

OPTIMIERTER

DATENAUSTAUSCH

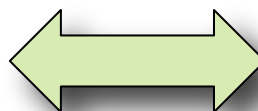
**Stoff
Datenverarbeitung
GmbH**

www.stoffdv.at

OPTIMIERTER

DATENAUSTAUSCH

Die heutige IT-Landschaft besteht aus vielen Businesssystemen. Der Datenaustausch zwischen diesen Systemen, ob es EDIfact, XML oder andere Files sind, soll **einfach** und **kostengünstig** erledigt werden. Zudem sollen die notwendigen Konvertiervverfahren eindeutig beschrieben werden. Hier werden sie mit Activity-Diagrammen nach UML **dokumentiert**. Die Daten selbst sind in einer Datenbank sicher abgelegt und jeder Konvertierschritt und Import/Export **nachvollziehbar** aufgezeichnet.



OPTIMIERTER

DATENAUSTAUSSCH

1. Sichere, nachvollziehbare Datenhaltung
2. In beliebige Jobsteuerungssysteme integrierbar
3. Standardkonvertierungen für EDIfact
4. Standardkonvertierungen für XML
5. Flexibler Konverter
6. Konvertierungen werden in UML dokumentiert
7. Verwendung als Prototyp für EAI-Systeme
8. Kosten

```
<?xml version = '1.0' encoding = 'UTF-8' ?>
<?xml version = '1.0' encoding = 'UTF-8' ?>
<?xml version = '1.0' encoding = 'UTF-8' ?>
<?xml xmi.version = '1.2' xmlns:UML = 'org.omg.xml.namespace.UML'>
  <UML:header>
    <UML:documentation>
      <UML:exporter:Netbeans <UML:Writer></UML:exporter>
      <UML:exporterVersion:1.0</UML:exporterVersion>
    </UML:documentation>
    <UML:metamodel xmi.name="UML" xmi.version="1.4"/> </UML:header>
  <UML:content>
    <UML:Model xmi.id = 'model1' name = 'null' isSpecification = 'false' isRoot = 'false' isLeaf = 'false'>
      <UML:Namespace.ownedElement>
        <UML:Class xmi.id = 'class4' name = 'CS_TransactClient' const='false' static='false' visibility='public'>
          <UML:ModelElement.stereotype>
            <UML:Stereotype xmi.idref = 'stereo8741' />
          </UML:ModelElement.stereotype>
          <UML:Classifier.feature>
            <UML:Attribute xmi.id='Varia5' name='iTrnr' const='false' static='false' visibility='public'>
              <UML:StructuralFeature.multiplicity>
                <UML:Multiplicity xmi.id = '1'>
                  </UML:StructuralFeature.type>
                </UML:Attribute>
            <UML:Attribute xmi.id='Varia7' name='bOk' const='false' static='false' visibility='public'>
```

```
<?xml version = '1.2' xmlns:UML = 'org.omg.xml.namespace.UML'>
<?xml version = '1.2' xmlns:UML = 'org.omg.xml.namespace.UML'>
<?xml version = '1.2' xmlns:UML = 'org.omg.xml.namespace.UML'>
<?xml xmi.version = '1.2' xmlns:UML = 'org.omg.xml.namespace.UML'>
  <UML:header>
    <UML:documentation>
      <UML:exporter:Netbeans <UML:Writer>
      <UML:exporterVersion:1.0</UML:exporterVersion>
    </UML:documentation>
    <UML:metamodel>
      <UML:metamodel xmi.name = 'UML' xmi.version = '1.4'>
    </UML:metamodel>
  <UML:content>
    <UML:Model>
      <UML:Model xmi.id = 'model1' name = 'null' isSpecification = 'false' isRoot = 'false' isLeaf = 'false'>
        <UML:Namespace.ownedElement>
          <UML:Class>
            <UML:Class xmi.id = 'class4' name = 'CS_TransactClient' const='false' static='false' visibility='public'>
              <UML:ModelElement.stereotype>
                <UML:Stereotype xmi.idref = 'stereo8741' />
              </UML:ModelElement.stereotype>
              <UML:Classifier.feature>
                <UML:Attribute>
                  <UML:Attribute xmi.id='Varia5' name='iTrnr' const='false' static='false' visibility='public'>
                    <UML:StructuralFeature.multiplicity>
                      <UML:Multiplicity xmi.id = '1'>
                        </UML:StructuralFeature.type>
                      </UML:Attribute>
                <UML:Attribute>
                  <UML:Attribute xmi.id='Varia7' name='bOk' const='false' static='false' visibility='public'>
```

