3 Variablen

3.1 Allgemeines

Variablen werden in

- Prozeduren,
- Mustern und
- Parameter-Dokumenten

definiert und verwendet und bei der Jobgenerierung durch die Werte, die ihnen zugewiesen werden, ersetzt.

Variablen werden zur Steuerung der Prozeduren verwendet, um (allgemeine) Muster an spezielle Aufgaben anzupassen. Darüber hinaus können sie zur Steuerung der Jobgenerierung - z.B. durch Schleifen oder die bedingte Generierung von Musterblöcken - verwendet werden.

3.2 Definition und Verwendung von Variablen

Eine Variable muss, ehe sie verwendet werden kann, zunächst in einem Dokument bekanntgemacht, d.h. als Variable **definiert** werden.

Bei der Definition wird ihr zugleich ein Anfangswert zugewiesen (Initialisierung). Eine Variable kann einen **einzelnen Wert** oder **mehrere Werte** (in einer oder mehreren Zeilen) erhalten; in diesem Fall handelt es sich um eine Listen- oder Tabellenvariable.

Eine definierte Variable kann auf verschiedene Weise **verwendet** werden:

- Sie kann einem anderen Ausdruck (z.B. einem Sprachelement in einer Prozedur oder einer anderen Variable) **zugewiesen** werden, der zur Testausführung ausgewertet wird und den aktuellen Wert der Variablen erhält.
- Sie kann in einem Muster **selbständig**, d.h. ohne Zuweisung an einen anderen Ausdruck, eingesetzt werden. Sie wird dann z.B. durch eine häufig benötigte Befehlsfolge oder eine aktuelle Pfadangabe ersetzt.

3.2.1 Variablen in Prozeduren

In Prozeduren werden Variablen durch ein "#$" vor dem Variablennamen **definiert**. Diese Definition kann in folgenden Bereichen erfolgen:

#GLOBAL

#PROCESSGROUP

#PROCESS

#STEP

Bei der anschließenden **Verwendung** in diesem Dokument werden Variablen durch ein "$" vor dem Variablennamen gekennzeichnet.

Beispiel

```
Definition: #$PARA = Testumgebung
Verwendung: #PARAMETER = $PARA
```

3.2.2 Variablen in Mustern

In Mustern werden Variablen durch ein "#$" vor dem Variablennamen **definiert**.

Bei der **Verwendung** in diesem Dokument werden Variablen durch ein "$" vor dem Variablennamen gekennzeichnet.

3.2.3 Variablen in Parameter-Dokumenten

Parameter-Dokumente dienen ausschließlich der **Definition** von Variablen.

In den Parameter-Dokumenten werden Variablen nur durch ein "$" vor dem Variablennamen **definiert.**

Diese Definition erfolgt durch eine direkte Wertzuweisung oder durch Zuweisung einer oder mehrerer anderer Variablen.

Beispiel

Direkte Wertzuweisung:

```
$VAR                    =     Wert
```

Zuweisung einer anderen Variablen:

```
$VAR                    =     $VAR1
```

Wenn einer Variablen eine Listenvariable zugewiesen wird, deren Werte wiederum Variablen enthalten, müssen alle Werte in einfache oder doppelte Hochkommata (' ' oder " ") gesetzt sein:

Beispiel:

```
$VAR = $VAR1
```

setzt voraus:

```
$VAR1 = *
'$DOKU'
'$PATH'
'Müller'
```

3.2.4 Format

Folgende formale Regeln gelten generell:

- Der Variablenname kann 40 Stellen umfassen.
- Nicht im Namen zugelassene Zeichen sind:
  `Blank $ , ; ' " % / : (`

  Doppelpunkt und geöffnete Klammer werden für Sonderfunktionen bei der Verwendung der Variablen (Formatierung und Index) benötigt.
- Bei der **Verwendung** werden Variablennamen durch einen Punkt "." ausdrücklich **abgeschlossen**. Dies ist nur erforderlich, wenn weitere (zulässige) Zeichen folgen (z.B. ein beliebiger Text). Wird dieser explizite Abschluss nicht gewählt, endet der Variablenname durch ein beliebiges nicht im Namen zugelassenes Zeichen.

  Beispiele

  ```
  $VAR1 = Text
  $VAR2 = $VAR1.beispiel
  ```

  Die Variable $VAR2 enthält jetzt den Inhalt "Textbeispiel". Der "." ist zum expliziten Abschluss der Variablen §VAR1 in der Zuweisung an $VAR2 notwendig.

  ```
  $VAR1 = Text
  $VAR2 = beispiel
  $VAR3 = $VAR1$VAR2
  ```

  Die Variable $VAR3 enthält jetzt ebenfalls den Inhalt "Textbeispiel". Ein Punkt wie im ersten Beispiel ist nicht erforderlich.

- Durch Anhängen eines **Doppelpunktes** kann zusätzlich eine **Formatierung** festgelegt werden:

  $<Variable>:n linksbündig
  $<Variable>:-n rechtsbündig

- Um auf ein Feld einer Listenvariable zuzugreifen, wird der **Index** des Feldes in **Klammern** angehängt.

  Beispiel:

  $VAR(3) greift auf das 3. Feld der Werteliste einer Listenvariablen zu.

- Um ein **"$" als Textkonstante** in Prozeduren, Mustern und Parametern darzustellen, wird es durch "$$" maskiert. Das gleiche gilt für "#", "%",
  " ' " und " " ".

  Beispiel:

  ```
  /* Der Preis wird in $$ angegeben.
  ```

  Fehlende Zeichen werden durch Blanks aufgefüllt; zu viele Zeichen werden abgeschnitten.

  Damit ein Dollarzeichen "$" in einem (k)SHELL-Skript nicht als SHELL-Varaible aufgelöst wird, wird es durch "\$" maskiert.

  Beispiel:

echo Der Preis wird in \$$ angegeben.

Job:
Echo Der Preis wird in \$ angegeben.

Ausgabe:
Der Preis wird in $ angegeben.

3.3 Wertzuweisung an Variablen bei der Definition

Einer Variablen können bei der Definition entweder ein einzelner Wert oder eine Werteliste zugewiesen werden.

3.3.1 Einzelwertzuweisung (einfache Variablen)

Bei der Einzelwertzuweisung wird einer Variablen, die innerhalb eines Musters oder eines Schrittes einer Prozedur benutzt wird, ein Wert zugewiesen.

Format

#$Variable[1]=[2]Wert[3]

Die hochgestellten Ziffern entsprechen den Ziffern der Feldbeschreibung.

Feldbeschreibung

1	***Variable***	Bezeichner der Variablen Die Länge von Variablennamen kann zwischen 1 und 40 Stellen betragen. 1. Es können auch **Standardvariablen** von SQS-TEST/Professional benutzt werden (siehe Abschnitt 3.5 - "Standardvariablen"). Der entsprechende Wert wird ihnen implizit zugewiesen, kann aber durch eine explizite Wertzuweisung überschrieben werden (Ausnahme: $COUNT).
2	***Operator =***	Vor und hinter dem Operator "=" ist ein Blank zwingend.
3	***Wert***	Wert der Variablen Der Wert darf Blanks enthalten; führende Blanks werden ignoriert. Die Wertzuweisung kann selber wiederum Variablen enthalten. Die maximale Länge des Eingabewertes ist durch die Länge der Zeile begrenzt. Ist die Länge größer als der vorhandene Platz, kann die Zuweisung über Werteliste (Typ zeilenweise) benutzt werden. Alternativ kann der Wert auf mehrere Variablen aufgesplittet und anschließend durch Zusammensetzen dieser Variablen zu einer neuen Variablen wiedererhalten werden (siehe Beispiel).
	' ' oder **" "**	Durch einfache oder doppelte Hochkommata wird ein String der Länge Null zugewiesen. Dies ist erforderlich bei der Definition der Variablen, wenn kein Wert zugewiesen wird.

Beispiele

Parameter-Dokument

```
$VAR1 = Beispielzuweisung
$VAR   = TEXT$VAR1.TEXTVAR2$VAR3

$VAR4 = Dies ist ein langer
$VAR5 = Eingabewert
$VAR_NEU = $VAR4$VAR5

/* Zuweisung eines Strings der Länge Null
$VAR6 = ''
```

3.3.2 Zuweisung einer Werteliste (Listen- und Tabellenvariablen)

Einer Variablen kann eine Werteliste zugewiesen werden. Diese Zuweisung kann zeilenweise oder wortweise erfolgen.

Format: Werteliste zeilenweise (Listenvariable)

$Variable[1]=[2]*[3]
Wert1[4]
Wert2[4]
Wertn[4]

Die hochgestellten Ziffern entsprechen den Ziffern der Feldbeschreibung.

