



# ibaPDA-Interface-Oracle

## Schreib-/Lesezugriff auf Datenbanken

Handbuch  
Ausgabe 1.4

Messsysteme für Industrie und Energie  
[www.iba-ag.com](http://www.iba-ag.com)

---

## Hersteller

iba AG  
Gebhardtstrasse 10-20  
90762 Fürth  
Deutschland

## Kontakte

Zentrale +49 911 97282-0  
Support +49 911 97282-14  
Technik +49 911 97282-13  
E-Mail [iba@iba-ag.com](mailto:iba@iba-ag.com)  
Web [www.iba-ag.com](http://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 2026, 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	Autor	Version SW
1.4	04-2026	Standard-Verbindungskonfiguration	rm, mm	8.4.0

Windows® und SQL Server® sind Marken und eingetragene Warenzeichen der Microsoft Corporation.

Oracle® und MySQL® sind Marken von Oracle und/oder seinen Partnern.

Andere in diesem Handbuch erwähnte Produkt- und Firmennamen können Marken oder Handelsnamen der jeweiligen Eigentümer sein.

## Inhalt

<b>1</b>	<b>Zu dieser Dokumentation .....</b>	<b>5</b>
1.1	Zielgruppe und Vorkenntnisse .....	5
1.2	Schreibweisen.....	6
1.3	Verwendete Symbole.....	7
<b>2</b>	<b>Systemvoraussetzungen .....</b>	<b>8</b>
<b>3</b>	<b>Schnittstelle SQL-Datenbank .....</b>	<b>9</b>
3.1	Allgemeine Informationen.....	9
3.2	Vergleich zu den DB/Cloud-Datenaufzeichnungen .....	10
3.3	Systemtopologien .....	10
3.4	Grundsätzliches Vorgehen .....	11
<b>4</b>	<b>Konfiguration und Projektierung ibaPDA .....</b>	<b>12</b>
4.1	Einstellungen der Schnittstelle .....	12
4.2	Datenbankverbindung erstellen .....	15
4.2.1	Oracle DB-Verbindung .....	16
4.2.1.1	Standard-Verbindungskonfiguration.....	18
4.2.1.2	Erweiterte Verbindungskonfiguration .....	20
4.3	SQL-Module .....	21
4.3.1	SQL-Modul hinzufügen .....	21
4.3.2	Allgemeine Moduleinstellungen.....	22
4.3.3	Puffer .....	25
4.4	SQL-Anweisungen.....	29
4.4.1	SQL-Anweisungen testen.....	30
4.4.2	SQL-Abfragen mit Eingangssignalen verknüpfen .....	31
4.4.3	Parameter verwenden .....	34
<b>5</b>	<b>Diagnose .....</b>	<b>36</b>
5.1	Lizenz .....	36
5.2	Sichtbarkeit der Schnittstelle .....	36
5.3	Protokolldateien .....	37
5.4	Verbindungsdiagnose mittels PING .....	38
5.5	Diagnosemodule.....	39

**6 Support und Kontakt.....45**

# 1 Zu dieser Dokumentation

Diese Dokumentation beschreibt die Funktion und Anwendung der Software-Schnittstelle *ibaPDA-Interface-Oracle*.

---

## Andere Dokumentation



Diese Dokumentation ist eine Ergänzung zum *ibaPDA*-Handbuch. Informationen über alle weiteren Eigenschaften und Funktionen von *ibaPDA* finden Sie im *ibaPDA*-Handbuch bzw. in der Online-Hilfe.

---

## 1.1 Zielgruppe und Vorkenntnisse

Diese Dokumentation richtet sich 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 der fachlichen Ausbildung, Kenntnisse und Erfahrungen sowie Kenntnis der einschlägigen Bestimmungen die übertragenen Arbeiten beurteilen und mögliche Gefahren erkennen kann.

Im Besonderen wendet sich diese Dokumentation an Personen, die mit Projektierung, Test, Inbetriebnahme oder Instandhaltung der unterstützten Datenbanken, Cloud- bzw. Cluster-Storage-Technologie befasst sind. Für den Umgang mit *ibaPDA-Interface-Oracle* sind folgende Vorkenntnisse erforderlich bzw. hilfreich:

- Grundkenntnisse *ibaPDA*
- Grundkenntnisse Datenbanken, Cloud- bzw. Cluster-Storage-Technologie
- Anwendung von SQL-Anweisungen

## 1.2 Schreibweisen

In dieser Dokumentation werden folgende Schreibweisen verwendet:

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

## 1.3 Verwendete Symbole

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

---

### Gefahr!



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

Beachten Sie die angegebenen Maßnahmen.

---

### Warnung!



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

Beachten Sie die angegebenen Maßnahmen.

---

### Vorsicht!



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

Beachten Sie die angegebenen Maßnahmen.

---

### Hinweis



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

---

### 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.

---

## 2 Systemvoraussetzungen

Folgende Systemvoraussetzungen sind für die Verwendung der SQL-Schnittstelle zu einer Datenbank vom Typ Oracle erforderlich:

- *ibaPDA* v8.0.0 oder höher
- Lizenz für *ibaPDA-Interface-Oracle*

In der *ibaPDA*-Dokumentation finden Sie weitere Anforderungen an die Computer-Hardware und die unterstützten Betriebssysteme.

### Lizenzinformation

Bestell-Nr.	Produktbezeichnung	Beschreibung
31.001113	ibaPDA-Interface-Oracle	Erweiterungslizenz für ein <i>ibaPDA</i> -System für das Lesen aus und Schreiben in Datenbanken des genannten Typs. Lizenz für 8 SQL-Module entsprechend 8 SQL-Anweisungen.
31.101113	one-step-up-Interface-Oracle	Erweiterungslizenz für weitere 8 SQL-Module. Max. 3 zulässig (32 SQL-Module insgesamt)

## 3 Schnittstelle SQL-Datenbank

### 3.1 Allgemeine Informationen

Die SQL-Schnittstelle von *ibaPDA* bildet die Basis für eine Reihe von Datenbankschnittstellen, über die *ibaPDA* Daten sowohl aus Datenbanken lesen als auch in Datenbanken schreiben kann.

Über die SQL-Schnittstelle konfigurieren und Verwalten Sie die Verbindungen zu einer oder mehreren Datenbanken. Pro Datenbank können Sie eine oder mehrere Verbindungen konfigurieren. Sie können alle DB-Verbindungen gleichzeitig nutzen. Die Datenbanken können vom gleichen oder von unterschiedlichen Systemtypen sein.

Unterstützt werden Datenbanksysteme unterschiedlicher Hersteller:

- Maria DB
- MySQL
- Oracle
- PostgreSQL
- SAP Hana<sup>1)</sup>
- SQL Server

Der Datenaustausch mit den Datenbanken erfolgt über entsprechende SQL-Module, die Datenbanksystem-spezifisch konfiguriert werden.

Der Zugriff auf die Daten wird mit SQL-Anweisungen realisiert.

- Eingangsrichtung (lesen)
  - Für den lesenden Zugriff werden benutzerdefinierte SQL-Abfragen (Query) wie z. B. SELECT verwendet und deren Ergebnisse auf Eingangssignale in *ibaPDA* abgebildet.
  - In den Abfragen können Sie optional Parameter einsetzen, die Signalen zugeordnet sind, um somit die Abfragen dynamisch während der laufenden Erfassung zu ändern.
- Ausgangsrichtung (schreiben)
  - Für den schreibenden Zugriff werden SQL-Befehle (Command) wie z. B. UPDATE, INSERT usw. verwendet, um Daten aus *ibaPDA* in die Datenbank zu schreiben.
  - In den Befehlen können Sie optional Parameter einsetzen, die Signalen zugeordnet sind, um somit Signal-Istwerte zu schreiben.

Abfragen und Befehle können Sie zyklisch oder getriggert ausführen.

<sup>1)</sup> DB-Client-Installation ".NET data provider" auf *ibaPDA*-Server erforderlich

### 3.2 Vergleich zu den DB/Cloud-Datenaufzeichnungen

Die SQL-Schnittstelle dient hauptsächlich dazu, neben den schnellen Messwerten, die über andere Schnittstellen erfasst werden, zusätzliche Daten zu einem Prozess, einer Anlage oder einem Produkt zu erfassen bzw. Istwerte, Betriebszustände oder berechnete Werte an eine Datenbank auszugeben. Die Zugriffe können auf Anforderung oder zyklisch erfolgen, wobei Sie stets die Reaktionszeit der Datenbank berücksichtigen müssen.

Für eine schnellere, kontinuierliche Datenschreibung in Datenbanken oder Cloud-Systeme sind die entsprechenden Datenaufzeichnungen (z. B. *ibaPDA-Data-Store-SQL-Server*, *ibaPDA-Data-Store-Oracle* usw.) zu verwenden.

SQL-Schnittstelle und DB-Datenaufzeichnungen unterstützen die gleichen Datenbanksysteme, sodass Sie die Wahl haben.

SQL-Schnittstelle	DB/Cloud-Datenaufzeichnung
<ul style="list-style-type: none"> <li>■ Konfiguration im I/O-Manager (lesen/schreiben)</li> </ul>	<ul style="list-style-type: none"> <li>■ Konfiguration bei Datenaufzeichnungen (nur schreiben)</li> </ul>
<ul style="list-style-type: none"> <li>■ Zugriff auf jede vorhandene, benutzerdefinierte Tabelle/View möglich</li> <li>■ Tabellen müssen vom Benutzer bereitgestellt werden, z. B. mit SQL-Anweisungen</li> </ul>	<ul style="list-style-type: none"> <li>■ Feste Tabellenstruktur, vorgegeben durch das Datenaufzeichnungsprofil</li> <li>■ Tabelle wird automatisch von <i>ibaPDA</i> erzeugt.</li> </ul>
<ul style="list-style-type: none"> <li>■ Zyklisches/getriggertes Lesen mit benutzerdefinierten SQL-Abfragen</li> <li>■ Zyklisches/getriggertes Einfügen oder Aktualisieren mit benutzerdefinierten SQL-Befehlen</li> </ul>	<ul style="list-style-type: none"> <li>■ Kontinuierliches/getriggertes Schreiben von Signalverläufen als Zeitreihendaten</li> </ul>
	<ul style="list-style-type: none"> <li>■ Lesen der Signalverläufe aus der Datenbank mit SQL-Trendabfragen in <i>ibaAnalyzer</i></li> <li>■ Lesen der Signalverläufe aus der Datenbank mit <i>ibaDaVIS</i></li> </ul>

Tab. 1: Unterschiede zwischen SQL-Schnittstelle und DB/Cloud-Datenaufzeichnung

### 3.3 Systemtopologien

Die Verbindungen zwischen Datenbanken und *ibaPDA* können über Standard-Ethernet-Schnittstellen des Rechners hergestellt werden.

**Hinweis**



Wenn hochzyklische Zugriffe oder sehr große Datenmengen zu erwarten sind, wird empfohlen, die TCP/IP-Kommunikation auf einem separaten Netzwerksegment durchzuführen, um eine gegenseitige Beeinflussung durch sonstige Netzwerkkomponenten, z. B. SPS-Systeme, auszuschließen.

### 3.4 Grundsätzliches Vorgehen

1. Prüfen Sie, ob für die geplante Datenbankverbindung zusätzliche Software und/oder Lizenzen des Datenbankherstellers erforderlich sind.
2. Installieren Sie ggf. die benötigte Software auf dem *ibaPDA*-Rechner, z. B. für SAP HANA-Verbindungen die Software "ADO.NET Data Provider".
3. Öffnen Sie in *ibaPDA* den I/O-Manager, Register *Eingänge*, und konfigurieren Sie auf der SQL-Schnittstelle eine oder mehrere Datenbankverbindungen.  
Siehe ↗ *Datenbankverbindung erstellen, Seite 15*.

#### Für Datenbankabfragen (Lesezugriff)

1. Fügen Sie unter der SQL-Schnittstelle im Register *Eingänge* ein Modul *SQL-Abfrage* hinzu und wählen Sie für das Modul eine der zuvor konfigurierten Datenbankverbindungen aus.
2. Formulieren Sie eine SQL-Anweisung zur Datenbankabfrage mit oder ohne Parameter.
3. Wenn Sie Parameter verwenden: Weisen Sie den Parametern die entsprechenden analogen und/oder digitalen Signale zu, deren Werte Sie für die Abfrage verwenden wollen.
4. Konfigurieren Sie die analogen und/oder digitalen Eingangssignale.

