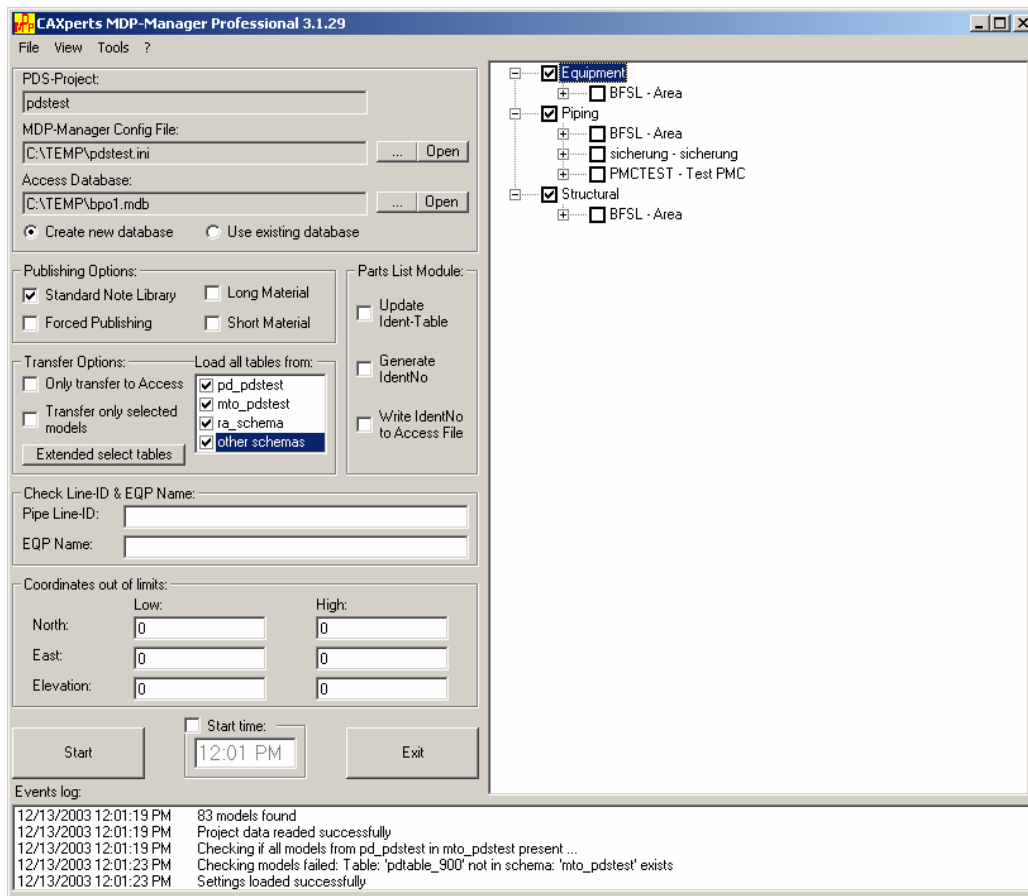


# CAXperts MDP-Manager (Basic/Professional)



CAXperts GmbH  
Telefon +49 (0)89 96 97 72 0

Freisinger Straße 1  
Fax +49 (0)89 96 97 72 11  
[www.caxperts.de](http://www.caxperts.de)

85737 Ismaning  
E-Mail [info@caxperts.de](mailto:info@caxperts.de)

# Inhaltsverzeichnis

<b>1</b>	<b>Beschreibung der Software</b>	<b>5</b>
<b>2</b>	<b>Hauptfenster</b>	<b>6</b>
2.1	PDS-Project	6
2.2	MDP-Manager Config-File	6
2.2.1	Oracle-Benutzer und Passwort	6
2.3	Projektstruktur	7
2.4	Access Database	8
2.5	Publishing Options	8
2.5.1	Forced Publishing	8
2.6	Transfer Options	9
2.6.1	Only Transfer to Access	9
2.6.2	Transfer only selected models	9
2.6.3	Load all tables from	9
2.6.4	Extended Select Tables	10
2.7	Quality Scripts	11
2.7.1	Check Line-ID & EQP Name	11
2.7.2	Coordinates out of Limit	12
2.8	Start Time	12
2.9	Events log	12
<b>3</b>	<b>Menü</b>	<b>13</b>
3.1	File	13
3.1.1	Configuration Wizard	13
3.1.2	Open Config file	13
3.1.3	Save Config file as	13
3.1.4	Create RIS dump	13
3.1.5	Transfer from RIS dump	13
3.1.6	Apply functions on Access Database	13
3.1.7	History	13

3.1.8	Exit .....	13
<b>3.2</b>	<b>View .....</b>	<b>14</b>
3.2.1	View MDP-Errors.....	14
3.2.2	Project Control.....	15
<b>3.3</b>	<b>Tools .....</b>	<b>16</b>
3.3.1	BadLines-Modul .....	16
3.3.2	DeleteModels-Modul.....	17
3.3.3	PartsList-Modul .....	18
3.3.4	.....	21
3.3.5	VerifyData-Modul.....	21
3.3.6	WriteToPDS-Modul.....	25
3.3.7	Schedule-Modul .....	27
3.3.8	Integrate with Batch Isogen Processing .....	27
<b>3.4</b>	<b>Hilfe.....</b>	<b>27</b>
3.4.1	Help .....	27
3.4.2	MDP-Manager on the Web.....	27
3.4.3	E-Mail Technical Support .....	27
3.4.4	Update License .....	28
3.4.5	About.....	28
<b>4</b>	<b>System-Administration .....</b>	<b>28</b>
4.1	Vorbereitungen für den MDP .....	28
<b>5</b>	<b>Batch Modus.....</b>	<b>29</b>
5.1	Daten in Access Datenbank entladen .....	29
5.2	Daten zurück in PDS schreiben.....	29
<b>6</b>	<b>Anhang.....</b>	<b>30</b>
6.1	Beschreibung der Konfigurationsdatei .....	30
6.1.1	Hauptkonfigurationsdatei.....	30
6.2	Projekt – Konfigurationsdatei.....	32
6.2.1	[MDPmgr] .....	32
6.2.2	[LoadTable] .....	33
6.2.3	[LoadAllTablesFromSchema] .....	34
6.2.4	[Quality] .....	35
6.2.5	[CustomIndex] .....	35
6.2.6	[SQL] .....	36

6.2.7	[Query].....	36
6.2.8	[ProjectControl].....	36
6.2.9	[RunBefore] .....	37
6.2.10	[RunAfter] .....	37
6.2.11	[Oracle].....	37
6.2.12	[ColDescr].....	37
6.2.13	[ColCopy].....	38
6.2.14	[Relation] .....	38
6.2.15	[Decode] .....	38
6.2.16	[DecodeSteel].....	38
6.2.17	[ReplacePath].....	39
6.2.18	[SchemasPass] .....	40
6.2.19	[Export2Excel] .....	40
6.2.20	[PartsListModule].....	40
6.2.21	[IdentAlloc].....	41
6.2.22	[MatReq].....	41
6.2.23	[WriteToPDS] .....	42
6.2.24	[WriteToPDSAllowUpdateTables].....	42
6.2.25	[WriteToPDSAllowUpdateColumns] .....	42
6.2.26	[Abfrage1].....	43
6.2.27	[VerifyDataRules].....	43

# 1 Beschreibung der Software

Intergraph liefert seit PDS 6.4 den Material Data Publisher mit PDS aus. Damit ist es erstmals möglich, fast alle Daten (aus Datenbank, Libraries und Designfiles), die in PDS vorhanden sind, in eine einzige Datenbank zu schreiben. Dies schafft ganz neue Möglichkeiten der Datenauswertung.

Mit dem MDP-Manager werden die Daten erst richtig nutzbar gemacht. Er bietet folgende Vorteile:

- Schnelle Extraktion in Microsoft Access Datenbank.
- Automatisches Dekodieren der Standard-Note-Library, 2D-Daten und Structural (Steel)-Daten.
- Qualitätsprüfung.
- Datum und Zeit werden in ein lesbare Format umgewandelt.
- Zollnennweiten werden entschlüsselt.
- Erstellung und Verwaltung von Stücklisten, Mehrerung/Minderung von Bauteilen.
- Leichte und effiziente Datenänderungen des Design-Schemas.
- Integrierung mit Batch Isogen Processing (Automatische Änderung der Daten in Design-Schemas VOR Generierung der Files).
- Models können direkt aus dem „MTO\_“-Schema gelöscht werden.
- Daten werden praxisgerecht indiziert (wesentlich schnellere Abfragen)
- Alle Felder können überprüft werden, ob die firmenspezifischen Eingaberegeln eingehalten wurden.
- Keine aufwändige und fehleranfällige Konfiguration von ODBC notwendig.
- Durchgängiges Schulungskonzept mit vielen Videos. So sind die Daten von jedermann auch ohne PDS-Kenntnisse anzuwenden.
- Daten können an Kunden, Lieferanten und Baustelle per Email versendet werden. Diese können Auswertungen leicht und schnell selbst erstellen.

## 2 Hauptfenster

### 2.1 PDS-Projekt

Ihr aktuelles PDS-Projekt wird auf Basis der Informationen aus dem Config-File (\*.ini) in diesem Feld angezeigt.

### 2.2 MDP-Manager Config-File

In dem Config-File sind alle notwendigen projektspezifischen Einstellungen wie z.B. Projektname gespeichert.

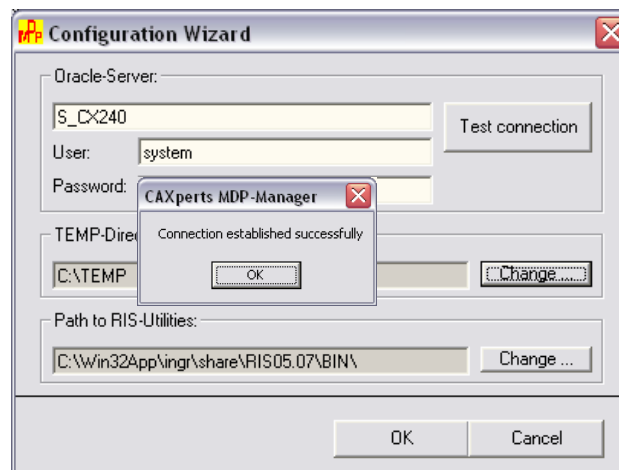
Nach dem Start der Software muss der projektspezifische Config-File entweder über File-Open-Dialog oder mit Hilfe von Drag&Drop ausgewählt werden. Man kann auch den Config-File direkt aus dem MDP-Manager-Fenster auf die Festplatte mit Drag&Drop kopieren.



Beim Klick auf den Open-Button kann der ausgewählte Config-File zum Editieren in einem Text-Editor geöffnet werden.

#### 2.2.1 Oracle-Benutzer und Passwort

Um Zugriff zur Oracle-Datenbank zu erhalten und diese Informationen auch nachhaltig ablegen zu können, gibt es in den Konfigurationsdateien des MDP Managers das Benutzernamen- und Passwortfeld. Dieses wird angelegt, wenn die Projektbearbeitung im 'Custom-Mode' des Configuration Wizards in 'Advanced Settings' erfolgt. Mit 'Test connection' kann die Einstellung getestet werden.



Es ist so möglich, für jedes Projekt eine eigene Server-, User- und Passwortdefinition zu erstellen und über ein 128Bit-Verschlüsselungsverfahren in der INI-Datei zu speichern. Somit ist eine hohe Sicherheit gewährleistet und dabei eine größtmögliche Flexibilität in der Datenbank- und Projektverwaltung gegeben. Es kann für jedes Projekt eine eigene Datenbankzugriffsdefinition erstellt werden. Siehe hierzu auch Punkt 6.2.11.

