



XML-Portoptimierung, Manager Presse Distribution XML

1	Einleitung	4
1.1	Einsatzzweck	4
1.2	XML-Portooptimierung	4
1.3	Manager Presse Distribution XML	4
1.4	Alternative Produkte	4
1.5	Zielgruppe dieses Handbuchs	4
2	Installation / Konfiguration / Updates	5
2.1	Systemanforderungen	5
2.2	Installation	5
2.3	Verzeichnisstruktur	6
2.4	Lizenzierung	6
2.5	Updates	7
2.5.1	Aktualisieren von Daten-Dateien	7
2.5.2	Aktualisieren der Software	7
3	Grundlagen	8
3.1	Architekturüberblick	8
3.2	Testanwendung	9
3.2.1	Zweck der Testanwendung	9
3.2.2	Aufbau der Testanwendung	9
3.2.3	Optimieren mit der Testanwendung	10
3.3	Konfiguration des Optimierungsaufrufs	11
4	Prozessintegration	13
4.1	Ansteuerung der Optimierungs-DLL	13
4.1.1	Beispiel in C#	14
4.1.2	Beispiel in Java	14
4.1.3	Beispiel in Visual Basic	15
4.2	Eingabedatei	15
4.2.1	Struktur der Eingabedatei	15
4.2.2	Ausgewählte Elemente der Eingabedatei	18
4.3	Ausgabedatei	19
4.3.1	Struktur der Ausgabedatei	19
4.3.2	Ausgewählte Elemente der Ausgabedatei	20
5	Anhang	24
5.1	Versionsübersicht	24
5.2	Funktionsüberblick XML-Portooptimierung	24
5.3	Funktionsüberblick Manager Presse Distribution XML	25
6	Index	26

1 Einleitung

1.1 Einsatzzweck

Bei der XML-Portooptimierung und dem Manager Presse Distribution XML handelt es sich um Software-Module, die Sie bei der Versandvorbereitung und -optimierung Ihrer Aussendungen unterstützen. Dazu werden von den Modulen die Adressmenge sowie die notwendigen Fertigungsparameter eingelesen, den Anforderungen und Allgemeinen Geschäftsbedingungen der Deutschen Post gemäß optimiert und das Ergebnis zur weiteren Verarbeitung in einer Datei zur Verfügung gestellt. Zusätzlich können die zur Einlieferung notwendigen Unterlagen in Form eines PDF-Dokuments erzeugt werden.

Beide Produkte sind zur Integration in bestehende Geschäftsprozesse im Batch-Betrieb konzipiert. Entsprechend müssen sie von einer Drittsoftware angesteuert und die Ergebnisse weiter verarbeitet werden.

1.2 XML-Portooptimierung

Verwenden Sie die XML-Portooptimierung, wenn Sie den Versand von Sendungen im Bereich „Brief“ vorbereiten möchten. Die XML-Portooptimierung unterstützt die Produkte Infopost, Infobrief, Katalog-Einzelversand sowie Infopost International, Infobrief International und Briefe International zum Kilotarif. Entsprechend Ihrer Vorgaben erfolgt eine Optimierung zur günstigsten Versandart.

1.3 Manager Presse Distribution XML

Verwenden Sie den Manager Presse Distribution XML zur Vorbereitung des Versands im Bereich „Presse“. Unterstützt werden Postvertriebsstücke, Pressesendungen, Streifenbandzeitungen sowie Presse und Buch International zum Kilotarif. Die Optimierung erfolgt anhand der Datei ZEBU (für den nationalen Versand).

1.4 Alternative Produkte

Wie bereits in *Einsatzzweck* beschrieben sind die XML-Portooptimierung und der Manager Presse Distribution XML zur Integration in einen bestehenden Prozess im Batch-Betrieb vorgesehen. Beide Module besitzen daher keine grafische Benutzerschnittstelle und werden ausschließlich über andere Prozesse und nicht unmittelbar durch einen Anwender gesteuert.

Wenn Sie eine Standalone-Lösung für die Versandvorbereitung benötigen, empfehlen wir die Produkte „Infopost-Manager“ und „Manager Presse Distribution“. Bei diesen handelt es sich um komfortable Anwendungen für das Betriebssystem MS Windows. Zusätzlich zur Versandvorbereitung und Optimierung bieten beide Produkte umfassende Features im Bereich Adressveredelung.

Weitere Informationen finden Sie im Internet:

► <http://www.infopost-manager.de> bzw. ► <http://www.manager-presse-distribution.de>

1.5 Zielgruppe dieses Handbuchs

Dieses Handbuch richtet sich an Systemintegratoren, die das Modul XML Portooptimierung oder Manager Presse Distribution XML in einen Geschäftsprozess einbinden möchten. Kenntnisse im Bereich von Windows-Betriebssystemen und XML-Dateien werden vorausgesetzt.

2 Installation / Konfiguration / Updates

Inhalt

2.1	Systemanforderungen	5
2.2	Installation	5
2.3	Verzeichnisstruktur	6
2.4	Lizenzierung	6
2.5	Updates	7
2.5.1	Aktualisieren von Daten-Dateien	7
2.5.2	Aktualisieren der Software	7

2.1 Systemanforderungen

Bei den Modulen XML-Portooptimierung und Manager Presse Distribution XML handelt es sich um einen Satz von funktionalen Dynamic Link Libraries (DLL) für das Betriebssystem Windows von Microsoft. Entsprechend muss auf dem PC-System, auf dem die Module ausgeführt werden sollen, das Betriebssystem Windows installiert sein.

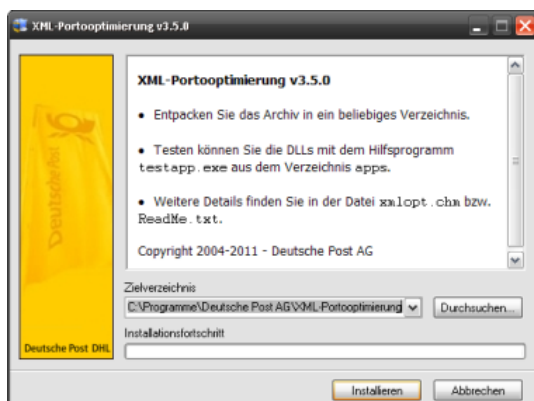
Die Module können mit jeder Programmiersprache, die den Aufruf von Windows-DLLs unterstützt, angesteuert werden. Beispiele für den Aufruf mit C#, Visual Basic und Java finden Sie im Kapitel *Prozessintegration* (auf Seite 13).

Für die Installation eines Moduls werden ca. 40 MB freie Festplattenkapazität benötigt. Bitte beachten Sie, dass während der Optimierung temporäre Dateien erzeugt werden, deren Größen von der Eingabedatei abhängig sind. Sorgen Sie daher immer für ausreichend freie Festplattenkapazität.

2.2 Installation

Die Module werden als selbstextrahierendes ZIP-Archiv in Form einer ausführbaren Windows-Anwendung ausgeliefert.

Durch das Ausführen der Anwendung bereiten Sie den Entpack-Vorgang vor.



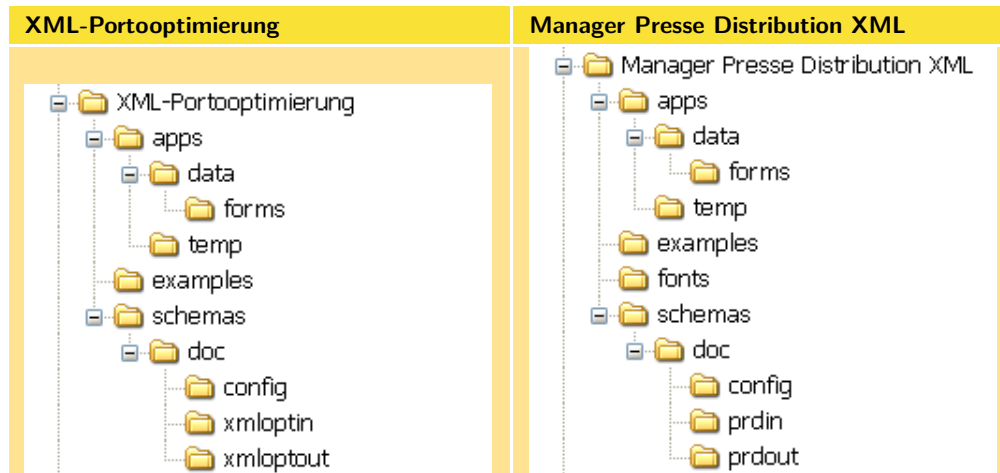
Sie können das Modul in ein beliebiges Verzeichnis, welches den Systemanforderungen entspricht, installieren. Klicken Sie auf die *Durchsuchen*-Schaltfläche, um ein Zielverzeichnis auszuwählen oder behalten Sie das vorgeschlagene Standardverzeichnis bei.

Starten Sie im Anschluss das Entpacken des Moduls durch einen Klick auf die *Installieren*-Schaltfläche.

Hinweis: Im Folgenden wird zur Vereinfachung davon ausgegangen, dass das Modul im Standard-Verzeichnis installiert wurde.

2.3 Verzeichnisstruktur

Der Installer erzeugt unterhalb des von Ihnen ausgewählten Verzeichnisses eine eigene Verzeichnisstruktur. Es wird empfohlen, diese Verzeichnisstruktur beizubehalten. So ersparen Sie sich zusätzlichen Konfigurationsaufwand.



Nachfolgend werden die einzelnen Verzeichnisse erläutert:

Verzeichnis	Inhalt
apps	In diesem Verzeichnis befinden sich die DLL-Dateien, die das Optimierungsmodul implementieren. Darüber hinaus finden Sie in diesem Verzeichnis eine Testanwendung (auf Seite 9) sowie einen Java-Wrapper (auf Seite 14) und eine Hilfedatei mit der Beschreibung der DLL-Aufrufe (auf Seite 13) und schließlich ein PDF-Dokument mit den Allgemeinen Geschäftsbedingungen für das Modul.
apps\data	In diesem Verzeichnis finden Sie Dateien, die von dem Optimierungsmodul zur Durchführung der Optimierung benötigt werden.
apps\data\forms	Hier sind Dateien abgelegt, die zur Erzeugung der Einlieferungs- und Fertigungsdokumente benötigt werden.
apps\temp	Arbeitsverzeichnis in dem während der Optimierung temporäre Dateien angelegt werden.
apps\examples	Der Ordner enthält Beispieldateien anhand derer sich der Aufbau der Ein- und Ausgabedateien erkennen lässt (auf Seite 10).
schemas	Enthält die Schema Dateien (xsd) für die Eingabe-, Ausgabe- und Konfiguration.
schemas\doc\config	Enthält die Beschreibung der Konfiguration als HTML-Dokument.
schemas\doc\xmloptin (prdin)	Enthält die Beschreibung der Eingabedatei als HTML-Dokument.
schemas\doc\xmloptout (prdout)	Enthält die Beschreibung der Ausgabedatei als HTML-Dokument.

