



Kommunikationsbibliothek

# Oracle for OfficeTalk

Version 1.00



# Inhaltsverzeichnis

Einleitung.....	1
Voraussetzungen.....	1
Kommunikationsbibliothek.....	1
Allgemeine Syntax.....	1
Service Syntax.....	1
Fehlerbehandlung.....	2
Services.....	2
beginTrans.....	2
closeDatabase.....	3
closeSession.....	3
column.....	4
commitTrans.....	4
createDynaset.....	5
createSQL.....	6
eof.....	6
executeSQL.....	6
item.....	8
lastServerErr.....	8
lastServerErrText.....	9
lastServerErrReset.....	9
moveFirst.....	10
moveNext.....	10
openDatabase.....	10
param.....	11
removeParam.....	12
removeParams.....	12
resetTrans.....	13
rollback.....	14



# Einleitung

Diese Dokumentation beschreibt die Anwendung der Services zur Kommunikation mit beliebigen Oracle Datenbanken im Rahmen von Skripts und Skriptmakros. Damit werden Sie in die Lage versetzt, mit OfficeTalk beliebige Datenbanken zu bearbeiten. Wie Sie Skripts und Skriptmakros erstellen, ändern und löschen, lesen Sie in der Dokumentation [OfficeTalk Business-Process-Management](#) und [OfficeTalk Skript](#).

## Voraussetzungen

Um die Kommunikationsbibliothek für Oracle Datenbanken zu benutzen, müssen folgende Voraussetzungen gegeben sein:

- OfficeTalk wird auf einer Windows Plattform betrieben
- Oracle Version 7 oder höher ist installiert
- Oracle Objects for OLE ist installiert

## Kommunikationsbibliothek

Um die Services im Skriptmakro anzuwenden, müssen Sie die zugehörige Bibliothek als erstes mit der Anweisung **Library** "`..\Library\OracleForOfficeTalk.pcl`" laden. Wenn sich die Bibliothek in einem anderen Verzeichnis befindet, tauschen Sie **Library** entsprechend aus.

## Allgemeine Syntax

Die generelle Syntax ist in der Dokumentation [OfficeTalk Skript](#) nachzulesen. Hier wird nur die Syntax, bezogen auf die Services der Kommunikationsbibliothek, beschrieben.

## Service Syntax

In diesem Kapitel wird die allgemeine Syntax der Bibliotheksservices beschrieben. Die Syntax entspricht weitgehend der Syntax von Microsoft® VisualBasic. Nachfolgend sind Anweisungen oder Anweisungsteile im Gegensatz zu den übrigen Texten in der Schriftart `Courier` geschrieben. Die Schreibweise der folgenden Syntax:

- `<` und `>` sind zu ersetzen durch den aktuellen Namen, Bezeichner oder ähnlichem.
- `[` und `]` umschließen optionale Angaben und werden nicht geschrieben.
- `...` kennzeichnen weitere gleichartige Angaben und werden nicht geschrieben.
- Mehrere Angaben durch `|` getrennt sind alternativ und mit `{ }` umschlossen.
- Wörter ohne obige Kennzeichnung sind unverändert zu übernehmen.

## Fehlerbehandlung

Die Behandlung von Fehlern wird meist ignoriert, ist aber für den stabilen Ablauf von Programmen ein nicht unwesentlicher Faktor. Die Ausführung eines Services kann aus verschiedensten Gründen fehlschlagen. Falsches Login, falscher Spaltenname oder Tabellename, falsche SQL-Syntax, und viele Gründe mehr. Wenn Sie keine Vorkehrung dagegen treffen, erhalten Sie beim Auftreten eines Fehlers eine Skriptmakrofehlermeldung mit der entsprechenden Fehlernummer des Adapters., und die Ausführung des Skriptmakros wird abgebrochen. Sie können jedoch mit der Try-Catch-Anweisung einen Fehler auffangen, und im Skriptmakroablauf entsprechend darauf reagieren. Lesen Sie dazu den Abschnitt *lastServerErr*, *lastServerErrText*, und die Dokumentation [OfficeTalk Skript](#), Kapitel *Anweisungen - Try-Catch*