#### Für Datenbankbefehle (Schreibzugriff)

1. Fügen Sie unter der SQL-Schnittstelle im Register *Ausgänge* ein Modul *SQL-Befehl* hinzu und wählen Sie für das Modul eine der zuvor konfigurierten Datenbankverbindungen aus.
2. Formulieren Sie eine SQL-Anweisung als SQL-Befehl mit oder ohne Parameter.
3. Wenn Sie Parameter verwenden: Weisen Sie den Parametern die entsprechenden analogen und/oder digitalen Signale zu, deren Werte Sie in die Datenbank schreiben wollen.

---

#### Vorsicht



**Durch hochzyklische DB-Zugriffe und/oder sehr große Datenmengen können Netzwerk und DB überlastet werden. In den SQL-Anweisungen können beliebige SQL-Befehle verwendet und damit auch Schaden angerichtet werden (z. B. "delete table", "drop database" usw.)**

Darum...

- Lassen Sie Erstellung/Änderung von SQL-Anweisungen nur von qualifizierten Anwendern vornehmen (→ Benutzerverwaltung, Security-Administration).
- Binden Sie Ihren DB-Administrator bei Fragen zur Konfiguration der SQL-Anweisungen mit ein. So sollte z. B. der DB-Benutzer, der in *ibaPDA* für die DB-Verbindung verwendet wird, nur die Rechte haben, die tatsächlich erforderlich sind.
- Stellen Sie eine angemessene Aktualisierungszeit ein (so klein wie nötig, so groß wie möglich).
- Schätzen Sie die Ergebnisse Ihrer SQL-Anweisung ab und testen Sie diese.

## 4 Konfiguration und Projektierung ibaPDA

Nachfolgend ist die Projektierung in *ibaPDA* beschrieben. Wenn alle Systemvoraussetzungen erfüllt sind, bietet *ibaPDA* im Schnittstellenbaum des I/O-Managers die Schnittstelle *SQL-Datenbank* an.

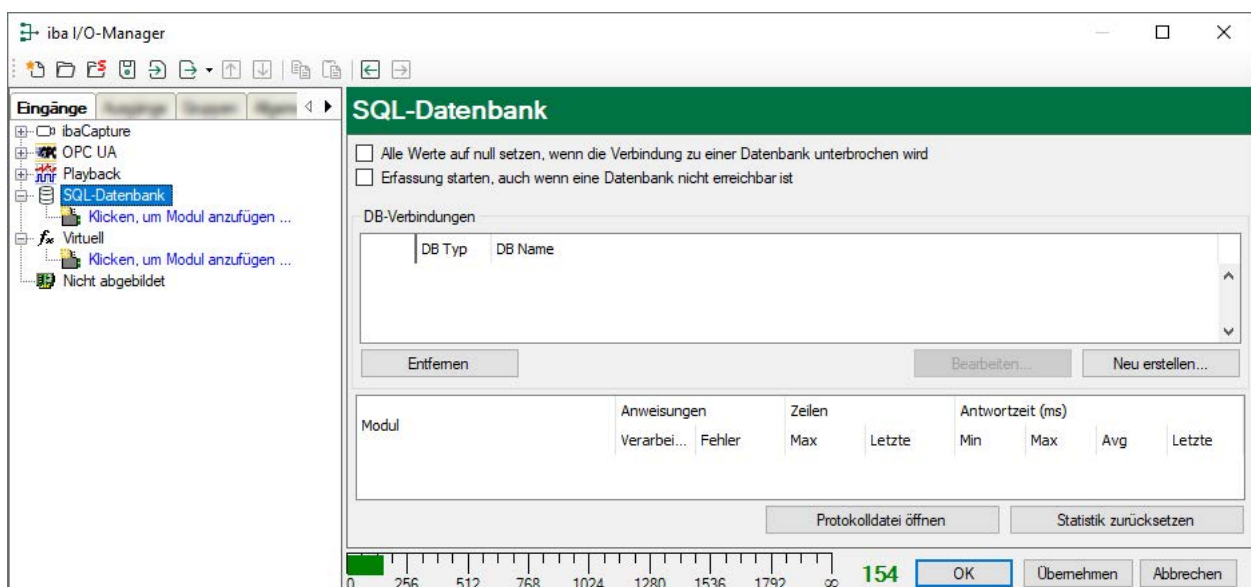
### Hinweis



Die SQL-Schnittstelle ist sichtbar, sobald eine der SQL-Interface-Lizenzen (SQL Server, Oracle, PostgreSQL, MySQL, SAP HANA) im Lizenzcontainer (Dongle oder Soft-Lizenz) vorhanden ist. Auch wenn nur ein Datenbanksystem lizenziert ist, erscheinen in der SQL-Schnittstellenkonfiguration alle unterstützten DB-Systeme. Für die Datenerfassung können Sie trotzdem nur das lizenzierte System verwenden.

### 4.1 Einstellungen der Schnittstelle

Die Schnittstelle hat folgende Funktionen und Konfigurationsmöglichkeiten.



#### Alle Werte auf null setzen, wenn die Verbindung zu einer Datenbank unterbrochen wird

Wenn diese Option aktiviert ist, werden alle Signalwerte in den Modulen, die einer unterbrochenen Datenbankverbindung zugeordnet sind, auf null gesetzt. *ibaPDA* versucht in regelmäßigen Abständen die Datenbankverbindung wiederherzustellen, bis dies entweder gelingt oder die Erfassung angehalten wird.

Wenn diese Option deaktiviert ist, dann behält *ibaPDA* den bei Verbindungsabbruch letzten gültigen Messwert im Speicher.

### Hinweis



Es kann einige Zeit dauern, bis die Signalwerte auf null gesetzt werden. Eine längere Reaktionszeit einer Datenbank, selbst im Bereich mehrerer Sekunden, ist nicht zwangsläufig ein Indiz für eine unterbrochenen Verbindung.

### Erfassung starten, auch wenn eine Datenbank nicht erreicht wird

Wenn diese Option aktiviert ist, startet die Erfassung auch dann, wenn eine oder mehrere der konfigurierten Datenbanken nicht erreichbar sind. Anstatt eines Fehlers wird im Prüfungsdialog eine Warnung ausgegeben. Wenn das System ohne Verbindung zu einer Datenbank gestartet wurde, dann versucht *ibaPDA* in regelmäßigen Abständen, eine Verbindung zur Datenbank herzustellen.

### DB-Verbindungen

Dieser Bereich zeigt die konfigurierten Datenbankverbindungen übersichtlich an. Jede Datenbankverbindung wird mit einer ID-Nummer, dem Datenbanksystem und der verbundenen Datenbank eindeutig identifiziert. Mit den Buttons können Sie Datenbankverbindungen erstellen, bearbeiten und entfernen. Die Konfiguration der Datenbankverbindungen ist in den nächsten Kapiteln beschrieben.

#### <Protokolldatei öffnen>

Wenn Verbindungen zu Steuerungen hergestellt wurden, dann werden alle verbindungsspezifischen Aktionen in einer Textdatei protokolliert. Diese Datei können Sie über diesen Button öffnen und einsehen. Im Dateisystem auf der Festplatte finden Sie die Protokolldatei zu dieser Schnittstelle im Pfad `...\ProgramData\iba\ibaPDA\Log`.

Der Dateiname der aktuellen Protokolldatei lautet `SchnittstelleLog.txt`, der Name der archivierten Protokolldateien lautet `SchnittstelleLog_yyyy_mm_dd_hh_mm_ss.txt`.

#### <Statistik zurücksetzen>

Über diesen Button können Sie die berechneten Zeitwerte und den Fehlerzähler in der Tabelle auf 0 setzen.

### Tabelle mit Verbindungsstatistik

Im unteren Bereich des Dialogs befindet sich eine Tabelle zum Zweck der Verbindungsdiagnose. Für jedes konfigurierte SQL-Modul werden statistische Informationen wie Zählerwerte und Zeiten angezeigt.

#### Hinweis



Aufgrund der Eigenart von relationalen Datenbanken können Fehlerzähler langsamer inkrementieren als Erfolgszähler. Beachten Sie dies bei der Diagnose von Verbindungsproblemen.

Spalte	Bedeutung
Modul	Name des SQL-Moduls (Abfragemodule und Befehlsmodule)
Anweisungen	Verarbeitet: Anzahl der ausgeführten SQL-Anweisungen Fehler: Anzahl der SQL-Anweisungen, bei denen Fehler aufgetreten sind. Entweder sind SQL-Anweisungen fehlgeschlagen (Datenbank antwortet mit Fehler) oder Anweisungen konnten nicht ausgeführt werden, weil die Datenbank nicht erreichbar war und der Puffer vollgelaufen ist. Ein leeres Ergebnis zu einer Anweisung wird nicht als Fehler betrachtet.

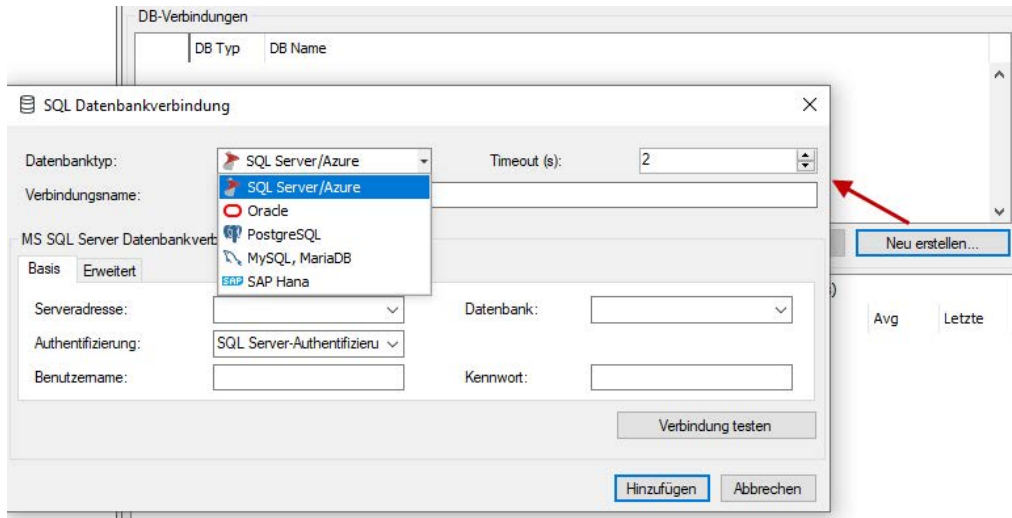
Spalte	Bedeutung
Zeilen	Max: Die höchste Anzahl Zeilen, die seit dem letzten Start der Erfassung oder dem letzten Rücksetzen der Statistik nach einer Anweisung zurückgegeben wurden. Letzte: Die Anzahl Zeilen, die nach der letzten Anweisung zurückgegeben wurden.
Antwortzeit	Zeitwerte für die Dauer zwischen Ausführung der SQL-Anweisung und Bestätigung der Ausführung durch die Datenbank. Min, Max, Avg: Diese Werte beziehen sich auf die ausgeführten Anweisungen seit dem Start der Erfassung bzw. dem letzten Rücksetzen der Zähler. Letzte: Antwortzeit der zuletzt ausgeführten Anweisung.

Tab. 2: Spalten der Verbindungsdiagnose

## 4.2 Datenbankverbindung erstellen

Bevor Sie Daten mit einer Datenbank austauschen können, müssen Sie zunächst eine Verbindung konfigurieren.

1. Wählen Sie im Schnittstellenbaum die Schnittstelle *SQL-Datenbank* aus und klicken Sie auf den Button <Neu erstellen>.



2. Wählen Sie das gewünschte Datenbanksystem aus.

Je nachdem, welchen Systemtyp Sie auswählen, verändert sich der untere Teil des Dialogs.

3. Geben Sie dann im unteren Teil des Dialogs die Informationen zur Datenbank an oder wählen Sie diese aus (z. B. DB-Server, Datenbankname, Authentifizierungsmethode, ggf. Benutzername und Kennwort).

Weitere Details dazu finden Sie im nächsten Kapitel.

Mit Angabe dieser Informationen wird automatisch für diese Verbindung ein Name generiert und in dem Feld *Verbindungsname* angezeigt (*Benutzername@Datenbank*). Sie können den Namen überschreiben.

