

Technische Anleitung zur Reportgestaltung

1. Allgemeines

Das Reporting im Calender 3 arbeitet mit Reportvorlagen im Richertextformat (RTF) Format. Dieses Format kann mit sehr vielen Editoren (z. B. Microsoft Wordpad, Microsoft Word) bearbeitet werden.

Die Reportvorlagen werden im Ordner REPORTS des Clients gespeichert und automatisch beim Zuweisen im Calender Client auf den Server kopiert. Beim Drucken wird die jeweilige Reportvorlage automatisch dem Client zur Verfügung gestellt. Die Datei-Endung der Reportvorlagen muss **.rtf** sein.

2. Die Reportvorlage

2.1. Allgemeines zur Formatierung

Alle internen Funktionen und Felder sind in der Reportvorlage mit einem beginnenden Backslash-Zeichen \ und einem endenden Backslash-Zeichen \ gekennzeichnet.

Alles andere ist normaler Text und wird 1:1 in den Druck übernommen.

Ein Report besteht immer aus einem Kopfabschnitt, dann einem sich wiederholenden Abschnitt und dann dem Fußabschnitt.

Dabei wird der sich wiederholende Abschnitt durch das SCAN-Konstrukt definiert. Alles was oberhalb liegt, ist dann der Kopfabschnitt alles was unterhalb liegt ist der Fußabschnitt.

Das SCAN-Konstrukt hat folgende Syntax:

```
\SCAN(),WHILE (AKTIV)\
```

(in dieser Schleife werden dann die Daten ausgegeben)

```
\ENDSCAN\
```

Beispiel:

Der folgende Report druckt eine Liste aller Notizen mit nur dem Notiztext.

```
Liste aller Notizen
\SCAN(),WHILE (AKTIV)\
\TEXT\
\ENDSCAN\
Ende der Liste
```

Der Kopf besteht in diesem Beispiel nur aus dem Text „Liste aller Notizen“ und einer Leerzeile.

Der Fuß besteht aus dem Text „Ende der Liste“.

Der wiederholende Abschnitt besteht aus dem SCAN Konstrukt und der Feldfunktion \TEXT\ und einer Leerzeile. Diese Funktion gibt den Text der Notiz aus. Die Feldfunktionen werden später genau beschrieben.

Die Ausgabe sieht dann folgendermaßen aus:

```
Liste aller Notizen
Dies ist meine erste Notiz
Dies ist meine zweite Notiz
Dies ist meine dritte Notiz
Ende der Liste
```

Der Wiederholungsabschnitt wiederholt sich solange, solange Daten zur Ausgabe vorhanden sind. Die Datenauswahl und Sortierung der Daten erfolgt im Drucken-Bildschirm des Calenders.

2.3. Arbeiten mit Variablen

Wir wollen nun die Notizen nach Anlagedatum sortiert ausgegeben und bei jedem neuen Datum soll eine Überschrift erscheinen.

```
Liste aller Notizen

Angelegt am 5.2.2005
Dies ist meine erste Notiz

Dies ist meine zweite Notiz

Angelegt am 7.2.2005
Dies ist meine dritte Notiz

Ende der Liste
```

Dazu muss man mit Variablen arbeiten.

```
\VAR (Anlagedatum)\ \SET(Anlagedatum,"")\
Liste aller Notizen
\SCAN(),WHILE (AKTIV)\
\if (Anlagedatum <> Angelegt)\
Angelegt am \Angelegt\ \SET(Anlagedatum,Angelegt)\ \endif\
\TEXT\

\ENDSCAN\
Ende der Liste
```

Durch das Konstrukt

```
\VAR(Anlagedatum)\
```

wird eine Variable namens Anlagedatum deklariert.

Deklariert bedeutet ab jetzt kann ich auf diese zugreifen, hat aber noch keinen Inhalt.

Mit `\SET(Anlagedatum,"")\`

wird nun der Variablen Anlagedatum ein Wert zugeordnet, hier der Wert "", eine leere Zeichenkette, also nichts.

Die Variable wird benötigt, da in dieser Variablen das letzte gedruckte Anlagedatum gespeichert werden soll.

