

# Xml parsing with Postscript The XML Parser

Jänner 21, 2010


# Die Idee

XML stackartig aufgebaut → Entwicklung eines XML Parsers

- Überprüfung auf Wohlgeformtheit
- Fehlerhafte XMLs wohlformen
- Fehleranalyse
- formatierte Ausgabe

# Fallbeschreibung: Fall 1

- schließendes Tag fehlt
- Erkennung: schließendes Tag des Vorgängers gefunden
- Lösung: beim Schließen des Vorgängers alle Nachfolger schließen

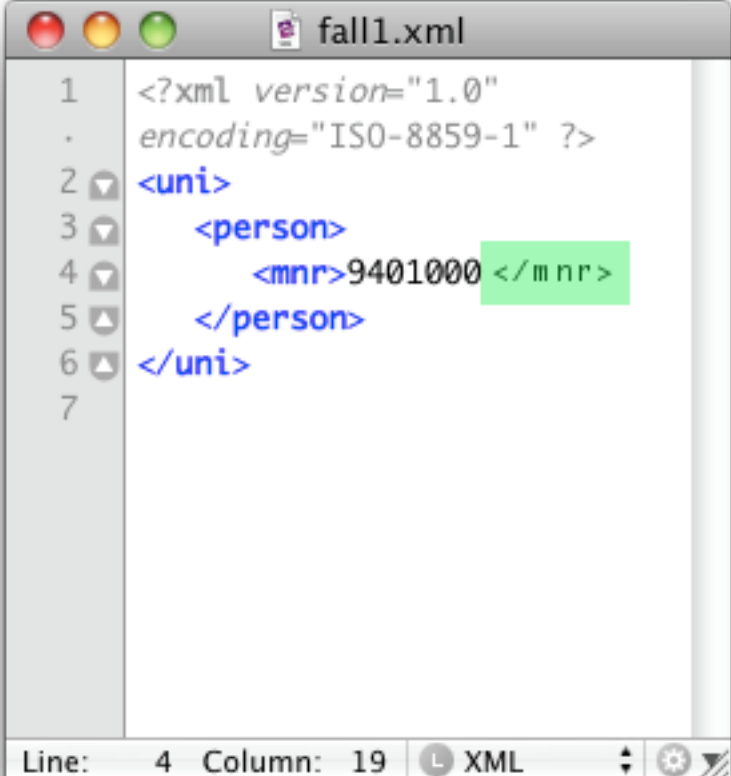


```
1  <?xml version="1.0"
.  encoding="ISO-8859-1" ?>
2  <uni>
3      <person>
4          <mnr>9401000
5      </person>
6  </uni>
7
```

The screenshot shows a text editor window with the title 'fall1.xml'. The editor contains an XML document. Line 1 is the XML declaration: `<?xml version="1.0" encoding="ISO-8859-1" ?>`. Line 2 is `<uni>`. Line 3 is `<person>`. Line 4 is `<mnr>9401000`. Line 5 is `</person>`. Line 6 is `</uni>`. Line 7 is empty. The status bar at the bottom indicates 'Line: 4 Column: 19' and 'XML'.

# Fallbeschreibung: Fall 1

- schließendes Tag fehlt
- Erkennung: schließendes Tag des Vorgängers gefunden
- Lösung: beim Schließen des Vorgängers alle Nachfolger schließen



```
1 <?xml version="1.0"
. encoding="ISO-8859-1" ?>
2 <uni>
3   <person>
4     <mnr>9401000 </mnr>
5   </person>
6 </uni>
7
```

Line: 4 Column: 19 XML

# Fallbeschreibung: Fall 2

- öffnendes Tag fehlt
- Erkennung: schließendes Tag gefunden, jedoch davor kein vorkommen eines öffnenden Tags
- Lösung: schließendes Tag verwerfen



```
1  <?xml version="1.0"
.  encoding="ISO-8859-1" ?>
2  <uni>
3      <person>
4          <mnr>9401000</mnr>
5          </fail>
6      </person>
7  </uni>
8
```

The screenshot shows a text editor window titled 'fall2.xml'. The code is XML. Line 1: `<?xml version="1.0"`, Line 2: `encoding="ISO-8859-1" ?>`, Line 3: `<uni>`, Line 4: `<person>`, Line 5: `<mnr>9401000</mnr>`, Line 6: `</fail>`, Line 7: `</person>`, Line 8: `</uni>`. The status bar at the bottom shows 'Line: 5 Column: 9' and 'XML'.