Im Fall von mehreren Verbindungen zum gleichen Systemtyp können Sie die Verbindung anhand dieses Namens unterscheiden. Außerdem wählen Sie die DB-Verbindung später bei der Modulkonfiguration anhand dieses Namens aus.

Der Verbindungsname wird in der Tabelle mit den DB-Verbindungen in der Spalte *DB Name* angezeigt.

Einmal erstellte DB-Verbindungen können Sie jederzeit bearbeiten, indem Sie die betreffende Verbindung in der Tabelle markieren und auf den Button <Bearbeiten> klicken.

Mit dem Button <Entfernen> entfernen Sie die markierte DB-Verbindung.

**Hinweis**

Wenn eine Datenbankverbindung entfernt werden soll, die bereits einem oder mehreren Modulen zugewiesen ist, erscheint ein Warnhinweis.

Da SQL-Module mit einer DB-Verbindung in der SQL-Schnittstelle verknüpft sind, müssen Sie darauf achten, dass die Module den Bezug zu der Verbindung nicht verlieren. Andernfalls laufen die SQL-Anweisungen ins Leere.

**4.2.1 Oracle DB-Verbindung**
**Datenbanktyp**

Wählen Sie in der Auswahlliste das Datenbanksystem aus, hier *Oracle*.

**Timeout**

Hier können Sie einen Wert für die Wartezeit (Timeout) in Sekunden bei Verbindungsaufbau einstellen. Ein Überschreiten der hier eingestellten Zeit führt dazu, dass *ibaPDA* den Verbindungsversuch abbricht und eine Fehlermeldung ausgibt.

Beachten Sie, dass nach dem Herstellen der Verbindung jeder an die Datenbank gesendete Befehl ein zusätzliches Timeout hat, das von der Anwendung, die ihn verwendet, eingestellt werden kann. Es gibt keine generische Konfiguration für den Befehls-Timeout.

**Verbindungsname**

Diese Zeile enthält den Namen für diese Datenbankverbindung. Der Name wird automatisch nach dem Schema *Benutzername@Datenbank* gebildet und eingetragen, sobald Sie die Felder darunter ausgefüllt haben. Sie können den Verbindungsnamen aber auch überschreiben.

Anhand dieses Namens können Sie die Datenbankverbindung in der Übersichtstabelle auf Schnittstellenebene identifizieren. Dort steht der Verbindungsname in der Spalte *DBName*.

**Oracle Datenbankverbindung**

Sie haben 2 Möglichkeiten, die Verbindung zur Oracle-Datenbank zu konfigurieren. Es wird empfohlen die Verbindung über das Register *TNS* herzustellen, siehe [↗ Standard-Verbindungskonfiguration, Seite 18](#). Die Datenbankverbindung über das Register *Basis* herzustellen wird für

DB-Administratoren und Entwickler empfohlen, siehe ↗ *Erweiterte Verbindungskonfiguration*, Seite 20.

In beiden Fällen müssen Sie sich authentifizieren:

**Benutzername/Kennwort**

Tragen Sie hier die nötigen Daten für die Anmeldung an der Datenbank ein. Erfragen Sie die korrekten Daten ggf. bei Ihrem DB-Administrator.

Diese Felder sind nur verfügbar, wenn Sie die Option *OS-Authentifizierung* deaktiviert haben.

**OS-Authentifizierung verwenden**

Wenn Sie diese Option aktivieren, wird das Benutzerkonto verwendet, unter dem der *ibaPDA*-Server läuft. Dies ist normalerweise das Systemkonto. Ob diese Methode die passende ist, erfragen Sie bitte bei Ihrem DB-Administrator.

**<Verbindung testen>**

*ibaPDA* prüft den Zugriff auf die gewählte Datenbank auf dem eingestellten DB-Server. Bei erfolgreicher Verbindung beinhaltet die Meldung die Version der Datenbank.

### 4.2.1.1 Standard-Verbindungskonfiguration

Es wird empfohlen die Verbindung über das Register *TNS* herzustellen.

Die Konfiguration mit TNS ermöglicht ebenfalls die Verwendung eines Failover-Clusters anstelle einer einzigen Server-Konfiguration für die Oracle-Datenbank. Wenden Sie sich zu diesem Thema an Ihren DB-Administrator.

#### TNS-Name oder Inhalt

Verwenden Sie für die TNS-Konfiguration eine der beiden nachfolgenden Möglichkeiten:

1. Geben Sie den Namen einer Verbindung ein, die in der Datei "tnsnames.ora" im Oracle-Client-Verzeichnis enthalten ist.
2. Geben Sie einen kompletten Abschnitt ein, wie er in der Datei "tnsnames.ora" enthalten ist.

Beispiel:

```
(DESCRIPTION= (ADDRESS_LIST= (ADDRESS= (PROTOCOL=TCP)
(HOST=192.168.99.101) (PORT=1521) ) ) (CONNECT_DATA= (SERVER=DEDICATED)
(SERVICE_NAME=xex)))
```

Geben Sie den kompletten Abschnitt in einer Zeile ein. Diese Eingabe hat den Vorteil, dass *ibaPDA* laufen kann, während ein Upgrade des Oracle-Servers durchgeführt wird. ODP.NET liest *tnsnames.ora* nur einmal. Wenn sich der Inhalt der Datei ändert, müssen Sie nur den geänderten Teil der Datei in die Oracle-Konfiguration im *ibaPDA*-Client einfügen und die neue Konfiguration übernehmen.

#### Pfad zu tnsnames.ora

Wenn bereits ein Oracle-Client auf dem Rechner installiert ist, kann die passende Datei *tnsnames.ora* gesucht werden. Zusätzlich können Sie den Pfad zu der Datei angeben oder über den Browser-Button auswählen.

Falls die Datei nicht vorhanden ist, muss eine passende Datei angelegt werden bzw. vom DB-Administrator zur Verfügung gestellt werden. Speichern Sie die Datei auf Ihrem Rechner und geben Sie den Pfad dorthin an.

Wenn Sie in der ersten Zeile (*TNS-Name oder Inhalt*) den kompletten Inhalt (content) der Datei *tnsnames.ora* angeben, dann kann das Feld *Pfad* leer bleiben. Geben Sie nur den TNS-Namen an, müssen Sie den Pfad auf die komplette Datei angeben.

Den Pfad zu *tnsnames.ora* können Sie über den Zahnrad-Button als Standard festlegen.

#### Automatische Suche

Die Suche erfolgt zuerst im Registry-Key TNS\_ADMIN. Dort können Sie als Administrator den Verweis auf *tnsnames.ora* eintragen.

Wenn die Datei dort gefunden wird, dann wird im Standard-Verzeichnis für einen Oracle-Client gesucht. Wenn die Datei auch dort gefunden wird, dann wird im *ibaPDA*-Server-Verzeichnis gesucht.

Wenn `tnsnames.ora` gefunden wird, dann wird der Pfad auch im Feld *Pfad* angezeigt.

Wenn Sie den Pfad manuell überschreiben, dann gilt der eingetragene Pfad.

Wird über die automatische Suche keine Datei `tnsnames.ora` gefunden, dann erscheint eine Fehlermeldung mit einem Dialog, in dem Sie die Datei auswählen können.

### 4.2.1.2 Erweiterte Verbindungskonfiguration

Basis	TNS		
Serveradresse:	<input type="text"/>	Port:	<input type="text" value="1521"/>
Dienstname/SID:	<input type="text"/>	<input type="checkbox"/>	SID statt Dienstname verwenden

Die Datenbankverbindung über das Register *Basis* herzustellen wird für DB-Administratoren empfohlen. Ein Failover-Cluster können Sie mit diesen Einstellungen nicht konfigurieren. Nutzen Sie dafür das Register *TNS*, siehe [↗ Standard-Verbindungskonfiguration, Seite 18](#).

#### Serveradresse

Geben Sie die IP-Adresse oder den Hostnamen des Datenbankservers ein.

#### Port

Über diesen Port kommuniziert der DB-Server. Den Standard-Port 1521 können Sie normalerweise beibehalten. Wenn ein anderer Port erforderlich ist, können Sie diesen Port hier eintragen.

#### Dienstname/SID

Der Name des Oracle-Dienstes, wie er im Oracle-Server konfiguriert ist.

#### SID statt Dienstname verwenden

In einigen seltenen Fällen muss die SID verwendet werden. Lesen Sie in der Oracle-Dokumentation nach, worin der Unterschied zwischen SID und Service-Name besteht.

## 4.3 SQL-Module

Bei den SQL-Modulen unter der SQL-Schnittstelle gibt es Eingangsmodule und Ausgangsmodulen unterschieden.

Ein SQL-Modul ist also entweder ein lesendes Modul mit einer SQL-Abfrage oder ein schreibendes Modul mit einem SQL-Befehl. Daher werden im folgenden Text auch die Begriffe SQL-Abfragemodul und SQL-Befehlsmodul verwendet.

Ein Modul kann nicht beides gleichzeitig sein. Daher gibt es auch kein SQL-Modul, das im I/O-Manager in beiden Registern *Eingänge* und *Ausgänge* gleichzeitig zu sehen ist.

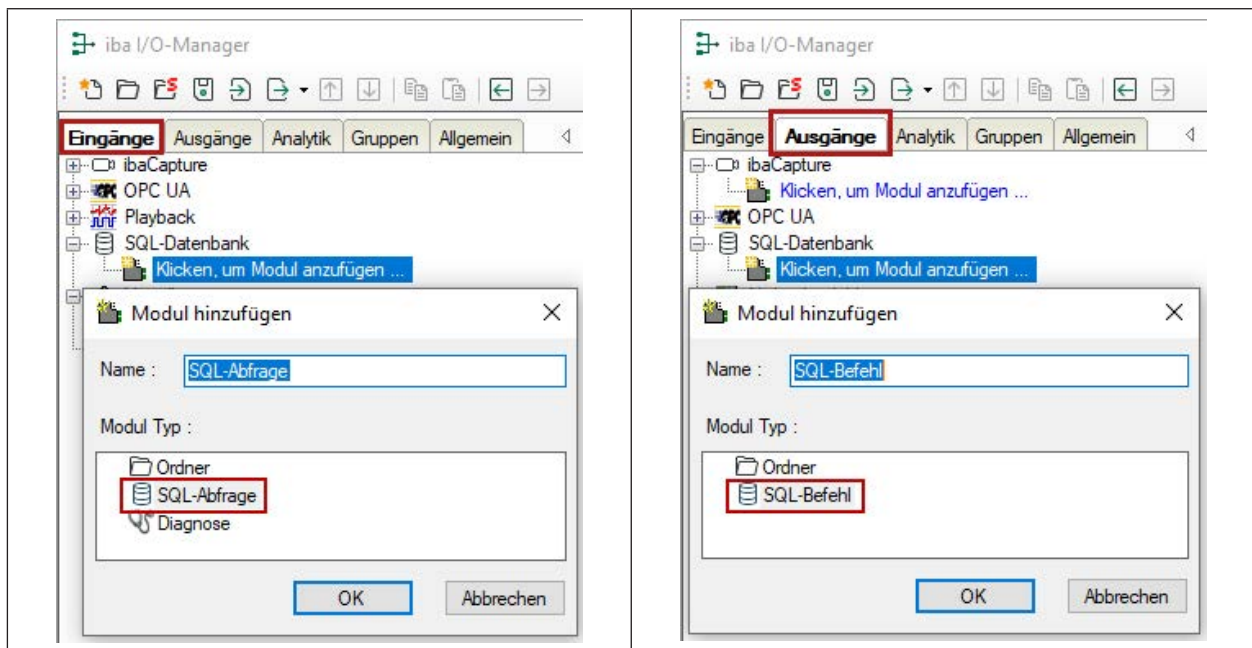
In ihrem Aufbau und den Bestandteilen sind die Module jedoch sehr ähnlich.

Für alle SQL-Module gilt:

- Die SQL-Module haben eine Referenz zu einer Datenbankverbindung in der SQL-Schnittstelle.
- Die SQL-Module enthalten eine SQL-Anweisung (Abfrage oder Befehl).
- Die SQL-Module bieten eine Zuweisungstabelle für in den SQL-Anweisungen verwendete Parameter.
- Die SQL-Module können zyklisch oder getriggert arbeiten.