## Services

Die Services werden alle aus dem Objekt **OfficeTalkSession** angewendet. Ein Objekt **OfficeTalkSession** verwaltet eine einzelne Datenbank. Wollen Sie gleichzeitig mehrere Datenbanken bearbeiten, müssen Sie mehrere **OfficeTalkSession** Objekte verwenden. Dabei wird zuerst eine Variable mit **Dim session As SmallCOM.OracleForOfficeTalk.OfficeTalkSession** dimensioniert. Dann wird eine Session mit **session = New SmallCOM.OracleForOfficeTalk.OfficeTalkSession** angelegt. Anschließend werden mit **session.** die Services aufgerufen. Sehen Sie dazu auch die Beispiele.

## beginTrans

Der Service startet die Transaktion. Lesen Sie bitte dazu auch die Dokumentation *Oracle Objects for OLE Help*.

### Syntax

```
beginTrans
```

Der Service liefert TRUE bei ordnungsgemäßer Ausführung ansonsten FALSE.

### Bemerkung

Wenn Oracle bei der Ausführung einen Fehler (z.B. die Transaktion ist nicht mit beginTrans eingeschaltet) erkennt, liefert das Service u.U. kein FALSE als Ergebnis, sondern es wird eine Ausnahme geworfen. Kapseln Sie den Service also am besten in einen Try-Catch Block. Weitere Informationen darüber finden Sie im Kapitel *Anweisungen - Try-Catch* in der Dokumentation *OfficeTalk Skript*.

### Beispiel

```
Library "..\Library\OracleForOfficeTalk.pcl"

Dim session As SmallCOM.OracleForOfficeTalk.OfficeTalkSession

session = New SmallCOM.OracleForOfficeTalk.OfficeTalkSession
session.openDatabase("OFFICETA", connect: "OfficeTalk/OfficeTalk",
connectType: 0)
session.beginTrans
```

```
session.param("shortcut", value: "schmack", paramMode: Enum(
SmallCOM.OracleInProcServer.ParamMode, ORAPARM_INPUT ))

session.createSQL("DELETE from bearbeiter WHERE shortcut = :shortcut")
```

## closeDatabase

Der Service schließt die Datenbank. Bei Bedarf können Sie mit `openDatabase` eine weitere Datenbank zur Bearbeitung öffnen. Lesen Sie bitte dazu auch die Dokumentation *Oracle Objects for OLE Help*.

### Syntax

```
closeDatabase
```

Der Service liefert `TRUE` bei ordnungsgemäßer Ausführung ansonsten `FALSE`.

### Beispiel

```
Library "..\Library\OracleForOfficeTalk.pcl"

Dim session As SmallCOM.OracleForOfficeTalk.OfficeTalkSession

session = New SmallCOM.OracleForOfficeTalk.OfficeTalkSession
session.openDatabase("OFFICETA", connect: "OfficeTalk/OfficeTalk",
connectType: 0)

session.param("shortcut", value: "schmack", paramMode: Enum(
SmallCOM.OracleInProcServer.ParamMode, ORAPARM_INPUT ))

session.createSQL("DELETE from bearbeiter WHERE shortcut = :shortcut")
session.closeDatabase
```

## closeSession

Der Service beendet die gesamte Verarbeitung. Bei Bedarf müssen Sie mit `New` ein neues Objekt `OfficeTalkSession` erzeugen.

### Syntax

```
closeSession
```

Der Service liefert `TRUE` bei ordnungsgemäßer Ausführung ansonsten `FALSE`.

