver blocks use same instance)			
	"ODK1500S_SQL_V3_CONTROL_DB".SQL_Select		Bool
	"ODK1500S_SQL_V3_CONTROL_DB".SQL_Insert		Bool
	"ODK1500S_SQL_V3_CONTROL_DB".SQL_Update		Bool
// Execute an insert or select statement using ODK1500S_SQL_V3_Execute FB			
	"ODK1500S_SQL_V3_CONTROL_DB".ODK1500S_SQL_Execute.REQ		Bool
	"ODK1500S_SQL_V3_CONTROL_DB".ODK1500S_SQL_Execute.affectedRows		DEC+/-
	"ODK1500S_SQL_V3_CONTROL_DB".ODK1500S_SQL_Execute.fctError		Bool
	"ODK1500S_SQL_V3_CONTROL_DB".ODK1500S_SQL_Execute.fctErrorCode		Hex
	"ODK1500S_SQL_V3_CONTROL_DB".ODK1500S_SQL_Execute.sqlErrorInfo		Unicode string
	"InstODK1500S_SQL_V3_Execute".statRuntime_ms		Floating-point ...
// Execute a delete statement using ODK1500S_SQL_V3_ExecuteStatementSimple FB			
	"ODK1500S_SQL_V3_CONTROL_DB".ODK1500S_SQL_ExecuteSimple.REQ		Bool
	"ODK1500S_SQL_V3_CONTROL_DB".ODK1500S_SQL_ExecuteSimple.affectedRows		DEC+/-
	"ODK1500S_SQL_V3_CONTROL_DB".ODK1500S_SQL_ExecuteSimple.fctError		Bool
	"ODK1500S_SQL_V3_CONTROL_DB".ODK1500S_SQL_ExecuteSimple.fctErrorCode		Hex
	"ODK1500S_SQL_V3_CONTROL_DB".ODK1500S_SQL_ExecuteSimple.sqlErrorInfo		Unicode string
// Update or insert data using ODK1500S_SQL_V3_InsertUpdate FB			
	"ODK1500S_SQL_V3_CONTROL_DB".ODK1500S_SQL_InsertUpdate.REQ		Bool
	"ODK1500S_SQL_V3_CONTROL_DB".ODK1500S_SQL_InsertUpdate.affectedRows		DEC+/-
	"ODK1500S_SQL_V3_CONTROL_DB".ODK1500S_SQL_ExecuteSimple.fctError		Bool
	"ODK1500S_SQL_V3_CONTROL_DB".ODK1500S_SQL_ExecuteSimple.fctErrorCode		Hex
	"ODK1500S_SQL_V3_CONTROL_DB".ODK1500S_SQL_ExecuteSimple.sqlErrorInfo		Unicode string
	"InstODK1500S_SQL_V3_InsertUpdate".statRuntime_ms		Floating-point ...
// Unload ODK1500S SQL driver V3 dll			
	"ODK1500S_SQL_V3_CONTROL_DB".ODK1500S_SQL_Unload.REQ		Bool

6.1.2 Laden der SQL-Treiber-DLL

Bevor der ODK1500S-SQL-Treiber genutzt werden kann muss die SQL-Treiber-DLL vom Software Controller geladen werden.

Die SQL-Treiber-DLL wird im Beispielprojekt automatisch geladen, sobald eine Verbindung zwischen dem Software Controller und dem Windows OS besteht. Dazu wird im OB1 der Baustein "GetWindowsDiag" aufgerufen. Dieser liefert den Zustand des Windows Betriebssystems über eine InOut-Struktur. Innerhalb dieser Struktur ist das Bit "WindowsIsRunning" hinterlegt. Dieses Bit ist mit dem "REQ"-Eingang des ODK1500S_SQL_V3_Load Bausteins verschaltet.

Nur wenn die SQL-Treiber-DLL gegen eine neue Version ausgetauscht werden soll ist es ggf. erforderlich die geladene DLL zu entladen, die neue Version der DLL zu installieren und anschließend die neue Version manuell zu laden.

