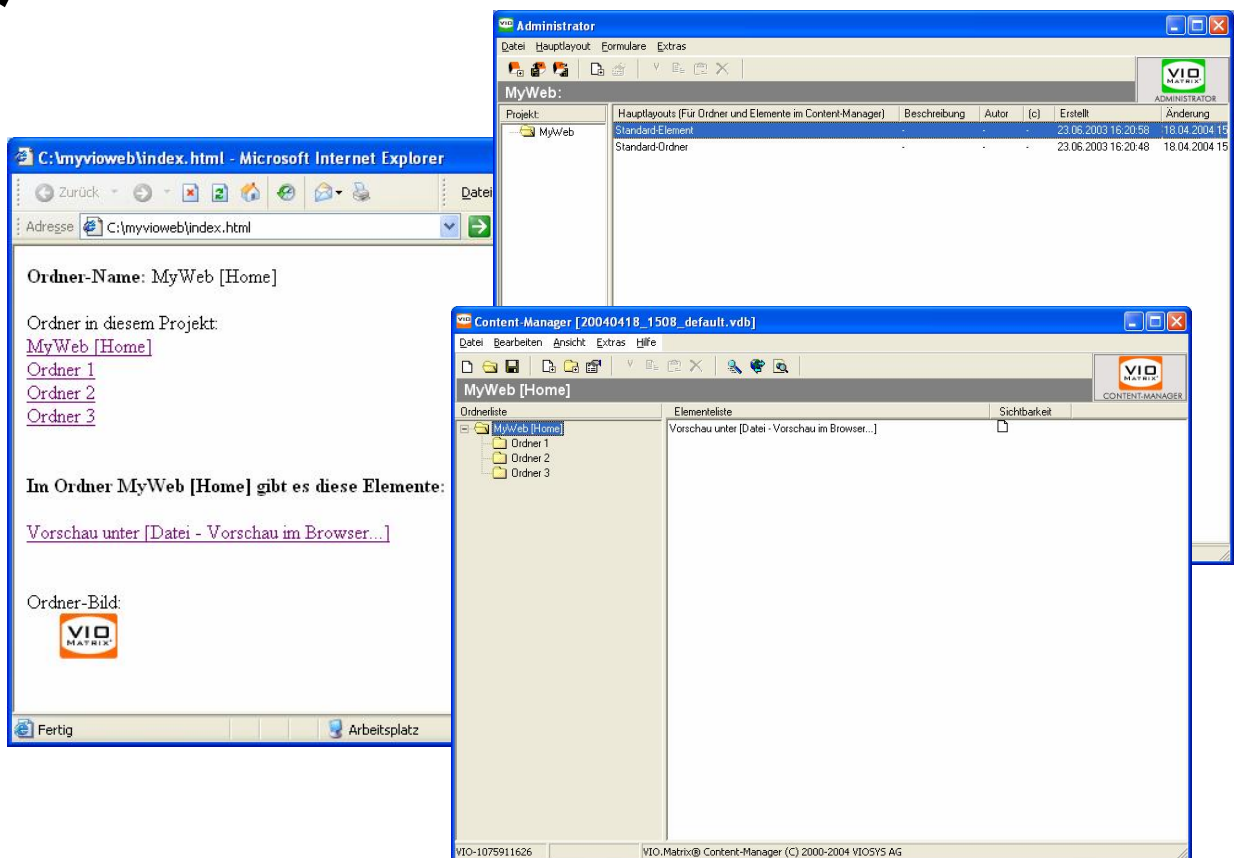


VIO.MATRIX

Parameter

Oktober 04, Rev. 06-1214

Webseiten mit Content Management einfacher und schneller als mit PHP oder ASP erstellen – durch VIO.Matrix-Technologie!



VIO.Matrix ist die Software für dynamische Webseiten, Content-Management und E-Commerce in einem System. Dieser User-Guide enthält eine Liste verfügbarer Parameter zur Integration von VIO.Matrix Webprojekten.

Diese Dokumentation ist urheberrechtlich geschützt.
Copyright © 2004 VIOSYS AG.
Alle Rechte vorbehalten.

Inhaltsverzeichnis

- 1 Verwendung von HTML-Platzhaltern
- 2 Aufruf von Ordner-/Elementdaten
- 3 Parameter für Ordner (Befehlsklasse OR)
- 4 Parameter für Elemente (Befehlsklasse EL)
- 5 Platzhalter für Eigenschaften von Elementen
- 6 Platzhalter für Ausprägungen der Eigenschaften von Elementen
- 7 Layoutparameter (Befehlsklasse LY)
- 8 #IF-Abfragen
- 9 Temporäre Variablen über #SET
- 10 Fehlercodes VIO.Matrix-CGI
- 11 Suchparameter (Befehlsklasse SE)
- 12 Warenkorbparameter (Befehlsklassen WK, WE und MW)
- 13 Parameter für Kunden (Befehlsklasse KD)
- 14 Parameter für Webnutzergruppen (Befehlsklasse WG)
- 15 Parameter für Webnutzer (Befehlsklasse WU)
- 16 CGI-Übergabeparameter
- 17 Systemparameter
- 18 Sonstige Parameter (Befehlsklasse SP)

VIO.MATRIX HOME EDITION

1 Die Syntax

Perfekte Arbeitsteilung durch Trennung von
Inhalt und Design mit zwei getrennten Programmen:
Content-Manager und Administrator

1 Verwendung von HTML-Platzhaltern

Der mit dem VIO.Matrix Administrator integrierte HTML-Code wird vom VIO.Matrix-CGI geparkt. Bei diesem Vorgang werden alle VIO.Matrix-Befehle durch die entsprechenden Werte ersetzt. Die somit generierte HTML-Seite wird dann an den Browser des Webbesuchers übermittelt. VIO.Matrix-Befehle haben immer folgenden Aufbau:

```
#INSERT_[Klasse]_[#Hierarchie]_[Befehlswort]_[Parameter]!
```

Typ	Beschreibung
Klasse	Bezeichnet die Befehlsklasse, d.h. welchen Bezug der Befehl besitzt. Mögliche Ausprägungen sind: KD (Kunde), EL (Element), OR (Ordner), SE (Suchergebnisse), SP (spezielle Funktionen)
Hierarchie	Angabe einer ID (bezogen auf die Klasse und damit nur bei OR und EL relevant), auf welche sich der folgende Befehl beziehen soll. Fehlt diese Angabe, wird die logisch aktuelle ID verwendet. D.h. wurde das CGI mit dem Parameter or=2 aufgerufen, wird hier automatisch der Ordner mit der ID 2 verwendet.
Befehlswort	Ein Befehlswort, welches bezogen auf die Klasse gültig ist. Es ist zu beachten, das einige Befehlswoorte einen Parameter erwarten (bspw. erwartet der Befehl „EXEC“ der Klasse „SP“ den absoluten Pfad und Namen einer ausführbaren Datei).
Parameter	Ein vom Befehlswort erwarteter Parameter. Der Parameter wird üblicherweise von einem Ausrufezeichen abgeschlossen. Innerhalb des Parameters können in geschweiften Klammern („{„, „}“) VIO.Matrix HTML-Platzhalter stehen. Dies ist bspw. sinnvoll wenn sich bestimmte Parameter aus vorher gesicherten Kundendaten ergeben. Format: Parameter = [...[[[{{codec:[codec]}[HTML-Platzhalter]...}...]]...] Beispiel: #INSERT_SP_EXEC_/usr/bin/perl board.pl \ „name={ { codec:web} #INSERT_KD_name! } &or={ #INSERT_OR_ID} !

2 Aufruf von Ordner-/Elementdaten

Parameter	Typ	Wert	Beschreibung
OR_	-	-	Ordnerparameter des aktuellen Ordners
OR_# [ID] _	int	OrdnerID	Holt die Inhalte aus dem angegebenen Ordner.
OR_# {parent} _	-	-	Holt die Inhalte aus dem übergeordneten Ordner.
OR_# {prev} _	-	-	Holt die Inhalte aus dem vorherigen Ordner derselben Ebene.
OR_# {next} _	-	-	Holt die Inhalte aus dem nachfolgenden Ordner derselben Ebene.
OR_# {level:[number]} _	-	-	Holt die Inhalte aus dem Ordner der angegebenen Ebene. Ebene 0 stellt den Wurzelordner dar.
OR_# {#INSERT_[...]} _	-	-	Holt die Inhalte aus dem Ordner mit der über den #INSERT-Befehl spezifizierten Ordner. Bspw.: <i>Eine in einer Kundenvariable gesicherte OrdnerID</i> <small>{#INSERT_OR_#{#INSERT_KD_meinordner!}_ZEILE}</small> .
EL_	-	-	Elementparameter des aktuellen Elementes
EL_# [ID] _	int	ElementID	Holt die Inhalte des angegebenen Elementes
EL_# {prev} _	-	-	Holt die Inhalte aus dem vorherigen Element derselben Ebene.
EL_# {next} _	-	-	Holt die Inhalte aus dem nachfolgenden Element derselben Ebene.
EL_# {#INSERT_[...]} _	-	-	Holt die Inhalte aus dem Element mit der über den #INSERT-Befehl spezifizierten Element. Bspw.: <i>Eine in einer Kundenvariable gesicherte ElementID</i> <small>{#INSERT_EL_#{#INSERT_KD_meinelement!}_NAME}</small> .

3 Parameter für Ordner (Befehlsklasse OR)

Parameter	Typ	Wert	Beschreibung
OR_ID	int	ID des Ordners	ID des aktuellen Ordners.
OR_NAME	string	Ordnername	Name des Ordners.
OR_OORDNER	int	ID des Oberordners	ID des übergeordneten Ordners
OR_OORDNER_NAME	string	Oberordnername	Name des übergeordneten Ordners
OR_ORANZAHL	int	Ordneranzahl	Anzahl der enthaltenen Unterordner
OR_ELANZAHL	int	Elementanzahl	Anzahl der enthaltenen Elemente
OR_FIRSTEL	int	ID eines Elementes	ID des ersten Elementes, welches sich im Ordner befindet oder -1 im Fehlerfall.
OR_LASTEL	int	ID eines Elementes	ID des letzten Elementes, welches sich im Ordner befindet oder -1 im Fehlerfall.
OR_FIRSTOR	int	ID eines Ordners	ID des ersten Unterordners, welcher sich im Ordner befindet oder -1 im Fehlerfall.
OR_LASTOR	int	ID eines Ordners	ID des letzten Unterordners, welcher sich im Ordner befindet oder -1 im Fehlerfall.
OR_PREVOR	int	ID eines Ordners	ID des vorhergehenden Ordners (gleiche Ebene) oder -1 im Fehlerfall
OR_NEXTOR	int	ID eines Ordners	ID des nächsten Ordners (gleiche Ebene) oder -1 im Fehlerfall
OR_FL	int	fil. Nummer	Fortlaufende Nummer (Beginnend bei 1) mit max. 3 führenden Nullen („0001“)
OR_CYLFARBE	string	Farbe	Wechselnde Farbe in Abhängigkeit von OR_FL.
OR_ZEILE	string	Ordnerzeile	Zeigt mit Hilfe der in „ordnerzeile“ definierten HTML-Layouts nacheinander alle in der Ordner enthaltenen Unterordnern an.
OR_ZEILE_[name]!	string	indiv. Ordnerzeile	Zeigt die in der data.txt unter dem Namen: ordnerzeile_[name] definierte Ordnerzeile an.
OR_ZEILE_[layouttyp]!	string	indiv. Ordnerzeile	Zeigt die als Layoutuntertyp unter dem Namen definierte Ordnerzeile an.

OR_ELZEILE	string	Elementliste	Die Liste aller Element, welche sich unter dem aktuellen Ordner befinden, wird angezeigt.
OR_ELZEILE_[name]!	string	indiv. Elementzeile	Zeigt die in der data.txt unter dem Namen: Elementzeile_[name] definierte Elementzeile an.
OR_ELZEILE_[layouttyp]!	string	indiv. Elementzeile	Zeigt die als Layoutuntertyp unter dem Namen definierte Elementzeile an.
OR_LIST_[varname]!	string	Listenzeile	fügt eine komplette Liste von Einträgen aus Textauswahlfeldern bzw. der vio_eingabe.dll ein; das zugehörige Layout steht in der Detailzeile; innerhalb der Zeile sind die Parameter #LTEXT (Text eines Listenelementes) und #FL (fortlaufende Nummerierung) zulässig
OR_LIST_[varname]!_[name]!	string	Listenzeile	fügt eine komplette Liste von Einträgen aus Textauswahlfeldern bzw. der vio_eingabe.dll ein; das zugehörige Layout steht in der mit [name] spezifizierten Detailzeile; innerhalb der Zeile sind die Parameter #LTEXT (Text eines Listenelementes) und #FL (fortlaufende Nummerierung) zulässig
OR_HIERARCHIE	string	Ordnerhierarchie	Anzeige der direkt übergeordneten Ordnern anhand der Vorlage aus „OR hierarchie“
OR_VAR_[varname]!	string	Variable	Anzeige des formatierten Inhaltes der Ordner zugeordneten Variable des Namens „varname“. Ist die Variable vom Typ Text (DLL) findet innerhalb von <a>-Tags ein Preparing statt.
OR_VAR_[varname]*length!	int	Variable	Länge des Variableninhaltes (Anzahl der Zeichen)
OR_LIST_[varname]!_[layoutname]!	string	Inhalte einer Variablenliste	Zeigt die Inhalte einer Textauswahlliste bzw. über eine vio_eingabe.dll erzeugte Liste an. Der „layoutname“ gibt den Titel eines Unterlayouts an, welches die darzustellenden Listenzeilen enthält. Neben allen #INSERT-Befehlen, können hier die Befehle #LTEXT für den Listeneintrag und #FL für

			eine fortlaufende Nummer genutzt werden.
OR_ABSTRACT[Anzahl]	string	Beginn der Beschreibung	Gibt die ersten „Anzahl“ Buchstaben der Ordnerbeschreibung aus.
OR_TYP	int	Ordnerart	Ausgabe des Ordnerart: Ordner enthält... nur Ordner nur Elemente Ordner und Elemente keine Ordner und keine Elemente 4
OR_RANDOR	int	Ordner-ID	Wählt einen zufälligen Ordner aus der Liste der dem Ordner untergeordneten Ordner aus.
OR_RANDEL	int	Elemente-ID	Wählt ein dem Ordner direkt untergeordnetes Element per Zufall aus.
OR_LEVEL	int	Hierarchieangabe	Gibt den Hierarchielevel(-ebene) des aktuellen Ordners aus (Wurzelordner = 0)
OR_LEVELID_[level]!	int	Ordner-ID	Gibt die Ordner-ID eines übergeordneten Ordners auf einem bestimmten Level an.
OR_VAR_[varname]*hex!	string	Farbwert	Ausgabe eines Farbwertes (Komponente Farbauswahlfeld) als Hexadezimalzahl (ohne führendes #-Zeichen)

4 Parameter für Elemente (Befehlsklasse EL)

Parameter	Typ	Wert	Beschreibung
EL_ID	int	ID eines Elementes	ID des aktuellen Elementes
EL_NAME	string	Elementname	Name des Elementes
EL_ORDNER	int	Ordner ID	ID der Elementordner, welcher Element untergeordnet ist.
EL_ORDNERNAME	string	Ordnername	Name der Elementordner, welcher Element untergeordnet ist.
EL_PREVEL	int	ID eines Elementes	ID des vorhergehenden Elementes oder -1 im Fehlerfall
EL_NEXTEL	int	ID eines Elementes	ID des nächsten Elementes oder -1 im Fehlerfall
EL_FL	int	fortlaufende Nummer	Fortlaufende Nummer, welche bspw. Elementlisten durchnummeriert. Es wird mit dem Wert 1 begonnen.
EL_CYLFARBE	string	wechselnde Farben	In Abhängigkeit der fortlaufenden Nummer (EL_FL) werden hier abwechselnd zwei Zeichenketten (bspw. Farben) ausgegeben.
EL_MENGE	int	Menge in Korb	Menge des Elementes im Warenkorb des Kunden.
EL_ESLISTE	string	Elementeigenschaften	Die Eigenschaften des Elementes werden angezeigt. Als Layoutvorgabe wird die „Eigenschaftenzeile“ verwendet.
EL_ESLISTE_[layouttyp]!	string	Elementeigenschaften	Die Eigenschaften des Elementes werden nach Vorlage des Layouttypes „layouttyp“ angezeigt.
EL_SUBSTITUTE	string	Elementesubstitute	Die Liste der verbundenen Elemente zum Element wird angezeigt.
EL_SUBSTITUTE_[layouttyp]!	string	Elementesubstitute	Die Liste der verbundenen Elemente zum Element werden nach Vorlage des Layouttyps „layouttyp“ angezeigt. Die HTML-Seite des Layouttyps enthält dann die Anweisung EL_SUBSTZEILE.
EL_SUBSTZEILE_[layouttyp]!	string	Liste der verbundenen Elemente	Die Liste der verbundenen Elemente zum Element wird angezeigt.
EL_SUBSTZEILE		Liste der verbundenen Elemente	Die Liste der verbundenen Elemente zum Element werden nach Vorlage des Layouttyps „layouttyp“ angezeigt. Die HTML-Seite des Layouttyps enthält dann die Anweisung EL_SUBSTZEILE.
EL_ABSTRACT[Anzahl]	string	Beginn der	Gibt die ersten „Anzahl“

		Beschreibung	Buchstaben der Elementbeschreibung aus.
EL_VAR_[varname]!	string	Variable	Anzeige des formatierten Inhaltes der dem Element zugeordneten Variable des Namens „varname“. Ist die Variable vom Typ Text (DLL) findet innerhalb von <a>-Tags ein Preparing statt.
EL_VAR_[varname]*length!	int	Variable	Länge des Variableninhaltes (Anzahl der Zeichen)
EL_VAR_[varname]*hex!	string	Farbwert	Ausgabe eines Farbwertes (Komponente Farbauswahlfeld) als Hexadezimalzahl (ohne führendes #-Zeichen)

5 Platzhalter für Eigenschaften von Elementen

Parameter	Typ	Wert	Beschreibung
ESFL	int	Nummer der Eigenschaft	4stellige Nummer der Eigenschaft (bspw. „0001“ beginnend mit 1.
ELID	int	ID des Elementes	6stellige ID des Elementes zu welchem Eigenschaften gehören.
WAHL	string	gewählte Ausprägung	Zeichenkette, welche die vom Kunden gewählte Eigenschaftsausprägung zeigt.
NAME	string	Name	Name der Eigenschaft
ANREXT	string	Elementnummererweiterung	Erweiterung der Elementnummer, welche durch Eigenschaft gegeben ist.
INDIVIDUELL	bool	1 : IndividualES 0 : Ausprägungen	Liefert eine „1“, wenn es sich um eine Individualeigenschaft handelt, ansonsten eine „0“.
AUSPRAEGUNG	string	Ausprägungen	Liefert das Layout, welches die Liste der Ausprägungen beinhaltet.
OPTION	string	Liste der Ausprägungen	Liefert eine Liste der mit der Eigenschaft verknüpften Ausprägungen bzw. das Eingabefeld für eine Individualeigenschaft.

6 Platzhalter für Ausprägungen der Eigenschaften von Elementen

Parameter	Typ	Wert	Beschreibung
NAME	string	Ausprägungsname	Titel der aktuellen Ausprägung
ANREXT	string	Elementnummererweiterung	Elementnummererweiterung, um welche Elementnummer inkl. Eigenschaftenerweiterung ergänzt wird.
SELECTED	string	„selected“	Liefert die Zeichenkette „selected“, wenn die Ausprägung im Warenkorb des Kunden selektiert ist ansonsten einen leeren String.

7 Layoutparameter (Befehlsklasse LY)

Parameter	Typ	Wert	Beschreibung
LY_[layoutname]!	string	Inhalt eines Layout	An die gewünschte Stelle wird das Layout mit dem bezeichneten Namen eingefügt.
LY_[layoutname]#[unterlayoutname]!	string	Inhalt eines Layout	An die gewünschte Stelle wird ein Unterlayout mit dem bezeichneten Namen eingefügt. Das Unterlayout muss sich im bezeichneten Layout befinden.

8 #IF-Abfragen

Parameter	Beschreibung
#IF	<p>Syntax: #IF ([Bedingung]) [Positivblock] [#ELSE [Negativblock]] #ENDIF</p> <p>#IF leitet die bedingte Anweisung ein. In runden Klammern folgt die Bedingung, die aus 2 Vergleichsteilen getrennt durch einen Vergleichsoperator besteht.</p> <p>Gültige Vergleichsoperatoren sind: "==" (gleich), "!=" (ungleich), "<=" (kleiner oder gleich), ">=" (größer oder gleich), "<" (streng kleiner) und ">" (streng größer).</p> <p>Es wird zwischen String- und Integervergleich unterschieden. Beim Stringvergleich müssen die beiden Vergleichsteile in doppelte Hochkommata ("[Vergleichsteil]") gesetzt werden, beim Integervergleich stehen beide Vergleichsteile direkt da.</p> <p><u>Beispiel Stringvergleich:</u></p> <pre>... #IF ("#INSERT_OR_NAME" == "VIOSYS") 1) ... #ENDIF ...</pre> <p><u>Beispiel Integervergleich:</u></p> <pre>... #IF (#INSERT_OR_ID > 1) ... #ENDIF ...</pre> <p><u>Hinweise:</u> Wird nur ein Vergleichsteil angegeben (und kein Vergleichsoperator) wird der Positivblock abgearbeitet, sobald beim Stringvergleich ein String existiert bzw. beim Integervergleich der Vergleichsausdruck ungleich 0 ist. In der Bedingung sowie im Positiv- und im Negativblock müssen alle Parameter in der Langform verwendet werden!</p>
#ELSE	Optional: Trennung von Positiv- und Negativblock
#ENDIF	Ende der bedingten Anweisung.

9 Temporäre Variablen über #SET

Parameter	Beschreibung
#SET	<p>Syntax: <code>#SET[Variablenname]=[Variableninhalt]#ENDSET</code></p> <p>Nutzen Sie den #SET #ENDSET-Anweisungsblock, um innerhalb eines HTML-Templates temporäre Variablen zu erzeugen und mit Inhalten zu versehen. Diese Anweisung wird genutzt, um Werte, welche sich bspw. innerhalb von Ordner- oder Elementelisten ändern, zwischenspeichern und innerhalb von Listen zu verwenden.</p> <p>Die Ausgabe der Variableninhalt erfolgt über den Befehl <code>#INSERT_SP_VAR_[Variablenname]!</code></p> <p>Bitte beachten:</p> <ul style="list-style-type: none"> • Der interne Variablentyp ist Text, d.h. Zahlenwerte werden als Zeichenkette gesichert • Leerzeichen werden aus dem Variablennamen und Inhalt entfernt, außer(!) sie befinden sich in Anführungszeichen (bspw. <code>#SET meintext = "#INSERT_EL_VAR_beschr!" #ENDSET</code>) • Nach Abarbeitung des Templates gehen die Variablen verloren, eine Speicherung von Werten über mehrere Sitzungen ist nur mit Hilfe der Webnutzer-/Kundenvariablen möglich. • über <code>#SET kd_[varname] = "... " #ENDSET</code> können auch Kundenvariablen innerhalb eines Templates geändert werden
#ENDSET	Ende des Anweisungsblocks zum Setzen temporärer Variablen

10 Fehlercodes VIO.Matrix-CGI

Rückgabewert	Fehlername	Beschreibung
200	ok	Seite wurde korrekt ausgeliefert
400	Fehlerhafte Anfrage	Eine fehlerhafte Anfrage ist beispielsweise, wenn ein Ordner bzw. Element zwar existiert, aber kein Layout zugewiesen ist oder das zugewiesene Layout nicht (mehr) existiert
401	Nutzer nicht autorisiert	Der Ordner bzw. das Element ist nur für bestimmte Webnutzerguppen freigegeben. Gehört der aktuelle Nutzer nicht zu einer dieser Gruppen (oder ist der Nutzer nicht angemeldet), kann die Seite nicht ausgeliefert werden, weil der Nutzer nicht autorisiert ist.
403	Zugriff verboten	Der Zugriff auf den Ordner bzw. das Element ist verboten, weil der Ordner bzw. das Element nicht sichtbar ist
404	Nicht gefunden	Der Ordner bzw. das Element wurde nicht gefunden, d.h. existiert nicht in der VIO.Matrix-Datenbank

Hinweis:

Für die Fehlercodes 4xx kann eine Weiterleitung auf eine entsprechende Fehlerseite eingerichtet werden. Dazu muss ein Systemparameter nach dem Schema `error[Rückgabewert]` mit dem Weiterleitungsziel im Fehlerfall belegt werden, z.B. `error404=http://www.kunden-viosys.de/error404.html` leitet im Fehlerfall 404 (Ordner bzw. Element wurde nicht gefunden) auf die Seite `http://www.kunden-viosys.de/error404.html` weiter.

11 Suchparameter (Befehlsklasse SE)

Definition eines Suchblockes

Parameter	Beschreibung
#SEARCH	<p><u>Syntax:</u> #SEARCH [Suchblock] #ENDSEARCH</p> <p>Angabe einer Suchdefinition und Ausführung einer Suche direkt im Template (die Suche findet nicht über neuen CGI-Aufruf statt). Damit sind mehrere Suchabfragen in einem Template möglich.</p> <p>Dabei definiert der Parameter search_id die ID des Suchblockes. Die ID des Suchblockes wird bei den #INSERT_SE_-Befehlen angegeben, um auf den entsprechenden Suchblock zuzugreifen (z.B. #INSERT_SE_#[ID des Suchblockes]_...).</p> <p>Wird bei den #INSERT_SE_-Befehlen keine Suchblock-ID angegeben, wird die Suche mit den über den CGI-Aufruf übergebenen Parametern durchgeführt.</p> <p>Außer dem Parameter search_id können innerhalb des #SEARCH-Statements die üblichen Suchparameter verwendet werden.</p> <p><u>Beispiel:</u></p> <pre>#SEARCH search_id = "1" search_varname0001 = "_nd" search_vartyp0001 = "100" search_varmin0001 = "[Suchbegriff]" #ENDSEARCH</pre> <p>Anzeige der Suchergebnisse: #INSERT_SE_#1_RESULTS_results!</p>
#ENDSEACRH	Ende des Suchblockes

Parameter	Typ	Wert	Beschreibung
SE_RESULTS[_name!]	string	Suchergebnisseite	Sind Suchergebnisse vorhanden (Elemente oder Ordner) werden diese hier eingefügt. „name“ gibt den Namen eines Unterlayouts an.
SE_NORESULTS[_name!]	string	Inhalt, wenn keine Suchergebnisse	Sind keine Suchergebnisse vorhanden (Elemente oder Ordner) wird das angegebene Unterlayout abgearbeitet.
SE_ZEILE[_name!]	string	Elementzeilen	Suchergebniszeilen mit allen gefundenen Elementen zu einer Suchanfrage.
SE_ORZEILE[_name!]	string	Ordnerzeilen	Suchergebniszeilen mit allen gefundenen Ordnern zu einer Suchanfrage.
SE_ANZEL	int	gefundene Elemente	Anzahl der aktuellen gefundenen Elemente
SE_ANZOR	int	gefundene Ordner	Anzahl der aktuellen gefundenen Ordner
SE_MAXEL	int	maximale Trefferzahl	Anzahl der Suchergebnisse (Elemente), bei der die Suche abgebrochen wird.
SE_MAXOR	int	maximale Trefferzahl	Anzahl der Suchergebnisse (Ordner), bei der die Suche abgebrochen wird.
SE_PREVEL	int	Elemente-ID	ID des vorherigen Elementes im Suchergebnis
SE_NEXTEL	int	Elemente-ID	ID des nächsten Elementes im Suchergebnis
SE_PREVOR	int	Ordner-ID	ID des vorherigen Ordners im Suchergebnis
SE_NEXTOR	int	Ordner-ID	ID des nächsten Ordners im Suchergebnis
SE_START	int	Blockstart	Nummer des Elementes oder Ordners, mit dem der aktuelle Suchergebnisblock begonnen werden soll
SE_BLOCK	int	Blockumfang	Anzahl der max. anzuzeigenden Suchergebnisse
SE_PREV	int	vorheriger Block	Blockstart des vorherigen Blockes
SE_NEXT	int	nächster Block	Blockstart des nächsten Blockes
SE_END	int	Blockende	Nummer des letzten angezeigten Suchergebnisses im aktuellen Block

12 Warenkorbparameter (Befehlsklassen WK, WE und MW)

Parameter	Beschreibung
#WKORB	<p><u>Syntax:</u> #WKORB [Warenkorblock] #ENDWKORB</p> <p>Innerhalb eines Templates kann mit den Parametern zur Ablage eines Elementes im Warenkorb ein Element direkt (ohne Übergabe der Parameter beim CGI-Aufruf) in den Warenkorb gelegt werden.</p> <p><u>Beispiel:</u></p> <pre>#WKORB wkorb0001 = "1" el0001 = "#INSERT_EL_ID" menge0001 = "1" eur0001 = "1,99" mwst0001 = "16,00" #ENDWKORB</pre>
#ENDWKORB	Ende des Warenkorbblockes

Tabelle 1 – Definition eines Warenkorbblockes

Parameter	Beschreibung
#INSERT_WK_#[Warenkorb-ID]_...	Wird über den Befehl #INSERT_WK_ ein Warenkorb angesprochen, muss immer eine Warenkorb-ID angegeben werden, die auf den gewünschten Warenkorb verweist.

