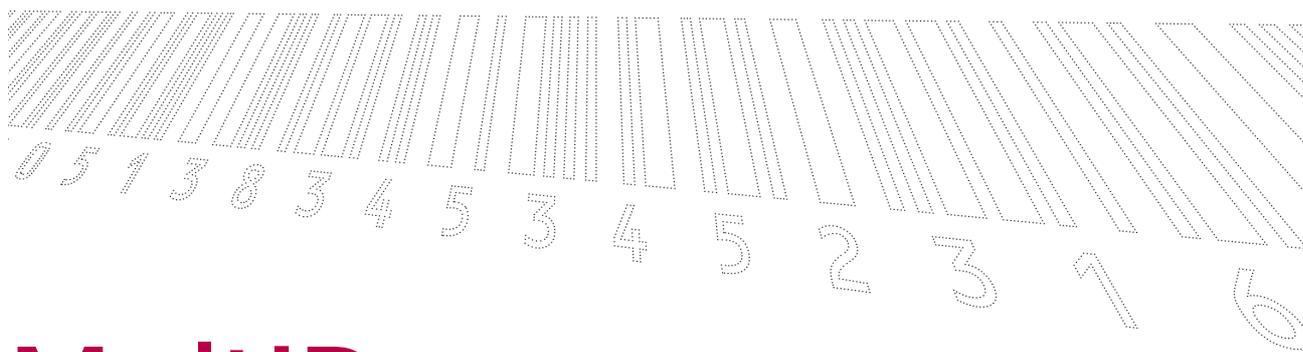


Funktionsübersicht



MultiParser

Für Automobilhersteller und Zulieferer

Inhaltsverzeichnis

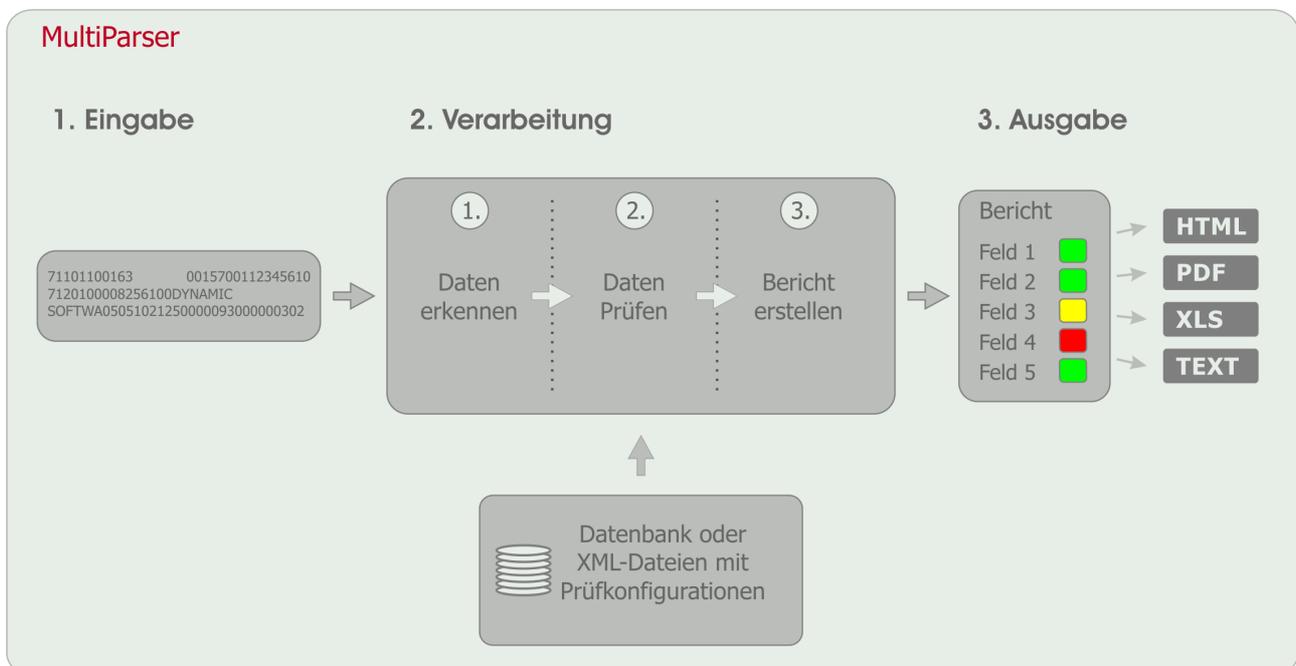
1 Funktionsübersicht.....	3
1.1 Eingabe.....	3
1.2 Verarbeitung.....	4
1.2.1 Daten erkennen und erfassen.....	4
1.2.2 Daten prüfen.....	4
1.2.3 Bericht erstellen.....	5
1.3 Ausgabe.....	5
2 PrüfregeIn.....	6
2.1 Standard Prüfungen.....	6
2.2 Addons.....	8
2.2.1 Normales Addon.....	8
2.2.2 Backend-Addon.....	8
3 Monitoring.....	9

1 Funktionsübersicht

Der MultiParser ist ein Werkzeug zur Analyse von Text-Daten in diversen Standardformaten (VDA, ODETTE, EDIFACT, iDOC), die im elektronischen Datenaustausch (EDI) verwendet werden.

Der MultiParser ist ein generisches Tool für die Textanalyse. Alle Textprüfungen werden in einer Konfigurationsdatenbank oder als XML-Datei hinterlegt. Das hat den entscheidenden Vorteil, dass zur Laufzeit - ohne Programmieraufwand - Anpassung an den Prüfroutinen durchgeführt werden können. Es können beliebig viele Textkonfigurationen hinterlegt werden (z.B. Lieferschein, Lieferabruf, Gutschriften etc.).

Schematischer Ablauf:



1.1 Eingabe

Der zu analysierende Text muss als Klartext (ASCII) vorliegen. Der Text kann an den MultiParser direkt über eine Methode als Parameter übergeben werden oder aus einer Datei eingelesen werden.

Der MultiParser kann folgende verschiedene Textarten verarbeiten:

Satzbasierte Texte, deren Inhalt durch einen Satzschlüssel definiert werden und entweder

- mit variabler Zeichenlänge durch Trennzeichen
- oder mit fester Zeichenlänge pro Zeile (geblockter Text) getrennt sind

1.2 Verarbeitung

Die Vorbedingung für die Textverarbeitung ist eine in der Datenbank oder in einem XML-File hinterlegte Konfiguration. Mit dem Aufruf zur Durchführung der Analyse muss immer eine aktuelle Konfigurationen übergeben werden (siehe auch Konfigurieren).

Das Grundprinzip der Textverarbeitung basiert auf 3 Schritten:

1. Daten erkennen und erfassen
2. Daten prüfen
3. Bericht erstellen

1.2.1 Daten erkennen und erfassen

In diesem ersten Arbeitsschritt wird der zu analysierende Text eingelesen und in eine Liste von Textzeilen zerlegt. Beim Einlesen wird das Zeilenende automatisch erkannt - unabhängig vom Betriebssystem (Windows oder Unix/Linux). Überflüssige Steuerzeichen wie z.B. CarriageReturn werden vom System entfernt. Der Anwender hat aber auch die Möglichkeit, in der Konfiguration ein eigenes Trennzeichen zu definieren (z.B. "|").

Anhand der Konfiguration werden die Textzeilen in einzelne Felder zerlegt. Zum Beispiel ist in der Konfiguration eine Satzart 'XYZ' hinterlegt. Das Feld Kundennummer steht im Satz 'XYZ' immer von der 5. bis zur 10. Stelle. Das Programm liest den Text von der 5. bis zur 10. Stelle und ordnet diesen Wert dem Feld Kundennummer zu. Im nächsten Arbeitsschritt wird dieser Wert auf Gültigkeit geprüft.

1.2.2 Daten prüfen

Der zweite Arbeitsschritt stellt das Herzstück des MultiParsers dar. Hier werden die Prüfungen der Inhalte durchgeführt. Basis für diese Prüfung ist die hinterlegte Konfiguration in der Datenbank oder einer XML-Datei (siehe Kap. 2.2). Satz für Satz und Feld für Feld werden die Prüfungen sequentiell bearbeitet. Jedes Feld hat eine ganz individuelle Liste von Prüfungen. Sobald eine Bedingung nicht erfüllt wird, wird dieses Feld im Ergebnisbericht als fehlerhaft gekennzeichnet. Der Fehler kann als Warnung oder als Error deklariert werden. Die Kategorie Error überschreibt immer Warnung.

1.2.3 Bericht erstellen

Alle Prüfungsergebnisse werden in einem systeminternen Bericht verwaltet. Dieser Bericht wird nur im Programmspeicher aufgebaut und hat somit kein bekanntes Dokumentenformat. Das hat den Vorteil, dass die Ausgabe (Rendering) des Berichts auf kein bestimmtes Format festgelegt bzw. reduziert wird (siehe Kap. 4. Ausgabe). Der systeminterne Bericht ist nicht persistent. Mit jedem neuen Prüfungslauf wird dieser wieder zurückgesetzt und die Ressourcen wieder freigegeben.

1.3 Ausgabe

In dem systeminternen Bericht werden alle Prüfungsergebnisse zu den jeweiligen Feldern abgelegt. Die Visualisierung des Berichts kann je nach Anforderung und Kundenumgebung implementiert werden. Eine einfach formatierte Textausgabe ist bereits enthalten. Ohne großen Aufwand kann der Bericht auch im CSV- oder HTML-Format ausgegeben werden. Die HTML-Generierung kann on-the-fly in einem Servlet oder in einer JSP-Seite durchgeführt werden.

Durch den Einsatz von externen Java-Bibliotheken kann der systeminternen Bericht problemlos zusätzlich auch in folgende Formate konvertiert werden:

- PDF
- XLS
- HTML
- TEXT

2 Prüfregelein

2.1 Standard Prüfungen

Name	Inhalt der Feldprüfung
Min	<p>Legt den minimalen Wert eines Feldes fest.</p> <p>Beispiel: Die Abladestelle muss größer oder gleich 100 sein.</p>
Max	<p>Legt den maximalen Wert eines Feldes fest.</p> <p>Beispiel: Bruttogewicht muss kleiner oder gleich 1.000 sein.</p>
Min. Stellen nach dem Komma	<p>Legt die Mindestanzahl der Stellen nach dem Komma fest.</p> <p>Beispiel: Preis muss 2 Stellen nach dem Komma haben.</p>
Max. Stellen nach dem Komma	<p>Legt die maximale Anzahl der Stellen nach dem Komma fest.</p> <p>Beispiel: Preis darf nicht mehr als 2 Stellen nach dem Komma haben.</p>
Min. Stellen vor dem Komma	<p>Legt die Mindestanzahl der Stellen vor dem Komma fest.</p> <p>Beispiel: Länge muss 4 Stellen vor dem Komma haben.</p>
Max. Stellen vor dem Komma	<p>Legt die maximale Anzahl der Stellen vor dem Komma fest.</p> <p>Beispiel: Länge darf nicht mehr als 4 Stellen vor dem Komma haben.</p>
Min Länge	<p>Legt die minimale Anzahl Zeichen des Feldes fest.</p> <p>Hinweis: Bei Nachrichtentypen mit fester Textlänge werden Leerzeichen (Blanks) nicht mitgezählt.</p>
Max Länge	<p>Legt die maximale Anzahl der Zeichen des Feldes fest.</p> <p>Hinweis: Bei Nachrichtentypen mit fester Textlänge werden Leerzeichen (Blanks) nicht mitgezählt.</p>
Datumsformat	<p>Definiert das Datumsformat für dieses Feld. Folgende reservierte Zeichen werden verwendet (case-sensitiv):</p> <ul style="list-style-type: none"> - YY Zweistelliges Jahr, z.B. 08 - YYYY Vierstelliges Jahr, z.B. 2008 - MM Zweistelliger Monat, z.B. 09 - DD Zweistelliger Tag, z.B. 27 - hh Zweistellige Stunde, z.B. 23 - mm Zweistellige Minute, z.B. 56 - ss Zweistellige Sekunde, z.B. 37 <p>Neben den Datumszeichen können beliebige weitere Zeichen (., - #) zur Prüfung verwendet werden.</p> <p>Beispiele: DD.MM.YYYY YYYY-MM-DD YYYYMMDD</p>

Name	Inhalt der Feldprüfung
	<p>Es können auch mehrere Datumsformatsvarianten angegeben werden. Diese können einfach mit dem Pipe (' ')-Zeichen aneinander gehängt werden.</p> <p>Beispiel: YYYYMMDDhhmm YYMMDD</p>
Regular Expression	<p>Hier können beliebige reguläre Ausdrücke zur Prüfung definiert werden. Aufgrund der Komplexität von regulären Ausdrücken wird hier auf die verwendete freie Bibliothek (jakarta-regexp) verwiesen: http://jakarta.apache.org/regexp/</p>
Nummer mit führenden Nullen	<p>Legt fest, ob der Wert mit führenden Nullen aufgefüllt werden muss (linksbündig).</p>
Feldtyp	<p>Freitextfeld um den Feldtyp zu beschreiben, derzeit ohne weitere Prüffunktion</p>
Erforderlich	<p>Definiert das Feld als Mussfeld</p>
Linksbündig	<p>Legt fest, dass der Wert des Feldes linksbündig gesetzt sein muss. Wird ein Leerzeichen als erstes linkes Zeichen gefunden, wird dieses Feld als fehlerhaft gekennzeichnet.</p>
Rechtsbündig	<p>Legt fest, dass der Wert des Feldes rechtsbündig gesetzt sein muss. Wird ein Leerzeichen als letztes rechtes Zeichen gefunden, wird dieses Feld als fehlerhaft gekennzeichnet.</p>
Verbotene Zeichen	<p>Hier können Zeichen definiert werden, die in diesem Feld nicht vorkommen dürfen. Die Angabe muss ohne Trennzeichen und Leerzeichen als ein Text (case-sensitiv) gemacht werden.</p> <p>Beispiel für keine Umlaute: äÄüÜöÖ</p>
Explizit erlaubte Zeichen	<p>Hier werden explizit erlaubte Zeichen definiert. Die Angabe erfolgt ohne Trennzeichen und Leerzeichen als ein Text (case-sensitiv).</p> <p>Beispiel für Mailadresse: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ_-.@.1234567890</p>
Verbotene Werte	<p>Gegenteil von 'Explizit erlaubte Werte' - Beschreibung nächster Eintrag</p>
Explizit erlaubte Werte	<p>Definiert eine Liste mit gültigen Werten.</p> <p>Beispiel: A01, A02, B03, X04, X05, Y06, Y07, Y08 und Z09</p> <p>Alternativ kann auch ein Pattern (Muster) verwendet werden. Dabei kann mit reservierten Zeichen als Platzhalter gearbeitet werden:</p> <ul style="list-style-type: none"> + Nur numerische Zeichen @ Nur Textzeichen ? Beliebiges Zeichen * Beliebige Zeichen nachfolgend oder beginnend <p>Das Beispiel könnte als folgendes Muster angegeben werden: @0+</p>

2.2 Addons

2.2.1 Normales Addon

Zu jeder Prüfkonfiguration (z.B. *bmw-werk1-vda4913*) kann ein Addon (Java-Code) zur hinzugefügt werden. Im Addon können kundenspezifische Spezialprüfungen durchgeführt werden. Beispielsweise kann geprüft werden, ob im Verwendungskennzeichen ein 'T' steht und der nach Satz ein 716er-Satz ist.

Beispielcode zum Vergleich des Eintreffdatumsoll gegen das Übertragungsdatum:

```
/*
 * 11. Check: 712 Eintreffdatum Soll muss >= 711 Übertragungsdatum sein
 */
if (VDAStatics.LK_712.equalsIgnoreCase(lineKey) && VDAStatics.FK_712_EINTREFFDATUM_SOLL.equalsIgnoreCase(fieldKey))
{
    // Format ist in Ordnung, Prüfung gegen Übertragungsdatum:
    String line711 = (String) listOfTextLines.get(0);
    String uebertragungsdatum = StringTools.getSafeSubstr(line711, 33,39, "");
    if(value!=null) {
        if (validator.checkFormat("YYMMDD", value)) {
            try {
                Date valueDate = FORMAT_DATE_YY_MM_DD.parse(value);
                Date uebertragungsdatumDate = FORMAT_DATE_YY_MM_DD.parse(uebertragungsdatum);
                if (valueDate.before(uebertragungsdatumDate)) {
                    listOfErrors = Master.addError(listOfErrors, getAddonPrefix()
                        + "EINTREFFDATUM_SOLL_GROESSER",
                        "Eintreffdatum Soll muss juenger sein als Uebertragungsdatum.");
                }
            } catch (ParseException e) {
                listOfErrors = Master.addError(listOfErrors, "FORM", "format error", "YYMMDD");
            }
        }
    }
}
```

2.2.2 Backend-Addon

Das Backend-Addon verwendet die selben Methoden wie das normale Addon. Die Besonderheit ist hier, dass Zugriffe auf kundenspezifische Backend-Systeme durchgeführt werden können.

3 Monitoring

Der Web-basierte Monitor ermöglicht die Überwachung der vom MultiParser geprüften Dateien (ASNs). In der oberen Suchleiste können verschiedene Kriterien definiert werden. Je mehr Suchkriterien angewendet werden, desto kleiner wird die Ergebnismenge.

Status	ID	SLB	Datum	Aktion	Kunde	Prüfkonfiguration	Dateiname	27
✖ ERROR	1017		11.05.2007 16:00:26		BMW AG	BMW STARD VDA W6	W6_Test-Case10.txt	
✔ OK	1014	04000068	11.05.2007 15:59:06		BMW AG	BMW STARD VDA W6	W6_Test-Case8.txt.edit	
! TODO	1015	04000068	11.05.2007 15:59:06		BMW AG	BMW STARD VDA W6	W6_Test-Case9.txt.edit	
! TODO	1011	00000000	11.05.2007 15:59:05		BMW AG	BMW STARD VDA W6	W6_Test-Case4.txt	
! TODO	1012	04000068	11.05.2007 15:59:05		BMW AG	BMW STARD VDA W6	W6_Test-Case5.txt	
! TODO	1013	04000068	11.05.2007 15:59:05		BMW AG	BMW STARD VDA W6	W6_Test-Case6.txt	
! TODO	1008	04000068	11.05.2007 15:59:04		BMW AG	BMW STARD VDA W6	W6_Test-Case1.txt	
! TODO	1009	04000068	11.05.2007 15:59:04		BMW AG	BMW STARD VDA W6	W6_Test-Case2.txt	
! TODO	1010	04000068	11.05.2007 15:59:04		BMW AG	BMW STARD VDA W6	W6_Test-Case3.txt	
! TODO	1048	04000068	11.05.2007 00:00:00		BMW AG	BMW STARD VDA W6	W6_Test-Case1.txt	
! TODO	1049	04000068	11.05.2007 00:00:00		BMW AG	BMW STARD VDA W6	W6_Test-Case2.txt	
! TODO	1050	04000068	11.05.2007 00:00:00		BMW AG	BMW STARD VDA W6	W6_Test-Case3.txt	
! TODO	1051	00000000	11.05.2007 00:00:00		BMW AG	BMW STARD VDA W6	W6_Test-Case4.txt	
! TODO	1052	04000068	11.05.2007 00:00:00		BMW AG	BMW STARD VDA W6	W6_Test-Case5.txt	
! TODO	1053	04000068	11.05.2007 00:00:00		BMW AG	BMW STARD VDA W6	W6_Test-Case6.txt	
✔ OK	1054	04000068	11.05.2007 00:00:00		BMW AG	BMW STARD VDA W6	W6_Test-Case8.txt.edit	
! TODO	1055	04000068	11.05.2007 00:00:00		BMW AG	BMW STARD VDA W6	W6_Test-Case9.txt.edit	
✖ ERROR	1057		11.05.2007 00:00:00		BMW AG	BMW STARD VDA W6	W6_Test-Case10.txt	
! TODO	1079	04000068	11.05.2007 00:00:00		BMW AG	BMW STARD VDA W6	W6_Test-Case1.txt	
! TODO	1080	04000068	11.05.2007 00:00:00		BMW AG	BMW STARD VDA W6	W6_Test-Case2.txt	
! TODO	1081	04000068	11.05.2007 00:00:00		BMW AG	BMW STARD VDA W6	W6_Test-Case3.txt	
! TODO	1082	00000000	11.05.2007 00:00:00		BMW AG	BMW STARD VDA W6	W6_Test-Case4.txt	
! TODO	1083	04000068	11.05.2007 00:00:00		BMW AG	BMW STARD VDA W6	W6_Test-Case5.txt	
! TODO	1084	04000068	11.05.2007 00:00:00		BMW AG	BMW STARD VDA W6	W6_Test-Case6.txt	

© 2007 - Dynamic Software GmbH