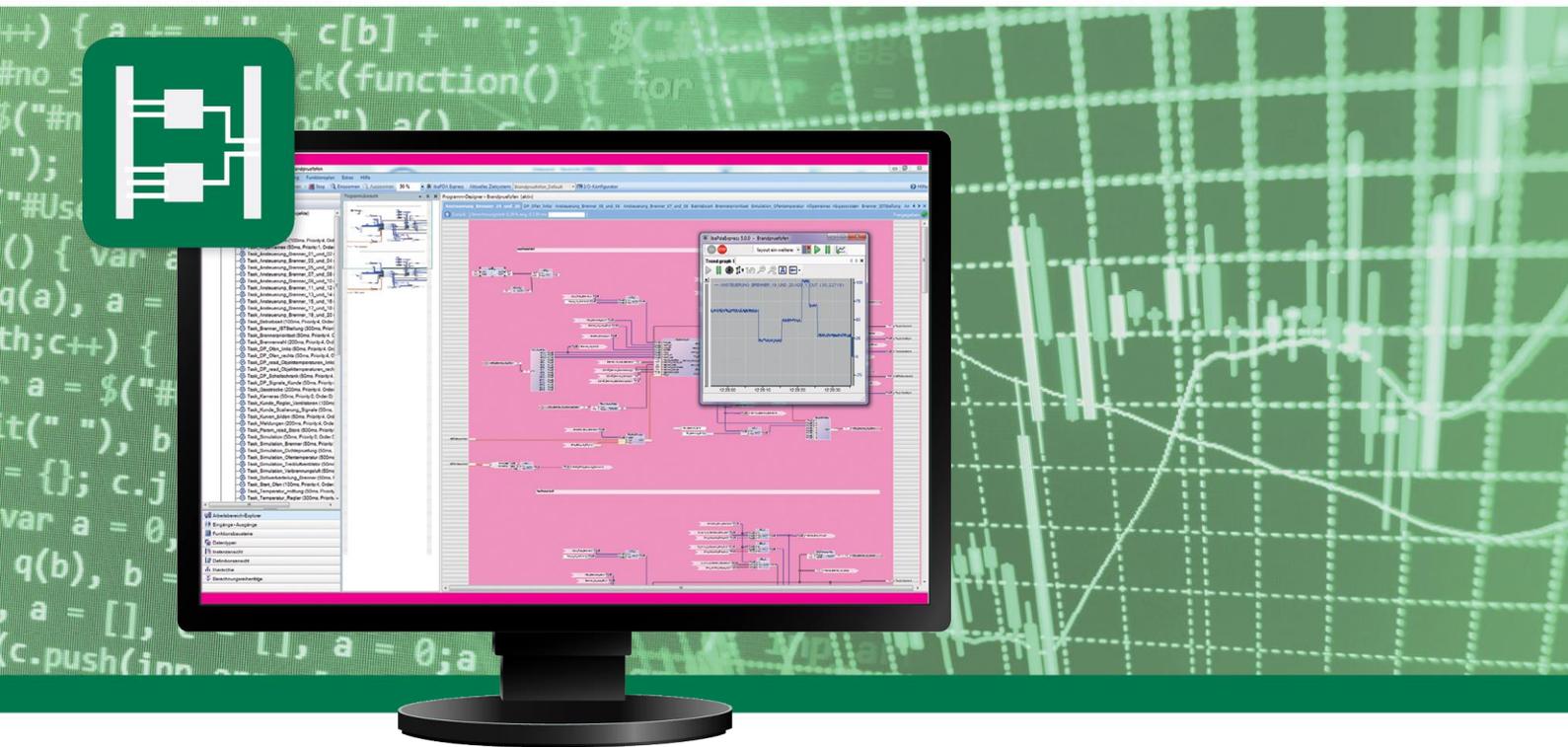




Das Ganze sehen



# ibaLogic ODBC\_ACCESS Funktionsbaustein

Handbuch

Ausgabe 1.0

**Messsysteme für  
Industrie und Energie**



---

Handbuch

# ibaLogic ODBC ACCESS

Ausgabe 1.0



iba AG

## Hersteller

iba AG  
Königswarterstr. 44  
90762 Fürth  
Deutschland

## Kontakte

Zentrale	+49 911 97282-0
Telefax	+49 911 97282-33
Support	+49 911 97282-14
Technik	+49 911 97282-13
E-Mail	iba@iba-ag.com
Web	www.iba-ag.com

Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts sind nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz.

© iba AG 2018, alle Rechte vorbehalten.

Der Inhalt dieser Druckschrift wurde auf Übereinstimmung mit der beschriebenen Hard- und Software überprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass für die vollständige Übereinstimmung keine Garantie übernommen werden kann. Die Angaben in dieser Druckschrift werden jedoch regelmäßig aktualisiert. Notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten oder können über das Internet heruntergeladen werden.

Die aktuelle Version liegt auf unserer Website [www.iba-ag.com](http://www.iba-ag.com) zum Download bereit.

Version	Datum	Revision - Kapitel / Seite	Autor	Version SW
1.0	27.09.2018	Erstausgabe	ST	5.4.0

Windows® ist eine Marke und eingetragenes Warenzeichen der Microsoft Corporation. Andere in diesem Handbuch erwähnte Produkt- und Firmennamen können Marken oder Handelsnamen der jeweiligen Eigentümer sein.

## Inhaltsverzeichnis

<b>1</b>	<b>Zu diesem Handbuch</b> .....	<b>2</b>
1.1	Zielgruppe.....	2
1.2	Schreibweisen.....	2
1.3	Verwendete Symbole.....	3
<b>2</b>	<b>Grundlagen</b> .....	<b>4</b>
<b>3</b>	<b>Konfiguration für Standard-SQL Befehle</b> .....	<b>5</b>
3.1	Anlegen einer Datenbank (MS SQL).....	5
3.2	Anlegen einer ODBC Verbindung zu MS SQL.....	7
3.3	Konfiguration des ODBC_ACCESS-Bausteins.....	10
3.4	Beispiel für StoredProcedure in MSSQL.....	16
<b>4</b>	<b>Konfiguration für Aufruf von StoredProcedure in Oracle</b> .....	<b>18</b>
4.1	ODBC Verbindung zu Oracle.....	20
<b>5</b>	<b>Anhang</b> .....	<b>23</b>
5.1	ODBC Datentypen (MSSQL/ORACLE).....	23
<b>6</b>	<b>Support und Kontakt</b> .....	<b>26</b>

# 1 Zu diesem Handbuch

Diese Dokumentation beschreibt die Funktion und die Anwendung des Funktionsbausteins ODBC\_ACCESS in *ibaLogic-V5*.

Dieses Dokument ist eine Ergänzung zu dem allgemeinen Handbuch zu *ibaLogic*, in dem die allgemeinen Funktionen und Bedienmöglichkeiten von *ibaLogic* beschrieben sind.

## 1.1 Zielgruppe

Im Besonderen wendet sich dieses Handbuch an ausgebildete Fachkräfte, die mit dem Umgang mit elektrischen und elektronischen Baugruppen sowie der Kommunikations- und Messtechnik vertraut sind. Als Fachkraft gilt, wer auf Grund seiner fachlichen Ausbildung, Kenntnisse und Erfahrungen sowie Kenntnis der einschlägigen Bestimmungen die ihm übertragenen Arbeiten beurteilen und mögliche Gefahren erkennen kann.

## 1.2 Schreibweisen

In dieser Dokumentation werden folgende Schreibweisen verwendet:

Aktion	Schreibweise
Menübefehle	Menü „Funktionsplan“
Aufruf von Menübefehlen	“Schritt 1 – Schritt 2 – Schritt 3 – Schritt x” Beispiel: Wählen Sie Menü „Funktionsplan – Hinzufügen – Neuer Funktionsblock”
Tastaturtasten	<Tastename> Beispiel: <Alt>; <F1>
Tastaturtasten gleichzeitig drücken	<Tastename> + <Tastename> Beispiel: <Alt> + <Strg>
Grafische Tasten (Buttons)	<Tastename> Beispiel: <OK>; <Abbrechen>
Dateinamen, Pfade	"Dateiname", "Pfad" Beispiel: "Test.doc"

## 1.3 Verwendete Symbole

Wenn in dieser Dokumentation Sicherheitshinweise oder andere Hinweise verwendet werden, dann bedeuten diese:



---

### **Gefahr! Stromschlag!**

Wenn Sie diesen Sicherheitshinweis nicht beachten, dann droht die unmittelbare Gefahr des Todes oder schwerer Körperverletzung durch einen Stromschlag!

---



---

### **Gefahr!**

Wenn Sie diesen Sicherheitshinweis nicht beachten, dann droht die unmittelbare Gefahr des Todes oder der schweren Körperverletzung!

---



---

### **Warnung!**

Wenn Sie diesen Sicherheitshinweis nicht beachten, dann droht die mögliche Gefahr des Todes oder schwerer Körperverletzung!

---



---

### **Vorsicht!**

Wenn Sie diesen Sicherheitshinweis nicht beachten, dann droht die mögliche Gefahr der Körperverletzung oder des Sachschadens!