Vor jeder Ausgabe des Notiztextes wird nun mit dem Konstrukt

```
\if (Anlagedatum <> Angelegt)\
```

```
\endif\
```

geprüft ob das Anlagedatum der Notiz, welches im Feld Angelegt gespeichert ist, sich von dem letzten Wert, welcher in der Variablen Anlagedatum steht, unterscheidet.

Wenn das so ist, dann wird der Text Angelegt am \Angelegt\ ausgegeben, wo \Angelegt\ ersetzt wird durch das Anlagedatum der Notiz.

Jetzt müssen wir nur noch die Variable Anlagedatum auf den aktuellen Wert des Anlagedatums der Notiz setzen.

Dies passiert mit dem Befehl

```
\SET(Anlagedatum,Angelegt)\
```

wo der Variablen der Wert des Feldes Angelegt zugewiesen wird.

Somit ist die Abfrage fertig gestellt.

Jetzt gibt es aber noch ein Problem. Im Feld Angelegt der Notiz steht nicht nur das Datum, sondern auch die Uhrzeit. Jetzt würde aber die Überschrift „Angelegt am“ immer kommen, da die Uhrzeit ja meist verschieden ist.

Somit müssen wir im Feld Angelegt nur das Datum und nicht die Uhrzeit auswerten.

Das Datum steht in der Variable Angelegt am Anfang und somit betrachten wir einfach die ersten 10 Zeichen der Variable. Dies funktioniert mit der internen Funktion COPY mit folgender Syntax:

```
\COPY(Variable,Startposition,Anzahl der Zeichen)\
```

wobei Variable das zu formatierende Feld ist, Startposition das 1. Zeichen ist, ab dem kopiert wird, und Anzahl der Zeichen die Anzahl der Zeichen ist, die kopiert werden.

Somit stellt `\COPY(Angelegt,1,10)\` die Variable Angelegt nur als Datum dar.

Jetzt wird die Funktion noch in den Vergleich (if), in die Ausgabe und in die Zuweisung gestellt und wir sind fertig:

```
\var (Anlagedatum)\ \set(Anlagedatum,"")\
Liste aller Notizen
\SCAN(),WHILE (AKTIV)\
\if (Anlagedatum <> COPY(Angelegt,1,10))\
Angelegt am \COPY(Angelegt,1,10)\ \SET(Anlagedatum,
COPY(Angelegt,1,10))\ \endif\
\TEXT\

\ENDSCAN\
Ende der Liste
```

Interessant ist hier noch, wenn wir Variablen in Funktionen hinein nehmen, entfällt der führende Backslash \ und der endende Backslash, da die Funktion ja schon mit Backslashes eingebunden ist.

Technische Anleitung zur Reportgestaltung

3. Feldfunktionen

3.1. Termine

Alle Feldfunktionen sind bezüglich eines einzelnen Termins zu sehen

Feldfunktion	Beschreibung
Terminart	Typ des Termins TER steht für normalen Termin, TS steht für eine Terminserie TSA steht für einen geänderten Termin einer Terminserie
Benutzername	Vollständiger Name des Benutzers
Benutzerlogin	Logins (3stellig) des Benutzers
Tag Terminbeginn	Datum, Beginn des Termins
Zeit Terminbeginn	Zeitbeginn des Termins
Beginn	Datum und Zeit, Beginn des Termins
Tag Terminende	Datum Ende des Termins
Zeit Terminende	Zeitende des Termins
Ende	Datum und Zeitende des Termins
Termindauer	Termindauer in Minuten
Termindauer in Tagen	Termindauer in Tagen
Erinnerung	Zeit wieviele Minuten/Stunden vorher an den Termin erinnert wird
Email-Erinnerung	Ja oder Nein ob eine Erinnerung erfolgt
Erinnerung Email erfolgt	Ja oder Nein ob eine Erinnerung schon erfolgt ist
Terminanfrage	Text ob der Termin angefragt, abgelehnt oder akzeptiert wurde
Anfragesteller	Vollständiger Name des Antragstellers
Anfrage gestellt	Datum und Uhrzeit wann die Anfrage gestellt wurde
Betreff	Betreff des Termins
Text	Text des Termins
Kategorie	Name der Kategorie
Priorität	Priorität
Ort	Ort
privat	Ja oder Nein ob Termin privat ist
Wiederholen	Beschreibung der Serienterminwiederholung
Geändert	Datum und Uhrzeit wann der Termin das letzte Mal geändert wurde
Geändert_von	Vollständiger Name des Benutzers, der den Termin geändert hat
Angelegt	Datum und Uhrzeit wann der Termin angelegt wurde
Angelegt_von	Vollständiger Name des Benutzers, der den Termin angelegt hat
Status	Status des Termins (erledigt)
Teilnehmer	Namen der Teilnehmer
Zusatzfeld_1	Inhalt des 1. Individualfeldes
Zusatzfeld_2	Inhalt des 2. Individualfeldes
Zusatzfeld_3	Inhalt des 3. Individualfeldes