#### Projekt-INI-Konfiguration

```
[Oracle]
```

```
User=system
```

```
Password=edacmncmlpcclp
```

Der Eintrag des Servernamens unter [MDPmgr]

```
OracleServer=S_CX240
```

gibt den Hostname des Datenbankservers an.

Ist der Eintrag [Oracle] nicht in der Projekt-INI-Datei vorhanden, wird der Eintrag in der MDPmgr.ini gesucht, dieser kann von einer Projektdatei kopiert werden, da dieser nicht durch den Configuration Wizard erstellt wird. Dieser gilt als dann als Standard-Wert für alle Projekte ohne Eintrag [Oracle]. Ist in beiden Dateien (projekt.ini und mdpmgr.ini) kein Oracle-Eintrag vorhanden, gilt der projektspezifische Eintrag [SchemasPass].

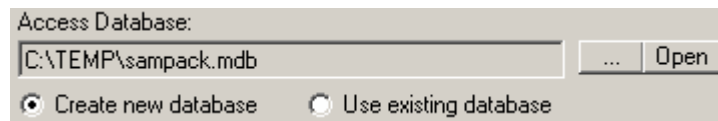
## 2.3 Projektstruktur

Im rechten Teil des Hauptfensters wird die Projektstruktur im Form eines Baumes angezeigt. Hier kann man Disziplinen, Design-Areas und Modelle auswählen, die extrahiert werden sollen.

Modelle, die rot markiert sind, fehlen im „mto“-Schema, existieren aber im „pd“-Schema. Diese können nicht ohne weiteres in die Access-Datenbank transferiert werden.

Um die fehlenden Modelle aus dem „pd“-Schema zu extrahieren, muss man sie durch Setzen eines Hakens in das rechteckige Feld davor aktivieren und den MDP-Manager mit ausgeschalteter Option „Only Transfer to Access“ starten.

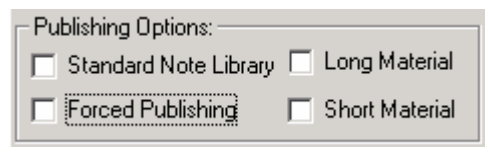
## 2.4 Access Database



Definiert den Access-File (\*.mdb), in den die Daten extrahiert werden. Sie können dabei einen bestehenden File auswählen („Use existing database“) oder einen neuen erzeugen („Create new database“).

Wenn im MDP-Manager Config-File „Access Seed Datenbank“ definiert ist, wird es nur bei „Create new database“ benutzt.

## 2.5 Publishing Options



In dem Bereich „Publishing Options“ wird angegeben, ob die Libraries mit extrahiert werden sollen oder nicht. Options sind nur dann aktiv, wenn der Schalter „Only Transfer to Access“ ausgeschaltet ist.

### 2.5.1 Forced Publishing

Der Schalter „Forced Publishing“ gibt an, ob die Daten erst komplett aus der Datenbank („mto\_“-Schema) gelöscht werden sollen, um dann komplett neu extrahiert zu werden ( Forced Publishing), oder ob nur die Änderungen berücksichtigt werden sollen ( Forced Publishing).

**Achtung:** Mit der Option Forced ( Forced Publishing) kann der Lauf bei großen Projekten mehrere Tage dauern.

## 2.6 Transfer Options

### 2.6.1 Only Transfer to Access

Ist der Schalter „Only Transfer to Access“ gesetzt, werden die Daten direkt aus dem bestehenden „mto\_“-Schema in die Access-Datenbank geladen.

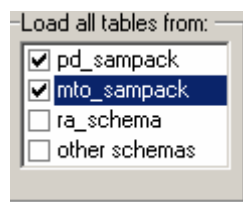
Bei nicht gesetztem Schalter wird vorher noch der Intergraph MDP durchgeführt und ein Update auf das „mto\_“-Schema ausgeführt.

### 2.6.2 Transfer only selected models

Wenn der Schalter „Transfer only selected Models“ gesetzt ist, werden nur ausgewählte Modelle aus dem „mto\_“-Schema in die Access-Datenbank geladen.

Andernfalls werden alle Daten aus dem „mto\_“-Schema geladen.

### 2.6.3 Load all tables from



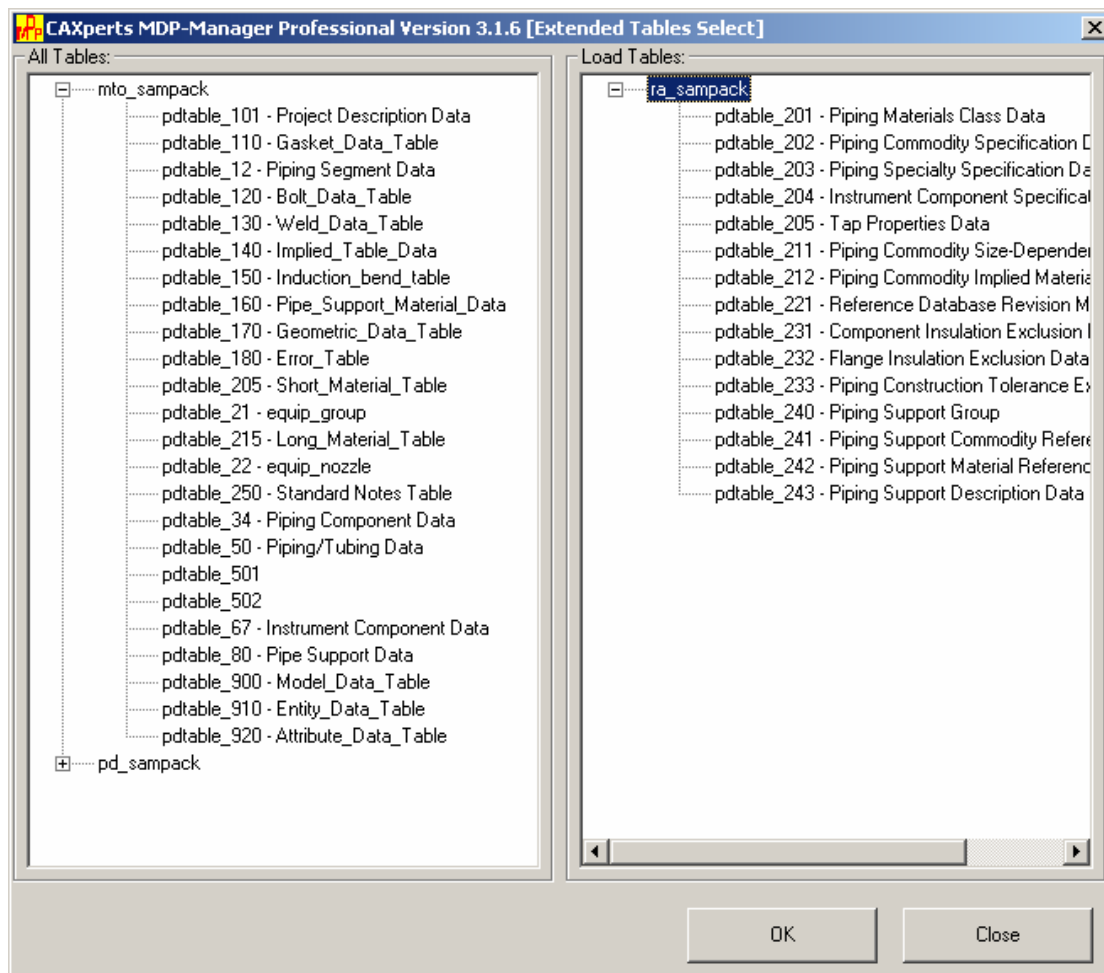
Es werden alle Tabellen aus den selektierten Schemen des gesamten Projekts geladen.

„other schemas“ – alle andere Schemen des Projektes, außer bereits aufgelisteten und „dd\_“-Schema.

Wenn im Fenster „Load all tables from“ ein Schema ausgewählt ist, werden alle Tabellen aus diesem Schema in die Access-Datenbank geladen. Mit dem „Extended Select Table“-Button ausgewählte Tabellen werden ignoriert und nicht geladen.

## 2.6.4 Extended Select Tables

Durch Klick auf den „Extended Select Table“-Button kann man zu ladende Tabellen aus der Liste aller Tabellen aus allen Schemen (außer „dd“-Schema) des Projektes auswählen.



Alle zu ladenden Tabellen des Schemas stehen im rechten Teil des Fensters. Alle Tabellen, die geladen werden können, sind links dargestellt.

Man kann die gewünschte Tabelle (oder sogar komplette Schemen) auswählen und auf die andere Seite des Fensters ziehen und loslassen (Drag&Drop-Methode).

Durch Klick auf den OK-Button werden alle Änderungen gespeichert und das Fenster „Extended Select Tables“ geschlossen.

Der Close-Button schließt das Fenster „Extended Select Tables“ ohne die vorgenommenen Änderungen zu speichern.

## 2.7 Quality Scripts

### 2.7.1 Check Line-ID & EQP Name

Check Line-ID & EQP Name:	
Pipe Line-ID:	cc-XXXnnn
EQP Name:	cccn

Durch diese Option werden Leitungen und Equipments gefunden, deren Namen (LineID) nicht den eingestellten Regeln entsprechen.

#### Regeln:

- n (kleiner Buchstabe) - beliebige Ziffer
- c (kleiner Buchstabe) - beliebiger Buchstabe
- Andere Zeichen (A, X, -, \_, .) werden wie geschrieben verwendet.

#### Beispiele:

nnn-ABC-ccc

Richtig	Falsch
<b>123-ABC-RKL</b>	12-ABC-RKL
007-ABC-NIX	007-ABX-NIX
471- -XXX	471-ABC-XX

Die fehlerhaften Leitungen können über das BadLines-Modul verwaltet werden.

## 2.7.2 Coordinates out of Limit

Durch diese Option werden Leitungen und Equipments gefunden, die außerhalb eines eingestellten Volumens sind. Dieses Volumen (durch zwei Raumpunkte) gibt die Außenbereiche der Anlage an.

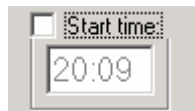
Auch hier werden die fehlerhaften Leitungen über das BadLines-Modul verwaltet.



	Low	High
North	0	9999999
East	0	9999999
Elevation	0	9999999

## 2.8 Start Time

Mit dieser Option kann man den Lauf des MDP-Managers zu einem späteren Zeitpunkt starten lassen.



1. Option aktivieren
2. Gewünschte Zeit einstellen
3. Auf den Start-Button klicken

Der MDP-Manager überprüft, ob alle Einstellungen korrekt sind, und wartet bis zur eingestellten Zeit. Nach Erreichen dieser eingestellten Zeit setzt der MDP-Manager seinen Lauf fort.

## 2.9 Events log

Zeigt 5 letzten Zeilen aus dem LOG-File. Beim Klicken öffnet die LOG-Datei in einem Text-Editor.

## 3 Menü

### 3.1 File

#### 3.1.1 Configuration Wizard

Erstellt eine neue oder ändert eine existierende Konfigurationsdatei. Siehe dazu auch den QuickStart Guide über die Verwendung des Configuration Wizards.

#### 3.1.2 Open Config file

Lädt die Einstellungen aus der angegebenen Konfigurationsdatei.

#### 3.1.3 Save Config file as

Speichert aktuelle Konfigurationsdatei unter angegebenem Namen.

#### 3.1.4 Create RIS dump

Entlädt „pd\_-“ und „mto\_-“ Schema des aktuellen Projektes, archiviert sie und speichert in die angegebene ZIP-Datei.

#### 3.1.5 Transfer from RIS dump

Lädt Daten aus dem RIS-Dump-File in das „mto\_-“ Schema des aktuellen Projektes, danach werden die Daten automatisch in die Access-Datenbank entladen.

#### 3.1.6 Apply functions on Access Database

Es werden alle Funktionen des MDP-Manager, die nach Entladen der Daten in Access Datenbank automatisch ausgeführt werden, wie Qualitätsprüfung, Dekodierung der Daten auf eine ausgewählte Access Datenbank angewendet.