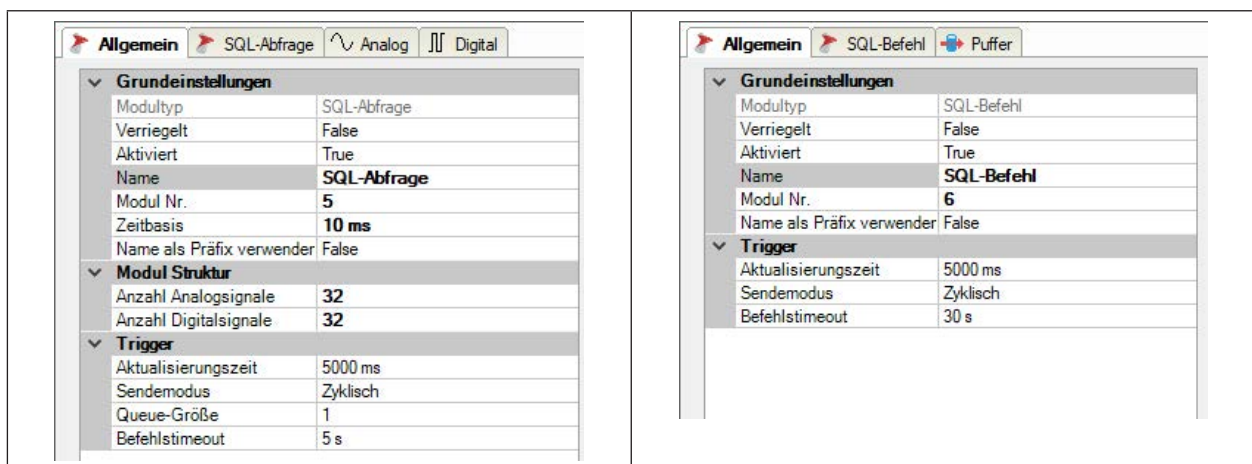
### 4.3.1 SQL-Modul hinzufügen

1. Klicken Sie auf den blauen Link *Klicken, um Modul anzufügen*, der sich unter jeder Datenschnittstelle im Register *Eingänge* oder *Ausgänge* befindet.
2. Wählen Sie im Dialogfenster den gewünschten Modultyp aus und vergeben Sie bei Bedarf einen Namen über das Eingabefeld.
3. Bestätigen Sie Ihre Auswahl mit <OK>.



Tab. 3: Hinzufügen eines SQL-Abfragemoduls (links) bzw. SQL-Befehlsmoduls (rechts)

### 4.3.2 Allgemeine Moduleinstellungen



Tab. 4: Allgemeine Moduleinstellungen für Eingangsmodule (li.) und Ausgangsmodule (re.)

#### Grundeinstellungen

##### Modultyp (nur Anzeige)

Zeigt den Typ des aktuellen Moduls an.

##### Verriegelt

Sie können ein Modul verriegeln, um ein versehentliches oder unautorisiertes Ändern der Einstellungen zu verhindern.

##### Aktiviert

Aktivieren Sie das Modul, um Signale aufzuzeichnen.

**Name**

Hier können Sie einen Namen für das Modul eintragen.

**Kommentar**

Hier können Sie einen Kommentar oder eine Beschreibung zum Modul eintragen. Dies wird dann als Tooltip im Signalbaum angezeigt.

**Modul Nr.**

Diese interne Referenznummer des Moduls bestimmt die Reihenfolge der Module im Signalbaum von *ibaPDA-Client* und *ibaAnalyzer*.

**Zeitbasis**

Alle Signale dieses Moduls werden mit dieser Zeitbasis erfasst.

**Modulname als Präfix verwenden**

Diese Option setzt den Modulnamen zusätzlich vor den Signalnamen.

**Modul Struktur (nur Eingangsmodule)****Anzahl Analogsignale/Anzahl Digitalsignale**

Stellen Sie die Anzahl der konfigurierbaren Analogsignale bzw. Digitalsignale in den Signaltabellen ein. Der Standardwert ist jeweils 32. Der Maximalwert beträgt 1000. Die Signaltabellen werden entsprechend angepasst.

**Trigger****Aktualisierungszeit**

Mit der Aktualisierungszeit in Millisekunden bestimmen Sie folgendes:

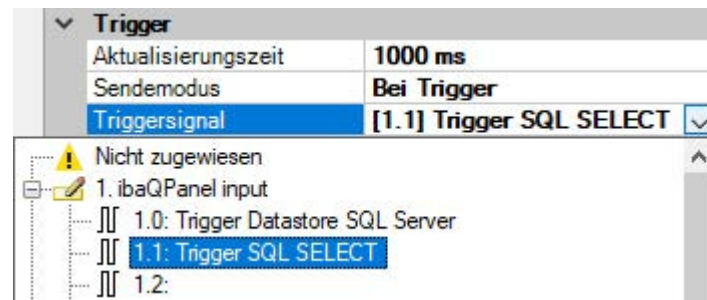
- Im Sendemodus *Zyklisch*: das Ausführungsintervall der SQL-Anweisungen
- Im Sendemodus *bei Änderung* und *bei Trigger*: die Totzeit nach Ausführung einer Anweisung

**Sendemodus**

Stellen Sie hier ein, wann die SQL-Anweisungen (Abfragen und Befehle) ausgeführt werden sollen.

- *Zyklisch*: Die SQL-Anweisung wird im Takt der Aktualisierungszeit ausgeführt.
- *Bei Änderung*: Die SQL-Anweisung wird bei jeder Änderung der Signaldaten ausgeführt. Bei Auswahl dieses Modus erscheint die Zeile *Triggersignal*. Nach einer Auslösung werden alle weiteren auslösenden Werteänderungen ignoriert, bis die Aktualisierungszeit abgelaufen ist. Erst dann kann wieder die SQL-Anweisung erneut ausgeführt werden.
- *Bei Trigger*: Bei Auswahl dieses Modus erscheint die Zeile *Triggersignal*. Nach einer Auslösung werden alle weiteren auslösenden Trigger ignoriert, bis die Aktualisierungszeit abgelaufen ist. Erst dann kann die SQL-Anweisung erneut ausgeführt werden.

Beispiel für eine DB-Abfrage in Abhängigkeit von einer QPanel-Eingabe:



### Triggersignal (nur für Sendemodus Bei Änderung oder Bei Trigger)

- Für Sendemodus *Bei Änderung*: Hier können Sie jedes analoge oder digitale Signal auswählen, das als Trigger dienen soll. Bei jeder Änderung des Signalwerts wird die SQL-Anweisung ausgeführt.
- Für Sendemodus *Bei Trigger*: Hier können Sie jedes Digitalsignal auswählen, das als Trigger dienen soll. Bei jeder steigenden Flanke des Signals wird die SQL-Anweisung ausgeführt.

### Queue-Größe

Die Queue (Warteschlange, Puffer) dient dazu, SQL-Anweisungen zu puffern, wenn folgende Situationen auftreten:

- Die Datenbankverbindung ist unterbrochen.
- Die Ausführung einer SQL-Anweisung dauert so lange, dass bereits wieder neue SQL-Anweisungen in *ibaPDA* getriggert werden, bevor die Ausführung abgeschlossen ist.

Mit der Queue-Größe bestimmen Sie, wie viele SQL-Anweisungen maximal zwischengespeichert werden. Mit Überschreitung dieser Anzahl werden die ältesten SQL-Anweisungen gelöscht.

- Bei Eingangsmodulen steht die Queue-Größe standardmäßig auf 1:  
Da die Ergebnisse einer SQL-Abfrage auf Signale gemappt werden, ist es in den meisten Fällen nicht sinnvoll Abfragen nachzuholen, die aufgrund einer unterbrochenen Verbindung nicht ausgeführt wurden. Die in der Queue gepufferten SQL-Abfragen würden dann in schneller Folge ausgeführt und die Ergebnisse auf die Signale übertragen. Mit der Queue-Größe 1 wird immer nur die letzte getriggerte SQL-Abfrage vorgehalten.
- Bei Ausgangsmodulen steht die Queue-Größe standardmäßig auf 1000:  
Über SQL-Befehle können beispielsweise Produktionsdaten geschrieben werden. Hier ist es wichtig, dass diese auch bei einem Ausfall der Verbindung gepuffert und nachgeholt werden.

### Befehlstimeout

Mit dem Befehlstimeout beschränken Sie die Zeit, die die Ausführung einer Anweisung maximal dauern darf. Wenn die Abfrage oder der Befehl nicht vollständig ausgeführt werden konnte, bevor der Befehlstimeout abgelaufen ist, dann wird die Ausführung abgebrochen. Die Standard-Einstellungen sind für SQL-Abfragen und SQL-Befehle unterschiedlich:

- SQL-Abfragen: 5 s
- SQL-Befehle: 30 s

### 4.3.3 Puffer

Für die Ausgangsrichtung der Schnittstelle steht ein Speicherpuffer und zusätzlich ein optional aktivierbarer Dateipuffer zur Verfügung.

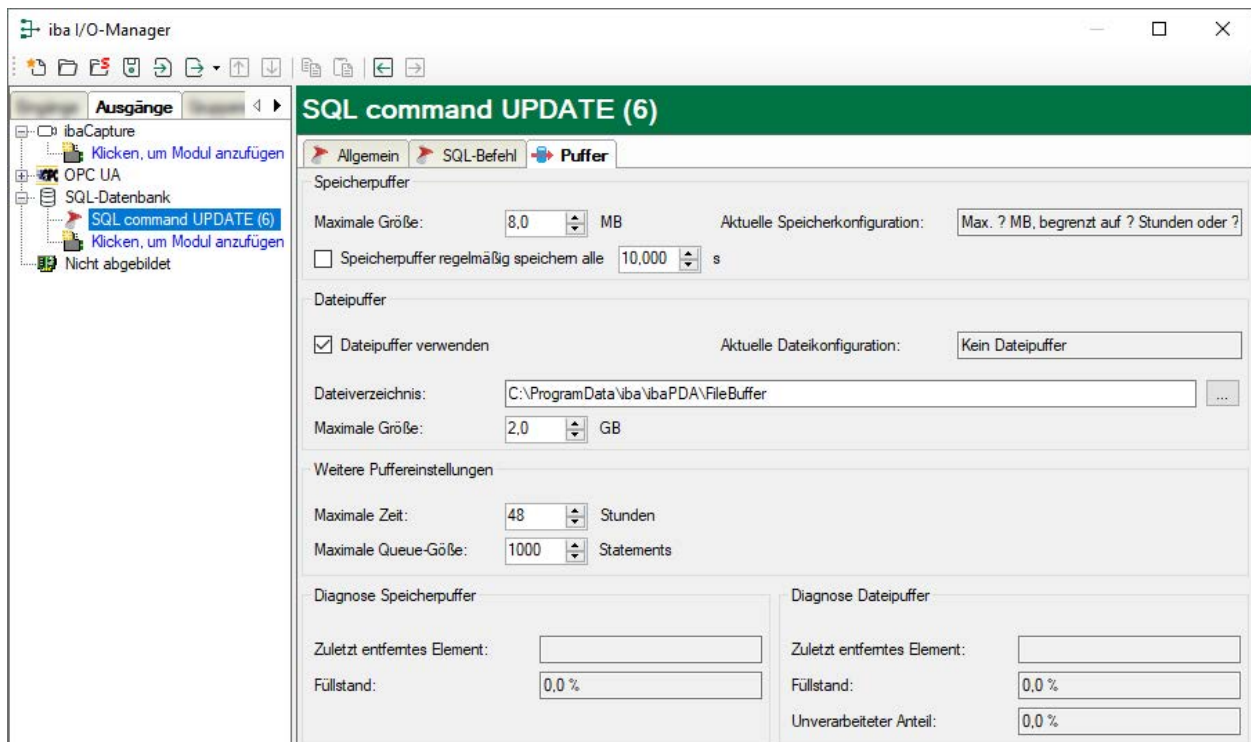
Daten, die an das Zielsystem gesendet werden sollen, durchlaufen immer den *ibaPDA*-internen Speicherpuffer. Bei bestehender Verbindung zum Zielsystem werden die Daten aus dem Speicherpuffer sofort dorthin gesendet. Geht die Verbindung verloren oder können die Daten nicht schnell genug verschickt werden, dann bleiben die Daten im Speicherpuffer erhalten. Der Speicherpuffer liegt im RAM des *ibaPDA*-Rechners, ist daher limitiert und flüchtig. Wird z. B. die Erfassung neu gestartet, gehen die gepufferten Daten verloren. Wächst der Speicherpuffer bei laufender Erfassung über die konfigurierte Größe hinaus an, werden die ältesten Werte gelöscht und gehen somit verloren.

