

zZip zZipSecure

Handbuch

Version 3

zZip/zZipSecure Handbuch

Copyright

Copyright © XPS Software GmbH

Alle Rechte vorbehalten.

Warenzeichen

Java ist ein Warenzeichen von Sun Microsystems, Inc.

Windows ist ein Warenzeichen der Microsoft Corporation.

MVS, OS/390, z/OS, VSE, VSE/ESA, z/VSE, VM/CMS, OS/400, TSO und CICS sind Warenzeichen der IBM Corporation.

PKWARE und PKZIP sind Warenzeichen der PKWare Incorporation.

Andere in diesem Handbuch erwähnten Marken- und Produktnamen sind Warenzeichen der jeweiligen Rechtsinhaber und werden hiermit anerkannt.

Inhaltsverzeichnis

Inhaltsverzeichnis	3
Abbildungen	5
Einführung	7
Allgemein.....	7
PKZIP Kompatibilität.....	8
Installation	9
Vorbemerkungen.....	9
zZip – z/OS.....	9
Einspielen des zZip-Installationsbandes.....	9
Einspielen des zZip-Installations-CD-Rom	10
Definition der XPSHFS/WFD-Datei	11
zZip – z/VSE	12
Einspielen des zZip-Installationsbandes.....	12
Definition der XPSHFS/WFD-Datei	12
Codepages	13
Konfiguration	16
Datenbibliothek 'XPSDATA'	16
z/OS.....	16
Job-Control	16
z/OS Parmlibs	17
z/VSE	18
Job-Control	18
Initialisierungsdatei 'XPSHDINI'	18
Systemeinstellungen	19
Variablen.....	35
Systemvariablen.....	36
zZip	38
Kontrollkarten.....	38
Wildcards und Filter	39
Aktionen	42
GZIP/GUNZIP – Komprimieren bzw. Entkomprimieren einzelner Dateien.....	42
ZIP/UNZIP – Komprimieren/Entkomprimieren von Archiven.....	57
Managementfunktionen für ZIP-Archive.....	78
ZIP,ADD - Dateien zu einem ZIP-Archiv hinzufügen	78
ZIP,COPY - Dateien von einem ZIP-Archiv in ein anderes kopieren	83
ZIP,DEL - Dateien aus einem ZIP-Archiv löschen.....	85
ZIP,UPD - Dateien in einem ZIP-Archiv ändern.....	86
ZIP,VIEW – Drucken des Inhaltsverzeichnisses eines ZIP-Archivs	91

ENCRYPT/DECRYPT – Verschlüsseln und Entschlüsseln von Daten	92
Programmausführungen.....	99
CMD – Ausführen eines Programmes unter der Kontrolle von HostDrive/J.....	99
WFD – Verarbeitung von Dateien mit dem Status 'waiting for delivery'	102
Eingabedaten.....	103
ARCHIVE – Lesen der Daten aus einem PKZIP-kompatiblen Archiv	104
IPT – Bereitstellung der Daten im Job-Control	106
LIB – Lesen der Daten aus einer Bibliothek	109
NET – Lesen der Daten von einem Netzwerklaufwerk.....	112
SEQ – Lesen der Daten aus einer sequentiellen Datei.....	115
TAPE – Lesen der Daten von Band.....	118
ESDS – Lesen der Daten aus einer VSAM ESDS Datei.....	122
KSDS – Lesen der Daten aus einer VSAM KSDS Datei.....	124
XPSHFS – Lesen der Daten vom XPS HFS-Dateisystem	127
FTP – Lesen der Daten von einem FTP-Server	129
HFS – Lesen der Daten aus dem OpenMVS HFS-Dateisystem	133
JES – Lesen der Daten aus dem z/OS Spoolsystem.....	135
POWER – Lesen der Daten aus dem z/VSE Spoolsystem.....	136
Ausgabedaten.....	139
ARCHIVE – Schreiben der Daten in ein PKZIP-kompatibles Archiv	140
LIB – Schreiben der Daten in eine Bibliothek	141
NET – Schreiben der Daten auf ein Netzwerklaufwerk	144
SEQ – Schreiben der Daten in eine sequentielle Datei	148
TAPE – Schreiben der Daten auf Band	152
ESDS – Schreiben der Daten in eine VSAM ESDS Datei	156
KSDS – Schreiben der Daten in eine VSAM KSDS Datei	160
XPSHFS – Schreiben der Daten in das XPS HFS Dateisystem.....	164
FTP – Versenden von Daten an einen FTP-Server	167
HFS – Schreiben der Daten in das OpenMVS HFS-Dateisystem	171
JES – Schreiben der Daten in das z/OS Spoolsystem.....	174
POWER – Schreiben der Daten in das z/VSE Spoolsystem	177
EMAIL – Versenden von Daten als E-Mail Anhang.....	181
Satzstrukturen.....	186
Struktur hinzufügen.....	186
Struktur löschen	187
Records.....	188
Exit.....	189
Verzeichnis der Beispieljobs.....	191
Beispieljobs für z/OS	191
Beispieljobs für z/VSE	193

