

# Übertragung von Terminen und Baustellen aus KWP nach virtic

In dieser Dokumentation wird die Anwendung und die Konfiguration der Programme beschrieben, die für die Übertragung der Baustellen und Termine aus der KWP SQL-Datenbank an das virtic-System verwendet werden können.

1	Voraussetzungen	2
2	Programmdateien	2
2.1	Dateien im Ordner fehlzeit	2
2.2	Dateien im Ordner einsatzplan	3
3	Programm KWP2VirticFehl	3
3.1	Funktionsbeschreibung	3
3.2	Fehlerprotokollierung	4
3.3	Schematischer Programmablauf	5
3.4	Konfigurationsdatei KWP2VirticFehl.exe.config	6
3.5	Aufbau KWP2VirticFehl.exe.config	6
3.5.1	<configuration><configSections>	6
3.5.2	<configuration><userSettings>	6
3.5.3	<configuration><applicationSettings>	6
3.6	Der Aufbau der SQL-Abfrage	8
3.6.1	Beispiel für den Inhalt einer SQL Abfragedatei:	9
4	Programm KWP2VirticEplBst	9
4.1	Funktionsbeschreibung	9
4.2	Fehlerprotokollierung	10
4.3	Schematischer Programmablauf	11
4.4	Konfigurationsdatei KWP2VirticEplBst.exe.config	12
4.4.1	<configuration> <configSections>	12
4.4.2	<configuration><userSettings>	12
4.4.3	<configuration><applicationSettings>	12
4.5	Der Aufbau der SQL-Abfrage	15
4.5.1	Beispiel für die Abfrage der Projektabelle	15
4.5.2	Beispiel für die Abfrage der Regietabelle	16
4.5.3	Der erweiterte Programmaufruf	17
4.5.4	Sonstiges	18
5	Kontakt	18

## 1 Voraussetzungen

- Auf den Systemen, auf denen die Programme ausgeführt werden sollen, muss das .Net-Framework© Version 3.5 oder höher installiert sein.
- Das Programm muss eine Verbindung zu der KWP SQL-Server-Datenbank aufbauen können, dafür muss eine ODBC-Datenquelle im Windows System definiert worden sein.
- Es muss eine HTTP-Verbindung ins Internet über den PORT 80 oder 443 aufgebaut werden können.

## 2 Programmdateien

Die Programme und die Konfigurationsdateien werden in Form einer komprimierten Archivdatei ausgeliefert (Download: [virtic.com/kwp](http://virtic.com/kwp)). In dieser Datei existieren wiederum die beiden Ordner `fehlzeit` und `einsatzplan`, in welchen sich jeweils eine ausführbare Datei und weitere Konfigurations- und Hilfsdateien befinden.

### 2.1 Dateien im Ordner `fehlzeit`

Dateiname	Beschreibung
KWP2VirticFehl.exe	Die ausführbare EXE-Datei
KWP2VirticFehl.XmlSerializers.dll	Von der EXE-Datei verwendete Datei
KWP2VirticFehl.exe.config	Vom Programm verwendete Konfigurationsdatei im Format XML. Diese muss den Bedürfnissen des Kunden angepasst werden
KrankheitUrlaubtermine.sql	Eine Musterdatei für eine Datenbankabfrage in SQL Syntax. Dies muss vom Benutzer hinsichtlich der korrekten Verwendung von Datenbank- und Tabellennamen kontrolliert und ggf. korrigiert werden und dann <b>unter einem anderen Namen</b> abgespeichert werden.

## 2.2 Dateien im Ordner einsatzplan

Dateiname	Beschreibung
KWP2VirticEplBst.exe	Die ausführbare EXE-Datei
KWP2VirticEplBst.XmlSerializers.dll	Von der EXE-Datei verwendete Datei
KWP2VirticEplBst.exe.config	Vom Programm verwendete Konfigurationsdatei im Format XML. Diese muss den Bedürfnissen des Kunden angepasst werden
einsatzplanProjekt.sql einsatzplanRegie.sql einsatzplanWartung.sql	Musterdateien für Datenbankabfragen in SQL-Syntax. Diese müssen vom Benutzer hinsichtlich der korrekten Verwendung von Datenbank- und Tabellennamen kontrolliert und ggf. korrigiert werden und <b>dann unter einem anderen Namen</b> abgespeichert werden.

Nach dem Entpacken des komprimierten Archivs können die Programme **KWP2VirticFehl** oder **KWP2VirticEplBst** direkt gestartet werden. Da es sich um Konsolenprogramme handelt, die kein Windows-Fenster öffnen, kann der Aufruf entweder auf Kommandozeilenebene durch die Eingabe des Programmnamens oder durch den Aufruf aus einer Stapeldatei heraus erfolgen. Dabei ist es wichtig, zuvor in das Verzeichnis zu wechseln, in dem das jeweilige Programm installiert wurde, damit das Programm die Konfigurationsdatei (.config) finden und einlesen kann.