Zur Verbesserung kann zusätzlich ein Dateipuffer aktiviert werden, der weitaus größere Datenmengen puffern kann. Die Daten werden dabei in Dateien in einem Verzeichnis eines lokalen Laufwerks des *ibaPDA*-Servers gespeichert. Bei aktiviertem Dateipuffer werden Daten aus dem überlaufenden Speicherpuffer in den Dateipuffer übertragen. Wird die Erfassung beendet oder neu gestartet (z. B. durch Anwenden einer geänderten IO-Konfiguration), werden Daten, die sich evtl. zu diesem Zeitpunkt im Speicherpuffer befinden, ebenfalls in den Dateipuffer übertragen.

Nach Wiederherstellung der Verbindung zum Zielsystem werden immer die ältesten Daten zuerst gesendet. Neuere Werte werden währenddessen dem Puffer hinzugefügt. Wenn beim Start der Erfassung noch gepufferte Daten im Dateipuffer vorhanden sind, werden diese genauso berücksichtigt und verarbeitet.

Für die SQL-Anweisungen bedeutet dies, dass die Anweisungen gem. der zum Zeitpunkt der Pufferung gültigen Konfiguration gepuffert werden. Wenn Sie während einer laufenden Pufferung die SQL-Anweisung ändern (im I/O-Manager ändern und übernehmen), dann werden nach Wiederherstellung der Verbindung trotzdem noch die alten Anweisungen ausgeführt, so wie sie zum Zeitpunkt der Pufferung erzeugt wurden.

Im Register *Puffer* bei den *Ausgängen* der jeweiligen SQL-Schnittstelle, nehmen Sie die Konfiguration der Pufferung vor (hier Beispiel SQL Server).



## Speicherpuffer

Der Speicherpuffer ist immer aktiviert. Sie können ihn nicht deaktivieren, da zu übertragende Daten immer den Puffer durchlaufen, bevor sie an das Zielsystem weitergegeben werden.

### Maximale Größe

Geben Sie hier die maximale Gesamtgröße für im Speicher gepufferte Elemente ein. Wenn die maximale Größe überschritten wird, gibt es 2 Möglichkeiten:

- Wenn die Dateipufferung deaktiviert ist, wird das älteste Element im Speicher gelöscht (und ist damit für immer verloren).
- Wenn die Dateipufferung aktiviert ist, wird der älteste Teil des Pufferspeichers in eine Pufferdatei verschoben.

### Speicherpuffer regelmäßig sichern alle ... s

Diese Option können Sie nur aktivieren, wenn der Dateipuffer aktiviert ist. Wenn die Option aktiviert ist, wird der gesamte Speicherpuffer periodisch in eine Pufferdatei ausgelagert.

Geben Sie eine Dauer ein, nach der der Speicherpuffer regelmäßig gesichert wird. Sie muss zwischen 10 s und 600 s liegen.

Mit dieser Option können Sie erreichen, dass möglichst wenig Daten bei einem Systemausfall verloren gehen.

### Aktuelle Speicherkonfiguration

Anzeige, welche Zeitspanne ungefähr mit den konfigurierten Einstellungen im Speicherpuffer zwischengespeichert werden kann. Angabe in d.hh:mm:ss.

## Dateipuffer

### Dateipuffer verwenden

Standardmäßig wird der Dateipuffer nicht verwendet. Hier können Sie die Dateipufferung aktivieren.

### Aktuelle Dateikonfiguration

Anzeige, welche Zeitspanne ungefähr mit den konfigurierten Einstellungen im Dateipuffer zwischengespeichert werden kann. Angabe in d.hh:mm:ss.

### Dateiverzeichnis

Im Feld *Dateiverzeichnis* können Sie einen Speicherort für die Dateien wählen. Sie können das Verzeichnis direkt in das Textfeld eingeben oder über den Browse-Button <...> auswählen. Das konfigurierte Dateiverzeichnis muss sich auf einer lokalen Festplatte des *ibaPDA*-Server-Rechners befinden.

Das gleiche Dateiverzeichnis können Sie für mehrere SQL-Befehlsmodule verwenden, da die Pufferdateien einen eindeutigen Namen haben. Dateien aus verschiedenen SQL-Befehlsmodulen können somit anhand ihres Namens unterschieden werden.

### Maximale Größe

Sie können die maximale Gesamtgröße der Pufferdateien eines SQL-Befehlsmoduls konfigurieren. Die Pufferdateien selbst haben die Dateierweiterung *.buf*, die Index-Datei zur Verwaltung der Pufferdateien hat die Endung *.info*. Die maximale Größe ist die Gesamtgröße aller dieser Dateien. Wenn die maximale Puffergröße überschritten wird, wird die älteste Pufferdatei gelöscht.

### Weitere Puffereinstellungen

#### Maximale Zeit

Gespeicherte Daten, die älter als die maximale Zeit sind, werden nicht an das Zielsystem übertragen. Dateien, die älter als die Maximalzeit sind, können gelöscht werden. Sie können einen Wert zwischen 1 und 1000 Stunden eingeben.

#### Maximale Queue-Größe

Hier können Sie eine Obergrenze für die Anzahl der gepufferten SQL-Anweisungen festlegen. Je nach Anwendungsfall kann es sinnvoller sein eine Anzahl von Anweisungen vorzugeben, anstatt einer maximalen Zeit. Führend ist trotzdem die Puffergröße.

### Diagnose Speicherpuffer/Diagnose Dateipuffer

#### Zuletzt entferntes Element

Zeigt an, wann das letzte Element aus diesem Teil des Puffers genommen wurde.

#### Füllstand

Der Füllstand gibt jeweils an, welcher prozentuale Anteil der Puffergröße aktuell mit gepufferten Daten gefüllt ist.

#### Unverarbeiteter Anteil

Im Dateipuffer werden Elemente, die an das Zielsystem übertragen wurden, nicht sofort gelöscht. Erst wenn eine Pufferdatei vollständig gelesen ist, wird sie gelöscht. Es ist also möglich, dass nur ein Teil einer Pufferdatei noch nicht übertragene Daten enthält. Der Füllstand bezieht

sich auf die vorhandenen Pufferdateien, der "unverarbeitete Anteil" gibt dagegen den prozentualen Anteil der noch nicht übertragenen Daten im Dateipuffer wieder.

---

**Tipp**

Wenn Sie für ein SQL-Befehlsmodul auch ein Diagnosemodul in der I/O-Konfiguration anlegen, können Sie Werte zum Füllstand von Speicherpuffer und Dateipuffer überwachen und aufzeichnen.

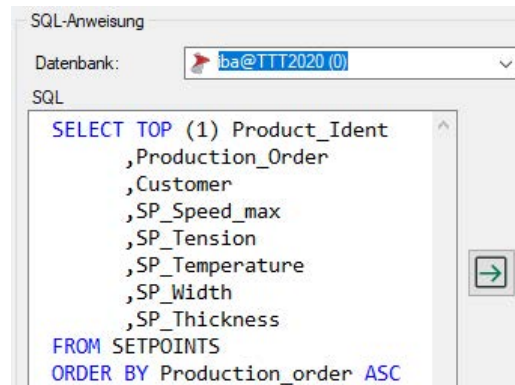
Siehe auch [➤ Diagnosemodule, Seite 39](#).

---

## 4.4 SQL-Anweisungen

Als SQL-Anweisungen werden bei SQL-Abfragen überwiegend SELECT-Anweisungen verwendet und bei SQL-Befehlen INSERT-, UPDATE- oder DELETE-Anweisungen.

Die SQL-Anweisung geben Sie im linken Fenster des Registers *SQL-Abfrage* bzw. *SQL-Befehl* des Moduls ein. Sie können die Anweisung direkt in das Fenster schreiben oder per Copy & Paste einfügen.



Verwenden Sie nur Anweisungen, die der verbundene Datenbank-Server ausführen kann. Die Syntax der SQL-Anweisung muss korrekt und im SQL-Dialekt des Datenbanksystems geschrieben sein. *ibaPDA* bietet keine Plausibilitätsprüfung der SQL-Anweisung!

Zur Unterstützung hebt *ibaPDA* einige Syntaxelemente farbig hervor:

- Schlüsselworte/Anweisungen/Operationen: blau
- Kommentare: grün
- Optionale SQL-Parameter: magenta

Die Hervorhebungen sind je nach DB-Systemtyp unterschiedlich, weil jeder SQL-Dialekt andere Schlüsselworte oder Verwendungsformen von Parametern hat. So werden z. B. bei Oracle Parameter mit einem Doppelpunkt ":", bei SQL Server mit einem at-Zeichen "@" gekennzeichnet.

Pro Modul ist eine SQL-Anweisung zulässig.

Unterschieden werden SQL-Anweisungen mit und ohne Parameter. Mit einer SQL-Abfrage ohne Parameter erhalten Sie als Ergebnis die aktuellen Werte der betreffenden, abgefragten Spalten aus der Datenbank und weisen diese Werte direkt den Signalen des Moduls zu.

Mithilfe von Parametern können Sie die Abfrage dynamisch an bestimmte Bedingungen knüpfen, um eine differenzierte Abfrage zu erstellen.

### 4.4.1 SQL-Anweisungen testen

Sowohl SQL-Anweisungen für Abfragen als auch für Befehle können Sie über den Button <Anweisung testen> testen.

Technisch wird bei dem Test die Anweisung in einer Transaktion ausgeführt, die anschließend wieder rückgängig gemacht wird (Rollback). Somit wird im Normalfall vermieden, dass die testweise Ausführung einer SQL-Anweisung zu Änderungen in der Datenbank führt. Gerade bei schreibenden SQL-Befehlen ist dies wichtig.

Je nach Komplexität der SQL-Anweisung und Aufbau der Datenbank kann aber trotzdem nicht gänzlich ausgeschlossen werden, dass nach dem Rollback Änderungen in der Datenbank verbleiben.

Im Unterschied zu den SQL-Befehlsmodulen (Ausgänge) gibt es bei den SQL-Abfragemodulen unten im Register *SQL-Abfrage* einen Bereich, in dem die Testergebnisse angezeigt werden.

The screenshot shows the 'SQL-Abfrage (5)' window. The 'Datenbank:' dropdown is set to 'iba@TTT2020 (0)'. The SQL query is: `SELECT TOP (1) Product_Ident, Production_Order, Customer, SP_Speed_max, SP_Tension, SP_Temperature, SP_Width, SP_Thickness FROM SETPOINTS ORDER BY Production_order ASC`. The 'Parameter in der SQL-Anweisung' table is empty. The 'Anweisung testen' button is visible. Below the query editor, a red box highlights the test results area, which includes a table with the following data:

Product_Ident	Production_Order	Customer	SP_Speed_max	SP_Tension	SP_Temperature	SP_Width	SP_Thickness
ABC000006	1	ibaKorea	140	13.7	550	1195	1.7

An dieser Stelle werden maximal die ersten 100 Zeilen des Abfrageergebnisses angezeigt.

Auch für SQL-Befehle können Sie die Anweisung testen. Allerdings erhalten Sie nur eine Rückmeldung, ob die Ausführung der Anweisung erfolgreich war oder nicht.

### 4.4.2 SQL-Abfragen mit Eingangssignalen verknüpfen

Bei SQL-Abfragen werden analoge und digitale Signale immer mit der Abfrage verknüpft. Somit stehen die abgefragten Werte als Signale für Visualisierung und Aufzeichnung zur Verfügung.

Die Signale können Sie in den Registern *Analog* und *Digital* des SQL-Abfragemoduls sehen.

Wenn der Abfragetest erfolgreich war, können Sie über den Button <Alle SQL-Ergebnisse hinzufügen> die abgefragten Werte in die Signaltabellen eintragen lassen.

