

Unified-E SQL Adapter Benutzerhandbuch

SQL-Endpunkte und Datenpunkte konfigurieren

Software-Version 3.1.0.0, zuletzt aktualisiert: Juli 2025

Herausgeber: Unified-E AG, Winterthur, Schweiz



Inhalt

1	Allgemeines	3
2	Adapter-Parameter	3
2.1	Allgemeine Parameter	3
2.2	Datenbankspezifische Parameter	5
2.2.1	Microsoft SQL Server	5
2.2.2	Oracle Datenbank	5
2.2.3	Microsoft Access Datenbank Datei	6
2.2.4	MySQL Datenbank	6
2.2.5	ODBC Datenquelle	7
2.2.6	Andere ADO.NET Datenquelle	8
3	Adressierung von Datenpunkten	9
3.1	Werte auslesen	9
3.2	Verwenden von Abfrage-Parametern	9
3.3	SQL Manipulationsbefehle ausführen	12
3.4	Stored Procedures (Skripte) verwenden	13
4	Performance-Hinweise	13
5	Sicherheitshinweise	14

1 Allgemeines

Mit dem Unified-E SQL Adapter ist es möglich, dass Daten aus einer Datenbank für die Bedien-App gelesen oder auch in die Datenbank geschrieben werden.

Wie bei allen Adaptern wird der Datenaustausch hier im Unified-E App Designer mit Hilfe von Datenpunkten konfiguriert. Diese Datenpunkte lassen sich genauso verwenden wie die Datenpunkte anderer Adapter.

Anwendungsbeispiele:

- Diagramme anzeigen
- Log-Listen anzeigen
- Auswahllisten anzeigen, zum Beispiel für das Auswählen eines Produktionsauftrages
- Eingabe von Datensätzen, zum Beispiel Produktionsnotizen

Beispiel: Kommunikation mit dem ERP-System:

Das ERP-System exportiert in regelmässigen Abständen die relevanten Daten für die Bedien-App-Visualisierung in eine Datenbank. Häufig wird für den Datenaustausch zum ERP-System eine Access-Datenbank verwendet.

Unterstützte Datenbanksysteme:

Der Adapter unterstützt ohne weitere Client-Software vom Datenbankhersteller bereits folgende Datenbanken:

- Microsoft SQL Server
- Oracle Datenbank
- Microsoft Access Datenbank Datei
- MySQL Datenbank

Weitere Datenbanksysteme können verwendet werden. Dafür muss allerdings ein entsprechender ODBC- oder ADO.NET Treiber installiert werden, der normalerweise beim Datenbankhersteller zu erwerben ist.

2 Adapter-Parameter

Die Parameter dienen der Konfigurationsanforderung, zum Beispiel die Adresse des Datenbankservers oder der Name der Datenbank.

2.1 Allgemeine Parameter

Verbindungstyp:

Dieser Parameter beschreibt das Protokoll bzw. mit welchem Datenbanksystem kommuniziert werden soll. Die spezifischen Parameter sind in den Folgekapiteln dokumentiert.

Verbindungskonfiguration:

- **Standard:**
Die Konfiguration erfolgt mit den typischen Verbindungsparametern, die jeweils abhängig sind vom Verbindungstyp.
- **Verbindungszeichenfolge:**
Die Verbindungszeichenfolge (Connection String) enthält alle Parameter und muss im ADO.NET Format sein. Diese Konfigurationsart sollte nur von Personen mit Datenbank-Kenntnissen verwendet werden.

Benutzer ID:

Der Datenbankbenutzer.

Passwort:

Das Passwort zur Benutzer ID. Es ist möglich, in der Verbindungszeichenfolge den Platzhalter {Passwort} zu verwenden, um das Passwort auch in der Verbindungszeichenfolge zu schützen.

Datenbank-Zeitzone:

Die Bedien-Apps erhalten alle Datum/Uhrzeit Werte immer im UTC-Format (universelle Zeitzone) vom Unified-E App Manager. Diese werden dann in die lokale Zeit des Bediengerätes umgewandelt. Für eine korrekte Darstellung muss die Zeitzone der in der Datenbank gespeicherten Datum/Uhrzeit Werte bekannt sein.