#### 3.1.7 History

Zeigt die letzten 5 Konfigurationsdateien in chronologischer Reihenfolge an.

#### 3.1.8 Exit

Beendet die Software.

## 3.2 View

### 3.2.1 View MDP-Errors

Zeigt decodierte MDP-Fehlertabelle (phtable\_180). Neben den Navigationsfunktionen wie „Erster“, „Letzter“, „Vorheriger“ und „Nächster“ Datensatz, gibt es zusätzlich die Möglichkeit den Fehler direkt in Microstation zu betrachten, indem auf den Button „Open in Microstation“ geklickt wird.

422 records								
index_n	where_string	error_c	error_code_decode	model_code	area_name	model_no	line_id	cp1_np
819235	@#. 160, 67, 56, 51, 34, 1981, 51	410	Linkage Error	3VFUR1	b2_pcs2_50	pb209TA10	PCS2-V	15
819266	@#. 160, 67, 57, 51, 34, 1977, 51	410	Linkage Error	3VFUR1	e_bpo1test	inggk	PCS2-V	40
819266	@#. 160, 67, 57, 51, 34, 1977, 51	410	Linkage Error	3VFUR1	b2_pcs2_50	pb209TA10	PCS2-V	40
819255	@#. 160, 67, 58, 51, 34, 1868, 51	410	Linkage Error	3VFUR1	e_bpo1test	inggk	PCS2-V	25
819255	@#. 160, 67, 58, 51, 34, 1868, 51	410	Linkage Error	3VFUR1	b2_pcs2_50	pb209TA10	PCS2-V	25
825780	@#. 160, 67, 9, 58, 34, 492, 58	410	Linkage Error	I2WAY	b2_pcs6_01	pb204CA11	PCSE-E	50
825830	@#. 160, 67, 15, 58, 34, 607, 58	410	Linkage Error	I2WAY	b2_pcs6_01	pb204CA11	PCS6-V	100
825832	@#. 160, 67, 24, 58, 34, 737, 58	410	Linkage Error	3VFUR1	b2_pcs6_01	pb204CA11	PCS6-V	80
825845	@#. 160, 67, 29, 58, 67, 30, 58	410	Linkage Error	3VFUR1	b2_pcs6_01	pb204CA11	PCS6-V	15
825846	@#. 160, 67, 30, 58, 34, 829, 58	410	Linkage Error	3VFUR1	b2_pcs6_01	pb204CA11	PCS6-V	15
825844	@#. 160, 67, 31, 58, 67, 29, 58	410	Linkage Error	VCA	b2_pcs6_01	pb204CA11	PCS6-V	15
819310	pmc=25C1203NB903' gask_item_name='6	670	failed to find GASKE	4CYL	b2_pcs6_01	pb204PM11a	PCS6-V	25
819311	@#. 670, 67, 9, 62, 34, 296, 62	410	Linkage Error	4CYL	b2_pcs6_01	pb204PM11a	PCS6-V	25
819312	@#. 160, 67, 9, 62, 34, 296, 62	410	Linkage Error	4CYL	b2_pcs6_01	pb204PM11a	PCS6-V	25
810412	@#. 160, 67, 4, 64, 34, 598, 64	410	Linkage Error	I2WAY	b2_pcs6_01	pb204DT11a	PCS6-V	80
810424	@#. 160, 67, 12, 64, 34, 585, 64	410	Linkage Error	I2WAY	b2_pcs6_01	pb204DT11a	PCS6-V	80
810435	@#. 160, 67, 13, 64, 34, 578, 64	410	Linkage Error	I2WAY	b2_pcs6_01	pb204DT11a	PCS6-V	80
810442	@#. 160, 67, 14, 64, 34, 591, 64	410	Linkage Error	I2WAY	b2_pcs6_01	pb204DT11a	PCS6-V	80
810461	@#. 160, 67, 16, 64, 34, 625, 64	410	Linkage Error	I2WAY	b2_pcs6_01	pb204DT11a	PCS6-V	80
810460	@#. 160, 67, 16, 64, 34, 625, 64	410	Linkage Error	I2WAY	b2_pcs6_01	pb204DT11a	PCS6-V	80
784834	@#. 160, 67, 2, 65, 34, 41, 65	410	Linkage Error	I2WAY	b2_pcs6_01	pb204DT11b	PCS6-V	15
801030	pmc=25C1203NB903' gask_item_name='6	670	failed to find GASKE	IND	b2_pcs6_01	pb204WT11a	PCS6-V	25
801482	pmc=25C1203NB903' gask_item_name='6	670	failed to find GASKE	IND	b2_pcs6_01	pb204WT11a	PCS6-V	25
801331	pmc=25C1203NB903' gask_item_name='6	670	failed to find GASKE	IND	b2_pcs6_01	pb204WT11a	PCS6-V	25
801588	pmc=25C1203NB903' gask_item_name='6	670	failed to find GASKE	IND	b2_pcs6_01	pb204WT11a	PCS6-V	25
801365	pmc=25C1203NB903' gask_item_name='6	670	failed to find GASKE	I2WAY	b2_pcs6_01	pb204WT11a	PCS6-V	50
801366	@#. 670, 67, 34, 66, 34, 1477, 66	410	Linkage Error	I2WAY	b2_pcs6_01	pb204WT11a	PCS6-V	50
801367	@#. 160, 67, 34, 66, 34, 1477, 66	410	Linkage Error	I2WAY	b2_pcs6_01	pb204WT11a	PCS6-V	50
801130	pmc=25C1203NB903' gask_item_name='6	670	failed to find GASKE	IND	b2_pcs6_01	pb204WT11a	PCS6-V	25
801165	@#. 670, 67, 36, 66, 34, 1508, 66	410	Linkage Error	I2WAY	b2_pcs6_01	pb204WT11a	PCS6-V	50
801164	pmc=25C1203NB903' gask_item_name='6	670	failed to find GASKE	I2WAY	b2_pcs6_01	pb204WT11a	PCS6-V	50
801166	@#. 160, 67, 36, 66, 34, 1508, 66	410	Linkage Error	I2WAY	b2_pcs6_01	pb204WT11a	PCS6-V	50
801193	@#. 160, 67, 50, 66, 34, 2048, 66	410	Linkage Error	I2WAY	b2_pcs6_01	pb204WT11a	PCS6-V	25
801191	pmc=25C1203NB903' gask_item_name='6	670	failed to find GASKE	I2WAY	b2_pcs6_01	pb204WT11a	PCS6-V	25
801192	@#. 670, 67, 50, 66, 34, 2048, 66	410	Linkage Error	I2WAY	b2_pcs6_01	pb204WT11a	PCS6-V	25
801388	@#. 160, 67, 51, 66, 34, 2107, 66	410	Linkage Error	I2WAY	b2_pcs6_01	pb204WT11a	PCS6-V	25

### 3.2.2 Project Control

Zeigt Auswahl-Abfragen aus MDP-Manager Config-File. Wenn in diesen SQL-Abfragen als erste Spalte „line\_id“ steht, wird beim rechten Mausklick in Project Control Fenster PopUp-Menü mit allen DGN-Zeichnungen dieser Leitung angezeigt. Man kann diese mit SmartPlantReview oder Microstation öffnen.

line_id	model_no	DN	ExtraktionsC	PlazierungsL
**88907C	p_burnersu	25	2/5/2003 11	9/30/2002 11
**88907C	p_burnersu	25	2/5/2003 11	9/30/2002 11
**88907C	p_burnersu	25	2/5/2003 11	9/30/2002 11
**FT	e_silencer	50	2/5/2003 11	11/26/2002 11
**FT	TESTRVZ	50	2/5/2003 11	11/26/2002 11
67001C	p_burnersu	40	2/5/2003 11	10/31/2002 11
67001C	p_burnersu	40	2/5/2003 11	10/31/2002 11
67001C	p_burnersu	25	2/5/2003 11	10/31/2002 11
67001C	p_burnersu	25	2/5/2003 11	10/31/2002 11
67001C	p_burnersu	25	2/5/2003 11	10/31/2002 11
67001C	p_burnersu	40	2/5/2003 11	10/31/2002 11

Open in SmartPlantReview \\NBOOK01\pds\sk\sampack\drv\p\_burnersu

Beispiel:

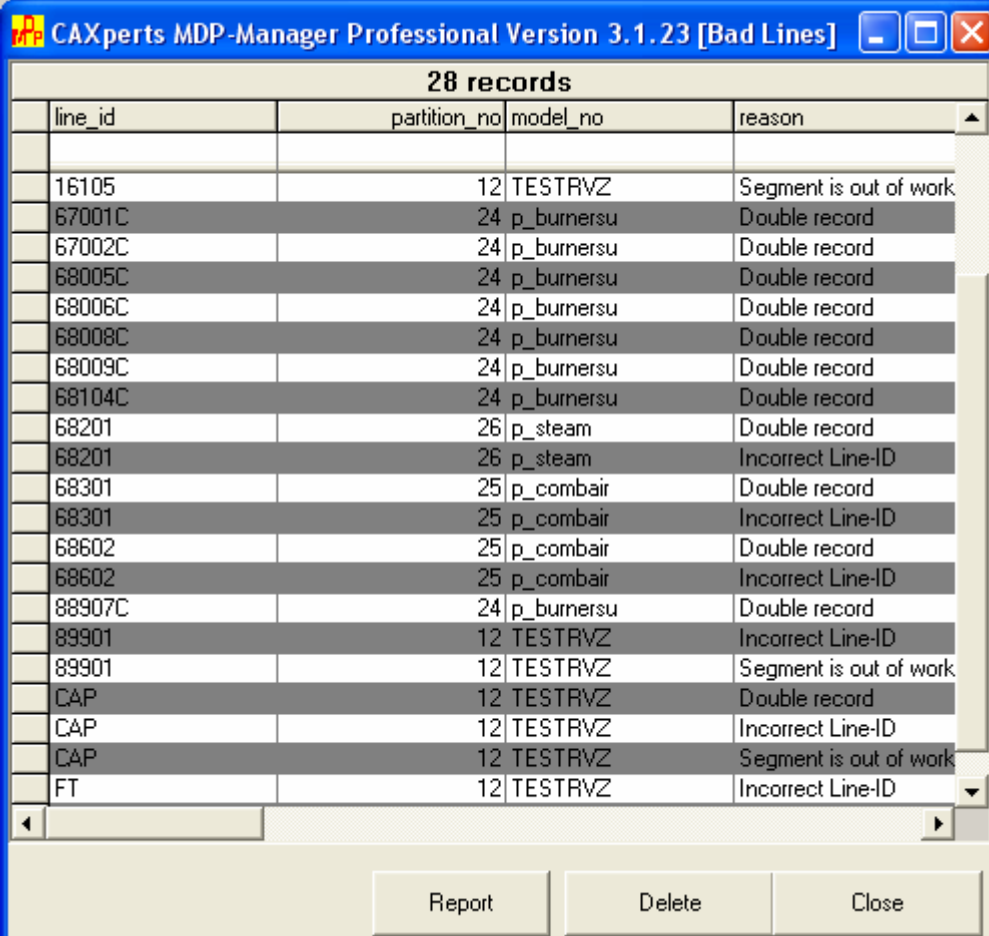
[ProjectControl]

LineList=SELECT line\_id FROM ptable\_12

## 3.3 Tools

### 3.3.1 BadLines-Modul

Verwaltung der fehlerhaften Leitungen in der Access-Datenbank.