Abbildungen

Abb. 1: Kompatible PKZIP JCL	8
Abb. 2: Installations-Job z/OS.....	10
Abb. 3: Beispieljob zur Erstellung der XPSHFS-Datei - z/OS.....	11
Abb. 4: Installations-Job z/VSE.....	12
Abb. 5: Beispieljob zur Erstellung der XPSHFS-Datei - z/VSE.....	13
Abb. 6: Codepage Datei	14
Abb. 7: Beispieljob z/OS	16
Abb. 8: Ausgabe des ISPF Kommandos '/d parmlib'	17
Abb. 9: Inhalt des Member XPSPRM00.....	17
Abb. 10: Beispieljob z/VSE.....	18
Abb. 11: Systemkennzeichen in XPSHDINI.....	19
Abb. 12: Variablenkennzeichen in XPSHDINI.....	36
Abb. 13: Variablen in Jobkarten	36
Abb. 14: Beispiel für Wildcard und Exclude-Filter	39
Abb. 15: Beispiel für Include-Filter.....	40
Abb. 16: Beispieljob für die Aktionen GZIP und GUNZIP z/OS.....	42
Abb. 17: Beispieljob für die Aktionen GZIP und GUNZIP z/VSE	43
Abb. 18: Beispieljob für die Aktionen ZIP und UNZIP z/OS	57
Abb. 19: Beispieljob für die Aktionen ZIP und UNZIP z/VSE.....	58
Abb. 20: Beispieljob für die Aktion ZIP,ADD z/OS.....	78
Abb. 21: Beispieljob für die Aktion ZIP,ADD z/VSE.....	79
Abb. 22: Beispieljob für die Aktion ZIP,COPY z/OS.....	83
Abb. 23: Beispieljob für die Aktion ZIP,COPY z/VSE.....	84
Abb. 24: Beispieljob für die Aktion ZIP,DEL z/OS.....	85
Abb. 25: Beispieljob für die Aktion ZIP,DEL z/VSE.....	85
Abb. 26: Beispieljob für die Aktion ZIP,UPD z/OS	86
Abb. 27: Beispieljob für die Aktion ZIP,UPD z/VSE.....	86
Abb. 28: Beispieljob für die Aktion ZIP,VIEW z/OS	91
Abb. 29: Beispieljob für die Aktion ZIP,VIEW z/VSE.....	91
Abb. 30: Beispieljob für die Aktionen ENCRYPT und DECRYPT z/OS.....	92
Abb. 31: Beispieljob für die Aktionen ENCRYPT und DECRYPT z/VSE.....	93
Abb. 32: Beispieljob für die Ausführung eines Netzwerkprogrammes - z/OS.....	99
Abb. 33: Beispieljob für die Ausführung eines Netzwerkprogrammes - z/VSE.....	100
Abb. 34: Beispieljob für die Veranlassung der WFD-Verarbeitung - z/OS	102
Abb. 35: Beispieljob für die Veranlassung der WFD-Verarbeitung - z/VSE	102
Abb. 36: Beispiel für Eingabekarten	103
Abb. 37: Beispieljob für IPT Eingabe z/OS	106
Abb. 38: Beispieljob für IPT Eingabe z/VSE	107
Abb. 39: Beispieljob für LIB Eingabe z/OS	109
Abb. 40: Beispieljob für LIB Eingabe z/VSE.....	110
Abb. 41: Beispieljob für NET Eingabe z/OS.....	112
Abb. 42: Beispieljob für NET Eingabe z/VSE.....	113
Abb. 43: Beispieljob für SEQ Eingabe z/OS.....	115
Abb. 44: Beispieljob für SEQ Eingabe z/VSE.....	116
Abb. 45: Beispieljob für TAPE Eingabe z/OS.....	118
Abb. 46: Beispieljob für TAPE Eingabe z/VSE	119
Abb. 47: Beispieljob für ESDS Eingabe z/OS	122
Abb. 48: Beispieljob für ESDS Eingabe z/VSE.....	122
Abb. 49: Beispieljob für KSDS Eingabe z/OS	124
Abb. 50: Beispieljob für KSDS Eingabe z/VSE.....	124
Abb. 51: Beispieljob für XPSHFS Eingabe z/OS.....	127
Abb. 52: Beispieljob für XPSHFS Eingabe z/VSE.....	127
Abb. 49: Beispieljob für KSDS Eingabe z/OS	129
Abb. 50: Beispieljob für KSDS Eingabe z/VSE.....	129
Abb. 55: Beispieljob für HFS Eingabe z/OS	133
Abb. 56: Beispieljob für JES Eingabe z/OS.....	135
Abb. 57: Definition JES Selektionskriterium.....	136
Abb. 58: Beispieljob für POWER Eingabe z/VSE	137
Abb. 59: Definition POWER Selektionskriterium.....	138