- **Lokale Zeitzone:**
Die Datum/Uhrzeit Werte in den Datenbank-Tabellen befinden sich in derselben Zeitzone wie der App-Manager PC.
- **Universelle Zeitzone:**
Die Datum/Uhrzeit Werte in den Datenbank-Tabellen werden als universell (UTC) interpretiert.

Timeout [s]:

Der Timeout-Wert wird sowohl beim Verbindungsaufbau und beim Ausführen einer SQL-Abfrage verwendet.

2.2 Datenbankspezifische Parameter

2.2.1 Microsoft SQL Server

Windows-Authentifizierung:

- Ja:
Der Login erfolgt mit dem Windows-Benutzer.
- Nein:
Es wird der Datenbank-Benutzer verwendet, so wie in den Feldern "Benutzer ID" und "Passwort" angegeben.

Serveradresse:

Die Adresse des Servers, z. B. URL, Rechnername, IP-Adresse.

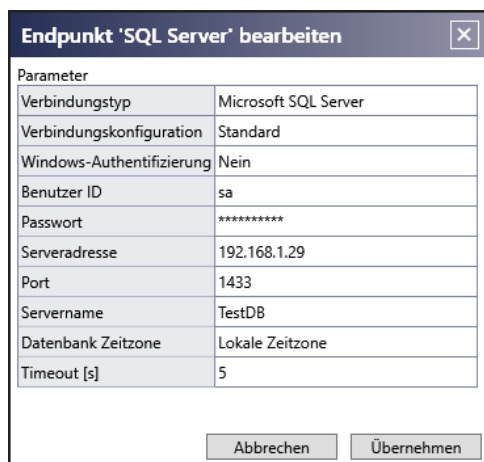
Port:

Der Port, über den der Server erreichbar ist.

Servename:

Der Name der zu verwendeten Server-Instanz.

Beispiel:



Parameter	
Verbindungstyp	Microsoft SQL Server
Verbindungskonfiguration	Standard
Windows-Authentifizierung	Nein
Benutzer ID	sa
Passwort	*****
Serveradresse	192.168.1.29
Port	1433
Servename	TestDB
Datenbank Zeitzone	Lokale Zeitzone
Timeout [s]	5

Abbrechen Übernehmen

2.2.2 Oracle Datenbank

Serveradresse:

Die Adresse des Servers, z. B. URL, Rechnername, IP-Adresse.

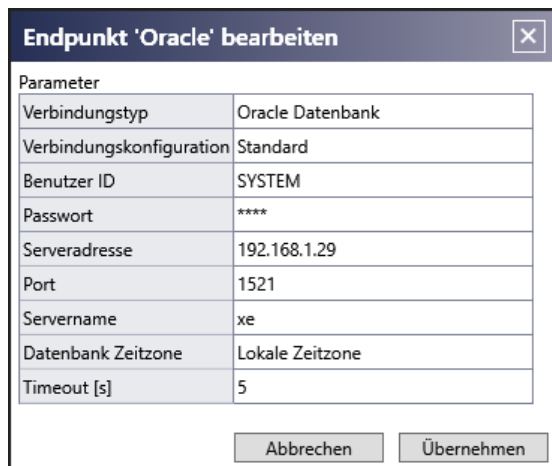
Port:

Der Port, über den der Server erreichbar ist.

Servername:

Der Name der zu verwendeten Server-Instanz.

Beispiel:



Parameter	
Verbindungstyp	Oracle Datenbank
Verbindungskonfiguration	Standard
Benutzer ID	SYSTEM
Passwort	****
Serveradresse	192.168.1.29
Port	1521
Servername	xe
Datenbank Zeitzone	Lokale Zeitzone
Timeout [s]	5

Abbrechen Übernehmen

2.2.3 Microsoft Access Datenbank Datei

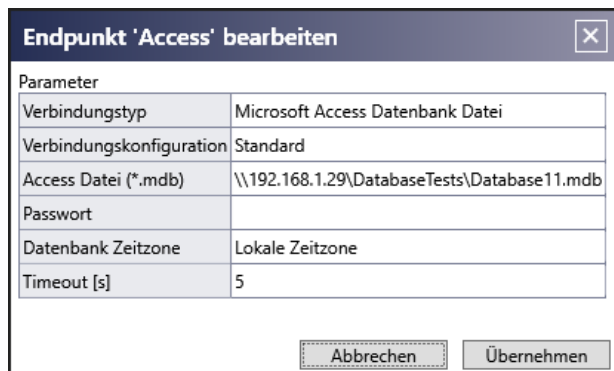
Der Zugriff auf die Access-Datei (*.mdb) erfolgt via OLE DB. Die erforderlichen Treiber sind auf Windows standardmässig installiert.