Bei der **zeilenweise Zuweisung** enthält eine Zeile der Werteliste **einen** Wert. Die verwendete Variable wird als **Listenvariable** bezeichnet. Die zeilenweise Zuweisung ist nur in Prozeduren und in allen Parameter-Dokumenten möglich (nicht in Mustern).

Format: Werteliste wortweise eine Zeile (Tabellenvariable)

#$Variable[1]=[2] Wert1[4] Wert2[4] Wert3[4]

Format: Werteliste wortweise mehrere Zeilen (Tabellenvariable)

$Variable[1]=[2]*[3]
Wert1[4] Wert2[4]
Wert3[4] Wert4[4]
Wert5[4] Wertn[4]

Bei der **wortweisen Zuweisung** besteht die Werteliste aus Zeilen, die mehrere Werte enthalten. Die gesamte Liste wird als **Tabelle** aufgefasst, d.h. alle Zeilen enthalten die gleiche Anzahl Werte.

Bei der **Tabellenvariablen** kann auf ein Feld einer Tabelle nur über einen **einwertigen Index** (Reihenfolge der Felder) zugegriffen werden. Um einen zweiwertigen Zugriff zu ermöglichen, kann folgendermaßen vorgegangen werden:

- über eine Schleife wird auf jeweils eine Zeile der Tabelle zugegriffen und
- über den einwertigen Index auf das n-te Element der jeweiligen Zeile, d.h. die Zählung der Felder beginnt bei der ausgewählten Zeile.

Feldbeschreibung

1	***$Variable***	Bezeichner der Variablen Die Länge von Variablennamen kann zwischen 1 und 40 Stellen sein
2	***Operator***	Die Leerstelle vor und hinter dem Operator "=" ist zwingend.

3	***Liste***	Symbol für Liste
	*	Dieses Symbol ist zwingend notwendig.
4	***Wert***	Wert der Variablen
		Der Wert darf Blanks enthalten; führende Blanks werden ignoriert.
	'' oder ""	Durch einfache oder doppelte Hochkommata wird ein String der Länge Null an ein Feld einer Werteliste zugewiesen. Dies ist erforderlich bei der Definition der Variablen, wenn kein weiterer Wert zugewiesen wird.

Die Wertzuweisung kann selber wiederum Variablen enthalten. Falls ein Wert in einer Listenvariable durch eine Variable ausgedrückt wird, müssen alle Werte mit einfachen oder doppelten Hochkommata (' ' oder " ") gekennzeichnet werden:

Beispiel

```
$VAR = *
'$DOKU'
'$PATH'
'Müller'
```

Auf die einzelnen Worte einer Liste kann über einen einwertigen **Index** zugegriffen werden.

"Index" ist eine Konstante (Zahl in Klammern hinter der Listenvariablen); sie liefert das entsprechende Wort aus der Werteliste. Ist die Werteliste kürzer als der Index, bricht die Jobgenerierung mit einem Fehler ab.

Beispiel

Zuweisung:

```
$LIS = *
TEST1
TEST2
TEST3
```

Alternative Zuweisung:

```
$LIS = *
TEST1 TEST2 TEST3
```

Verwendung:

`$LIS(2)` liefert den Inhalt des zweiten Elementes

`TEST2`

Eine Zuweisung an $LIS(3) ist nicht möglich, da eine Zuweisung auf Felder einer Liste nicht möglich ist.

Die Indexfunktion enthält eine **Zusatzfunktion**: wurden bei den einer Listenvariablen zugewiesenen Werten Hochkommata (' ' bzw. " ") verwendet, werden diese bei der Ersetzung der Variablen bei der Jobgenerierung gelöscht.

Beispiele

Definition im Parameter-Dokument

```
$VAR = 'Hugo'
```

Verwendung	Ergebnis
`$VAR`	`'Hugo'`
`$VAR(1)`	`Hugo`

Definition im Parameter-Dokument

```
$VAR = *
'Hugo' 'Berta'
```

Verwendung	Ergebnis
`$VAR`	`'Hugo' 'Berta'`
`$VAR(2)`	`Berta`

3.4 Verwendung von Variablen

Definierte Variablen können Ausdrücken verschiedener Art **zugewiesen** werden:

- Es ist möglich, eine Variable einer anderen Variable zuzuweisen.
- Eine Variable kann den Sprachelementen #ACTIVE, #TEMPLATE, #GENPARM, #FILEPARM sowie #TESTITEMPARM in einer Prozedur zugewiesen werden
- Innerhalb von Schleifen kann auch an das #STEP-Sprachelement eine Variable zugewiesen werden.
- In Mustern können Variablen an die Sprachelemente #CALL, #FILEPARM, #GENPARM, #TESTITEMPARM sowie #INCLUDE und #EXINCL (unter Beachtung der Einschränkungen, siehe die

Beschreibung dieser Elemente im Kapitel 5 - "Muster erstellen") zugewiesen werden.