## 3 Programm KWP2VirticFehl

### 3.1 Funktionsbeschreibung

Das Programm **KWP2VirticFehl** liest aus einer KWP SQL-Datenbank

- Termine und
- die Zeiträume von
  - Urlauben,
  - Krankheiten und
  - sonstigen Fehlzeiten

für einzelne Mitarbeiter. Der zu filternde Zeitraum für die Erzeugung der Datenmenge wird durch einen Parameter in der Konfigurationsdatei (**KWP2VirticFehl.exe.config**) bestimmt.

Dieser Wert und eine Abfragezeichenkette in SQL naher Syntax, welche aus einer Datei entnommen wird, deren Name ebenfalls in der Konfigurationsdatei enthalten ist, wird dann zu einer SQL Abfrage kombiniert und anschließend ausgeführt.

Nachdem die Datenmenge nach der Abfrage zur Verfügung steht, wird diese per Aufruf des virtic-Webservices an den virtic-Server übertragen und dort in der Datenbank des Kunden abgelegt, und steht sofort im Rahmen der Fehlzeitenplanung und -verwendung für die Zeitwirtschaft zur Verfügung.

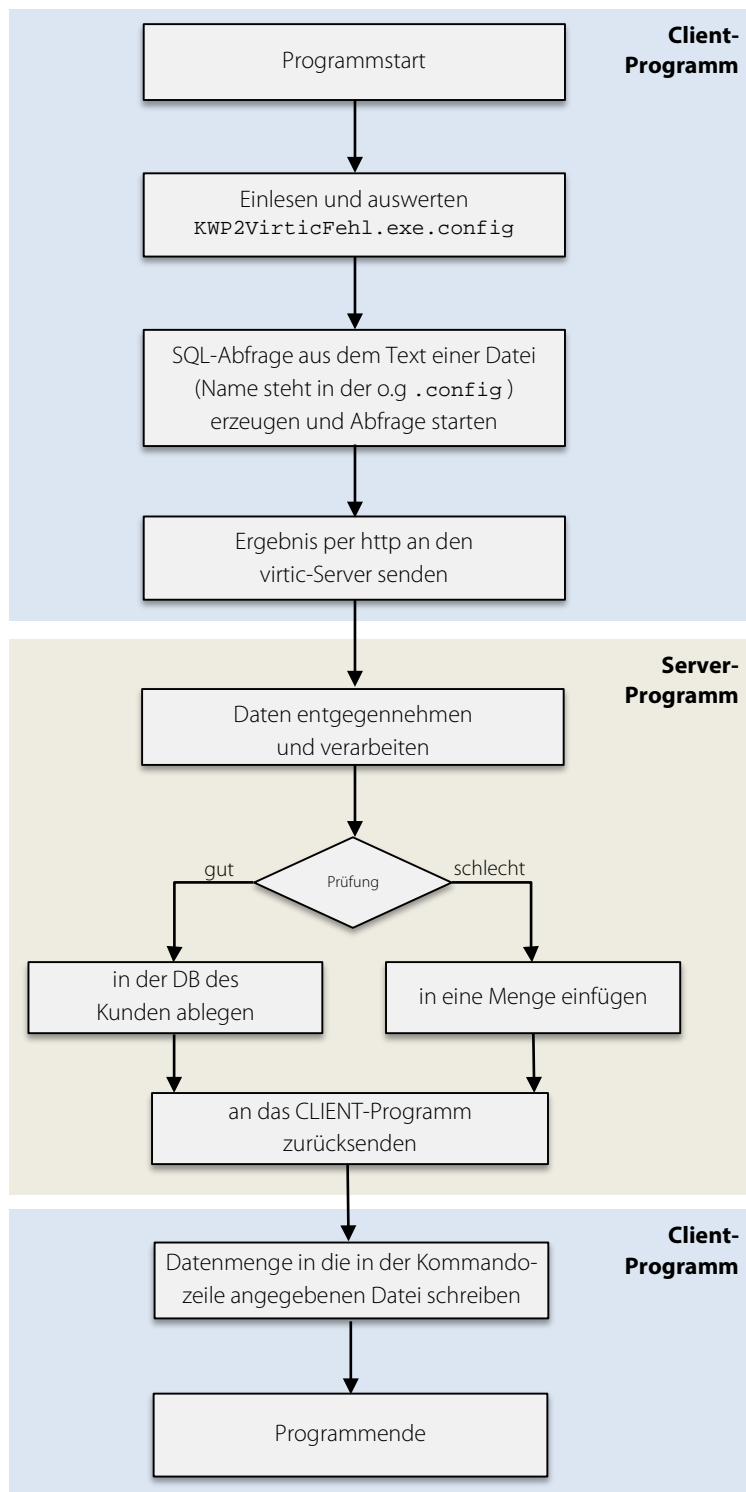
### **3.2 Fehlerprotokollierung**

Im Rahmen der Übertragung der Termine erzeugt der Server eine Liste der Datensätze, welche nicht gespeichert werden konnten. Diese werden in eine Datei geschrieben, deren Name beim Programmaufruf angegeben werden muss.