OPTIMIERTER

DATENAUSTAUSCH

JanusStoffDV - Blätter recherchieren, verwalten; Cluster 00

Menü Hilfe

Kopfdaten

BlattId 290/xmlsap.xml
eingefügt 08.07.04 12:31:25 XML2Tab/Fritz, besetzt 03.12.04 10:02:26 Dbh4F
freigegeben 08.07.04 12:31:25 XML2Tab/Fritz, erledigt 03.12.04 10:02:26 Db
Schlagworte
Typ XMLtab
Schnittste

Alle Datenpakete werden unveränderbar in ein Datenbanksystem eingebracht. Jeder Schritt wie Import/Export und Konvertierung wird nachvollziehbar mitdokumentiert. Darüber sind Recherchen durchführbar.

Treffer 18/

183 Umläute	ZFIDCCP02 IDOC EDI_DC40 < SEGMENT ="1" <
210 JVC	ZFIDCCP02 IDOC EDI_DC40 TABNAM < < EDI_DC40
222 822498.xml	ZFIDCCP02 IDOC EDI_DC40 < SEGMENT ="1" <
239 x1.xml	ZFIDCCP02 IDOC EDI_DC40 MANDT < < 200
253 x1.xml	ZFIDCCP02 IDOC EDI_DC40 < SEGMENT ="1" <
254 x1.xml	ZFIDCCP02 IDOC EDI_DC40 DOCNUM < < 0000000010571248
274 822498.xml	ZFIDCCP02 IDOC EDI_DC40 < SEGMENT ="1" <
279 x2.xml	ZFIDCCP02 IDOC EDI_DC40 DOCREL < < 620
280 x3.xml	ZFIDCCP02 IDOC EDI_DC40 < SEGMENT ="1" <
281 x4.xml	ZFIDCCP02 IDOC EDI_DC40 STATUS < < 41
282 x5.xml	ZFIDCCP02 IDOC EDI_DC40 < SEGMENT ="1" <
283 x5.xml	ZFIDCCP02 IDOC EDI_DC40 DIRECT < < 1
284 x4.xml	ZFIDCCP02 IDOC EDI_DC40 < SEGMENT ="1" <
285 x3.xml	ZFIDCCP02 IDOC EDI_DC40 OUTMOD < < 2
286 x2.xml	ZFIDCCP02 IDOC EDI_DC40 < SEGMENT ="1" <
287 xmlsap.xml	ZFIDCCP02 IDOC EDI_DC40 IDOC TYP < < ZFIDCCP02
289 xmlsap.xml	ZFIDCCP02 IDOC EDI_DC40 < SEGMENT ="1" <
290 xmlsap.xml	ZFIDCCP02 IDOC EDI_DC40 < SEGMENT ="1" <

Sichere, nachvollziehbare Datenhaltung

OPTIMIERTER

DATENAUSTAUSCH

Das Datenverwaltungs- und Konvertiersystem „Janus“ lässt sich innerhalb der Microsoft-Betriebssystemfamilie in beliebige Jobsteuerungssysteme integrieren.



```
ProgLog_Dbh4Files_20080205_141353Pid2344.txt - Editor
Datei Bearbeiten Format Ansicht ?
Programm-Start 05.02.2008 14:13:53
OS-Info: 2600 5 1
Programmversion erstellt: Mar 20 2007 11:39:21
Parameter ->
  Test:nein
  Cluster:00
  Blattnr:
  Blatt:
  Files:*.xml
  Aktive:ja
  Freigegeben:ja
  Typ:XML
  Schlagworte:
  Schnittstelle:
  Erstellt:
  X-Flow:
  Proglog(Dir):ja
1_TEAMTRANS-_50156955_20080104_1059_BEX_TPA_OUT_36.xml wird verarbeitet
Blatt 904 erstellt
Zeile 1, Spalte 1
```