Spezifische Parameter:

Access Datei (*.mdb):

Der Pfad zur Access-Datei. Stellen Sie sicher, dass dieser auch vom Unified-E Windows-Dienst erreichbar ist. Verwenden Sie keine gemappten Laufwerke, damit der Unified-E Dienst auf die Datei zugreifen kann.

Beispiel:



Parameter	
Verbindungstyp	Microsoft Access Datenbank Datei
Verbindungskonfiguration	Standard
Access Datei (*.mdb)	\\192.168.1.29\DatabaseTests\Database11.mdb
Passwort	
Datenbank Zeitzone	Lokale Zeitzone
Timeout [s]	5

Abbrechen Übernehmen

2.2.4 MySQL Datenbank

Serveradresse:

Die Adresse des Servers, z. B. URL, Rechnername, IP-Adresse.

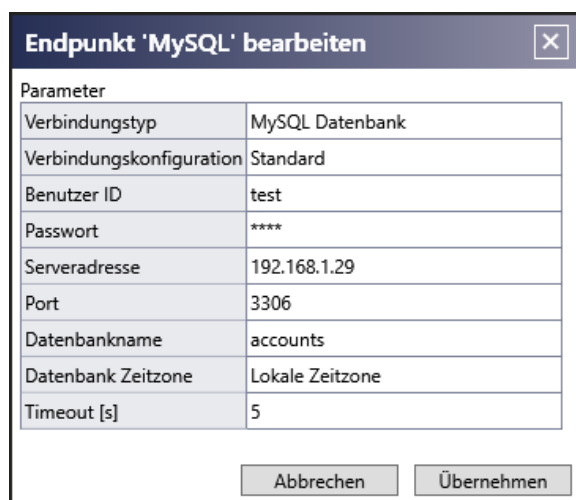
Port:

Der Port, über den der Server erreichbar ist.

Datenbankname:

Der Name der zu verwendeten Server-Instanz.

Beispiel:



Parameter	
Verbindungstyp	MySQL Datenbank
Verbindungskonfiguration	Standard
Benutzer ID	test
Passwort	****
Serveradresse	192.168.1.29
Port	3306
Datenbankname	accounts
Datenbank Zeitzone	Lokale Zeitzone
Timeout [s]	5

Abbrechen Übernehmen

2.2.5 ODBC Datenquelle

Ablauf:

Das Einrichten einer Verbindung zu einer ODBC-Datenbank erfolgt in folgenden Schritten:

1. ODBC-Treiber für das Datenbanksystem installieren
2. Die Verbindungskonfiguration festlegen
 - a. Datei-DSN (via Systemsteuerung -> ODBC Datenquellen-Administrator)
 - b. System-DSN (via Systemsteuerung -> ODBC Datenquellen-Administrator)
 - c. Direkt
3. Verbindungsparameter in Unified-E setzen

Spezifische Parameter:

Verbindungskonfiguration:

System-DSN: Die Konfiguration wird aus einer vorhandenen System-DSN ausgelesen. Diese kann mit dem ODBC Datenquellen-Administrator der Windows-Systemsteuerung eingerichtet werden.

Datei-DSN: Die Konfiguration wird aus einer vorhandenen Datei-DSN ausgelesen. Diese kann mit dem ODBC Datenquellen-Administrator der Windows-Systemsteuerung eingerichtet werden.

Verbindungszeichenfolge: Die Konfiguration erfolgt manuell (siehe Beispiel unten).