Für das manuelle Laden der DLL gehen Sie bitte wie folgt vor:

- Öffnen Sie die Beobachtungstabelle ODK1500S_V3_ControlTable
- Steuern Sie das Bit "ODK1500S_SQL_V3_CONTROL_DB.ODK1500S_SQL_V3_Load.REQ" auf 1

Es wird der Baustein ODK1500S_SQL_V3_Load im OB1 aufgerufen.

6.1.3 Ausführen von SQL-Anweisungen

Mit dem Baustein ODK1500S_SQL_V3_Execute können beliebige SQL-Anweisungen mit flexiblen Argumenten ausgeführt werden.

Das grundsätzliche Vorgehen beim Ausführen von SQL-Anweisungen in der Beispielapplikation besteht aus zwei Schritten:

1. Gewünschte SQL-Anweisung anwählen
2. Request-Bit der Funktion anwählen

Dazu öffnen Sie zunächst die Beobachtungstabelle ODK1500S_SQL_V3_ControlTable:

In der Beispielapplikation werden die in den Datenbausteinen ODK1500S_SQL_V3_STRINGS und ODK1500S_SQL_V3_ARGUMENTS definierten SQL-Anweisungen verwendet.

Hinweis Für die korrekte Funktion der Beispielapplikation ist darauf zu achten, dass immer nur eine SQL-Anweisung (Select, Insert, Update) angewählt ist! Hintergrund ist, dass in der Beispielapplikation bei einigen SQL-Treiber FBs mehrfach die gleiche Instanz verwendet wird.

Ausführen einer INSERT oder SELECT Anweisung

- Setzen Sie ggf. die Anwahl einer anderen SQL-Anweisung zurück
- Steuern Sie je nach gewünschter Beispiel-SQL-Anweisung das Bit "ODK1500S_SQL_V3_CONTROL_DB.SQL_Insert" oder "ODK1500S_SQL_V3_CONTROL_DB.SQL_Select" auf 1
- Steuern Sie das Bit "ODK1500S_SQL_V3_CONTROL_DB.ODK1500S_SQL_V3_Execute.REQ" auf 1

Es wird der Baustein ODK1500S_SQL_V3_SetStatement im OB1 aufgerufen. Nach erfolgreicher Ausführung wird das Bit "ODK1500S_SQL_V3_CONTROL_DB.ODK1500S_SQL_V3_Execute.REQ" wieder zurückgesetzt.

6.1.4 Prozessdaten schreiben

Mit dem Baustein ODK1500S_SQL_V3_InsertUpdate können Daten aus einem Datenbaustein ohne SQL-Strings (SQL-Anweisungen) in die Datenbank geschrieben werden.

- Setzen Sie ggf. die Anwahl einer anderen SQL-Anweisung zurück
- Steuern Sie je nach gewünschter Funktion das Bit "ODK1500S_SQL_V3_CONTROL_DB.SQL_Insert" (fügt einen neuen Datensatz in die Datenbank ein) oder "ODK1500S_SQL_V3_CONTROL_DB.SQL_Update" auf 1 (aktualisiert/überschreibt den Datensatz mit der ID 500 in der Datenbank)
- Steuern Sie das Bit "ODK1500S_SQL_V3_CONTROL_DB.ODK1500S_SQL_V3_InsertUpdate.REQ" auf 1

6.2 Anpassen der Beispielapplikation an eigene Anforderungen

Es wird der Baustein ODK1500S_SQL_V3_InsertUpdate im OB1 aufgerufen. Nach erfolgreicher Ausführung wird das Bit "ODK1500S_SQL_V3_CONTROL_DB.ODK1500S_SQL_V3_InsertUpdate.REQ" wieder zurückgesetzt.

6.1.5 SQL-Treiber-DLL entladen

Ein manuelles entladen ist nur zum Tausch der DLL notwendig. Dazu gehen Sie wie folgt vor:

- Öffnen Sie die Beobachtungstabelle ODK1500S_SQL_V3_ControlTable
- Steuern Sie das Bit "ODK1500S_SQL_V3_CONTROL_DB.ODK1500S_SQL_V3_Unload.REQ" auf 1

Es wird der Baustein ODK1500S_SQL_V3_Unload im OB1 aufgerufen.