### Beispiel

```
Library "..\Library\OracleForOfficeTalk.pcl"

Dim session As SmallCOM.OracleForOfficeTalk.OfficeTalkSession

session = New SmallCOM.OracleForOfficeTalk.OfficeTalkSession
session.openDatabase("OFFICETA", connect: "OfficeTalk/OfficeTalk",
connectType: 0)

session.param("shortcut", value: "schmack", paramMode: Enum(
SmallCOM.OracleInProcServer.ParamMode, ORAPARM_INPUT ))

session.createSQL("DELETE from bearbeiter WHERE shortcut = :shortcut")
session.closeSession
```

## column

Der Service liefert den Inhalt der Spalte mit der angegebenen Nummer aus dem aktuellen Record eines vorangegangenen `createDynaset` Services. Lesen Sie bitte dazu auch die Dokumentation *Oracle Objects for OLE Help*. i.d.R. wird statt dessen der Service `item()` verwendet, da dort die Spalte mit Namen referenziert werden kann. Spezielle Select Varianten (z.B. `SELECT <Sequenzname>.nextval FROM sys.dual`) liefern jedoch nur unbenannte Spalten.

### Syntax

```
column( <Spaltennummer> )
```

1. Spaltennummer Die Spaltennummer im aktuellen Records. Die Zählbasis ist 0.  
<Ausdruck | Funktion | Literal | Variable>

### Beispiel

```
Library "..\Library\OracleForOfficeTalk.pcl"

Dim session As SmallCOM.OracleForOfficeTalk.OfficeTalkSession
Dim sequence As Integer

session = New SmallCOM.OracleForOfficeTalk.OfficeTalkSession
session.openDatabase( "OFFICETA", connect: "OfficeTalk/OfficeTalk",
connectType: 0 )
session.beginTrans

session.createDynaset( "SELECT OFFICETALK.SERVICE$SEQ.nextval FROM
sys.dual", queryOption: 0 )
session.commitTrans

If session.eof = False
    Then
        sequence = session.column( 0 )
        MsgBox( "SEQ: " & sequence )
    End If
```

## commitTrans

Der Service schließt die laufende Transaktion. Alle bisherigen Änderungen werden dauerhaft gemacht. Lesen Sie bitte dazu auch die Dokumentation *Oracle Objects for OLE Help*.

### Syntax

```
commitTrans
```

Der Service liefert `TRUE` bei ordnungsgemäßer Ausführung ansonsten `FALSE`.

### Bemerkung

Wenn Oracle bei der Ausführung einen Fehler (z.B. die Transaktion ist nicht mit `beginTrans` eingeschaltet) erkennt, liefert der Service u.U. kein `FALSE` als Ergebnis, sondern es wird eine Ausnahme geworfen. Kapseln Sie den Service also am besten in einen Try-Catch Block. Weitere Informationen darüber finden Sie im Kapitel *Anweisungen – Try-Catch* in der Dokumentation *OfficeTalk Skript*.



## Beispiel

```
Library "..\Library\OracleForOfficeTalk.pcl"

Dim session As SmallCOM.OracleForOfficeTalk.OfficeTalkSession

session = New SmallCOM.OracleForOfficeTalk.OfficeTalkSession
session.openDatabase("OFFICETA", connect: "OfficeTalk/OfficeTalk",
connectType: 0)
session.param("shortcut", value: "schmack", paramMode: Enum(
SmallCOM.OracleInProcServer.ParamMode, ORAPARM_INPUT ) )
session.beginTrans
session.createSQL("DELETE from bearbeiter WHERE shortcut = :shortcut")
session.commitTrans
```

## createDynaset

Der Service führt die angegebene SQL Anweisung aus. Es muß sich um eine `SELECT` Anweisung handeln. Lesen Sie bitte dazu auch die Dokumentation *Oracle Objects for OLE Help*.

### Syntax

```
createDynaset( <SQL Anweisung>, queryOption: <Zahl> )
```

Der Service liefert `TRUE` bei ordnungsgemäßer Ausführung ansonsten `FALSE`.

1. SQL Anweisung

Die auszuführende SQL Anweisung.

<Ausdruck | Funktion | Literal | Variable>

2. Zahl

Hier schreiben Sie derzeit immer 0.

### Bemerkung

Wenn Oracle bei der Ausführung einen Fehler (z.B. fehlerhafte SQL-Anweisung) erkennt, wird eine Ausnahme geworfen. Kapseln Sie den Service also am besten in einen `Try-Catch` Block. Weitere Informationen darüber finden Sie im Kapitel *Anweisungen – Try-Catch* in der Dokumentation *OfficeTalk Skript*.