Die vollständige Syntax beim Programmaufruf lautet dann

**KWP2VirticFehl.exe fehlerzeilen.dat**

### 3.3 Schematischer Programmablauf



### 3.4 Konfigurationsdatei KWP2VirticFehl.exe.config

### 3.5 Aufbau KWP2VirticFehl.exe.config

Die Konfigurationsdatei verwendet die XML-Syntax.

Sie ist durch 3 Hauptelemente gegliedert:

```
<configuration> <configSections>
```

und

```
<configuration> <userSettings>
```

und

```
<configuration> <applicationSettings>
```

#### 3.5.1 <configuration><configSections>

Innerhalb des Elementes <configuration><configSections> dürfen keine Änderungen vorgenommen werden.

#### 3.5.2 <configuration><userSettings>

Die Elemente /Attribute Innerhalb des Elementes <configuration><userSettings>

```
<setting name="BenutzerId" ..>  
    <value>Benutzername</value>
```

Der Wert **Benutzername** muss durch den Wert ersetzt werden, den sie vom Support als Benutzernamen genannt bekommen haben.

```
<setting name="Passwort" ...>  
    <value>00000</value>
```

Der Wert **Passwort** muss durch den Wert ersetzt werden, den sie vom Support als Kennwort genannt bekommen haben.

#### 3.5.3 <configuration><applicationSettings>

Die Elemente innerhalb des Elementes <configuration><applicationSettings>

```
<setting name="FirmenId" serializeAs="String">  
    <value>00000000</value>
```

Tragen sie hier die vom Support mitgeteilte eindeutige ID der der Firma mit, für die Daten übertragen werden.

```
<setting name="UseProxy" serializeAs="String">  
    <value>False</value>
```

Hier kann durch den Wert `True` die Verwendung eines Proxy-Servers bestimmt werden, der die http-Anfrage weiterleitet. Ob es in ihrem Umfeld einen Proxy-Server gibt, der für den Zugang zum Internet verwendet werden muss und wie dieser heißt und mit welchen Anmeldedaten man sich mit ihm verbinden muss, wird man beim Netzwerkadministrator erfragen können.

```
<setting name="ProxyUrl" serializeAs="String">  
    <value>http://einProxy:8081</value>
```

Der Name und das Protokoll für die Verbindung mit dem Proxy-Server.  
Dieser Wert ist nur dann relevant, wenn der Wert `UseProxy` gleich `True` ist.

```
<setting name="KWP2VirticFehl_de_hasenpusch_virtic_virsoapmanagedService "  
    <value>http://www.virtic24.de/wsa/wsdl</value>
```

Der Name des virtic-Webservice-Servers mit der Angabe eines Pfades, wie er bei der Verbindung zum Webservice verwendet werden soll. Hier dürfen nur auf Anweisung des virtic-Supports Änderungen vorgenommen werden.

```
<setting name="Zeichensatz" serializeAs="String">  
    <value>ISO8859-1</value>
```

Der Zeichensatz der Datenbank

```
<setting name="fegObjId" serializeAs="String">  
    <value>U=45823609,K=45823601</value>
```

Um die referentielle Integrität der Daten zu erhalten, muss jedem Datensatz ein Fremdschlüssel für den Verweis auf die in virtic existierenden Fehlgründe zugewiesen werden. Zurzeit müssen für die Datensätze welche, im Statusfeld mit `U` oder `K` gekennzeichnet sind, die entsprechenden ID-Werte für den Verweis auf die Fehlgründe angegeben werde. Die Werte sind beim virtic-Support zu erfragen.

```
<setting name="DBConn" serializeAs="String">  
    <value>DSN=kwp</value>
```

Hier muss der „**System DataSource Name**“ der ODBC Datenquelle, die vom Benutzer auf dem selben System definiert wurde, angegeben werden.

