

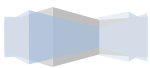
Bundesnetzagentur
Referat IS12 • Entwicklung von IT-Verfahren
Fehrbelliner Platz 3
D-10707 Berlin

Erläuterungen zu den Methoden des XML Webservice Strom 2008

1

Berlin, 01. April 2016

Erläuterungen zum Aufruf der Methoden des XML Webservice 2008 für Versorgungsunterbrechungen Strom



Inhaltsverzeichnis

| | |
|--|----|
| Einleitung und Vorbemerkungen | 5 |
| Zweck des Verfahrens und Erfassungsmöglichkeiten..... | 5 |
| Auswahl und Wechsel des Datenübermittlungsverfahrens | 5 |
| Kontaktaufnahme mit dem XML-Webservice | 10 |
| Nutzung von TLS 1.2 mit .NET 4.5.x | 11 |
| Umfang der nachfolgenden Dokumentation | 12 |
| Regeln für die Nutzung des XML Webservice 2008 | 13 |
| Beispiel: Initialer Aufruf der Proxy-Klasse | 15 |
| Beispiel: Aufruf der DoBeginn()-Methoden | 16 |
| Beispiel: Aufrufreihenfolge „einmalige Netzdaten als Leermeldung“ | 19 |
| Beispiel: Aufrufreihenfolge „einmalige Netzdaten und Versorgungsunterbrechungen“ | 20 |
| Beispiel: Abbruch einer Transaktion (durch den Aufrufer) | 21 |
| Beispiel: Testaufruf (für Software-Entwickler) | 22 |
| Beispiel: Senden der Versorgungsunterbrechungen (fiktive Daten)..... | 23 |
| Beispiel: Rückgabe von Einmaldaten (fiktive Daten)..... | 24 |
| Beispiel: Rückgabe von Unterbrechungen (fiktive Daten) | 25 |



| | |
|--|----|
| Abbildung 1: Anzeige des aktuellen Übermittlungsverfahrens..... | 6 |
| Abbildung 2: Anpassung des Übermittlungsverfahrens..... | 7 |
| Abbildung 3: Bestätigung des Verfahrenswechsels - Sicherheitsabfrage | 8 |
| Abbildung 4: Verfahrenswechsel - Bestätigungsmeldung | 9 |
| Abbildung 5: Aktuelles Übermittlungsverfahren XML-Webservice 2008 | 10 |
| Abbildung 6: Diagramm - Abbruch einer Transaktion | 27 |
| Abbildung 7: Diagramm - Echtdaten ohne Rückgabe der Daten..... | 28 |



Einleitung und Vorbemerkungen

Zweck des Verfahrens und Erfassungsmöglichkeiten

Zweck des Verfahrens ist die Übermittlung von Versorgungsunterbrechungen (Strom). Betreiber von Energieversorgungsnetzen (Elektrizität und Gas) sind gemäß § 52 Energiewirtschaftsgesetz (EnWG) dazu verpflichtet, der Bundesnetzagentur jeweils **bis zum 30. April eines Jahres** einen Bericht über die in ihrem Netz aufgetretenen Versorgungsunterbrechungen vorzulegen.

Zur Übertragung der Versorgungsunterbrechungen der Netzbetreiber und der zugehörigen allgemeinen Netzdaten bietet die Bundesnetzagentur **zwei Datenübermittlungswege** an:

1. **Web-Eingabemasken** (Webformulare) für die manuelle Eingabe der Versorgungsunterbrechungen bzw. für die Angabe der Leermeldung, falls im Datenerhebungszeitraum keine Versorgungsunterbrechungen aufgetreten sind (auch **Leermeldungen** müssen abgegeben werden). Die Nutzung dieser Eingabemasken bietet sich eher für eine kleinere Anzahl an Meldungen an.
2. **XML Webservice** (Version 2008) für eine automatisierte Übertragung der Versorgungsunterbrechungen. Der Webservice ist für die effizientere Übertragung einer größeren Anzahl von Versorgungsunterbrechungen geeignet. Die durch die Bundesnetzagentur definierte XML-Schnittstelle ist über folgenden Link erreichbar:

<https://app.bundesnetzagentur.de/WsVersorgUnterbrStrom/WsVersUnterbrechungenStrom.aspx>

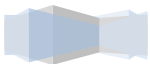
Auswahl und Wechsel des Datenübermittlungsverfahrens

Für Netzbetreiber, die sich **neu bei der Bundesnetzagentur registriert** haben (sog. Stammdaten-Erfassung), ist kein **Datenübermittlungsweg** ausgewählt. Die Registrierung läuft (Stand: März 2016) in drei Phasen ab, wobei die 3. Phase die Erstanmeldung im Energiedaten-Portal der Bundesnetzagentur mit den postalisch zugesandten Zugangsdaten darstellt. Vor einem Wechsel des **Datenübermittlungswegs** muss im Energiedatenportal der Datenübermittlungsweg **ausgewählt werden**. Das Energiedaten-Portal ist über folgenden Link erreichbar:

<https://app.bundesnetzagentur.de/Energie/Login.aspx>

5

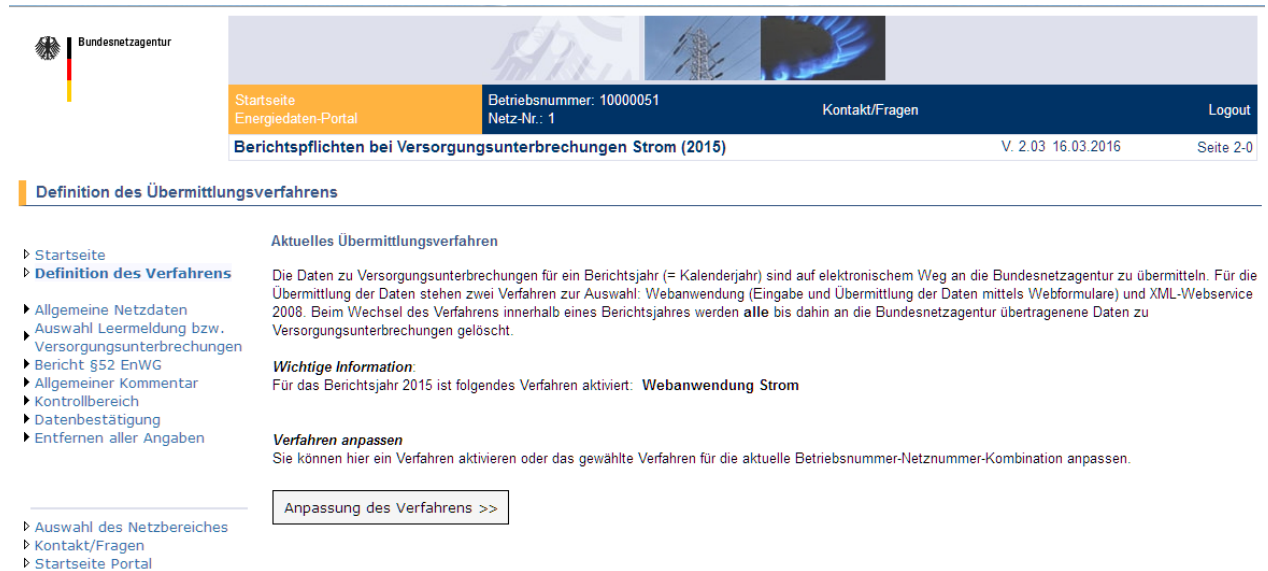
Mit der erstmaligen Anmeldung im Energiedaten-Portal durch den Kommunikationsbevollmächtigten (KBV) ist die 3. Phase der Registrierung abgeschlossen und das Portal somit nutzbar.