Beispiel für die Abfrage einer Wertereihe (der ersten) aus einer Tabelle mit 8 Spalten:

**SQL-Abfrage (5)**

Allgemein
SQL-Abfrage
Analog
Digital

	Name	Einheit	Gain	Offset	Spalte	Zeile	Typ	Aktiv	Istwert
0	Product_Ident		1	0	0	0	STRING	<input checked="" type="checkbox"/>	ABC000006
1	Production_Order		1	0	1	0	DINT	<input checked="" type="checkbox"/>	1
2	Customer		1	0	2	0	STRING	<input checked="" type="checkbox"/>	ibaKorea
3	SP_Speed_max		1	0	3	0	FLOAT	<input checked="" type="checkbox"/>	140
4	SP_Tension		1	0	4	0	FLOAT	<input checked="" type="checkbox"/>	13.7
5	SP_Temperature		1	0	5	0	FLOAT	<input checked="" type="checkbox"/>	550
6	SP_Width		1	0	6	0	FLOAT	<input checked="" type="checkbox"/>	1195
7	SP_Thickness		1	0	7	0	FLOAT	<input checked="" type="checkbox"/>	1.7
8			1	0	0	0	FLOAT	<input type="checkbox"/>	
9			1	0	0	0	FLOAT	<input type="checkbox"/>	
10			1	0	0	0	FLOAT	<input type="checkbox"/>	
11			1	0	0	0	FLOAT	<input type="checkbox"/>	
12			1	0	0	0	FLOAT	<input type="checkbox"/>	
13			1	0	0	0	FLOAT	<input type="checkbox"/>	
14			1	0	0	0	FLOAT	<input type="checkbox"/>	

Testen Sie die Abfrage, um die Ergebnisse unten anzuzeigen. Wenn dort Abfrageergebnisse verfügbar sind, können Sie alle Ergebnisse als Signale hinzufügen.

Abgerufene Zeilen (max 100): 1

**Alle SQL-Ergebnisse hinzufügen**

	Product_Ident	Production_Order	Customer	SP_Speed_max	SP_Tension	SP_Temperature	SP_Width	SP_Thickness
▶	ABC000006	1	ibaKorea	140	13.7	550	1195	1.7

Falls nicht genügend Signale im Modul zur Verfügung stehen, werden Sie gefragt, ob Sie die Anzahl der Signale erhöhen wollen.

Der Bezug der hinzugefügten Signale zu den Abfrageergebnissen wird über die Spalten- und Zeilennummern hergestellt, die automatisch in die Signaltabelle eingetragen werden, wenn Sie auf den Button <Alle SQL-Ergebnisse hinzufügen> klicken. Wenn die Abfrage mehr als ein Ergebnis liefert, dann werden die Ergebnissätze untereinander nach Zeilennummer in die Signaltabelle eingefügt.

Eine SQL-Abfrage der ersten 3 Zeilen einer Tabelle mit der folgenden Anweisung würde z. B. das nachfolgende Ergebnis liefern.

```

SELECT TOP (3) Product_Ident
,Production_Order
,Customer
,SP_Speed_max
,SP_Tension
,SP_Temperature
,SP_Width
,SP_Thickness
FROM SETPOINTS
ORDER BY Production_order ASC
    
```

### SQL-Abfrage (5)

➤ Allgemein
➤ SQL-Abfrage
⌂ Analog
⏏ Digital

Name	Einheit	Gain	Offset	Spalte	Zeile	Typ	Aktiv	Istwert
0 Product_Ident (0)			1	0	0	STRING	<input checked="" type="checkbox"/>	ABC000006
1 Production_Order (0)			1	1	0	DINT	<input checked="" type="checkbox"/>	1
2 Customer (0)			1	2	0	STRING	<input checked="" type="checkbox"/>	ibaKorea
3 SP_Speed_max (0)			1	3	0	FLOAT	<input checked="" type="checkbox"/>	140
4 SP_Tension (0)			1	4	0	FLOAT	<input checked="" type="checkbox"/>	13.7
5 SP_Temperature (0)			1	5	0	FLOAT	<input checked="" type="checkbox"/>	550
6 SP_Width (0)			1	6	0	FLOAT	<input checked="" type="checkbox"/>	1195
7 SP_Thickness (0)			1	7	0	FLOAT	<input checked="" type="checkbox"/>	1.7
8 Product_Ident (1)			1	0	1	STRING	<input checked="" type="checkbox"/>	ABC000002
9 Production_Order (1)			1	1	1	DINT	<input checked="" type="checkbox"/>	2
10 Customer (1)			1	2	1	STRING	<input checked="" type="checkbox"/>	ibaBenelux
11 SP_Speed_max (1)			1	3	1	FLOAT	<input checked="" type="checkbox"/>	145
12 SP_Tension (1)			1	4	1	FLOAT	<input checked="" type="checkbox"/>	11.9
13 SP_Temperature (1)			1	5	1	FLOAT	<input checked="" type="checkbox"/>	510
14 SP_Width (1)			1	6	1	FLOAT	<input checked="" type="checkbox"/>	1220
15 SP_Thickness (1)			1	7	1	FLOAT	<input checked="" type="checkbox"/>	2.7
16 Product_Ident (2)			1	0	2	STRING	<input checked="" type="checkbox"/>	ABC000003
17 Production_Order (2)			1	1	2	DINT	<input checked="" type="checkbox"/>	3
18 Customer (2)			1	2	2	STRING	<input checked="" type="checkbox"/>	ibaChina
19 SP_Speed_max (2)			1	3	2	FLOAT	<input checked="" type="checkbox"/>	150
20 SP_Tension (2)			1	4	2	FLOAT	<input checked="" type="checkbox"/>	12.5
21 SP_Temperature (2)			1	5	2	FLOAT	<input checked="" type="checkbox"/>	495
22 SP_Width (2)			1	6	2	FLOAT	<input checked="" type="checkbox"/>	1225
23 SP_Thickness (2)			1	7	2	FLOAT	<input checked="" type="checkbox"/>	3.1
24			1	0	0	FLOAT	<input type="checkbox"/>	

Testen Sie die Abfrage, um die Ergebnisse unten anzuzeigen. Wenn dort Abfrageergebnisse verfügbar sind, können Sie alle Ergebnisse als Signale hinzufügen.
 Abgerufene Zeilen (max 100): 3

Product_Ident	Production_Order	Customer	SP_Speed_max	SP_Tension	SP_Temperature	SP_Width	SP_Thickness
ABC000006	1	ibaKorea	140	13.7	550	1195	1.7
ABC000002	2	ibaBenelux	145	11.9	510	1220	2.7
ABC000003	3	ibaChina	150	12.5	495	1225	3.1

### Tipp



Nutzen Sie bei SQL-Abfragen mit mehrzeiligen Ergebnissen die Anweisung ORDER BY, um eine definierte Sortierung der Ergebnisse und damit eine korrekte Referenzierung der Signale über die Zeilennummer zu erreichen.

Signale können auch manuell konfiguriert werden, indem Sie die korrekten Spalten- und Zeilennummern entsprechend dem zu erwartenden Abfrageergebnis eintragen.

### Hinweis



Wenn Sie die zu erwartenden Signale der Abfrage nicht von Hand in die Signaltabelle eintragen wollen, dann sind die Aktionen <Anweisung testen> und anschließend <Alle SQL-Ergebnisse hinzufügen> zwingend erforderlich. Ansonsten werden die Signale nicht angelegt und das Modul wird später im Signalbaum nicht angezeigt.

Bei SQL-Befehlen findet eine Verbindung mit ibaPDA-Signalen nur mittelbar über Parameter statt.

**Hinweis**

Falls ein Signal in *ibaPDA* den Wert "NaN" hat (möglich, z. B. beim Teilen durch Null oder ein falsches Format in der Steuerung) und dieser Wert mit einem SQL-Befehl in eine DB-Tabelle geschrieben wird, dann ersetzt *ibaPDA* diesen durch "NULL". Damit wird berücksichtigt, dass die meisten Datenbanken den Wert "NaN" bei Parametern nicht unterstützen. Voraussetzung dafür ist, dass die entsprechende Spalte in der DB-Tabelle "NULLable" ist, also der Wert "NULL" erlaubt ist.

Falls beim Lesen über die SQL-Schnittstelle der Wert "NULL" zurückgegeben wird, behandelt *ibaPDA* diesen als "NaN". Alternativ können Sie die SQL-Abfrage so schreiben, dass "NULL" in andere Ersatzwerte überführt wird.

---

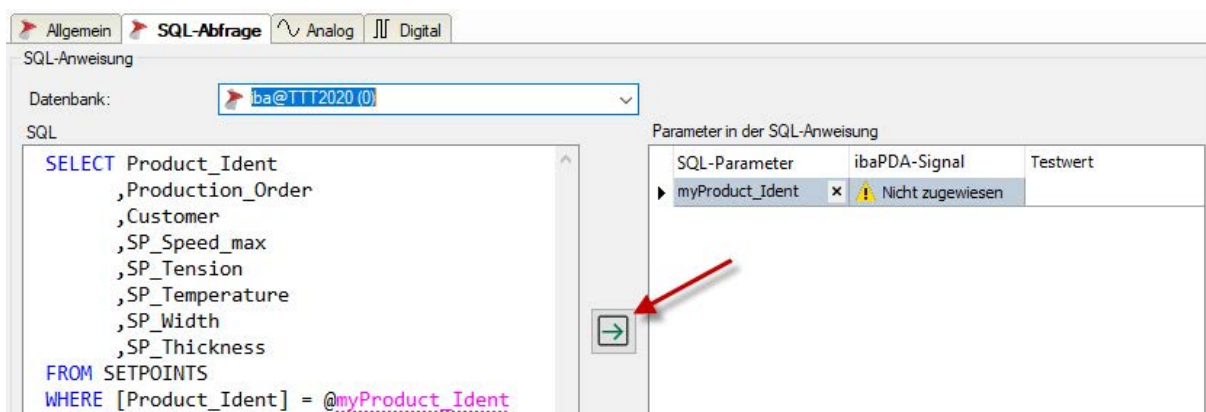
### 4.4.3 Parameter verwenden

Sowohl in SQL-Abfragen als auch in SQL-Befehlen können zusätzliche Parameter verwendet werden. Mithilfe dieser Parameter können Sie Abfragen dynamisch gestalten oder Bedingungen schaffen, die den Ergebnisraum erweitern bzw. einengen. Bei Befehlen können Sie mit Parametern Signalwerte in eine Datenbank schreiben.

Grundsätzlich können Parameter nur Werte ersetzen.

Parameter definieren Sie in den SQL-Anweisungen und verknüpfen sie mit Signalen in *ibaPDA*. Dabei gehen Sie wie folgt vor:

1. Schreiben Sie eine SQL-Anweisung, die Parameter enthält. Parameter werden bei den verschiedenen Datenbanksystemen unterschiedlich mit dem ersten Zeichen im Parameternamen gekennzeichnet.
  - Bei SQL Server, PostgreSQL und MySQL ist es "@".
  - Bei Oracle und SAP HANA ist es ":".
2. Klicken Sie anschließend auf den grünen Pfeil zwischen den Fenstern. Im nachfolgenden Beispiel SQL-Abfrage in Abhängigkeit von der Produkt-ID mit einem Parameter "myProduct\_Ident".



Alle Parameter in der Anweisung werden in die Tabelle im rechten Fenster übernommen. Sollten sich bereits Parameter dort befinden, werden Sie gefragt, ob überzählige Parameter gelöscht werden sollen.

3. In der Spalte *ibaPDA-Signal* können Sie nun ein Signal auswählen, das den Wert des Parameters bestimmen soll. Zulässig sind alle gängigen Signaltypen: Analogsignale, Digitalsignale und Textsignale.
4. Bei Bedarf können Sie einzelne Parameter aus der Liste löschen. Klicken Sie auf den gewünschten Parameter und anschließend auf den Button <X> in der Spalte *SQL-Parameter*.

SQL-Parameter	ibaPDA-Signal	Testwert
myident	Av 3:7: Product_Ident	ABC
mycustomer	Av 3:8: Customer	iba AG
myspeed	Diesen Parameter entfernen (Del)	
mytension	~ 3:4: AV_Tension	234
mytemperature	~ 3:5: AV_Temperatur	345
myprodent	~ 3:6: Product_Cycle_1	

Das ausgewählte Signal liefert den Wert für den Parameter bei Ausführung der Anweisung.

Damit eine Anweisung ausgeführt werden kann, z. B. beim Testen oder bei der Validierung zum Start der Messung, können Sie in der Spalte *Testwert* einen Wert eintragen. Geben Sie den Wert einfach als Text ein. Anhand des Typs des zugewiesenen Signals wird der Testwert korrekt interpretiert.

Digitalsignale haben die Werte 0 oder 1.

---

#### Hinweis



Beim Testen einer Anweisung können in der Parametertabelle mehr Parameter enthalten sein als die, die in der Anweisung verwendet werden. Beim Testen werden überzählige Parameter ignoriert und es erfolgt nur ein Warnhinweis. Vor dem Starten der Erfassung müssen Sie aber alle nicht verwendeten Parameter aus der Tabelle löschen, da sie bei der Erfassung nicht ignoriert werden und es dadurch zu Fehlern kommen kann.