## Beispiel

```
Library "..\Library\OracleForOfficeTalk.pcl"

Dim session As SmallCOM.OracleForOfficeTalk.OfficeTalkSession

session = New SmallCOM.OracleForOfficeTalk.OfficeTalkSession
session.openDatabase("OFFICETA", connect: "OfficeTalk/OfficeTalk",
connectType: 0)
session.param("shortcut", value: "schmack", paramMode: Enum(
SmallCOM.OracleInProcServer.ParamMode, ORAPARM_INPUT ) )
session.createDynaset("SELECT id, shortcut FROM bearbeiter",
queryOption: 0)
```

## createSQL

Der Service führt die angegebene SQL-Anweisung aus. Die Anweisung wird intern gespeichert und kann mit `executeSQL` wiederholt ausgeführt werden. Die SQL Anweisung darf keine `SELECT` Anweisung sein, da nur der Service `createDynaset` verarbeitbare Selektionsergebnisse liefert. Lesen Sie bitte dazu auch die Dokumentation *Oracle Objects for OLE Help*.

### Syntax

```
createSQL( <SQL-Anweisung> )
```

Der Service liefert `TRUE` bei ordnungsgemäßer Ausführung ansonsten `FALSE`.

1. SQL Anweisung benennt die auszuführende SQL Anweisung. Es darf sich um keine `SELECT` Anweisung handeln.  
<Ausdruck | Funktion | Literal | Variable>

### Bemerkung

Wenn kein `beginTransaction` vorangestellt ist, werden `DELETE`, `UPDATE` und `INSERT` Anweisungen automatisch mit `commitTrans` abgeschlossen. Wenn Oracle bei der Ausführung einen Fehler (z.B. fehlerhafte SQL-Anweisung) erkennt, wird eine Ausnahme geworfen. Kapseln Sie den Service also am besten in einen `Try-Catch` Block. Weitere Informationen darüber finden Sie im Kapitel *Anweisungen – Try-Catch* in der Dokumentation *OfficeTalk Skript*.

### Beispiel

```
Library "..\Library\OracleForOfficeTalk.pcl"

Dim session As SmallCOM.OracleForOfficeTalk.OfficeTalkSession

session = New SmallCOM.OracleForOfficeTalk.OfficeTalkSession
session.openDatabase("OFFICETA", connect: "OfficeTalk/OfficeTalk",
connectType: 0)

session.param("shortcut", value: "schmack", paramMode: Enum(
SmallCOM.OracleInProcServer.ParamMode, ORAPARM_INPUT ) )

session.createSQL("DELETE from bearbeiter WHERE shortcut = :shortcut")
```

## eof

Der Service meldet, ob das Ergebnisset einer `SELECT` Anweisung am Ende steht. `TRUE` bedeutet Ende erreicht. `FALSE` bedeutet Ende nicht erreicht. Lesen Sie bitte dazu auch die Dokumentation *Oracle Objects for OLE Help*.

### Syntax

```
eof
```

### Beispiel

Siehe `moveNext`

## executeSQL

Der Service führt die angegebene SQL-Anweisung aus. Die Anweisung darf keine `SELECT` Anweisung sein, da nur der Service `createDynaset` verarbeitbare Selektionsergebnisse

liefert. Wenn Sie eine leere Zeichenkette übergeben, führt der Service die zuvor mit `createSQL` ausgeführte SQL Anweisung erneut aus. Dabei muß die Anweisung nicht noch mal geparsed werden. Mit dem Service `param` veränderte Parameter werden dabei beachtet. Lesen Sie bitte dazu auch die Dokumentation *Oracle Objects for OLE Help*.

### Syntax

```
executeSQL( [<SQL-Anweisung>] )
```

Der Service liefert als Ergebnis die Anzahl der behandelten Datensätze oder bei einer leeren Anweisung 1.