Im Energiedaten-Portal stehen verschiedene Funktionen zur Verfügung; der erste Unterpunkt lautet „Versorgungsunterbrechungen Strom“. Nach der Anmeldung und der Auswahl der Funktion steht der Menüpunkt „**Definition des Übermittlungsverfahrens**“ zur Verfügung, worüber das Datenübermittlungsverfahren von Webformularen hin zum Webservice bzw. vom Webservice hin zur Webanwendung gewechselt werden kann.

Die folgenden Abbildungen stellen einen Wechsel unter Verwendung einer Test-Betriebsnummer exemplarisch dar.

Abbildung 1: Anzeige des aktuellen Übermittlungsverfahrens



The screenshot displays the 'Definition des Übermittlungsverfahrens' page. The header includes the Bundesnetzagentur logo and navigation links: 'Startseite Energiedaten-Portal', 'Betriebsnummer: 10000051 Netz-Nr.: 1', 'Kontakt/Fragen', and 'Logout'. The main title is 'Berichtspflichten bei Versorgungsunterbrechungen Strom (2015)'. The left sidebar lists navigation options: 'Startseite', 'Definition des Verfahrens', 'Allgemeine Netzdaten', 'Auswahl Leermeldung bzw. Versorgungsunterbrechungen', 'Bericht §52 EnWG', 'Allgemeiner Kommentar', 'Kontrollbereich', 'Datenbestätigung', and 'Entfernen aller Angaben'. The main content area is titled 'Definition des Übermittlungsverfahrens' and contains the following text:

Aktuelles Übermittlungsverfahren

Die Daten zu Versorgungsunterbrechungen für ein Berichtsjahr (= Kalenderjahr) sind auf elektronischem Weg an die Bundesnetzagentur zu übermitteln. Für die Übermittlung der Daten stehen zwei Verfahren zur Auswahl: Webanwendung (Eingabe und Übermittlung der Daten mittels Webformulare) und XML-Webservice 2008. Beim Wechsel des Verfahrens innerhalb eines Berichtsjahres werden **alle** bis dahin an die Bundesnetzagentur übertragenen Daten zu Versorgungsunterbrechungen gelöscht.

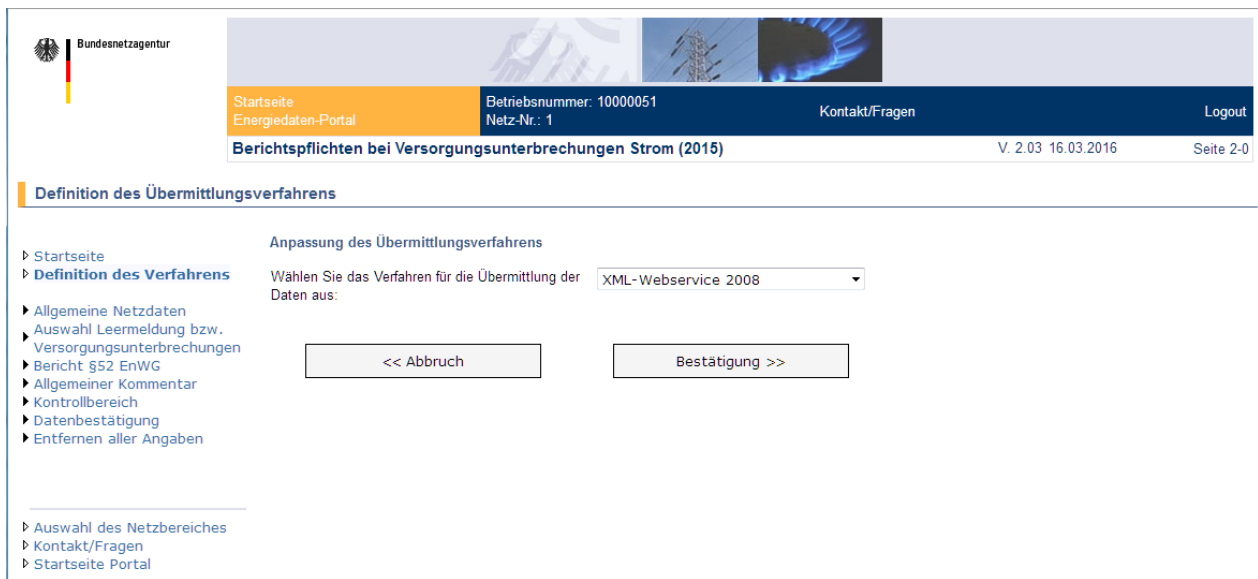
Wichtige Information:
Für das Berichtsjahr 2015 ist folgendes Verfahren aktiviert: **Webanwendung Strom**

Verfahren anpassen
Sie können hier ein Verfahren aktivieren oder das gewählte Verfahren für die aktuelle Betriebsnummer-Netznummer-Kombination anpassen.

At the bottom, there is a button labeled 'Anpassung des Verfahrens >>'.

Abbildung 2 zeigt die beiden hinterlegten Verfahren und wie sie angepasst werden können.

Abbildung 2: Anpassung des Übermittlungsverfahrens



Bundesnetzagentur

Startseite
Energiedaten-Portal

Betriebsnummer: 10000051
Netz-Nr.: 1

Kontakt/Fragen

Logout

Berichtspflichten bei Versorgungsunterbrechungen Strom (2015)

V. 2.03 16.03.2016

Seite 2-0

Definition des Übermittlungsverfahrens

► Startseite

► **Definition des Verfahrens**

► Allgemeine Netzdaten

► Auswahl Leermeldung bzw. Versorgungsunterbrechungen

► Bericht §52 EnWG

► Allgemeiner Kommentar

► Kontrollbereich

► Datenbestätigung

► Entfernen aller Angaben

Anpassung des Übermittlungsverfahrens

Wählen Sie das Verfahren für die Übermittlung der Daten aus:

XML-Webservice 2008

<< Abbruch

Bestätigung >>

► Auswahl des Netzbereiches

► Kontakt/Fragen

► Startseite Portal

Abbildung 3 zeigt die Sicherheitsabfrage, die im Rahmen des Verfahrenswechsels gestellt wird. Beachten Sie, dass durch den Verfahrenswechsel ggf. bereits zum Datenerhebungsjahr eingegebene Daten unwiderbringlich **gelöscht** werden.