2.4 Lizenzierung

Vor der ersten Verwendung muss das Optimierungsmodul lizenziert werden. Dies geschieht über eine Lizenzdatei, die speziell für Sie/Ihr Unternehmen personalisiert wurde.

Zur Lizenzierung kopieren Sie die Lizenzdatei `license.dat`, die Sie von der Deutschen Post erhalten haben, in das `apps`-Verzeichnis Ihrer Installation - siehe *Verzeichnisstruktur* (auf dieser Seite).

Zu Testzwecken stellt die Deutsche Post zeitlich befristete Demolizenzen zur Verfügung. Der Lizenzierungsvorgang unterscheidet sich nicht von dem einer Volllizenzierung.

2.5 Updates

2.5.1 Aktualisieren von Daten-Dateien

Sowohl die XML-Porttooptimierung als auch der Manager Presse Distribution XML benötigen zur Optimierung Dateien, die einer regelmäßigen Aktualisierung unterliegen. Dabei handelt es sich zum Beispiel um das Postleitzahlenverzeichnis, die Datei ZEBU oder die Preis- und Produktliste.

Die Deutsche Post stellt die aktualisierten Dateien in Form von ZIP-Archiven zum Download bereit. Die Installation der Updates gestaltet sich einfach. Sie müssen die Dateien des ZIP-Archivs lediglich in das **Data**-Verzeichnis (auf der vorherigen Seite) Ihrer Installation entpacken.

2.5.2 Aktualisieren der Software

Die Software-Module „XML-Porttooptimierung“ und „Manager Presse Distribution XML“ unterliegen einer ständigen Weiterentwicklung. Dabei werden neben der Erweiterung des Funktionsumfangs der Module auch Anpassungen an geänderte Allgemeine Geschäftsbedingungen der Deutschen Post in die Software übernommen. Damit Ihre Einlieferungen stets konform zu den gültigen AGB sind, wird der Abschluss eines Software-Updates-Abonnements empfohlen.

Manche Änderungen an den Modulen erfordern Anpassungen am Schema der Ein- (auf Seite 15) und/oder Ausgabe-Datei (auf Seite 19). Dabei wird versucht, eine Abwärtskompatibilität zu gewährleisten. Da dies aber nicht in allen Fällen möglich ist, kann es vorkommen, dass nach dem Einspielen eines Updates Anpassungen in Ihrem Geschäftsprozess notwendig sind. Die Änderungen am Ein- und Ausgabeschema können sie der Dokumentation des jeweiligen Produkts entnehmen.

Das Software-Update wird als selbstentpackendes Archiv (EXE-Datei) als Download zur Verfügung gestellt. Nach dem Ausführen müssen Sie im Installationsdialog unter „Zielverzeichnis“ das Verzeichnis Ihrer Installation angeben und im Anschluss auf *Installieren* klicken. Sollte der Dialog *Ersetzen von Dateien bestätigen*“ erscheinen, dann bestätigen Sie diesen bitte mit *Alle*.

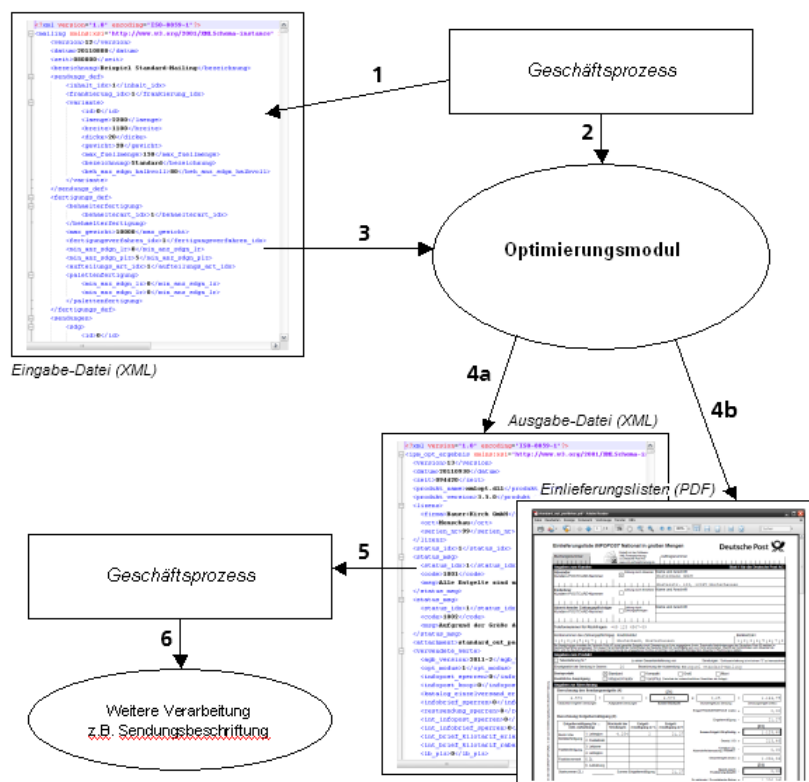
3 Grundlagen

Inhalt

3.1	Architekturüberblick	8
3.2	Testanwendung	9
3.2.1	Zweck der Testanwendung	9
3.2.2	Aufbau der Testanwendung	9
3.2.3	Optimieren mit der Testanwendung	10
3.3	Konfiguration des Optimierungsaufrufs	11

3.1 Architekturüberblick

Wie bereits in der *Einleitung* (auf Seite 4) beschrieben, ist das Optimierungsmodul für den Batch-Betrieb konzipiert. Die folgende Grafik zeigt schematisch die Integration des Optimierungsmoduls in den Geschäftsprozess:



- In Schritt 1 erstellt der Geschäftsprozess die Eingabedatei für das Optimierungsmodul. Die Eingabedatei enthält alle Parameter, die für die Optimierung notwendig sind. Dies sind z.B. Angaben zu Absender, Einlieferer und Zahlungspflichten, Sendungsgewicht und Fertigungsvorgaben für die Palettenfertigung. Darüber hinaus enthält die Eingabedatei auch die Postleitzahlen und Länderkennungen der anzuschreibenden Adressen.
- In Schritt 2 startet der Geschäftsprozess das Optimierungsmodul.
- Dieses liest in Schritt 3 die Eingabedatei ein und führt die Optimierung aus.
- Als Ergebnis der Optimierung wird im Schritt 4 die Ausgabedatei und auf Wunsch die Einlieferungslisten erzeugt.

- In den weiteren Schritten erfolgt dann die Verarbeitung der Ausgabedatei durch den Geschäftsprozess. Dazu wird die Ausgabedatei zunächst vom Geschäftsprozess eingelesen (Schritt 5).
- Die eigentliche Verarbeitung besteht dann zum Beispiel darin, die Versandstücke anhand der von der Optimierung ermittelten und in der Ausgabedatei gespeicherten Reihenfolge zu beschriften (Schritt 6).

3.2 Testanwendung

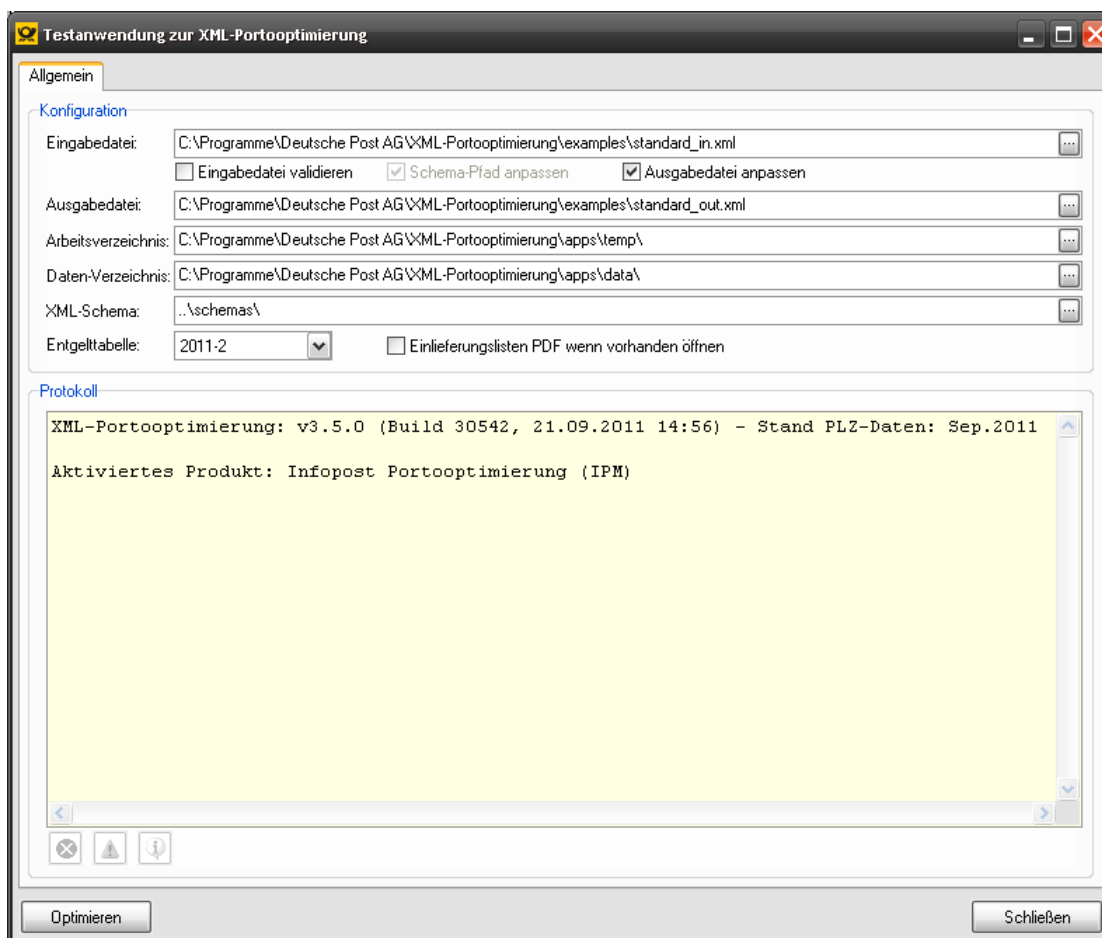
3.2.1 Zweck der Testanwendung

Im `app`-Verzeichnis (auf Seite 6) befindet sich eine Testanwendung (`testapp.exe`). Mit Hilfe dieser Anwendung können Sie das Optimierungsmodul zu Testzwecken ansteuern. Sie haben dadurch die Möglichkeit, den vollen Funktionsumfang des Optimierungsmoduls zu testen, ohne gleich die Integration in Ihren Geschäftsprozess implementieren zu müssen. Später kann die Anwendung ebenfalls genutzt werden, um Fehler in einer erzeugten XML-Eingabedatei zu finden.