Nachdem Sie die SQL-Treiber dll gegen eine neue Version ausgetauscht haben, können Sie das Bit ODK1500S_SQL_V3_CONTROL_DB.ODK1500S_SQL_V3_Unload.REQ wieder zurücksetzen. Dabei wird in der Demoapplikation automatisch die neue SQL-Treiber dll geladen.

6.1.6 Fehleranalyse

Bei nicht erfolgreicher Ausführung eines SQL-Funktionsbausteins können die von den Bausteinen zurückgegebenen Fehlermeldungen online im Datenbaustein SQL_CONTROL_DB oder in der Beobachtungstabelle eingesehen werden.

Nähere Informationen zu den Fehler-Codes finden Sie im Kapitel 7: [Fehlermeldungen](#).

6.2 Anpassen der Beispielapplikation an eigene Anforderungen

Es wird empfohlen zunächst das STEP 7-Beispielprojekt zu verwenden und anschließend dieses an eigene Anforderungen anzupassen.

Hinweis Erfahrene STEP 7 Anwender können auch alternativ die mitgelieferte SCL-Quelle verwenden, um die SQL-Treiber-Bausteine in einem bestehenden TIA-Portal zu importieren und zu generieren.

6.2.1 Anpassen der Verbindungsinformationen

Passen Sie hierzu im Datenbaustein ODK1500S_SQL_V3_ConnectionStrings die Verbindungsinformationen an, die für die Verbindung zu Ihrer Datenbank erforderlich sind. Orientieren Sie sich an der Syntax der Beispiele im Datenbaustein ODK1500S_SQL_V3_ConnectionStrings.

- Driver: Name des zu verwendenden ODBC Treibers (z.B. „ODBC Driver 17 for SQL Server“)
- Server: IP-Adresse und Name des Servers (z.B. „192.168.0.211\SQLEXPRESS“)
- Database: Name der Datenbank (z.B. „DEMO_MSSQL_DATABASE“)
- Uid: Benutzername für den Login
- Pwd: Passwort für den Login

6.2.2 Anpassen der SQL-Anweisungen

SQL-Anweisungen bestehen wie zuvor beschrieben aus zwei Teilen:

- SQL-String: enthält die eigentliche SQL-Anweisung mit Platzhaltern („?“) für die SQL-Argumente
 - SQL-Argument: hier werden die eigentlichen Daten definiert
1. Tragen Sie im Datenbaustein ODK1500S_SQL_V3_STRINGS Ihre SQL-Anweisungen mit den Platzhaltern („?“) für die Argumente ein. (Die maximale SQL-Stringlänge beträgt ca. 16000 Zeichen)
 2. Im Datenbaustein ODK1500S_SQL_V3_ARGUMENTS definieren Sie eine Struktur mit den Daten, in der gleichen Reihenfolge wie die Platzhalter im SQL-String.
 3. Generieren Sie mit dem Tool ConfigDBCcreator die Konfigurationsdaten für den Baustein ODK1500S_SQL_V3_ARGUMENTS.

Die drei Datenbausteine bzw. die Strukturen werden an den Baustein ODK1500S_SQL_V3_Execute übergeben.

Wenn Sie SQL-Select Anweisungen nutzen passen Sie noch die Reihenfolge und die Struktur der Daten im Datenbaustein ODK1500S_SQL_V3_SELECTED_DATA an.

7 Fehlermeldungen

7.1 Status und Error Codes ODK1500S

Der ODK1500S-SQL-Treiber wurde mit dem ODK1500S (Open Development Kit) entwickelt. Das ODK kann ebenfalls Status- und Fehlercodes generieren, die an den Funktionsbausteinen am Ausgang "STATUS" zurückgegeben werden.