Die Datenquelle muss eine Windows-Authentifizierung verwenden, damit die Abfragen korrekt ausgeführt werden können. Die Verwendung einer Benutzer-Datenquelle ist NICHT möglich.

```
<setting name="SQLFile" serializeAs="String">  
    <value>Abfrage.sql</value>
```

Der Name der Datei, welcher die SQL-Abfrage enthält

```
<setting name="Von2Bis" serializeAs="String">  
    <value>6</value>
```

Anzahl der zusätzlichen Tage vom Zeitpunkt des Programmstarts aus gerechnet, auf die sich der Abfragezeitraum erstrecken soll.

### 3.6 Der Aufbau der SQL-Abfrage

Die SQL Abfrage muss eine Tabelle mit den folgenden Spalten erzeugen

FeldName	Typ	/Beschreibung
TerminID	String	Dieser Wert ist ein Schlüsselwert für das Erzeugen/Aktualisieren eines Datensatzes in der virtic-Datenbank
PersNr	String	Die Personalnummer des Mitarbeiters aus dem KWP System. Der Mitarbeiter muss im virtic-System unter dieser externen Nummer vorhanden sein.
TerminVon	Datetime	Beginn des Fehlens
TerminBis	Datetime	Ende des Fehlens
status	String	κ oder υ



### 3.6.1 Beispiel für den Inhalt einer SQL Abfragedatei:

```
SELECT [BNWINS].[dbo].MontStrgTermine.TerminID, [BNWINS].[dbo].MontStrgPers.PersNr,
       [BNWINS].[dbo].MontStrgTermine.Typ,
[BNWINS].[dbo].MontStrgTermine.TerminVon,
       [BNWINS].[dbo].MontStrgTermine.TerminBis,
       status = [BNWINS].[dbo].MontStrgTermine.Status

FROM   [BNWINS].[dbo].MontStrgTermine LEFT OUTER JOIN
       [BNWINS].[dbo].MontStrgPers ON [BNWINS].[dbo].MontStrgTermine.TerminID =
[BNWINS].[dbo].MontStrgPers.fkTerminID

WHERE  (([BNWINS].[dbo].MontStrgTermine.Typ = 'U') OR
([BNWINS].[dbo].MontStrgTermine.Typ = 'K'))

      AND (([BNWINS].[dbo].MontStrgTermine.TerminVon >= '#VON# 00:00:00') AND
([BNWINS].[dbo].MontStrgTermine.TerminVon <= '#BIS# 23:59:59'))
```

Die Zeichenketten #von# und #bis# werden vor der Ausführung der Abfrage in eine Datumszeichenkette des Formats tt.mm.yyyy ersetzt. Der #von# Wert entspricht dabei dem Datum zum Zeitpunkt der Ausführung des Programm, der #bis# Wert entspricht dem vorgenannten Datum plus die Anzahl der Tage, die in der Konfigurationsdatei angegeben worden sind.

## 4 Programm KWP2VirticEplBst

### 4.1 Funktionsbeschreibung

Das Programm **KWP2VirticEplBst** liest aus einer KWP SQL-Datenbank Termine, die Zeiträume von Einsätzen auf Baustellen für einzelne Mitarbeiter beschreiben. Der zu filternde Zeitraum für die Erzeugung der Datenmenge wird durch einen Parameter in der Konfigurationsdatei (**KWP2VirticEplBst.exe.config**) bestimmt.

Dieser Wert und eine Abfragezeichenkette in SQL-naher Syntax, welche aus einer Datei entnommen wird, deren Name ebenfalls in der Konfigurationsdatei enthalten ist, wird dann zu einer SQL-Abfrage kombiniert und anschließend ausgeführt.

Sobald die Datenmenge nach der Abfrage zur Verfügung steht, wird diese per Aufruf des virtic- Webservices an den virtic-Server übertragen und dort in der Datenbank des Kunden abgelegt, und steht sofort im Rahmen der Einsatzplanung und -verwendung für die Zeitwirtschaft zur Verfügung.

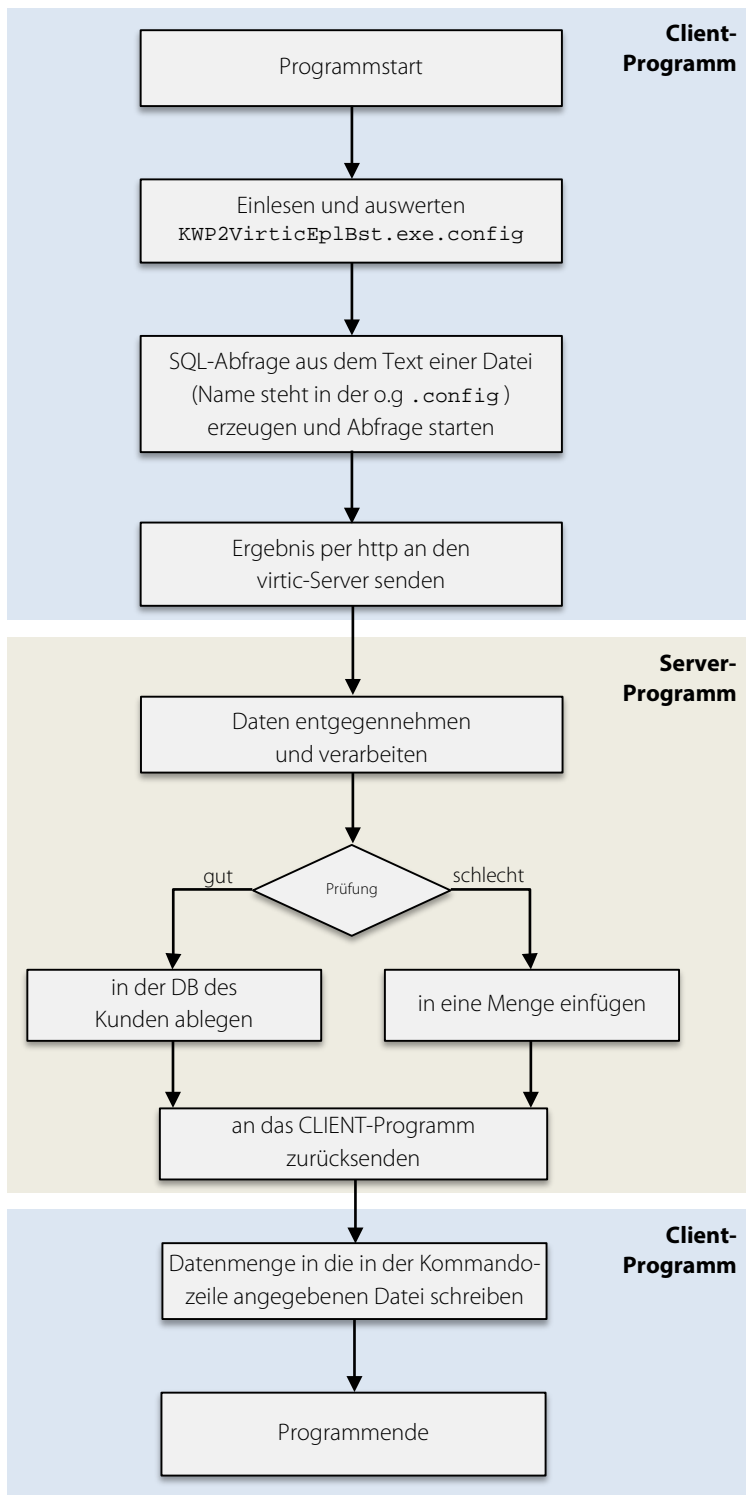
## 4.2 Fehlerprotokollierung

Im Rahmen der Übertragung der Einsatzplanung erzeugt der Server eine Liste der Datensätze, welche nicht gespeichert werden konnten. Diese werden in eine Datei geschrieben, deren Name beim Programmaufruf angegeben werden muss.

Die vollständige Syntax beim Programmaufruf lautet dann

**KWP2VirticEplBst.exe fehlerzeilen.dat**

### 4.3 Schematischer Programmablauf



## 4.4 Konfigurationsdatei KWP2VirticEplBst.exe.config

Die Konfigurationsdatei verwendet die XML-Syntax.

Sie ist durch 3 Hauptelemente gegliedert

```
<configuration> <configSections>
```

und

```
<configuration> <userSettings>
```

und

```
<configuration> <applicationSettings>
```

### 4.4.1 <configuration><configSections>

Innerhalb des Elementes <configuration><configSections> dürfen keine Änderungen vorgenommen werden.

### 4.4.2 <configuration><userSettings>

Die Elemente /Attribute Innerhalb des Elementes <configuration><userSettings>

```
<setting name="BenutzerId" ..>  
    <value>Benutzername</value>
```

Der Wert Benutzername muss durch den Wert ersetzt werden, den sie vom Support als Benutzernamen genannt bekommen haben.

```
<setting name="Passwort" ...>  
    <value>00000</value>
```

Der Wert Passwort muss durch den Wert ersetzt werden, den sie vom Support als Kennwort genannt bekommen haben.

### 4.4.3 <configuration><applicationSettings>

Die Elemente innerhalb des Elementes <configuration><applicationSettings>

```
<setting name="FirmenId" serializeAs="String">  
    <value>00000000</value>
```

Tragen sie hier die vom virtic-Support mitgeteilte eindeutige ID der der Firma mit, für die Daten übertragen werden.

```
<setting name="UseProxy" serializeAs="String">  
    <value>False</value>
```

Hier kann durch den Wert **True** die Verwendung eines Proxy-Servers bestimmt werden, der die http-Anfrage weiterleitet. Ob es in ihrem Umfeld einen Proxy-Server gibt, der für den Zugang zum Internet verwendet werden muss und wie dieser heißt und mit welchen Anmeldedaten man sich mit ihm verbinden muss, wird man beim Netzwerkadministrator erfragen können.

```
<setting name="ProxyUrl" serializeAs="String">  
    <value>http://einProxy:8081</value>
```

Der Name und das Protokoll für die Verbindung mit dem Proxy-Server.  
Dieser Wert ist nur dann relevant, wenn der Wert **UseProxy** gleich **True** ist.

```
<setting name="KWP2VirticEplBst_de_hasenpusch_virtic_virsoapmanagedService"  
    <value>http://www.virtic24.de/wsa/wsa1</value>
```

Der Name des virtic-Webservice-Servers mit der Angabe eines Pfades, wie er bei der Verbindung zum Webservice verwendet werden soll. Hier dürfen nur auf Anweisung des virtic-Supports Änderungen vorgenommen werden.

```
<setting name="Zeichensatz" serializeAs="String">  
    <value>ISO8859-1</value>
```

Der Zeichensatz der Datenbank

```
<setting name="EpTyp" serializeAs="String">  
    <value>0</value>
```

Dieser Wert darf nur auf Anweisung vom Support geändert werden.

```
<setting name="EzBez" serializeAs="String">  
    <value>0</value>
```

Die eindeutige Nummer der Entfernungzone, die der Baustelle des Termins zugewiesen werden soll.  
Der Wertebereich erstreckt sich von **0** bis **5**.

```
<setting name="orten" serializeAs="String">
```

```
<value>False</value>
```

Hier muss immer der Wert **FALSE** angegeben werden

```
<setting name="KstBez" serializeAs="String">  
  <value>0</value>
```

Um die referentielle Integrität der Daten zu erhalten, muss jedem Datensatz ein Fremdschlüssel für den Verweis auf den in virtic existierenden Kostenträger zugewiesen werden. Der Kostenträger mit der Nummer **0** ist immer vorhanden. Soll hier ein anderer Kostenträger verwendet werden, stimmen sie das mit dem virtic-Support ab.

```
<setting name="EvmBez" serializeAs="String">  
  <value>Buchung</value>
```

Wenn für die Baustellen im Rahmen der Verwendung innerhalb einer Buchung weitere Daten abgefragt werden sollen, dann ist hier der Name der Liste, die Informationen über diese zusätzlichen Buchungen erhält, anzugeben. Bitte stimmen das mit dem virtic-Support ab.

```
<setting name="DBConn" serializeAs="String">  
  <value>DSN=kwp</value>
```

Hier muss der „**System DataSource Name**“ der ODBC Datenquelle, die vom Benutzer auf demselben System definiert wurde, angegeben werden.

Die Datenquelle muss eine Windows-Authentifizierung verwenden, damit sie Abfragen korrekt ausgeführt werden können. Die Verwendung einer Benutzer-Datenquelle ist NICHT möglich.

```
<setting name="SQLFile" serializeAs="String">  
  <value>Abfrage.sql</value>
```

Der Name der Datei, welcher die SQL-Abfrage enthält.

```
<setting name="Von2Bis" serializeAs="String">  
  <value>6</value>
```

Anzahl der zusätzlichen Tage vom Zeitpunkt des Programmstarts aus gerechnet, auf die sich der Abfragezeitraum erstrecken soll.

## 4.5 Der Aufbau der SQL-Abfrage

Die SQL-Abfrage muss eine Tabelle mit den folgenden Spalten erzeugen

FeldName	Typ	Beschreibung
ProjNr	String	Schlüsselwert für das Erzeugen/Aktualisieren eines Datensatzes in der Tabelle <code>baustelle</code> der virtic-Datenbank
ProjBezeichnung	String	Der Name der Baustelle in der Virtic-Baustelle
Strasse	String	Straßenname der Baustelle
PLZ	String	PLZ des Ortes, in dem sich die Baustelle befindet
ORT	String	Ort, in dem sich die Baustelle befindet
TerminID	String	Termin ID im Einsatzplan
PersNr	String	Die Personalnummer des Mitarbeiters aus dem KWP System. Der Mitarbeiter muss im virtic-System unter dieser externen Nummer vorhanden sein.
TerminVon	Datetime	Beginn des Termins
TerminBis	Datetime	Ende des Termins

### 4.5.1 Beispiel für die Abfrage der Projekttable

Beachten Sie bitte, dass hier der Projektnummer die Zeichenkette `x#` vorangestellt werden muss. Durch diese Option wird auf der Seite des Servers sichergestellt, dass der Baustelle die Kategorie `x` zugeordnet wird. Dadurch kann während der weiteren Verarbeitung im Rahmen der Berechnung der Zeiten eine gewisse Differenzierung erfolgen. Das Zeichen `#` dient als Trennzeichen zwischen der Werten `kategorie` und `baustellennummer`. Welche Kategorien Sie verwenden können, klären sie bitte mit dem virtic-Support.