3.2. Aufgaben

Alle Feldfunktionen sind bezüglich eines einzelnen Termins zu sehen

Feldfunktion	Beschreibung
Aufgabenart	Typ der Aufgabe Allgemeine Aufgabe, privat, delegierte ..
Aufgabe_für	Vollständiger Name des Benutzers für den die Aufgabe ist
Aufgabe_von	Vollständiger Name des Benutzers von dem die Aufgabe erstellt wurde
Beginn	Datum wann die Aufgabe beginnen soll
fällig_am	Datum wann die Aufgabe fällig ist
erledigt_am	Datum wann die Aufgabe erledigt wurde
Betreff	Betreff
Text	Text
Kategorie	Kategorie
Priorität	Priorität
Ort	Ort
Geändert	Datum und Uhrzeit wann die Aufgabe das letzte Mal geändert wurde
Geändert_von	Vollständiger Name des Benutzers, der die Aufgabe geändert hat
Angelegt	Datum und Uhrzeit wann die Aufgabe angelegt wurde
Angelegt_von	Vollständiger Name des Benutzers, der die Aufgabe angelegt hat
Status	Status der Aufgabe
Zusatzfeld_1	Inhalt des 1. Individualfeldes
Zusatzfeld_2	Inhalt des 2. Individualfeldes
Zusatzfeld_3	Inhalt des 3. Individualfeldes

Technische Anleitung zur Reportgestaltung

3.3. Notizen

Alle Feldfunktionen sind bezüglich eines einzelnen Termins zu sehen

Feldfunktion	Beschreibung
Notizart	Notiz von mir oder Notiz an mich
Notiz_für	Vollständiger Name des Benutzers für den die Notiz ist
Notiz_von	Vollständiger Name des Benutzers von dem die Notiz erstellt wurde
Text	Text
Geändert	Datum und Uhrzeit wann die Notiz das letzte Mal geändert wurde
Geändert_von	Vollständiger Name des Benutzers der die Notiz geändert hat
Angelegt	Datum und Uhrzeit wann die Notiz angelegt wurde
Angelegt_von	Vollständiger Name des Benutzers, der die Notiz angelegt hat

4. Funktionen

Die folgenden Funktionen ändern keine Werte sondern geben geänderte Werte zurück. Der Funktionsname ist immer **grün**, die Parameter der Funktion sind **blau** gekennzeichnet.

4.1. Spezielle Calendarfunktionen

KW(DATUM)

Gibt die Kalenderwoche des Parameters DATUM zurück. Ohne Parameter liefert dies Funktion die aktuelle Kalenderwoche.

KWlang(DATUM)

Wie die Funktion KW, nur das der Kalenderwoche noch das Jahr vorangestellt wird.

Wochevonbis(DATUM)

Liefert das Datum des ersten Tages in der Woche in der der Parameter DATUM liegt gefolgt von einem - Zeichen und dem Datum des letzten Tages in dieser Woche zurück. Der Montag ist hier der erste Tag in der Woche, der Sonntag der letzte.

Wochentagkurz(DATUM)

Liefert den kurzen Wochentagnamen des Parameters DATUM zurück. z.B. Mo, Di ..

Wochentaglang(DATUM)

Wie Wochentagkurz, nur hier wird der komplette Name des Wochentags zurückgeliefert. z.B. Montag, Dienstag ...