Tabelle 7-1: ODK Status und Error Codes

Status Codes (HEX)	Beschreibung
0x0000	Success
0x7000	<p>(Un)Load the ODK driver No active (un)loading</p> <p>Execute the ODK driver No active process</p>
0x7001	<p>(Un)Load the ODK driver (Un)Loading in progress, first call</p> <p>Execute the ODK driver First call (asynchronous)</p>
0x7002	<p>(Un)Load the ODK driver (Un)Loading in progress, ongoing call</p> <p>Execute the ODK driver Continuous call (asynchronous)</p>
0x7100	<p>Load the ODK driver CPU function library is already loaded</p>
Error Codes (HEX)	Beschreibung
0x8090	<p>Load the ODK driver CPU function library could not be loaded. An exception occurred during execution of the "OnLoad()" function.</p> <p>Execute the ODK driver CPU function library could not be executed. An error occurred during execution.</p> <p>Unload the ODK driver An exception occurred during the unloading of the CPU function library. The CPU function library has been unloaded nevertheless.</p>
0x8091	<p>Execute the ODK driver CPU function library could not be executed. A "STOP" occurred during the function call.</p>
0x8092	<p>Load the ODK driver CPU function library could not be loaded because the library name is invalid.</p>
0x8093	<p>Load the ODK driver</p>

7 Fehlermeldungen

7.1 Status und Error Codes ODK1500S

	CPU function library could not be loaded because the CPU function library could not be found. Check the file name and path of the file.
0x8094	<p>Load the ODK driver CPU function library could not be loaded. The CPU function library was created for the Windows user context, but no user is logged on.</p>
0x8095	<p>Load the ODK driver CPU function library could not be loaded due to the following reasons:</p> <ul style="list-style-type: none"> • The DLL file is not a CPU function library • An attempt has been made to load a 64-bit application into a 32-bit system • Dependencies on other Windows DLL files could not be resolved. <ul style="list-style-type: none"> – Check that the release build of the CPU function library is used. – Check whether the "Visual C++ Redistributables" are installed for the Visual Studio version you are using. • The CPU does not support the utilized ODK version.
0x8096	<p>Load the ODK driver The CPU function library could not be loaded because the internal identification is already being used by another loaded CPU function library.</p> <p>Execute the ODK driver CPU function library could not be executed because the CPU function library was not loaded or unloading is not yet finished.</p> <p>Unload the ODK driver CPU function library could not be unloaded because the CPU function library was not loaded or unloading is not yet finished.</p>
0x8098	<p>Load the ODK driver The CPU function library could not be loaded because the CPU function library is currently being unloaded.</p> <p>Execute the ODK driver CPU function library could not be executed because the function is not supported.</p>
0x8099	<p>Execute the ODK driver CPU function library could not be executed because the maximum amount of input data (1 MB) was exceeded (declarations with "In" and "InOut")</p>
0x809A	<p>Execute the ODK driver CPU function library could not be executed because the maximum amount of output data (1 MB) was exceeded (declarations with "Out" and "InOut")</p>
0x809B	<p>(Un)Load the ODK driver The CPU function library could not be (un)loaded and returns an invalid value (the values 0x0000 and 0xF000 - 0xFFFF are allowed)</p> <p>Execute the ODK driver The function returns an invalid value (a value between 0x0000 and 0x6FFF; 0xF000 and 0xFFFF is permitted)</p>
0x809C	<p>Function uses an invalid data type:</p> <ul style="list-style-type: none"> • IN_DATA • INOUT_DATA

	<ul style="list-style-type: none"> • OUT_DATA
0x80A4	<p>Load the ODK driver CPU function library could not be loaded. Start the ODK service manually or restart Windows.</p> <p>Execute the ODK driver CPU function library could not be executed for the following reasons:</p> <ul style="list-style-type: none"> • The "<STEP7Prefix>_Unload" instruction was executed during a function execution. The function execution was aborted at the CPU end. Windows terminates the execution of the function normally. No return value is sent to the CPU. <p>Wait until the "<STEP7Prefix>_Unload" instruction has ended. Then load the CPU function library again.</p> <ul style="list-style-type: none"> • Windows is not available • ODK service is not running <p>Start the ODK service manually or restart Windows.</p> <p>Unload the ODK driver CPU function library could not be unloaded for the following reasons:</p> <ul style="list-style-type: none"> • Windows is not available <p>Start the ODK service manually or restart Windows.</p>
0x80C2	<p>(Un)Load the ODK driver CPU function library could not be (un)loaded. There are currently not enough resources available from Windows. Reload the CPU function library after a few seconds.</p> <p>Execute the ODK driver CPU function library could not be executed. The CPU currently does not have enough resources. Execute the CPU function library again after a few seconds.</p>
0x80C3	<p>(Un)Load the ODK driver CPU function library could not be (un)loaded. The CPU currently does not have enough resources. Reload the CPU function library after a few seconds.</p> <p>Execute the ODK driver CPU function library could not be executed. The CPU currently does not have enough resources. Execute the CPU function library again after a few seconds.</p>
0xF000	<p>Load the ODK driver ODK1500S SQL driver license file is expired. Please purchase a full license</p>
0xF001	<p>Load the ODK driver ODK1500S SQL driver license file is invalid. (The license file must not be manipulated!)</p>
0xFFFF	<p>Load the ODK driver ODK1500S SQL driver license file does not exist in ODK1500S folder</p>

