

Features des SAS Display Managers: Tipps für eine effiziente Arbeitsumgebung

Renate Scheiner-Sparna
iOMEDICO AG
Hanferstraße 28
79108 Freiburg i.Br.
renate.scheiner-sparna@iomedico.com

Zusammenfassung

Die bekannte SAS Programmierumgebung mit Enhanced Editor, Log Fenster und Output Fenster begleitet den SAS Programmierer täglich. Sie zu optimieren heißt, Effizienz, Sicherheit und Konzentration auf das Wesentliche zu unterstützen.

Optionen aus dem Menü, einige sehr hilfreiche Belegungen von Funktionstasten, Keyboard Macros, Kommandos und Fenster-Einstellungen werden gezeigt. Es entsteht ein großes, aufeinander abgestimmtes Bild von Funktionsweisen und Arbeitswerkzeugen.

Schlüsselwörter: Display Manager, Keys, Toolbar, Enhanced Editor, SAS Log, Viewtable

1 Einleitung

Der Display Manager ist in SAS seit der Version 5 (1985) implementiert. Er besteht im Kern aus dem Editor (seit Version 8 Enhanced Editor), Output und Log. Der Enhanced Editor ist die Neuerfindung des ursprünglichen Program Editors.

Zu diesen Fenstern kommen inzwischen viele weitere hinzu, z.B. Viewtable, Libname, Results etc. Je nach Arbeitsweise bekommen diese mehr oder weniger Bedeutung.

Genauso wie der Display Manager schon viele Jahre existiert, sind auch die Dinge, die ich hier zusammengestellt habe, zumeist lange bekannt. Ich halte diese Sammlung für wichtig, weil diese Informationen hier sehr kondensiert dargestellt werden, weil sie sehr viel Aufwand sparen und weil sie insbesondere Anfängern oder Skeptikern Hilfen an die Hand geben, wie sie effizient mit dem Display Manager arbeiten können. Der Wert dieser Einstellungen offenbart sich nach einer Gewöhnungszeit erst, wenn man einmal nicht die eigenen Einstellungen zur Verfügung hat und sie schmerzlich vermisst.

Nach der Beschreibung einiger Fenster-unabhängiger Einstellungen wird es hier schwerpunktmäßig um die drei Fenster Enhanced Editor, Log und Viewtable gehen.

2 Fenster-unabhängige Einstellungen

Einige sehr hilfreiche Features lassen sich über das Menü einstellen und werden dann automatisch im SASUSER Profil gespeichert. Sie müssen nur einmalig entsprechend gesetzt werden und stehen dann in jeder folgenden SAS Sitzung zur Verfügung.

2.1 Command Bar

Der Command Bar ermöglicht den Einsatz von kurzen Befehlen. So können Nutzer die Tastatur dem Wechsel zur Maus vorziehen. Die Befehle sind abhängig von dem zuletzt vorher aktivierten Fenster, wie weiter unten beschrieben.

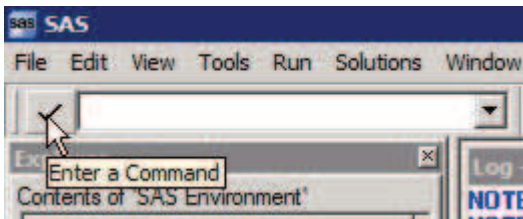


Abbildung 1: Command Bar

Wenn der Command Bar nicht angezeigt wird, kann seine Anzeige über das Menü Tools – Customize im Reiter Toolbars aktiviert werden.