Die Testanwendung dient lediglich der Ansteuerung des Optimierungsmoduls. Sie kann keine Eingabedateien erzeugen. Für erste Optimierungsversuche und zur Verdeutlichung des Dateiaufbaus finden Sie aber im `examples`-Verzeichnis (auf Seite 6) einige fertige Eingabedateien.

3.2.2 Aufbau der Testanwendung

Starten Sie die Testanwendung `testapp.exe` aus dem `app`-Verzeichnis (auf Seite 6). Der folgende Dialog wird angezeigt:



Die Testanwendung gliedert sich in die Bereiche „Konfiguration“ und „Protokoll“ und besitzt neben der Schaltfläche *Schließen* zum Beenden der Anwendung nur noch die Schaltfläche *Optimieren* um den Optimierungsvorgang zu starten.

Im Konfigurationsabschnitt werden die zum Starten des Optimierungsmoduls notwendigen Einstellungen vorgenommen. Im Einzelnen sind dies:

Verzeichnis	Inhalt
Eingabedatei	Geben Sie hier an, welche Eingabedatei das Optimierungsmodul verwenden soll.
Ausgabedatei	Geben Sie hier an, in welche Datei das Optimierungsmodul die Ergebnisse schreiben soll.
Arbeitsverzeichnis	Geben Sie hier den Pfad zum temp-Verzeichnis aus dem Installationsprozess an (auf Seite 6).
Datenverzeichnis	Geben Sie hier den Pfad zum data-Verzeichnis aus dem Installationsprozess an (auf Seite 6).
XML-Schema	Geben Sie hier den Pfad zum schemas-Verzeichnis aus dem Installationsprozess an (auf Seite 6)

Über die Option *Eingabedatei validieren* können Sie das Optimierungsmodul dazu veranlassen, die XML-Eingabedatei vor der Verarbeitung gegen das XML-Schema (xsd) zu validieren. Treten dabei Fehler auf, wird keine Optimierung durchgeführt. Sollte die Eingabedatei nicht den richtigen Pfad zum Schema besitzen, so können Sie durch Aktivieren der Option *Schema-Pfad anpassen*, diesen Pfad automatisch vom Optimierungsmodul korrigieren lassen. Das Optimierungsmodul verwendet dabei den unter "XML-Schema" angegebenen Pfad.

Die Option *Ausgabedatei anpassen* bewirkt, dass der Name der zu erzeugenden Ausgabedatei automatisch an den Namen der Eingabedatei angepasst wird. Diese Anpassung erfolgt unmittelbar nach der Auswahl der Eingabedatei über die ...-Schaltfläche. Enthält der Dateiname der Eingabedatei den Text `_in`, wird dieser in der Ausgabedatei durch `_out` ersetzt. Ist kein entsprechender Text enthalten, wird als Name für die Ausgabedatei der Name der Eingabedatei erweitert um den Zusatz `_out` verwendet.

Über die Liste "Entgelttabelle" können Sie festlegen, mit welcher Entgelttabelle bzw. mit welcher AGB-Version die Optimierung durchgeführt werden soll, falls in der Eingabedatei kein Übergabedatum spezifiziert ist.

Das Setzen der Option *Einlieferungslisten PDF wenn vorhanden öffnen* führt dazu, dass eine vom Optimierungsmodul erzeugte PDF-Datei mit Einlieferungslisten automatisch angezeigt wird. Bitte beachten Sie, dass dazu eine Anwendung auf dem PC registriert sein muss, die die Anzeige von PDF-Dokumenten ermöglicht.

Im Protokollabschnitt werden Statusinformationen zum Optimierungsvorgang ausgegeben. Dabei kann es sich um allgemeine Informationen, Fehler und Warnungen handeln. Über die drei Schaltflächen unterhalb des Ausgabefensters können Sie unmittelbar zu den einzelnen Typen springen.

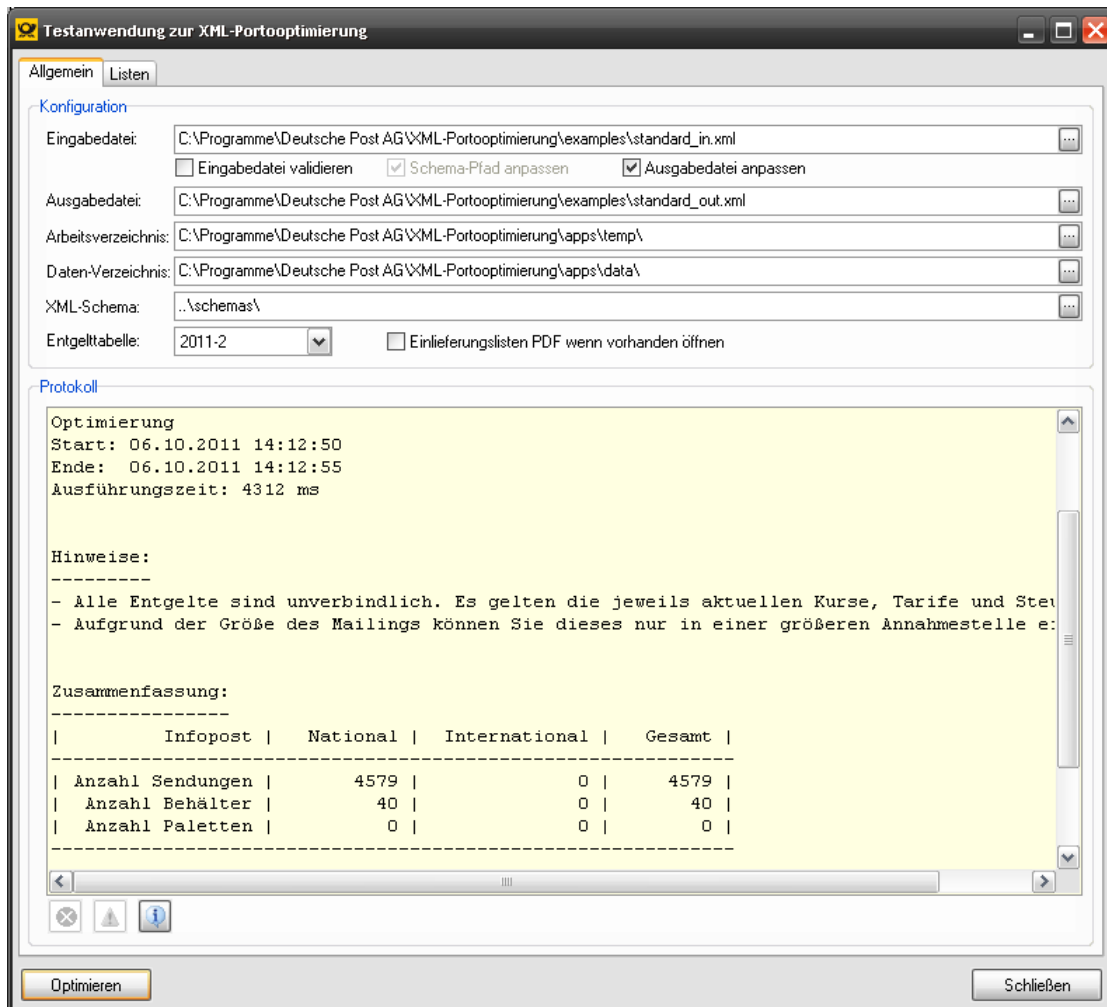
3.2.3 Optimieren mit der Testanwendung

Starten Sie die Testanwendung und aktivieren Sie die Option *Ausgabedatei anpassen*. Setzen Sie nun die Eingabedatei auf die Datei `standard_in.xml` aus dem `examples`-Verzeichnis Ihrer Installation. Nachdem dies geschehen ist, sollte der Pfad und Dateiname der Ausgabedatei automatisch angepasst worden sein, d.h. der Pfad ist der gleiche wie der der Eingabedatei und der Dateiname lautet `standard_out.xml`.

Überprüfen Sie, ob die übrigen Verzeichnisse Ihrer Installation entsprechend eingestellt sind. Sollte dies nicht der Fall sein, korrigieren Sie diese.

Wählen Sie in der Liste „Entgelttabelle“ die aktuellste Liste aus und klicken Sie anschließend auf *Optimieren*.

War alles richtig eingestellt, sollte nach kurzer Optimierung im Protokoll-Fenster eine einfache Tabelle mit einem Teil der Optimierungsergebnisse zu sehen sein. Zusätzlich wird die Start- und Endzeit sowie die Dauer der Optimierung ausgegeben. Wenn bei der Optimierung Hinweise erzeugt wurden, sind diese ebenfalls im Protokollfenster zu lesen.



Öffnen Sie das `examples`-Verzeichnis. In diesem ist eine neu erstellte Datei `standard_out.xml` mit allen Optimierungsergebnissen enthalten. Diese Datei können Sie mit einem (XML/Text)-Editor Ihrer Wahl öffnen.

Weiterhin existiert eine PDF-Datei mit dem Name `standard_out_postlisten.pdf`. Wenn Sie diese öffnen, sehen Sie die bei diesem Optimierungsdurchgang erzeugten Einlieferungsunterlagen.

3.3 Konfiguration des Optimierungsaufrufs

Wie Sie bereits wissen, sind an einer Optimierung mindestens zwei Dateien beteiligt, die Eingabe- und die Ausgabe-Datei. Das Verhalten der Optimierung kann jedoch noch über weitere Parameter beeinflusst werden (dazu später mehr). Zunächst ist wichtig, dass alle Parameter beim Start der Optimierung an das Optimierungsmodul übergeben werden müssen.