---



---

### **Hinweis**

Hinweis, wenn es etwas Besonderes zu beachten gibt, wie z. B. Ausnahmen von der Regel usw.

---



---

### **Wichtiger Hinweis**

Hinweis, wenn etwas Besonderes zu beachten ist, z. B. Ausnahmen von der Regel.

---



---

### **Tipp**

Tipp oder Beispiel als hilfreicher Hinweis oder Griff in die Trickkiste, um sich die Arbeit ein wenig zu erleichtern.

---



---

### **Andere Dokumentation**

Verweis auf ergänzende Dokumentation oder weiterführende Literatur.

---



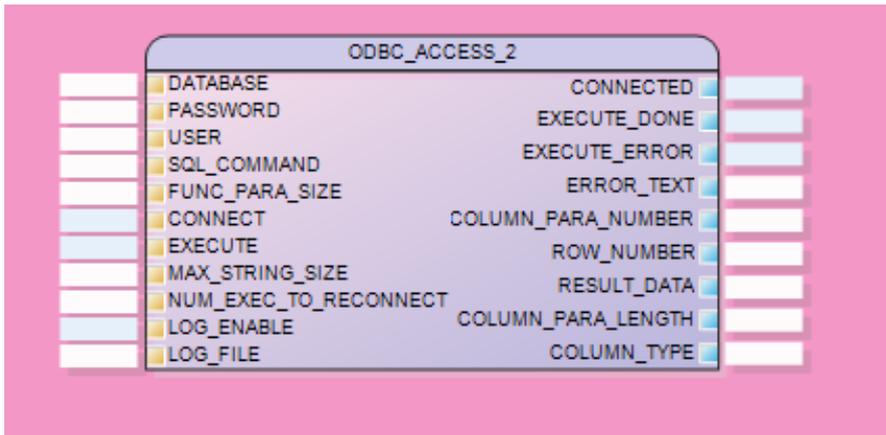
---

### **Beispiel**

Konfigurations- und Anwendungsbeispiele zum besseren Verständnis

---

## 2 Grundlagen



Mit dem ODBC\_ACCESS Baustein kann man in Datenbanken schreiben und aus Datenbanken lesen. Dazu muss eine ODBC-Verbindung zu der jeweiligen Datenbank möglich sein und eingerichtet werden.

Der ODBC\_ACCESS-Baustein ist nur auf einem ibaLogic-System lauffähig, welches auf einem PC läuft. ibaPADU-S-IT-2x16 kann den Baustein systembedingt nicht verwenden.

Es sind Standard-SQL-Befehle erlaubt. StoredProcedure-Aufrufe sind ebenfalls möglich.

Der ODBC\_ACCESS Baustein ist lizenzpflichtig und muss über eine entsprechende Freischaltung im Dongle aktiviert werden.

Bestell-Nr.	Produktbezeichnung	Beschreibung
32.500030	ibaLogic-V5-DB ACCESS	Lizenz für ibaLogic-V5 Funktionsbaustein für DB Access

Zur Anwendung legt man den SQL-Befehl auf den Eingang des Bausteins. Der Ergebnis-Ausgang kann direkt auf die erforderliche Ergebnis-Struktur gelegt werden. Sind mehrere unterschiedliche Ergebnisstrukturen zu erwarten, kann dies über entsprechende Multiplexer realisiert werden.

Nachfolgend werden unterschiedliche Konfigurationen beschrieben:

- Konfiguration für Standard-SQL Befehle wie SELECT/INSERT/DELETE etc.
- Konfiguration für den Aufruf von StoredProcedure wie CALL ....

### 3 Konfiguration für Standard-SQL Befehle

Notwendige Schritte für einen Datenbankzugriff über ODBC:

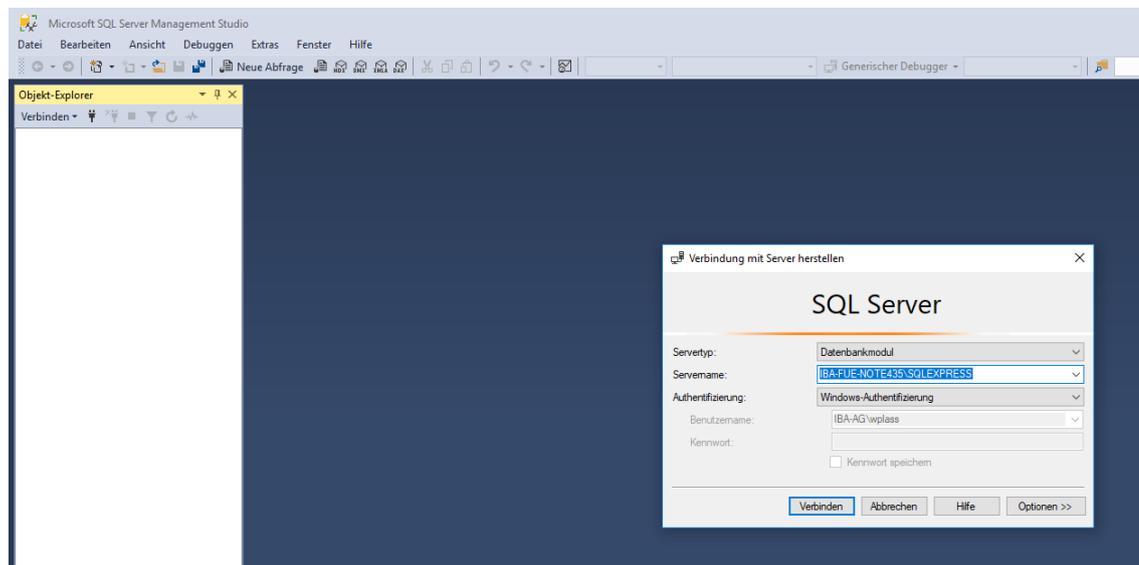
1. Anlegen der Datenbank und Tabelle(n) für den Zugriff, wenn diese noch nicht existieren sollten
2. Erstellen einer ODBC Verbindung zu dieser Datenbank
3. Konfigurieren des ODBC-Bausteins in ibaLogic mit
  - ODBC-Verbindungsname
  - SQL-Befehl
  - Logik für Ausführung des SQL-Befehls
  - Dekodierung des Ergebnisses in entsprechende Ergebnisstrukturen

#### 3.1 Anlegen einer Datenbank (MS SQL)

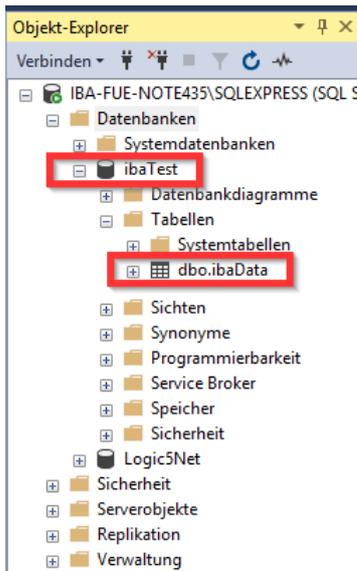
Ist keine spezielle Datenbank vorhanden, kann man auch die installierte Datenbank von ibaLogic benutzen. Das ist normalerweise eine MS-SQLEXPRESS-Datenbank.

Für den Zugriff auf die Datenbank und deren Tabellen, ist ein Download von Microsoft SQL Server Management Studio erforderlich.

Starten Sie das SQL Server Management Studio und loggen sich mit Ihrer Windows-Authentifizierung ein.



Legen Sie sich eine Datenbank und eine neue Tabelle an (hier: ibaTest und ibaData). Das Präfix dbo. wird vom System erzeugt und ist nicht Teil des Tabellennamens, wenn man diese später von außen anspricht.



Hinweis: Logic5Net ist die ibaLogic-eigene DB.

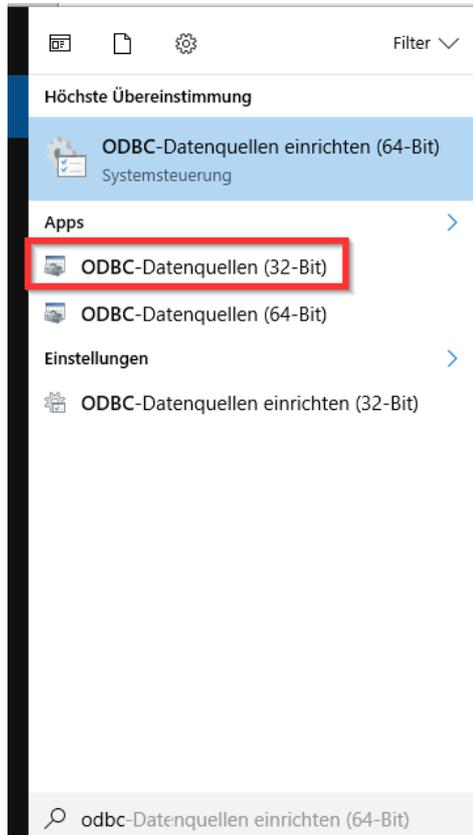
Das SQL Server Management Studio kann auch verwendet werden, um Testdaten anzulegen oder um Änderungen, die durch den ODBC\_ACCESS Baustein von ibaLogic aus gemacht wurden, zu überprüfen.

## 3.2 Anlegen einer ODBC Verbindung zu MS SQL

Hier wird exemplarisch die ODBC-Verbindung zu einer MS-SQL-Datenbank beschrieben.

Hinweis: Es gibt ODBC Einstellungen für 32/64-Bit-Systeme.

Im Beispiel wird die ODBC-Einstellungen für 32-Bit benötigt, da es sich um eine 32-Bit Applikation handelt (auch wenn es ein 64-Bit-System ist).



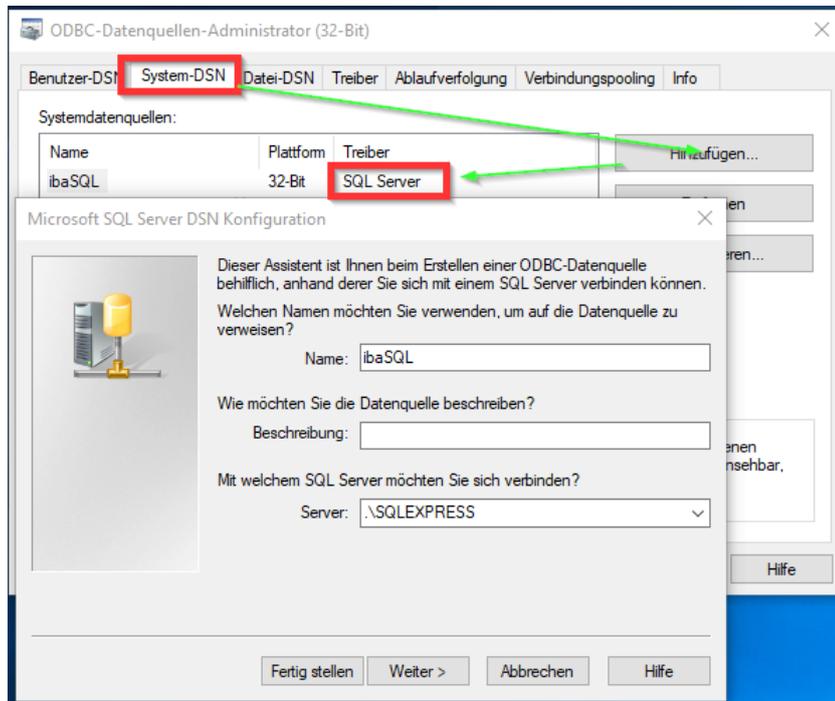
### Hinweis

Findet man die passenden ODBC Einstellungen nicht über die Suche im Windows-System, kann man sie auch direkt starten `C:\Windows\SysWOW64\odbcad32.exe`.

Weitere Hinweise sind hier zu finden:

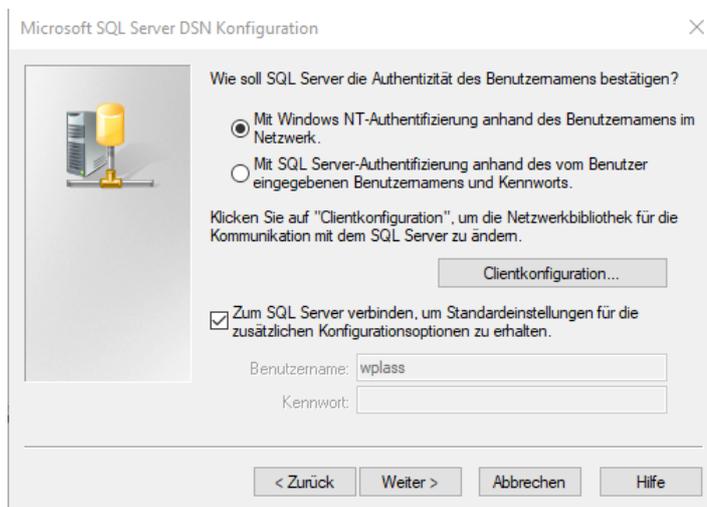
<https://support.microsoft.com/de-de/help/942976/odbc-administrator-tool-displays-both-the-32-bit-and-the-64-bit-user-d>

1. Tragen Sie unter System-DSN eine neue Verbindung ein und wählen den passenden ODBC-Treiber aus (hier: SQL Server).
2. Geben Sie der ODBC-Verbindung einen frei wählbaren Namen (hier: ibaSQL)
3. Tragen Sie den Namen des SQL-Servers ein. Diesen finden Sie über das SQL Server Management Studio heraus.

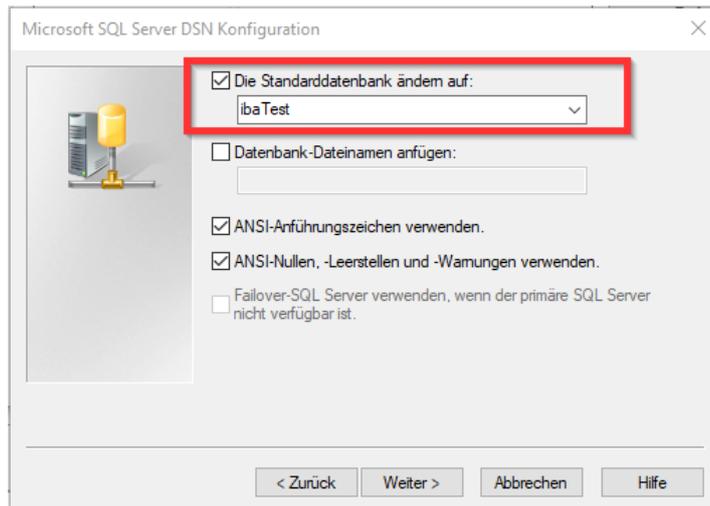


Klicken Sie <Weiter>.

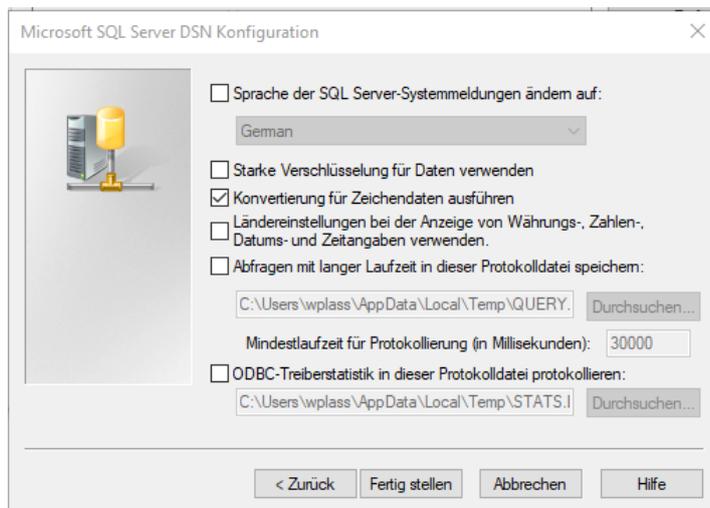
4. Prüfen Sie diese Einstellungen und klicken <Weiter>.



5. Tragen Sie den Tabellennamen ein und klicken <Weiter>.



6. Prüfen Sie diese Einstellungen:



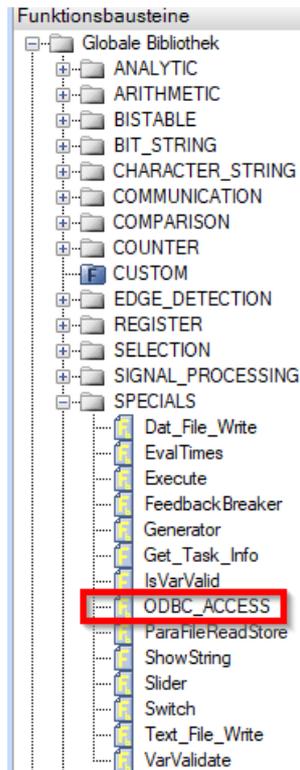
7. Schließen Sie mit <Fertig stellen> ab.

Damit ist die ODBC Verbindung eingerichtet.

### 3.3 Konfiguration des ODBC\_ACCESS-Bausteins

Hierzu gibt es das Beispiel ODBC\_SQL\_example.il5 auf der DVD „iba Software & Manuals“.

Der ODBC\_ACCESS Baustein ist hier zu finden:



Ziehen Sie den Baustein per Drag & Drop in das Layout.

Die Konfiguration wird über einen Select-Befehl exemplarisch gezeigt. Das nachfolgende Beispiel zeigt den Inhalt einer abzurufenden Datenbank-Tabelle.

Die Tabelle enthält ID (Int) und WEIGHT (Real) und entsprechende Werte.

The screenshot shows the 'Objekt-Explorer' (Object Explorer) on the left and a SQL query window on the right. The query window shows the following SQL code:

```

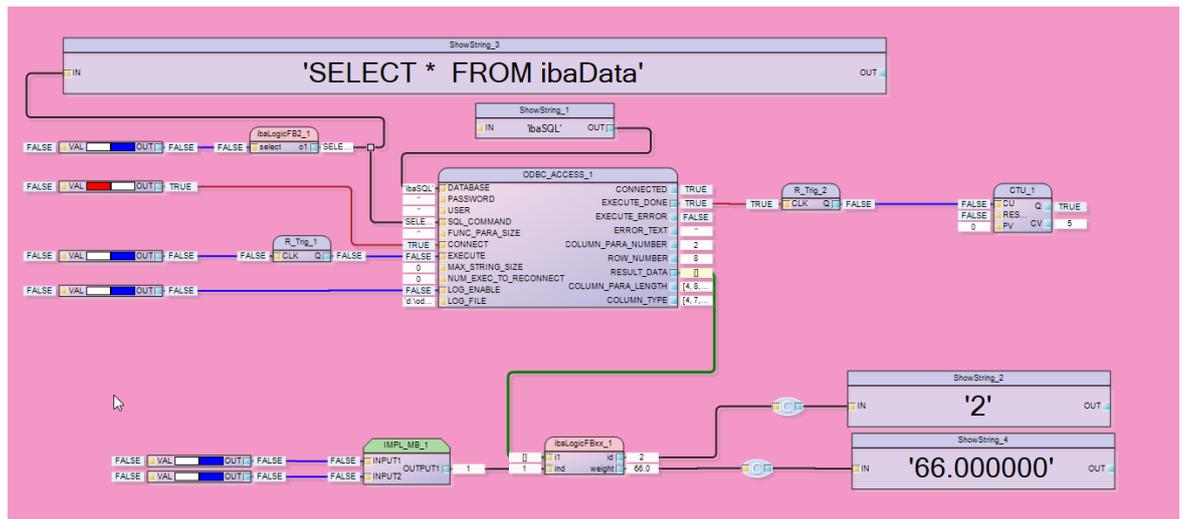
/***** Skript für SelectTopNRows-Befehl aus SSMS *****/
SELECT TOP (1000) [id]
,[weight]
FROM [ibaTest].[dbo].[ibaData]
  
```

The 'Ergebnisse' (Results) pane shows the following data:

	id	weight
1	1	45
2	2	66
3	20	13,45
4	21	66,45
5	33	33,33
6	38	33,33
7	...	...

The status bar at the bottom indicates: IBA-FUE-NOTE435\SQLEXPRESS ... | IBA-AG\wplass (54) | ibaTest | 00:00:00 | 8 Zeilen

Abfrage per ODBC\_ACCESS:



Eingänge

Konnektor	Datentyp	Bedeutung/Verwendung
DATABASE	String	ODBC-Datenquellenname
PASSWORD / USER	String	Anmeldedaten bei zugriffsgeschützter Datenbank
SQL_COMMAND	String	Standard-SQL-Befehl (hier: SELECT * from ibaDATA)
FUNC_PARA_SIZE	String	Datengröße der Parameter entsprechend der Datenstruktur (nur bei StoredProcedure-Aufruf), z.B.: 'INT,REAL,STRING10,REAL'
CONNECT	Bool	TRUE: Verbindung zur Datenbank wird aufgebaut
EXECUTE	Bool	Impulsgesteuerter Eingang zur Ausführung des SQL-Befehls
MAX_STRING_SIZE	Dint	0 : 1024 Byte als maximale String-Länge. Um den Speicherbedarf für das Ergebnis zu reduzieren, kann man die maximale String-Länge begrenzen. Dies ist beim Ergebnis durch eine entsprechende String derived type Definition mit dieser Länge zu berücksichtigen.
NUM_EXEC_TO_DISCONNECT	Dint	Anzahl der Execute-Befehle, nach denen ein automatischer Reconnect auf die Datenbank erfolgt. Dies kann notwendig sein, wenn Datenbanken temporäre Daten nicht abbauen. 0: 50 Execute-Befehle (fest eingestellter Defaultwert)
LOG_ENABLE	Bool	Freigabe zum Schreiben eines Log-Files.
LOG_FILE	String	Der Log-File Name besteht aus Pfad und Dateiname, z.B. 'D:\ODBC_ACCESS.log' Alle Execute-Befehle und Fehlermeldungen werden mit Datum/Uhrzeit in eine Textdatei geschrieben. (Kein automatischer Aufräum-Mechanismus).

**Ausgänge:**

Konnektor	Datentyp	Bedeutung/Verwendung
CONNECTED	Bool	TRUE, wenn die Verbindung zur Datenbank besteht.
EXECUTE_DONE	Bool	TRUE, wenn der SQL Befehl ausgeführt wurde. Hinweis: Bei jeder Ausführung gibt der Ausgang eine Low-Flanke aus, sodass mit einem nachgeschalteten Zähler-Baustein die Ausführungen mitgezählt bzw. nachfolgende Aktionen getriggert werden können.
EXECUTE_ERROR	Bool	TRUE bei Auftreten eines Fehlers.
ERROR_TEXT	String	Klartext der Fehlermeldung
COLUMN_PARA_NUMBER	Dint	Anzahl der Spalten des Ergebnisses. (Hier: 2, da es zwei Werte ID und WEIGHT in der Datenbank sind)
ROW_NUMBER	Dint	Anzahl der Datensätze des Ergebnisses. (Hier: 8, da es 8 Datensätze in der Datenbank sind und alle abgerufen wurden)
COLUMN_PARA_LENGTH	ICPBUF_INT	Array mit Anzeige der Datenlänge in BYTE für jede Ergebniszelle (Hier: 4 Byte für den INT Datentyp in der Datenbank und 8 Byte für den REAL Datentyp der Datenbank)
COLUMN_TYPE	ICPBUF_INT	Array mit Anzeige des SQL-Datentyps für jede Ergebniszelle. (Hier: 4 für den Datenbank-Typ INTEGER und 7 für den Datenbank-Typ REAL) 0 = unbekannter Typ 1 = CHAR 2 = NUMERIC 3 = DECIMAL 4 = INTEGER 5 = SMALLINT 6 = FLOAT 7 = REAL 8 = DOUBLE 9 = DATETIME 12 = VARCHAR
RESULT_DATA	Any	Ergebnis der zurückgegebenen Datensätze. Zur Weiterverarbeitung muss eine passende Struktur angelegt werden.

Die angelegte Struktur muss zu den empfangenden Daten passen.

**Hinweis**

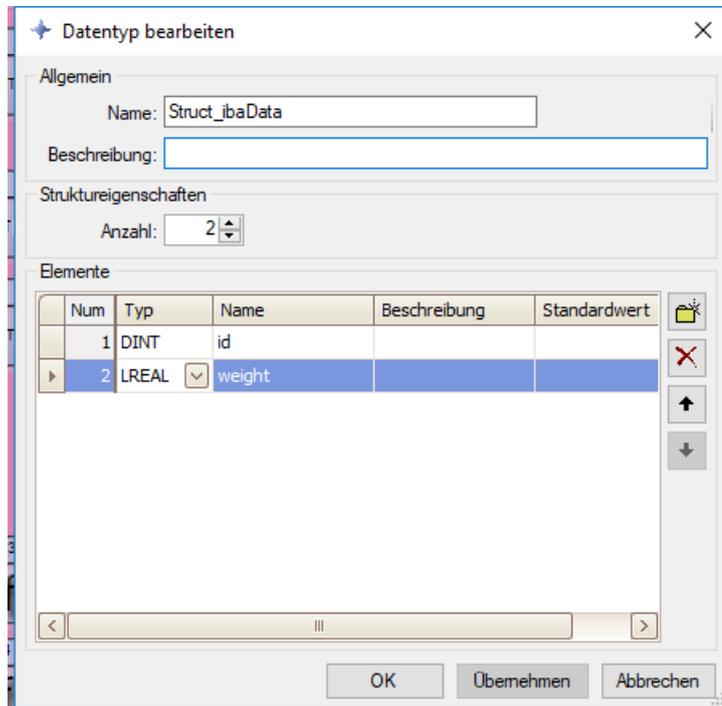
Hat man unterschiedliche Abfragen, die unterschiedliche Datensatzarten zurückgeben, muss man die Empfangsdaten multiplexen. Dies wird im zweiten Beispiel „Konfigurieren von StoredProcedure Calls in Oracle“ angewendet und dort beschrieben.

In unserem Beispiel brauchen wir eine Struktur mit zwei Members ID und WEIGHT. Dies entspricht einem Datensatz. Die Anzahl der zurückgegebenen Datensätze muss dann in ein passendes Array von Datensätzen geschrieben werden.

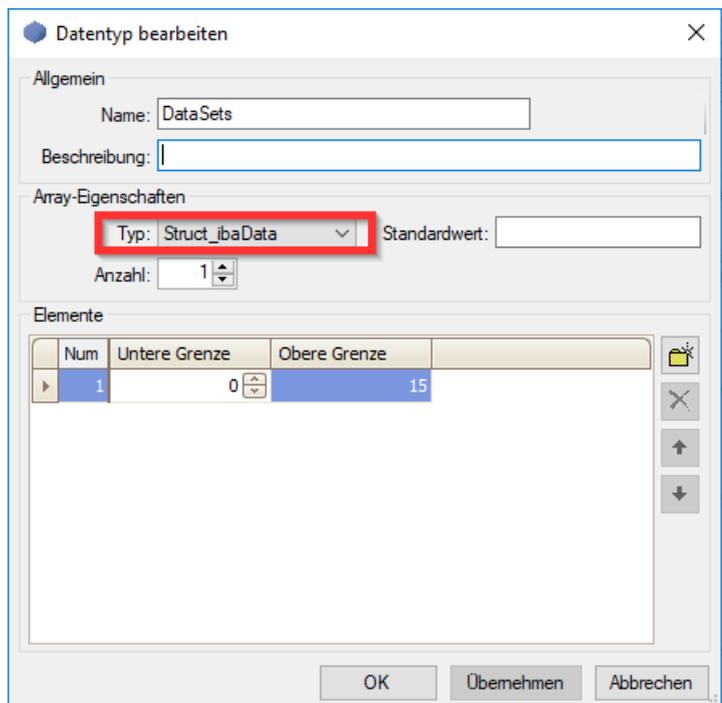
Daher legen wir uns eine Datensatz-Struktur an und verwenden diese dann in einem Array.

Zuerst muss die Datensatzstruktur deklariert werden. Dabei ist zu beachten, dass die Datentypen von MS-SQL und ibaLogic teilweise verschiedene BYTE-Länge haben.

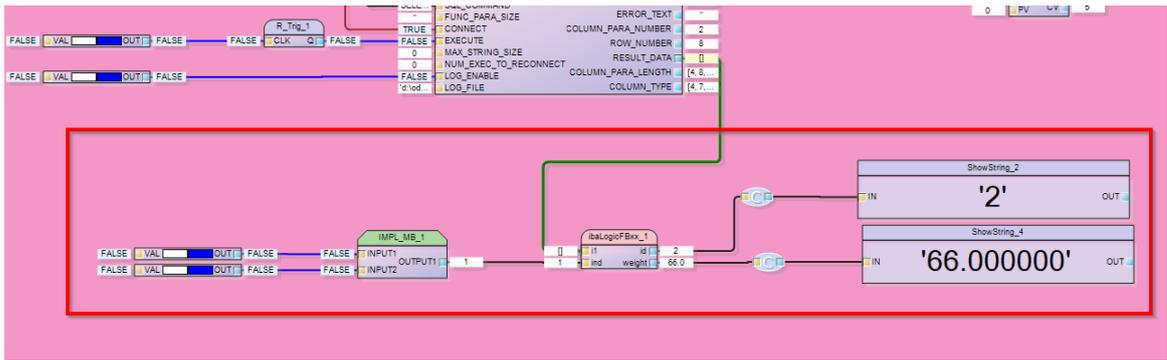
In unserem Fall entspricht ein MS-SQL INTEGER einem ibaLogic DINT und ein MS-SQL REAL einem ibaLogic LREAL. Daher legen wir die folgende Datensatzstruktur an.



Diese verwenden wir in einem Array, welches die maximal zu erwartenden Datensätze aufnehmen können sollte. (hier: 16)



Nun können wir die Datensätze zur Weiterverarbeitung auslesen. Nachfolgend ein Beispiel des Zugriffs:



Über den IMPL\_MB Macro kann ein Index = Datensatznummer angewählt werden. Der Baustein ibaLogicFBxx gibt dann die Werte des jeweiligen Datensatzes aus.

Der Inhalt des ibaLogicFBxx zeigt, wie man den Datensatz anspricht.

Funktionsblock bearbeiten

Definitionenname: ibaLogicFBxx

Beschreibung:

Instanzname: ibaLogicFBxx 1

Anzahl Eingänge: 2    Anzahl Ausgänge: 2    Anzahl Variablen: 0

Argumente

Variablen

Variablentyp	Index	Datentyp	Name	Standardwert	Beschreibung	Aktueller Istwert...
Variabletyp: Input	1	DataSets	i1			
	2	INT	ind	0		1
Variabletyp: Output	1	DINT	id	0		2
	2	LREAL	weight	0.0		66.0

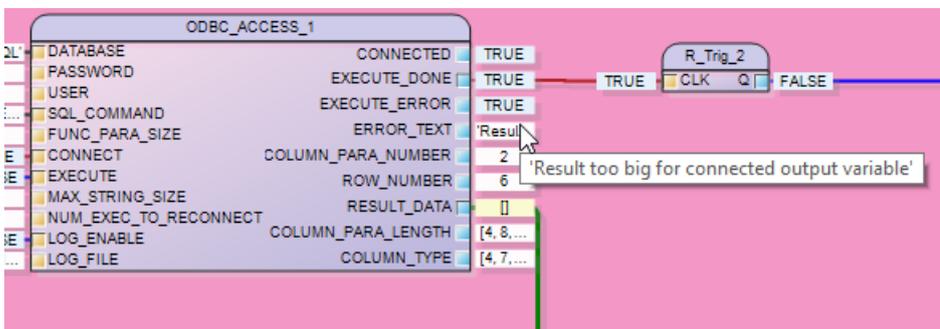
Structured Text

```

1 weight:= il[ind].weight;
2 id:=il[ind].id;
3
4
5
6
7
8
9
    
```

ST prüfen    Intellisense deaktivieren    OK    Abbrechen

Ist das Datensatz-Array zu klein um alle Empfangsdaten aufzunehmen, erhält man eine Fehlermeldung.



Beispiel bei einer Datensatzstruktur, die nur 6 Datensätze zulässt, aber eigentlich mehr abgerufen worden sind:

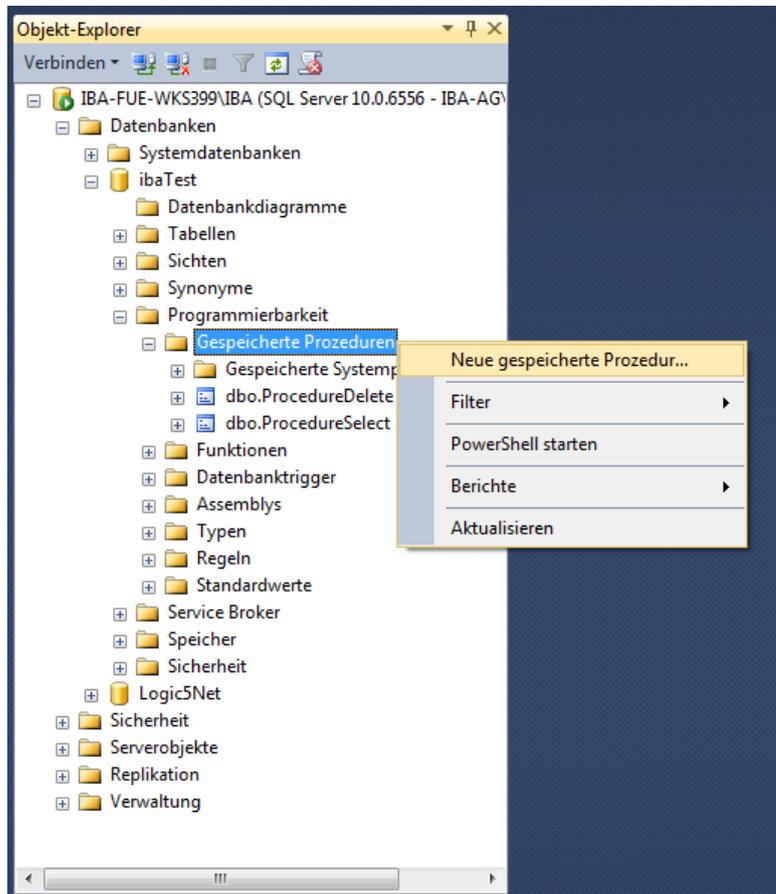
Es werden die 6 möglichen übertragen (Row number = 6) und eine Fehlermeldung gibt den Hinweis, dass eigentlich mehr Datensätze vorhanden sind. Der Benutzer muss dann seine Struktur vergrößern oder gezielt weniger Datensätze abfragen.

### 3.4 Beispiel für StoredProcedure in MSSQL

Das Beispiel für StoredProcedure in MSSQL kann in ibaLogic importiert werden. Die Importdatei ODBC\_StoredProcedure\_MSSQL.il5 befindet sich auf der DVD „iba Software & Manuals“.

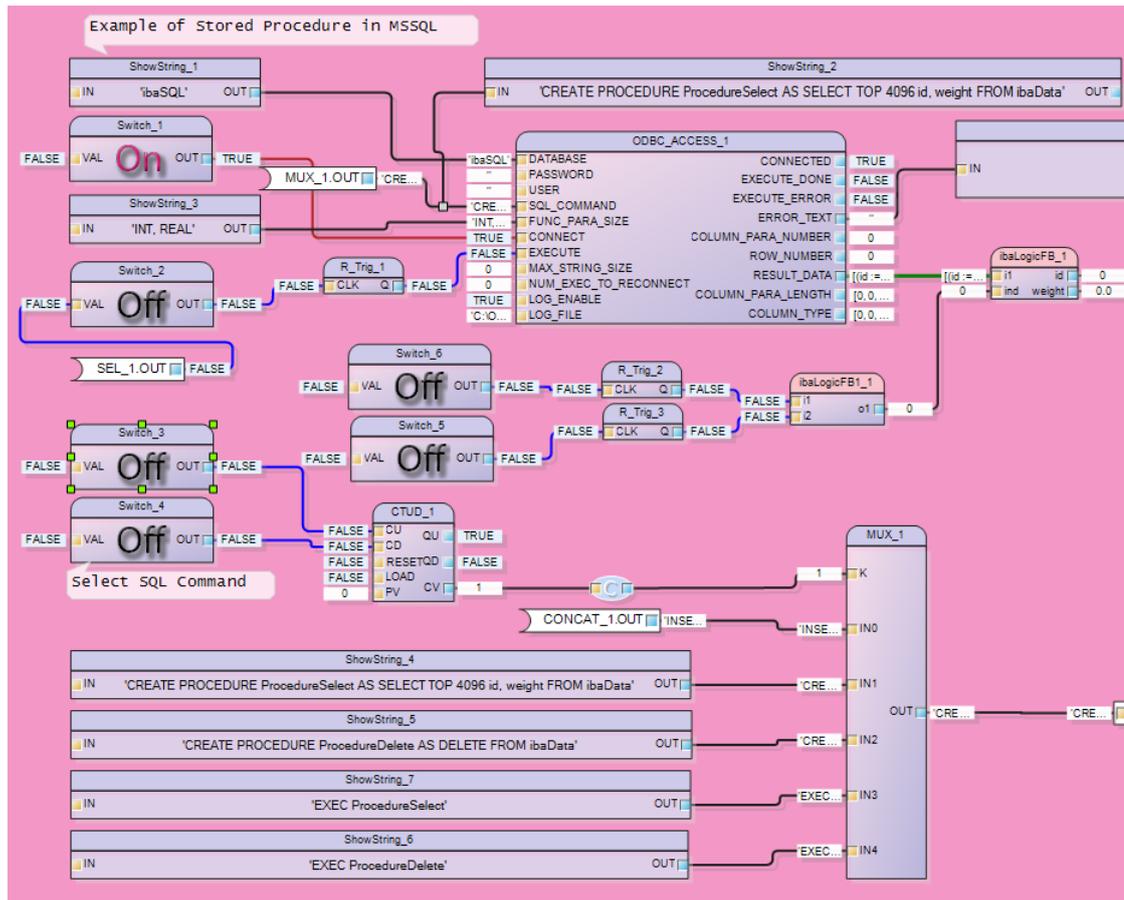
Ein StoredProcedure kann im Microsoft SQL Server Management Studio selbst (siehe Abbildung unten) oder mit Hilfe des ODBC\_ACCESS Bausteins über den Befehl „CREATE PROCEDURE procedurname AS ...“ erstellt werden.

Mit dem Beispielprogramm kann man eine SELECT- und eine DELETE-Prozedur erstellen und diese dann mit dem Befehl „EXEC procedurname“ ausführen.



Der Einfachheit halber wird die bereits konfigurierte Datenbankverbindung ibaSQL mit der Tabelle ibaTest und den darin enthaltenen Spalten ID (Int) und WEIGHT (Real) genutzt.

**Beispiel**



Beim Starten der Berechnung liegt ein INSERT-Befehl am Eingang des ODBC\_ACCESS Bausteins. Um die Datenbank zuerst mit Werten zu befüllen, kann der Schalter Switch\_7 betätigt werden. Es werden im Sekundentakt Testdaten in die Spalten geschrieben.

Über die Taster Switch\_3 und Switch\_4 können die entsprechenden SQL-Befehle ausgewählt werden.

Betätigen Sie den Taster Switch\_3, um den Befehl „CREATE PROCEDURE ProcedureSelect AS SELECT TOP 4096 id, weight FROM ibaData“ bzw. „CREATE PROCEDURE ProcedureDelete AS DELETE FROM ibaData“ auszuwählen. Durch Betätigen des Tasters Switch\_2 wird der Befehl ausgeführt.

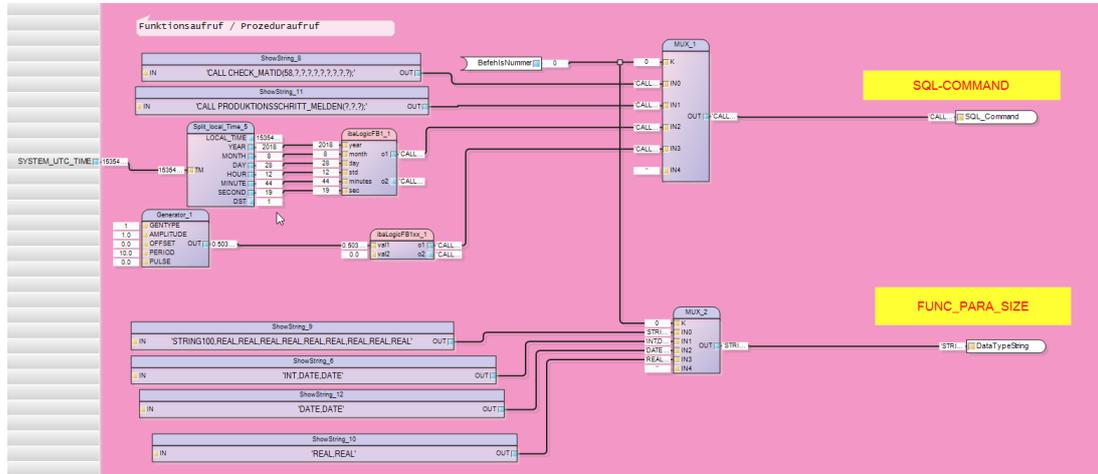
Sie können im Microsoft SQL Server Management Studio prüfen, ob die Prozeduren angelegt wurden und diese auch dort ausführen. Um die Prozeduren von ibaLogic aus auszuführen, gehen Sie wie beim Erstellen der Prozeduren vor und wählen die entsprechenden Befehle „EXEC ProcedureSelect“ bzw. „EXEC ProcedureDelete“ aus und betätigen Sie Taster Switch\_2, um die Ausführung zu triggern.

Am Ausgang RESULT\_DATA werden die empfangenen Daten in einem Array angezeigt, die Sie einzeln durch Betätigen der Schalter Switch\_5 und Switch\_6 ansehen können.

## 4 Konfiguration für Aufruf von StoredProcedure in Oracle

Hierzu gibt es das Beispiel ODBC\_StoredProcedure\_Oracle.il5 auf der DVD „iba Software & Manuals“. Hier werden nur die Besonderheiten bezogen auf die StoredProcedure beschrieben.

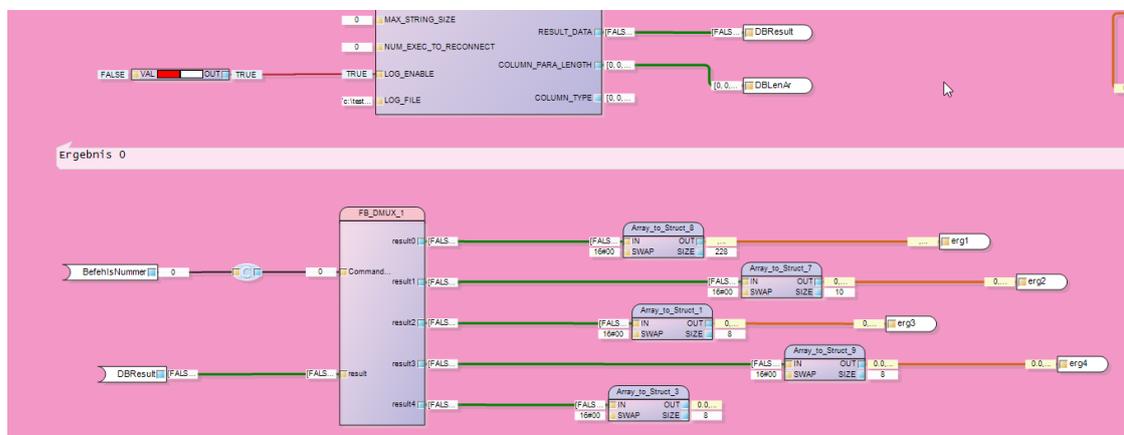
Eine StoredProcedure besteht aus dem Aufruf CALL procedurename (parameter).



Dabei kann man Werte übergeben und bekommt Werte zurück. Die Rückgabewerte sind im Aufruf mit ? versehen. Damit die Datenübergabe funktioniert, müssen auch die Datentypen mit übergeben werden. Dies erfolgt durch Klartext-String am Eingang FUNC\_PARA\_SIZE.

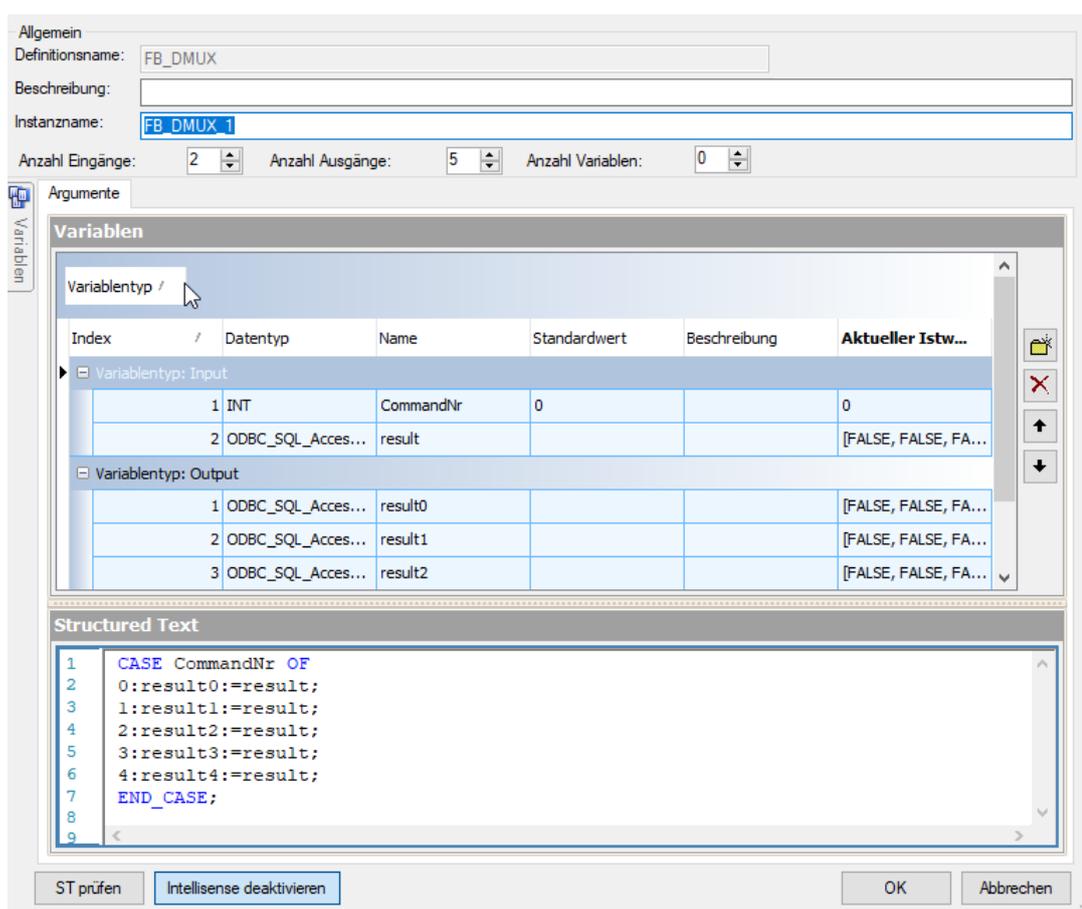
In diesem Beispiel werden unterschiedliche Prozeduren aufgerufen. Mit unterschiedlichen Parametern und damit mit unterschiedlichen Ergebnissen.

Man muss abschätzen, wie groß die maximalen Empfangsdaten sind und ein entsprechendes Byte-Array am Ausgang anlegen. Je nach Anforderung wird dann auf dieses Array eine andere Struktur, entsprechend dem zu erwartenden Ergebnis, gelegt. Dazu dient vor allem der ARRAY\_TO\_STRUCT Baustein.



Die Befehlsnummer bestimmt die Anfrage und schaltet auch das Ergebnis über den FB-Mux auf den richtigen Ausgang. Dort wird jeweils eine entsprechende Struktur über einen ARRAY\_TO\_STRUCT Baustein versorgt. Diese Struktur kann dann weiterverarbeitet werden.

Der Multiplexer sieht wie folgt aus:

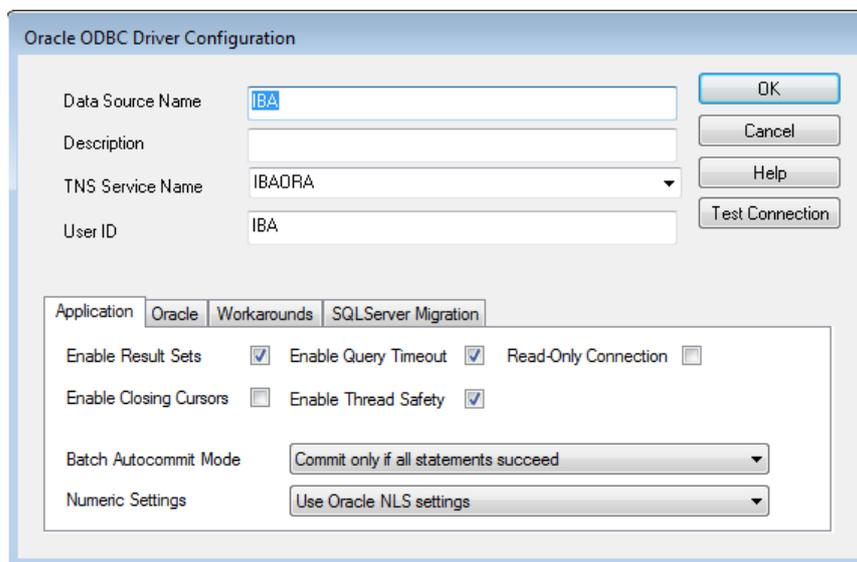
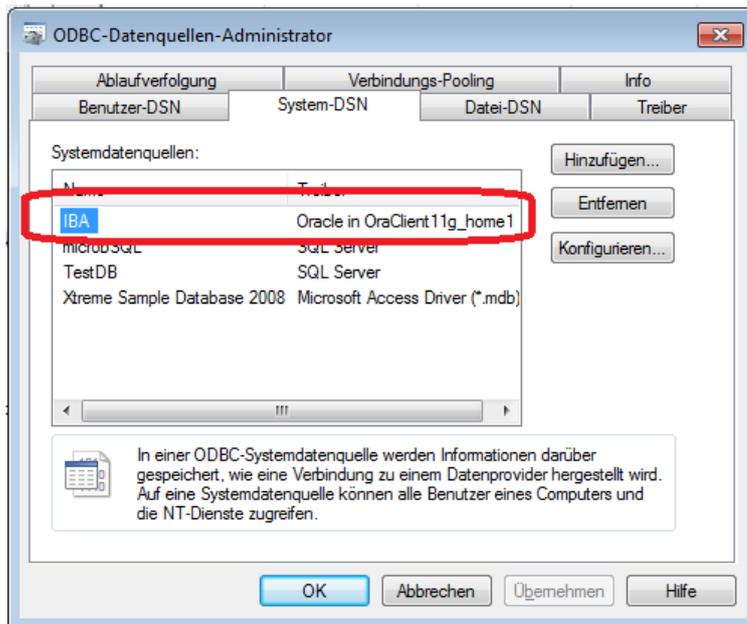


Bei den Strukturen ist zu beachten:

- Eine DATE-Größe kommt als UDINT (UTC-Zeit) an.
- Ein String-Wert muss entsprechend seiner Größe in der Struktur vorhanden sein. So ist ein STRING100 als Datentyp „String der Größe 100" zu definieren (siehe String derived type).
- Der Eingang MaxStringSize hat Vorrang. Steht dieser auf 50 werden alle Strings nicht größer als 50 zurückgegeben. Steht er auf 0 wird die ibaLogic String-Größe 1024 verwendet.

## 4.1 ODBC Verbindung zu Oracle

Für die ODBC Verbindung zu Oracle sind folgende Einstellungen im Test gemacht worden. Dabei muss auf dem ibaLogic-Rechner auch ein Oracle-Client installiert sein, um den ODBC Treiber zur Verfügung zu haben.



**Oracle ODBC Driver Configuration**

Data Source Name: IBA  
 Description:   
 TNS Service Name: IBAORA  
 User ID: IBA

OK  
 Cancel  
 Help  
 Test Connection

Application | Oracle | Workarounds | SQLServer Migration

Fetch Buffer Size: 64000  
 Enable LOBs:   
 Enable Statement Caching:   
 Cache Buffer Size: 20

Failover  
 Enable Failover:   
 Retry: 10  
 Delay: 10

**Oracle ODBC Driver Configuration**

Data Source Name: IBA  
 Description:   
 TNS Service Name: IBAORA  
 User ID: IBA

OK  
 Cancel  
 Help  
 Test Connection

Application | Oracle | Workarounds | SQLServer Migration

Bind TIMESTAMP as DATE:   
 Force SQL\_WCHAR Support:   
 Disable Microsoft Transaction Server:   
 Set Metadata Id Default to SQL\_TRUE:

Disable SQLDescribeParam:   
 Bind NUMBER As FLOAT:   
 Disable RULE Hint:

**Oracle ODBC Driver Configuration**

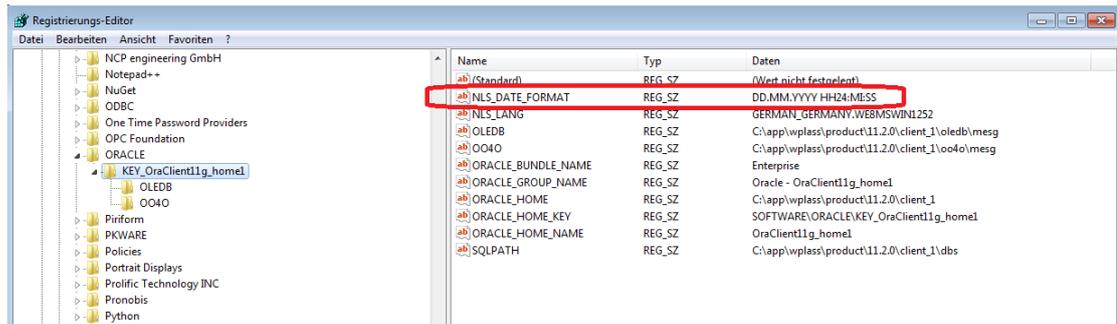
Data Source Name: IBA  
 Description:   
 TNS Service Name: IBAORA  
 User ID: IBA

OK  
 Cancel  
 Help  
 Test Connection

Application | Oracle | Workarounds | SQLServer Migration

Enable EXEC Syntax:  Schema:

Für das DATE-Format ist zu beachten, dass DD.MM.YYYY HH:MM:SS erwartet wird. Dieses Format muss eingestellt sein, z. B. in der Registry HKEY\_LOCAL\_MACHINE\SOFTWARE\ORACLE\KEY\_OraClient11g\_home1.



### Fehlermeldungen:

**INVALID NUMBER:** Dieser Fehler kann auftreten, wenn für Real-Werte ein Punkt angegeben ist, aber das System ein Komma erwartet. Dies liegt an den NLS-Settings von Oracle.

Abhilfe schaffen kann eventuell ein Eintrag im REGEDIT unter HKEY\_LOCAL\_MACHINE—ORACLE—KEY\_OraClient11g\_home1 (oder ähnlich).

## 5 Anhang

### 5.1 ODBC Datentypen (MSSQL/ORACLE)

MS SQL DATENTYPEN					
Name	MSSQL Datentyp	Code	Length	ibaLogic Typ	Bemerkung
bit	BIT	-7	1	BYTE	NULL = 0
tinyint	TINYINT	-6	1	USINT	NULL = 0
smallint	SMALLINT	5	2	INT	NULL = 0
int	INT	4	4	DINT	NULL = 0
bigint	BIGINT	-5	19	STRING(19)	NULL = 'NULL'
smallmoney	SMALLMONEY	3	8	LREAL	NULL = NaN.0
money	MONEY	3	8	LREAL	NULL = NaN.0
numeric	NUMERIC(18,0)	2	8	LREAL	NULL = NaN.0
decimal	DECIMAL(18,0)	3	8	LREAL	NULL = NaN.0
real	REAL	7	8	LREAL	NULL = NaN.0
float	FLOAT	6	8	LREAL	NULL = NaN.0
smalldatetime	SMALLDATETIME	11	16	STRUCT(INT,INT,INT,INT,INT,INT,DINT)	NULL = 0, 'Rückgabe-Struktur 'Year/Month/day/hour/minute/second/fraction'
datetime	DATETIME	11	16	STRUCT(INT,INT,INT,INT,INT,INT,DINT)	NULL = 0, 'Rückgabe-Struktur 'Year/Month/day/hour/minute/second/fraction'
datetime2	DATETIME2	-9	27	STRING(27)	NULL = 'NULL'
datetimeoffset	DATETIMEOFFSET	-9	34	STRING(34)	NULL = 'NULL'
date	DATE	-9	10	STRING(10)	NULL = 'NULL'
time	TIME7	-9	16	STRING(16)	NULL = 'NULL'
char	CHAR(10)	1	10	STRING (10)	NULL = 'NULL'
varchar	VARCHAR(len)	12	x	STRING(x)	NULL = 'NULL'
varchar(max)	VARCHAR(MAX)	-10	1024	STRING	NULL = 'NULL', SELECT mit CAST und CONVERT <sup>1</sup>
text	TEXT	-8	1024	STRING	NULL = 'NULL', SELECT mit CAST und CONVERT <sup>1</sup>
nchar	NCHAR(10)	-8	10	STRING(10)	NULL = 'NULL'
nvarchar	NVARCHAR(50)	-9	50	STRING(50)	NULL = 'NULL'
nvarchar(max)	NVARCHAR(MAX)	-10	1024	STRING	NULL = ''
ntext	NTEXT	-10	1024	STRING	NULL = ''
binary	BINARY(50)	-2	50	BYTE(50)	NULL = 0
varbinary	VARBINARY(50)	-3	50	BYTE(50)	NULL = 0
varbinary(max)	VARBINARY(MAX)	x	x	nicht unterstützt	kein SELECT
image	IMAGE	x	x	nicht unterstützt	kein SELECT

<sup>1</sup> SELECT CAST(Spaltenname AS CHAR(1024)) FROM Tabellename  
 SELECT CONVERT(CHAR(1024), Spaltenname) FROM Tabellename  
 Die Zielgröße von 1024 ist hier willkürlich gewählt. Es geht auch jede andere Größe, solange sie in den Zieldatentyp passt.

MS SQL DATENTYPEN					
Name	MSSQL Datentyp	Code	Length	ibaLogic Typ	Bemerkung
sqlvariant	SQL_VARIANT	-150	1024	STRING	NULL = 'NULL'
xml	XML	-10	128	nicht unterstützt	
uniqueidentifier	UNIQUEIDENTIFIER	-11	16	nicht unterstützt	
timestamp	TIMESTAMP	-2	8	nicht unterstützt	wird selber versorgt von DB
geography				nicht unterstützt	
geometry				nicht unterstützt	
hierarchyid				nicht unterstützt	
cursor				nicht unterstützt	
table				nicht unterstützt	

ORACLE DATENTYPEN					
Name	Oracle Datentyp	Code	Length	ibaLogic Typ	Bemerkung
bfile	BFILE	?	?	nicht unterstützt	kein SELECT
binary_double	BINARY_DOUBLE	8	8	LREAL	NULL = NaN.0
binary_float	BINARY_FLOAT	7	8	LREAL	NULL = NaN.0
blob	BLOB	12	1024	STRING	NULL = 'NULL'
character	CHAR(1 BYTE)	1	1	STRING(1)	NULL = 'N'
clob	CLOB	12	1024	STRING	NULL = 'NULL'
date	DATE	12	19	STRING TO_CHAR	NULL = 'NULL'
real	FLOAT	6	8	LREAL	NULL = NaN.0
float	FLOAT	6	8	LREAL	NULL = NaN.0
double_precision	FLOAT	6	8	LREAL	NULL = NaN.0
interval_day	INTERVAL DAY(2) TO SECOND(6) !!TO_CHAR!	12	20	STRING(20)	NULL = 'NULL'
interval_year	INTERVAL YEAR(2) TO MONTH !!TO_CHAR	12	7	STRING(7)	NULL = 'NULL'
nchar	NCHAR(1 CHAR)	-8	1	STRING(1)	NULL = 'N'
national_char	NCHAR(1 CHAR)	-8	1	STRING(1)	NULL = 'N'
number	NUMBER	6	8	LREAL	NULL = NaN.0
int	NUMBER(38,0)	3	8	LREAL	NULL = NaN.0
integer	NUMBER(38,0)	3	8	LREAL	NULL = NaN.0
numeric	NUMBER(38,0)	3	8	LREAL	NULL = NaN.0
dec	NUMBER(38,0)	3	8	LREAL	NULL = NaN.0
decimal	NUMBER(38,0)	3	8	LREAL	NULL = NaN.0
nchar_varying	NVARCHAR2(20 CHAR)	-9	20	STRING(20)	NULL = 'NULL'
national_char_varying	NVARCHAR2(20 CHAR)	-9	20	STRING(20)	NULL = 'NULL'
nvarchar2	NVARCHAR2(20 CHAR)	-9	20	STRING(20)	NULL = 'NULL'
raw	RAW	12	1024	nicht unterstützt	kein SELECT

ORACLE DATENTYPEN					
Name	Oracle Datentyp	Code	Length	ibaLogic Typ	Bemerkung
rowid	ROWID	12	18	nicht unterstützt	kein SELECT
timestamp	TIMESTAMP(6)	12	19	STRING TO_CHAR	NULL = 'NULL'
urowid	UROWID	12	18	nicht unterstützt	kein SELECT
character_varying	VARCHAR2(20 BYTE)	12	20	STRING(20)	NULL = 'NULL'
varchar2 (20)	VARCHAR2(20 BYTE)	12	20	STRING(20)	NULL = 'NULL'
char_varying	VARCHAR2(20 BYTE)	12	20	STRING(20)	NULL = 'NULL'
long_raw				nicht unterstützt	
nclob				nicht unterstützt	
smallint				nicht unterstützt	

## 6 Support und Kontakt

### Support

Tel.: +49 911 97282-14  
Fax: +49 911 97282-33  
E-Mail: support@iba-ag.com



---

### Hinweis

Wenn Sie Support benötigen, dann geben Sie die Seriennummer (iba-S/N) des Produktes an.

---

### Kontakt

#### Zentrale

iba AG  
Königswarterstraße 44  
90762 Fürth  
Deutschland

Tel.: +49 911 97282-0  
Fax: +49 911 97282-33  
E-Mail: iba@iba-ag.com  
Kontakt: Harald Opel

#### Regional und weltweit

Weitere Kontaktadressen unserer regionalen Niederlassungen oder Vertretungen finden Sie auf unserer Webseite

**[www.iba-ag.com](http://www.iba-ag.com)**.