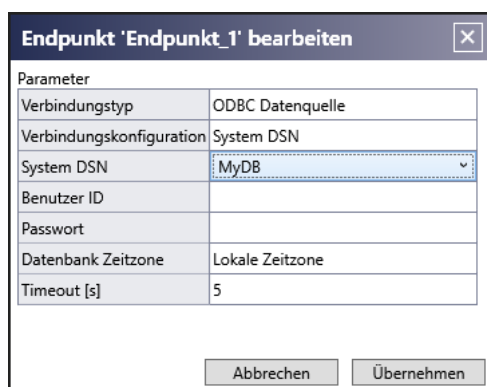
System-DSN:

Die zu verwendende System-DSN ist zu wählen.

DSN Datei (*.dsn):

Die zu verwendende Datei-DSN ist zu wählen.

Beispiele:



The screenshot shows a dialog box titled "Endpoint 'Endpoint_1' bearbeiten". It contains a table with the following parameters:

Parameter	
Verbindungstyp	ODBC Datenquelle
Verbindungskonfiguration	System DSN
System DSN	MyDB
Benutzer ID	
Passwort	
Datenbank Zeitzone	Lokale Zeitzone
Timeout [s]	5

At the bottom of the dialog box, there are two buttons: "Abbrechen" and "Übernehmen".



The screenshot shows a dialog box titled "Endpoint 'Endpoint_1' bearbeiten". It contains a table with the following parameters:

Parameter	
Verbindungstyp	ODBC Datenquelle
Verbindungskonfiguration	Verbindungszeichenfolge
Verbindungszeichenfolge	Driver=Driver do Microsoft Access (*.mdb); DBQ=C:\Work\Logs.mdb
Passwort	
Datenbank Zeitzone	Lokale Zeitzone
Timeout [s]	5

At the bottom of the dialog box, there are two buttons: "Abbrechen" and "Übernehmen".

2.2.6 Andere ADO.NET Datenquelle

Provider Name:

Wählen Sie den gewünschten ADO.NET Provider aus der Auswahlliste.

Verbindungszeichenfolge:

Die spezifische Verbindungszeichenfolge für den ADO.NET Provider. Der genaue Aufbau der Zeichenfolge ist Provider-spezifisch und ist der jeweiligen Dokumentation zu entnehmen.

3 Adressierung von Datenpunkten

3.1 Werte auslesen

Der Datenpunkt-Wert lässt sich mit Hilfe eines SELECT-Befehls adressieren.

Beispiel: Adresse eines Tabellen-Datenpunktes

SELECT TOP 100 CreatedOnTimeStamp, CreatedByUser, IsUrgent, NoteText FROM Notes
ORDER BY createdontimestamp DESC

Bezeichnung	Adresse	Zugriff	Datentyp	Steuer-Passw
NotesList	SELECT TOP 100 CreatedOnTimeStamp, CreatedByUser, IsUrgent, NoteText FROM Notes ORDER BY createdontimestamp DESC	Lesen	Tabelle	Nein
NoteText	name=NoteText; defaultvalue=; datatype=String	Schreiben, Lesen	Text	Nein
IsUrgent	name=IsUrgent; defaultvalue=0; datatype=Boolean	Schreiben, Lesen	Ja/Nein	Nein

Der Datentyp muss auf "Tabelle" gesetzt sein, um die Tabelle in einem Listen- oder Diagramm-Bereich verwenden zu können.

Beispiel: Auslesen eines numerischen Wertes

SELECT max(Id) FROM notes WHERE IsUrgent=true and (createdbyuser <> {system:currentuser})

Bezeichnung	Adresse	Zugriff	Datentyp	Steuer-Passw
NotesList	SELECT TOP 100 CreatedOnTimeStamp, CreatedByUser, IsUrgent, NoteText FROM Notes ORDER BY createdontimestamp DESC	Lesen	Tabelle	Nein
NoteText	name=NoteText; defaultvalue=; datatype=String	Schreiben, Lesen	Text	Nein
IsUrgent	name=IsUrgent; defaultvalue=0; datatype=Boolean	Schreiben, Lesen	Ja/Nein	Nein
AddNoteTrigger	insert into Notes (CreatedOnTimeStamp, CreatedByUser, IsUrgent, NoteText)	Schreiben, Lesen	Numerisch	Nein
LastUrgentNote	SELECT max(Id) from notes where IsUrgent=true	Lesen	Numerisch	Nein

Beim Auslesen eines einzelnen Wertes muss der Datentyp entsprechend gesetzt werden, in diesem Beispiel auf "Numerisch".