Die Abfrage erstreckt sich über zwei Datenbanken (`BNWINS`, `BNWINS2`), da in diesem Fall im KWP-System zwei Mandanten angelegt worden sind. Es werden Termine aus der Projekttable abgefragt.

```
SELECT projNr=('r#' + [BNWINS].[dbo].Projekt.ProjNr),
        projBezeichnung=(ISNULL([BNWINS].[dbo].adrAdressen.Name + ';', '') +
        (ISNULL(substring([BNWINS].[dbo].Projekt.ProjBezeichnung, 1, 30), ''))),
        Strasse = ISNULL([BNWINS].[dbo].adrAdressen.Strasse, ''),
        PLZ = ISNULL([BNWINS].[dbo].adrOrte.PLZ, ''),
        Ort = ISNULL([BNWINS].[dbo].adrOrte.Ort, ''),
        status = [BNWINS].[dbo].MontStrgTermine.Status,
        [BNWINS].[dbo].MontStrgTermine.TerminID,
        [BNWINS].[dbo].MontStrgTermine.TypNr, [BNWINS].[dbo].MontStrgPers.PersNr,
        [BNWINS].[dbo].MontStrgTermine.TerminVon,
[BNWINS].[dbo].MontStrgTermine.TerminBis
FROM      [BNWINS].[dbo].MontStrgTermine
        JOIN  [BNWINS].[dbo].MontStrgPers
            ON [BNWINS].[dbo].MontStrgTermine.TerminID =
[BNWINS].[dbo].MontStrgPers.fkTerminID
        JOIN  [BNWINS].[dbo].Projekt
            ON [BNWINS].[dbo].MontStrgTermine.TypNr = [BNWINS].[dbo].Projekt.ProjNr
        JOIN  [BNWINS].[dbo].adrAdressen
        JOIN  [BNWINS].[dbo].adrOrte
            ON [BNWINS].[dbo].adrAdressen.Ort = [BNWINS].[dbo].adrOrte.OrtID
            ON Projekt.RechAdr = adrAdressen.AdrNrGes
WHERE     ([BNWINS].[dbo].MontStrgTermine.TerminVon >= '#VON# 00:00:00')
        AND ([BNWINS].[dbo].MontStrgTermine.TerminVon <= '#BIS# 23:59:59')
```

Die Zeichenketten #von# und #bis# werden vor der Ausführung der Abfrage in eine Datumszeichenkette des Formats `tt.mm.yyyy` ersetzt. Der #von# Wert entspricht dabei dem Datum zum Zeitpunkt der Ausführung des Programm, der #bis# Wert entspricht dem vorgenannten Datum plus die Anzahl der Tage, die in der Konfigurationsdatei angegeben worden sind.

### 4.5.2 Beispiel für die Abfrage der Regietabelle

Beachten Sie bitte, dass hier der Projektnummer die Zeichenkette `r#` vorangestellt werden muss. Durch diese Option wird auf der Seite des Servers sichergestellt, dass der Baustelle die Kategorie `r` zugeordnet wird. Dadurch kann während der weiteren Verarbeitung im Rahmen der Berechnung der Zeiten eine gewisse Differenzierung erfolgen. Das Zeichen # dient als Trennzeichen zwischen der Werten `kategorie` und `Baustellennummer`. Welche Kategorien Sie verwenden können, klären sie bitte mit dem virtic-Support.