Damit das Optimierungsmodul die Parameter verarbeiten kann, müssen diese in einer XML-Struktur übergeben werden.

Wenn Sie mit der Testanwendung bereits eine Optimierung durchgeführt haben, finden Sie im `apps`-Verzeichnis die Datei `config.xml`. Diese Datei enthält die Konfigurationsstruktur für den letzten Aufruf:

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<configuration>
  <input_xml>C:\Programme\Deutsche Post AG\XML-Portoptimierung\examples\standard_in.xml</input_xml>
  <output_xml>C:\Programme\Deutsche Post AG\XML-Portoptimierung\examples\standard_out.xml</output_xml>
  <projectdir>C:\Programme\Deutsche Post AG\XML-Portoptimierung\apps\temp\</projectdir>
  <datadir>C:\Programme\Deutsche Post AG\XML-Portoptimierung\apps\data\</datadir>
  <schemadir>..\schemas\</schemadir>
  <agb_version>2011-2</agb_version>
  <delete_tempfiles>0</delete_tempfiles>
  <internet>
  </internet>
  <am>
  </am>
</configuration>

```

In der Konfiguration erforderlich sind die Tags `<input_xml>` und `<output_xml>`. Ersterer muss den Dateinamen (inkl. Pfad) zur Eingabedatei enthalten, letzterer den der Ausgabedatei, die von dem Optimierungsmodul erzeugt werden soll.

Die Struktur der Konfiguration finden Sie in der Dokumentation im Verzeichnis `schemas\doc\config`. Die Beschreibung liegt im Html-Format vor. Beginnen Sie mit der Datei `index.html`. Wenn Sie geübt im Umgang mit dem XML-Dateien sind, können Sie die Informationen auch der Schemadatei `config.xsd` im `schemas`-Verzeichnis entnehmen.

Auf einzelne Konfigurations-Parameter wird in den nachfolgenden Kapiteln noch eingegangen.

4 Prozessintegration

Inhalt

4.1 Ansteuerung der Optimierungs-DLL	13
4.1.1 Beispiel in C#	14
4.1.2 Beispiel in Java	14
4.1.3 Beispiel in Visual Basic	15
4.2 Eingabedatei	15
4.2.1 Struktur der Eingabedatei	15
4.2.2 Ausgewählte Elemente der Eingabedatei	18
4.3 Ausgabedatei	19
4.3.1 Struktur der Ausgabedatei	19
4.3.2 Ausgewählte Elemente der Ausgabedatei	20

4.1 Ansteuerung der Optimierungs-DLL

Die Integration der Optimierung in den Geschäftsprozess besteht aus mehreren Schritten:

- Erstellen der Eingabe-Datei (XML)
- Ansteuerung der Optimierungs-DLL
- Verarbeiten der Optimierungsergebnisse (XML)

Auch wenn es erst an zweiter Stelle steht, so soll doch zunächst auf die Ansteuerung der Optimierungs-DLL eingegangen werden.

Wie bereits in der Einführung gesagt, handelt es sich bei den Optimierungsmodulen um Windows-DLLs. Im Falle der XML-Portooptimierung ist es die `xmlopt.dll`, beim Manager Presse Distribution XML die `xm1prd.dll` im `apps`-Verzeichnis der jeweiligen Installation. Beide rufen ihrerseits für den eigentlichen Optimierungsvorgang wiederum DLLs auf und verwenden Dateien, die im `data`-Verzeichnis liegen.

In der Beschreibung des *Optimierungsaufrufs* (auf Seite 11) wurde bereits festgestellt, dass das Optimierungsmodul mit verschiedenen Parametern ausgestattet werden muss, damit eine Optimierung durchgeführt werden kann. Ebenfalls ist aus diesem Kapitel bereits bekannt, dass diese Parameter in einer XML-Struktur bereitgestellt werden müssen.

Um eine Optimierung auszuführen, muss die aufrufende Anwendung die Funktion

```
XMLOPT_InitAndExecute(Configuration, ExecResult, ExecResultSize);
```

der Optimierungs-DLL aufrufen.

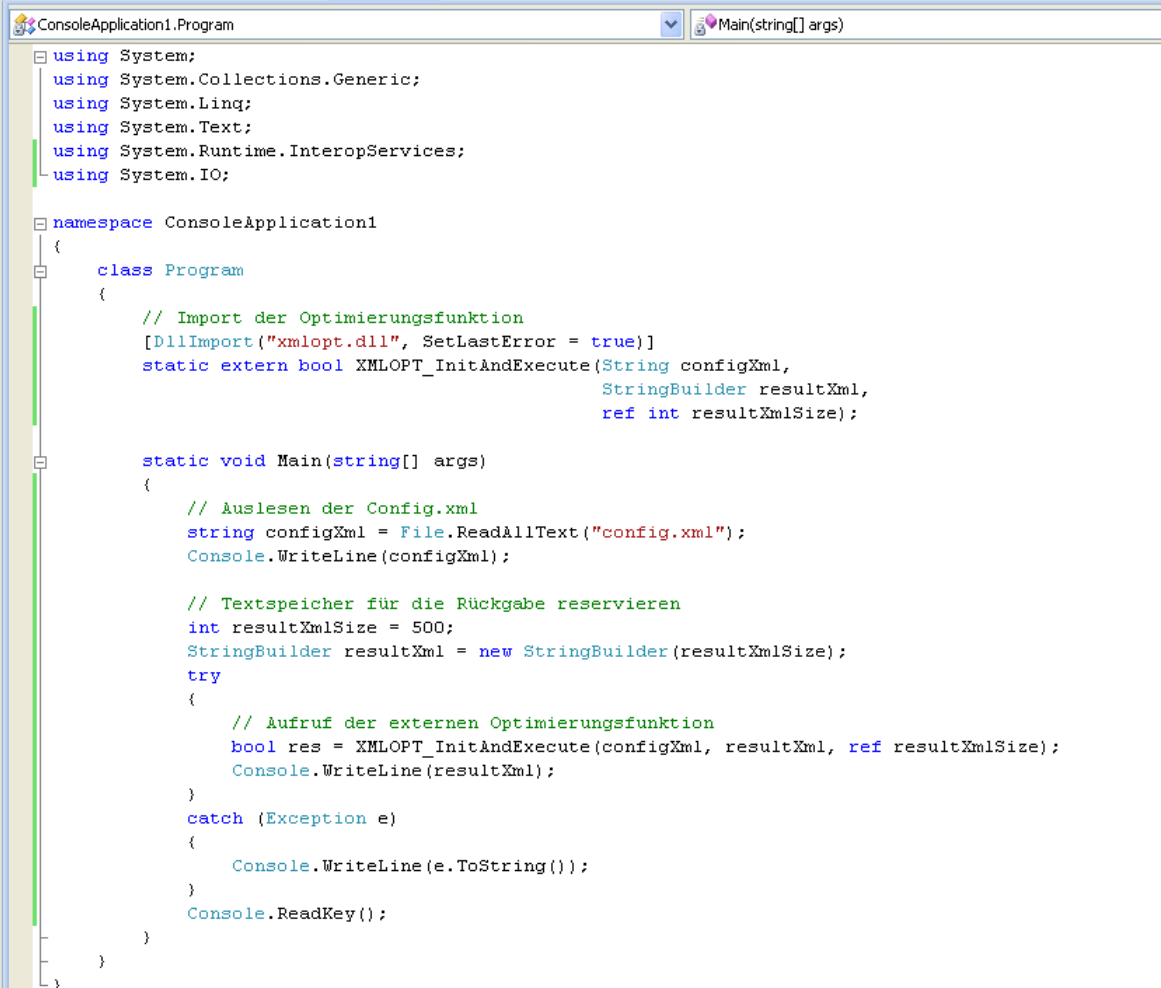
Der Rückgabewert der Funktion ist vom Typ `Boolean` und `TRUE`, wenn während der Optimierung kein Fehler aufgetreten ist. Anderenfalls ist er `FALSE` und im zweiten Parameter (Typ `PChar`) `ExecResult` werden Informationen (Text) zum Fehler zurückgeliefert. Die Länge des Rückgabetextes enthält der dritte Parameter (Typ `Integer`).

Bislang noch nicht beschrieben wurde der erste Parameter `Configuration`. Bei diesem handelt es sich ebenfalls um einen Parameter vom Typ `PChar`, also einem Zeiger auf eine Zeichenkette. Dieser Zeiger muss auf einen String zeigen, der die Konfigurationsstruktur enthält. Zur Verdeutlichung finden Sie in den nächsten Abschnitten Beispiele in C#, Visual Basic und Java.

Neben dem Aufruf `XMLOPT_InitAndExecute` stellen die Optimierungs-DLLs (`xmlopt.dll`, `xm1prd.dll`) noch weitere Funktionen zur Verfügung. Eine detaillierte Beschreibung aller Funktionsaufrufe finden Sie in der entsprechenden Hilfedatei (`xmlopt.chm`, `xm1prd.chm`) im `apps`-Verzeichnis.

4.1.1 Beispiel in C#

Die nachfolgende Abbildung zeigt beispielhaft den Aufruf der XML-Portooptimierung aus einer mit C# erstellten Konsolenanwendung:



```

ConsoleApplication1.Program
Main(string[] args)
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Runtime.InteropServices;
using System.IO;

namespace ConsoleApplication1
{
    class Program
    {
        // Import der Optimierungsfunktion
        [DllImport("xmlopt.dll", SetLastError = true)]
        static extern bool XMLOPT_InitAndExecute(String configXml,
            StringBuilder resultXml,
            ref int resultXmlSize);

        static void Main(string[] args)
        {
            // Auslesen der Config.xml
            string configXml = File.ReadAllText("config.xml");
            Console.WriteLine(configXml);

            // Textspeicher für die Rückgabe reservieren
            int resultXmlSize = 500;
            StringBuilder resultXml = new StringBuilder(resultXmlSize);
            try
            {
                // Aufruf der externen Optimierungsfunktion
                bool res = XMLOPT_InitAndExecute(configXml, resultXml, ref resultXmlSize);
                Console.WriteLine(resultXml);
            }
            catch (Exception e)
            {
                Console.WriteLine(e.ToString());
            }
            Console.ReadKey();
        }
    }
}

```

4.1.2 Beispiel in Java

Für einen vereinfachten Aufruf der Optimierung aus einer Java-Anwendung heraus werden spezielle Wrapper-Klassen zur Verfügung gestellt. Verwenden können Sie diese, indem Sie das JAR-Archiv `xmlopt.jar` aus dem Verzeichnis `apps` in Ihre Java-Anwendung einbinden.

Der folgende Quelltextausschnitt zeigt beispielhaft den Aufruf der Optimierung mit Hilfe der Wrapper-Klassen:

```

import java.io.FileNotFoundException;
import java.io.IOException;

import de.bauerkirch.ipm.Optimierung;
import de.bauerkirch.xmlopt.ConfigReader;
import de.bauerkirch.xmlopt.OptimierungsException;

public class OptimierungTest (

public static void main(String[] args) (
    try (
        if (args.length == 1) (
            Optimierung opt = new Optimierung();
            System.out.println("Version der Optimierung: " + opt.GetVersion());
            System.out.println("Version der Postleitdaten: " +

```

```

        opt.GetPLZDataBuildDate("data"));
        System.out.println("Ergebnis der Optimierung: " +
            opt.Optimiere(ConfigReader.readFile(args[0])));
    ) else (
        System.out.println("Java Anbindung der XML-Portooptimierung");
        System.out.println("USAGE: java -jar xmlopt.jar <configfile>");
    )
) catch (FileNotFoundException e) (
    System.out.println("Konfigurationsdatei nicht gefunden:" + e.getMessage());
) catch (IOException e) (
    System.out.println("Fehler beim Lesen der Konfigurationsdatei:" + e.getMessage());
) catch (OptimierungsException e) (
    System.out.println("Fehler beim Optimieren:" + e.getMessage());
)
)
)
)

```

4.1.3 Beispiel in Visual Basic

Die nachfolgende Abbildung zeigt beispielhaft den Aufruf der XML-Portooptimierung aus einem VBA-Modul in Microsoft Access:

```

db1 - Modul1 (Code)
(Allgemein) (Deklarationen)
Option Compare Database

Declare Function XMLOPT_InitAndExecute Lib "C:\Programme\Deutsche Post AG\XML-Portooptimierung\apps\xmlopt.dll" _
    (ByVal Configuration As String, ByVal ExecResult As String, ByRef ExecResultSize As Long) As Boolean

Public Sub ExecuteOptimierung()

    Dim Configuration As String
    Dim ResultText As String
    Dim ResultSize As Long
    Dim bRes As Boolean

    ' Config-String setzen
    Configuration = "<?xml version=""1.0"" encoding=""ISO-8859-1""?>" & _
        "<configuration>" & _
        "<input_xml>C:\Programme\Deutsche Post AG\XML-Portooptimierung\examples\standard_in.xml</input_xml>" & _
        "<output_xml>C:\Programme\Deutsche Post AG\XML-Portooptimierung\examples\standard_out.xml</output_xml>" & _
        "<projectdir>C:\Programme\Deutsche Post AG\XML-Portooptimierung\apps\temp</projectdir>" & _
        "<datadir>C:\Programme\Deutsche Post AG\XML-Portooptimierung\apps\data</datadir>" & _
        "<schemadir>..\schemas</schemadir>" & _
        "<agb_version>2011-2</agb_version>" & _
        "<delete_tempfiles>0</delete_tempfiles>" & _
        "</configuration>"

    ' Rückgabe-XML String ausreichend dimensionieren
    ResultText = String(2000, Chr(32))
    ResultSize = Len(ResultText)

    ' Optimierung ausführen
    bRes = XMLOPT_InitAndExecute(Configuration, ResultText, ResultSize)

    If bRes Then
        MsgBox "Optimierung erfolgreich durchgeführt"
    Else
        MsgBox Trim$(ResultText)
    End If

End Sub

```

4.2 Eingabedatei

Über die Eingabedatei erhält das Optimierungsmodul alle zur Optimierung notwendigen Daten. Dies sind zum Beispiel Angaben zur Sendung, Frankierung und Fertigung, aber auch die anzuschreibenden Adressen.

Die Eingabedatei muss vom Geschäftsprozess erzeugt werden.

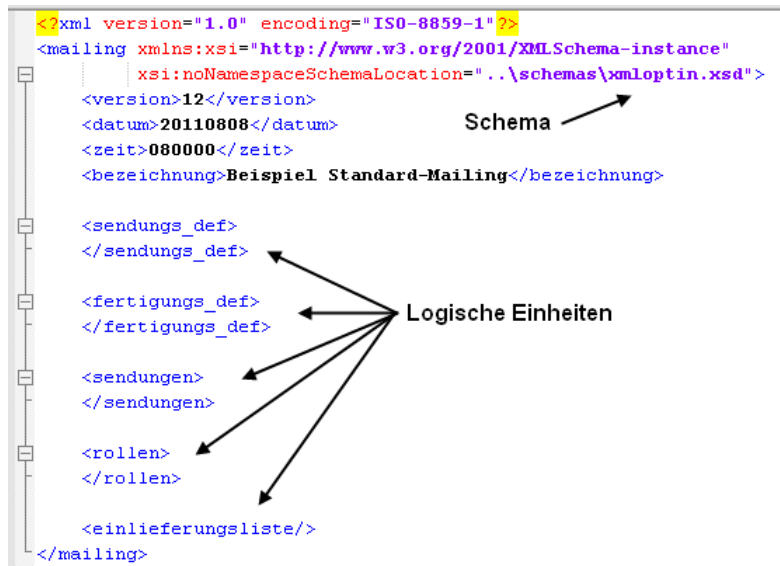
4.2.1 Struktur der Eingabedatei

Damit das Optimierungsmodul die Daten verarbeiten kann, müssen diese in einer XML-Struktur zur Verfügung gestellt werden. Den vollständigen Aufbau der Struktur können Sie der Dokumentation

im Verzeichnis `schemas\docs\xmloptin` für die XML-Portooptimierung bzw. `schemas\docs\prdin` entnehmen. Die Dokumentation liegt in beiden Fällen im HTML-Format vor, Ausgangspunkt ist jeweils die Datei `index.html`. Wenn Sie im Umgang mit XML-Dateien geübt sind, können Sie den Aufbau der `xsd`-Datei im jeweiligen `schemas`-Verzeichnis entnehmen.

Generell ist der Aufbau der Eingabedatei für beide Optimierungsmodule ähnlich. Unterschiede gibt es bei produktspezifischen Angaben. Bei Postvertriebsstücken müssen beispielsweise Angaben zu Beilagen gemacht werden, während dies bei Infopost nicht notwendig ist und deshalb der entsprechende Abschnitt auch nicht in der XML-Struktur für die XML-Portooptimierung enthalten ist.

Nachfolgend soll der prinzipielle Aufbau der Eingabedatei beschrieben werden. Dabei werden Grundkenntnisse im Bereich XML vorausgesetzt.



Jede Eingabedatei besitzt genau ein root-Element. Dieses lautet bei Eingabedateien für die XML-Portooptimierung `<mailing>` und für Eingabedateien beim Manager Presse Distribution XML `<presse>`.

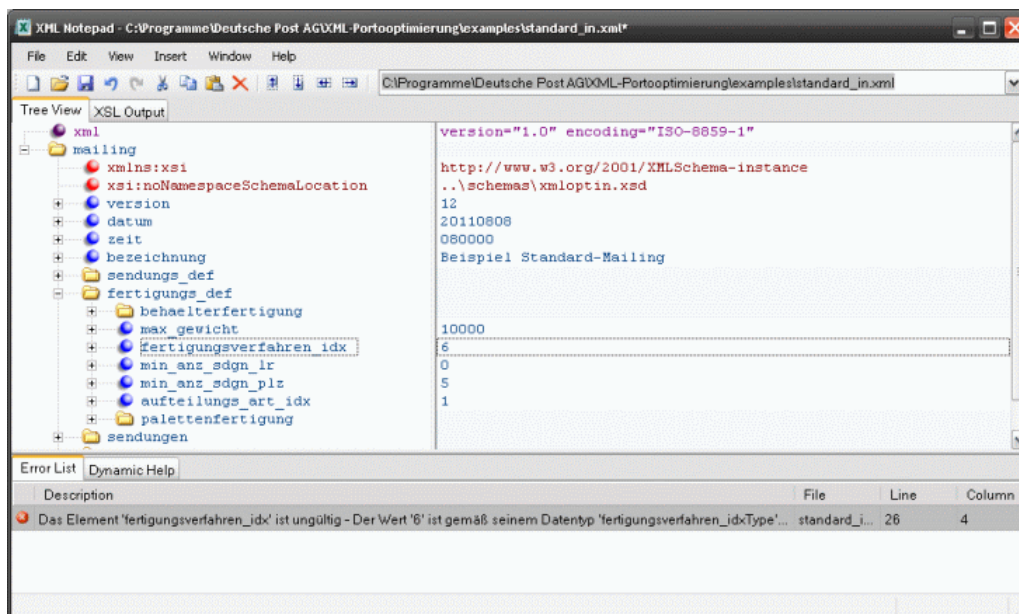
Innerhalb des root-Elements gibt es weitere Elemente, die jeweils einen logischen Block bilden. D.h. innerhalb eines solchen Blocks befinden sich Angaben, die logisch zusammen gehören. So sind beispielsweise im Element `<sendungs_def>` alle Angaben enthalten, die das Versandstück betreffen. Dazu gehören Daten wie das Sendungsgewicht und die Abmessungen der Sendung. Im Element `<fertigungs_def>` werden dagegen Vorgaben zur Fertigung gemacht, wie das maximale Paletten-gewicht oder ob eine Bund- oder Behälterfertigung erfolgen soll.

Damit die logischen Einheiten nicht zu unübersichtlich werden, sind diese selber wieder in logische Einheiten unterteilt. Innerhalb des Elements `<sendungs_def>` existiert beispielsweise das Element `<variante>`, welches die logische Einheit für die physikalischen Eigenschaften der Sendung bildet. Weitere Einheiten bilden die Elemente `<bundfertigung>` und `<behaelterfertigung>`. Diese vereinen Fertigungsparameter für die entsprechenden Fertigungsarten und stehen deshalb im Element `<fertigungs_def>`.


```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mailing xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noName
  <version>12</version>
  <datum>20110808</datum>
  <zeit>080000</zeit>
  <bezeichnung>Beispiel Standard-Mailing</bezeichnung>
  <sendungs_def>
    <inhalt_idx>1</inhalt_idx>
    <frankierung_idx>1</frankierung_idx>
    <variante>
      <id>0</id>
      <laenge>2200</laenge>
      <breite>1100</breite>
      <dicke>20</dicke>
      <gewicht>20</gewicht>
      <max_fuellmenge>150</max_fuellmenge>
      <bezeichnung>Standard</bezeichnung>
      <beh_anz_sdgn_halfvoll>80</beh_anz_sdgn_halfvoll>
    </variante>
  </sendungs_def>
  <fertigungs_def>
    <behaelterfertigung>
      <behaelterart_idx>1</behaelterart_idx>
    </behaelterfertigung>
    <max_gewicht>10000</max_gewicht>
    <fertigungsverfahren_idx>1</fertigungsverfahren_idx>
    <min_anz_sdgn_lr>0</min_anz_sdgn_lr>
    <min_anz_sdgn_plz>5</min_anz_sdgn_plz>
    <aufteilungs_art_idx>1</aufteilungs_art_idx>
    <palettenfertigung>
      <min_anz_sdgn_lz>0</min_anz_sdgn_lz>
      <min_anz_sdgn_lr>0</min_anz_sdgn_lr>
    </palettenfertigung>
  </fertigungs_def>
</mailing>
```

Durch die Verschachtelung und die Vielzahl der Elemente kann eine recht komplexe Eingabedatei entstehen. Um sicherzustellen, dass die Datei formal korrekt ist, besteht die Möglichkeit diese zu prüfen. Dazu werden mit dem Optimierungsmodul entsprechende Schemadefinitionen in Form von xsd-Dateien ausgeliefert. Diese finden Sie im Verzeichnis `schemas` der Installation.

Um nun eine Eingabedatei zu überprüfen, muss in ihr das Schema, gegen welches geprüft werden soll, angegeben sein. Dies erfolgt durch Angabe des Schemas im Attribut `xsi:noNamespaceSchemaLocation`. Die Validierung kann dann mit einem geeigneten XML-Tool (z.B. dem kostenlosen „XML Notepad“ von Microsoft) durchgeführt werden.



Sie können auch die Testanwendung zum Validieren verwenden. Setzen Sie dazu die Option *Eingabedatei validieren*. Wenn Sie zusätzlich die Option *Schema-Pfad anpassen* aktivieren, wird die Schema-Angabe in der Eingabedatei automatisch gesetzt (auf Seite 9).

Das Validieren der Eingabedatei ist besonders bei den ersten selbst erzeugten Eingabedateien ein

wichtiges Hilfsmittel, um Fehler zu finden.

Ein guter Ansatz, ein Verständnis für den Aufbau der Eingabedatei zu bekommen ist es auch, mit den Beispiel-Eingabedateien zu experimentieren. So könnten Sie zum Beispiel zunächst Maximalgewichte in den Fertigungsangaben ändern und sehen, wie sich dies auf die Ausgabedatei auswirkt. Oder Sie machen aus einer Behälterfertigung eine Bundfertigung.

4.2.2 Ausgewählte Elemente der Eingabedatei

In Folgenden soll auf einige besondere Elemente der Eingabedatei eingegangen werden:

Element `<version>`

Die Optimierungsmodule unterliegen einer ständigen Weiterentwicklung. Daraus ergeben sich auch Änderungen in der XML-Struktur. Über das Tag `<version>` teilen Sie dem Optimierungsmodul mit, welche Struktur-Version Sie in der Eingabedatei verwendet haben. Zu beachten ist dabei, dass nicht alle Versionen in Verbindung mit jeder AGB (Übergabetermin) verwendet werden können. Welche Kombinationen gültig sind, können Sie der Hilfedatei entnehmen.

Element `<bezeichnung>`

Über dieses Tag kann ein Name für die Aussendung bestimmt werden. Der Name wird u.a. auf den Einlieferungslisten im Feld „Bezeichnung der Aussendung“ verwendet.

Element `<sendungs_def>`

In diesem Element werden im wesentlichen die Angaben zu den Eigenschaften der Sendung gemacht. Detaillierte Informationen finden Sie in der Dokumentation oder in der Schema-Datei.

Element `<fertigungs_def>`

In diesem Element werden im wesentlichen die Fertigungsparameter an das Optimierungsmodul übergeben. Detaillierte Informationen finden Sie in der Dokumentation oder in der Schema-Datei.

Element `<sendungen>`

In diesem Element werden die anzuschreibenden Adressen an die Optimierung übergeben. Dazu kann das Element beliebig viele Unterelemente vom Typ `<sdg>` besitzen.

Für jede anzuschreibende Adresse muss ein `<sdg>`-Element erzeugt werden. Jedes dieser `<sdg>`-Elemente muss seinerseits eine eindeutige numerische ID erhalten, über welche sich die Adresse in der Ausgangsmenge referenzieren lässt. Darüber hinaus muss jedes `<sdg>`-Element noch eine Angabe zur Zustell-Postleitzahl im Tag `<plz>` besitzen.

Die eindeutige ID mit Bezug auf die Zustelladresse ist notwendig, weil dem Optimierungsmodul nicht die gesamte Adresse, sondern lediglich die Postleitzahl übergeben wird. Soll am Ende des Optimierungsvorgangs das Versandstück mit der Adresse beschriftet werden, so muss die konkrete Adresse anhand der ID aus der Ausgangsdatei ermittelt werden (auf Seite 20).

Element `<rollen>`

In diesem Element werden Angaben über die am Projekt beteiligten Rollen, also Absender, Einlieferer und Zahlungspflichtiger an das Optimierungsmodul übergeben. Detaillierte Informationen finden Sie in der Dokumentation oder in der Schema-Datei.

Element `<einlieferungslisten>`

Wenn Sie wünschen, dass das Optimierungsmodul eine PDF-Datei mit den Einlieferungsdokumenten generiert, müssen Sie dieses Tag angeben. Anderenfalls wird nur die Ausgabe-XML-Datei erzeugt.

Element `<ausgabe>` (Manager Presse Distribution XML)

In diesem Element werden Angaben zur Ausgabe (z.B. ZKZ, Art der Distribution, Zeitungsnummer) gemacht. Detaillierte Informationen finden Sie in der Dokumentation oder in der Schema-Datei.

Element <beilage> (Manager Presse Distribution XML)

In diesem Element werden Angaben zu den Beilagen der Sendungen gemacht. Dazu gehören Daten wie die Art und Gewicht der Beilage. Detaillierte Informationen finden Sie in der Dokumentation oder in der Schema-Datei.

Element <beilagenkombination> (Manager Presse Distribution XML)

Verschiedene Beilagen können zu Beilagenkombinationen zusammengefasst werden. Auf diese kann dann bei den einzelnen Sendungen (<sdg>) verwiesen werden, so dass bei der Sendung nicht jede Beilage einzeln aufgeführt werden muss. Detaillierte Informationen finden Sie in der Dokumentation oder in der Schema-Datei.

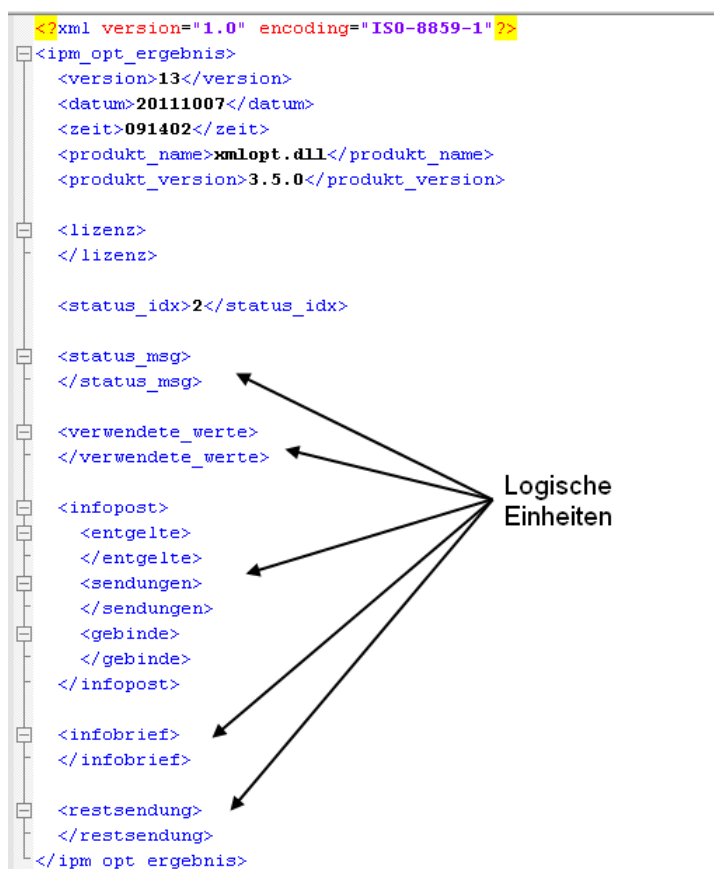
4.3 Ausgabedatei

Das Optimierungsmodul erzeugt eine Ausgabedatei mit den Ergebnissen. Auch diese Datei besitzt eine XML-Struktur. Den genauen Aufbau inkl. aller Elemente finden Sie in der HTML-Dokumentation im Verzeichnis `schemas\doc\xmlout` bzw. `schemas\doc\xmlprout` (beginnen Sie mit der Datei `index.html`). Wenn Sie im Umgang mit XML-Dateien geübt sind, können Sie den Aufbau der xsd-Datei im jeweiligen `schemas`-Verzeichnis entnehmen.

Die Ausgabedatei ist eine Art „Steuerungsdatei“ und muss vom Geschäftsprozess weiterverarbeitet werden. Anhand der Ergebnisse in dieser Datei müssen z.B. die Versandstücke beschriftet und Behälter- oder Palettenleitzettel gedruckt werden.

4.3.1 Struktur der Ausgabedatei

Wie die Eingabedatei verfügt auch die Ausgabedatei über ein root-Element und logische Einheiten:



Wichtigstes Element in der Ausgabedatei ist zunächst das Tag <status_idx>. Dieses informiert darüber, ob die Optimierung erfolgreich durchgeführt werden konnte oder ob Hinweise, Warnungen oder Fehler aufgetreten sind. Sollte dies der Fall sein, enthält das Element <status_msg> eine entsprechende Beschreibung des Vorkommnisses. Das <status_msg>-Element kann mehrfach vorkommen.

Das Element <verwendete_werte> spiegelt im Prinzip die Eingabedatei wieder. Dabei werden allerdings auch die Elemente ausgegeben, die in der Eingabedatei nicht enthalten waren, für die aber

Standardwerte definiert sind. Es kann auch vorkommen, dass das Optimierungsmodul Werte aus der Eingabedatei anpasst, weil sie z.B. nicht den AGBs entsprechen. In solchen Fällen wird unter `<verwendete_werte>` der korrigierte Wert ausgegeben. Zusätzlich wird in diesen Fällen ein Element `<status_msg>` mit entsprechendem Hinweis (Codenummer 1xxx) ausgegeben.

An den Abschnitt `<verwendete_werte>` schließen sich die eigentlichen Optimierungsergebnisse an. Das Optimierungsmodul erzeugt für jedes durch den Optimierungsvorgang erzeugte Postprodukt, also z.B. Infopost, Infobrief oder Streifbandzeitungen ein separates Element.

In der nachfolgenden Abbildung wird zur weiteren Erläuterung als Beispiel ein `<infopost>`-Element in verkürzter Form dargestellt:

```
<infopost>
  <format_idx>1</format_idx>
  <entgelte>
    <anz_sdg>4579</anz_sdg>
    <anz_aufz>0</anz_aufz>
    <anz_sdg_gesamt>4579</anz_sdg_gesamt>
    <entgelt_sdg>25</entgelt_sdg>
    <entgelt_sdg_gesamt>114475</entgelt_sdg_gesamt>
    <entgelt_erm_geb_lr>2127</entgelt_erm_geb_lr>
    <gewicht_sdg>20</gewicht_sdg>
    <gewicht_gesamt>91500</gewicht_gesamt>
  </entgelte>
  <sendungen>
    <sdg><id>0</id><oid>3139</oid><bid>0</bid><banf>1</banf></sdg>
    <sdg><id>1</id><oid>2866</oid><bid>0</bid></sdg>
    <sdg><id>2</id><oid>2102</oid><bid>0</bid></sdg>
    <sdg><id>3</id><oid>1404</oid><bid>0</bid></sdg>
    ...
    <sdg><id>149</id><oid>2745</oid><bid>0</bid><bend>1</bend></sdg>
  </sendungen>
  <gebinde>
    <b><id>0</id><anz_sdg>150</anz_sdg><gev_n>3000</gev_n><art_idx>3</art_idx><kennung>0-9</kennung></b>
    <b><id>1</id><anz_sdg>87</anz_sdg><gev_n>1740</gev_n><art_idx>3</art_idx><kennung>0-9</kennung></b>
    <b><id>2</id><anz_sdg>88</anz_sdg><gev_n>1760</gev_n><art_idx>3</art_idx><kennung>0-9</kennung></b>
    <b><id>3</id><anz_sdg>150</anz_sdg><gev_n>3000</gev_n><art_idx>2</art_idx><kennung>40</kennung><rabattiert>1</b>
    ...
  </gebinde>
</infopost>
```

Im Element `<entgelte>` werden die vom Optimierungsmodul ermittelten Entgelte für die Aussendung ausgegeben. Dabei werden sowohl Einzelentgelte als auch Aufsummierungen und Rabatte ausgegeben. Sie können diese Ergebnisse nutzen, um die Einlieferungslisten auszufüllen, falls Sie diese nicht vom Optimierungsmodul erstellen lassen.

Das Element `<sendungen>` enthält für jede Zustelladresse aus der Eingabedatei, für die das Optimierungsmodul als günstigste Versandart „Infopost“ ermittelt hat, ein Element vom Typ `<sdg>`. Jedes Sendungselement besitzt eine eindeutige ID im Tag `<id>`. Über das Tag `<oid>` wird die Originalsendung aus der Eingabedatei referenziert. Über das Tag `<bid>` wird das Gebinde (Bund oder Behälter, vgl. Element `<gebinde>`) referenziert, in welches die Sendung bei der Fertigung gepackt werden muss.

Das Element `<gebinde>` enthält eine Liste mit allen Gebinden, die als Infopost gefertigt werden. Dabei kann es sich um Bunde oder Behälter handeln, je nachdem welcher Typ in der Eingabedatei definiert wurde. Das Tag `<kennung>` enthält die Beschriftung für das Gebinde, also den variablen Teil der Bund- oder Behälterzettel.

Der Aufbau der Ergebnis-Elemente ähnelt sich für alle Postprodukte. Es kann jedoch sein, dass nicht alle Elemente vorhanden sind. So werden bei Paketsendungen beispielsweise keine Entgelte ermittelt. Streifbandzeitungen werden einzeln eingeliefert, so dass es kein Element `<gebinde>` gibt.

Die genaue Beschreibung aller Elemente entnehmen Sie bitte der Dokumentation.

4.3.2 Ausgewählte Elemente der Ausgabedatei

In Folgenden soll auf einige besondere Elemente der Eingabedatei eingegangen werden:

Element `<version>`

Die Optimierungsmodule unterliegen einer ständigen Weiterentwicklung. Daraus ergeben sich auch Änderungen in der Struktur der Ausgabedatei. Das Element `<version>` gibt an, welche Version der Ausgabedatei erzeugt wurde. In der Regel wird dies die aktuellste sein. Unter Umständen ist es aber erforderlich, das Optimierungsmodul dazu zu veranlassen, eine andere Version zu erzeugen. Dies kann durch Einfügen des Tags `<output_xml_version>` im Konfig-String bewirkt werden. Bitte beachten Sie, dass nicht jede Ausgabeversion bei jeder AGB-Version möglich ist.

Element <status_idx>

Das Element informiert über das Ergebnis des Optimierungsprozesses. Je nach dem ob dieser erfolgreich verlaufen ist oder ob Informationen, Warnungen oder gar Fehler erzeugt wurden, wird der Status entsprechend gesetzt.

status_idx	Bedeutung
1	Die Optimierung wurde ohne Probleme bzw. Unregelmäßigkeiten durchgeführt.
2	Die Optimierung wurde durchgeführt – es sind jedoch Warnhinweise (siehe Element <status_msg>) vorhanden.
3	Die Optimierung wurde abgebrochen – Gründe hierfür sind nicht zulässige Parameter bzw. Parameterkombinationen in der Eingabedatei.
4	Die Optimierung wurde abgebrochen – Gründe hierfür sind interne Fehler, Systemfehler oder andere nicht näher spezifizierbare Probleme.

Element <status_msg>

Wenn während des Optimierungsvorgangs Hinweise, Warnungen oder Fehler aufgetreten sind, werden diese im Klartext in diesem Element ausgegeben. Das Element kann mehrfach auftreten.

Zur internen Klassifizierung ist jeder Meldung eine Codenummer zugeordnet. Hinweise mit den Codenummern 1xxx geben z.B. statistische Informationen oder Optimierungsmöglichkeiten wieder. Warnungen mit den Codenummern 2xxx deuten meist auf eine ungünstige oder unübliche Parametereinstellung in der Eingabedatei hin. Bei fehlenden oder nicht zulässigen Parametereinstellungen werden Fehlermeldungen mit dem Code 3xxx generiert.

Beispiel:

```
<status_idx>3</status_idx>
<status_msg>
  <status_idx>3</status_idx>
  <code>3334</code>
  <msg>Das Versandnetzwerk ist nicht angegeben.</msg>
</status_msg>
```

Die Optimierung wurde abgebrochen, weil das Versandnetzwerk in der Eingabedatei nicht angegeben wurde (Element <versandservice_idx>).

Element <verwendete_werte>

Wie beim Punkt *Struktur der Ausgabedatei* (auf Seite 19) bereits beschrieben, enthält dieses Element die Angaben aus der Eingabedatei ergänzt um Standardwerte und - falls notwendig - korrigierte Werte. Detaillierte Informationen finden Sie in der Dokumentation oder in der Schema-Datei.

Element <sendungen>

Wie Sie bereits aus der Beschreibung der Eingabedatei wissen (auf der vorherigen Seite) wurde der Optimierung keine vollständige Adresse, sondern lediglich die Zustell-PLZ der Adresse zusammen mit einer ID übergeben. Mit dieser Zustell-PLZ hat das Optimierungsmodul die Optimierung durchgeführt und unter anderem die Fertigungsreihenfolge der einzelnen Sendungen ermittelt. Die Sendungen müssen in der Reihenfolge gefertigt werden (innerhalb eines Produktes), wie Sie im Element <sendungen> vorkommen. Das Tag <id> des <sdg>-Elements verdeutlicht diese noch einmal: Es nummeriert die Sendungen in der richtigen Fertigungsreihenfolge aufsteigend durch. Der Bezug zur Originaladresse erfolgt über das Tag <oid>. Dieses beinhaltet die ID der Sendung aus der Eingabedatei. Nachfolgende Abbildung verdeutlicht den Zusammenhang zwischen der Adressenliste, der Ein- und der Ausgabedatei.

Adressliste

1	Behringer Spezielle Studiotechnik GmbH	Otto-Brenner-Str. 4	47877	Willich	Herr		
2	Colt International GmbH	Am Dämmig 16/18	37339	Teistungen	Herr	Gottfried	Wül
3	Nau Industrie-Packmittel	Hermann Nau	Salder Str. 35	31226	Peine	Herr	Ge
4	Kaiser, Nachfolger GmbH & Co. KG	Metallgießerei, Maschinenbau	Rathausstr. 12	20095	Hau		
5	bela-pharm GmbH & Co. KG	Arzneimittelfabrik	Hohenlohestr. 40	28209	Bremen	Herr	
6	...						
7							
8							

Eingabedatei

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mailing xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  <version>12</version>
  <datum>20110808</datum>
  <zeit>080000</zeit>
  ...
  <sendungen>
    <sdg><id>1</id><plz>47877</plz> </sdg>
    <sdg><id>2</id><plz>37339</plz> </sdg>
    <sdg><id>3</id><plz>31226</plz> </sdg>
    <sdg><id>4</id><plz>20095</plz> </sdg>
    <sdg><id>5</id><plz>28209</plz> </sdg>
    ...
  </sendungen>
  ...
</mailing>
```

Ausgabedatei

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ipm_opt_ergebnis xmlns:xsi="http://www.w3.org/2001/XMLSchema-insta
  <version>13</version>
  <datum>20111010</datum>
  <zeit>133158</zeit>
  ...
  <infopost>
    <format_idx>1</format_idx>
    ...
  <sendungen>
    <sdg><id>0</id><oid>4</oid><bid>0</bid><banf>1</banf></sdg>
    <sdg><id>1</id><oid>5</oid><bid>0</bid></sdg>
    <sdg><id>2</id><oid>3</oid><bid>0</bid></sdg>
    <sdg><id>3</id><oid>2</oid><bid>0</bid></sdg>
    <sdg><id>4</id><oid>1</oid><bid>0</bid></sdg>
    ...
  </sendungen>
  ...
</ipm_opt_ergebnis>
```

Adressliste

1	Behringer Spezielle Studiotechnik GmbH	Otto-Brenner-Str. 4	47877	Willich	Herr		
2	Colt International GmbH	Am Dämmig 16/18	37339	Teistungen	Herr	Gottfried	Wül
3	Nau Industrie-Packmittel	Hermann Nau	Salder Str. 35	31226	Peine	Herr	Ge
4	Kaiser, Nachfolger GmbH & Co. KG	Metallgießerei, Maschinenbau	Rathausstr. 12	20095	Hau		
5	bela-pharm GmbH & Co. KG	Arzneimittelfabrik	Hohenlohestr. 40	28209	Bremen	Herr	
6	...						
7							
8							

Sendungsbeschriftung

Kaiser, Nachfolger GmbH & Co. KG
 Metallgießerei, Maschinenbau
 Rathausstr. 12
 20095 Hamburg

bela-pharm GmbH & Co. Kg
 Arzneimittelfabrik
 Hohenlohestr. 40
 28209 Bremen

Nau Industrie-Packmittel
 Hermann Nau
 Salder Str. 35
 31226 Peine

Colt International GmbH
 Am Dämmig 16/18
 37339 Teistungen

Behringer Spezielle Studio GmbH
 Otto-Brenner-Str. 4
 47877 Willich

...

In Schritt 1 generiert der Geschäftsprozess aus der Adressliste die Eingabedatei. Als eindeutige ID für jede Sendung kann dabei beispielsweise die Datensatz-Nummer verwendet werden. Diese wird im Element `<sdg>` im Tag `<id>` eingetragen.

Das Optimierungsmodul erzeugt in Schritt 2 die Ausgabedatei. In dieser sind die `<sdg>`-Elemente in der Reihenfolge enthalten, in der sie gefertigt werden müssen (Tag `<id>`). Zusätzlich enthält jedes Element `<sdg>` das Tag `<oid>`. Dieses verweist auf das `<sdg>`-Element in der Eingabedatei und damit auch auf die Adresse in der Adressliste.

Sollen nun im Geschäftsprozess die Sendungen mit den Adressen beschriftet werden, so muss folgendermaßen vorgegangen werden:

- Aus der Ausgabedatei wird die `<oid>` des ersten `<sdg>`-Elements ausgelesen, im Beispiel „4“.
- Der Wert der `<oid>` wird in den Verweis auf die Originaladresse zurückgewandelt. Im Beispiel entspricht die `<oid>` der Datensatznummer. Die erste Sendung muss also mit der Adresse des Datensatzes 4 beschriftet werden.
- Der Geschäftsprozess liest die Adresse Nr. 4 aus der Adressliste (Schritt 3). Im Beispiel „Kaiser, Nachfolger GmbH & Co. KG...“ und druckt diese auf die Sendung (Schritt 4).
- Anschließend wird die `<oid>` des nächsten `<sdg>`-Elements aus der Adressdatei gelesen („5“) und die Punkte 2 und 3 wiederholt.

Der Punkt 4 wird für alle weiteren `<sdg>`-Elemente wiederholt. Am Ende sind alle Sendungen in der korrekten Reihenfolge beschriftet.

Das Tag `<bid>` ist ein Verweis auf das Gebinde (Bund oder Behälter), in welches die Sendung gepackt werden muss (siehe Element `<gebinde>`).

Element `<gebinde>`

Alle zu fertigenden Gebinde (Bunde oder Behälter) sind in dieser Auflistung enthalten. Für jedes Gebinde gibt es genau ein Element `` mit einer eindeutigen ID (`<id>`) und weiteren Angaben wie Anzahl der Sendungen im Gebinde, das Gewicht sowie Art und Ziel des Gebindes. Bei aktivierter Palettenfertigung existiert darüber hinaus mit dem Tag `<pid>` ein Verweis auf die Palette, auf die das Gebinde gepackt werden muss (siehe auch Element `<paletten>`). Diese Auflistung wird u.a. zur Erstellung der Leitzettel für die Bunde oder Behälter benötigt.

Element `<paletten>` (bei aktivierter Palettenfertigung)

Dieses Element enthält eine Auflistung aller Paletten eines Produkts. Für jede Palette existiert ein `<pal>`-Element mit einer eindeutigen ID (`<id>`), Angaben zur Anzahl der Sendungen und der Gebinde, die sich auf dieser Palette befinden sowie Angaben zur Art, zum Gewicht und zum Ziel der Palette. Diese Angaben werden für die Einlieferungsunterlagen und für die Erstellung der Palettenleitzettel benötigt.

5 Anhang

Inhalt

5.1	Versionsübersicht	24
5.2	Funktionsüberblick XML-Portoptimierung	24
5.3	Funktionsüberblick Manager Presse Distribution XML	25

5.1 Versionsübersicht

Dieses Integrationshandbuch wurde möglichst allgemein gehalten und versionsunabhängig geschrieben. Im Einzelfall können die hier beschriebenen XML-Tags jedoch von der aktuellen Version abweichen.

Folgende Programmversionen liegen dieser Anleitung zu Grunde:

Anwendung	Version	Datum
XML-Portoptimierung	3.6	25.11.2011
Manager Presse Distribution	2.1	11.11.2011

5.2 Funktionsüberblick XML-Portoptimierung

Die XML-Portoptimierung verteilt die Sendungen aus dem Bereich „Brief“ in der portoptimierten Reihenfolge auf verschiedenen Behälter bzw. Bunde und diese ggf. auf die notwendigen Paletten.

Dazu liefert die XML-Portoptimierung alle zur Auflieferung relevanten Dokumente wie z.B. die Einlieferungslisten im PDF-Format sowie die Fertigungs- und Packreihenfolge der Sendungen in einer Ausgabedatei im XML-Format.

Funktionen im Einzelnen:

- Einfache Integration in den Arbeitsprozess
- Ansteuerung mit nahezu jeder Programmiersprache die DLLs aufrufen kann
- Berücksichtigung der aktuellen Postbestimmungen
- Minimierung der Portokosten
- Unterstützung von Teileinlieferungen
- Unterstützung von Katalogeinzelsendungen
- Unterstützung von VarioPlus-Mailings und Werbesendungen
- Unterstützung von Infopost/Infobrief und VarioPlus international
- Unterstützung von Briefen zum Kilotarif
- Unterstützung von Premiumadressen
- Fertigung über Parameter umfangreich konfigurierbar
- Fertigung im Einstrom- und Mehrstromverfahren (auch variantenrein)
- Ankündigung der Einlieferung via AM.exchange (Auftragsmanagement)
- Überprüfung von Postleitzahlen auf Existenz

- Erstellung aller zur Einlieferung benötigten Unterlagen (PDF)
- Alle Informationen zur Erstellung der Aufschriftzettel
- Alle Informationen zur Erstellung der Fertigungsunterlagen (z.B. Bund- und Palettenlisten)
- Ausführliche Dokumentation der Schnittstelle (XSD und HTML)
- Beispiel-XML-Dateien (Input/Output)
- Testumgebung

5.3 Funktionsüberblick Manager Presse Distribution XML

Der Manager Presse Distribution XML verteilt die Sendungen aus dem Bereich „Presse“ anhand der Datei ZEBU auf verschiedene Bunde bzw. Behälter und diese ggf. auf die notwendigen Paletten.

Dazu liefert der Manager Presse Distribution XML alle zur Auflieferung relevanten Dokumente wie z.B. die Einlieferungslisten im PDF-Format sowie die Fertigungs- und Packreihenfolge der Sendungen in einer Ausgabedatei im XML-Format.

Funktionen im Einzelnen:

- Einfache Integration in den Arbeitsprozess
- Ansteuerung mit nahezu jeder Programmiersprache die DLLs aufrufen kann
- Berücksichtigung der aktuellen Postbestimmungen
- Verteilung anhand Datei ZEBU
- Unterstützung von Postvertriebsstücken, Pressesendungen, Streifbandzeitungen
- Unterstützung von Presse und Buch International zum Kilotarif
- Unterstützung von alternativen Versandformen (Pakete, Pakete International)
- Unterstützung von Beilagen
- Unterstützung von Mehrfachbeziehern
- Unterstützung von Premiumadress
- Fertigung über Parameter umfangreich konfigurierbar
- Fertigung im Einstrom- und Mehrstromverfahren (auch beilagenkombinationsrein)
- Erzeugung des Einlieferungsdatensatzes (EDS)
- Ankündigung der Einlieferung via AM.exchange (Auftragsmanagement)
- Erstellung aller zur Einlieferung benötigten Unterlagen (PDF)
- Alle Informationen zur Erstellung der Aufschriftzettel
- Alle Informationen zur Erstellung der Fertigungsunterlagen (z.B. Bund- und Palettenlisten)
- Ausführliche Dokumentation der Schnittstelle (XSD und HTML)
- Beispiel-XML-Dateien (Input/Output)
- Testumgebung

6 Index

Alternative Produkte 4
Anhang 24
Ansteuerung der Optimierungs-DLL 13
Architekturüberblick 8
Ausgabedatei 19

Eingabedatei 15
Einleitung 4
Einsatzzweck 4

Funktionsüberblick Manager Presse Distribution
XML 25
Funktionsüberblick XML-Portoptimierung 24

Grundlagen 8

Installation 5
Installation / Konfiguration / Updates 5

Konfiguration des Optimierungsaufrufs 11

Lizenzierung 6

Manager Presse Distribution XML 4

Prozessintegration 13

Systemanforderungen 5

Testanwendung 9

Updates 7

Versionsübersicht 24
Verzeichnisstruktur 6

XML-Portoptimierung 4

Zielgruppe dieses Handbuchs 4

Stand: 01/2012
Mat.-Nr. 675-601-036