Abbildung 3: Bestätigung des Verfahrenswechsels - Sicherheitsabfrage

Definition des Übermittlungsverfahrens

▶ Startseite
 ▶ **Definition des Verfahrens**
 ▶ Allgemeine Netzdaten
 ▶ Auswahl Leermeldung bzw. Versorgungsunterbrechungen
 ▶ Bericht §52 EnWG
 ▶ Allgemeiner Kommentar
 ▶ Kontrollbereich
 ▶ Datenbestätigung
 ▶ Entfernen aller Angaben

▶ Auswahl des Netzbereiches
 ▶ Kontakt/Fragen
 ▶ Startseite Portal

Anpassung des Übermittlungsverfahrens

Wählen Sie das Verfahren für die Übermittlung der Daten aus: XML-Webservice 2008

Wichtiger Hinweis

Durch den Verfahrenswechsel werden alle bisher von Ihnen eingegebenen Daten für das laufende Berichtsjahr gelöscht! Hatten Sie bereits ein Verfahren gewählt und wechseln nun dieses (z.B. von 'Webanwendung' zu 'XML-Webservice 2008'), so wird zudem eine neue Erhebung angelegt. Im Anschluss an diesen Vorgang können Sie über die [Startseite des Portals](#) diesen Bereich erneut aktivieren.

Bitte bestätigen Sie nochmals den Verfahrenswechsel!

<< Abbruch
 Bestätigung und Weiter >>

Abbildung 4 zeigt den Bestätigungsdialog nach dem Verfahrenswechsel.

Abbildung 4: Verfahrenswechsel - Bestätigungsmeldung



The screenshot displays the web interface of the Bundesnetzagentur. At the top, there is a header bar with the agency's logo and name on the left, and navigation links for 'Startseite', 'Energiedaten-Portal', 'Betriebsnummer: 10000051', 'Netz-Nr.: 1', 'Kontakt/Fragen', and 'Logout' on the right. Below the header, a blue banner contains the text 'Berichtspflichten bei Versorgungsunterbrechungen Strom (2015)', the version 'V. 2.03 16.03.2016', and 'Seite 2-0'. The main content area is titled 'Verfahrenswechsel - Bestätigung'. It contains a message addressed to the network operator, stating that the procedure change has been successfully completed and that data can now be transferred using the new procedure. The message is signed 'Mit freundlichen Grüßen' and 'Ihre Bundesnetzagentur'. At the bottom of the message area, there are two buttons: '<< Portal' and 'Auswahl des Netzbereiches >>'.

Bundesnetzagentur

Startseite
Energiedaten-Portal

Betriebsnummer: 10000051
Netz-Nr.: 1

Kontakt/Fragen

Logout

Berichtspflichten bei Versorgungsunterbrechungen Strom (2015)

V. 2.03 16.03.2016

Seite 2-0

Verfahrenswechsel - Bestätigung

Verfahrenswechsel - Bestätigung

Sehr geehrter Netzbetreiber,

der von Ihnen in Auftrag gegebene Verfahrenswechsel ist erfolgreich durchgeführt worden. Sie können nun Ihre Daten über das neu gewählte Verfahren übertragen.

Mit freundlichen Grüßen

Ihre Bundesnetzagentur

<< Portal

Auswahl des Netzbereiches >>

Hinweis: Für den Verfahrenswechsel wird eine iTAN aus der iTAN-Liste benötigt und entwertet, die dem KBV vorliegt.

Abbildung 5 zeigt die Definition des Verfahrens (zur Kontrolle).

Abbildung 5: Aktuelles Übermittlungsverfahren XML-Webservice 2008

 Bundesnetzagentur

Startseite
Energiedaten-Portal

Betriebsnummer: 10000051
Netz-Nr.: 1

Kontakt/Fragen

Logout

Berichtspflichten bei Versorgungsunterbrechungen Strom (2015)

V. 2.03 16.03.2016

Seite 2-0

Definition des Übermittlungsverfahrens

Startseite

Definition des Verfahrens

Auswahl des Netzbereiches

Kontakt/Fragen

Startseite Portal

Aktuelles Übermittlungsverfahren

Die Daten zu Versorgungsunterbrechungen für ein Berichtsjahr (= Kalenderjahr) sind auf elektronischem Weg an die Bundesnetzagentur zu übermitteln. Für die Übermittlung der Daten stehen zwei Verfahren zur Auswahl: Webanwendung (Eingabe und Übermittlung der Daten mittels Webformulare) und XML-Webservice 2008. Beim Wechsel des Verfahrens innerhalb eines Berichtsjahres werden **alle** bis dahin an die Bundesnetzagentur übertragenene Daten zu Versorgungsunterbrechungen gelöscht.

Wichtige Information:
Für das Berichtsjahr 2015 ist folgendes Verfahren aktiviert: **XML-Webservice 2008**

Verfahren anpassen
Sie können hier ein Verfahren aktivieren oder das gewählte Verfahren für die aktuelle Betriebsnummer-Netznummer-Kombination anpassen.

Anpassung des Verfahrens >>

Kontaktaufnahme mit dem XML-Webservice

Ein Webservice (auch Webdienst genannt) stellt eine Maschine-zu-Maschine-Schnittstelle dar, mit deren Hilfe Daten auf der Basis von **XML (eXtensible Markup Language)** zwischen heterogenen Systemen (unterschiedliche Hersteller, unterschiedliche Programmierumgebungen) serviceorientiert ausgetauscht werden können. Webservices liefern eine (maschinengenerierte) maschinenlesbare Schnittstellen-Beschreibung als XML-Artefakt (**WSDL, Webservice Description Language**), mit deren Hilfe sich sog. **Proxy-Klassen** –ebenfalls maschinell- erstellen lassen. Diese Proxy-Klassen werden für die Einbindung in die vom Netzbetreiber oder dem externen Dienstleister favorisierte Programmiersprache benötigt. Bei technischen Änderungen am Webservice, die direkt die WSDL betreffen, müssen die Proxy-Klassen neu generiert werden. Dazu bieten sich, je nach verwendeter Programmierumgebung, üblicherweise Tools an, die diese Aufgabe übernehmen.

Auf Grund von allgemeinen Sicherheitsbestimmungen des Bundesamtes für Sicherheit in der Informationstechnologie (BSI) ist die Bundesnetzagentur gehalten, für die Sicherung des Übertragungsweges (hier: Internet, Wirtschaft-zu-Behörden-Kommunikation) die aktuellen

Sicherheitsprotokolle zu nutzen. Als Standard ist nach §8 Abs.1 Satz 1 BSIG das **Protokoll TLS 1.2 (Transport Layer Security, Version 1.2)** definiert (Stand: März 2016).

Das Dokument ist hier als PDF zu finden:

https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Mindeststandards/Mindeststandard_BSI_TLS_1_2_Version_1_0.pdf?__blob=publicationFile&v=2

Auszug aus dem Dokument, S. 8:

„Das SSL-Protokoll existiert in den Versionen 1.0, 2.0 und 3.0, wobei die Version 1.0 nicht veröffentlicht wurde. SSL in der Version 2.0 wurde schon im Jahre 1996 wegen zahlreicher funktionaler Schwachstellen von der IETF abgekündigt und die Nutzung des von Netscape entwickelten SSL der Version 3.0 empfohlen. TLS 1.0 ist eine direkte Weiterentwicklung von SSL 3.0. Dessen Spezifikation wurde Anfang 1999 als [RFC2246] veröffentlicht. Des Weiteren gibt es für das TLS-Protokoll Sicherheitsanpassungen in den Versionen 1.1 und 1.2, welche 2006 als [RFC4346] bzw. 2008 als [RFC5246] spezifiziert wurden. **Seit 2011 sind mehrere Angriffe gegen SSL/TLS bekannt geworden.** Die entsprechenden Schwachstellen können durch Nutzung entsprechender Cipher-Suites gemäß [TR-02102-2] in **TLS 1.2** behoben werden.“

Nutzung von TLS 1.2 mit .NET 4.5.x

Für die Programmierung des Webservice-**Clients** stehen viele Technologien zur Auswahl. Die folgende technische Dokumentation am Beispiel von Microsoft .NET 4.5.2 bzw. 4.6.1 stellt keine Technologie-Empfehlung dar, sondern ist nur zur Demonstration der technischen Umsetzbarkeit gedacht.

Microsoft bietet das kostenlose .NET Framework an; zum jetzigen Stand (März 2016) ist die Version 4.6.1 aktuell. Die jeweils aktuelle Framework-Version kann beim Hersteller eingesehen werden:

<https://msdn.microsoft.com/de-de/library/bb822049%28v=vs.110%29.aspx>

Oracle bietet ebenfalls ein Framework an; und zwar das weit verbreitete Java Development Kit (JDK).

Vgl. <http://www.oracle.com/technetwork/java/javase/overview/index.html>

(Auf die Java-Technologie wird im Folgenden nicht weiter eingegangen).

Die vom .NET Framework **unterstützen Protokoll-Versionen** sind bei Microsoft hinterlegt:

<https://msdn.microsoft.com/de-de/library/system.security.authentication.sslprotocols%28v=vs.110%29.aspx>

Bitte beachten Sie, dass ältere .NET - Versionen das neuere TLS 1.2-Protokoll nicht unterstützen!



Bezüglich der Protokoll-Unterstützung anderer Programmiersprachen / -umgebungen wie Delphi, PHP, SAP,... usw. wenden Sie sich bitte an den Hersteller der jeweiligen Programmiersprache bzw. – umgebung.

Um die Default-Einstellung auf die Verwendung des Protokolls TLS 1.2 zu ändern, *kann* ein Registrierungsschlüssel implementiert werden. **Bitte beachten Sie, dass durch die Verwendung eines Registrierungsschlüssels, der sich de facto auf alle .NET-Programme auswirkt, Probleme mit anderen Programmen entstehen können („Seiteneffekte“).** Es empfiehlt sich die Nutzung eines separaten Entwickler-Computers sowie der ausgiebige Test vor dem Rollout der Registrierungseinstellung in die Fläche. Die Nutzung geschieht auf eigene Gefahr!

Im Bereich

COMPUTER\HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\.NETFramework\v4.0.30319
einen Wert vom Typ REG_DWORD hinzufügen:

SchUseStrongCrypto 0x000000001

Bitte beachten Sie weiterhin, dass bei der Verwendung von **32-Bit-Framework-Versionen** und **64-Bit-Framework-Versionen** zu jeder Version ein Schlüssel hinzugefügt werden sollte.

Danach den Computer ggf. neu booten.

Umfang der nachfolgenden Dokumentation

Die nachfolgende Dokumentation konzentriert sich auf die wesentlichen, zur Kommunikation mit dem Webservice benötigten Komponenten und ist insofern unvollständig. Dokumentiert sind die wesentlichen Methoden, Eigenschaften, Objekte und Enumerationen. **Beachten Sie auch die Liste der MeldungsCodes, die Beschreibung der Datenfelder sowie die HTML-Hilfedatei.**

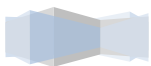


Regeln für die Nutzung des XML Webservice 2008

1. Die Anzahl der Versorgungsunterbrechungen ist zwar theoretisch unbegrenzt; jedoch ist aus Sicherheitsgründen ein Timeout auf der Webserver-Farm eingestellt.
2. Die Versorgungsunterbrechungen sind in fortlaufende, nummerierte Pakete aufzuteilen.
3. Auch wenn es sich um nur eine Meldung handelt, muss ein Paket generiert werden (ggf. Web-Formulare nutzen?).
4. Innerhalb eines Paketes dürfen **maximal 2000** Versorgungsunterbrechungen (VU's) gesendet werden.
5. Die Angaben zu den Paketen werden zu Beginn der Transaktion gemacht.
6. Bei einer Leermeldung müssen die Angaben zu Versorgungsunterbrechungen auf 0 gesetzt werden.
7. Sofern der Netzbetreiber mehrere (physische oder logische) Netzte betreibt, können für jedes betroffene Netzgebiet Transaktionen durchgeführt werden.
8. Die Pakete mit Versorgungsunterbrechungen sind zu nummerieren und müssen in der richtigen Reihenfolge gesendet werden (Paket 1, Paket 2, Paket 3, ..., Paket n).
9. Das letzte Paket kann den Rest der Versorgungsunterbrechungen enthalten (d.h., die nicht durch 2.000 teilbare Restmenge).

Beispiel: Es liegen 7.583 VU's vor => 3 Pakete zu je 2.000 VU's; 4. Paket mit 1.583 VU's.

10. Die Versorgungsunterbrechungen (innerhalb des Paketes bzw. der Pakete) müssen fortlaufend sein.
11. Es kann maximal drei Details zu einer Versorgungsunterbrechung geben. Dies ist abhängig von der getroffenen Auswahl bei VSU_1_14 bis VSU_1_17.
12. Die Aufrufreihenfolge ist unbedingt einzuhalten.
13. Nur Transaktionen mit Abschluss werden von der Bundesnetzagentur weiter verarbeitet.



14. Das **Ergebnis der Vorprüfung** kann auf Grund von internen Betriebsprozessen **erst nach 24 Stunden** abgerufen werden. Hierzu ist die Transaktionsnummer aufzubewahren. Dies bezieht sich auch auf Rückfragen an die Fachabteilung der Bundesnetzagentur. Nur mit Hilfe der **Transaktionsnummer** kann -bei der Vielzahl der Netzbetreiber- die entsprechende Datenübermittlung schnell und sicher identifiziert werden.

Bitte teilen Sie bei Anfragen an die Bundesnetzagentur ebenfalls die achtstellige **Betriebsnummer** sowie die **Netznummer** mit. **Bitte senden Sie auf keinen Fall vertrauliche Informationen wie Kontrollnummer, Schlüssel oder iTAN mit!** Sollte die Bundesnetzagentur Kenntnis davon erlangen, dass sensitive Daten kompromittiert wurden, wird aus Sicherheitsgründen der **Zugang unverzüglich gesperrt.**

15. Enthält die Vorprüfliste in der Vorprüfantwort keine Einträge, so wurden die Daten von der Bundesnetzagentur angenommen.
16. Die Rückmeldung des Dateneingangs (Datenquittung) wird zusätzlich als verschlüsselte xlsx.dat-Datei im Excel 2007-Format im **Download-Bereich** des Energiedaten-Portals für den manuellen Download zur Verfügung gestellt (vgl. Rückgabe-Methoden). Die Datenquittung ist üblicherweise 48 Stunden nach Eingang der korrekten Datenübermittlung verfügbar, je nach Auslastung der Systeme aber auch erst später.

Hinweise zu den Code-Beispielen:

- In den folgenden Beispielen wird auf die Methode **DoLvItem** verwiesen. Diese dient der Befüllung / Neuzeichnung eines ListView-Controls (GUI) und ist hier nicht weiter dargestellt.
- Mit dem Begriff „**Proxy**“ kann hier der **Proxy-Server** als Zugangssystem für den Internet-Zugriff gemeint sein, oder auch die maschinell generierte **Proxy-Klasse**, die sich aus der **WSDL** ergibt.
- Einige Variablen sind an anderen Stellen (z.B. global) definiert.
- Es wird ein transparenter System-Proxyserver, wie z.B. im Edge-Browser / Internetoptionen hinterlegt, genutzt, über den auf das Internet zugegriffen wird.
- Die benötigten Webservice-Proxy-Klassen wurden zuvor bereits aus der WSDL generiert.
- Die Log-Klasse ist eine generische statische Klasse (Trace-Ausgabe in lokale Textdatei) und hier nicht kolportiert.



Beispiel: Initialer Aufruf der Proxy-Klasse

```
(  
#region Webservice initialisieren  
  
    /// <summary>  
    /// Instanz des Webservice anlegen und den Proxy zuweisen.  
    /// </summary>  
    /// <returns>Instanz der Webservice-Proxyklasse</returns>  
    WsVersUnterbrechungenStrom GetWS()  
    {  
        Debug.Assert(!String.IsNullOrEmpty(_sWSUri), "Webservice-  
URI ist null!");  
  
        toolStripLblStatus.Text = String.Format("Verbinde mit Webservice: {0}", _sWSUri);  
        Application.DoEvents();  
  
        Log.Append(0, toolStripLblStatus.Text);  
  
        DateTime dStart = DateTime.Now;  
  
        WsVersUnterbrechungenStrom oWS = new WsVersUnterbrechungenStrom();  
        oWS.Url = _sWSUri;  
        oWS.Proxy = myProxy;  
  
        DateTime dEnde = DateTime.Now;  
  
        DoLvItem(WsVersUnterbrechungenStrom.TransStatusEnum.Akzeptiert, dStart, dEnde, "Konstrukt  
or",  
        String.Format("Verbinde mit Webservice: {0}", toolStripCboBoxURL.Text), "", null);  
  
#pragma warning disable 0618  
        toolStripLblStatus.Text = String.Format("Verbunden mit Webservice: {0} [Proxy: {1}:{2}]",  
  
#pragma warning restore 0618  
  
        Application.DoEvents();  
  
        Log.Append(0, toolStripLblStatus.Text);  
  
        return oWS;  
    }  
)
```



```
}
#endregion
```

Beispiel: Aufruf der DoBeginn()-Methoden

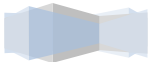
(Hinweis: Drei überladene Methoden; Ziel ist der Start einer Transaktion.)

#region DoBeginn-Methoden

```
    /// <summary>
    /// Start einer Transaktion. Ermittelt eine Transaktionsnummer und speichert diese als globale Variable.
    /// </summary>
    void DoBeginn()
    {
        DoBeginn(_strTestBetrieb.sTestKNR, true, _strTestBetrieb.iTestNNR, 0, 0, 0, iBerichtsJahr, TransBeginnArtEnum.EchtDaten);
    }

    /// <summary>
    /// Start einer Transaktion. Ermittelt eine Transaktionsnummer und speichert diese als globale Variable.
    /// </summary>
    /// <param name="sKontrNr"></param>
    /// <param name="iNNR"></param>
    void DoBeginn(String sKontrNr, int iNNR)
    {
        DoBeginn(sKontrNr, false, iNNR, 0, 0, 0, iBerichtsJahr, TransBeginnArtEnum.EchtDaten);
    }

    /// <summary>
    /// Start einer Transaktion. Ermittelt eine Transaktionsnummer und speichert diese als globale Variable.
    /// </summary>
    /// <param name="sKontrNr"></param>
    /// <param name="bLeer"></param>
    /// <param name="iNNR"></param>
    /// <param name="iPaketeGesamt"></param>
    /// <param name="iAnzahlUntGesamt"></param>
    /// <param name="iAnzahlUntInPaket"></param>
```




```

        /// <param name="iBJ"></param>
        /// <param name="oEnArt"></param>

        /// <returns>true, wenn eine Transaktionsnummer zurück kam (Status Akzeptiert), sonst false</returns>

        private bool DoBeginn(String sKontrNr, bool bLeer, int iNNr, int iPaketeGesamt, int iAnzahlUntGesamt, int iAnzahlUntInPaket, int iBJ,
            WSVersUnterbrechungenStrom.TransBeginnArtEnum oEnArt)
        {
            // Webservice instanziiieren
            WSVersUnterbrechungenStrom oWS = GetWS();

            DateTime dStart = DateTime.Now;

            // Transaktion beginnen

            Log.Append(0, String.Format("DoBeginn {0} {1} {2} {3} {4} {5} {6} {7} {8}",
                _strTestBetrieb.iTestBNR, sKontrNr, _strTestBetrieb.iTestNNR, iBJ, bLeer,
                iPaketeGesamt, iAnzahlUntGesamt, iAnzahlUntInPaket, oEnArt));

            WSVersUnterbrechungenStrom.BeginnTranAntwort oAnt =
                oWS.BeginnTransaktion(_strTestBetrieb.iTestBNR, sKontrNr, _strTestBetrieb.iTestNNR, iBJ, bLeer,
                    iPaketeGesamt, iAnzahlUntGesamt, iAnzahlUntInPaket, oEnArt);

            // Transaktionsnummer global speichern
            _sTranNr = oAnt.Transaktionsnummer;
            Log.Append(2, String.Format("Transaktionsnummer: {0}", _sTranNr));

            DateTime dEnde = DateTime.Now;

            TranAntwort oAntTest = new TranAntwort();

            oAntTest.Meldung = String.Format("NetzNr: {0}, BJahr: {1}, Leerm.:{2}, Pakete ges.: {3},
                VU ges.: {4}, VU/Paket: {5}",
                iNNr, iBJ, bLeer, iPaketeGesamt, iAnzahlUntGesamt, iAnzahlUntInPaket);

            DoLvItem(oAnt.StatusWs, dStart, dEnde, "BeginnTransaktion",
                "Info zum Aufruf",
                String.Format("Art: {6}, NetzNr: {0}, BJahr: {1}, Leerm.:{2}, Pakete ges.: {3}, VU ges.: {4}, VU/Paket: {5}",