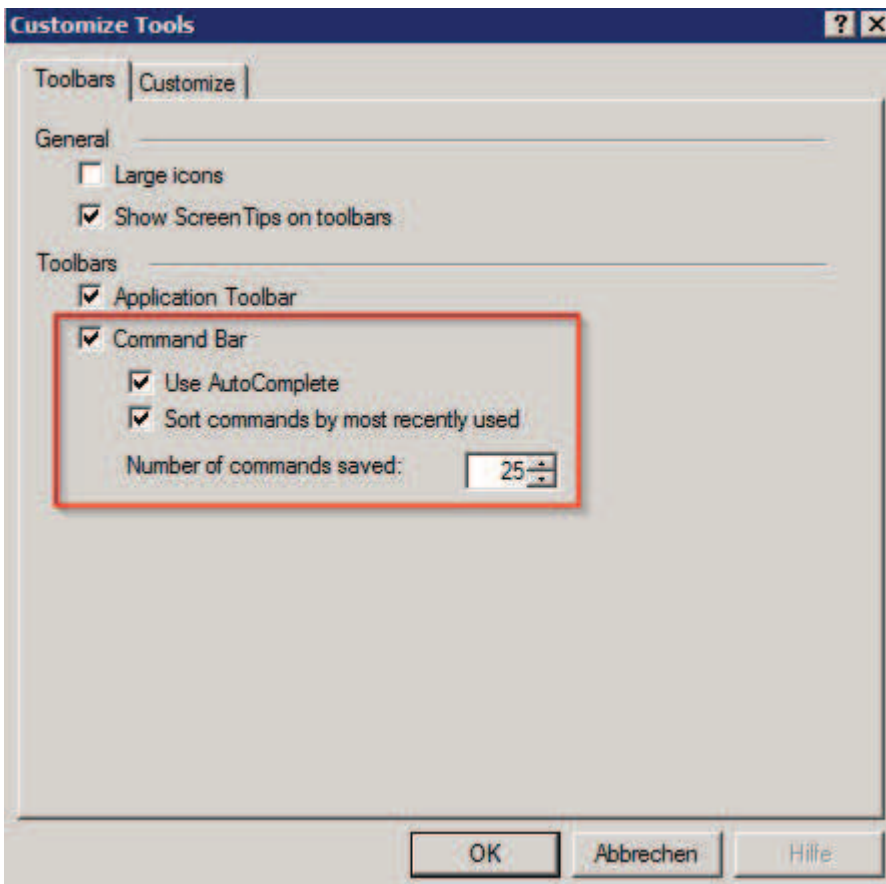


Abbildung 2: Aktivierung des Command Bars

2.2 Fenster-Anordnung

Die einzelnen Fenster des Display Managers können mit dem Cursor in Position und Größe angepasst werden wie in jeder anderen Windows-Anwendung auch. Für manche Fenster, z.B. den SAS Explorer, gibt es die Option Docked, mit der das Fenster an einer Seite angeheftet werden kann. Sie ist über das Menü Window – Docked zu setzen bzw.

aufzuheben. Mit dem Befehl „wsave all“ können die Einstellungen aller Fenster im SASUSER Profil gespeichert werden [3].

3 Enhanced Editor

Der Enhanced Editor birgt viele Möglichkeiten, bleibt mit seinen Standard-Einstellungen aber auch in manchem hinter der Funktionalität anderer Editoren zurück. Vieles lässt sich den Bedürfnissen anpassen, „Enhancing the Enhanced Editor“ [1].

Warum kann es überhaupt eine gute Möglichkeit sein, mit dem Enhanced Editor zu arbeiten? Dafür spricht die Transparenz und Unmittelbarkeit in der Programmierung, die letztlich auch der Handhabbarkeit und einfacheren Fehlersuche zugute kommt. Eine unüberschaubare Struktur von Umgebungsparametern sollte verschlankt werden. Der Programmierer soll wissen, was er tut, auch Anfänger sollen sich in der Programmstruktur zurechtfinden. Man kann nur sicher und effizient arbeiten, wenn man einen Überblick hat.

Dagegen spricht, dass viele Programmier-Umgebungen mit einer Vielzahl an Makrovariablen arbeiten, ohne die Programmläufe nicht mehr funktionieren. Eine Vielzahl von Umgebungsparametern wird bereits außerhalb des zu bearbeitenden Programms gesetzt. Programme werden zum Teil in anderen Editoren geschrieben.

Für eine begrenzte Anzahl von Makrovariablen kann man sich leicht helfen. Sie lassen sich mit einem normalerweise auskommentierten Block am Anfang eines Programms einfach initialisieren:

```

1 /*
2 %let FLAG=FASFL;
3 %let SG=treat1;
4 %let NUMBER=5;
5 */;
6
7 data base;
8     set ads.adsl;
9     where &sg.;
10 run;
```

Ein solcher Block kann problemlos von Zeile 2 an bis zum Ende des Programms fehlerfrei ablaufen. Dann sind die Makrovariablen initialisiert. Das Semikolon am Ende von Zeile 5 beendet die Zeile fehlerfrei, auch wenn kein /* wie in der ersten Zeile vorher mit ausgeführt wurde. Die Zeilen stören also einerseits nicht in einem kompletten Programmlauf, andererseits kann man ab Zeile 2 markieren und den diesem Block folgenden Code gleich mit ausführen, ohne dass ein Fehler auftritt.

Die Makrovariablen geben dem Programmierer die Möglichkeit, an genau der Stelle gezielt zu arbeiten, die er braucht, ohne immer gleich eine Vielzahl von Programmen ablaufen zu lassen.

3.1 Schriftgröße

Über das Menü Tools – Options – Enhanced Editor lassen sich im Reiter Appearance sowohl die Farben des Editors, als auch die Schriftgröße einstellen. Eine Änderung der Farb-Codierung des Programmcodes kann ich nicht empfehlen, das kann sehr irritierende Auswirkungen haben, die bei der Fehlersuche stören. Die Schriftgröße ist jedoch mit standardmäßig eingestellten 10 pt für viele anstrengend zu lesen.