Abb. 60: Beispiel für Ausgabekarten	139
Abb. 61: Beispieljob für LIB Ausgabe z/OS	141
Abb. 62: Beispieljob für LIB Ausgabe z/VSE.....	142
Abb. 63: Beispieljob für NET Ausgabe z/OS.....	144
Abb. 64: Beispieljob für NET Ausgabe z/VSE.....	145
Abb. 65: Beispieljob für SEQ Ausgabe z/OS.....	148
Abb. 66: Beispieljob für SEQ Ausgabe z/VSE.....	148
Abb. 67: Beispieljob für TAPE Ausgabe z/OS	152
Abb. 68: Beispieljob für TAPE Ausgabe z/VSE	152
Abb. 69: Beispieljob für ESDS Ausgabe z/OS.....	156
Abb. 70: Beispieljob für ESDS Ausgabe z/VSE.....	157
Abb. 71: Beispieljob für KSDS Ausgabe z/OS	160
Abb. 72: Beispieljob für KSDS Ausgabe z/VSE.....	161
Abb. 73: Beispieljob für XPSHFS Ausgabe z/OS.....	164
Abb. 74: Beispieljob für XPSHFS Ausgabe z/VSE.....	165
Abb. 75: Beispieljob für FTP Ausgabe z/OS	167
Abb. 76: Beispieljob für FTP Ausgabe z/VSE.....	167
Abb. 77: Beispieljob für HFS Ausgabe z/OS.....	171
Abb. 78: Beispieljob für JES Ausgabe z/OS	174
Abb. 79: Beispieljob für POWER Ausgabe z/VSE	177
Abb. 80: Beispieljob für E-Mail Ausgabe z/OS	181
Abb. 81: Beispieljob für E-Mail Ausgabe z/VSE	181
Abb. 82: Job zum Strukturimport z/OS	186
Abb. 83: Job zum Strukturimport z/VSE	186
Abb. 84: Satzstrukturanzeige Online HostDrive/z.....	188
Abb. 85: Beispiel für die Verwendung einer RECORD-Angabe.....	188
Abb. 86: Beispiel für die Verwendung von 'R'-Steuerkarten.....	189
Abb. 87: COBOL Beispielprogramm für XPSZZIP Exit	190

Einführung

Allgemein

Die Software Produkte zZip und zZipSecure von der XPS Software GmbH erweitern das Portfolio der XPS im Bereich der Mainframe-Connectivity um eine PKWARE/PKZIP-kompatible Applikation zur Komprimierung, Entkomprimierung und zur optionalen Ver- bzw. Entschlüsselung von Daten, die auf einem IBM System z Mainframe gespeichert sind bzw. dort abgelegt werden sollen.

Darüber hinaus stellen zZip und zZipSecure eine Reihe von Verfahren für den Austausch von Daten zwischen dem Mainframe und anderen Plattformen zur Verfügung.

Auf Grund der Heterogenität der heute eingesetzten Netzwerkstrukturen sind sichere und kontrollierbare Mechanismen zum Austausch von Daten zwischen unterschiedlichen Plattformen von entscheidender Bedeutung. Diese Anforderung ist in vielen Unternehmen häufig sogar kritisch und muss erfüllt sein, damit die Betriebsfähigkeit gewährleistet werden kann.

Mit zZip bzw. zZipSecure stellt XPS ein Software System zur Verfügung, das diesen Ansprüchen unter besonderer Berücksichtigung der IBM zSeries Betriebssysteme gerecht wird.

Gegenstand dieses Handbuchs ist die Beschreibung der zZip und zZipSecure Batchprozessoren, die es ermöglichen, die bereit gestellten Funktionen im Rahmen der Ausführung von Batchjobs unter z/OS bzw. z/VSE zu nutzen.

Unter z/OS können darüber hinaus rechenintensive Prozesse auf zIIP (IBM System z Integrated Information Processor) ausgelagert werden, was zum Einen zur Reduzierung von Kosten führt, die durch Prozessornutzung entstehen und zum Anderen zur besseren Nutzung der verfügbaren z/OS Ressourcen.

Für die optionale Kommunikation mit dem jeweils zu Grunde liegenden Spoolsystem – JES2/JES3 unter z/OS, Power unter z/VSE – nutzen zZip bzw. zZipSecure die Funktionalität von PrintEx, einem von XPS entwickelten System zur Erweiterung der Druckmöglichkeiten auf IBM Mainframes.

Der Zugriff auf Datenquellen bzw. auf Datenspeicher außerhalb des Mainframes wird – mit Ausnahme von FTP und E-Mail - über eine Kommunikation mit HostDrive/J abgewickelt. HostDrive/J ist eine Serverapplikation, die auf jeder Java-fähigen Plattform ausgeführt werden kann.

Weitere Informationen zu PrintEx und HostDrive/J sind den jeweiligen Handbüchern zu entnehmen.

zZip wird in zwei verschiedenen Editionen angeboten. Die zZipSecure Edition enthält den gesamten Funktionsumfang während in der zZip Edition keine kryptographischen Funktionen zur Verfügung stehen. Außer zum Zweck der expliziten Unterscheidung wird in diesem Handbuch immer die Bezeichnung zZip verwendet, die in jedem Fall zZipSecure beinhaltet.

PKZIP Kompatibilität

Durch die Verwendung von Adapterprogrammen unterstützt zZip unter MVS bzw. zSeries das JCL-Format von PKZIP.

Damit können PKZIP Anwender ihre Jobs zur Komprimierung und Entkomprimierung – auch mit Ver- und Entschlüsselung - prinzipiell ohne Anpassung mit zZip ausführen.

Die einzig erforderliche Änderung beschränkt sich darauf, die STEPLIB der Jobs so anzupassen, dass statt den PKZIP Lademodulen die zZip Lademodule geladen werden.

Obwohl zum gegenwärtigen Zeitpunkt noch nicht alle PKZIP Optionen unterstützt werden, kann doch der Großteil der PKZIP Jobs ohne weitere Änderung von zZip ausgeführt werden. Die Durchführung konkreter Tests kann im Zweifelsfall Klarheit schaffen.

Die XPS Software GmbH vergrößert die Anzahl der unterstützten PKZIP Optionen fortwährend, wobei explizite Kunden- bzw. Interessentenanforderungen selbstverständlich vorrangig implementiert werden.

Die nachfolgende Abbildung zeigt eine PKZIP JCL, bei der lediglich das STEPLIB-Statement auf die zZip Ladebibliothek verweist.

Darüber hinaus wurde noch eine spezielle '-ZZIP_' Option eingefügt. Diese speziellen Optionen dienen lediglich dazu, bestimmte zZip Trace- und Analysefunktionen zu aktivieren, und sind im Rahmen der normalen Job Ausführung nicht erforderlich.

```
//JPKZIP JOB CLASS=A,MSGCLASS=M
//*****
//* DATEI PACKEN *****
//*****
//STEPZIP EXEC PGM=PKZIP,REGION=32M
//STEPLIB DD DISP=SHR,DSN=XPSHDRV.V300.LOADLIB
//SYSIN DD *
-ECHO
-ARCHIVE(PKZIP.XPS.TEST.REPAIR.LIT.SORT.ZIP)
-ARCHUNIT(3390)
-ADD
-DATA_TYPE(TEXT)
PKZIP.XPS.TEST.REPAIR.LIT.SORT
/*
```

Abb. 1: Kompatible PKZIP JCL

Erläuterung:

Die zZip PKZIP Präprozessoren 'PKZIP' bzw. 'PKUNZIP' (Aliasnamen für 'XPSPKZIP' bzw. 'XPSPKUNZ') analysieren die jeweilige PKZIP JCL und generieren daraus zZip kompatible Eingabekarten.