```



```

        iNNr, iBJ, bLeer, iPaketeGesamt, iAnzahlUntGesamt, iAnzahlUntInPaket, oEnAnt), oAntTest
    );

    if (oAnt.StatusWs != WSVersUnterbrechungenStrom.TransStatusEnum.Akzeptiert)
    {
        toolStripLblStatus.Text = String.Format("Nicht mit Webservice verbunden, Status: {0}", oAnt.StatusWs);
        Log.Append(4, toolStripLblStatus.Text);

        DoLvItem(oAnt.StatusWs, dStart, dEnde, "BeginnTransaktion",
            "Transaktion beginnen", String.Format("{0}, {1}", oAnt.StatusWs, oAnt.Meldung), oAnt);

        return false;
    }

    toolStripLblStatus.Text = String.Format("Verbunden mit Webservice, BeginnTransaktion Rückgabe: {0}", _sTranNr);
    Log.Append(2, toolStripLblStatus.Text);

    DoLvItem(oAnt.StatusWs, dStart, dEnde, "BeginnTransaktion", "Transaktion beginnen", _sTranNr, oAnt);

    // Transaktions-ID im ToolStrip anzeigen
    toolStripTxtBoxTransID.Text = _sTranNr;

    Application.DoEvents();

    return true;
}

#endregion

```



Beispiel: Aufrufreihenfolge „einmalige Netzdaten als Leermeldung“ (fiktive Betriebsnummer)

```
/// <summary>
    /// Leermeldung mit Einmal-Objekt senden.<br/>
    /// <list type="bullet">
    /// <item>Instanz des Webservice erzeugen</item>
    /// <item>Transaktion beginnen</item>
    /// <item>Einmaldaten senden</item>
    /// <item>Abschluss der Transaktion</item>
    /// </list>
    /// </summary>
    /// <param name="oEinmalig">einmalige Netzdaten</param>
    /// <returns>true bei Erfolg (Status Akzeptiert)</returns>
    private bool DoEinmaligLeer(Einmalig oEinmalig)
    {
        Log.Append(0, "DoEinmaligLeer(Einmalig)");

        // 1) Instanz des Webservice erzeugen
        WsVersUnterbrechungenStrom oWS = new WsVersUnterbrechungenStrom();

        // 2) Transaktion beginnen
        BeginnTranAntwort oAnt =

oWS.BeginnTransaktion(999999999, "1234567A", 1, iBerichtsJahr, true,

0, 0, 0, WsVersUnterbrechungenStrom.TransBeginnArtEnum.EchtDaten);

        if (oAnt.StatusWs != TransStatusEnum.Akzeptiert)
        {
            Log.Append(4, String.Format("BeginnTranAntwort.StatusWs [Beginn]: {0}", oAnt.StatusWs.ToString()));
            return false;
        }

        // 3) Einmaldaten senden

        TranAntwort oAntEinmalig = oWS.TransEinmalig(oAnt.Transaktionsnummer, oEinmalig);
        if (oAntEinmalig.StatusWs != TransStatusEnum.Akzeptiert)
        {
            Log.Append(4, String.Format("TranAntwort.StatusWs [Einmalig]: {0}", oAntEinmalig.StatusWs.ToString()));
            return false;
        }

        // 4) Abschluss der Transaktion
        TranAntwort oAntAbschluss = oWS.AbschlussTransaktion(_sTranNr);
        if (oAntAbschluss.StatusWs != TransStatusEnum.Akzeptiert)
        {
            Log.Append(4, String.Format("TranAntwort.StatusWs [Abschluss]: {0}", oAntAbschluss.StatusWs.ToString()));
            return false;
        }
    }
};
```



```

    }
    return true;
}

```

Beispiel: Aufrufreihenfolge „einmalige Netzdaten und Versorgungsunterbrechungen“

```

/// <summary>
/// Sendet Einmaldaten.
/// </summary>
/// <param name="oEinmalig">Instanz der Klasse Einmalig</param>
/// <returns>true, wenn der Status Akzeptiert zurück gegeben wurde</returns>
private bool DoEinmalig(Einmalig oEinmalig)
{
    Log.Append(0, "--- Start DoEinmalig ---");

    WSVersUnterbrechungenStrom.WsVersUnterbrechungenStrom oWS = GetWS();
    DateTime dStart = DateTime.Now;

    WSVersUnterbrechungenStrom.TranAntwort oAntEinmalig = oWS.TransEinmalig(_sTranNr, oEinmalig);

    toolStripLblStatus.Text = String.Format("Einmaldaten zur Transaktionsnummer: {0}", _sTranNr);
    Log.Append(0, toolStripLblStatus.Text);

    DateTime dEnde = DateTime.Now;

    DoLvItem(oAntEinmalig.StatusWs, dStart, dEnde, "TransEinmalig", "Einmaldaten senden", oAntEinmalig.Mel
dung, oAntEinmalig);

    if (oAntEinmalig.StatusWs == TransStatusEnum.Akzeptiert)
    {
        Log.Append(0, "--- DoEinmalig OK ---");
        return true;
    }

    Log.Append(0, "--- DoEinmalig Fehler ---");

    return false;
}

```



Beispiel: Abbruch einer Transaktion (durch den Aufrufer)

```
/// <summary>
/// Transaktion abbrechen.
/// </summary>
private void DoTransAbbruch()
{
    WSVersUnterbrechungenStrom.WsVersUnterbrechungenStrom oWS = GetWS();
    DateTime dStart = DateTime.Now;

    WSVersUnterbrechungenStrom.TranAntwort oAnt = oWS.AbbruchTransaktion(_sTranNr);

    DateTime dEnde = DateTime.Now;

    toolStripLbl1Status.Text = String.Format("Abbruch Tran", oAnt.StatusWs);
    Log.Append(2, toolStripLbl1Status.Text);
    DoLvItem(oAnt.StatusWs, dStart, dEnde, "AbbruchTransaktion", "Abbruch Tran",
        String.Format("{0}, {1}", oAnt.StatusWs, oAnt.Meldung), oAnt);
}
```

Beispiel: Abschluss einer Transaktion (durch den Aufrufer)

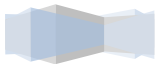
```
/// <summary>
/// Transaktionsabschluss
/// </summary>
private void DoTransAbschluss()
{
    WSVersUnterbrechungenStrom.WsVersUnterbrechungenStrom oWS = GetWS();
    DateTime dStart = DateTime.Now;

    WSVersUnterbrechungenStrom.TranAntwort oAnt = oWS.AbschlussTransaktion(_sTranNr);

    DateTime dEnde = DateTime.Now;

    toolStripLbl1Status.Text = String.Format("Abschluss Transaktion", oAnt.StatusWs);
    Log.Append(2, toolStripLbl1Status.Text);

    DoLvItem(oAnt.StatusWs, dStart, dEnde, "AbschlussTransaktion",
        "Abschluss Tran",
        String.Format("{0}, {1}", oAnt.StatusWs, oAnt.Meldung), oAnt);
}
```



Beispiel: Testaufruf (für Software-Entwickler)