3.2 Tabulator

Für die Ordnung im Programmcode und die Weitergabe an andere Programmierer ist es notwendig, Konventionen zur der Verwendung der Tabulator-Taste anzuwenden. Dies lässt sich einfach umsetzen, indem man Tabulator-Zeichen durch eine bestimmte Anzahl Leerzeichen ersetzt. Über das Menü Tools – Options – Enhanced Editor kann man im Reiter General sowohl die Anzahl, als auch die Ersetzung festlegen.

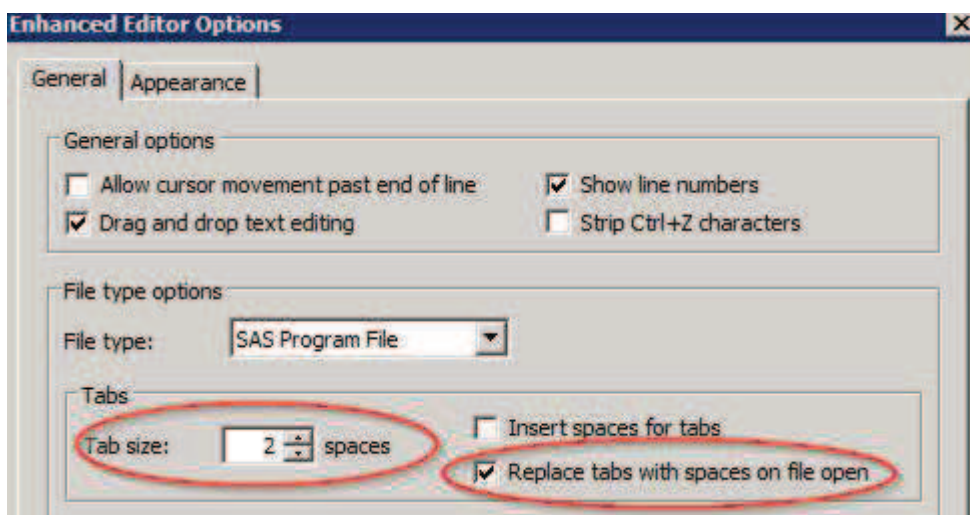


Abbildung 3: Tabulator-Ersetzung

3.4 Zeilennummern

Im selben Enhanced Editor Options Fenster, wie unter 3.3 beschrieben, können auch Zeilennummern für den Enhanced Editor eingestellt werden. Sie sind sehr nützlich zur Orientierung in langen Programmen und zum direkten Anspringen, wie in Abschnitt 3.5 erläutert wird.

Alternativ kann im Command Bar der Befehl „nums“ die Zeilennummern generieren. Wichtig ist, dass der Cursor unmittelbar vorher in einem Enhanced Editor Fenster stand.

3.5 Springen zu einer bekannten Zeile

Die Tastenkombination strg + g lässt ein Fenster erscheinen, das die Eingabe einer Zeilennummer fordert und mit drücken der Enter-Taste dorthin springt [9].



Abbildung 4: Go To Line Fenster

3.6 Textmarken setzen und anspringen (Bookmarking)

Mit der Tastenkombination `strg + F2` setzt man eine Textmarke in der Zeile, in der der Cursor steht. Es können mehrere Textmarken gesetzt werden. Mit `F2` springt man von einer zur nächsten vorwärts, mit `shift + F2` rückwärts [7].

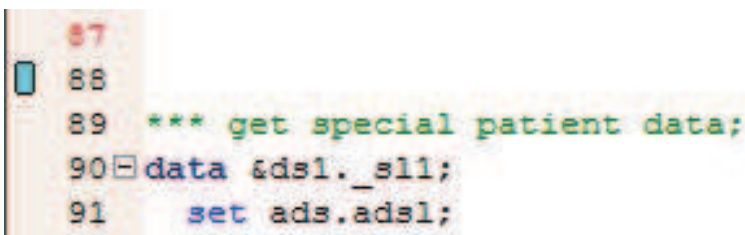


Abbildung 5: Textmarke in Zeile 88

3.7 Klammern kontrollieren

Steht der Cursor vor einer Klammer, kann mit der Tastenkombination `strg + 9` die paarige Klammer gesucht werden. Diese 9-Taste ist hierfür sehr einleuchtend, da sie auch für `)` und `]` verwendet wird. Die Suche nach der paarigen Klammer funktioniert sowohl für beide Klammerarten, als auch in jede Richtung.

3.8 Blockweise kommentieren

Markierte Zeilen kann man mit der Tastenkombination `strg + 7` auskommentieren. Es wird jede einzelne Zeile in `/* */` eingebettet. Da die 7-Taste auch für `/` verwendet wird, ist das sehr leicht zu merken. Blockweises Ent-Kommentieren funktioniert mit `strg + shift + 7` [8, 11].