1. SQL-Anweisung benennt die auszuführende SQL Anweisung. Es darf sich um keine `SELECT` Anweisung handeln.  
<Ausdruck | Funktion | Literal | Variable>

### Bemerkung

Wenn kein `beginTrans` vorangestellt ist, werden `DELETE`, `UPDATE` und `INSERT` Anweisungen automatisch mit `commitTrans` abgeschlossen. Wenn Oracle bei der Ausführung einen Fehler (z.B. fehlerhafte SQL-Anweisung) erkennt, wird eine Ausnahme geworfen. Kapseln Sie den Service also am besten in einen `Try-Catch` Block. Weitere Informationen darüber finden Sie im Kapitel *Anweisungen – Try-Catch* in der Dokumentation *OfficeTalk Skript*.

### Beispiel

```
Library "..\Library\OracleForOfficeTalk.pcl"

Dim session As SmallCOM.OracleForOfficeTalk.OfficeTalkSession

session = New SmallCOM.OracleForOfficeTalk.OfficeTalkSession
session.openDatabase("OFFICETA", connect: "OfficeTalk/OfficeTalk",
connectType: 0)
session.param("shortcut", value: "schmack", paramMode: Enum(
SmallCOM.OracleInProcServer.ParamMode, ORAPARM_INPUT ) )
session.executeSQL("DELETE from bearbeiter WHERE shortcut =
:shortcut")
```

### Beispiel 2

```
Library "..\Library\OracleForOfficeTalk.pcl"

Dim session As SmallCOM.OracleForOfficeTalk.OfficeTalkSession

session = New SmallCOM.OracleForOfficeTalk.OfficeTalkSession
session.openDatabase("OFFICETA", connect: "OfficeTalk/OfficeTalk",
connectType: 0)
session.param("shortcut", value: "schmack", paramMode: Enum(
SmallCOM.OracleInProcServer.ParamMode, ORAPARM_INPUT ) )
session.createSQL("DELETE from bearbeiter WHERE shortcut = :shortcut")
session.param("shortcut", value: "roland", paramMode: Enum(
SmallCOM.OracleInProcServer.ParamMode, ORAPARM_INPUT ) )
session.executeSQL
```

## item

Der Service liefert den Inhalt der benannten Spalte des aktuellen Records eines vorangegangenen `createDynaset Services`. Lesen Sie bitte dazu auch die Dokumentation *Oracle Objects for OLE Help*.

### Syntax

```
item( <Spaltenname> )
```

2. Spaltenname benennt die Spalte des Records.  
<Ausdruck | Funktion | Literal | Variable>

### Beispiel

Siehe `moveNext`

## lastServerErr

Der Service liefert die Nummer des zuletzt aufgetretenen Fehlers als Integer. Lesen Sie bitte dazu auch die Dokumentation *Oracle Objects for OLE Help*. Um die Fehlernummer des Service ausgeben zu können, müssen Sie die Datenbankanweisungen in einen Try-Catch Block kapseln. Damit werden Fehler des Oracle-Datenbankadapters aufgefangen. Wenn Sie keinen Try-Catch-Block verwenden, erhalten Sie bei einem Datenbankadapterfehler nur die Fehlermeldung mit der allgemeinen Fehlernummer. Lesen Sie dazu auch die Dokumentation [OfficeTalk Skript](#), Kapitel *Anweisungen - Try-Catch*.

### Syntax

```
lastServerErr
```

### Beispiel

```
Library "..\Library\OracleForOfficeTalk.pcl"

Dim session As SmallCOM.OracleForOfficeTalk.OfficeTalkSession

session = New SmallCOM.OracleForOfficeTalk.OfficeTalkSession
session.openDatabase("OFFICETA", connect: "OfficeTalk/OfficeTalk",
connectType: 0)

session.param("bearbeiter", value: "Huber", paramMode: Enum(
SmallCOM.OracleInProcServer.ParamMode, ORAPARM_INPUT ) )

Try
    session.executeSQL("DELETE from service WHERE bearbeiter =
:bearbeiter")
End Try

Catch
    MsgBox("Löschfehler: " & session.lastServerErr)
End Catch
```

## lastServerErrText

Der Service liefert die Beschreibung des zuletzt aufgetretenen Fehlers als String. Lesen Sie bitte dazu auch die Dokumentation *Oracle Objects for OLE Help*. Um einen möglichen Fehler des Service ausgeben zu können, müssen Sie die Datenbankanweisungen in einen Try-Catch Block kapseln. Damit werden Fehler des Oracle-Datenbankadapters aufgefangen. Wenn Sie keinen Try-Catch-Block verwenden, erhalten Sie bei einem Datenbankadapterfehler nur die Fehlermeldung mit der allgemeinen Fehlernummer. Lesen Sie dazu auch die Dokumentation [OfficeTalk Skript](#), Kapitel *Anweisungen - Try-Catch*.

### Syntax

```
lastServerErrText
```

### Beispiel

```
Library "..\Library\OracleForOfficeTalk.pcl"

Dim session As SmallCOM.OracleForOfficeTalk.OfficeTalkSession

session = New SmallCOM.OracleForOfficeTalk.OfficeTalkSession
session.openDatabase("OFFICETA", connect: "OfficeTalk/OfficeTalk",
connectType: 0)
session.param("bearbeiter", value: "Huber", paramMode: Enum(
SmallCOM.OracleInProcServer.ParamMode, ORAPARM_INPUT ) )

Try
    session.executeSQL("DELETE from service WHERE bearbeiter =
:bearbeiter")
End Try

Catch
    MsgBox("Löschfehler: " & session.lastServerErrText)
End Catch
```

## lastServerErrReset

Der Service löscht den zuletzt aufgetretenen Fehler der mit lastServerErr oder lastServerErrText abgeholt werden kann. Lesen Sie bitte dazu auch die Dokumentation *Oracle Objects for OLE Help*.

### Syntax

```
lastServerErrText
```

### Beispiel

```
Library "..\Library\OracleForOfficeTalk.pcl"

Dim session As SmallCOM.OracleForOfficeTalk.OfficeTalkSession

session = New SmallCOM.OracleForOfficeTalk.OfficeTalkSession
session.openDatabase("OFFICETA", connect: "OfficeTalk/OfficeTalk",
connectType: 0)
```

```
session.param("shortcut", value: "schmack", paramMode: Enum(
SmallCOM.OracleInProcServer.ParamMode, ORAPARM_INPUT ) )

session.executeSQL("DELETE from bearbeiter WHERE shortcut =
:shortcut")

session.lastServerErrReset
```

## moveFirst

Der Service positioniert das Ergebnisset eines vorangegangenen createDynaset Services auf den ersten Record. Lesen Sie bitte dazu auch die Dokumentation *Oracle Objects for OLE Help*.

### Syntax

MoveFirst

Der Service liefert TRUE bei ordnungsgemäßer Ausführung ansonsten FALSE.

### Beispiel

Siehe moveNext

## moveNext

Der Service positioniert das Ergebnisset eines vorangegangenen createDynaset Services auf den nächsten Record. Lesen Sie bitte dazu auch die Dokumentation *Oracle Objects for OLE Help*.

### Syntax

MoveNext

Der Service liefert TRUE bei ordnungsgemäßer Ausführung ansonsten FALSE.

### Beispiel

```
Library "..\Library\OracleForOfficeTalk.pcl"
```

```
Dim session As SmallCOM.OracleForOfficeTalk.OfficeTalkSession
```

```
session = New SmallCOM.OracleForOfficeTalk.OfficeTalkSession
session.openDatabase("OFFICETA", connect: "OfficeTalk/OfficeTalk",
connectType: 0)
session.createDynaset("SELECT id, shortcut FROM bearbeiter",
queryOption: 0)
While session.eof = False
    MsgBox("ID: " & session.item("id") & " Name: " &
    session.item("shortcut")

    session.moveNext
Wend
```

## openDatabase

Mit dem Service wird eine Datenbank zur weiteren Bearbeitung geöffnet. Falls die Session noch nicht geöffnet ist, wird auch sie geöffnet.

## Syntax

```
openDatabase ( <Servicename> , connect: <Login>/<Passwort> ,
connectType: <eine Zahl> )
```

Der Service liefert TRUE bei ordnungsgemäßer Ausführung ansonsten FALSE.