7.2 Error Codes ODK1500S SQL-Treiber

7.2.1 Function Error Codes

Diese Meldungen beschreiben Fehler der ODK-Funktionen des SQL-Treibers. Zum Beispiel wenn ein Datentyp in einem Datenbaustein nicht mit einem ausgelesenen Wert von der Datenbank übereinstimmt. Der Errorcode wird am Ausgang „fctErrorCode“ ausgegeben.

Tabelle 7-2: Function Error Codes

Error-Code	Beschreibung
9500	Die Argument Config Daten (sqlArgumentsConfigData) enthalten einen nicht unterstützenden Datentyp Identifier.
9501	Die Länge der Daten am Parameter sqlArguments passt nicht zu der Länge der Daten in sqlArgumentsConfigData
9502	Es ist eine Exception beim Prüfen der Config Daten aufgetreten. Für weitere Informationen bitte sqlErrorInfo prüfen.
9503	Es ist eine Exception beim Lesen der Argumente (sqlArguments) aufgetreten. Für weitere Informationen bitte sqlErrorInfo prüfen.
9510	Die von der Datenbank empfangenen Daten überschreiten die Größe der Daten am Parameter sqlReceivedData.
9511	Die von der Datenbank empfangenen Daten enthalten einen nicht unterstützten Datentyp.
9512	Es ist eine Exception beim Schreiben der empfangenen Daten (sqlReceivedData) aufgetreten. Für weitere Informationen bitte sqlErrorInfo prüfen.
9513	Es ist eine Exception beim Prüfen der Länge der empfangenen Daten aufgetreten. Für weitere Informationen bitte sqlErrorInfo prüfen.
9514	Die Länge der empfangenen Blob Daten überschreitet die maximale Länge von 65535 Bytes.
9515	Die Anzahl der zurückgegebenen Zeichen in einem Sting ist größer als die maximal unterstützte Länge von 254 Zeichen.
9520	Es ist eine Exception beim Ausführen der SQL Anweisung aufgetreten. Für weitere Informationen bitte sqlErrorInfo prüfen.
9521	Der Parameter tableName darf nicht leer sein.
9530	Ungültiger Wert am Parameter writeMode. Wert muss 0 (Insert) oder 1 (Update) sein.

7.2.2 SQL Error Info

Am Ausgang sqlErrorInfo wird die Fehlermeldung als Klartext ausgegeben. Die Fehlermeldungen können je nach Datenbank und ODBC Treiber variieren.

Hier werden auch Warnungen bzw. Hinweise ausgegeben. In diesem Fall wird der Ausgang fctError nicht gesetzt.

8 Ausgabe und Anzeige von Log-Informationen

Die Logging-Funktion des SQL-Treibers unterstützt bei der Konfiguration der SQL-Statements und der Überprüfung der vom Treiber gesendeten und empfangenen Nachrichten. Außerdem werden auftretende Fehlermeldungen dargestellt. Es besteht zudem die Möglichkeit, die Log-Informationen als Datei zu exportieren.

Für die Anzeige der Log-Informationen auf dem Runtime-Rechner kann z.B. das kostenlose Open Source Tool „DebugView++“ verwendet werden. Dieses kann unter dem folgenden Link heruntergeladen werden:

<https://github.com/CobaltFusion/DebugViewPP/releases>

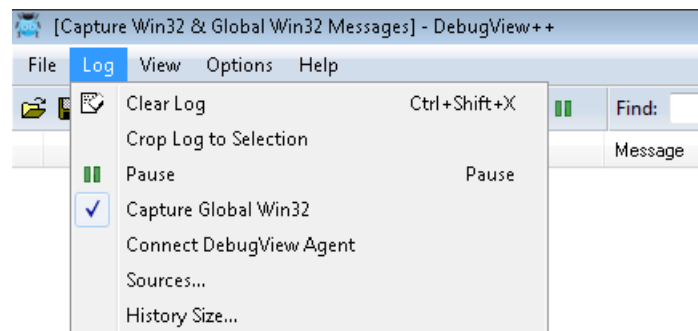
Es wird die aktuelle 64 bit release Version (DebugView++.zip) empfohlen.