Trotz standardmäßiger Vorbelegung dieser Tastenkombination muss dies erfahrungsgemäß evtl. nochmals eingestellt werden. Hierzu wählt man über das Menü `Tools – Options – Enhanced Editor Keys` [4, 5], dort in der Commands Liste „Comment the selection with lin...“, entfernt die zugewiesene Current keys-Kombination durch `Reset` (Abb. 6) und erzeugt sie neu.

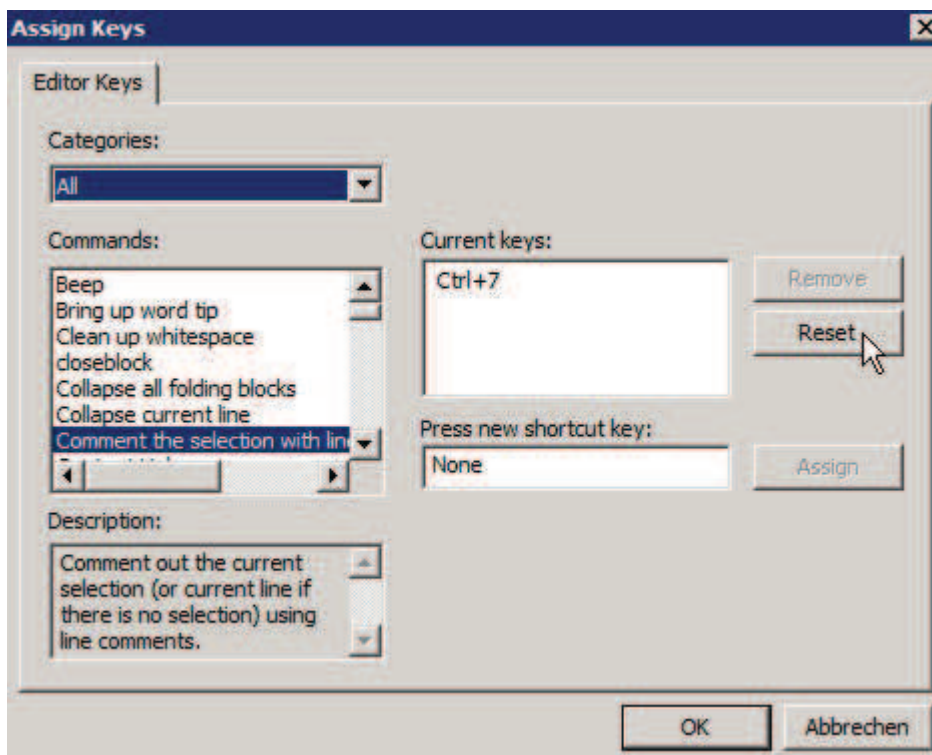


Abbildung 6: Tastenkombination für Blockkommentierung

3.9 Blockweise markieren

Hält man die Alt-Taste fest, kann im Enhanced Editor mit dem Cursor Block- bzw. Spaltenweise markiert werden. Das ist sehr praktisch, wenn der Code sauber untereinander geschrieben ist und wie im Beispiel von Abbildung 7 alle Zeilen gleichartig bearbeitet werden sollen. Hier soll ein optionaler Makroparameter aus einer Reihe von Makroaufrufen entfernt werden.

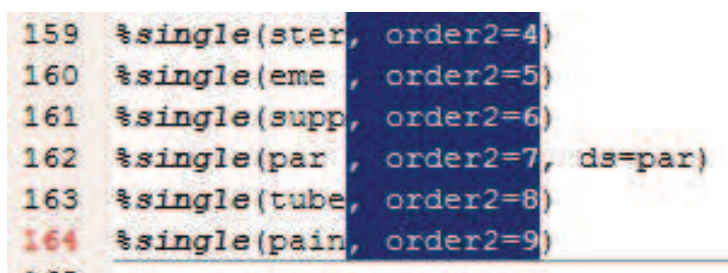


Abbildung 7: Blockweise markieren

Blockweises Entfernen von Code funktioniert auf diese Weise, blockweises Einfügen leider nicht.

3.10 Groß- /Kleinbuchstaben

Markierter Text kann per Tastenkombination in Groß- bzw. Kleinbuchstaben umgesetzt werden. Für Großbuchstaben verwendet man strg + shift + U (upper case), für Kleinbuchstaben strg + shift + L (lower case) [12].

3.11 Abbreviations / Keyboard Macros

Über das Menü Tools – Add Abbreviations kann einem Textstring ein Programm Code zugewiesen werden. Es empfiehlt sich, komplexere Texte im Enhanced Editor vorzuschreiben und mit copy/paste einzufügen und dann nur mit ok zu bestätigen (siehe Abb. 9). Wird danach im Enhanced Editor der Textstring getippt, erscheint ein fly-over Text mit dem hinterlegten Programm Code (Abb. 10). Nun hat man die Möglichkeit, mit Enter oder Tabulator den Textstring durch den Programm Code zu ersetzen. Möchte man das nicht, kann man den eingblendeten Text ignorieren und einfach weiter tippen.