4.2. Allgemeine Funktionen

Verschiedene arithmetische und logische Ausdrücke können in der Reportvorlage verwendet werden.

arithmetische Operatoren:

>, <, =, <=, >=, <>, != (not equal), +, -, *, /, % (mod)

Logische Operatoren:

&& (and), || (or), ! (not)

zum Beispiel:

```
\IF( (Ort <> „Nürnberg“ || (ORT = „München“) ) \
.....
\ENDIF\
```

4.3. Variablenfunktionen

VAR(VAR1, ..., VAR N) – Deklariert die Variablen VAR1, ... bis VAR N wenn Sie nicht existieren

SET(VarName, WERT) – Weist den Wert WERT der Variablen VarName zu.

IIF(Logischer_Ausdruck, WERT1, WERT2) – Liefert einen von zwei Werten abhängig von dem logischen Ausdruck zurück. Wenn der logische Ausdruck wahr ist, dann ist der Rückgabewert WERT1, ansonsten WERT2.

4.4. Zeichenfunktionen

COPY(S,StartPosition,[Optional ANZAHL])

MID(S, StartPosition,[Optional ANZAHL])

SUBSTR(S, StartPosition,[Optional ANZAHL])

Technische Anleitung zur Reportgestaltung

Diese Funktionen geben Zeichenfolgen aus der Zeichenkette **S** zurück.

StartPosition ist die Startposition ab der die Zeichen aus der Zeichenkette heraus kopiert werden. **ANZAHL** ist die Anzahl der Zeichen, die kopiert werden.

FORMATFLOAT(Formatierung, WERT)

FormatFloat formatiert die Gleitkommazahl **WERT** in ein Format, welches durch den Wert in **Formatierung** definiert wird.

STR(ZAHL,[LÄNGE, GENAUIGKEIT])

Liefert eine Zeichenkette zurück, wobei die ZAHL im Parameter **ZAHL** mit der Länge **LÄNGE** und den Nachkommastellen **GENAUIGKEIT** formatiert wird.

VAL(s)

Wandelt die Zeichenfolge s in eine Zahl um.

UPPER(s)

Wandelt die Zeichenfolge s in Grossbuchstaben um.

LOWER(s)

Wandelt die Zeichenfolge s in Kleinbuchstaben um.

TRIM(s)

Entfernt aus s am Anfang und am Ende alle Leerzeichen.

4.5. Numerische Funktionen

ROUND(ZAHL,BREITE)

Rundet die Gleitkommazahl **ZAHL** auf die Länge **BREITE** und gibt eine ganze Zahl zurück.

INT(ZAHL)

Schneidet die Nachkommastellen der Gleitkommazahl ZAHL ab und gibt eine ganze Zahl zurück.

FRAC(ZAHL)

Gibt die Nachkommaanteil der Gleitkommazahl ZAHL zurück.

POWER(BASIS, EXPONENT)

Diese Funktion potenziert die Zahl **BASIS** mit dem Exponenten **EXPONENT**.

INTPOWER(BASIS, EXPONENT)

IntPower gibt das Ergebnis von BASIS hoch **EXPONENT** zurück.

Unterschied zu Power ist, dass **EXPONENT** eine ganze Zahl sein muss.

4.6. Datum und Zeit Funktionen

NOW()

Gibt das aktuelle Datum und die aktuelle Uhrzeit zurück.

DATE()

Gibt das aktuelle Datum zurück.

TIME()

Gibt die aktuelle Zeit zurück.

DATETOSTR(DATUM)

Konvertiert den Datumswert **DATUM** zu einer Zeichenkette und gibt diese zurück.

DATETIMETOSTR(DATUMZEIT)

Konvertiert den Datums- und Zeitwert **DATUMZEIT** zu einer Zeichenkette und gibt diese zurück.

TIMETOSTR(ZEIT)

Konvertiert den Zeitwert **ZEIT** zu einer Zeichenkette und gibt diese zurück.

STRTO DATE(ZEICHENKETTE)

Konvertiert die Zeichenkette **ZEICHENKETTE** in einen Datumswert.

STRTO DATETIME(ZEICHENKETTE)

Konvertiert die Zeichenkette **ZEICHENKETTE** in einen Datums- und Zeitwert.

STRTO TIME(ZEICHENKETTE)

Konvertiert die Zeichenkette **ZEICHENKETTE** in einen Zeitwert.

YEAR(DATUM)

Gibt das Jahr des Parameters **DATUM** zurück. Der Parameter **DATUM** ist ein Datums- und Zeitwert. Der Rückgabewert ist eine Zahl.