- In einem Muster kann eine Variable auch einen Ausdruck (z.B. eine Befehlsfolge) **vertreten**, ohne dabei einem anderen Ausdruck zugewiesen werden zu müssen.
- Werden in Mustern oder Prozeduren Variablen verwendet, für die **keine Wertzuweisungen** existieren, werden sie bei der Generierung eines Jobs in einer Fehlerliste ausgewiesen. Solche Aufträge werden nicht gestartet, sondern enden mit einer Fehlermeldung.
- Variablen werden maximal über **zehn Stufen** ersetzt, danach wird mit einer Fehlermeldung abgebrochen.

3.5 Standardvariablen

$PROJ	Name des aktuellen SQS-TEST/Professional-Projektes
$PZ	Name der Prozedur
$PZ1	1. Stelle des Prozedurnamens
$PZ2 - $PZn	2. bis - n-te Stelle des Prozedurnamens
$USER	Benutzername oder -account des SQS-TEST/Professional-Anwenders
$PROCESS-GROUP	Name der Aufgabe (#PROCESSGROUP-Bereich)
$PROCESS	Name des Auftrags (#PROCESS-Bereich)
$STEP	Name des Schritts (#STEP-Bereich)
$COUNT	Zählvariable, Startwert = 0 Der Wert von $COUNT kann durch die Anweisung "#COUNT n" im Muster um n erhöht werden. Fehlt n bei dieser Anweisung, wird um 1 erhöht. $COUNT gilt musterübergreifend innerhalb eines Auftrags. Mit $COUNT kann der Inhalt dargestellt oder abgefragt werden, jedoch nur, wenn diese Variable in einer Prozedur oder in einem Parameter-Dokument nicht schon einmal deklariert wurde. Der Wert von $COUNT wird auf 0 zurückgesetzt, wenn im Muster oder in einem Schritt eine #JOBCARD-Anweisung ausgeführt wird.

Die folgenden Standardvariablen sind durch die entsprechenden Angaben in der Environment-Datei "SQSPVS.ENV" definiert.

$TSCLIENT	Angemeldeter Client, der den Job ausführt
$TSCLIENTIP	IP-Adresse des Clients
$TSENV	Aktuelle Umgebung (Environment)
$TSSYS	Programmverzeichnis von SQS-TEST/Professional
$TSGLOBAL	Globalverzeichnis des aktuellen Projektes

$TSTMP	Jobverzeichnis
$TSPRJ	Pfad des aktuellen Projektes
$TCPROJ	Name des aktuellen Projektes

3.6 Prioritäten beim Ersetzen von Variablen

Das Ersetzen der Variablen unterliegt folgenden Prioritäten. Dabei entspricht "1" der höchsten und "8" der niedrigsten Priorität.

1	STANDAD (spezielles Parameter-Dokument)
2	Spezifikation im Muster
3	Spezifikation im #STEP-Bereich
3.1	- im Parameter-Dokument
3.2	- in der Prozedur
4	#INPUT
4.1	- im #PROCESSGROUP-Bereich
4.2	- im #PROCESS-Bereich
4.3	- im #GLOBAL-Bereich
5	Spezifikation im #PROCESS-Bereich
5.1	- im Parameter-Dokument
5.2	- in der Prozedur
6	Spezifikation im #PROCESSGROUP-Bereich
6.1	- im Parameter-Dokument
6.2	- in der Prozedur
7	Spezifikation im #GLOBAL-Bereich
7.1	- im Parameter-Dokument
8	Zuweisung an Standard-Variablen

Das bedeutet:

Falls eine Variable in einem STANDAD-Dokument und an einer anderen Stelle einen Wert (z.B. in einem Muster) erhalten hat, erhält die Variable bei der Joberstellung immer den Wert aus STANDAD.

Die Priorisierung von mehreren Parameter-Dokumenten vom Typ STANDAD wird im Kapitel 6 - "Parameter-Dokumente bearbeiten" beschrieben.