Die Abfrage erstreckt sich über zwei Datenbanken (BNWINS, BNWINS $\mathcal{P}$ ), da in diesem Fall im KWP-System zwei Mandanten angelegt worden sind. Es werden Termine aus der Regietabelle abgefragt.

```
SELECT   ProjNr=('r#' + [BNWINS $\mathcal{P}$ ].[dbo].Regie.ProjNr),
         projBezeichnung=(ISNULL([BNWINS].[dbo].adrAdressen.Name + ';', '') +
         (ISNULL(substring([BNWINS $\mathcal{P}$ ].[dbo].Regie.ProjBezeichnung, 1, 30), ''))),
         Strasse = ISNULL([BNWINS].[dbo].adrAdressen.Strasse, ''),
         PLZ = ISNULL([BNWINS].[dbo].adrOrte.PLZ, ''),
         Ort = ISNULL([BNWINS].[dbo].adrOrte.Ort, ''),
         status = [BNWINS].[dbo].MontStrgTermine.Status,
         [BNWINS $\mathcal{P}$ ].[dbo].Regie.AbtNr, [BNWINS].[dbo].MontStrgTermine.TerminID,
         [BNWINS].[dbo].MontStrgPers.PersNr,
         [BNWINS].[dbo].MontStrgTermine.TerminVon,
         [BNWINS].[dbo].MontStrgTermine.TerminBis

FROM     [BNWINS].[dbo].adrOrte
        RIGHT OUTER JOIN [BNWINS].[dbo].adrAdressen
            ON [BNWINS].[dbo].adrOrte.OrtID = [BNWINS].[dbo].adrAdressen.Ort
        RIGHT OUTER JOIN [BNWINS $\mathcal{P}$ ].[dbo].Regie
        INNER JOIN [BNWINS].[dbo].MontStrgTermine
            ON [BNWINS $\mathcal{P}$ ].[dbo].Regie.ProjNr = [BNWINS].[dbo].MontStrgTermine.TypNr
        LEFT OUTER JOIN [BNWINS].[dbo].MontStrgPers
            ON [BNWINS].[dbo].MontStrgTermine.TerminID =
            [BNWINS].[dbo].MontStrgPers.fkTerminID
            ON [BNWINS].[dbo].adrAdressen.AdrNrGes = [BNWINS $\mathcal{P}$ ].[dbo].Regie.RechAdr

WHERE    ([BNWINS].[dbo].MontStrgTermine.TerminVon >= '#VON# 00:00:00')
        AND ([BNWINS].[dbo].MontStrgTermine.TerminVon <= '#BIS# 23:59:59')
```

### 4.5.3 Der erweiterte Programmaufruf

```
KWP2VirticEplBst.exe fehlerzeilen.dat TEST
```

Der Parameter **TEST** führt zu einem veränderten Verhalten während der Programmausführung. Die Abfrage erfolgt wie gewöhnlich aber die Daten werden nicht an den Server übermittelt, sondern in die Datei `fehlerzeilen.dat` geschrieben. Dies ermöglicht eine Kontrolle der Ergebnismenge.

#### 4.5.4 Sonstiges

Es empfiehlt sich, für mehrere Abfragen jeweils ein gesondertes Unterverzeichnis zu erstellen. Innerhalb der Unterverzeichnisse sollten sich dann die Konfigurationsdateien und die Abfragedateien befinden. Für jedes dieser Verzeichnisse erfolgt dann ein Programmaufruf `KWP2VirticEplBst`.

## 5 Checkliste

1. Programmpaket herunterladen
2. Paket entpacken
3. Ein ODBC-System mit Datenquellendefinition für den Microsoft MSSQL Server erstellen
4. Die Daten für den Zugang beim Support erfragen (**BenutzerID**, **Passwort**, **FirmenID**) und den Namen der System-Datenquelle in beide Konfigurationsdateien eintragen
5. In die Konfigurationsdatei `KWP2VirticEplBst.exe.config` weitere firmenspezifische Änderungen einbringen
6. In die Konfigurationsdatei `KWP2VirticFehl.exe.config` weitere firmenspezifische Änderungen anbringen
7. Die SQL-Abfrage-Dateien erstellen und testen. Anschließend daraus eine Vorlage erstellen, ersetzbare Zeichenketten einfügen(**#von#**, **#bis#**) und in einer Datei speichern.
8. Den Namen der Datei in die betreffende Konfigurationsdatei eintragen.

Die Schritte 7 und 8 wiederholen Sie für weitere Abfragen, wenn also beispielsweise sowohl für die Termine aus der Projektabelle als auch für die Termine aus der Regietabelle Daten abgefragt und übertragen werden sollen, dann empfiehlt es sich, hierfür zwei Unterverzeichnisse anzulegen.

Sollen Daten für mehrere Mandanten übertragen werden, müssen für alle Mandanten Unterverzeichnisse angelegt und die darin enthaltenen Konfigurations- und Abfragedateien (`*.sql`) angepasst werden.

Es empfiehlt sich, den Aufruf zeitgesteuert erfolgen zu lassen.

Erkundigen Sie sich bitte beim Vertrieb über die Bedingungen, um zyklisch Daten an den virtic-Server übertragen zu können.

## 6 Kontakt

virtic GmbH & Co. KG  
Rheinlanddamm 199  
44139 Dortmund

Support:  
Tel: +49 231 24936-24  
eMail: [technik@virtic.com](mailto:technik@virtic.com)