MONTH(DATUM)

Gibt das Monat des Parameters **DATUM** zurück. Der Parameter **DATUM** ist ein Datums- und Zeitwert. Der Rückgabewert ist eine Zahl.

DAY(DATUM)

Gibt den Tag des Parameters **DATUM** zurück. Der Parameter **DATUM** ist ein Datums- und Zeitwert. Der Rückgabewert ist eine Zahl.

SYEAR(DATUM)

Gibt das Jahr des Parameters **DATUM** zurück. Der Parameter **DATUM** ist ein Datums- und Zeitwert. Der Rückgabewert ist eine Zeichenkette.

SMONTH(DATUM)

Gibt das Monat des Parameters **DATUM** zurück. Der Parameter **DATUM** ist ein Datums- und Zeitwert. Der Rückgabewert ist eine Zeichenkette.

SDAY(DATUM)

Gibt den Tag des Parameters **DATUM** zurück. Der Parameter **DATUM** ist ein Datums- und Zeitwert. Der Rückgabewert ist eine Zeichenkette.

DTOS(DATUM)

Gibt den Datumsparameter **DATUM** als Zeichenkette in der Form `yyyymmdd` zurück. Dabei ist `yyyy` die Jahreszahl, `mm` der Monat, `dd` der Tag.

STOD(ZEICHENKETTE)

Wandelt die Zeichenkette `ZZEICHENKETTE` in ein Datum um. `ZEICHENKETTE` muß die Form `yyyymmdd` haben.

4.7. Numerische Formatierungs-Funktionen

Numerische Werte können in der Reportvorlage in einer wissenschaftlichen Notation, in einer festen Länge Notation, in einer numerischen Notation und in Währungsnotation dargestellt werden. Dazu dienen die Funktionen `fexp`, `ffix`, `ffixr`, `fnum`, `fnumr` und `fcurl`

z.B. `\fnum(Zahl)\`, `\fcurl(Zahl,2,4)\`, `\fexp(Zahl,10:2)\`

Die allgemeine Syntax der Formatierungsfunktionen ist `\func(Zahl[, [Genauigkeit:]Breite])\`
Wobei `func` der Name der Formatierungsfunktion ist.
Die Parameter in eckigen Klammern sind optional.

Alle folgenden Funktionen liefern eine Zeichenkette zurück.

fexp(Zahl [, [Genauigkeit:] Breite])

Umwandeln in die wissenschaftliche Notation der Form `"-d.ddd...E+dddd"`.

Das Ergebnis beginnt mit einem Minuszeichen, wenn die Zahl negativ ist. Dann mindestens eine Ziffer, gefolgt vom Dezimalseparator. Durch den Parameter `Breite` wird die Anzahl der Ziffern festgelegt. Danach kommt ein `E` gefolgt von einem `+` oder `-` Zeichen und maximal 4 Stellen für den Exponenten. Der Parameter `Genauigkeit` gibt die minimale Anzahl von Stellen im Exponenten an. (muss zwischen 0 und 4 sein)

ffix(Zahl [, [Genauigkeit:] Breite])

Umwandeln in eine feste-Länge-Notation. Die Zahl wird in ein Format der Form `"-ddd.ddd..."` umgewandelt. Das Ergebnis beginnt mit einem Minus, wenn die Zahl negativ ist, dann mindestens eine Ziffer vor dem Dezimalseparator, dann der Dezimalseparator, dann die Nachkommastellen. Die Parameter `Breite` gibt die Anzahl der Stellen hinter dem Komma an, der Parameter `Genauigkeit` die Anzahl vor dem `Breite` Dezimalseparator.

fnum(Zahl [, [Genauigkeit:] Breite])

Umwandlung in ein Format der Form `"-d,ddd,ddd.ddd..."`. Wie die Funktion `ffix`, nur das zusätzlich ein Tausendertrennzeichen mit eingefügt wird.

fcurl(Zahl [, [Genauigkeit:] Breite])

Umwandlung in das Währungsformat. Hier werden die Windowsvorgaben aus der Systemsteuerung =>Ländereinstellungen) => Währung verwendet.

Sollten Sie noch Fragen haben, wenden Sie sich bitte an:

