

# NetSlave cKonto.NET Anleitung

cKonto validiert Kontonummern deutscher Banken und Sparkassen in Verbindung mit der Bankleitzahl durch die Berechnung einer Prüfziffer nach den [Berechnungsmethoden der Deutschen Bundesbank](#). cKonto kann somit eine Aussage treffen, ob die Bankverbindung theoretisch existieren kann.

cKonto wurde insbesondere für den Einsatz in eCommerce- und Onlineshopsystemen entwickelt. Durch ein Interface im CGI-Standard (Format der GET-Datenübergabe) kann cKonto schnell in jede Applikation integriert werden. Hier finden Sie die Anleitung für die Integration der cKonto Binary in Windows Anwendungen über eine .NET DLL.

cKonto ist Shareware. Ohne Eingabe eines gültigen Lizenzschlüssels ist die Funktion auf die ersten 10 der über 125 verschiedenen Prüfalgorithmen beschränkt. Die Nutzung von cKonto als Shareware ist durch Eingabe des Lizenzschlüssels "demo" möglich und nicht weiter beschränkt.

## Inhalt

---

- Einbinden der cKonto.NET.dll
- Kontonummer / BLZ prüfen
- IBAN prüfen
- Bankensuche

## Einbinden der cKonto.NET.dll

---

### 1. Verweis auf die cKonto.NET.dll hinzufügen

### 2. Namespace laden:

```
using NetSlave.cKontoNET;
```

### 3. Objekt anlegen

```
var cKonto = new cKontoNET();
```

### 4. Lizenzkey und Pfad zur cKonto.exe setzen:

```
cKonto.SetLicenseKey("demo");  
cKonto.SetExecutablePath("cKonto.exe");
```

## Kontonummer / Bankleitzahl prüfen (VerifyAccount)

---

cKonto prüft nach dem Aufruf die Plausibilität der eingegebenen Daten, sucht die Bank und die von ihr verwendete Prüfmethode, führt die Prüfzifferberechnung durch. Das Ergebniss ist ein CkontoResult Objekt das die Rückgabewerte (Status und Bankdaten) enthält.

### Die verfügbaren Parameter:

name	(string)	=	Bezeichnung der Bank - max. 58 Zeichen
zip	(string)	=	Postleitzahl der Bank - max. 5 Zeichen

### Beispiel (C#):

```
string Kto = "12345";
string Blz = "12345678";

var Result = cKonto.VerifyAccount(Kto, Blz);

this.tbVerifyBankname.Text = Result.BankName;
this.tbVerifyZIP.Text = Result.ZipCode.ToString();

if (Result.Status == ExitCodes.AccountNumberValid)
{
    // Konto ist gültig
}
else
{
    // Konto ist ungültig (ggf. Rückgabecode auswerten)
}
```

## IBAN-Nummer prüfen (VerifyIBAN)

---

Mit cKonto können Sie auch die IBAN-Nummer (Format: "DE" + 22-stellig numerisch) für deutsche Konten prüfen lassen.

Die IBAN beinhaltet die Kontonummer, Bankleitzahl sowie eine 2-stellige Prüfziffer. Dadurch können Tippfehler bei IBAN-Prüfungen mit einer theoretischen Sicherheit von 99,9% erkannt und von cKonto zurückgewiesen werden.

### Die verfügbaren Parameter:

IBAN (string) = IBAN  
BIC (string) = BIC (optional)

### Beispiel (C#):

```
string IBAN = "DE07100500006603032331";

var Result = cKonto.VerifyIBAN(IBAN);

this.tbIBANBankName.Text = Result.BankName;
this.tbIBANZip.Text = Result.ZipCode.ToString();
this.tbIBANLocation.Text = Result.Location;

if (Result.Status == ExitCodes.AccountNumberValid)
{
    // Konto ist gültig
}
else
{
    // Konto ist ungültig (ggf. Rückgabecode auswerten)
}
```

## IBAN Generierung (GenerateIBAN)

---

Über diese Funktion ist die Generierung von IBANs aus Kontonummer und Bankleitzahl für Deutschland, gemäß den IBAN-Regeln der Bundesbank, möglich.

### Die verfügbaren Parameter:

AccountNumber (string) = Kontonummer  
BankCode (string) = Bankleitzahl (BLZ)  
CountryCode (string) = Länderkürzel (momentan nur „DE“ möglich; optional)

### Beispiel (C#):

```
string Kto = "12345";  
string Blz = "12345678";  
  
var Result = cKonto.GenerateIBAN(Kto, Blz);  
  
this.tbIBAN.Text = Result.IBAN;  
this.tbBIC.Text = Result.BIC;  
  
if (Result.Status == ExitCodes.IBANGenerationSuccessfull)  
{  
    // IBAN wurde erfolgreich generiert  
}  
else  
{  
    // Bei der Generierung gab es Probleme (ggf. Fehlercode auswerten)  
}
```

## Bankensuche (SearchBank)

---

Mit cKonto können Sie nun auch Banken nach verschiedenen Kriterien suchen lassen. Für die Nutzung dieser Funktion ist kein Lizenzschlüssel erforderlich.

### Die verfügbaren Parameter:

BankName (string) = Bank name  
BankCode (string) = Bank code (BLZ)  
ZIP (string) = ZIP (PLZ)  
BankLocation (string) = Bank location  
MaxResults (int) = Result limit

Nach dem Aufruf der *SearchBank-Methode* erhalten Sie ein *CkontoSearchResult* Objekt, das den Exit-Code (Status) und eine Liste der gefundenen Banken enthält.

### Das Rückgabe Objekt (CkontoSearchResult):

Status (ExitCodes) = cKonto Rückgabe-Codes  
SearchResults (List<CKontoResult>) = SearchResults Liste

## Beispiel (C#):

```
string Name = "Sparkasse";
string BankCode = "12345678";
string ZIP = "";
string Location = "Berlin";
int Max = 20;

var Res = cKonto.SearchBank(Name, BankCode, ZIP, Location, Max);

if (Res.Status == ExitCodes.SearchSuccessful
    || Res.Status == ExitCodes.SearchOutputLimitReached)
{
    // Suche war erfolgreich

    // Suchergebnisse durchgehen
    foreach (var SearchResult in Results.SearchResults)
    {
        MessageBox.Show(SearchResult.BankName);
    }
}
else {
    // Suche war nicht erfolgreich (ggf. Rückgabecode auswerten)
}
```

## cKonto Rückgabe Objekt (CkontoResult)

---

Bei dem *CkontoResult*-Objekt handelt es sich um einen Container, der die Rückgabe Werte einer cKonto Abfrage enthält.

### Folgende Parameter können ausgelesen werden:

Status	(ExitCodes)	= cKonto Rückgabe-Codes
BankName	(string)	= Bank name
AccountNumber	(string)	= Kontonummer
BankCode	(string)	= Bank code (BLZ)
ZipCode	(string)	= ZIP (PLZ)
Location	(string)	= Bank Location
Country	(string)	= Bank Country
BIC	(string)	= BIC
IBAN	(string)	= IBAN

## cKonto Rückgabe-Codes (ExitCodes)

---

Mithilfe der *ExitCodes (Enum)* lässt sich sehr einfach überprüfen ob die Abfrage erfolgreich war oder ob ein Fehler aufgetreten ist.

### Folgende Status Codes sind definiert:

AccountNumberInvalid	Die Kontonummer ist ungültig, sie kann von diesem Institut nicht vergeben werden
AccountNumberValid	Die Kontonummer oder IBAN ist gültig, sie kann von diesem Institut vergeben werden
ErrorBankCodeNotFound	Die Bankleitzahl wurde in der Datenbank nicht gefunden
ErrorBICNotFound	Der BIC wurde nicht gefunden.

ErrorGeneralInputError	Genereller Eingabefehler des Übergabeparameters
ErrorInvalidParameterFormat	Fehler im Format des Übergabeparameters
ErrorNotAvailableInDemoMode	Die Prüfmethode ist im Demonstrations-Modus nicht verfügbar
ErrorNoVerifyMethodOrSpecialAccount	Die Kontonummer kann nicht geprüft werden, da die Bank entweder keine Prüfziffern verwendet oder es sich um eine spezielle Kontonummer handelt
ErrorWrongCountryCode	Eingabefehler beim Ländercode.
ErrorWrongIBANFormat	Eingabefehler IBAN
ErrorWrongInputAccountNumber	Fehler bei der Eingabe der Kontonummer (Sie ist entweder zu lang oder enthält Buchstaben)
ErrorWrongInputBankCode	Fehler bei der Eingabe der Bankleitzahl (Sie ist entweder nicht 8-stellig oder enthält Buchstaben)
ErrorWrongInputBankName	Eingabefehler Bank Name
ErrorWrongInputBIC	Fehler bei der Eingabe der BIC.
ErrorWrongInputFormat	Fehler: Eingabestring hat falsches Format
ErrorWrongInputIBAN	Eingabefehler IBAN
ErrorWrongInputLocation	Eingabefehler Ort
ErrorWrongInputParameters	Fehler: Eingabeformat unbekannt (z.B. mehr als 3 Parameter)
ErrorWrongInputSearchLimit	Eingabefehler Max
ErrorWrongInputZIP	Eingabefehler Postleitzahl
IBANGenerationSuccessfull	IBAN wurde erfolgreich erzeugt
IBANNotClearManualProcessingRecommended	IBAN nicht eindeutig, Überprüfung empfohlen
IBANValidBICWrongOrUntestable	Die IBAN ist korrekt, BIC nicht prüfbar oder unpassend.
IBANWrong	IBAN ungültig
SearchNoResults	Keine Fehler, Suche ergab keine Treffer
SearchOutputLimitReached	Ausgabe-Begrenzung (Max.) erreicht
SearchSuccessful	Keine Fehler, Suche erfolgreich
UnknownExitCode	Unbekannter Exit-Code

Letzte Änderung: 29. Mai 2013 – cKonto.NET DLL - Version: 3.0.0.0  
 Copyright © 2013 - NetSlave GmbH, [www.netslave.de](http://www.netslave.de) - [www.ckonto.de](http://www.ckonto.de)