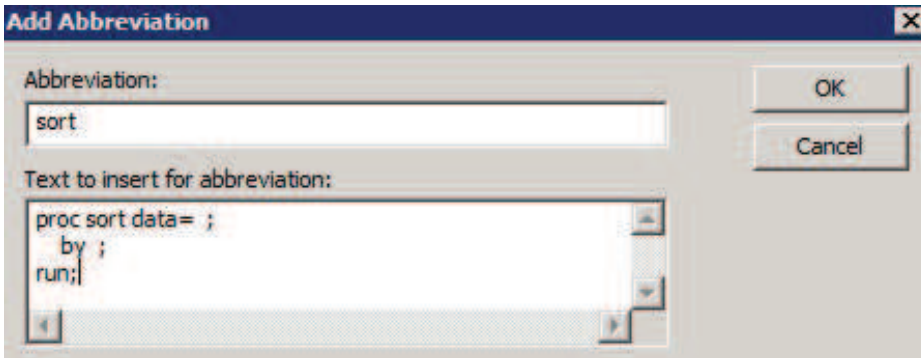


Abbildung 8: Add Abbreviation Fenster

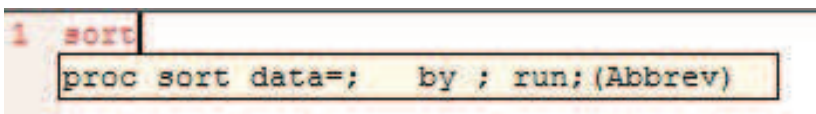


Abbildung 9: Fly-over Text im Enhanced Editor

Um einen Überblick über die vorhandenen Abbreviations zu bekommen oder diese zu bearbeiten kann man über das Menü Tools – Keyboard Macros – Macros ein Fenster bekommen, das alle vorhandenen Abkürzungen und ihre Eigenschaften aufführt. Sie können hier auch bearbeitet werden, z.B. indem man über Assign keys eine Tastenkombination zuweist, mit der der Text in den Enhanced Editor eingefügt werden soll. Auch Import und Export sind hier möglich [10], was sehr praktisch ist, um diese Funktionalität in einer Abteilung einheitlich zu handhaben.

Mit der beschriebenen Funktionalität werden Programmbausteine als Text in den Enhanced Editor eingefügt, aber nicht ausgeführt. Zum Ausführen von Programmbausteinen empfiehlt es sich, eine Funktionstaste mit dem Befehl zu belegen, wie in Abb. 10 gezeigt ist. Hierzu ist es wichtig darauf zu achten, dass die DM Keys geöffnet sind (Befehl „keys“ im Command Bar, nachdem der Cursor unmittelbar vorher in einem Enhanced Editor Fenster stand). Der Programmbaustein wird in einem Makro gekapselt (hier: kill1). Die Verwendung von doppelten Anführungsstrichen ist an dieser Stelle unerlässlich, damit das Makro ausgeführt wird.

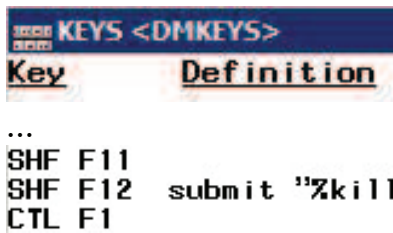


Abbildung 10: DM Key zur Programmausführung

3.13 Ausführen von Programmcode

Standardmäßig ist die Taste F8 oder der Mausklick auf das Submit-Symbol für das Ausführen von Programmcode vorgesehen. F8 entspricht dem Befehl „submit“, evtl. auch „submit; log“, was die Aktivierung des Log Fensters nach einem Programmablauf beinhaltet. Eine sehr geschätzte Alternative ist die Erweiterung auf „out; clear; log; clear; wpgm; submit; log;“ [3, 12]. Diese Kombination von Befehlen stellt sicher, dass sich keine veralteten Inhalte mehr im Output oder Log Fenster befinden. Es hat sich als sehr praktisch erwiesen, beide Möglichkeiten eines Submits zu haben, z.B. die einfache Variante auf F12 und die erweiterte auf strg + F12.

3.14 Automatisches Speichern

Der Zeitabstand, in dem die Inhalte des Enhanced Editors automatisch gespeichert werden, lässt sich über das Menü Tools – Options im Registerblatt Edit einstellen. Es ist Versions- und Systemabhängig, wo das gespeicherte File zu finden ist. Hierzu wurde eine SAS Note veröffentlicht (Usage Note 12392), die unter <http://support.sas.com/kb/12/392.html> zu finden ist.

4 Log Fenster

4.1 Zwei Konzepte zur Arbeit mit dem Log Fenster

Es gibt grundsätzlich zwei verschiedene Arten, mit SAS Logs zu arbeiten. Entweder wird das Log schon während des Programmablaufs als Datei gespeichert und dann automatisch per Makro nach Schlüsselworten durchsucht, oder der Programmierer bedient sich des Log Fensters direkt. Vorteile der zweiten Variante sind die Geschwindigkeit und Unmittelbarkeit des Zugriffs. Man muss nicht immer ein gesamtes System von Programmen ablaufen lassen. Die Farben im Log Fenster geben einen raschen Überblick. Detailarbeit ist einfacher, weil man exakt an einer bestimmten Stelle im Programm kleinschrittig arbeiten kann.

Im Alltag erweist es sich als sehr praktisch, je nach Situation beides alternativ zu verwenden:

- unmittelbar das Log Fenster während der Programmierung einzelner Abschnitte bis zur Fertigstellung

- komplette Programmdurchläufe mit automatischem Durchsuchen herausgeschriebener Log Files für fertige Blöcke

Für die unkomplizierte Bereitstellung einer Reihe von benötigten Makrovariablen an einer bestimmten Stelle im Programmablauf sei auf Abschnitt 3 verwiesen.

4.2 Find Keys

Im Enterprise Guide sind Buttons zum Suchen von ERROR und WARNING im Log implementiert, jedoch nur für diese beiden Begriffe [6]. Jedes der vielfältigen SAS Makros zum Durchsuchen gespeicherter Log Files arbeitet nach dem Prinzip, die Textzeilen wieder einzulesen und zu durchsuchen.

Für die unmittelbare Arbeit mit dem Log Fenster hat es sich sehr bewährt, Find-Befehle auf Tastenkombinationen zu legen, wie in Abb. 11 gezeigt ist.

```
CTL B    libname
CTL D    clear
CTL E    find 'uninitialized'
CTL G
CTL H    find 'MERGE statement has'
CTL I    find 'missing values'
CTL J    find 'No observations'
CTL K    find 'by zero'
CTL L    find 'has been replaced'
CTL M    mark
CTL Q    find ERROR
CTL R    find 'repeats of BY'
CTL T    find converted
CTL U    find 'too small'
CTL W    find WARNING
CTL Y
```

Abbildung 11: DM Keys mit Find Befehlen

Diese Tasten liegen auf der Tastatur günstig zusammen. Durch die feste Belegung einiger Tasten im Computersystem müssen manche Tasten ausgespart werden. Die Anwendung muss kurz eingeübt werden, geht aber so schnell in Routine über, dass dies kein Problem darstellt.

Nach einem Submit unter Verwendung von F12 oder Strg + F12 steht der Cursor automatisch im Log Fenster, wie im Abschnitt 3.13 dargestellt. Man springt nun zuerst mit Strg + Pos1 zum Anfang des Logs. Hierbei ist zu beachten, dass diese Tastenkombination nur auf den ersten Bildschirm des Logs springt, jedoch nicht an die allererste Position. Alles, was oberhalb des Cursors steht, kann nicht von der folgenden Suche gefunden werden. Der erste Bildschirm ist jedoch in der Regel gut zu überblicken.

Steht der Cursor im ersten Bildschirm und oberhalb sind keine verdächtigen Log-Meldungen, hält man die Strg-Taste gedrückt und geht nacheinander die Tasten Q W E R T U I H J K L. Die Reihenfolge wird durch einen Blick auf Abb. 11 plausibel.



Abbildung 12: Find Befehle auf der Tastatur

Während man diese Tastenbefehle ausführt, behält man die Message Line im Blick, die sich ganz links unten im SAS zu finden ist, siehe Abb. 13.

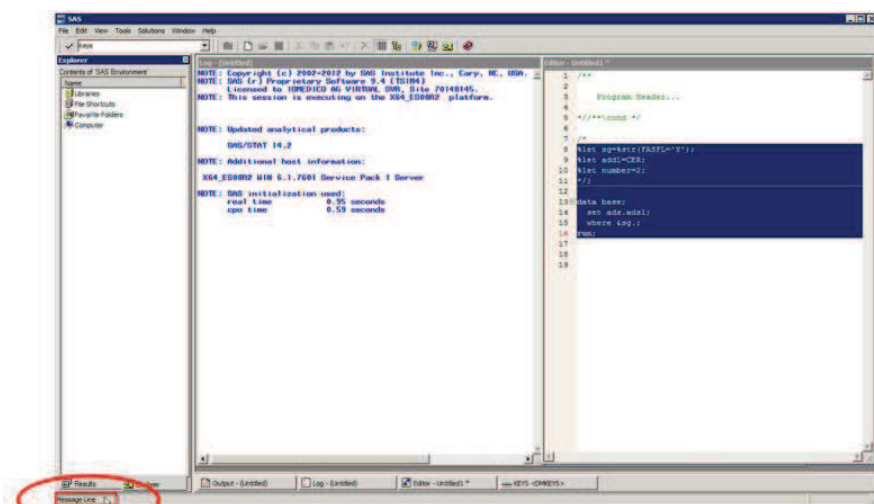


Abbildung 13: Message Line

Bei einem sauberen Log erscheinen hier nacheinander die Meldungen, dass der gesuchte Text nicht gefunden wurde. Tabelle 1 listet die Meldungen nach Tastenkombination.