(Hinweise: Fiktive Betriebsnummer. Als Berichtsjahr ist das abgelaufene Kalenderjahr zu verwenden. Das Verfahren muss korrekt eingestellt sein, und der Zugang durch die Fachabteilung der Bundesnetzagentur genehmigt sein. Es können entweder fiktive Nummern verwendet werden, oder – besser – die von der Fachabteilung für diese Zwecke zugeteilten Test-Betriebsnummern-/Netznummern-Kombination (Zuteilung auf Anfrage). **Beachten Sie die Hinweise zum Datenschutz in Bezug auf die Verwendung von Test-Betriebsnummern!**)

```
/// <summary>
    /// Testaufruf.<br/>
    /// Betriebsnummer und Kontrollnummer können fiktiv sein.<br/>
    /// <list type="bullet">
    /// <item>Instanz des Webservice erzeugen</item>
    /// <item>Transaktion mit TestVerweigert beginnen</item>
    /// <item>Transaktion mit TestNichtAkzeptiert beginnen</item>
    /// </list>
    /// </summary>
    private void DoTestAufruf()
    {
        Log.Append(0, "DoTestAufruf()");

        // 1) Instanz des Webservice erzeugen
        WsVersUnterbrechungenStrom oWS = new WsVersUnterbrechungenStrom();

        // 2) Transaktion mit TestVerweigert beginnen

        BeginnTranAntwort oAnt = oWS.BeginnTransaktion(99999999, "1234567A", 1, iBerichtsJahr, true, 0, 0, 0,
            WsVersUnterbrechungenStrom.TransBeginnArtEnum.TestVerweigert);

        if (oAnt.StatusWs != TransStatusEnum.Verweigert)
        {
            // Test OK
            Log.Append(0, "Test OK");
        }

        oAnt = oWS.BeginnTransaktion(99999999, "1234567A", 1, iBerichtsJahr, true, 0, 0, 0,
            WsVersUnterbrechungenStrom.TransBeginnArtEnum.TestNichtAkzeptiert);

        if (oAnt.StatusWs != TransStatusEnum.TestTrans)
        {
            // Testmodus

            TranAntwort oAntEinmalig = oWS.TransEinmalig(oAnt.Transaktionsnummer, new Einmalig());
            if (oAntEinmalig.StatusWs != TransStatusEnum.NichtAkzeptiert)
            {
                // Test OK
                Log.Append(0, "Test OK");
            }
        }
    }
}
```



```

    }
}
}

```

Beispiel: Senden der Versorgungsunterbrechungen (fiktive Daten)

```

/// <summary>
    /// Senden der Versorgungsunterbrechungen.<br/>
    /// <list type="bullet">
    /// <item>Instanz des Webservice erzeugen</item>
    /// <item>Transaktion beginnen</item>
    /// <item>Einmaldaten senden</item>
    /// <item>VU's senden</item>
    /// <item>Abschluss der Transaktion</item>
    /// </list>
    /// </summary>
    /// <param name="oEinmalig">einmalige Netzdaten</param>
    /// <param name="oVUs">Versorgungsunterbrechungen als Array</param>
    /// <returns>true bei Erfolg (Status Akzeptiert)</returns>
    private bool DoEinmaligAndVUs(Einmalig oEinmalig, Unterbrechung[] oVUs)
    {
        Log.Append(0, "DoEinmaligAndVUs");

        // 1) Instanz des Webservice erzeugen
        WsVersUnterbrechungenStrom oWS = new WsVersUnterbrechungenStrom();

        // 2) Transaktion beginnen
        BeginnTranAntwort oAnt =

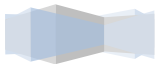
oWS.BeginnTransaktion(999999999, "1234567A", 1, iBerichtsJahr, false,
100, 100, 1, WsVersUnterbrechungenStrom.TransBeginnArtEnum.EchtDaten);

        if (oAnt.StatusWs != TransStatusEnum.Akzeptiert)
        {
            Log.Append(4, String.Format("BeginnTranAntwort.StatusWs [Beginn]: {0}", oAnt.StatusWs.ToString()));
            return false;
        }

        // 3) Einmaldaten senden
        TranAntwort oAntEinmalig = oWS.TransEinmalig(oAnt.Transaktionsnummer, oEinmalig);
        if (oAntEinmalig.StatusWs != TransStatusEnum.Akzeptiert)
        {
            Log.Append(4, String.Format("TranAntwort.StatusWs [Einmalig]: {0}", oAntEinmalig.StatusWs.ToString()));
            return false;
        }

        // 4) VU's senden

```



```

TranUnterbrAntwort oAntVUs = oWS.TransUnterbrechungen(oAnt.Transaktionsnummer, 1, oVUs);
    if (oAntVUs.StatusWs != TransStatusEnum.Akzeptiert)
    {

Log.Append(4, String.Format("TranUnterbrAntwort.StatusWs [Unterbr]: {0}", oAntVUs.StatusWs.ToString()));
        return false;
    }

    // 5) Abschluss der Transaktion
    TranAntwort oAntAbschluss = oWS.AbschlussTransaktion(_sTranNr);
    if (oAntAbschluss.StatusWs != TransStatusEnum.Akzeptiert)
    {

Log.Append(4, String.Format("TranAntwort.StatusWs [Abschluss]: {0}", oAntAbschluss.StatusWs.ToString()));
        return false;
    }

    return true;
}

```

Beispiel: Rückgabe von Einmaldaten (fiktive Daten)

```

/// <summary>
/// Rückgabe von Einmaldaten.
/// </summary>
private void DoRueckgabeEinmalig()
{
    WSVersUnterbrechungenStrom.WSVersUnterbrechungenStrom oWS = GetWS();

    DateTime dStart = DateTime.Now;
    WSVersUnterbrechungenStrom.RueckEinmaligAntwort oAnt;

    oAnt = oWS.RueckgabeEinmalig(_sTranNr);

    DateTime dEnde = DateTime.Now;

    toolStripLblStatus.Text = String.Format("Rueckgabe Unterbrechungen", oAnt.StatusWs);
    Log.Append(0, toolStripLblStatus.Text);

    DoLvItem(oAnt.StatusWs, dStart, dEnde, "RueckgabeEinmalig",
        "Rueckgabe Einmalig",
        String.Format("{0}, {1}", oAnt.StatusWs, oAnt.Meldung), oAnt);

    if (oAnt.EinmalDaten != null)
    {
        DoLvItem(oAnt.StatusWs, dStart, dEnde, "RueckgabeEinmalig",
            "Info zur Rückgabe",
            "Einmaldaten empfangen",
            oAnt.EinmalDaten);
    }
}

```




```

        Log.Append(0, "Einmaldaten empfangen");
    }
    else
    {
        Log.Append(4, "Einmaldaten sind NULL");
    }
}