---

#### Hinweis



Beachten Sie bei der Verwendung von Parametern in einer Abfragebedingung (WHERE-Anweisung), dass bei einer Änderung des Parameterwerts im Betrieb mehr oder weniger Zeilen im Ergebnis sein können, als ursprünglich bei der Konfiguration des Moduls vorgesehen.

In diesem Fall stimmen Anzahl Ergebnisse und Anzahl Signale nicht mehr überein.

Enthält das Ergebnis mehr Zeilen als entsprechende Signale vorhanden sind, werden die überzähligen Ergebnisse nicht berücksichtigt.

Enthält das Ergebnis weniger Zeilen als entsprechende Signale vorhanden sind, werden einige Signale nicht aktualisiert.

---

## 5 Diagnose

### 5.1 Lizenz

Falls die Schnittstelle *SQL-Datenbank* nicht im Signalbaum angezeigt wird, können Sie entweder in *ibaPDA* im I/O-Manager unter *Allgemein – Einstellungen* oder in der *ibaPDA* Dienststatus-Applikation überprüfen, ob Ihre Lizenz für eine oder mehrere der unterstützten Datenbanken ordnungsgemäß erkannt wird. Die Anzahl der lizenzierten Verbindungen ist in Klammern angegeben.

Sie benötigen **eine** der folgenden Lizenzen:

*ibaPDA-Interface-MySQL*, *ibaPDA-Interface-Oracle*, *ibaPDA-Interface-PostgreSQL*,  
*ibaPDA-Interface-SAP-HANA*, *ibaPDA-Interface-SQL-Server*

Lizenzinformationen		Lizenzen:
Lizenzcontainer:	3-1000000	ibaPDA-Interface-MySQL (8) ibaPDA-Interface-Oracle (8) ibaPDA-Interface-PostgreSQL (8) ibaPDA-Interface-SAP-HANA (8) ibaPDA-Interface-SQL-Server (8)
Kundenname:	ibapda Testsystem	
Nutzungsdauer:	Unbegrenzt	
Containertyp:	WIBU CmStick v4.40	
Container-Host:	ibapda-testsystem	
Erforderl. EUP-Datum:	01.02.2023	
EUP-Datum:	31.12.2025	

### 5.2 Sichtbarkeit der Schnittstelle

Ist die Schnittstelle trotz gültiger Lizenz nicht zu sehen, ist sie möglicherweise verborgen.

Überprüfen Sie die Einstellung im Register *Allgemein* im Knoten *Schnittstellen*.

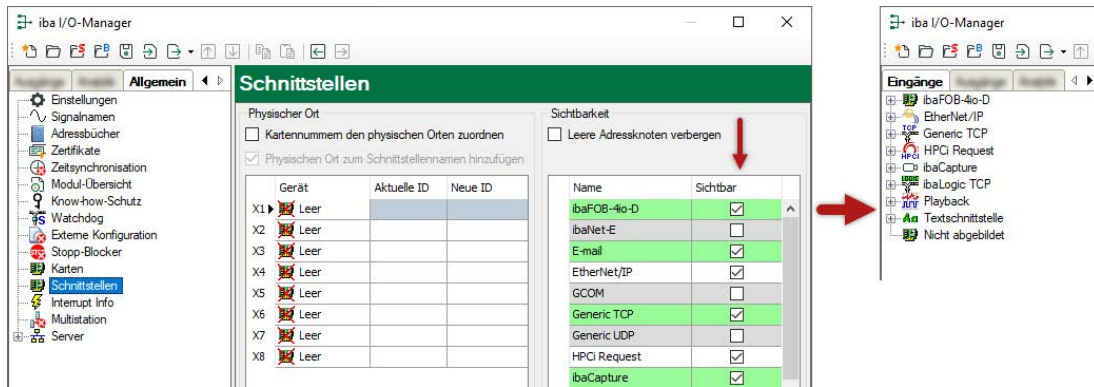
#### Sichtbarkeit

Die Tabelle *Sichtbarkeit* listet alle Schnittstellen auf, die entweder durch Lizenzen oder installierte Karten verfügbar sind. Diese Schnittstellen sind auch im Schnittstellenbaum zu sehen.

Mithilfe der Häkchen in der Spalte *Sichtbar* können Sie nicht benötigte Schnittstellen im Schnittstellenbaum verbergen oder anzeigen.

Schnittstellen mit konfigurierten Modulen sind grün hinterlegt und können nicht verborgen werden.

Ausgewählte Schnittstellen sind sichtbar, die anderen Schnittstellen sind verborgen:



### 5.3 Protokolldateien

Wenn Verbindungen zu Zielsystemen bzw. Clients hergestellt wurden, dann werden alle verbindungs-spezifischen Aktionen in einer Textdatei protokolliert. Diese (aktuelle) Datei können Sie z. B. nach Hinweisen auf mögliche Verbindungsprobleme durchsuchen.

Die Protokolldatei können Sie über den Button <Protokolldatei öffnen> öffnen. Der Button befindet sich im I/O-Manager:

- bei vielen Schnittstellen in der jeweiligen Schnittstellenübersicht
- bei integrierten Servern (z. B. OPC UA-Server) im Register Diagnose.

Im Dateisystem auf der Festplatte finden Sie die Protokolldateien von *ibaPDA*-Server (... \ProgramData\iba\ibaPDA\Log). Die Dateinamen der Protokolldateien werden aus der Bezeichnung bzw. Abkürzung der Schnittstellenart gebildet.

Dateien mit Namen *Schnittstelle.txt* sind stets die aktuellen Protokolldateien. Dateien mit Namen *Schnittstelle\_yyyy\_mm\_dd\_hh\_mm\_ss.txt* sind archivierte Protokoll-dateien.

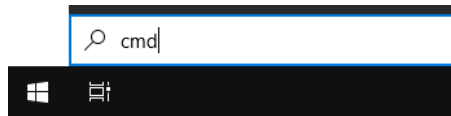
Beispiele:

- *ethernetipLog.txt* (Protokoll von EtherNet/IP-Verbindungen)
- *AbEthLog.txt* (Protokoll von Allen-Bradley-Ethernet-Verbindungen)
- *OpcUAServerLog.txt* (Protokoll von OPC UA-Server-Verbindungen)

## 5.4 Verbindungsdiagnose mittels PING

Ping ist ein System-Befehl, mit dem Sie überprüfen können, ob ein bestimmter Kommunikationspartner in einem IP-Netzwerk erreichbar ist.

1. Öffnen Sie eine Windows Eingabeaufforderung.



2. Geben Sie den Befehl "ping" gefolgt von der IP-Adresse des Kommunikationspartners ein und drücken Sie <ENTER>.

→ Bei bestehender Verbindung erhalten Sie mehrere Antworten.

```
Administrator: Eingabeaufforderung
Microsoft Windows [Version 10.0]
(c) Microsoft Corporation. Alle Rechte vorbehalten.

C:\Windows\system32>ping 192.168.81.10

Ping wird ausgeführt für 192.168.81.10 mit 32 Bytes Daten:
Antwort von 192.168.81.10: Bytes=32 Zeit=1ms TTL=30
Antwort von 192.168.81.10: Bytes=32 Zeit<1ms TTL=30
Antwort von 192.168.81.10: Bytes=32 Zeit<1ms TTL=30
Antwort von 192.168.81.10: Bytes=32 Zeit<1ms TTL=30

Ping-Statistik für 192.168.81.10:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
    (0% Verlust),
    Ca. Zeitangaben in Millisek.:
    Minimum = 0ms, Maximum = 1ms, Mittelwert = 0ms

C:\Windows\system32>
```

→ Bei nicht bestehender Verbindung erhalten Sie Fehlermeldungen.

```
Administrator: Eingabeaufforderung
Microsoft Windows [Version 10.0]
(c) Microsoft Corporation. Alle Rechte vorbehalten.

C:\Windows\system32>ping 192.168.81.10

Ping wird ausgeführt für 192.168.81.10 mit 32 Bytes Daten:
Antwort von 192.168.81.10: Zielhost nicht erreichbar.
Zeitüberschreitung der Anforderung.
Zeitüberschreitung der Anforderung.
Zeitüberschreitung der Anforderung.

Ping-Statistik für 192.168.81.10:
    Pakete: Gesendet = 4, Empfangen = 1, Verloren = 3
    (75% Verlust),
    Ca. Zeitangaben in Millisek.:
    Minimum = 0ms, Maximum = 1ms, Mittelwert = 0ms

C:\Windows\system32>
```

## 5.5 Diagnosemodule

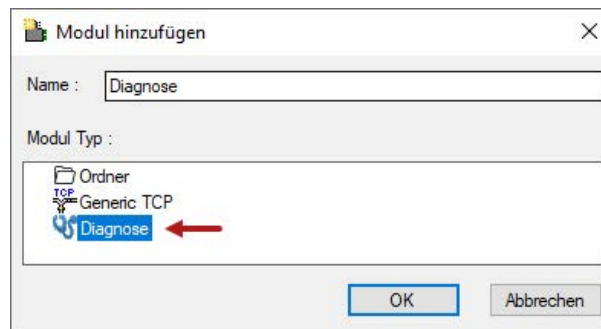
Diagnosemodule sind für die meisten Ethernet-basierten Schnittstellen und Xplorer-Schnittstellen verfügbar. Mit einem Diagnosemodul können Sie Informationen aus den Diagnoseanzeigen (z. B. Diagnoseregister und Verbindungstabellen einer Schnittstelle) als Signale erfassen.

Ein Diagnosemodul ist immer einem Datenerfassungsmodul derselben Schnittstelle zugeordnet und stellt dessen Verbindungsinformationen zur Verfügung. Durch die Nutzung eines Diagnosemoduls können die Diagnoseinformationen auch im *ibaPDA*-System durchgängig aufgezeichnet und ausgewertet werden. Diagnosemodule verbrauchen keine Verbindung der Lizenz, da sie keine Verbindung aufbauen, sondern auf ein anderes Modul verweisen.

Nutzungsbeispiele für Diagnosemodule:

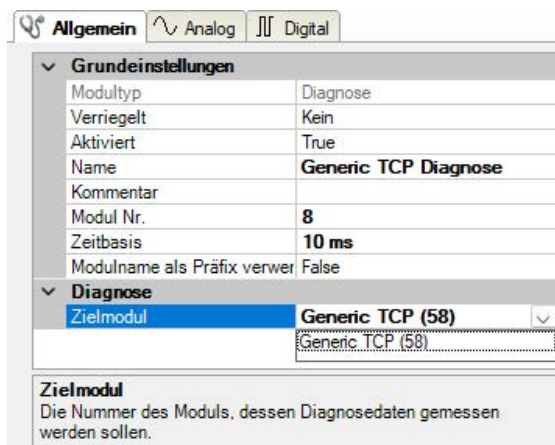
- Wenn der Fehlerzähler einer Kommunikationsverbindung einen bestimmten Wert überschreitet oder eine Verbindung abbricht, kann eine Benachrichtigung generiert werden.
- Bei einem Störfall können die aktuellen Antwortzeiten im Telegrammverkehr in einem Störungsreport dokumentiert werden.
- Der Status der Verbindungen kann in *ibaQPanel* visualisiert werden.
- Diagnoseinformationen können über den in *ibaPDA* integrierten SNMP-Server oder OPC DA/UA-Server an übergeordnete Überwachungssysteme wie Netzwerkmanagement-Tools weitergegeben werden.

Wenn für eine Schnittstelle ein Diagnosemodul verfügbar ist, wird im Dialog *Modul hinzufügen* der Modultyp "Diagnose" angezeigt (Beispiel: Generic TCP).



### Moduleinstellungen Diagnosemodul

Bei einem Diagnosemodul können Sie folgende Einstellungen vornehmen (Beispiel: Generic TCP):



Die Grundeinstellungen eines Diagnosemoduls entsprechen denen der anderen Module. Sie müssen nur eine für das Diagnosemodul spezifische Einstellung vornehmen: das Zielmodul auswählen.