1. Servicename gibt den Namen des Datenbankservices an. Auf Windows Plattformen ist das der Name des Systemdienstes, unter dem die gewünschte Datenbank erreichbar ist.  
<Ausdruck | Funktion | Literal | Variable>
2. Login  
Gibt den Benutzernamen zur Datenbank anmeldung an  
<Ausdruck | Funktion | Literal | Variable>
3. Passwort  
Gibt das Passwort zur Datenbank anmeldung an  
<Ausdruck | Funktion | Literal | Variable>
4. eine Zahl  
Eine Zahl. Das Argument wird derzeit nicht verwendet.

## Bemerkung

Wenn Oracle bei der Ausführung einen Fehler (z.B. falsches Passwort) erkennt, wird eine Ausnahme geworfen. Kapseln Sie den Service also am besten in einen Try-Catch Block. Weitere Informationen darüber finden Sie im Kapitel *Anweisungen – Try-Catch* in der Dokumentation *OfficeTalk Skript*.

## Beispiel

```
Library "..\Library\OracleForOfficeTalk.pcl"

Dim session As SmallCOM.OracleForOfficeTalk.OfficeTalkSession

session = New SmallCOM.OracleForOfficeTalk.OfficeTalkSession
session.openDatabase("OFFICETA", connect: "OfficeTalk/OfficeTalk",
connectType: 0)
```

## param

Mit dem Service geben Sie für eine nachfolgende SQL-Anweisung im Service createDynaset, executeSQL oder createSQL benannte Parameter an.

## Syntax

```
param( <Parametername>, value: <Parameterwert>, paramMode: Enum(
SmallCOM.OracleInProcServer.ParamMode, <Modus>) )
```

Der Service liefert TRUE bei ordnungsgemäßer Ausführung ansonsten FALSE.

1. Parametername benennt den Parameter.  
<Ausdruck | Funktion | Literal | Variable>
2. Parameterwert  
Ist der Wert mit dem der Parameter in der nachfolgenden Service createDynaset angewandt wird.  
<Ausdruck | Funktion | Literal | Variable>
3. Modus  
Benennt den Modus, unter dem der Parameter verwendet wird. Lesen Sie dazu weiteres in der Dokumentation *Oracle Objects for OLE Help* nach.

Für Modus schreiben Sie

ORAPARM\_INPUT Der Parameter wird als Eingabe verwendet.

ORAPARM\_OUTPUT Der Parameter wird als Ausgabe verwendet.

ORAPARM\_BOTH Der Parameter wird als Eingabe und Ausgabe verwendet.

### Bemerkung

Wenn sie einen Parameterwert aktualisieren wollen, müssen Sie diesen zuerst mit dem Service removeParam entfernen bevor Sie ihn neu besetzen können.

### Beispiel

```
Library "..\Library\OracleForOfficeTalk.pcl"

Dim session As SmallCOM.OracleForOfficeTalk.OfficeTalkSession

session = New SmallCOM.OracleForOfficeTalk.OfficeTalkSession
session.openDatabase("OFFICETA", connect: "OfficeTalk/OfficeTalk",
connectType: 0)
session.param("shortcut", value: "schmack", paramMode: Enum(
SmallCOM.OracleInProcServer.ParamMode, ORAPARM_INPUT ) )
```

## removeParam

Der Service entfernt einen mit param eingefügten Parameter. Lesen Sie bitte dazu auch die Dokumentation *Oracle Objects for OLE Help*.

### Syntax

```
removeParam( <Name oder Position> )
```

Der Service liefert TRUE bei ordnungsgemäßer Ausführung ansonsten FALSE.

1. Name oder Position benennt den Namen oder die Position des Parameters.  
<Ausdruck | Funktion | Literal | Variable>

### Beispiel

```
Library "..\Library\OracleForOfficeTalk.pcl"

Dim session As SmallCOM.OracleForOfficeTalk.OfficeTalkSession

session = New SmallCOM.OracleForOfficeTalk.OfficeTalkSession
session.openDatabase("OFFICETA", connect: "OfficeTalk/OfficeTalk",
connectType: 0)
session.param("shortcut", value: "schmack", paramMode: Enum(
SmallCOM.OracleInProcServer.ParamMode, ORAPARM_INPUT ) )
session.removeParam("shortcut")
```

## removeParams

Der Service entfernt alle mit param eingefügten Parameter. Lesen Sie bitte dazu auch die Dokumentation *Oracle Objects for OLE Help*.