In beliebige Jobsteuerungssysteme integrierbar

OPTIMIERTER

DATENAUSTAUSSCH

```
UNA:+. ? '  
UNB+UNOC:3;Senderkennung+Empfaengerkennung+060620:0931+1++1234567'  
UNH+1+ORDERS;D:96A:UN'  
BGM+220+B10001'  
DTM+4:20060620:102'  
NAD+BY+++Bestellername+Strasse+Stadt++23436+xx'  
LIN+1++Produkt Schrauben:SA'  
QTY+1:1000'  
UNS+S'  
CNT+2:1'  
UNT+9+1'  
UNZ+1+1234567'
```

Für EDIfact werden die offiziellen Definitionen der UNO verwendet. Natürlich können kundenspezifische Abweichungen zusätzlich eingepflegt werden.

```
UNB;UNOC;3;Senderkennung;;;Empfaengerkennung;;;060620;0931;1;;;1234567  
UNH;1;ORDERS;D;96A;UN  
BGM;220;;;B10001  
DTM;4;20060620;102  
NAD;BY;;;;;;;;;Bestellername;;;;;;;;;Strasse;;;Stadt;;;;;23436;xx  
LIN;1;;Produkt Schrauben;SA  
QTY;1;1000
```

Standardkonvertierungen für EDIfact

DATENAUSTAUSCH

Eingehende XML-Nachrichten werden nach den Standards für das interne Konvertierformat zerlegt. Danach lassen sich die Inhalte komfortabel umsetzen. Ausgehende XML-Nachrichten werden aus dem internen Konvertierformat erstellt.