Mit der Auswahl des Zielmoduls weisen Sie das Diagnosemodul dem Modul zu, dessen Verbindungsinformationen erfasst werden sollen. Die Auswahlliste zeigt die unterstützten Module derselben Schnittstelle. Jedem Diagnosemodul können Sie genau ein Datenerfassungsmodul zuordnen. Wenn Sie ein Modul ausgewählt haben, werden in den Registern *Analog* und *Digital* sofort die verfügbaren Diagnosesignale hinzugefügt. Welche Signale das sind, hängt vom Schnittstellentyp ab. Im nachfolgenden Beispiel sind die Analogwerte eines Diagnosemoduls für ein Generic TCP-Modul aufgelistet.

Name	Einheit	Gain	Offset	Aktiv	Istwert
0 IP-Adresse (Teil 1)			1	0	<input checked="" type="checkbox"/>
1 IP-Adresse (Teil 2)			1	0	<input checked="" type="checkbox"/>
2 IP-Adresse (Teil 3)			1	0	<input checked="" type="checkbox"/>
3 IP-Adresse (Teil 4)			1	0	<input checked="" type="checkbox"/>
4 Port			1	0	<input checked="" type="checkbox"/>
5 Telegrammzähler			1	0	<input checked="" type="checkbox"/>
6 Unvollständig			1	0	<input checked="" type="checkbox"/>
7 Paketgröße (aktuell)	Bytes		1	0	<input checked="" type="checkbox"/>
8 Paketgröße (max)	Bytes		1	0	<input checked="" type="checkbox"/>
9 Zeit zwischen Daten (aktuell)	ms		1	0	<input checked="" type="checkbox"/>
10 Zeit zwischen Daten (min)	ms		1	0	<input checked="" type="checkbox"/>
11 Zeit zwischen Daten (max)	ms		1	0	<input checked="" type="checkbox"/>

Die IP(v4-)Adresse eines Generic TCP-Moduls, z. B. (siehe Abbildung), wird entsprechend der 4 Bytes bzw. Oktetts in 4 Teile zerlegt, um sie leichter lesen und vergleichen zu können. Andere Größen, wie Portnummer, Zählerstände für Telegramme und Fehler, Datengrößen und Laufzeiten für Telegramme werden ebenfalls ermittelt. Im nachfolgenden Beispiel sind die Digitalwerte eines Diagnosemoduls für ein Generic TCP-Modul aufgelistet.

Name	Aktiv	Istwert
0 Aktiver Verbindungsmodus	<input checked="" type="checkbox"/>	
1 Ungültiges Paket	<input checked="" type="checkbox"/>	
2 Verbinde	<input checked="" type="checkbox"/>	
3 Verbunden	<input checked="" type="checkbox"/>	

### Diagnosesignale

Abhängig vom Schnittstellentyp stehen folgende Signale zur Verfügung:

Signalname	Bedeutung
Aktiv	Nur für redundante Verbindungen relevant. Aktiv bedeutet, dass die Verbindung zur Messung der Daten verwendet wird, d. h. bei redundanten Standby-Verbindungen steht der Wert 0. Bei normalen/nicht redundanten Verbindungen steht immer der Wert 1.
Aktualisierungszeit (Istwert/konfiguriert/max/min/Mittelwert)	Gibt die Aktualisierungszeit an, in der die Daten aus der SPS, der CPU oder vom Server abgerufen werden sollen (konfiguriert). Standard ist gleich dem Parameter "Zeitbasis". Während der Messung kann die reale aktuelle Aktualisierungszeit (Istwert) höher sein als der eingestellte Wert, wenn die SPS mehr Zeit zur Übertragung der Daten benötigt. Wie schnell die Daten wirklich aktualisiert werden, können Sie in der Verbindungstabelle überprüfen. Die minimal erreichbare Aktualisierungszeit wird von der Anzahl der Signale beeinflusst. Je mehr Signale erfasst werden, desto größer wird die Aktualisierungszeit.  Max/min/Mittelwert: statische Werte der Aktualisierungszeit seit dem letzten Start der Erfassung bzw. Rücksetzen der Zähler
Anforderungen Sendewiederholung	Anzahl der nochmals angeforderten Datentelegramme (in) bei Verlust oder Verspätung
Antwortzeit (aktuell/konfiguriert/max/min/Mittelwert)	Antwortzeit ist die Zeit zwischen Messwertanforderung von <i>ibaPDA</i> und Antwort von der SPS bzw. Empfang der Daten.  Aktuell: Istwert  Max/min/Mittelwert: statische Werte der Antwortzeit seit dem letzten Start der Erfassung bzw. Rücksetzen der Zähler
Anzahl Anforderungsbefehle	Zähler für Anforderungstelegramme von <i>ibaPDA</i> an die SPS/CPU
Aufgebaute Verbindungen (in)	Anzahl der aktuell gültigen Datenverbindungen für den Empfang
Aufgebaute Verbindungen (out)	Anzahl der aktuell gültigen Datenverbindungen für das Senden
Ausgangsdatenlänge	Länge der Datentelegramme mit Ausgangssignalen in Bytes ( <i>ibaPDA</i> sendet)
Datenlänge	Länge der Datentelegramme in Bytes
Datenlänge des Inputs	Länge der Datentelegramme mit Eingangssignalen in Bytes ( <i>ibaPDA</i> empfängt)
Datenlänge O->T	Größe des Output-Telegramms in Byte
Datenlänge T->O	Größe des Input-Telegramms in Byte

Signalname	Bedeutung
Definierte Topics	Anzahl der definierten Topics
Empfangene Telegramme seit Konfiguration	Anzahl der empfangenen Datentelegramme (in) seit Beginn der Erfassung
Empfangene Telegramme seit Verbindungsstart	Anzahl der empfangenen Datentelegramme (in) seit Beginn des letzten Verbindungsaufbaus
Empfangszähler	Anzahl der empfangenen Telegramme
Exchange ID	ID des Datenaustauschs
Falscher Telegrammtyp	Anzahl der Empfangstelegramme mit falschem Telegrammtyp
Fehlerzähler	Zähler der Kommunikationsfehler
Gepufferte Anweisungen	Anzahl der noch nicht ausgeführten Anweisungen im Zwischenspeicher
Gepufferte Anweisungen sind verloren	Anzahl der gepufferten aber nicht ausgeführten und verlorenen Anweisungen
Gesendete Telegramme seit Konfiguration	Anzahl der gesendeten Datentelegramme (out) seit Beginn der Erfassung
Gesendete Telegramme seit Verbindungsstart	Anzahl der gesendeten Datentelegramme (out) seit Beginn des letzten Verbindungsaufbaus
ID der Verbindung O->T	ID der Verbindung für Output-Daten (vom Zielsystem an <i>ibaPDA</i> ) Entspricht der Assembly-Instanznummer
ID der Verbindung T->O	ID der Verbindung für Input-Daten (von <i>ibaPDA</i> an Zielsystem) Entspricht der Assembly-Instanznummer
IP-Adresse (Teil 1-4)	4 Oktets der IP-Adresse des Zielsystems
IP-Quelladresse (Teil 1-4) O->T	4 Oktets der IP-Adresse des Zielsystems Output-Daten (vom Zielsystem an <i>ibaPDA</i> )
IP-Quelladresse (Teil 1-4) T->O	4 Oktets der IP-Adresse des Zielsystems Input-Daten (von <i>ibaPDA</i> an Zielsystem)
IP-Zieladresse (Teil 1-4) O->T	4 Oktets der IP-Adresse des Zielsystems Output-Daten (vom Zielsystem an <i>ibaPDA</i> )
IP-Zieladresse (Teil 1-4) T->O	4 Oktets der IP-Adresse des Zielsystems Input-Daten (von <i>ibaPDA</i> an Zielsystem)
KeepAlive-Zähler	Anzahl der vom OPC UA-Server empfangenen KeepAlive-Telegramme
Lesezähler	Anzahl der Lesezugriffe/Datenanforderungen
Multicast Anmeldefehler	Anzahl der Fehler bei Multicast-Anmeldung
Nachrichtenzähler	Anzahl der empfangenen Telegramme
Paketgröße (aktuell)	Größe der aktuell empfangenen Telegramme
Paketgröße (max)	Größe des größten empfangenen Telegramms
Ping-Zeit (Istwert)	Antwortzeit für ein Ping-Telegramm
Port	Portnummer für die Kommunikation
Producer ID (Teil 1-4)	Producer-ID als 4 Byte unsigned Integer

Signalname	Bedeutung
Profilzähler	Anzahl der vollständig erfassten Profile
Pufferdateigröße (aktuell/mittl./max)	Größe der Pufferdatei zum Zwischenspeichern der Anweisungen
Pufferspeichergröße (aktuell/mittl./max)	Größe des belegten Arbeitsspeichers zum Zwischenspeichern der Anweisungen
Schreibverlustzähler	Anzahl missglückter Schreibzugriffe
Schreibzähler	Anzahl erfolgreicher Schreibzugriffe
Sendezähler	Anzahl der Sendetelegramme
Sequenzfehler	Anzahl Sequenzfehler
Synchronisation	Gerät wird für die isochrone Erfassung synchronisiert
Telegramme pro Zyklus	Anzahl der Telegramme im Zyklus der Aktualisierungszeit
Telegrammzähler	Anzahl der empfangenen Telegramme
Topics aktualisiert	Anzahl der aktualisierten Topics
Trennungen (in)	Anzahl der aktuell unterbrochenen Datenverbindungen für den Empfang
Trennungen (out)	Anzahl der aktuell unterbrochenen Datenverbindungen für das Senden
Unbekannter Sensor	Anzahl unbekannter Sensoren
Ungültiges Paket	Ungültiges Datenpaket erkannt
Ungültige Datenpunkte	Anzahl empfangener Datenpunkte mit fehlender Konfiguration
Unvollständig	Anzahl unvollständiger Telegramme
Unvollständige Fehler	Anzahl unvollständiger Telegramme
Verarbeitete Anweisungen	Anzahl der ausgeführten SQL-Anweisungen seit dem letzten Start der Erfassung
Verbinde	Verbindung wird aufgebaut
Verbindungsphase (in)	Zustand der ibaNet-E Datenverbindung für den Empfang
Verbindungsphase (out)	Zustand der ibaNet-E Datenverbindung für das Senden
Verbindungsversuche (in)	Anzahl der Versuche, die Empfangsverbindung (in) aufzubauen
Verbindungsversuche (out)	Anzahl der Versuche, die Sendeverbindung (out) aufzubauen
Verbunden	Verbindung ist aufgebaut
Verbunden (in)	Eine gültige Datenverbindung für den Empfang (in) ist vorhanden
Verbunden (out)	Eine gültige Datenverbindung für das Senden (out) ist vorhanden
Verlorene Images	Anzahl der verlorenen Images (in), die selbst nach einer Sendewiederholung nicht empfangen wurden
Verlorene Profile	Anzahl unvollständiger/fehlerhafter Profile

Signalname	Bedeutung
Zeilen (letzte)	Anzahl der Ergebniszeilen der letzten SQL-Abfrage (innerhalb der projektierten Anzahl von Ergebniszeilen)
Zeilen (Maximum)	Höchste Anzahl der Ergebniszeilen einer SQL-Abfrage seit dem letzten Start der Erfassung (maximal gleich der projektierten Anzahl von Ergebniszeilen)
Zeit zwischen Daten (aktuell/max/min)	Zeit zwischen zwei korrekt empfangenen Telegrammen Aktuell: zwischen den letzten zwei Telegrammen Max/min: statistische Werte seit Start der Erfassung oder Rücksetzen der Zähler
Zeit-Offset (Istwert)	Gemessene Zeitdifferenz der Synchronität zwischen dem ibaNet-E-Gerät und <i>ibaPDA</i>

## 6 Support und Kontakt

### Support

Tel.: +49 911 97282-14

E-Mail: [support@iba-ag.com](mailto:support@iba-ag.com)

---

### Hinweis



Wenn Sie Support benötigen, dann geben Sie bitte bei Software-Produkten die Nummer des Lizenzcontainers an. Bei Hardware-Produkten halten Sie bitte ggf. die Seriennummer des Geräts bereit.

---

### Kontakt

#### Hausanschrift

iba AG  
Gebhardtstraße 10-20  
90762 Fürth  
Deutschland

Tel.: +49 911 97282-0

E-Mail: [iba@iba-ag.com](mailto:iba@iba-ag.com)

#### Postanschrift

iba AG  
Postfach 1828  
90708 Fürth

#### Warenanlieferung, Retouren

iba AG  
Gebhardtstraße 10  
90762 Fürth

#### Regional und weltweit

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

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