**Syntax**

```
removeParams
```

Der Service liefert TRUE bei ordnungsgemäßer Ausführung ansonsten FALSE.

**Beispiel**

```
Library "..\Library\OracleForOfficeTalk.pcl"

Dim session As SmallCOM.OracleForOfficeTalk.OfficeTalkSession

session = New SmallCOM.OracleForOfficeTalk.OfficeTalkSession
session.openDatabase("OFFICETA", connect: "OfficeTalk/OfficeTalk",
connectType: 0)
session.param("shortcut", value: "schmack", paramMode: Enum(
SmallCOM.OracleInProcServer.ParamMode, ORAPARM_INPUT ))
session.removeParams
```

**resetTrans**

Der Service beendet alle Transaktionen und verwirft alle Datenbankänderungen der laufenden Transaktionen. Lesen Sie bitte dazu auch die Dokumentation *Oracle Objects for OLE Help*.

**Syntax**

```
resetTrans
```

Der Service liefert TRUE bei ordnungsgemäßer Ausführung ansonsten FALSE.

**Bemerkung**

Wenn Oracle bei der Ausführung einen Fehler (z.B. die Transaktion ist nicht mit beginTrans eingeschaltet) erkennt, liefert der Service u.U. kein FALSE als Ergebnis, sondern es wird eine Ausnahme geworfen. Kapseln Sie den Service also am besten in einen Try-Catch Block. Weitere Informationen darüber finden Sie im Kapitel *Anweisungen – Try-Catch* in der Dokumentation *OfficeTalk Skript*.

**Beispiel**

```
Library "..\Library\OracleForOfficeTalk.pcl"

Dim session As SmallCOM.OracleForOfficeTalk.OfficeTalkSession

session = New SmallCOM.OracleForOfficeTalk.OfficeTalkSession
session.openDatabase("OFFICETA", connect: "OfficeTalk/OfficeTalk",
connectType: 0)
session.beginTrans
session.param("shortcut", value: "schmack", paramMode: Enum(
SmallCOM.OracleInProcServer.ParamMode, ORAPARM_INPUT ))
session.createSQL("DELETE from bearbeiter WHERE shortcut = :shortcut")
session.resetTrans
```

## rollback

Der Service beendet die laufende Transaktion und verwirft alle seit `beginTransaction` erfolgten Datenbankänderungen. Lesen Sie bitte dazu auch die Dokumentation *Oracle Objects for OLE Help*.

### Syntax

```
rollback
```

Der Service liefert `TRUE` bei ordnungsgemäßer Ausführung ansonsten `FALSE`.

### Bemerkung

Wenn Oracle bei der Ausführung einen Fehler (z.B. die Transaktion ist nicht mit `beginTransaction` eingeschaltet) erkennt, liefert das Service u.U. kein `FALSE` als Ergebnis, sondern es wird eine Ausnahme geworfen. Kapseln Sie den Service also am besten in einen Try-Catch Block. Weitere Informationen darüber finden Sie im Kapitel *Anweisungen – Try-Catch* in der Dokumentation *OfficeTalk Skript*.

### Beispiel

```
Library "..\Library\OracleForOfficeTalk.pcl"

Dim session As SmallCOM.OracleForOfficeTalk.OfficeTalkSession

session = New SmallCOM.OracleForOfficeTalk.OfficeTalkSession
session.openDatabase("OFFICETA", connect: "OfficeTalk/OfficeTalk",
connectType: 0)
session.beginTransaction
session.param("shortcut", value: "schmack", paramMode: Enum(
SmallCOM.OracleInProcServer.ParamMode, ORAPARM_INPUT ))
session.createSQL("DELETE from bearbeiter WHERE shortcut = :shortcut")
session.rollback
```