3.2 Verwenden von Abfrage-Parametern

Datenpunkte können als Abfrage-Parameter festgelegt werden und dann in einer SQL-Abfrage verwendet werden.

Abfrage-Parameter sind immer Benutzereingaben, deshalb müssen diese Datenpunkte als "Schreiben, Lesen" konfiguriert werden.

Anwendungsfälle:

- Liste filtern
- Zeitraum für Diagramm-Daten festlegen
- Benutzereingabe für UPDATE oder INSERT Befehle

Aufbau der Adresse:

Beispiel: name=NoteText; defaultvalue=; datatype=String

Bezeichnung	Adresse	Zugriff
NotesList	select top 100 CreatedOnTimeStamp, CreatedByUser, IsUrgent, NoteText from Notes	...
NoteText	name=NoteText; defaultvalue=; datatype=String	... Schreiben, Lesen
IsUrgent	name=IsUrgent; defaultvalue=0; datatype=Boolean	... Schreiben, Lesen

Adress-Komponenten:

Name:

Der Name, so wie er im Platzhalter des SQL-Befehls innerhalb {} verwendet wird. Wenn im SQL-Befehl geschweifte Klammern nicht für den Platzhalter verwendet werden sollen (z. B. im String-Literal), dann müssen diese wiederholt werden.

Defaultvalue:

Der initiale Wert, wenn noch keine Benutzereingaben vorgenommen wurden.

Datatype:

Der Datentyp des Parameters.

- Boolean: 0, 1
- Char: Unicode character
- SByte: Signed 8-bit integer
- Byte: Unsigned 8-bit integer
- Int16: Signed 16-bit integer
- UInt16: Unsigned 16-bit integer
- Int32: Signed 32-bit integer
- UInt32: Unsigned 32-bit integer
- Int64: Signed 64-bit integer
- UInt64: Unsigned 64-bit integer
- Single: IEEE 32-bit float

- Double: IEEE 64-bit double
- Decimal: Decimal
- DateTime: Datum/Uhrzeit
- String: Unicode character string

Wildcardtype (optional):

Diese Option ist nur für String-Parameter möglich. Beispiel: Eine Liste soll nach Namen gefiltert werden. Dann kann eine tolerante Filterung mit Wildcard (%) eingestellt werden.

- None: Keine Wildcards hinzufügen
- Before: Prozentzeichen vor Parameter-Wert
- After: Prozentzeichen nach Parameter-Wert
- BeforeAndAfter : Prozentzeichen vor und nach Parameter-Wert

Adresse im Editor konfigurieren:

Adresse konfigurieren

Adresstyp
☒ Parameter ☐ SQL-Befehl

Parameter konfigurieren

Parametername
NoteText

Standardwert

Datentyp
String

Wildcard hinzufügen
Ohne Wildcards

Abbrechen Übernehmen

Built-In Parameter:

{system:currentuser}:

Der Wert des Platzhalter repräsentiert den Namen des aktuellen Bedien-App Benutzer, so wie er im Unified-E App Manager registriert ist.

3.3 SQL Manipulationsbefehle ausführen

Wenn der Datenpunkt-Wert eines SQL-Befehls auf den Wert "1" gesetzt wird dann wird der Manipulationsbefehl ausgeführt.

Im Manipulationsbefehl sind normalerweise auch Platzhalter für die Parameter-Datenpunkte enthalten.