28 records			
line_id	partition_no	model_no	reason
16105	12	TESTRVZ	Segment is out of work
67001C	24	p_burnersu	Double record
67002C	24	p_burnersu	Double record
68005C	24	p_burnersu	Double record
68006C	24	p_burnersu	Double record
68008C	24	p_burnersu	Double record
68009C	24	p_burnersu	Double record
68104C	24	p_burnersu	Double record
68201	26	p_steam	Double record
68201	26	p_steam	Incorrect Line-ID
68301	25	p_combair	Double record
68301	25	p_combair	Incorrect Line-ID
68602	25	p_combair	Double record
68602	25	p_combair	Incorrect Line-ID
88907C	24	p_burnersu	Double record
89901	12	TESTRVZ	Incorrect Line-ID
89901	12	TESTRVZ	Segment is out of work
CAP	12	TESTRVZ	Double record
CAP	12	TESTRVZ	Incorrect Line-ID
CAP	12	TESTRVZ	Segment is out of work
FT	12	TESTRVZ	Incorrect Line-ID

Nach der Generierung der Access-Datenbank werden Quality Scripts ausgeführt, die aus folgenden Teilen bestehen:

- Überprüfung des Leitungsnamens und des Equipmentnamens nach bestimmten Regeln; (Konfigurierbar über Check Line-ID & EQP Name)
- Überprüfung, ob alle Komponenten der Leitung in dem bestimmten Arbeitsbereich liegen; (Konfigurierbar über Coordinates Out of Limits)
- Überprüfung, ob keine doppelte Datensätze in der Datenbank vorhanden sind.

Alle fehlerhafte Leitungen werden in der Tabelle „bad\_lines“ gespeichert und können mit dem BadLines-Modul verwaltet werden.

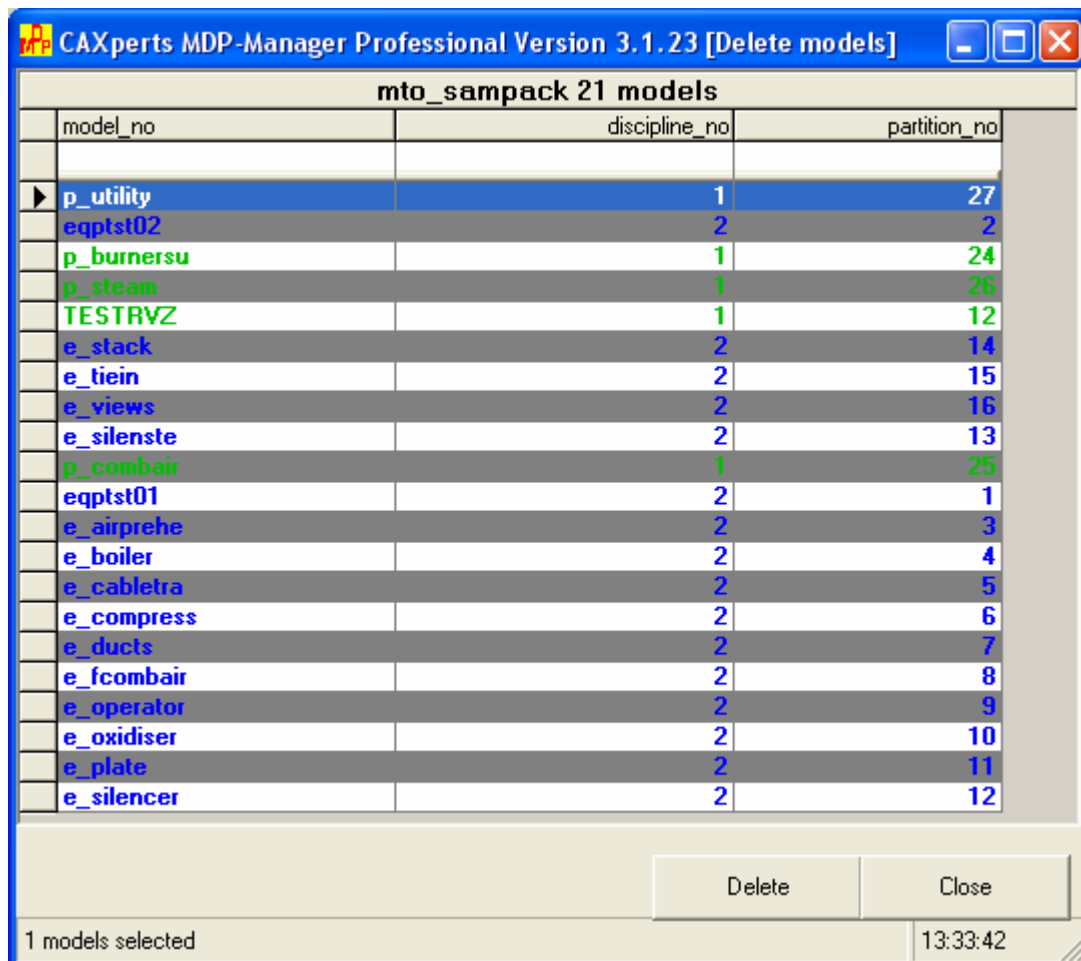
Es können bestimmte Leitungen aus der Access-Datenbank gelöscht werden indem eine oder mehrere Leitungen selektiert und durch Klick auf Delete-Button bestätigt werden. Report-Button erzeugt ein Bericht, der gedruckt werden kann.

Durch Klick auf den Spaltennamen können die Werte sortiert werden. Es können nur die Ergebnisse angezeigt werden, die bestimmte Kriterien erfüllen, indem ein Filter gesetzt wird.

Um ein Filter zu definieren, muss in die entsprechende Spalte der ersten Zeile ein Kriterium eingetragen werden. Dabei können gleichzeitig in mehreren Spalten Filterkriterien gesetzt sein.

### 3.3.2 DeleteModels-Modul

Verwaltung der Modelle im „mto\_“-Schema des PDS-Projektes.



Es können ein oder mehrere Modelle aus dem „mto\_“-Schema des PDS-Projektes gelöscht werden, ohne noch einmal den MDP-Job zu starten, indem ein oder mehrere Modelle ausgewählt werden und mit Klick auf Delete-Button bestätigt wird.

Blaue Datensätze sind Equipment-Modelle

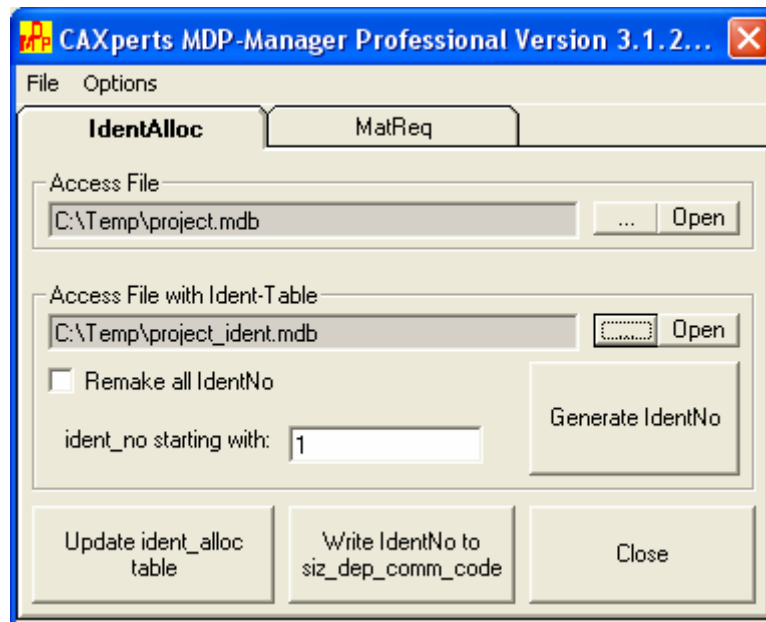
Grüne Datensätze sind Piping-Modelle

Rote Datensätze sind Structural-Modelle

Durch Klick auf den Spaltennamen können die Werte sortiert werden. Es können nur die Ergebnisse angezeigt werden, die bestimmte Kriterien erfüllen, indem ein Filter gesetzt wird.

Um einen Filter zu definieren, muss in die entsprechende Spalte der ersten Zeile ein Kriterium eingetragen werden. Es können gleichzeitig in mehreren Spalten Filterkriterien gesetzt sein.

### 3.3.3 PartsList-Modul



Das PartsList-Modul dient zur Erstellung und Verwaltung von Stücklisten.

Es wird eine Tabelle (ident\_alloc) mit sämtlichen Bauteilen des Projekts erstellt. Für jedes Bauteil wird automatisch oder manuell eine eindeutige Nummer (Ident-Nummer) vergeben. Anhand dieser Nummer können Stücklisten zusammengestellt und die Mehrung/Minderung von Bauteilen festgestellt werden.

Wenn neue Bauteile ins Projekt hinzugefügt werden, wird die Tabelle (ident\_alloc) mit allen Bauteilen aktualisiert und weitere Nummer werden für neue Bauteile vergeben.

#### **Ablauf:**

1. Eine aktuelle Access-Datenbank mit den Projektdaten auswählen

2. Access-Datenbank für die Tabelle (ident\_alloc) mit Ident-Nummer auswählen

In dieser Datenbank wird eine Tabelle mit sämtlichen Bauteilen aus dem Projekt erstellt. Die Tabelle enthält auch für jedes Bauteil eine eindeutige Nummer. Diese Datenbank muss nach der Erstellung gesichert werden, damit bei dem nächsten Lauf eine bereits existierende Datenbank benutzt werden kann.

3. „Update ident\_alloc table“-Button Klick

Es wird eine neue Tabelle mit allen Bauteilen aus dem Projekt erstellt. Falls die Tabelle bereits existiert, werden die fehlenden Bauteile hinzugefügt.

Die erstellte bzw. aktualisierte Tabelle „ident\_alloc“ enthält alle Bauteile aus dem Projekt. Diese Tabelle kann erweitert werden, indem eigene Spalte hinzugefügt werden.

4. „Generate IdentNo“-Button klicken

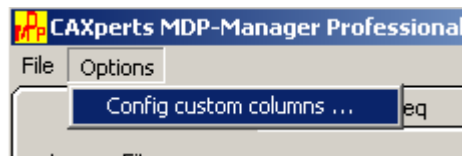
Es werden für alle Bauteile, denen noch keine Nummer zugeordnet wurde, Nummern vergeben. Die erste Nummer wird aus dem Feld „starting IdentNo with“ ausgelesen. Man kann auch die Nummer selbst vergeben, indem man direkt in der Tabelle „ident\_alloc“ in die Spalte „ident\_no“ die entsprechende Nummer einträgt (Alphanumerischer Wert).

Wenn man alle Nummern neu vergeben will, muss man den Schalter „Remake all IdentNo“ setzen. Dabei werden alle alten Ident-Nummern überschrieben.

5. „Write IdentNo to siz\_dep\_comm\_code“-Button Klick

Die vergebenen Nummern werden zurück in die Datenbank mit den Projektdaten geschrieben.

Es können eigene Spalten in die „ident\_alloc“-Tabelle hinzugefügt werden indem man Menü Options->Config custom columns aufruft:



Für jede eigene Spalte, die zur Tabelle „ident\_alloc“ hinzugefügt werden muss, ist eine Zeile vorgesehen.

1. Den Namen für neue Spalte definieren (col\_name).

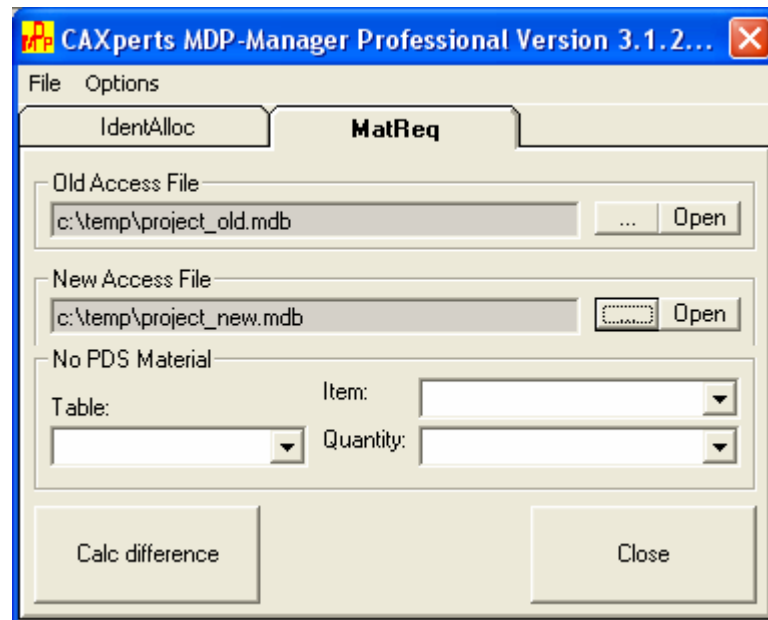
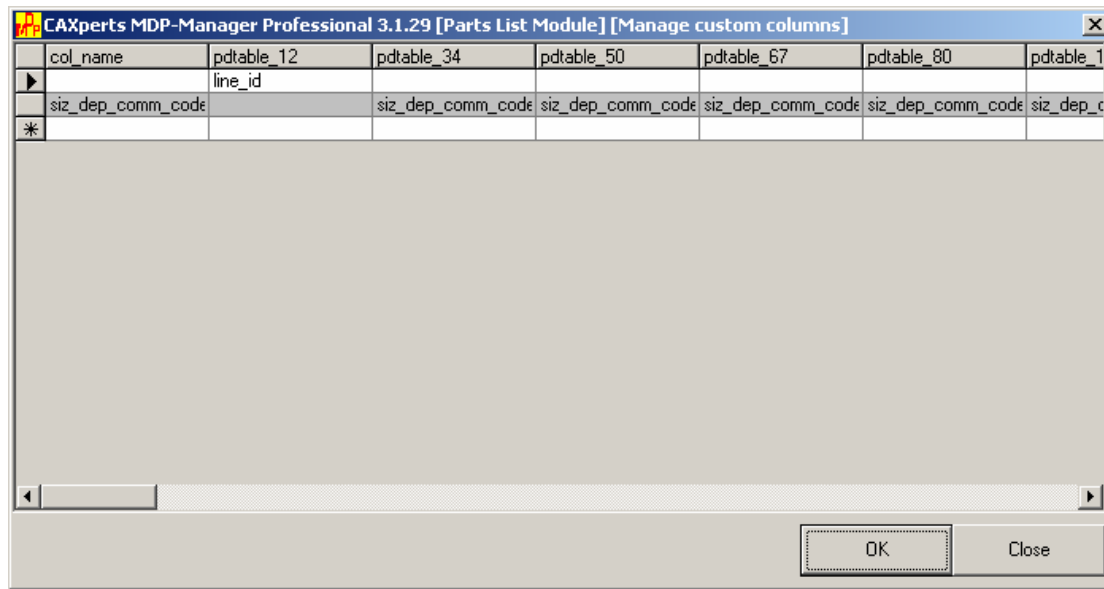
Man kann auch keinen Namen definieren.

2. Für jede Tabelle eine Spalte auswählen, aus der die Daten in die Tabelle „ident\_alloc“ hinzugefügt werden müssen.

Die Daten können entweder aus pdtable\_12 oder aus anderen Tabelle ausgewählt werden.

Es müssen nicht für alle Tabellen die Spalten definiert werden.

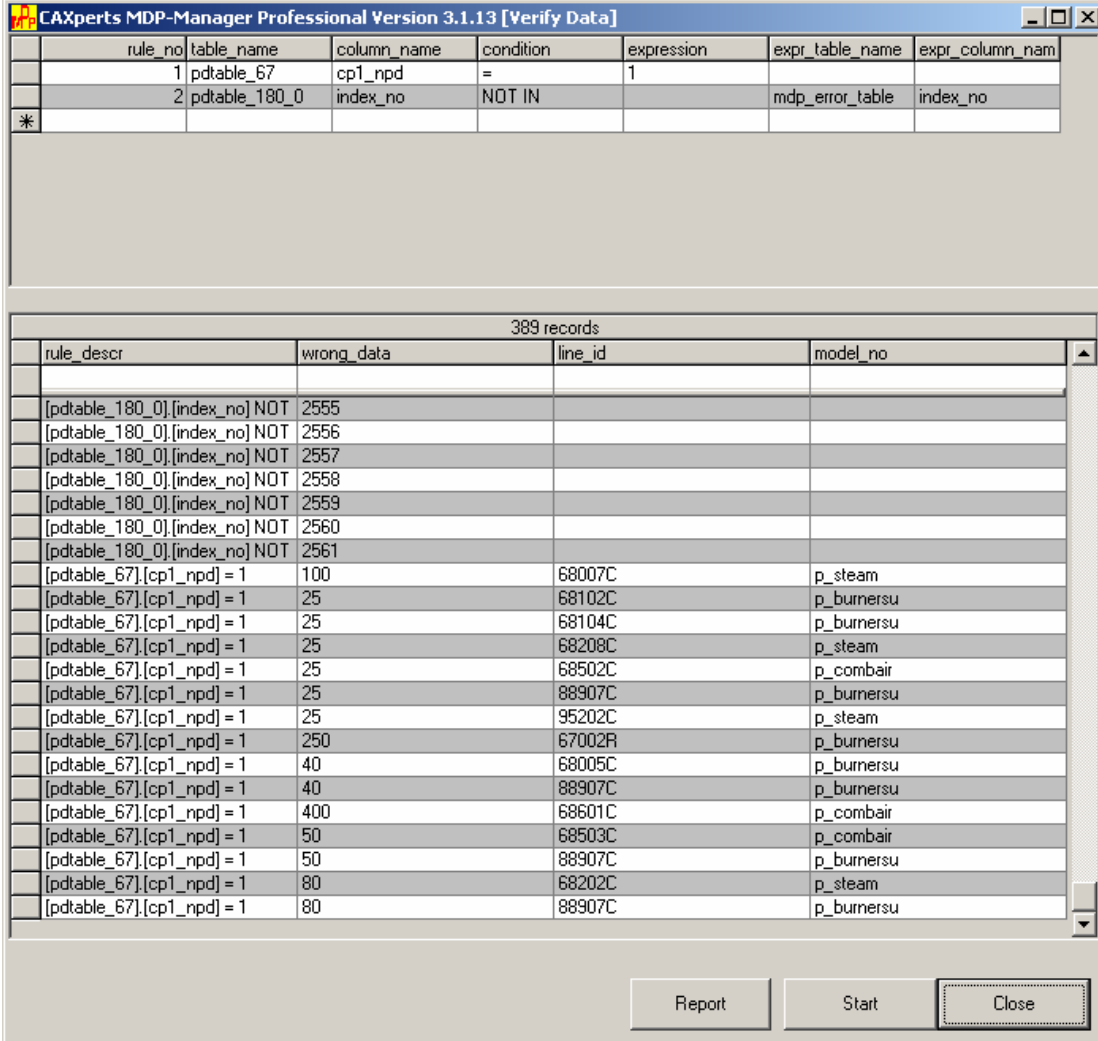
3. Falls weitere Spalten nötig sind, kehre zum Schritt 1 zurück.
4. Auf „OK“-Button drücken, damit die Änderungen gespeichert werden.  
„Close“-Button schließt das Fenster ohne Änderungen zu speichern.



### 3.3.4

Mit Hilfe der Funktion MatReq werden zwei Access-Dateien miteinander verglichen und dabei die Mehrung oder Minderung über den Button „Calc difference“ in eine Differenzliste geladen. Dadurch können die vorhergehend erzeugten Bauteiltabellen mit ihren Massenveränderungen ausgewertet werden.

### 3.3.5 VerifyData-Modul



rule_no	table_name	column_name	condition	expression	expr_table_name	expr_column_name
1	pdtable_67	cp1_npd	=	1		
2	pdtable_180_0	index_no	NOT IN		mdp_error_table	index_no

399 records			
rule_descr	wrong_data	line_id	model_no
[pdtable_180_0].[index_no] NOT	2555		
[pdtable_180_0].[index_no] NOT	2556		
[pdtable_180_0].[index_no] NOT	2557		
[pdtable_180_0].[index_no] NOT	2558		
[pdtable_180_0].[index_no] NOT	2559		
[pdtable_180_0].[index_no] NOT	2560		
[pdtable_180_0].[index_no] NOT	2561		
[pdtable_67].[cp1_npd] = 1	100	68007C	p_steam
[pdtable_67].[cp1_npd] = 1	25	68102C	p_burnersu
[pdtable_67].[cp1_npd] = 1	25	68104C	p_burnersu
[pdtable_67].[cp1_npd] = 1	25	68208C	p_steam
[pdtable_67].[cp1_npd] = 1	25	68502C	p_combair
[pdtable_67].[cp1_npd] = 1	25	88907C	p_burnersu
[pdtable_67].[cp1_npd] = 1	25	95202C	p_steam
[pdtable_67].[cp1_npd] = 1	250	67002F	p_burnersu
[pdtable_67].[cp1_npd] = 1	40	68005C	p_burnersu
[pdtable_67].[cp1_npd] = 1	40	88907C	p_burnersu
[pdtable_67].[cp1_npd] = 1	400	68601C	p_combair
[pdtable_67].[cp1_npd] = 1	50	68503C	p_combair
[pdtable_67].[cp1_npd] = 1	50	88907C	p_burnersu
[pdtable_67].[cp1_npd] = 1	80	68202C	p_steam
[pdtable_67].[cp1_npd] = 1	80	88907C	p_burnersu

Buttons: Report, Start, Close

Dies ist ein sehr mächtiges Werkzeug, mit dessen Hilfe leicht und effektiv Eingabefehler frühzeitig erkannt werden können.

Die Prüfung wird in folgender Reihenfolge durchgeführt:



1. Zu prüfende Tabelle wählen

rule_no	table_name	column_name	co
1	pdtable_67	cp1_npd	=
2	table_name		
*	pdtable_250		
	pdtable_34		
	pdtable_50		
	pdtable_501		
	pdtable_502		
	pdtable_67		
	pdtable_80		
	pdtable_900		
	pdtable_910		
	pdtable_920		
	Pipeliste		
	Rohrlänge		
	Surface per line		
	verify_data		
	Volume per Medium		
	Weidlist		

2. Spalte wählen

rule_no	table_name	column_name	condition	exp
1	pdtable_67	cp1_npd	=	1
2	pdtable_180_0	column_name		
*		cp1_npd		
		cp2_idx_12		
		cp2_npd		
		cp3_idx_12		
		cp3_npd		
		cp4_idx_12		
		cp4_npd		
		cp5_idx_12		
		cp5_npd		
		design_resp		
		dimension_a		
		dimension_b		
		dimension_c		
		empty_weight		
		fabrication_cat		
		fabrication_cat_decoded		

### 3. Logik auswählen


	rule_no	table_name	column_name	condition	expression	exp
	1	pdtable_67	cp1_npd	=	1	
	2	pdtable_180_0	index_no	condition		
				>		
				<		
				=		
				<>		
				LIKE		
				NOT LIKE		
				IN		
				NOT IN		

Falls man Logik: „IN“ oder „NOT IN“ ausgewählt hat, muss „expr\_table\_name“ und „expr\_column\_name“ gesetzt werden (s. Schritt 1 und 2). Bei dieser Logik können keine Werte in die Spalte „expression“ geschrieben werden.

Für die andere Logik muss „expression“ eingegeben werden.

expression	e
1	

4. Falls weitere Regeln nötig sind, kehre zum Schritt 1 zurück.

5.  zeigt die Ergebnisse auf dem Bildschirm an und schreibt sie in verify\_data-Tabelle.

6.  erzeugt einen Bericht, der gedruckt werden kann.

Beim Klicken auf dem Spaltennamen können die Werte sortiert werden. Man kann nur die Ergebnisse anzeigen, die bestimmte Kriterien erfüllen, indem ein Filter gesetzt wird. Um einen Filter zu definieren, muss in die entsprechende Spalte der ersten Zeile (s. roter Pfeil) ein Kriterium eingetragen werden. Es können gleichzeitig in den mehreren Spalten Filterkriterien gesetzt sein.

rule_descr	wrong_data	line_id	model_no
	25		p_§
[pdtable_67].[cp1_npd] = 1	25	68208C	p_steam
[pdtable_67].[cp1_npd] = 1	25	95202C	p_steam

Die Regeln definieren richtige Daten. Werte, die die Regeln NICHT erfüllen werden angezeigt.

Für die Prüfung von Textwerten steht eine sehr mächtige Prüffunktion zur Verfügung.

Entsprechung	Muster	Entsprechung (gibt True zurück)	Keine Entsprechung (gibt False zurück)
Mehrere Zeichen	a%a	aa, aBa, aBBBa	ABC
Mehrere Zeichen	%ab%	abc, AABB, Xab	aZb, bac
Sonderzeichen	a[%]a	A%a	Aaa
Mehrere Zeichen	ab%	abcdefg, abc	cab, aab
Einzelne Zeichen	A_a	aaa, a3a, aBa	ABBBa
Einzelne Ziffern	a#a	a0a, a1a, a2a	aaa, a10a
Zeichenbereich	[a-z]	f, p, j	2, &
Außerhalb des Bereichs	[!a-z]	9, &, %	b, a
Keine Ziffern	[!0-9]	A, a, &, ~	0, 1, 9
Kombination	a[!b-m]#	An9, az0, a99	abc, aj0

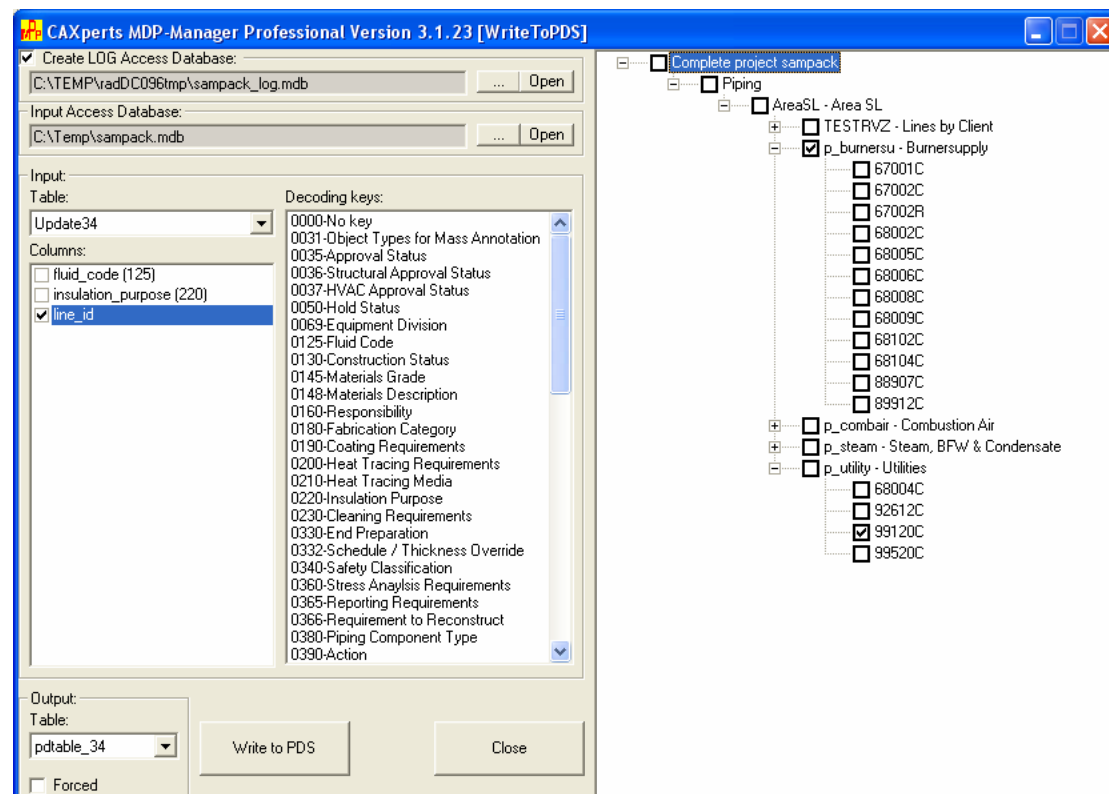
### 3.3.6 WriteToPDS-Modul

(Nur in Professional Version ausführbar)

WriteToPDS-Modul ermöglicht es, leicht und effizient Änderungen im Design-Schema des Projektes vorzunehmen.

**WriteToPDS-Modul kann Ihr PDS-Projekt zerstören. Machen Sie VOR Benutzung dieses Moduls eine Sicherungskopie Ihres PDS-Projektes.**

**CAXperts GmbH übernimmt keine Verantwortung für möglichen Datenverlust.**



#### Ablauf:

1. Eine Input Access-Datenbank auswählen. Enthält Tabelle(n) oder Abfrage(n) mit den Daten, die in PDS geschrieben werden.
2. Eine Input-Tabelle/Abfrage aus der Input-Access Datenbank auswählen. Alle Spaltennamen und Datentypen aus der Input-Tabelle/Abfrage müssen mit den Spaltennamen und Datentypen der Output-Tabelle übereinstimmen.

3. Input-Spalten entschlüsseln. Es können die Werte in den Spalten der Input-Tabelle/Abfrage entschlüsselt werden.

Zur Entschlüsselung der Spalte muss der entsprechende Schlüssel über die zu entschlüssende Spalte gezogen und losgelassen werden (Drag&Drop-Methode). Wenn man die Entschlüsselung löschen will, muss man den Schlüssel „0000-No key“ über die entsprechende Spalte ziehen und loslassen.

4. Kriterium-Spalten definieren. Kriterium-Spalten stellen Bedingungen dar, die beim Schreiben der Daten in PDS berücksichtigt werden. Daten aus Kriterium-Spalten werden nicht in PDS geschrieben.

Es können mehrere Kriterium-Spalten definiert sein, sind aber nicht zwingend notwendig.

5. Output-Tabelle auswählen. Tabelle aus dem Projekt Design-Schema, in welche die Daten geschrieben werden sollen.

6. Forced-Schalter

Wenn der Forced-Schalter ausgeschaltet ist, werden nur die Werte in PDS geschrieben, die vorher nicht existierten. (Alte Werte werden nicht überschrieben). Wenn der Forced-Schalter gesetzt ist, werden alle Werte in PDS geschrieben und dabei werden eventuell alte Daten überschrieben.

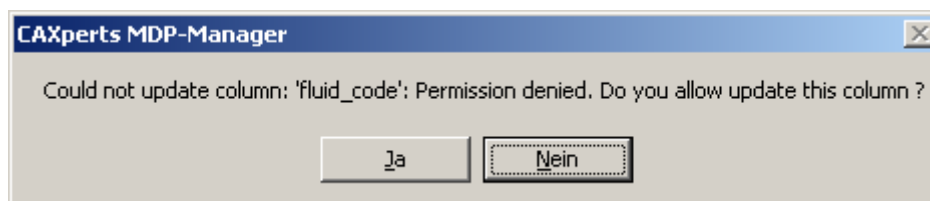
7. Änderungsbereich auswählen. Die Änderungen werden im kompletten Projekt, in den bestimmten Areas, Modellen oder sogar nur in den bestimmten Leitungen vorgenommen.

8. LOG einschalten und LOG-Access Datenbank auswählen. In der LOG-Access-Datenbank werden alle Änderungen im Projekt Design-Schema protokolliert.

9. WriteToPDS-Button klick. Die Daten werden in PDS geschrieben.

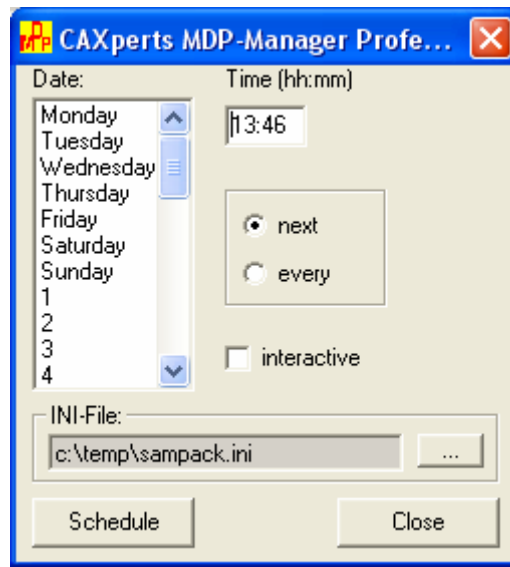
Standardmäßig darf WriteToPDS-Modull keine Daten in PDS ändern.

Man muss explizit jede Spalte freigeben indem man folgende Warnungsfrage mit „Ja“ beantwortet.



### 3.3.7 Schedule-Modul

Mit dem Taskplaner kann man die Ausführung von MDP-Manager für einen beliebigen Zeitpunkt planen. Es muss eine MDP-Manager Config-Datei ausgewählt werden, die zum ausgewählten Zeitpunkt dem MDP-Manager als Parameter übergeben wird.



### 3.3.8 Integrate with Batch Isogen Processing

Integriert MDP-Manager in Batch Isogen Processing. Dies ermöglicht vor Generierung der ISO-Files die Daten in Design-Schemas des Projektes zu aktualisieren bzw zu ändern mittels WriteToPDS-Moduls des MDP-Managers.

## 3.4 Hilfe

### 3.4.1 Help

Zeigt die Hilfe-Datei an.

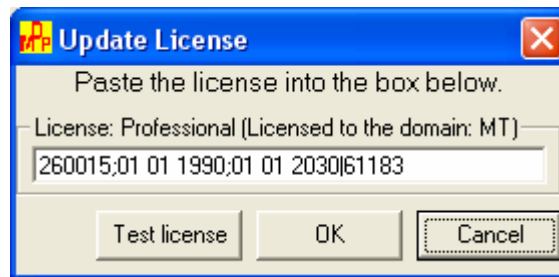
### 3.4.2 MDP-Manager on the Web

Öffnet die Homepage von MDP-Manager ([www.CAXperts.de](http://www.CAXperts.de)) mit dem Standard-Webbrowser.

### 3.4.3 E-Mail Technical Support

Generiert eine E-Mail für technische Support ([support@CAXperts.de](mailto:support@CAXperts.de)).

### 3.4.4 Update License



„Test license“-Button prüft eingetragene Lizenznummer.

„OK“-Button speichert die eingetragene Lizenznummer.

„Cancel“-schließt das Fenster ohne Änderungen zu speichern.

### 3.4.5 About

Zeigt ein Dialogfeld mit Informationen über MDP-Manager an, z. B. Version, Lizenzdaten.

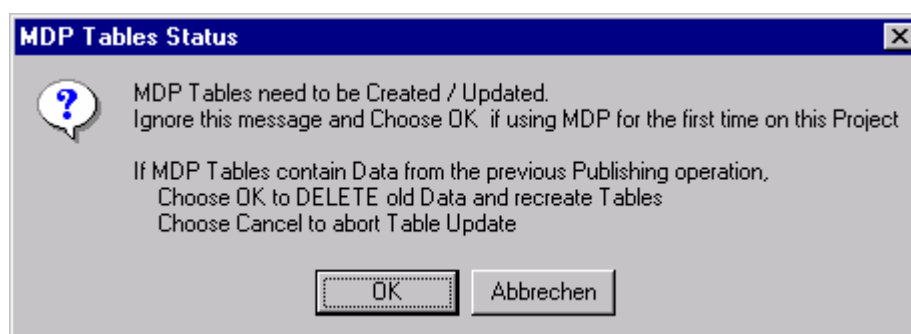
## 4 System-Administration

### 4.1 Vorbereitungen für den MDP

„mto-“-Datenbankuser und RIS-Schemen für das Projekt müssen angelegt sein. Vor dem ersten MDP-Lauf eines neuen Projektes müssen die Datenbank-Tabellen angelegt werden.

#### Ablauf:

1. Intergraph MDP starten
2. Projekt auswählen
3. Meldung mit OK bestätigen



4. MDP beenden

## 5 Batch Modus

Für den Benutzer, der Befehle im Batchbetrieb ausführt, müssen alle erforderlichen Umgebungsvariablen (RIS\_PARAMETERS, TEMP...) richtig gesetzt sein. Außerdem muss der Benutzer entsprechende Rechte haben.

### 5.1 Daten in Access Datenbank entladen

Um die Software im Batchbetrieb zu starten, muss MDP-Manager mit einem Parameter: Config-File aufgerufen werden.

Beispiel:

```
c:\path\mdpmgr.exe c:\path\project.ini
```

### 5.2 Daten zurück in PDS schreiben

Es ist auch möglich die Daten im Batchbetrieb zurück in PDS schreiben.

In diesem Fall muss MDP-Manager mit zwei Parameter aufgerufen werden:

1. Parameter: Der Ordner mit Config-Dateien oder der Pfad bis zu einem Config-File.;
2. Parameter: Isogen Batch Input-File;

Beispiel:

```
c:\path\mdpmgr.exe c:\path\to\config_files\  
c:\path\isobatch.txt
```

oder

```
c:\path\mdpmgr.exe c:\path\project.ini c:\path\isobatch.txt
```

Wenn der erste Parameter ein Ordner mit Config-Dateien ist, müssen Config-Dateien wie Projektname heißen (z.B. Die Einstellungen für PDS-Project „project1“ müssen in der Datei „project1.ini“ gespeichert sein).

## 6 Anhang

### 6.1 Beschreibung der Konfigurationsdatei

#### 6.1.1 Hauptkonfigurationsdatei

Die Hauptkonfigurationsdatei mdpmgr.ini muss im gleichen Verzeichnis wie mdpmgr.exe liegen.

##### **[MDP-Manager]**

License=1234567

Software-Lizenznummer.

##### **[PDTablesDescr]**

Kurze Beschreibung der Tabellen, die im „Extended Select Table“ - Fenster angezeigt werden.

Format: Tabellenname=Beschreibung

Beispiel:

[PDTableDescr]

phtable\_101=Project Description Data

##### **[MTOTablesDescr]**

s. Abschnitt [PDTableDescr]

##### **[RaTablesDescr]**

s. Abschnitt [PDTableDescr]

##### **[DecodedMDPErrors]**

Beschreibung der MDP-Fehler, die bei der Dekodierung der phtable\_180 benutzt wird. (s. mdp\_error\_table)

Format: Fehlernummer=Beschreibung

Beispiel:

[DecodedMDPErrors]

210=failed to read linkage

**[MDPErrorViewerColumns]**

Die Länge der Spalten aus ViewMDPErrors-Modul.

Format: Spaltenname=Länge

Beispiel:

```
[MDPErrorViewerColumns]
error_code_decoded=1530.142
area_name=1514.835
```

**[Oracle]**

Definiert das decodierte Passwort und den Benutzernamen für die Oracle-Datenbank und gilt übergeordnet, falls im Projekt dieser Eintrag nicht definiert ist. Siehe hierzu auch Punkt 2.2.1.

```
[Oracle]
User=system
Password=edacmncmlpcclp
```

**[History]**

Listet die fünf zuletzt benutzten Config-Files. Abrufbar über Menü File.

Beispiel:

```
[History]
INIFile1=c:\temp\config1.ini
INIFile2= c:\temp\config2.ini
```

## 6.2 Projekt – Konfigurationsdatei

### 6.2.1 [MDPmgr]

**Project**=projname

PDS-Projektname, mit dem MDP-Manager arbeitet.

**MDBFile**=C:\Temp\Project.mdb

Der Pfad zur Access Datenbank, in die Daten extrahiert werden. Wenn kein Dateiname angegeben ist, wird automatisch vom MDP-Manager einer generiert.

**Temp**=C:\TEMP

Verzeichnis, in dem die temporären Daten abgelegt werden. Wenn Temp-Variable im Konfigurationsfile des MDP-Managers gesetzt ist, wird sie benutzt. Sonst wird die Systemvariable TEMP verwendet.

**RISPath**=C:\Win32App\RIS\

Verzeichnis mit RIS-Utilities. Falls die Pfade bis zu RIS-Utilities nicht in der Systemvariable eingetragen sind, muss man den richtigen Pfad zur RIS-Utilities hier setzen.

**OracleServer**=ServerName

Oracle-Server mit PDS-Projekten. Wenn eine Verbindung zum Oracle Server ‚ServerName‘ aufgebaut werden kann, werden Daten aus PDS Oracle Client entladen. Über Oracle Client kann MDP-Manager um vielfaches schneller als über RIS auf die Daten zugreifen

**MDBSeed**= C:\Temp\mdbseed.mdb

Der Pfad zur Standard-Vorlage MS Access Datenbank. Das File wird VOR dem Laden der Daten aus PDS in **MDBFile** kopiert. Wenn `CreateNewAccessFile=0` wird Seed-File ignoriert.

**DGNSeed**= C:\Temp\dgnseed.dgn

Der Pfad zur Standard DGN-Zeichnung. DGN Seed File wird benutzt, um Fehler in ViewMDPErrors-Modul anzuzeigen.

**OnlyTransferToAccess=0**

Wenn **OnlyTransferToAccess=1**, werden Daten aus mto\_schema extrahiert **ohne** MDP-Batch Job zu starten. Wenn **OnlyTransferToAccess=0**, wird **zuerst** MDP-Batch Job gestartet, **danach** werden die Daten aus mto\_schema extrahiert.

**StdNote=1**

(Re)Publishing Standard Note Library

**LongMat=1**

Resolve Long Material Descriptions

**ShortMat=1**

Resolve Short Material Descriptions

**Forced=1**

Forced Publishing of Selection Set specified

**CreateNewAccessFile=1**

Wenn **CreateNewAccessFile=1** werden Daten in **neues** Access-File geladen. Dabei wird das alte File gesichert. Wenn **CreateNewAccessFile=0** werden die Daten in **vorhandenes** Access-File geladen. Dabei werden nur die Tabellen überschrieben. Abfragen bleiben unberührt.

**TransferOnlySelectedModels=1**

Wenn **TransferOnlySelectedModels=1** werden nur die Daten für die selektierte Modelle aus PDS in die Access Datenbank geladen. Sonst werden alle Daten geladen. Diese Einschränkung gilt nur für mto\_schema. Aus allen anderen Schemen werden alle Daten unabhängig davon geladen.

## 6.2.2 [LoadTable]

Der Abschnitt definiert Tabellennamen, die in Access Datenbank geladen werden sollen.

Format: Table[1-100]=Tabellenname|Präfix des Schemennamen

Wenn kein Präfix des Schemanamens angegeben ist, wird Tabelle aus mto\_schema geladen. Wenn ra\_schema des Projektes in anderem Format als

„ra\_projektname“ ist, kann man nach dem Pipe-Zeichen den kompletten Namen des ra\_schemas eingeben.

Beispiel:

Das ra\_schema für das PDS-Projekt „test1“ heißt ra\_test1. Dies wird dann in MDP-Manager Config-Datei so aussehen:

```
[LoadTable]
Table1=pdtable_500|ra
Table2=pdtable_12
Table3=pdtable_113|pd
```

Wenn aber ra\_schema für den PDS-Projekt test1 ra\_standart heißt, muss man die Konfigurationsdatei so ändern:

```
[LoadTable]
Table1=pdtable_500|ra_standart
Table2=pdtable_12
Table3=pdtable_113|pd
```

Dieser Abschnitt [LoadTable] wird komplett ignoriert, wenn der Abschnitt [LoadAllTablesFromSchema] nicht leer ist. Es können bis zu 100 Tabellen definiert werden.

### 6.2.3 [LoadAllTablesFromSchema]

Der Abschnitt definiert Schemanamen, die in Access Datenbank geladen werden sollen.

Format: Schemenname=1

Der Eintrag „other schemas“ bedeutet alle andere Schemen außer pd\_schema, mto\_schema, ra\_schema und dd\_schema.

Beispiel:

```
[LoadAllTablesFromSchema]
pd_test1=1
mto_test1=1
ra_test1=1
other schemas=1
```

Wenn der Abschnitt [LoadAllTablesFromSchema] nicht leer ist, wird der Abschnitt [LoadTable] komplett ignoriert.

## 6.2.4 [Quality]

Wenn Koordinatengrenzen gesetzt sind, wird MDP-Manager überprüfen, ob alle Bauteile innerhalb dieser Grenzen liegen. Fehlerhafte Datensätze werden in die Tabelle „bad\_lines“ hinzugefügt.

Beispiel:

```
[Quality]
Loweast=0
LowElev=0
HighNorth=999999999
HighEast=999999999
HighElev=999999999
```

Wenn PipeLineID und/oder EQPLineID gesetzt sind, werden Leitungen (Line-ID) und/oder Equipments (EQP-Name) geprüft, ob sie diesen Regeln entsprechen.

### **Regeln:**

- a. n (kleiner Buchstabe) - beliebige Ziffer
- b. c (kleiner Buchstabe) - beliebiger Buchstabe
- c. Andere Zeichen (A, X, -, \_, .) werden wie geschrieben verwendet.

Beispiel:

```
[Quality]
PipeLineID=nnn-ccc-nnn
EQPLineID= nnn-ccc-nnn
```

## 6.2.5 [CustomIndex]

In diesem Abschnitt können Spalten definiert werden, für die nach dem Laden der Daten aus PDS ein Index erstellt wird.

Format: Index[1-100]=Tabellenname|Spaltenname.

Beispiel:

```
[CustomIndex]
Index1=pdtable_12|LINE_SEQUENCE_NO
Index2=...
```

Es können bis zu 100 Einträge in dem Abschnitt gemacht werden.

### 6.2.6 [SQL]

Der Abschnitt beschreibt SQL-Statements, die nach dem Laden aller Tabellen ausgeführt werden. Format: SQL[1-100]=SQL-Statement.

Beispiel:

[SQL]

```
SQL1=INSERT INTO table SELECT line_id FROM phtable_12;
```

```
SQL2=...
```

Es können bis zu 100 SQL-Strings definiert werden.

### 6.2.7 [Query]

Der Abschnitt [Query] definiert Abfragen, die in Access Datenbank importiert werden sollen. Format: Query[1-100]=Abfragenname|Abfragetext.

Beispiel:

[Query]

```
Query1=Surface per line|SELECT * FROM phtable_12;
```

```
Query2=...
```

Es können bis zu 100 Abfragen definiert werden.

### 6.2.8 [ProjectControl]

In diesem Abschnitt können Auswahl-Abfragen definiert werden. Inhalt dieser Abfragen kann über MDP-Manager Menü (Tools->Project Control->"Name") angezeigt werden. Wobei „Name“ der Name der Abfrage ist (in diesem Fall „LineList“). Wenn in diesen SQL-Abfragen als erste Spalte „line\_id“ steht, wird beim rechten Mausklick in Project Control Fenster PopUp-Menü mit allen DGN-Zeichnungen dieser Leitung angezeigt. Man kann diese mit SmartPlantReview oder Microstation öffnen.

Beispiel:

[ProjectControl]

```
LineList=SELECT line_id FROM phtable_12
```

### 6.2.9 [RunBefore]

Hier können beliebige Kommandozeilebefehle definiert werden, die vor der Extraktion ausgeführt werden.

Format: Run[1-100]=DOS-Befehle

Beispiel:

```
[RunBefore]
```

```
Run1=copy c:\temp\project.mdb c:\temp\project.mdb.old
```

```
Run2=...
```

Es können bis zu 100 Einträge in dem Abschnitt gemacht werden.

### 6.2.10 [RunAfter]

Hier können beliebige Kommandozeilebefehle definiert werden, die am Ende der Extraktion ausgeführt werden.

Format: Run[1-100]=DOS-Befehle

Beispiel:

```
[RunAfter]
```

```
Run1=copy c:\temp\project.mdb c:\temp\project.mdb.old
```

```
Run2=...
```

Es können bis zu 100 Einträge in dem Abschnitt gemacht werden.

### 6.2.11 [Oracle]

Hier werden das projektspezifische dekodierte Passwort und der zugehörige Benutzername definiert. Per Copy&Paste kann dieser Eintrag in die mdpmgr.ini kopiert werden und dadurch für alle Projekte verfügbar gemacht werden.

```
[Oracle]
```

```
User=system
```

```
Password=edacmncmlpcclp
```

### 6.2.12 [ColDescr]

In diesem Abschnitt können Feldbeschreibungen für Access Datenbank definiert werden.

Format: Desc1=Tabellenname|Spaltenname|Beschreibung.

Beispiel:

```
[ColDescr]
```

```
Descr1=bad_lines|line_id|Leitung
```

Es können bis zu 100 Einträge in dem Abschnitt gemacht werden.

### 6.2.13 [ColCopy]

Im Abschnitt [ColCopy] können zu kopierende Spalten definiert werden.

Format: Copy[1-100]=Quelle\_tabelle.Quelle\_spalte|Ziel\_tabelle.Ziel\_spalte

Die Quelle- und Zielspalten müssen in einer Tabelle sein. Wenn Zielspalte nicht existiert, wird sie erstellt.

Beispiel:

```
[ColCopy]
```

```
Copy1=pdtable_34.commodity_code|pdtable_34.siz_dep_comm_code
```

```
Copy2=...
```

Es können bis zu 100 zu kopierende Spalten definiert werden.

### 6.2.14 [Relation]

Neue Beziehung zwischen Tabellen definieren.

Format: Relation[1-

100]=Table|Column1[,Column2[,ColumnN]]|ForeignTable|ForeignColumn1[,ForeignColumn2[,ForeignColumnN]]

Beispiel:

```
[Relation]
```

```
Relation1=pdtable_12|system_unique_no,partition_no|pd-  
table_80|idx_12,partition_no
```

```
Relation2=pdtable_12|partition_no|pdtable_900|par-  
tition_no
```

### 6.2.15 [Decode]

Zusätzlich zu den von MDP-Manager automatisch dekodierten Spalten kann man eigene Spalten definieren, die dekodiert werden.

Format: Decode1=Tabellenname|Spaltenname|Dekodierungsschlüssel

Beispiel:

```
[Decode]
```

```
Decode1=pdtable_12|fluid_code|125
```

```
Decode2=...
```

Man kann bis zu 100 Einträge in dem Abschnitt machen.

### 6.2.16 [DecodeSteel]

Dekodieren der Spalten aus der Tabelle pdtable\_532 (Steel).

Format: Decode1=Spaltenname|AttributName

Beispiel:

[DecodeSteel]

Decode1=design\_status|Design\_Status

Decode2=fabr\_note|Fabricator\_Note

Man kann bis zu 100 Einträge in dem Abschnitt machen.

### 6.2.17 [ReplacePath]

Ersetzen der Pfade zu den DGN-Dateien in ViewMDPErrors-Modul.

Format: Path1=SuchenNach|ErsetzenDurch

Wenn man mit dem ViewMDPErrors-Modul den Fehler betrachtet, werden die entsprechenden DGN-Dateien in MicroStation angezeigt. Die Pfade zu diesen Dateien werden automatisch aus dem PDS ermittelt. Wenn DGN-Dateien über diese Pfade nicht erreichbar sind (z.B. man hat keine Rechte, es sind nicht alle Laufwerke/Ordner freigegeben), kann man diese ersetzen.

Beispiel:

MDP-Manager hat den folgenden Pfad für die Datei test1.dgn ermittelt:

[\\PDS-Server\d\pds\project\test1.dgn](#). Der Rechner, auf dem MDP-Manager ausgeführt wird, hat ein Netzwerklaufwerk O:, der mit folgendem Ordner auf dem PDS-Server verbunden ist: \\PDS-Server\d\pds.

d.h. Man kann die Datei über zwei Wege erreichen:

[\\PDS-Server\d\pds\project\test1.dgn](#)

O:\project\test1.dgn

Für diesen Fall kann man folgende Ersetzungsregel in MDP-Manager Konfigurationsdatei eintragen:

[ReplacePath]

Path1=\\PDS-Server\d\pds|O:

Jetzt wird MDP-Manager auf die Datei test1.dgn über den Pfad [\\PDS-Server\d\pds\project\test1.dgn](#) zugreifen.

Es können bis zu 100 Einträge in dem Abschnitt gemacht werden.

### 6.2.18 [SchemasPass]

Setzen der Schema-Kennwörter für den Zugriff auf die Daten über Oracle-Client.

Format: Schemaname=Kennwort

Das Kennwort muss in der umgekehrten Reihenfolge geschrieben werden. Gross- und Kleinschreibung für die Schemennamen spielen keine Rolle, für Kennwörter aber schon. Wenn fürs Schema kein Kennwort definiert ist, wird das Kennwort automatisch auf den Schemennamen gesetzt.

Beispiel:

für das Schema pd\_test1 wird das Kennwort „pass1“ definiert:

```
[SchemasPass]
```

```
PD_TEST1=1ssap
```

### 6.2.19 [Export2Excel]

Exportierung der Daten aus der Access Datenbank nach Excel. Die Excel-Datei wird automatisch im gleichen Verzeichnis mit gleichem Namen wie Access Datenbank-Datei angelegt.

Format: Export[1-100]=Tabellen-/Abfragenname

Wenn man alle Tabellen und Abfragen exportieren will, kann man folgende Syntax verwenden:

```
Export1=*
```

Beispiel:

```
[Export2Excel]
```

```
Export1=pdtable_12
```

```
Export2=pdtable_34
```

### 6.2.20 [PartsListModule]

Nachdem MDP-Manager fertig ist, wird Parts-List Modul gestartet. Wenn im Parts-List Modul keine Access Datenbank definiert ist, wird MDP-Manager Datenbank genommen. Wenn keine Access Datenbank für die Ident-Tabelle definiert ist, wird automatisch eine erstellt.

Wenn keine Startnummer definiert ist, wird als Startwert für die Ident-Nummer 1 genommen.

Wenn **UpdateIdentTable=1** wird ident\_tabelle erstellt/updatet.

Wenn **GenerateIdentNo=1** werden für Bauteile ohne entsprechende Nummer IdentNo generiert.

Wenn **writeIdentNo=1** wird IdentNo zurück in die Access Datenbank in die Spalte **size\_dep\_comm\_code** geschrieben.

Beispiel:

```
[PartsListModule]
UpdateIdentTable=1
GenerateIdentNo=1
WriteIdentNo=1
```

### 6.2.21 [IdentAlloc]

Hier werden die Einstellungen für PartsList-Modul gespeichert.

**startIdentNo=1**

Startwert für die Ident-Nummer.

**MDBFile= C:\Temp\Project.mdb**

Access Datenbank wohin die Ident-Nummer geschrieben werden sollen.

**IdentMDBFile= C:\Temp\Project\_ident.mdb**

Access Datenbank mit ident\_alloc-Tabelle.

### 6.2.22 [MatReq]

Hier werden die Einstellungen für PartsList-Module gespeichert.

**NewMDBFile= C:\Temp\Project\_new.mdb**

Neue Access Datenbank, die verglichen werden soll

**OldMDBFile= C:\Temp\Project\_old.mdb**

Alte Access Datenbank, die verglichen werden soll

**NoPDSMatColumnItem=item**

Spaltenname mit dem Schlüssel für Nicht-PDS-Material

**NoPDSMatColumnQuantity=quantity**

Spaltenname mit den Werten für Nicht-PDS-Material

**NoPDSMatTable=test**

Tabellenname, aus der Item- und Quantity-Spalten stammen.

### 6.2.23 [WriteToPDS]

**OutputTable**=pdtable\_12

PDS-Tabellenname, wohin Date zurückgeschrieben werden

**InputTable**=Abfrage1

Input Tabelle oder Abfrage

**Forced**=0

Wenn 1, werden alle Daten in PDS überschrieben. Wenn 0, werden nur die Daten mit dem Wert „NULL“ geändert.

**InputMDBFile**=c:\Beispiele\demo.mdb

Input Access Datenbank.

Wenn keine Input Access Datenbank definiert ist, wird MDP-Manager Datenbank verwendet.

**CreateLogFile**=1

Wenn 1, wird eine LOG Access Datei erstellt.

**LogMDBFile**=c:\Beispiele\demo\_log.mdb

LOG Access Datenbank.

Wenn keine LOG Access Datenbank definiert ist, wird automatisch eine erstellt.

### 6.2.24 [WriteToPDSAllowUpdateTables]

Hier werden Tabellen definiert, die vom WriteToPDS-Modul geändert werden dürfen.

Format: Tabellenname=1 oder \*=1 (Alle Tabellen dürfen verändert werden)

Beispiel:

```
[WriteToPDSAllowUpdateTables]
```

```
pdtable_12=1
```

### 6.2.25 [WriteToPDSAllowUpdateColumns]

Hier werden Spalten definiert, die von WriteToPDS-Module geändert werden dürfen.

Format: Spaltenname=1 oder \*=1 (Alle Spalten dürfen verändert werden)

Beispiel:

```
[WriteToPDSAllowUpdateColumns]
```

```
fluid_code=1
```

### 6.2.26 [Abfrage1]

In dem WriteToPDS-Modul wird für jede Input Tabelle/Abfrage ein eigener Abschnitt erstellt, der wie Input Tabelle/Abfrage heißt. In diesem Abschnitt werden Spalten-Einstellungen (Bedingungsspalte, Output-Spalte, Spalten-Schlüssel) in folgendem Format gespeichert:

Spaltenname=[0,1]|Dekodierungs-Schlüssel

Wenn ein Dekodierungs-Schlüssel vorhanden ist, wird diese Spalte mit Hilfe von phtable\_250 dekodiert.

Beispiel:

```
[Abfrage1]
```

```
fluid_code=0|125
```

*fluid\_code-Spalte aus der Tabelle/Abfrage ist eine Outputspalte, die über die Tabelle phtable\_250 mit dem Schlüssel 125 dekodiert werden muss.*

```
line_id=1|
```

*line\_id-Spalte aus der Tabelle/Abfrage ist eine Bedingungsspalte (Inputspalte)*

```
pipng_mater_class=0|
```

*pipng\_mater\_class -Spalte aus der Tabelle/Abfrage ist eine Outputspalte*

### 6.2.27 [VerifyDataRules]

In diesem Abschnitt werden die Regeln für VerifyData-Modul gespeichert.

Format:

Für die Logik: „IN“ oder „NOT IN“:

Rule[1-100]=Tabellenname.Spaltenname|Logik|Tabellenname.Spaltenname

Für andere Typen von Logik:

Rule[1-100]=Tabellenname.Spaltenname|Logik|Wert

Beispiel:

```
[VerifyDataRules]
```

```
Rule1=phtable_12.npd|<>|0
```

```
Rule2=...
```

Es können bis zu 100 Einträge in dem Abschnitt gemacht werden.