Tabelle 2 –Warenkorbparameterrufe allgemein

Parameter	Typ	Wert	Beschreibung																
WK_ZEILE	Liste	Warenkorbliste	Liste der Einträge im Warenkorb																
WK_VAR_[Variablenname]!		Summe einer Warenkorbbeitragsvariablen oder Elementvariablen	<p>Summierung einer speziellen Warenkorb- oder Elementvariablen</p> <p><u>spezielle Warenkorbvariablen:</u></p> <table> <tr> <td>_brutto</td> <td>Bruttobetrag (Gesamtpreis)</td> </tr> <tr> <td>_netto</td> <td>Nettobetrag (Gesamtpreis)</td> </tr> <tr> <td>_mwstbetrag</td> <td>Mehrwertste</td> </tr> <tr> <td>uerbetrag</td> <td>Einzelpreis</td> </tr> <tr> <td>_preis</td> <td>Mehrwertste</td> </tr> <tr> <td>_mwst</td> <td>Mehrwertste</td> </tr> <tr> <td>uersatz</td> <td>Menge</td> </tr> <tr> <td>_menge</td> <td>Menge</td> </tr> </table> <p><u>Hinweis:</u> WK_VAR__mwstbetrag! und WK_MWSTBETRAG können verschieden sein</p>	_brutto	Bruttobetrag (Gesamtpreis)	_netto	Nettobetrag (Gesamtpreis)	_mwstbetrag	Mehrwertste	uerbetrag	Einzelpreis	_preis	Mehrwertste	_mwst	Mehrwertste	uersatz	Menge	_menge	Menge
_brutto	Bruttobetrag (Gesamtpreis)																		
_netto	Nettobetrag (Gesamtpreis)																		
_mwstbetrag	Mehrwertste																		
uerbetrag	Einzelpreis																		
_preis	Mehrwertste																		
_mwst	Mehrwertste																		
uersatz	Menge																		
_menge	Menge																		

			(Rundungsproblem), weil bei WK_VAR__mwstbetrag! die einzelnen Mehrwertsteuerbeträge addiert werden und bei WK_MWSTBETRAG die Bruttobeträge addiert werden und daraus die Mehrwertsteuer berechnet wird
WK_COUNT	int	Anzahl	Anzahl der Elemente im Warenkorb
WK_CLEAR		Warenkorb löschen	löscht den angegebenen Warenkorb
WK_MWSTZEILE	Liste	Mehrwertsteuerliste	gibt die Liste der Mehrwertsteuerstaffeln aus
WK_MWSTBETRAG	Preis	Mehrwertsteuerbetrag	Summe der Mehrwertsteuerbeträge der Mehrwertsteuerstaffeln (auf Basis der Bruttosummen)
WK_NETTO	Preis	Nettobetrag	Summe der Nettobeträge der Mehrwertsteuerstaffeln
WK_BRUTTO	Preis	Bruttobetrag	Summe der Bruttobeträge der Mehrwertsteuerstaffeln

Tabelle 3 - HTML-Platzhalter für Warenkorbdaten

Parameter	Typ	Wert	Beschreibung
WE_ID	int	Warenkorbeintrags-ID	Warenkorbeintrags-ID
WE_ELID	int	Elemente-ID	ID des Elementes im Warenkorb
WE_NAME	string	Elementenname	Name des Elementes im Warenkorb
WE_VAR_[Variablenname]!		Warenkorbvariable	Darstellung einer speziellen Warenkorb- oder Elementevariablen <u>spezielle Warenkorbvariablen:</u> _brutto Bruttobetrag (Gesamtpreis) _netto Nettobetrag (Gesamtpreis) _mwstbetrag Mehrwertsteuerbetrag _preis Einzelpreis _mwst Mehrwertsteuersatz _menge Menge
WE_WKORBID	int	Warenkorb-ID	ID des Warenkorbes, in dem dieser Warenkorbeintrag ist
WE_MENGE	int	Menge	Menge des Warenkorbeintrages
WE_MENGE++	int	Menge + 1	Menge des Warenkorbeintrages + 1
WE_MENGE--	int	Menge - 1	Menge des Warenkorbeintrages - 1
WE_PREVEL	int	Elemente-ID	ID des vorherigen Elementes im Warenkorb
WE_NXTTEL	int	Elemente-ID	ID des nächsten Elementes im

			Warenkorb
--	--	--	-----------

Tabelle 4 - HTML-Platzhalter für einen Warenkorbbeitrag

Parameter	Typ	Wert	Beschreibung
MW_BRUTTO	Preis	Bruttobetrag	Bruttobetrag der aktuellen Mwst-Staffel
MW_NETTO	Preis	Nettobetrag	Nettobetrag der aktuellen Mwst-Staffel
MW_MWSTBETRAG	Preis	Mwst-Betrag	Mwst-Betrag der aktuellen Mwst-Staffel
MW_MWSTSATZ	Preis	Mwst-Satz	Mwst-Satz der aktuellen Mwst-Staffel
MW_ANZEL	int	Elementanzahl	Anzahl der Elemente im Warenkorb, die zur aktuellen Mehrwertsteuerstaffel gehören

Tabelle 5 - HTML-Platzhalter für eine Mehrwertsteuerzeile

13 Parameter für Kunden (Befehlsklasse KD)

Parameter	Typ	Wert	Beschreibung
KD_ID	string	Sitzungs-ID	ID des Kunden während der aktuellen Sitzung
Individuelle Kundenvariablen			
KD_VAR_[varname]!	string	individueller Wert	Ausgabe eines über kd_[varname], im Webnutzerformular oder über #SET kd_[varname] = "..." #ENDSET definierter Parameters Ist die Variable vom Typ Text (DLL) findet innerhalb von <a>-Tags ein Preparing statt. Bspw.: KD_VAR_hausnummer!
KD_VAR_[varname]*length!	int	Variable	Länge des Variableninhaltes Anzahl der Zeichen)
Geschützter Webnutzerebereich			
KD_LOGIN_[Unterlayout]!	string	Layout	Das unter "Unterlayout" hinterlegte Layout wird angezeigt, wenn der Kunde in das System eingeloggt ist.
KD_NOLOGIN_[Unterlayout]!	string	Layout	Das unter "Unterlayout" hinterlegte Layout wird angezeigt, wenn der Kunde nicht in das System eingeloggt ist.
KD_CHECKLOGIN	int	[0/1]	Liefert 1, wenn in der Sitzung ein gültiges Webnutzerlogin gespeichert ist, ansonsten 0.

Tabelle 6 - HTML-Platzhalter für individuelle Kundendaten

14 Parameter für Webnutzergruppen (Befehlsklasse WG)

Parameter	Typ	Wert	Beschreibung
WG_ID	int	Webnutzergruppen-ID	ID der Webnutzergruppe
WG_NEXTWG	int	Webnutzergruppen-ID	ID der nächsten Webnutzergruppe
WG_PREVWG	int	Webnutzergruppen-ID	ID der vorherigen Webnutzergruppe
WG_WUANZAHL	int	Anzahl	Anzahl enthaltener Webnutzer (es werden nur Webnutzer gezählt, die aktuell Zugriffsberechtigung haben; gesperrte Webnutzer werden nicht mitgezählt)
WG_WGANZAHL	int	Anzahl	Anzahl enthaltener Webnutzergruppen; nur die Startgruppe enthält Webnutzergruppen (aber keine Webnutzer), die anderen Webnutzergruppen enthalten ausschließlich Webnutzer (aber keine weiteren Webnutzergruppen)
WG_NAME	string	Webnutzergruppenname	Name der Webnutzergruppe
WG_FIRSTWU	int	Webnutzer-ID	ID des ersten sichtbaren Webnutzers in dieser Webnutzergruppe (im Fehlerfall -1)
WG_LASTWU	int	Webnutzer-ID	ID des letzten sichtbaren Webnutzers in dieser Webnutzergruppe (im Fehlerfall -1)
WG_FIRSTWG	int	Webnutzergruppen-ID	ID der ersten Webnutzeruntergruppe (im Fehlerfall -1)
WG_LASTWG	int	Webnutzergruppen-ID	ID der letzten Webnutzeruntergruppe (im Fehlerfall -1)
WG_FL	int	fortlaufende Nr.	fortlaufende Nummer bei Webnutzergruppenlisten
WG_CYLFARBE	string	Farbwert	abwechselnde Farbe bei Listendarstellungen
WG_ZEILE_[layoutname]!	Liste	Liste der Webnutzergruppen	Darstellung der Liste der Webnutzergruppen (die Angabe des Parameters layoutname ist optional)
WG_WUZEILE_[layoutname]!	Liste	Liste der Webnutzer	Darstellung der Liste der Webnutzer (die Angabe des Parameters layoutname ist optional)

Tabelle 7 - HTML-Platzhalter für Webnutzerguppen

15 Parameter für Webnutzer (Befehlsklasse WU)

Parameter	Typ	Wert	Beschreibung
WU_ID	int	Webnutzer-ID	ID des aktuellen Webnutzers
WU_NEXTWU	int	Webnutzer-ID	ID des nächsten sichtbaren Webnutzers
WU_PREVWU	int	Webnutzer-ID	ID des vorherigen sichtbaren Webnutzers
WU_VAR_[varname]!		Webnutzervariable	Ausgabe einer auf dem Webnutzerformular erfassten Variable (oder einer internen Webnutzervariable) Hinweis: Über die Variable <code>_lg</code> kann das Login und über <code>_pw</code> kann das Passwort eines Webnutzers ausgegeben werden. Bei unachtsamer Integration kann die Verwendung der Variable <code>_pw</code> ein Sicherheitsproblem darstellen!
WU_NAME	string	Name des Webnutzers	Ausgabe des Namen des Webnutzers (Variable <code>_na</code>)
WU_CYLFARBE	int	fortlaufende Nr.	fortlaufende Nummer bei Webnutzerlisten
WU_FL	string	Farbwert	abwechselnde Farbe bei Listendarstellungen

Tabelle 8 - HTML-Platzhalter für Webnutzer

16 CGI-Übergabeparameter

Parameter	Typ	Wert	Beschreibung
Standardparameter			
kd	string	SitzungsID	Angabe der Sitzungs-ID zur Wiedererkennung eines Besuchers z.B. für Warenkorbfunktionalität oder geschützte Webnutzerebereiche
or	int	OrdnerID	ID des aktuellen Ordners
el	int	ElementID	ID des aktuellen Elementes
typ	string	Unterlayout	Name des zu verwendenden Unterlayouts
Angaben zur Suche im Datenbestand			
search_id	int	ID des Suchblocks	innerhalb eines #SEARCH-Statements wird mit search_id die ID des Suchblocks angegeben
search_name	string	Elementbezeichnung	Zu suchende Elementbezeichnung (entspricht search_varname0001 = _na)
search_ordner	int	Ordner	Element oder Ordner muss in diesem Ordner stehen.
search_stop	string	Liste von OrdnerIDs	Liste von OrdnerIDs (durch Komma getrennt); bei diesen Ordnern wird die Suche sofort abgebrochen, d.h. darunter liegende Ordner/Elemente werden nicht durchsucht
search_use1st	int	0 1	0 : Der in search_ordner angegebene Wurzelordner wird nicht in die Suche einbezogen. 1 : Der in search_ordner angegebene Wurzelordner wird einbezogen.
search_max	int	Maximale Trefferzahl	Suche nach Ordnern bzw. Elementen wird abgebrochen, wenn search_max erreicht wurde.
search_exact	int	0 1	1 = nur exakte Begriffsübereinstimmung in Element suchen
search_varname####	string	Variablenname	Zu durchsuchender Variablenname oder mit Komma getrennte Liste von Variablen, welche nach dem Suchbegriff durchsucht werden sollen. Im letzten Fall muss der Suchbegriff lediglich in einer der Variablen vorhanden sein.
search_vartyp####	int	100 : Textfeld 101 : Text (DLL) 102 : Datei 103 : Textauswahl 200 : Integer 201 : Gleitkomma 202 : Preis 203 : Prozent 204 : Wahrheitsfeld 208 : Datum	Typ der zu durchsuchenden Variable Hinweis: Soll eine Liste von Variablen nach einem Suchbegriff durchsucht werden, muss hier lediglich der Obertyp (100, 200 oder 300) angegeben werden. Format:

		209 : Zeit 210 : Farbauswahl	208 : „yyyymmdd“ 209 : „hhmmss“
search_varquantor####	int	0 : include 1 : exclude	Soll der Suchparameter in der Variable gefunden oder nicht vorhanden sein?
search_varcsense####	int	0 : nein 1 : ja	Soll für die Variable die Groß-/Kleinschreibung beachtet werden?
search_varexact####	int	0 : nein 1 : ja	Soll für die Variable eine exakte Suche durchgeführt werden?
search_varbegin####	int	0 : nein 1 : ja	Muss der Suchbegriff am Anfang des Variableninhaltes stehen?
search_varmulti####	int	0 : nein 1 : ODER-Verknüpf. 2 : UND-Verknüpf.	Wenn 1 oder 2, dann besteht Suchtext (search_varmin####) aus mit Komma getrennter Liste von Suchbegriffen. Ein Element wird gefunden, wenn es mind. einen der Suchbegriffe enthält (Oder-Verknüpfung) bzw. aller der Suchbegriffe enthält (UND-Verknüpfung).
search_varmin####	string	Mindestwert	Mindestwert für zu suchende Variable
search_varmax####	string	Maximalwert	Maximalwert für zu suchende Variable
search_sortvar1 search_sortvar2	string	Sortiervariable1 Sortiervariable2	Variablenname nach dem sortiert werden soll na : Elementenname
search_sorttyp1 search_sorttyp2	int	Sortierrichtung1 0 1 2 Sortierrichtung2 0 1	Richtung der Sortierung: 0 : Aufwärts A – Z 1 : Abwärts Z - A 2 : zufällige Sortierung
search_grname	string	Ordnername	Zu suchender Ordnername
search_grexact	int	0 1	1 = nur exakte Begriffsübereinstimmung in Ordnern suchen
search_grvarname####	string	Variablenname	Zu durchsuchender Variablenname oder mit Komma getrennte Liste von Variablen, welche nach dem Suchbegriff durchsucht werden sollen. Im letzten Fall muss der Suchbegriff lediglich in einer der Variablen vorhanden sein.
search_grvartyp####	int		(siehe search_vartyp – hier lediglich für Ordner)
search_grvarquantor####	int	0 : include 1 : exclude	Soll der Suchparameter in der Variable gefunden oder nicht vorhanden sein?
search_grvarcsense####	int	0 : nein 1 : ja	Soll für die Variable die Groß-/Kleinschreibung beachtet werden?
search_grvarexact####	int	0 : nein 1 : ja	Soll für die Variable eine exakte Suche durchgeführt werden?
search_grvarbegin####	int	0 : nein 1 : ja	Muss der Suchbegriff am Anfang des Variableninhaltes stehen?
search_grvarmulti####	int	0 : nein 1 : ODER-Verknüpf. 2 : UND-Verknüpf.	Wenn 1 oder 2, dann besteht Suchtext (search_grvarmin####) aus mit Komma getrennter Liste von Suchbegriffen. Ein Ordner wird

			gefunden, wenn er mind. einen der Suchbegriffe enthält (Oder-Verknüpfung) bzw. aller der Suchbegriffe enthält (UND-Verknüpfung).
search_grvarmin####	string	Mindestwert	Mindestwert für zu suchende Variable
search_grvarmax####	string	Maximalwert	Maximalwert für zu suchende Variable
search_grsortvar1 search_grsortvar2	string	Sortiervariable1 Sortiervariable2	Variablenname nach dem sortiert werden soll na : Ordnername
search_grsorttyp1 search_grsorttyp2	int	Sortierrichtung1 0 1 2 Sortierrichtung2 0 1	Richtung der Sortierung: 0 : Aufwärts A – Z 1 : Abwärts Z - A 2 : zufällige Sortierung
search_start	int	Blockstart	Nummer des Elementes oder Ordners, mit dem Suchergebnisse beginnen sollen (0 : Beginn)
search_block	int	Blockumfang	Anzahl anzuzeigender Suchergebnisse
search_hidden	int	Versteckte Ordner durchsuchen	Angabe, ob versteckte Ordner (ohne Internetfreigabe) in die Suche einbezogen werden sollen. Die Ordner selbst werden nicht(!) gefunden aber durchsucht. 0 : (Default) versteckte Ordner nicht durchsuchen 1 : versteckte Ordner durchsuchen
Geschützter Webnutzerebereich			
kd_login	string	Webuser-Login	Loginname eines Webusers, welcher sich damit für die folgende Sitzung authentifizieren möchte.
kd_passwd ¹	string	Webuser-Passwort	Passwort eines Webuser, welcher sich in Zusammenhang mit Loginname für die folgende Sitzung authentifizieren möchte.
Individuelle Kundenparameter			
kd_[name]	string	Initialisierung eines Kundenparameter	Alle Übergabeparameter, die mit kd_ beginnen (z.B. in Links oder Formularfelder) werden in der Sitzung als sogenannte Kundenvariablen gespeichert und können jederzeit über #INSERT_KD_VAR_[Variablenname]! wieder abgerufen werden.
Angaben zur Ablage von Elementen im Warenkorb			
wkorb####	int	ID des Warenkorb	Übergabe der ID, in welchen Warenkorb das Element abgelegt werden soll; die
el####	int	Elemente-ID	ID des Elementes, welches in den Warenkorb gelegt werden soll
menge####	int	Menge	Menge, in der das Element in den Warenkorb gelegt werden soll (Menge>0)

¹ Bitte beachten Sie: Das Paßwort wird hier unverschlüsselt übertragen! Daher sollte zumindest das POST-Verfahren zur Übermittlung von sicherheitsrelevanten Formulardaten genutzt werden (bspw. <form action="#INSERT_SP_CGI" method="post">...</form>).

eur####	int	individ. EUR-Preis (Bruttopreis)	individueller Preis (Bruttopreis), mit dem das Element in den Warenkorb gelegt werden soll; die Option wkorb_pricing muss gesetzt sein (yes); andernfalls wird der Standardpreis (Variable_pe) verwendet Beispiel: eur0001 = 7,85
mwst####	int	individ. MwSt.	individueller Mehrwertsteuersatz; wird kein Mehrwertsteuersatz übergeben, wird die Elementvariable _mw abgefragt oder der Mehrwertsteuersatz aus dem Systemparameter mwst gelesen Beispiel: mwst0001 = 16,00
wem#	int	Mengenänderung	Mengenänderung eines bestehenden Warenkorbeintrages; # steht dabei für die ID des Warenkorbeintrages (#INSERT_WE_ID); bei Übergabe der Menge 0 wird das Element aus dem Warenkorb entfernt
es#####	string	gewählte Auspräg.	##### - ID des Elementes #### - lf. Nr. der Eigenschaft
weap#####	string	Ausprägung ändern	Ändert die Ausprägung einer Eigenschaft eines Eintrages im Warenkorb: #### - ID des Warenkorbeintrages #### - lfd. Nr. der Eigenschaft
mmcheck	string	absoluter Multimediapfad	Bei Übergabe des Parameters mmcheck mit einem gültigen absoluten Multimediapfad werden alle Dateien dieses Verzeichnisses auf Vorhandensein in der VIO.Matrix-Datenbank (in Ordner-, Elemente- oder Kundenformular sowie als statisches Bild im Layout) durchsucht. Je nach Angabe von mmcheck_output werden alle Bilder, nur die vorhandenen oder nur die nicht vorhandenen Bilder aufgelistet. Standardmäßig werden nur die nicht vorhandenen Bilder aufgelistet (mmcheck_output=0). Um Missbrauch zu vermeiden, ist das Verzeichnis /cgi-bin sowie ein Wechsel in übergeordnete Verzeichnisse über ../ nicht möglich. Außerdem kann über den Systemparameter mmcheckpath ein zwingender Pfadbestandteil angegeben werden.
mmcheck_output	string	Typ der Multimedialiste [0/1/2]	Werte: 0: Ausgabe aller nicht in der VIO.Matrix-Datenbank gefundenen Dateien 1: Ausgabe aller Dateien 2: Ausgabe aller gefundenen Dateien

Tabelle 9 - Übergabeparameter aus dem Web

17 Systemparameter

Parameter	Typ	Wert	Beschreibung
Position der Datenbankdateien			
ElementDB	string	Dateiname	Name der Elementedatenbank (ohne .DAT)
OrdnerDB	string	Dateiname	Name der OrdnerDB (ohne .DAT)
RabatteDB	string	Dateiname	Name der Rabattedatenbank (ohne .DAT)
webuser	string	Dateiname	Name der Kundendatenbank (ohne .DAT)
webusergroup	string	Dateiname	Name der Kundengruppendatenbank (ohne .DAT)
cylfarbe1 cylfarbe2	string	Farbwert	Angabe von zwei Farbwerten, welche in Listen abwechselnd angezeigt werden können.
clientpath	string	Pfadname	Name des Pfades (ausgehend von vio.matrix-Verzeichnis) in welches die einzelnen Kundendateien gelegt werden.
Parameter für die Webstatistik			
logfile	string	Logfilename	Dateiname des zu erstellenden Logfiles (bspw. matrix.log). Jeder Aufruf von VIO.Matrix wird daraufhin protokolliert. Dazu wird das in logformat spezifizierte Format bzw. im Standardfall das CLF (Common Log Format) verwendet. Das Logfile kann anschließend von jedem Weblog-Analyser gelesen werden (bspw. webalizer, wusage,..).
logformat	string	Logformat	Formatierung einer Zeile des Logfiles im CLF. Derzeit werden folgende Formatanweisungen unterstützt (ansonsten wird ein „-“ eingesetzt): %b Größe des angezeigten Layout in Byte %h Remotehost (Name oder IP) %t Zeitpunkt des Request im Standardformat ([Tag/Monat/Jahr:Stunde:Minute:Sekunde GMT Abweichung]) %r Request in der Form (O E S):Titel wobei O (Ordner), E (Element), S (Sonstiges) und Titel den Titel des Ordners/Elementes beinhaltet. %s Statuscode 200 – OK, 400 – kein Layout vorhanden, 404 – kein gültiger Ordner/Element
Parameter zur Warenkorbverwaltung			
wkorb_uniqueelements	string	Warenkorbelemente	"yes" ein Element (ID) kann mehrfach in den Warenkorb gelegt werden "no" jedes Element (ID) kann nur einmal im Warenkorb liegen
wkorb_pricing	string	individuelle Preise	"yes" über den Parameter eur#### dürfen Elemente mit individuellem Preis

			in den Warenkorb gelegt werden; wird eur#### nicht angegeben, wird der Standardpreis verwendet "no" alle Elemente werden bei der Übernahme in den Warenkorb mit dem Standard-Euro-Preis übernommen
mwst	string	Standard-MwSt	Angabe der Standardmehrwertsteuer (z.B. 16,00)
Multimedialiste			
mmcheckpath	string	Pfadangabe	Pfadangabe, die bei Verwendung des CGI-Übergabeparameters mmcheck zwingend enthalten sein muss. Dadurch kann eine missbräuchliche Nutzung, d.h. das missbräuchliche Auslesen anderer Verzeichnisse, verhindert werden.
HTML-Vorschau			
domain_preview	string	Domain	Domain, unter der lokal nicht vorhandene Bilder im Web gefunden werden können.

Tabelle 10 - Wichtige Parameter der "data.txt"

18 Sonstige Parameter (Befehlsklasse SP)

Parameter	Typ	Wert	Beschreibung
PARAM_[Varname]!	string	Parameter	Liefert den Inhalt des Systemparameters. In Ordner- und Elementezeilen muss dazu der Befehl #PAR_[name]! genutzt werden.
SP_EXEC_[Programm]!	string	Prg.-aufruf	Startet das angegebene Programm mit den entsprechenden Parametern und fügt die Ausgabe an Stelle des Parameters. z.B.: SP_EXEC_cat hallo.txt!
			Hinweis: Innerhalb des Programmaufrufes können VIO.Matrix HTML-Parameter eingefügt werden. Bsp: #INSERT_SP_EXEC_cat /messages/{#INSERT_KD_file!}!
SP_CGI	string	CGI-Aufruf	Fügt den Inhalt der Variable „shopcgi“ ein. Diese enthält den Aufruf des viodb-CGI.
SP_VAR_[varname]!	string	temporäre Variable	Anzeige einer temporären Variablen, deren Gültigkeit sich nur über das aktuelle Layout erstreckt.
SP_VAR_[varname]*length!	int	Variable	Länge des Variableninhaltes (Anz. d. Zeichen)
SP_ENVVAR_[Umgeb.-var.]!	string	Umgebungsvariable	Ausgabe einer Umgebungsvariable, z.B.: #INSERT_SP_ENVVAR_REMOTE_ADDR!
Datumsangaben			
SP_DATE	string	aktuelles Datum	Fügt das aktuelle Datum im Format TT.MM.JJJJ ein.
SP_MDAY	string	aktueller Tag	Fügt den aktuellen Tag im Format TT ein.
SP_WDAY	string	aktueller Wochentag	Fügt den aktuellen Wochentag (0-6) ein. 0=Sonntag, 1=Montag, ...
SP_MONTH	string	aktueller Monat	Fügt den aktuellen Monat im Format MM ein.
SP_YEAR	string	aktuelles Jahr	Fügt das aktuelle Jahr im Format JJJJ ein.
SP_YY	string	aktuelles Jahr	Fügt das aktuelle Jahr im Format JJ ein.
SP_HOUR	string	aktuelle Stunde	Fügt das aktuelle Jahr im Format HH ein.
SP_MIN	string	aktuelle Minute	Fügt das aktuelle Jahr im Format MM ein.
SP_SEC	string	aktuelle Sekund.	Fügt das aktuelle Jahr im Format SS ein.
Zufällige Passwortvergabe			
SP_RANDOMPASSWORD_[Länge]!	string	zufälliges Passwort	fügt ein zufälliges, aus Groß- und Kleinbuchstaben sowie Ziffern bestehendes Passwort in der gewünschten Länge ein

Die in dieser Dokumentation beschriebene Software und Hardware wird Ihnen gemäß den Bedingungen eines Lizenzabkommens zur Verfügung gestellt und darf nur unter den darin beschriebenen Bedingungen eingesetzt werden.

Diese Dokumentation ist urheberrechtlich geschützt. © 2004 VIOSYS AG.
Alle Rechte vorbehalten.

Alle in diesem Handbuch erwähnten Beispiele mit Namen, Firmennamen oder Firmen sind frei erfunden. Ähnlichkeiten mit tatsächlichen Namen, Firmen oder Einrichtungen sind unbeabsichtigt und rein zufällig.

Die Dokumentation wird in der vorliegenden Form geliefert, und die VIOSYS AG übernimmt keine Garantie für die Genauigkeit oder die Verwendung. Die Verwendung der Dokumentation oder der darin enthaltenen Informationen hat der Benutzer zu verantworten. Die Dokumentation kann technische oder andere Ungenauigkeiten oder typografische Fehler enthalten. Die VIOSYS AG behält sich das Recht vor, Änderungen ohne vorherige Ankündigung vorzunehmen.

VIO.Matrix® ist ein eingetragenes Warenzeichen der VIOSYS AG.
Alle anderen Warenzeichen sind Warenzeichen ihrer jeweiligen Inhaber.

Weitere Informationen zu VIO.Matrix® finden Sie auf diesen Webseiten:
www.viosys.de, www.viomatrix.de, www.akademie.viomatrix.de, www.nophp.de.