

ZIP Dateien mit SAS erstellen und einlesen

Frank Biedermann
 Grüenthal GmbH
 Zieglerstraße 6
 52078 Aachen
 Frank.Biedermann@grunenthal.com

Zusammenfassung

SAS bietet die Möglichkeit, ZIP Dateien direkt einzulesen und zu erstellen. Seit der SAS Version 9.2 können ZIP Dateien mit dem ODS Statement ODS PACKAGE erstellt werden. Mit SAS 9.4 kommt die Option hinzu, mit dem „FILENAME ZIP“ Statement ZIP Dateien zu erstellen und einzulesen. In diesem Tipp & Trick Beitrag wird gezeigt, wie ZIP Dateien mit den beiden aufgeführten Optionen (ODS Statement ODS PACKAGE und Statement FILENAME ZIP) genutzt werden können. In Bezug auf das „FILENAME ZIP“ Statement wird auch dargestellt, wie man sich eine Inhaltsübersicht anzeigen lassen kann. Und zu guter Letzt wird veranschaulicht, wie man mit Hilfe der Applikation 7-Zip, von SAS aus, passwortgeschützte ZIP Dateien verarbeitet werden.

Schlüsselwörter: ODS PACKAGE, filename ZIP, 7-ZIP, WinZIP, WinRAR

1 ODS PACKAGE

1.1 ODS PACKAGE ZIP – FILENAME ZIP

In der folgenden Tabelle werden die wichtigsten Eigenschaften des ODS Ausgabekanals PACKAGE ZIP und des Statements FILENAME ZIP gegenübergestellt.

Tabelle 1: ODS PACKAGE – FILENAME ZIP

Eigenschaften	ODS PACKAGE ZIP (1)	FILENAME ZIP (2)
Erstellen	Ja	Ja
Hinzufügen	Ja	Ja
Unterordner	Ja	Ja
Alle Dateitypen	Ja	Ja
Öffnen (ZIP/GZIP/GZ ...)	Nein	Ja
Inhalte anzeigen	(Ja)	Ja
Passwortgeschützte Dateien	Nein	Nein

(1) Verfügbar seit SAS Version 9.2

(2) Verfügbar seit SAS Version 9.4

1.2 ODS PACKAGE ZIP – Aufbau

In der folgenden Tabelle wird der Aufbau des ODS PACKAGE ZIP erläutert.

Tabelle 2: ODS PACKAE ZIP – Aufbau

<pre>ods package(newzip) open nopf;</pre>	<p>Mit diesem Statement wird der ODS PACKAGE Ausgabekanal geöffnet. Die NOPF Option unterdrückt, dass zusätzliche Metadaten erstellt werden. Standardmäßig ist die Option an, die Option fügt einen Paket Metadaten Eintrag hinzu, dieser Eintrag kann von anderen SAS Applikationen gelesen und für die Weiterverarbeitung genutzt werden.</p>
<pre>ods package(newzip) add file=datei1 path="datei/";</pre>	<p>Die <datei1> wird dem Paket hinzugefügt, mit der Option PATH kann eine Unterordnerstruktur erstellt werden. In meinem Beispiel wird der Ordner <datei> erstellt.</p>
<pre>ods package(newzip) add file=datei2;</pre>	<p>Eine zweite Datei <datei2> wird dem Paket hinzugefügt.</p>
<pre>ods package(newzip) publish archive properties(archive_name="ausgabe.zip" archive_path="C:\temp");</pre>	<p>In dem PUBLISH ARCHIVE Statement werden die Ausgabeparameter festgelegt, archive_name = Name der ZIP Datei + Dateiendung, archive_path = Ausgabepfad, wo die ZIP Datei gespeichert werden soll.</p>
<pre>ods package(newzip) close;</pre>	<p>Mit diesem Statement wird der ODS PACKAGE Ausgabekanal geschlossen und die ZIP Datei wird erstellt.</p>

1.3 ODS PACKAGE ZIP – Beispiel Programm ZIP Datei erstellen

Der folgenden SAS Programmcode erstellt zwei Dateien: shoes.csv und cars.rtf.

```
%let ausgabedir = C:\temp;
ods listing close;
filename datei1 "&ausgabedir.\shoes.csv";
ods csv file=datei1;
proc print data=sashelp.shoes(obs=10) noobs;
  format _all_;
run;
filename datei2 "&ausgabedir.\cars.rtf";
ods rtf file=datei2;
proc print data=sashelp.cars(obs=10) noobs;
```

```
format _all_;
run;
ods _all_ close;
ods listing;
filename _all_ clear;
```

Im nächsten Schritt erstelle ich eine ZIP Datei mit dem Namen `ausgabe.zip`. Die beiden Dateien `shoes.csv` und `cars.rtf` aus dem vorherigen Programmcode Beispiel werden in die ZIP Datei geschrieben. Die Datei `shoes.csv` wird in den Unterordner `datei` gespeichert und die Datei `cars.rtf` auf den ersten Level der ZIP Datei.

```
%let ausgabedir = C:\temp;
ods package(newzip) open nopf;
ods package(newzip) add file=datei1 path="datei/";
ods package(newzip) add file=datei2;
ods package(newzip) publish archive
  properties (
    archive_name="ausgabe.zip"
    archive_path="&ausgabeDir."
  );
ods package(newzip) close;
```

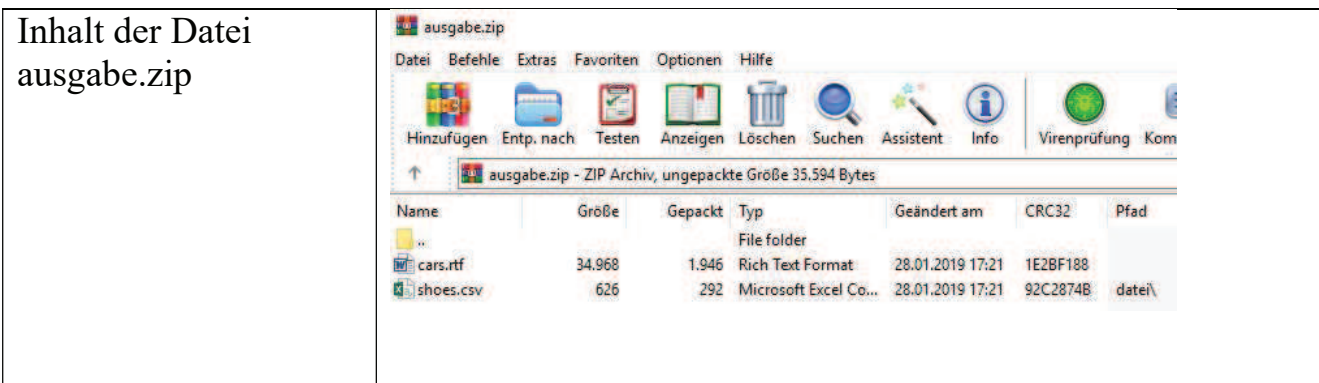


Abbildung 1: Datei `ausgabe.zip` Inhalt

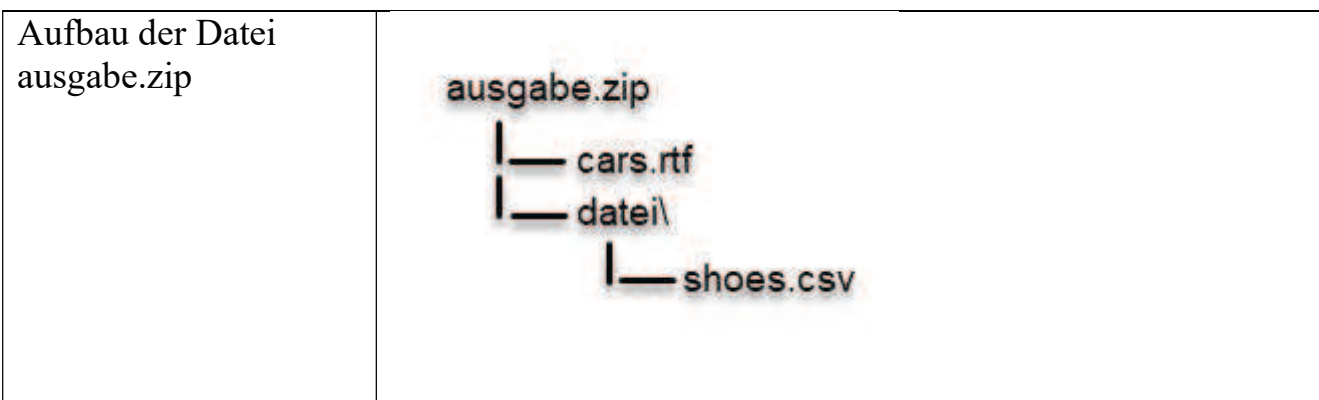


Abbildung 2: Datei `ausgabe.zip` Aufbau

2 Filename ZIP

2.1 Filename ZIP – ZIP Datei erstellen

In dem folgenden Beispiel wird eine ZIP Datei mit dem Filename ZIP Statement erstellt.

Tabelle 3: Filename ZIP – ZIP Datei erstellen

<pre>filename inData "C:\Program Files\SASHome\SASFoundation\9.4 \core\sashelp\class.sas7bdat";</pre>	<p>Als erstes wird ein Filename Statement zur Datei, welche in die ZIP Datei gespeichert werden soll, erstellt.</p>
<pre>filename outZip zip "c:\temp\sas_dateien.zip"</pre>	<p>Danach folgt ein Filename Statement zur ZIP Datei, die erstellt werden soll.</p>
<pre>member='class.sas7bdat';</pre>	<p>Der Dateiname der Datei, auf die im Filename Statement <code>inData</code> referenziert wurde, ist Teil des Filename ZIP Statements.</p>
<pre>data _null_; infile inData recfm=n; file outZip recfm=n; input byte \$char1. @; put byte \$char1. @; run;</pre>	<p>Die ZIP Datei wird mit Hilfe eines <code>data _null_</code> Schrittes erstellt.</p>
<pre>filename _all_ clear;</pre>	<p>Die Verknüpfung zwischen SAS und den Dateien <code>class.sas7bdat</code> und <code>sas_dateien.zip</code> wird gelöst.</p>

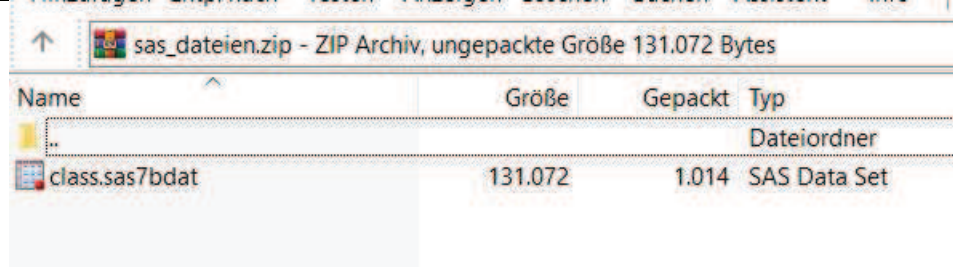
<p>Inhalt der Datei <code>sas_dateien.zip</code></p>	
--	--

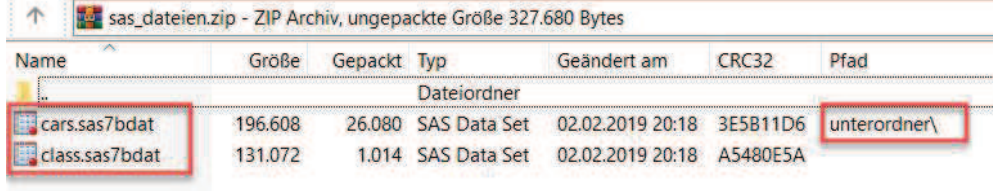
Abbildung 3: Datei `sas_dateien.zip` Inhalt

2.2 Filename ZIP – Dateien in eine existierende ZIP schreiben

In diesem Beispiel wird eine existierende ZIP Datei erweitert.

Tabelle 4: Filename ZIP – ZIP Datei erweitern

<pre>filename inData "C:\Program Files\SASHome\SASFoundation\9.4 \core\sashelp\cars.sas7bdat";</pre>	<p>Wie im vorherigen Beispiel, wird als erstes ein Filename Statement zur Datei, die der ZIP Datei hinzugefügt werden soll, erstellt.</p>
<pre>filename outZip zip "c:\temp\sas_dateien.zip"</pre>	<p>Danach folgt ein Filename Statement zur vorhandenen ZIP Datei.</p>
<pre>member='unterordner/ cars.sas7bdat';</pre>	<p>Der Dateiname der Datei auf die im Filename Statement <code>inData</code> referenziert wurde ist Teil des Filename ZIP Statements. Zusätzlich wird in diesem Beispiel ein Unterordner erstellt, in dem die SAS Datei <code>cars.sas7bdat</code> gespeichert wird.</p>
<pre>data _null_; infile inData recfm=n; file outZip recfm=n; input byte \$char1. @; put byte \$char1. @; run;</pre>	<p>Die SAS Datei <code>cars.sas7bdat</code> wird in die vorhandene ZIP Datei <code>sas_dateien.zip</code> geschrieben.</p>
<pre>filename _all_ clear;</pre>	<p>Die Verknüpfung zwischen SAS und den Dateien <code>cars.sas7bdat</code> und <code>sas_dateien.zip</code> wird gelöst.</p>

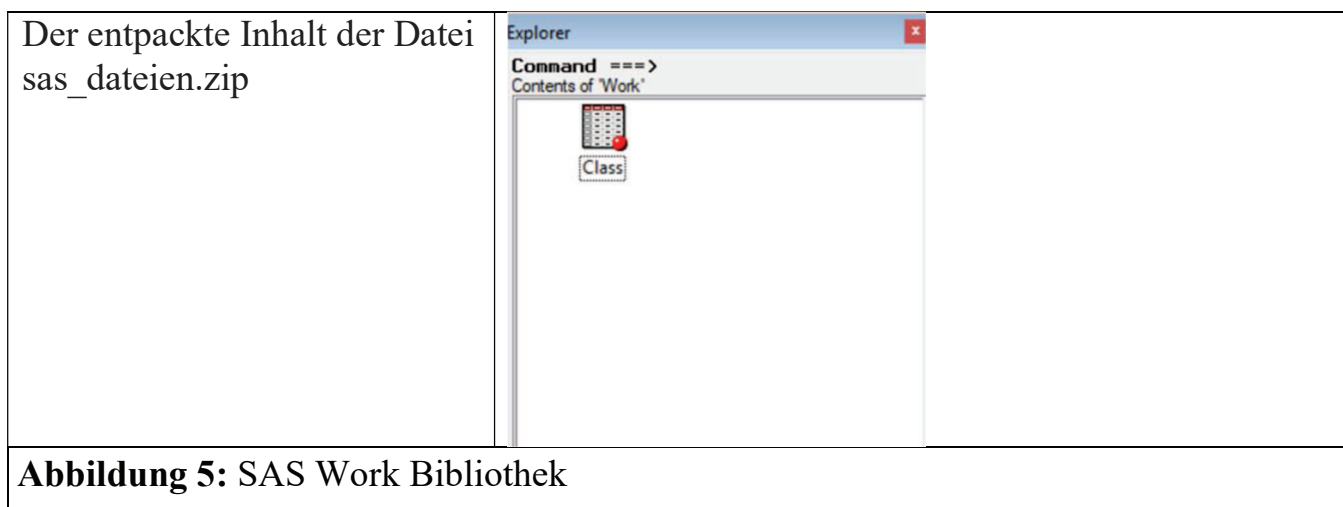
<p>Inhalt der Datei <code>sas_dateien.zip</code></p>	 <table border="1"> <thead> <tr> <th>Name</th> <th>Größe</th> <th>Gepackt</th> <th>Typ</th> <th>Geändert am</th> <th>CRC32</th> <th>Pfad</th> </tr> </thead> <tbody> <tr> <td colspan="7" style="text-align: center;">Dateiordner</td> </tr> <tr> <td>cars.sas7bdat</td> <td>196.608</td> <td>26.080</td> <td>SAS Data Set</td> <td>02.02.2019 20:18</td> <td>3E5B11D6</td> <td>unterordner\</td> </tr> <tr> <td>class.sas7bdat</td> <td>131.072</td> <td>1.014</td> <td>SAS Data Set</td> <td>02.02.2019 20:18</td> <td>A5480E5A</td> <td></td> </tr> </tbody> </table>	Name	Größe	Gepackt	Typ	Geändert am	CRC32	Pfad	Dateiordner							cars.sas7bdat	196.608	26.080	SAS Data Set	02.02.2019 20:18	3E5B11D6	unterordner\	class.sas7bdat	131.072	1.014	SAS Data Set	02.02.2019 20:18	A5480E5A	
Name	Größe	Gepackt	Typ	Geändert am	CRC32	Pfad																							
Dateiordner																													
cars.sas7bdat	196.608	26.080	SAS Data Set	02.02.2019 20:18	3E5B11D6	unterordner\																							
class.sas7bdat	131.072	1.014	SAS Data Set	02.02.2019 20:18	A5480E5A																								
<p>Abbildung 4: Datei <code>sas_dateien.zip</code> Inhalt</p>																													

2.3 Filename ZIP – ZIP Dateien entpacken

In dem nächsten Beispiel wird gezeigt wie man mit dem Filename ZIP Statement eine ZIP Datei entpackt.

Tabelle 5: Filename ZIP – ZIP Datei entpacken

<pre>filename inZip zip "C:\temp\sas_dateien.zip";</pre>	<p>Im ersten Schritt wird das Filename Statement zur ZIP Datei, die entpackt werden, erstellt.</p>
<pre>filename outDs "%SYSFUNC(pathname(work))/class .sas7bdat";</pre>	<p>Danach folgt ein Filename Statement, wohin die Datei entpackt werden soll. In diesem Beispiel in die SAS Work Bibliothek.</p>
<pre>data _null_; infile inZip(class.sas7bdat) lrecl=256 recfm=F length=length eof=eof unbuf; file outDs lrecl=256 recfm=N; input; put _infile_ \$varying256. length; return; eof: stop; run;</pre>	<p>Die ZIP Datei sas_dateien.zip wird entpackt, der Inhalt class.sas7bdat wird in der SAS Work Bibliothek gespeichert.</p>
<pre>filename _all_ clear;</pre>	<p>Die Verknüpfung zwischen SAS und den Dateien class.sas7bdat und sas_dateien.zip wird gelöst.</p>



2.4 Filename ZIP – Inhalt einer ZIP Datei anzeigen

Um das Entpacken einer ZIP Datei zu automatisieren, sollte man den Inhalt kennen. Im nächsten Beispiel wird gezeigt, wie man eine Inhaltsangabe von einer ZIP erstellt und diese in einer SAS Datei speichert.

Tabelle 6: Filename ZIP – Inhalte anzeigen

<pre>filename inZip zip "C:\temp\sas_dateien2.zip";</pre>	<p>Wie auch bei den anderen Beispielen, muss zuerst das Filename Statement zur ZIP Datei erstellt werden.</p>
<pre>data work.contents(keep=memname isFolder); length memname \$200 isFolder 8; fid=dopen("inzip"); if fid=0 then stop; memcount=dnum(fid); do i=1 to memcount; memname=dread(fid,i); isFolder = (first(reverse(trim (memname)))=' '); output; end; rc=dclose(fid); run;</pre>	<p>Die ZIP Datei wird mit dem Befehl dopen geöffnet, mit dem Befehl dread wird der Inhalt gelesen und der Befehl dclose schließt die ZIP Datei wieder. Zusätzlich wird überprüft, ob kein führender Slash vorhanden ist. Die Inhaltsübersicht wird dann in die SAS Datei work.contents geschrieben.</p>
<pre>filename _all_ clear;</pre>	<p>Die Verknüpfung zwischen SAS und der Datei sas_dateien2.zip wird gelöst.</p>

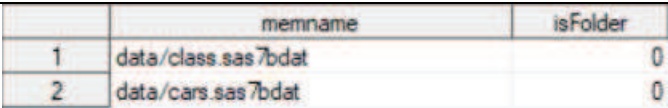
<p>Inhalt der SAS Datei contents</p>	 <table border="1"> <thead> <tr> <th></th> <th>memname</th> <th>isFolder</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>data/class.sas7bdat</td> <td>0</td> </tr> <tr> <td>2</td> <td>data/cars.sas7bdat</td> <td>0</td> </tr> </tbody> </table>		memname	isFolder	1	data/class.sas7bdat	0	2	data/cars.sas7bdat	0
	memname	isFolder								
1	data/class.sas7bdat	0								
2	data/cars.sas7bdat	0								

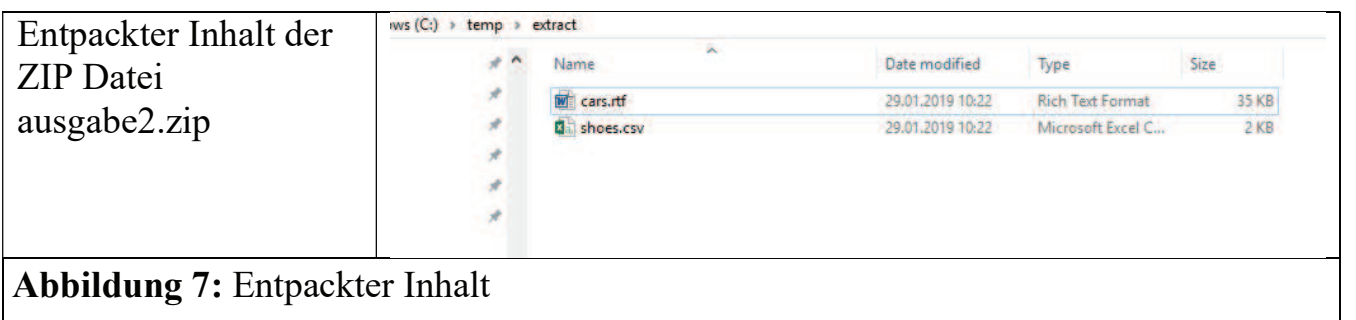
Abbildung 6: Datei contents Inhalt

3 Passwortgeschützte ZIP Datei entpacken

Passwortgeschützte ZIP Dateien können nicht mit SAS Boardmitteln entpackt werden. Dafür werden externe Tools wie z.B. WinZIP, WinRAR oder 7-ZIP benötigt. Im nächsten Beispiel wird dargestellt, wie man passwortgeschützte ZIP Dateien mit 7-ZIP entpackt.

Tabelle 7: Passwortgeschützte ZIP Dateien entpacken

<pre>options noxwait noxsync;</pre>	<p>Da ich 7-ZIP über die Windows Command Line aufrufe, sollten folgende SAS System Optionen zu Anfang gesetzt werden: <code>noxwait</code>, <code>noxsync</code></p>
<pre>%let source_path=C:\temp; %let source_filename=ausgabe2.zip; %let passwd=Ksfe1234!;</pre>	<p>Zur Einfachheit, erstelle ich drei SAS Makro Variablen: <code>source_path</code> = Pfad, wo die ZIP Datei gespeichert wurde, <code>source_filename</code> = Name der ZIP Datei und <code>passwd</code> = Das Passwort, um die ZIP Datei zu entpacken.</p>
<pre>data _null_; newdir=dcreate("extract", "&source_path."); run;</pre>	<p>Mit dem Befehl <code>dcreate</code> wird ein neuer Ordner erstellt, in diesen wird die ZIP Datei entpackt.</p>
<pre>%SYSEXEC "C:\temp\executable\7_zipportable\ App\ 7-Zip64\7z.exe" x "&source_path. &source_filename." -o "&source_path.\extract" -p"&passwd." ;</pre>	<p>Im nächsten Schritt rufe ich mit dem SAS Makro Befehl <code>%SYSEXEC</code> das 7-ZIP Command Line Tool auf. Das Command Line Tool kann im folgenden 7-ZIP Ordner gefunden werden: <code>App\7-Zip64\7z.exe</code></p> <p>Übersicht der verwendeten Kommandos: X = Entpacken, der volle Pfad wird benötigt. -O = Ausgabeverzeichnis -P = Passworteingabe</p>



Literatur

- [1] Chris Hemedinger 2015, Using FILENAME ZIP to unzip and read data files in SAS <https://blogs.sas.com/content/sasdummy/2015/05/11/using-filename-zip-to-unzip-and-read-data-files-in-sas/>
- [2] Jack Hamilton 2013, Creating ZIP Files with ODS. <http://support.sas.com/resources/papers/proceedings13/131-2013.pdf>
- [3] Eric Gebhart 2009, ODS Packages: Putting Some Zip in Your ODS Output. <https://www.lexjansen.com/nesug/nesug09/ap/AP15.pdf>
- [4] 7-ZIP Command Line Switches <https://sevenzip.osdn.jp/chm/cmdline/switches/index.htm>
- [5] 7-ZIP Command Line Commands <https://sevenzip.osdn.jp/chm/cmdline/commands/index.htm>