```

Beispiel: Rückgabe von Unterbrechungen (fiktive Daten)

```

/// <summary>
/// Gibt die Unterbrechungen zu einem Paket zurück.<br/>
/// Ausgabe erfolgt im Control lvItem (ListView).
/// </summary>
/// <param name="iPaket">Nummer des Pakets</param>
private void DoRueckgabeUnterbrechungen(int iPaket)
{
    WSVersUnterbrechungenStrom.WsVersUnterbrechungenStrom oWS = GetWS();

    DateTime dStart = DateTime.Now;
    WSVersUnterbrechungenStrom.RueckUntbrAntwort oAnt;

    oAnt = oWS.RueckgabeUnterbrechungen(_sTranNr, iPaket);

    DateTime dEnde = DateTime.Now;

    toolStripLblStatus.Text = String.Format("Rueckgabe Unterbrechungen", oAnt.StatusWs);
    Log.Append(0, toolStripLblStatus.Text);

    DoLvItem(oAnt.StatusWs, dStart, dEnde, "RueckgabeUnterbrechungen",
        "Rueckgabe Unterbrechungen",
        String.Format("{0}, {1}", oAnt.StatusWs, oAnt.Meldung), oAnt);

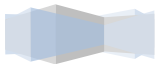
    if (oAnt.Unterbrechungen != null)
    {
        if (oAnt.Unterbrechungen.Length > 0)
        {
            int iVon = oAnt.Unterbrechungen[0].LfdNr;

            int iBis = oAnt.Unterbrechungen[oAnt.Unterbrechungen.Length - 1].LfdNr;

            DoLvItem(oAnt.StatusWs, dStart, dEnde, "RueckgabeUnterbrechungen",
                "Info zur Rückgabe",
                String.Format("Von: {0} bis {1}, Anzahl: {2} empfangen", iVon, iBis, oAnt.Unterbrechungen.Length), oAnt.Unterbrechungen);

            Log.Append(0, String.Format("Von: {0} bis {1}, Anzahl: {2} empfangen", iVon, iBis, oAnt.Unterbrechungen.Length));
        }
    }
}

```



```
    }  
    }  
    else  
    {  
        Log.Append(4, "Unterbrechungen sind NULL");  
    }  
}
```

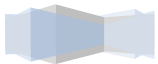


Abbildung 6: Diagramm - Abbruch einer Transaktion

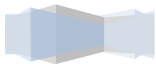
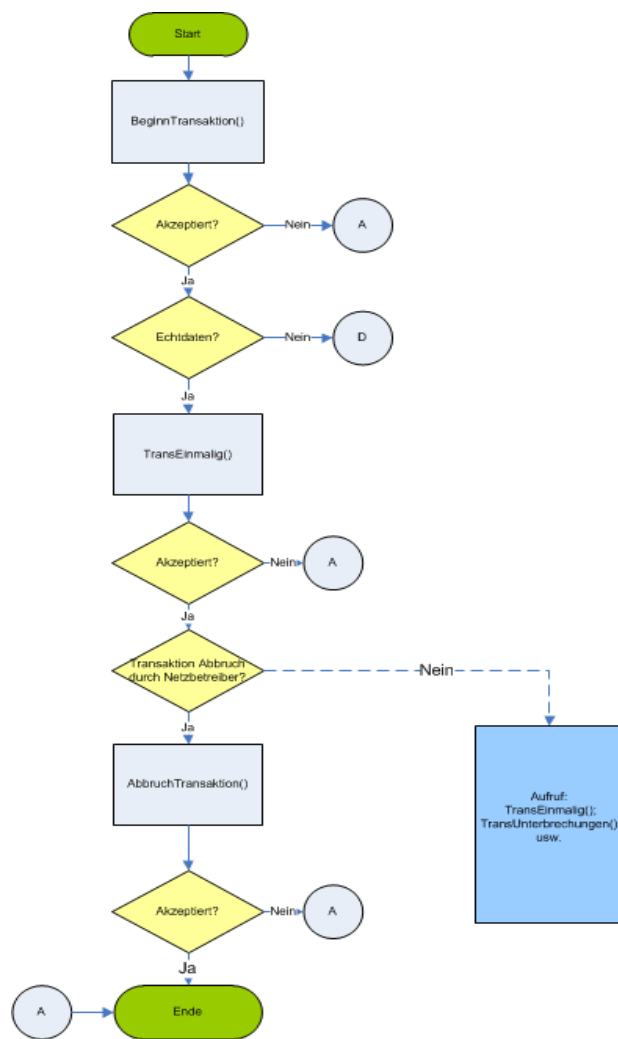
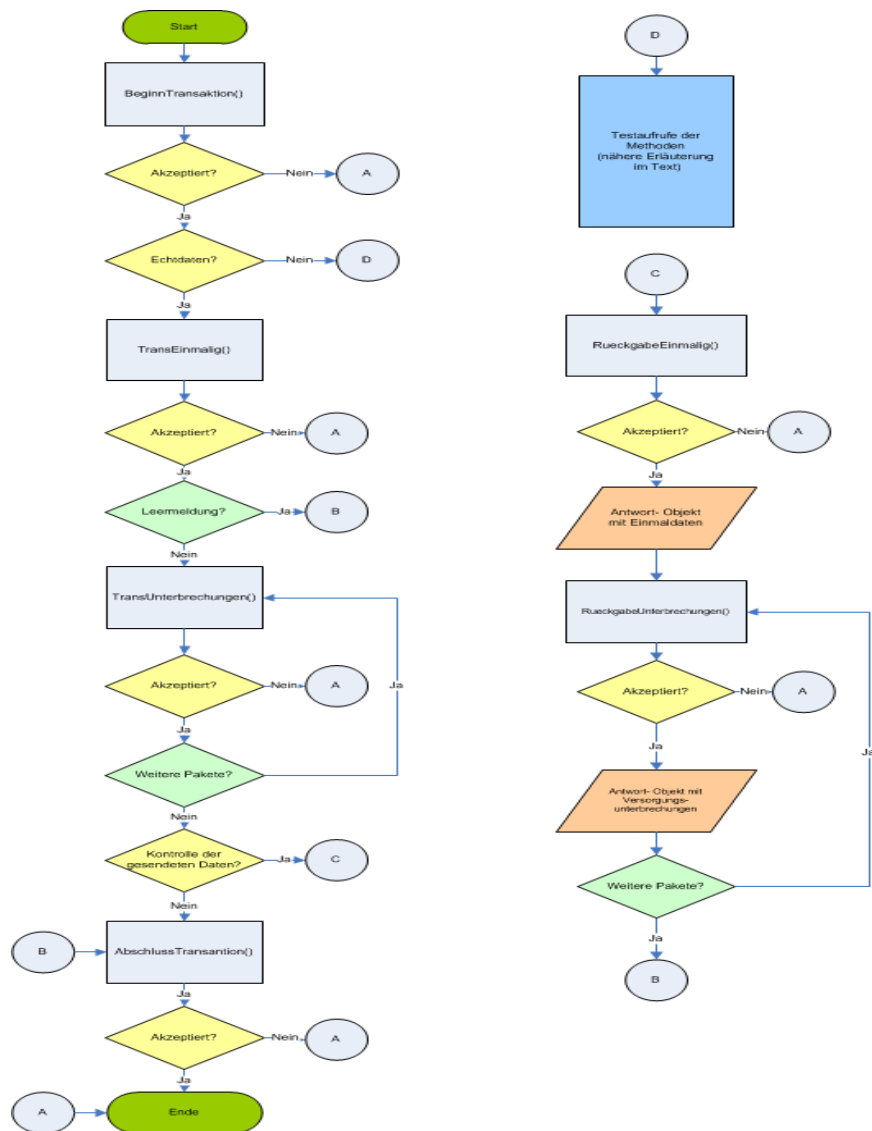


Abbildung 7: Diagramm - Echtdaten ohne Rückgabe der Daten



(Ende des Dokuments)

