

DECUS 2004

Erfolgreiche Softwareportierung proprietärer Lösungen

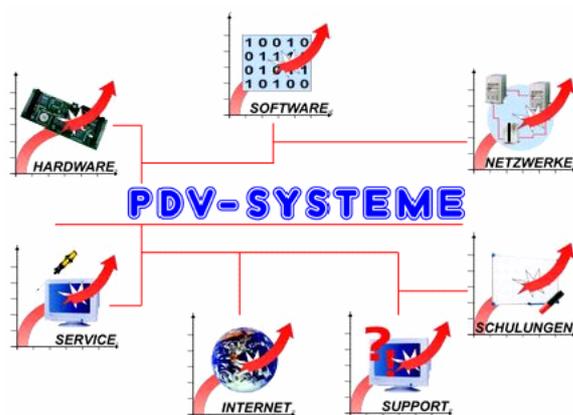
oder

Die Herausforderung, geistiges Eigentum anderer zu verstehen!

Tristan Niewisch, Leiter Softwareabteilung

PDV-SYSTEME

Das Unternehmen



PDV-SYSTEME

Geschichte

1980	IWP- Ingenieurbüro für Wärme- und Prozeßtechnik
1983	Erweiterung zum Systemhaus PDV-SYSTEME oHG
1989	Umfirmierung zur PDV-SYSTEME Gesellschaft für Systemtechnik mbH
1991	Gründung der PDV-SYSTEME SACHSEN als eigenständige GmbH
1995	Expansion der Software-Distribution
1998	Ausrichtung als Service- und Dienstleistungsunternehmen
2001	Gründung der Niederlassungen München und Bremen
2003	Gründung der Niederlassung Wetzlar
Heute	Über 80 Mitarbeiter an 5 Standorten



PDV-SYSTEME

PDV-SYSTEME in Zahlen

- Geschäftsstellen Goslar, Bremen, München, Wetzlar - z.Zt. ca. 56 Mitarbeiter
- PDV-SYSTEME Sachsen, eigenständig, z.Zt. ca. 25 Mitarbeiter
- Geschäftsfläche Goslar 800 m² + 600 m² Lager



PDV-SYSTEME

Bereich Hardware

Produkte

- INTEL/Alpha Systeme von **HP** u.a.
- UNIX- und VMS-Systeme von **HP**
- UltraSPARC-Systeme von **SUN**
- Storage von **HP, Baydel** und **TMS**
- Bandsysteme von **HP** und **Overland Storage**



Lösungen

- Hochverfügbarkeit
- Storage / NAS / SAN / iSCSI
- Datensicherheit / Backup
- Komplettlösungen (Server, Storage, Backup)



PDV-SYSTEME

Bereich Service & Support

Service

- „End of life“-Service für **Digital-** und **Compaq**-Systeme
- Internet-Outsourcing
- Vor-Ort-Unterstützung
- Ersatzteilhaltung und Ersatzteilvertrieb
- Wartungsverträge



Support

- Erstellung von Domain- und Migrationskonzepten
- Implementierung von Systemmanagement-Lösungen
- Unterstützung für die Plattformen **MS-Windows**, **Linux**, **Unix** (Tru64, HP-UX, Solaris) und **OpenVMS**
- Backup- und Restore-Konzepte
- User Help Desk



PDV-SYSTEME

Bereich Software

Eigenentwicklung

- Industrielle Software-Lösungen
- Kaufmännische Software-Lösungen
- Individuelle Programmentwicklung
- Softwareanpassungen
- Softwarewartung







networking solutions



Distribution

- Hummingbird Ltd.
- Process-Software
- Ericom Software
- Attachmate
- eigene Software-Produkte

PDV-SYSTEME

Eingesetzte Sprachen

- FORTRAN IV (PDP, VAX, Alpha)
- FORTRAN 77 (VAX, Alpha, DOS (Watcom))
- Assembler (PDP, VAX, Alpha)
- DEC-Pascal (VAX, Alpha)
- DEC-C (PDP, VAX, Alpha)
- DEC-C++ (VAX, Alpha)
- DCL (VAX, Alpha)
- VAX-C (VAX)
- MS Visual C++ 4.0, 4.2, 5.0, 6.0 (WIN 9x, WIN-NT)
- MFC / ATL / STL unter C++ (WIN 9x, WIN-NT)
- OPEN GL (WIN-NT)
- MS Visual Basic (WIN 9x, WIN-NT/2000/XP)

- VBA (MS-Office, Excel, Word, Access, Outlook)
- Objekt Pascal
- Perl (OpenVMS, div. Unix)
- JavaScript (WIN 9x, WIN-NT/2000/XP)
- Visual J++ (WIN 9x, WIN-NT)
- Java Developer Kit 1.4 (WIN 9x, WIN-NT/2000/XP)
- Visual Fortran (WIN 9x, WIN-NT)
- Delphi 1.,2.,3.,4.,5.
- J-Developer
- C++ Builder 1.,3.,4.,5.,6
- C++ Linux, Tru64 Unix
- PHP, PL/SQL, PERL etc.

PDV-SYSTEME

Methoden

Das in mittlerweile 24 Jahren entstandene Know-how ermöglicht uns Projekte aus der „EDV-Steinzeit“ – also der Zeit der ersten DECUS Veranstaltungen - softwaretechnisch effektiv auf den aktuellsten Stand zu bringen.

Für die Erneuerung von Software gibt es drei (vier) grundsätzlich unterschiedliche Ansätze

- Rebuild-from-Kernel
- Portierung
- GUI-fizierung
- Portallösung, Framework, Information Management System

PDV-SYSTEME

Rebuild from Kernel

Vorhandene Funktionalität wird analysiert und komplett neu erstellt, ohne dabei auf vorhandenen Code Rücksicht zu nehmen.

- Erweiterte neue Funktionalitäten können eingebunden werden
- Pflichtenheft entspricht Ursprungsprogramm plus neu zu beschreibender Funktionalität
- im Programm verankertes - nicht offensichtliches Wissen - geht verloren
- bedingt einen CUT, an dem das alte System abgeschaltet und das neue fehlerfrei laufen muss
- Datenbestände müssen auf den Stichtag konvertiert werden
- Neues Programm darf keine Fehler aufweisen, bei denen Daten verloren gehen - Rückschritt meist nicht möglich

PDV-SYSTEME

LIS-II

Method	Typ	Kommentar
AM	3	AM berechnet in LIS
AUTL_D	1	Letztstand auf Automatik für Dosierung
	1	Letztstand auf Automatik für Sichteinstellung
	1	POLAB auf Automatik für Dosierung
	1	POLAB auf Automatik für Sichteinstellung
Da_1_2		
FA03		
Freib ab		
K20	1	K20
KDMP_1	1	
KDMP_2	1	
KDMP_3	1	
KDMP_4	1	
KDMP_5	1	
KST	3	KST berechnet in LIS mit S03 - Berücksichtigung
MAI_NR	1	Manuale
MgD	1	MgD von SRS
NI	1	NI von SRS
Na03	1	Na03 von SRS
Na03_A	3	Na03-Argument
NI	1	NI von SRS
Obbau	1	
PRSB_GLEICH	1	PRSB Steigungsparameter von Malven
PRSB_LAGE	1	PRSB Lageparameter v' von Malven
Pradial	1	
SLO_NR	1	SloNr
SM	3	SM berechnet in LIS
S03	1	S03 von SRS
S02	1	S02 von SRS
T02	1	T02 von SRS
mgm25	1	
man040	1	
man062	1	

PDV-SYSTEME

Portierung

Vorhandener Programmcode wird analysiert und auf die neue Systemumgebung angepasst.

- gleiche Funktionalität
- erlaubt ggf. Anbindung „alter“ Peripherie
- mögliche Integration/Nutzung vorhandener Schnittstellenprogramme
- Schwierigkeit liegt im Erkennen von versteckten Strukturen

PDV-SYSTEME

Portierung

The terminal window displays the following configuration commands:

```

$! the port number of the PDU-IMS Time Server running on the host in the
$! format "host:port" (the port defaults to 5000 if omitted).
$!
$! device == "SVSSDCF77"
$
$! Define the speed of the serial interface for DCF or GPS time providers.
$! DCF usually runs on 9600 baud and GPS on 4800 baud.
$!
$! speed == "4800"  Ifor GPS receiver
$! speed == "9600"  Ifor DCF clock
$
$! Define the options for PDU-IMS. For a complete list and description of
$! the several options see the user manual.
$!
$! options == "/INI=600 /DELTA=10 /OUT=SVSSIMS:TIME.LOG /SWTIME /RESTART"
$
$! Define the TDF (time differential factor, i.e. the offset from UTC to
$! local system time) for your system during standard time and daylight
$! savings time. Please note that the offset is given in seconds.
$!
$! standard_tdf == "+3600"  ! in seconds !
$! daylight_tdf == "+7200"  ! in seconds !

```

The GUI configuration dialog box, titled "figuration", shows the following settings:

- Zeitgeber: Synchronisation
- Abgleichmethode: [Dropdown]
- Intervall (s): 600
- Logfile: RA*\1\PDV-SY*\1\PDVTMS.log
- maximale Dateigröße einstellen: 8 KB

PDV-SYSTEME

GUI-fizierung

Das alte Programm bleibt erhalten, bekommt aber eine neue Oberfläche

- Parallelbetrieb, ohne dass die Daten oder die Funktionalität geändert werden
- Erweiterte Funktionalität eingeschränkt möglich (durch Zusammenfassung von anwenderfreundlichere Bedienung)
- Programmlogik muss vom Guifizierer © verstanden werden
- Zusätzlicher Overhead/Fehlerquelle – Keine Performancesteigerung
- Löst keine Probleme der vorhandenen Implementierung
- Nachentwicklung der Oberfläche bei Änderungen am unterlagertem Programm

PDV-SYSTEME

MEDAS Classic

AXPSW KEAI 340
MFDAS PDU-SYSTEME
VRTUELLE KANAEL
14.02.2002 14:16:22

Kanalnummer : 18961
 Bezeichnung : Berechneter Füllstand Zementsilo 1 Kammer 4
 Kurzbezeichnung : FUELLS_ZS1_K4

Aktiviert : J (J/N) Zykluszeit : 00:10:00
 Dimension : T0
 Berechnung : 0

Parameteranzahl :
 Parameter :
 MPS-Formel : $erg := (wert(19924) + wert(19925) + wert(19926) + wert(19927))$

Speicherung : 0
 Speicherzyklus : Toleranzband :
 Lebensdauer MDB : Einträge in MDB :

Modifizieren
Alarmbearbeitung
Beenden

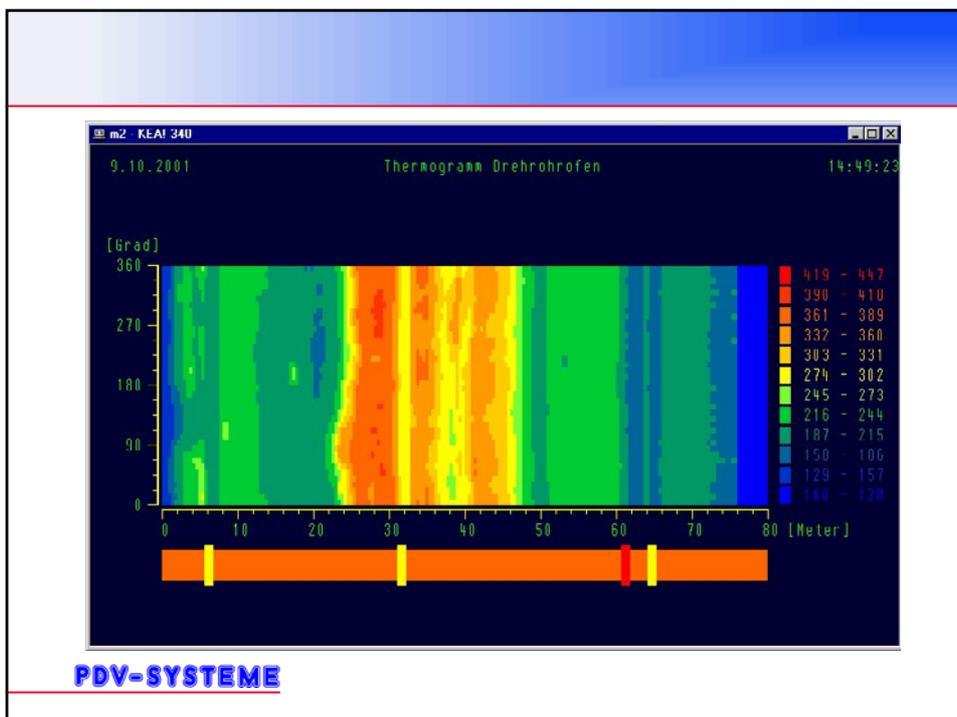
Auswahl
<CURSOR TASTEN>

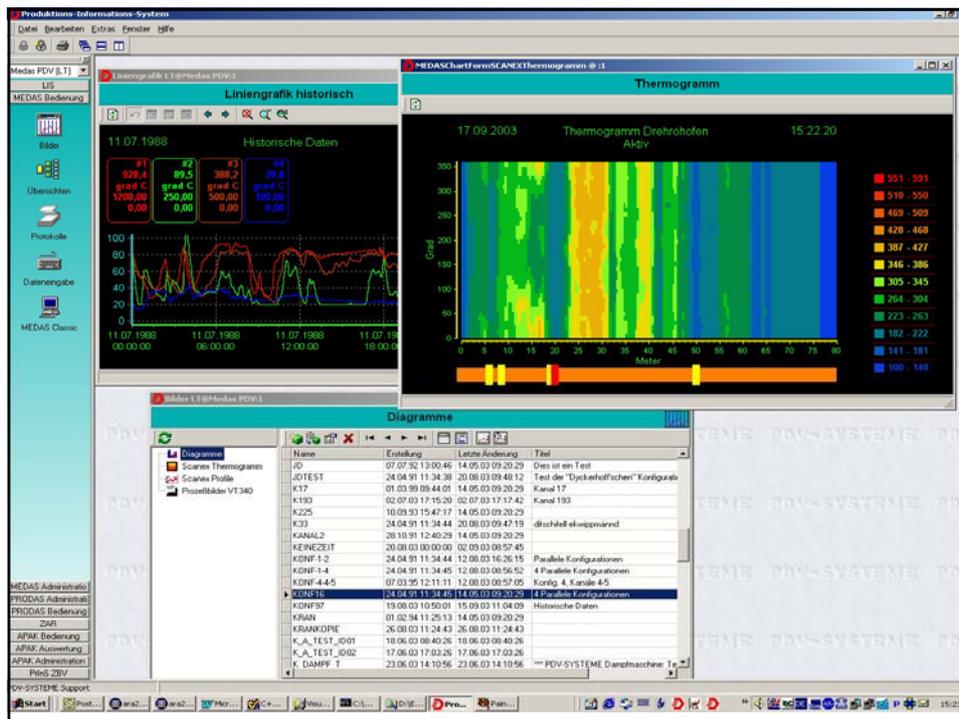
Ausführung
<RETURN>

Ende
<PF4>



PDV-SYSTEME





Portallösung

Eine Portallösung besteht aus konfigurierbaren Modulen und Komponenten welche über intelligente Zugriffsmechanismen und -pfade vorhandene Informationen sammeln und im Ergebnis wie eine programmierte Anwendung erscheinen – d.h. es wird konfiguriert statt programmiert.

Der Schwerpunkt für Portallösungen liegt z.Zt. mehr im kommerziellen Bereich.

Portallösungen erlauben den Zugriff auf alle Unternehmensdaten und Ressourcen von einer zentralen Arbeitsoberfläche.

Portallösungen haben ihre Stärken wenn es um die Verwaltung, Verknüpfung, Verdichtung und Visualisierung von Datenbeständen geht – also das was man früher als Datenverarbeitung umschrieben hat.

Portallösungen sind nicht geeignet für steuerungstechnische Abläufe oder systemnahe Prozesse.

PDV-SYSTEME

Hummingbird Enterprise

- Hummingbird Portal
- Hummingbird Connectivity
- Hummingbird DM
- Hummingbird RM
- Hummingbird BI
- Hummingbird Collaboration
- Hummingbird ETL

PDV-SYSTEME

Portierungsklappen und -löcher

**Die Schwierigkeit, geistiges Eigentum
anderer zu verstehen!**

oder

**Portierungsklappen und -löcher
für Softwareentwickler**

Eine Auswahl

PDV-SYSTEME

Erster Eindruck

GHH-RAND Schraubenverdichter Eingabe
 CG80 : V00.48

Kennwort : CS80/1 Datum/Uhrzeit : 30.01.2002 14
 Type : CS80/1
 Auftragsnummer :
 Besteller :
 Maschinenummer : 502406
 Erprobungsmedium : Luft
 Bemerkung : Oelfreier Luftkompressor

Bemerkung z. Erprobung :
 Erprobungsleiter : KL

Eingabewerte korrigieren (J/N) [N] :

GHH-RAND Schraubenverdichter Eingabe
 CG80 : V00.48

Eingabe der Maschinenummer
 Seriennummer : 00000
 Maschinenummer : 502406

Standardmaschine :
 Maschine nochmals prüfen (J/N)

Erprobungsleiter : KL

Eingabewerte korrigieren (J/N) [N] :

PDV-SYSTEME

Das Ergebnis

The screenshot shows the PDV-SYSTEME software interface. The main window is titled 'Kompressoren - Oelingspritzt zweiufig - CF128'. On the left, there is a navigation tree with categories like 'Probelauf', 'Druckprobe', 'Freigabe', 'Datenangabe', 'Statistik', 'Tagesprotokolle', 'Systemprotokoll', and 'Kalibrierung'. The main area displays a table of compressor data:

Kompressor	Maschinenummer	Maschinenr HD	Erfassungzeit	verschrotet
CF128	672431		15.09.2003 12:42:17	<input type="checkbox"/>
CF128	div		02.06.2003 19:22:00	<input type="checkbox"/>

Below the table, there is a detailed view for the selected compressor (CF128). The 'Eigenschaften' section shows:

- Kategorie: Oelingspritzt, zweiufig
- Maschinenummer: 672431
- Artikelnummer: [empty]
- Registrierenummer: [empty]
- Auftragsnummer: 0548785
- Zeitpunkt des letzten Probelaufs: 27.07.2003 00:00:00
- Status des letzten Probelaufs: [empty]

The 'Probelaufe' section shows:

- Probelauf erforderlich:
- Druckprobe erforderlich:
- Erfassungzeit: 27.07.2003 00:00:00
- Zeitpunkt des letzten Druckprobe: [empty]
- Status der letzten Druckprobe: [empty]

Portierungsklappen und -löcher

Verschiedene Entwickler haben (ggf. über Jahre) an einem Programmcode gearbeitet. Da gab es Kompetente, Vorsichtige, Faule (Dokumentation), Pragmatiker und Visionäre - und vor allem jede Menge Code, der erkennbar unter Zeitdruck entstanden ist.

```

Rev:1.0 24.08.90 HL / HS / KH
Rev:2.0 28.09.90 HL
Rev:3.0 06.08.92 SPR
Rev:4.0 22.02.95 FB
Rev:5.0 04.07.95 FB
Rev:6.01 30.01.96 MF
Ver: 7.02 12.02.96 MB
Ver: 7.32 05.08.97 MF
Ver: 8.51 12/19/98 FB
Ver: 0.24 MF

```

Status 2003 - alle Entwickler sind bis auf einen verstorben (bzw. nicht mehr greifbar), der letzte Verbliebene geht in 10 Monaten in Rente.

PDV-SYSTEME

Portierungsklappen und -löcher

Wenn im einem existierenden Programm innerhalb eines Produktionsprozesses ein Fehler auftritt, dann bleibt meist keine Zeit für Schönheit. Provisorien in der Software halten länger als die, welche ein Zahnarzt temporär einsetzt.

.....

```

VORG(10)=VORG(8)/VORG(6)
If (VORG(5).GT.100.) VORG(5)=VORG(3)/VORG(5)
VORG(4)=VORG(13)*1000/VORG(3)*0.10472
If (VORG(5).GT.100.) VORG(5) = VORG(3)/VORG(5)
VORG(10)=VORG(8)/VORG(6)
VORG(12)=VORG(12)
VORG(14)= VORG(13)/VORG(12)*60.
VORG(17) = 2.40
VORG(4) = VORG (13)/0.93
110  FORMAT (F6.2)
      IF (VORG (3) . LE. 2000.) THEN FAK(1) = FAK (26)
      usw., usw. usw.

```

PDV-SYSTEME

Beispiele

Codezeilen sind über die Zeit bedeutungslos geworden, Messgrößen werden nicht mehr benutzt.

Programmzeilen sind auskommentiert –
Wofür waren sie da? Werden sie irgendwann doch noch benötigt?

Im Fehlerfall wurden auskommentierte Programmzeilen ggf. wieder editiert und das Programm neu compiliert – wann treten diese Fehler auf? Wird das ggf. noch benötigt ?

```
C      CALL FPUT (MELD12',MELD1')
C      WRITE (IUNZD,1108) SEC
      IF (IFOFL=1) GOTO 1520
C      CALL WFOR80 (3)
      IFOFL=1
1520   CONTINUE
C      END IF
```

PDV-SYSTEME

Beispiele

Namen von Variablen werden beibehalten, repräsentieren aber einen Inhalt, der nicht dem Namen entspricht. Das ist für Entwickler schwierig, wenn man versucht die Logik zu verstehen

Ein Beispiel von Vielen :

Eine Variable mit dem Namen GASKON (Gaskonstante?) entpuppt sich als Wert der die Reynoldszahl (ein Wert, welcher einen Rückschluss auf die Art der Strömung gibt)

Eine andere Größe LDREHZ (Hauptläuferdrehzahl ?) taucht im Code immer wieder auf, wird aber nirgendwo Ausgegeben und hat auch keine Funktion mehr, bzw heißt anders.

PDV-SYSTEME

Beispiele

„GoTo-Gewurschtel“

Sprünge, die quer durch das ganze Programm und zurück führen, ohne das sich der Sinn dafür sofort erschließt - Eine besondere Herausforderung, wenn diese Entwicklung in eine strukturierte Sprachumgebung portiert werden soll.

```
1359      IF (INDIG(1).EQ.1) GOTO 1360
          GOTO 1840
1360      CONTINUE
          IOUDIG(3)=IOU1
          GOTO 1820
```

..... 5 Seiten weiter

```
1820      CONTINUE
C         STMELD= 'STOERMELDUNG VOM PRUEFSTAND'
1840      CONTINUE
```

.....
PDV-SYSTEME

Beispiele

„Dynamische“ Schleifenzähler

Beispiel: Ein Programm soll bei einem Beharrungszustand abbrechen.

```
          IZA1=0
1440      CONTINUE
          IF (IZA1.LT.17) GOTO 3060
3060      CONTINUE
C         IF(IENDE.EQ.1) GOTO 1680
1580      IZA1=7
```

div. Programmcode über 4 Seiten mit diversen Ein-, Um- und Blocksprüngen

```
          IF (IZA1.GE.7) GOTO 1680
          IZA1=IZA1+1
          GOTO 1440
1680      CONTINUE
          IF (IZA1.GE.17) GOTO 1740
          IZA1=IZA1+1
          GOTO 1440
1740      CONTINUE
```

.....
PDV-SYSTEME

Beispiele

Alte Speichereinsparungslasten:

Ein Programm, das in Zeiten teurer Speicher und enger Adressierungsräume erstellt wurde, enthält manchmal „Kunstgriffe“ (z.B. Common Arrays), die sehr exzessiv verwendet werden

Beispiel: Array mit dem Namen Zeit(3), wobei:

Zeit(1)=Uhrzeit,
Zeit(2)=Datum,
Zeit(3)=Drehzahl (!?) ist

PDV-SYSTEME

Beispiele

Schlaue (?) Lösungen und Programmiertricks – Ein Beispiel:

Ein Probelauf soll beendet werden, wenn die Temperatur der vorhergehenden Messung mit der aktuellen Messung konstant ist.