Bedienung

Kopieren Sie das Tool auf den Runtime-Rechner (IPC auf dem der S7-1500 Software Controller läuft) und starten Sie das Tool mit Administrator Rechten.

Damit das Tool die Log-Ausgaben vom SQL-Treiber empfangen kann müssen Sie die Log-Einstellungen wie folgt festlegen:

Abbildung 8-1: Log-Einstellungen

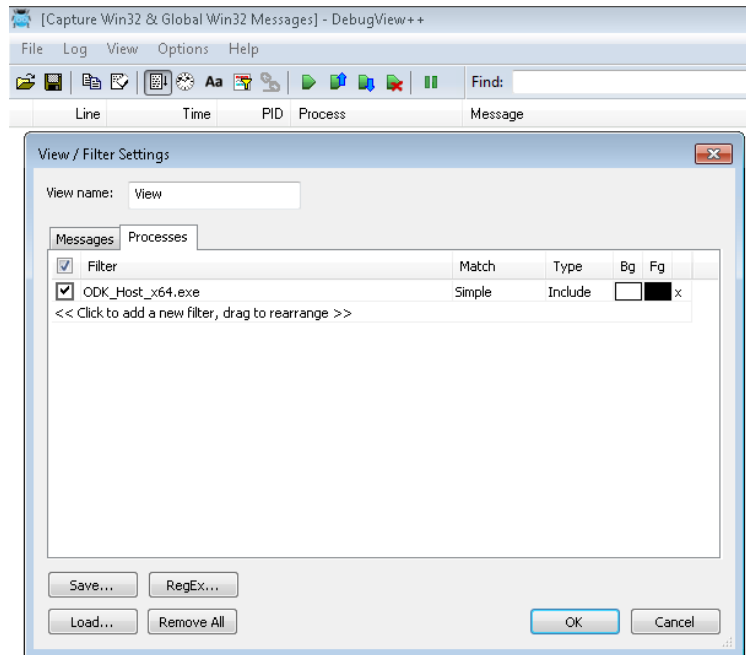


Damit nur Log-Informationen vom SQL-Treiber ausgegeben werden wird zudem empfohlen, den Filter über den Filter-Button auf den „ODK_Host_x64.exe“ Prozess wie folgt festzulegen:

8 Ausgabe und Anzeige von Log-Informationen

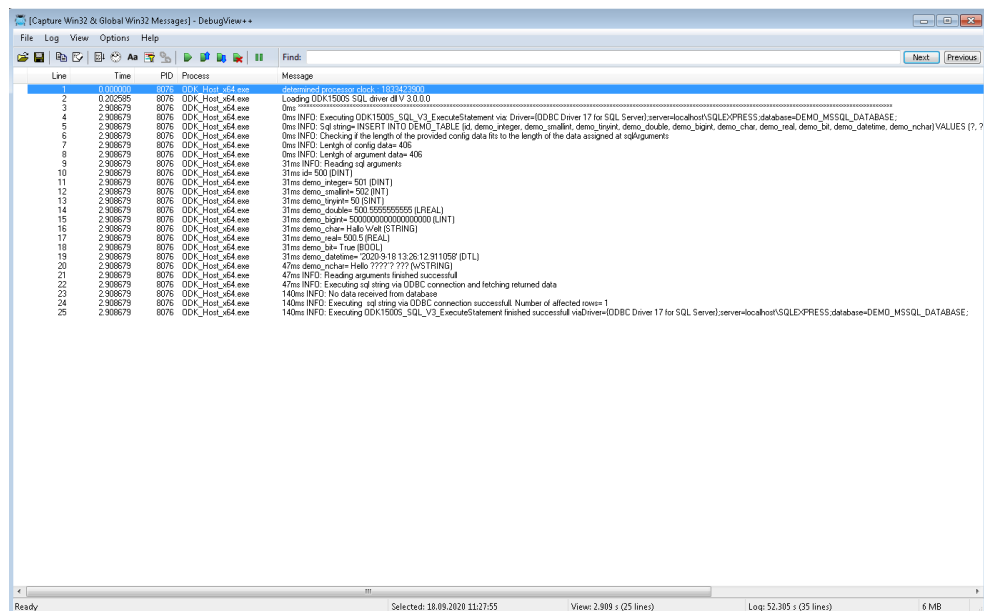
8.1

Abbildung 8-2: Filtereinstellung DebugView++



Nun werden von allen SQL-Treiber Bausteinen Log-Informationen ausgegeben, an denen der Parameter „enableLog“ aktiviert ist. Die Log-Informationen werden im SQL-Treiber gepuffert und erst ausgegeben, wenn die Abarbeitung des FBs abgeschlossen ist. Damit wird gewährleistet, dass bei paralleler Ausführung von SQL-Treiber FBs die Log-Informationen konsistent sind.

Abbildung 8-3: Ausgabe von Log-Informationen mit DebugView++



Über den Menüpunkt File->SaveLog können die Log-Informationen als Datei gespeichert werden.

9 Literaturhinweise

Tabelle 9-1

	Themengebiet	Titel
\1\	Siemens Industry Online Support	http://support.industry.siemens.com
\2\	Downloadseite des Beitrages	https://support.industry.siemens.com/cs/ww/de/view/109479140

10 Ansprechpartner

Bei Fragen zu dem ODK1500S SQL-Treiber oder Problemen bei der Anwendung können Sie über folgende Email-Adresse eine kostenlose Support-Anfrage stellen:

applications.aud.koe.nrh.rd@siemens.com

11 Historie

Tabelle 11-1

Version	Datum	Änderung
V1.0	08/2015	Erste Ausgabe
V1.1	06/2016	Unterstützung von 64-Bit Gleitkommazahlen Direkte Eingabe von SQL Statements möglich (Trennung von Daten und String nicht mehr zwingend erforderlich) Erstellung der Demodatenbank per Skript
V2.0	06/2017	Umsetzung mit ODK1500S V2 Neue Anwenderschnittstelle Neuer Funktionsbaustein zum Schreiben von Prozessdaten ohne SQL-Strings Neuer Funktionsbaustein zum Ausführen von SQL Anweisungen ohne Argumente Unterstützung von optimierten Datenbausteinen Unterstützung weiterer Datentypen (datetime, bit...) SQL-Anweisungen mit bis zu 16382 Zeichen aus dem Anwenderprogramm
V2.1	08/2017	Unterstützung vom Datentyp WString bzw. nvarchar (ermöglicht Unterstützung von Unicode) Unterstützung variabler Stringlänge in Send- und Empfangsrichtung (bis zu 254 Zeichen) Datenschnittstelle von 8kByte auf 265kByte vergrößert (sqlArguments / sqlReceivedData) Verbesserte Diagnose mit SQL Viewer Bugfixes
V2.2	10/2018	Timeout-Parameter für FB Connect und ExecuteStatement Kompatibilitätsmodus für SQL datetime Datentyp Ausgabe einer Warnung sowie Information über SQLViewer wenn über ein Select Statement ein Wert mit dem Inhalt "NULL" ausgelesen wird Allg. Bugfixes
V2.3	05/2020	Unterstützung (lesend) für BLOB-Datentypen sowie Dokumentation Versionierung-Prinzip.
V3.0	10/2020	Aktualisierung der Dokumentation im Bezug auf den ODK1500S SQL-Treiber V3.
V3.0	27.10.20	Update: Informationen für Oracle DB hinzugefügt
V3.1	07.12.20	Doku: Informationen für PostgreSQL Datentypen hinzugefügt. Treiber: Unterstützung variabler Stringlänge für das Lesen von MS SQL varchar und nvarchar Datentypen Optimierung der Treiber FB-Schnittstellen zur Reduzierung der Daten-Arbeitsspeicherauslastung in der PLC