Beispiel-Adresse:

Adresse konfigurieren

Adresstyp
☐ Parameter ☒ SQL-Befehl

SQL-Befehl konfigurieren
Platzhalter für Parameter einfügen: NoteText >>

insert into Notes (CreatedOnTimestamp, Createdbyuser, isUrgent, notetext)
Values (Now(), {system:currentuser}, {IsUrgent}, {NoteText})

Abbrechen Übernehmen

Beispiel: Manipulationsbefehl im Werteingabe-Bereich triggern:

In den Eigenschaften eines Eingabebereichs kann durch ein Betätigen der Übernehmen-Schaltfläche der INSERT-Befehl wie folgt getriggert werden:

Senden

☐ Eingabewerte sofort senden
☒ Eingabewerte mit Schaltfläche senden

Sende-Trigger (optional)
MDB: AddNoteTrigger

Triggerwert
1

Analog kann ein Trigger beim Drücken einer Schaltfläche im Schaltflächen-Bereich ausgelöst werden.

3.4 Stored Procedures (Skripte) verwenden

Bei Stored Procedure-Aufrufen bzw. Skript-Aufrufen ist zu unterscheiden, ob dessen Rückgabe-Wert bzw. die Rückgabe-Tabelle in einer Ansicht verwendet werden soll, oder ob ähnlich wie bei Manipulationsbefehlen der Aufruf die Daten in der Datenbank verändert.

Abfrage-Skripte:

Bei Abfrage-Skripten wird davon ausgegangen, dass der Rückgabewert verwendet wird und keine Daten in der Datenbank verändert werden. Beim SQL-Befehl muss daher bei der Datenpunkt-Adresse noch „QUERY“ vorangestellt werden. Bei diesen Datenpunkten ist der Zugriff daher auf "Lesen" zu setzen.

Beispiel für Datenpunkt-Adresse bei MS SQL Server:

QUERY EXEC <Prozedurname>

Command-Skripte:

Bei Command-Skripten, das heisst, Skripte, die keine Tabelle für die Anzeige liefern, sondern nur Daten in der Datenbank verändern, muss bei der Datenpunkt-Adresse COMMAND vorangestellt werden. Sie können so zum Beispiel beim Schaltflächen-Klick in der App ein Skript mit Parametern starten. Command-Skripts sind logischerweise immer mit „Schreiben, Lesen“ zu konfigurieren und werden durch Setzen des Wertes mit „1“ ausgeführt.

Beispiel für Datenpunkt-Adresse bei MS SQL Server:

COMMAND EXEC <Prozedurname>

4 Performance-Hinweise

Ansichten-Anzeige optimieren:

Beim Anzeigen einer Ansicht in der Bedien-App werden alle relevanten Daten in ca. 1 bis 3 Sekunden vom Endpunkt via Unified-E App Manager gepollt und die Ansicht wird entsprechend aktualisiert.

Bei einer Listenansicht oder Diagrammansicht mit grossen Tabellen kann dies zu Performanceproblemen führen. Um die regelmässige Aktualisierung zu deaktivieren, muss dies explizit in den Ansichten-Eigenschaften deaktiviert werden.

Anzeigeoptionen
☐ Start-Ansicht
☐ Ansichtentitel ausblenden
☒ Periodische Aktualisierung der Datenpunkt-Werte deaktivieren. Eine Aktualisierung der Ansicht erfolgt automatisch bei Benutzereingaben

Bild

 <Nicht ausgewählt> 

Auswerterate für Trigger bei Push-Nachrichten optimieren:

Standardmässig wird im 1-Sekundentakt geprüft, ob die Trigger-Bedingung zum Versenden einer Push-Nachricht erfüllt ist. Dabei wird jedes Mal der aktuelle Wert vom Endpunkt gelesen (Performance).

Dieser Wert kann in der Spalte "Auswerterate [s]" in der Push-Nachrichten Tabelle angepasst werden, um die Datenbanklast zu verringern.

5 Sicherheitshinweise

Passwörter in Verbindungszeichenfolgen sollten immer via Platzhalter gesetzt werden.

Dazu als Passwort-Platzhalter die Zeichenfolge {Password} verwenden und im Passwort-Parameter das Passwort setzen.

Beispiel:

Adapterparameter

Verbindungstyp	Microsoft SQL Server
Verbindungskonfiguration	Verbindungszeichenfolge
Verbindungszeichenfolge	Data Source=tcp:192.168.1.29,1433;Initial Catalog=TestDB;Integrated Security=False;User ID=sa;Password={Password};Connect Timeout=5
Passwort	*****
Datenbank Zeitzone	Lokale Zeitzone

Verbindung testen...