# Fallbeschreibung: Fall 2

- öffnendes Tag fehlt
- Erkennung: schließendes Tag gefunden, jedoch davor kein vorkommen eines öffnenden Tags
- Lösung: schließendes Tag verwerfen



```
1 <?xml version="1.0"
. encoding="ISO-8859-1" ?>
2 <uni>
3   <person>
4     <mnr>9401000</mnr>
5   </person>
6 </uni>
7
8
```

Line: 5 Column: 9 XML

# Fallbeschreibung: Fall 3

- XML besteht nur aus öffnenden Tags
- Erkennung: am Ende noch Tags offen
- Lösung: nach der Reihe Tags schließen

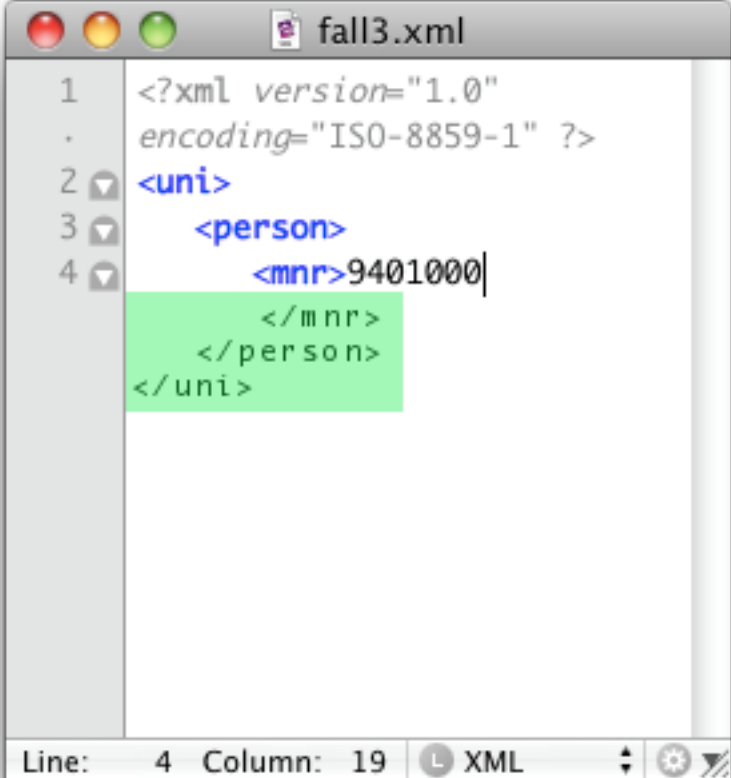


```
1 <?xml version="1.0"
. encoding="ISO-8859-1" ?>
2 <uni>
3   <person>
4     <mnr>9401000|
```

The screenshot shows a text editor window with the title 'fall3.xml'. The editor contains an XML document. Line 1 is the XML declaration: `<?xml version="1.0" encoding="ISO-8859-1" ?>`. Line 2 is `<uni>`. Line 3 is `<person>`. Line 4 is `<mnr>9401000|`. The cursor is at the end of line 4. The status bar at the bottom indicates 'Line: 4 Column: 19' and 'XML'.

# Fallbeschreibung: Fall 3

- XML besteht nur aus öffnenden Tags
- Erkennung: am Ende noch Tags offen
- Lösung: nach der Reihe Tags schließen



```
1 <?xml version="1.0"
. encoding="ISO-8859-1" ?>
2 <uni>
3   <person>
4     <mnr>9401000|
      </mnr>
      </person>
    </uni>
```

Line: 4 Column: 19 XML



# Features

- Unterstützung von
  - Processor Instructions
  - Comments
  - CDATA
  - sofort schließende Tags
- Wohlformungen
  - Erkennen der Fälle 1-3
  - Wohlformungen von Mischfällen
- Eingabe über Inputparameter
- Debugausgaben (wahlweise ausblendbar)
- Formatierte Ausgabe

# Workflow