Tabelle 1: DM Keys und ihre Meldungen in der Message Line bei sauberem Log

DM Key	Meldung Message Line	Sinn
Strg + Q	WARNING: No occurrences of „ERROR“ found.	ERROR
Strg + W	WARNING: No occurrences of „WARNING“ found.	WARNING
Strg + E	WARNING: No occurrences of „uninitialized“ found.	Variable ... is uninitialized
Strg + R	WARNING: No occurrences of „repeats of BY“ found.	...repeats of BY values
Strg + T	WARNING: No occurrences of „converted“ found.	...converted to character ...converted to numeric
Strg + U	WARNING: No occurrences of „too small“ found.	...too small to be printed
Strg + I	WARNING: No occurrences of „missing values“ found.	Missing values
Strg + H	WARNING: No occurrences of „MERGE statement has“ found.	...MERGE statement has more than one dataset with repeats of by values

Strg + J	WARNING: No occurrences of „No observations“ found.	Dataset has no observations
Strg + K	WARNING: No occurrences of „by zero“ found.	Division by zero detected
Strg + L	WARNING: No occurrences of „has been replaced“ found.	Where clause has been replaced

Wenn ein Text String im Log gefunden wird, springt der Cursor an diese Position und bei erneuter Anwendung der Tastenkombination von einem dieser Texte zum nächsten, falls es einen weiteren weiter unten im Log gibt. So kommt man schnell zu den zu prüfenden Stellen, auch wenn das Log als Ganzes unübersichtlich erscheint.

Hiermit steht dem Programmierer ein Mittel zur Verfügung, die Texte zuverlässig und unmittelbar zu finden. Die Gewichtung der Texte hängt von den Anforderungen an das Programm ab. Je nach Validierungsstufe, Auftraggeber oder eigenem Anspruch kann der Programmierer entscheiden, ob er einzelnen Meldungen nachgeht oder nicht. Das entspricht inhaltlich dem Vorgehen mit der Suche von Texten per SAS Makro.

5 Viewtable Fenster

Das Viewtable Fenster bietet die Möglichkeit, SAS Data Sets als Datentabelle komfortabel anzuschauen. Wer im Display Manager arbeitet und unmittelbar die Ergebnisse eines Programmierschrittes anschauen möchte, kann dies hier in wenigen Sekunden verwirklichen.

5.1 Öffnen und Schließen eines Viewtable Fensters

Das Viewtable Fenster kann man entweder durch Mausklick auf das Data Set im SAS Explorer Fenster öffnen, oder durch Eingabe des Namens im Command Bar. Wenn die DM Keys entsprechend gesetzt sind, geht das ganz einfach mit F11 (hinterlegter Befehl: command focus) [3] und Schreiben des Namens. Da der Command Bar sich die letzten verwendeten Befehle merken kann und selbst vervollständigt (siehe 2.1), muss man in der Regel nur sehr wenige Buchstaben tippen und kann gleich mit Enter das Viewtable erhalten. Der Command Bar löst dabei auch Makrovariablen auf, die bei einer gewissen Standardisierung oft in Data Set Namen vorkommen.

Eine extrem vereinfachte Optimierung dieses Vorgangs ist es, die System-Makrovariable SYSLAST zu verwenden. Sie wird als Befehl vt „&syslast.“ auf eine Funktionstaste in den DM Keys gelegt, wie in Abb. 14 gezeigt. So kann mit einer einzigen Taste immer das zuletzt erzeugt Data Set geöffnet werden. Genauso schnell kann es wieder geschlossen werden, wenn man den Befehl winclose in den Viewtable (VT) Keys auf die benachbarte Funktionstaste legt (Abb. 14). Es ist hier zu beachten, dass die Viewtable Keys ein ganz eigenes Set von Keys bilden, die nur vom Viewtable aus ausgeführt werden können. Zum Editieren geöffnet werden diese VT Keys mit dem Befehl Keys im Command Bar, nachdem zuletzt ein Viewtable das aktive Fenster war. Ich persönlich habe auch in den VT Keys den SYSLAST Befehl auf F3 gelegt für den Fall,

dass ich mich in einem anderen Viewtable befinde und von dort aus das zuletzt erzeugt Data Set zum Vergleich öffnen möchte.