Standardkonvertierungen für XML

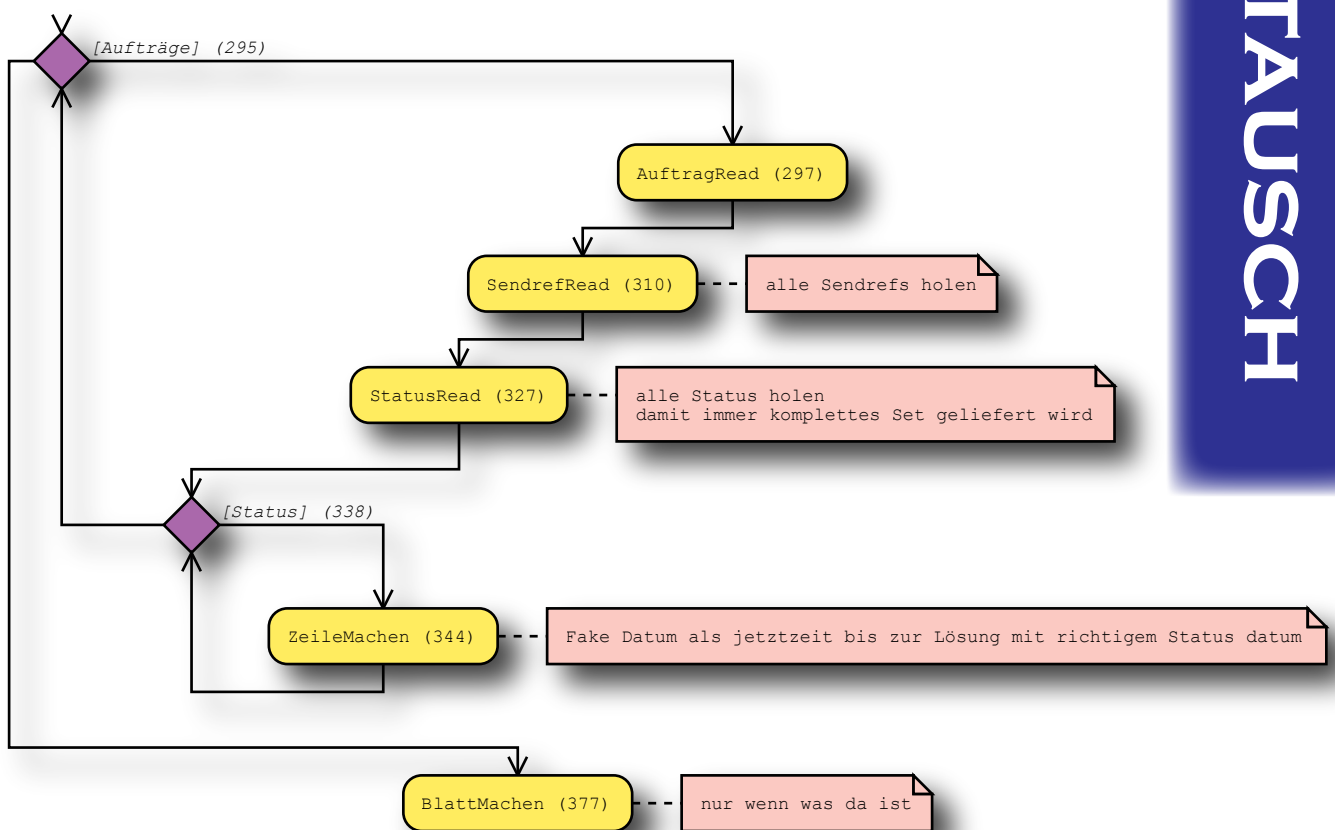
DATENAUSTAUSCH

Alle Datenpakete werden über Schablonen beschrieben. Damit können die Daten unverändert interpretiert werden. Die in den Schablonen definierten Bezeichnungen werden im Konvertiermodul verwendet. Für die Umsetzung werden mächtige Konvertierklassen eingesetzt. Da als Basis C++ genutzt wird, sind auch komplizierteste Umsetzungen möglich.

OPTIMIERTER

DATENAUSTAUSCH

Die eingesetzten Konvertiermodule werden aus der im Quellcode mitgeführten Dokumentation in UML-Diagrammen dargestellt.



Konvertierungen werden in UML dokumentiert

Filename: KonTest.h
Line: 77

clsParseXMLdoc

```
clsParseXMLdoc()  
clsParseXMLdoc(P: const clsParseXMLdoc &)  
operator =(P: const clsParseXMLdoc &): clsParseXMLdoc &  
fnGibXMLstring(P: const clsParseXMLdoc &): clsUCCstring  
fnNimmXMLstring(S: const clsUCCstring &, iPos: unsigned int): unsigned int  
fnStufeM(): unsigned int  
fnStufe(S: const clsUCCstring &): unsigned int  
fnStufeP(S: const clsUCCstring &): unsigned int  
fnGibStufen(): unsigned int  
fnGibStufeAttrVector(iStufe: unsigned int, vectorAttr: vector<strXMLtagattr> &): clsUCCstring  
fnGibLetzteStufeAttrVector(vectorAttr: vector<strXMLtagattr> &): clsUCCstring  
fnLetzteStufeLeer(): bool  
fnLetzteStufeNurAttr(): bool  
fnGibInhalt(): clsUCCstring  
fnNimmInhalt(S: const clsUCCstring &): void
```

Verwendung als Prototyp für EAI-Systeme

Die erarbeiteten Klassen, (Datenbank-)strukturen und Prozeduren können mit dem Nachweis der Funktionsfähigkeit in EAI-Systeme portiert werden.

OPTIMIERTER

DATENAUSTAUSCH

OPTIMIERTER

DATENAUSTAUSSCH

Für die Einrichtung und den Betrieb sind die vereinbarten Sätze zu entrichten.

Lizenzkosten entfallen!

Den tatsächlichen Gewinn zeigt eine Kostenanalyse.

Kosten

OPTIMIERTER

DATENAUSTAUSCH

Stoff Datenverarbeitung GmbH

Anton Baumgartnerstr. 44/A8/173 | A-1230 Wien

Tel.: +43 699 12241677 | E-Mail: f.stoff@stoffdv.at

www.stoffdv.at