Als der Entwickler die zusätzliche Aufgabe bekam, auf eine Eingabe hin den Probelauf trotzdem weiterzuführen – also einen Dauerlauf der Messungen zu ermöglichen - wurde dieses gelöst, in dem man auf den aktuellen Messwert 70 Grad (jede andere Zahl hätte es auch getan) addiert und dann vergleicht.

Der Beharrungszustand wird so nie erreicht; das Programm terminiert nicht. Der Probelauf wurde dann mit einem Programmabbruch <CTRL E> beendet. Funktioniert auch...!

PDV-SYSTEME

Beispiele

Kompressoren

Kompressor	Maschinennummer	Maschinen- HD	Erfassungszeit	Zeitpunkt der letzten Druckprobe	Status der letzten Druckprobe	Zeitpunkt des letzten Probebaus
CF1280	deg		16.07.2003 09:49:07			
CF128	672431 (we)		23.07.2003 00:00:00			
CF128	de		02.08.2003 15:22:00			02.08.2003 00:00:00

Kompressor

Eigenschaften:

Kategorie	Kompressortyp
Übergeordnet	CF1280
Maschinennummer	Maschinennummer HD
Site	
Anlagennummer	
Registrierungsnummer	<input type="checkbox"/> Druckprobe erforderlich
Auftragsnummer	
Erfassungszeit	16.07.2003 09:49:07
Zeitpunkt des letzten Probebaus	
Zeitpunkt der letzten Druckprobe	
Status des letzten Probebaus	
Status der letzten Druckprobe	

APAM - Auswertung
APAK Administration
PDV-SYSTEME Support

PDV-SYSTEME

Beispiele

Systemprotokoll

Zeit	Kategorie	Meldung
17.07.2003 13:57:36	Information	ORA-04091: table APAK.ERPROBUNGSVORGABE is mutating, trigger/function may not see it
17.07.2003 10:42:19	Fehler	ORA-04091: table APAK.ERPROBUNGSVORGABE is mutating, trigger/function may not see it
17.07.2003 10:43:05	Fehler	ORA-04091: table APAK.ERPROBUNGSVORGABE is mutating, trigger/function may not see it
17.07.2003 10:43:09	Fehler	ORA-04091: table APAK.ERPROBUNGSVORGABE is mutating, trigger/function may not see it
20.07.2003 12:57:23	Information	Beginn des Imports
20.07.2003 12:57:26	Information	Ende des Imports
20.07.2003 13:00:08	Information	Beginn des Imports
20.07.2003 16:54:03	Information	Ende des Imports
07.08.2003 17:00:29	Information	Beginn des Imports
13.08.2003 09:02:30	Information	Beginn des Imports
20.08.2003 15:00:27	Fehler	ORA-04091: table APAK.ERPROBUNGSVORGABE is mutating, trigger/function may not see it
20.08.2003 15:00:48	Fehler	ORA-04091: table APAK.ERPROBUNGSVORGABE is mutating, trigger/function may not see it
21.08.2003 09:53:41	Information	Ende des Imports
21.08.2003 09:53:44	Information	Stop-Flag ist gesetzt, Import wird nicht gestartet
02.09.2003 00:00:00	Information	Uhrzeit 0:00:00

Allgemein

Eigenschaften:

Zeitpunkt	Kategorie
16.07.2003 13:57:36	Information
Element	Modul
CFD = IMPORT	APAK Import
Funktion	Phase
ImportDate@bearbeiten	

Meldungstext:
Ende des Bearbeitungslaufs, Dauer = 0:23 min, gespeicherte Merkmale = 596, fehlerhafte Merkmale = 0

APAM - Auswertung
PDV-SYSTEME Support

PDV-SYSTEME