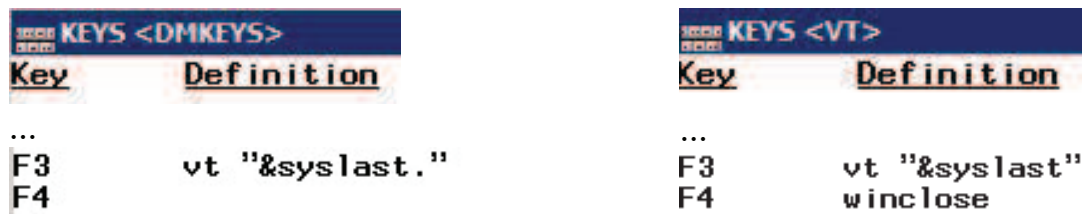


Abbildung 14: SYSLAST Key und WINCLOSE

5.2 Nützliche Befehle im Viewtable Fenster

Abschließend zum Viewtable Fenster sind in Tabelle 2 einige sehr nützliche Befehle zusammengefasst, die die Selektion von Zeilen und Spalten sowie deren Anordnung und Darstellung ermöglichen. Alle diese Befehle können auch in kleinen Buchstaben getippt werden, prägen sich aber in Großbuchstaben besser optisch ein.

Tabelle 2: Befehle zur Zeilen- / Spaltenselektion und Darstellung im Viewtable Fenster

COLL	Column Labels
COLN	Column Names
WHERE <Bedingung wie im Data Step>	Zeilenselektion
WHERE UNDO	Zeilenselektion rückgängig machen
SHOW '<Variablennamen>	Spaltenselektion / -reihenfolge
SHOW _all_	Spaltenselektion / -reihenfolge rückgängig
HIDE	Einzelne Spalten ausblenden
UNHIDE (optional: _all_)	Einzelne oder alle Spalten einblenden

Die Auto-Vervollständigung der Befehle im Command Bar nach erstmaliger Verwendung beschleunigt die Selektion sehr. Mit den folgenden wenigen Schritten kann man nun nach einem abgelaufenen Programmabschnitt das Ergebnis unmittelbar anschauen:

- F3 (=vt „&SYSLAST.“)
- F11 (=command focus)
- Selektionsbefehle eingeben und mit Enter bestätigen
- F4 (=WINCLOSE)

6 Einheitliche Keys unter Kollegen

Wer einzelne oder einige der in diesem Beitrag beschriebenen Keys schätzen gelernt hat, möchte sie unter Kollegen in einer Abteilung teilen oder sie gar zu Standard-Einstellungen machen. Da sie im SASUSER Profil abgelegt werden, brauchen sie nur einmalig gesetzt zu werden. Dazu kann man PROC CATALOG verwenden [2]:

```
libname standard '<pfad>';
PROC CATALOG CATALOG=SASUSER.PROFILE;
    COPY OUT=standard.standardPROFILE;
    SELECT DMKEYS.KEYS VT.KEYS;
```

```

QUIT;
libname standard;

*syntax to get the standard key definitions into the personal profile;

libname standard '<pfad>' access=readonly;
PROC CATALOG CATALOG= standard.standardPROFILE;
    COPY OUT=SASUSER.PROFILE;
    SELECT DMKEYS.KEYS VT.KEYS;
QUIT;
libname standard;

```

Alternativ können die Key Definitionen auch in der AUTOEXEC.SAS eingefügt werden.

Literatur

- [1] A.L. Carpenter: Doing More with the SAS Display Manager: From Editor to ViewTable – Options and Tools You Should Know. SAS Global Forum 2012. <http://support.sas.com/resources/papers/proceedings12/151-2012.pdf>
- [2] L. Weiler: END;END;END;END and Other Useful Shortcuts Using Function Keys. Sugi 1993. <http://www.sascommunity.org/sugi/SUGI93/Sugi-93-02%20Weiler.pdf>
- [3] <https://stackoverflow.com/questions/1233056/sas-display-manager-commands>
- [4] <http://morgan.dartmouth.edu/Docs/sas92/support.sas.com/documentation/cdl/en/hostvms/62450/HTML/default/alp-custenv-keydef.htm#a002489693>
- [5] https://www.reddit.com/r/sas/comments/35werh/the_dm_statement_a_relic_of_the_past/
- [6] <http://bi-notes.com/2012/04/sas-enterprise-guide-log-errors/>
- [7] http://support.sas.com/documentation/cdl/en/hostwin/63285/HTML/default/viewer.htm#editor_enhanced.htm
- [8] <https://communities.sas.com/t5/Base-SAS-Programming/Enhanced-Editor-shortcut-keys/td-p/43552>
- [9] <http://www.lexjansen.com/nesug/nesug10/ff/ff08.pdf>
- [10] <http://support.sas.com/resources/papers/proceedings15/3502-2015.pdf>
- [11] <http://www.technoleros.com/quickly-comment-and-uncomment-sas-code/>
- [12] <http://blogs.sas.com/content/sastraining/2012/02/13/3-sas-display-manager-shortcuts/>