Der jeweils aktive Präprozessor lädt nach Analyse bzw. Konvertierung der PKZIP Steuerkarten den zZip Hauptprozessor 'XPSZZIP' und übermittelt die generierten zZip Steuerkarten.

Installation

Vorbemerkungen

zZip ist Bestandteil der HostDrive/z Distribution. Die nachfolgend beschriebenen Schritte zur Installation von zZip sind nicht erforderlich, wenn HostDrive/z auf dem Zielrechner bereits installiert ist.

Falls die Installation der optionalen Komponenten PrintEx oder HostDrive/J gewünscht wird, sind die entsprechenden Handbücher zu Rate zu ziehen.

zZip – z/OS

zZip läuft unter OS/390 ab Version 1.3, sowie unter z/OS ab Version 1.1. Hierbei wird TCPIP/MVS ab Version 3.1 bzw. der Communications Server for OS/390 ab Version 2.4 vorausgesetzt.

In Bezug auf die Hardware ist die Installation auf einen IBM System/390 oder kompatiblen Prozessor ausgelegt, auf dem die zuvor genannten Softwarekomponenten lauffähig sind. Für die Installation werden ca. 400 MB Festplattenplatz für die Bibliotheken benötigt. Zum Einspielen des Installationsdatenträgers wird entweder ein Band-/Cartridgelaufwerk oder ein Client-PC mit einem CD-Laufwerk und einer FTP-Anbindung an den Host benötigt.

Einspielen des zZip-Installationsbandes

Folgende Bibliotheken sind für die Installationsdateien vorzubereiten:

Name	Space	Lrecl	Blksz	Recfm
XPSHDRV.V300.LOADLIB	6144,(300,50,50)		6144	U
XPSHDRV.V300.MACLIB	3200,(30,5,10)	80	3200	FB

Die 'Loadlib' enthält den ausführbaren Programmcode. In der 'Maclib' befinden sich die Installationsjobs, Beispiele zur Anpassung an die bestehende Systemumgebung, sowie Dateien, die zur Laufzeit von zZip benötigt werden, wie z. B. die Lizenzdatei.

Beispieljob

```
//XPSH300 JOB , 'INSTALL XPS-ZZIP'
// CLASS=c,MSGCLASS=x
//TAPL EXEC PGM=IEBCOPY
//LOADIN DD DISP=(OLD,PASS),VOL=(,RETAIN,SER=XPSH20),
// LABEL=(1,SL),DSN=XPSH300.LOADLIB,
// UNIT=cart
```

```
//LOADOUT DD DISP=(NEW,CATLG),DSN=xpshdrv.V300.loadlib,
//          SPACE=(6144,(300,50,50),,,ROUND),DCB=SYS1.LINKLIB,
//          VOL=SER=mvs001,UNIT=dasd
//MACIN DD DISP=(OLD,PASS),VOL=(,RETAIN,SER=XPSH20),
//          LABEL=(2,SL),DSN=XPSH300.MACLIB,
//          UNIT=cart
//MACOUT DD DISP=(NEW,CATLG),DSN=xpshdrv.V300.maclib,
//          SPACE=(3200,(30,5,10),,,ROUND),DCB=SYS1.MACLIB,
//          VOL=SER=mvs001,UNIT=dasd
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COPY INDD=LOADIN,OUTDD=LOADOUT
COPY INDD=MACIN,OUTDD=MACOUT
```

Abb. 2: Installations-Job z/OS

Einspielen des zZip-Installations-CD-Rom

Die Installationsbibliotheken auf der Installations-CD-Rom müssen mit Hilfe eines FTP-Clientprogramms zum Hostrechner übertragen werden, auf dem zZip installiert werden soll. Die Bibliotheken stehen im TSO-Transmit-Format auf der CD-Rom und sind binär zum Host zu übertragen. Vor der Übertragung sind die Dateien auf dem empfangenden Server zu definieren und anzulegen. Dazu sind folgende Werte anzugeben:

Name	Space	Lrecl	Blksz	Recfm
XMIT.XPSHDRV.V300.LOADLIB	600,(100)	80	3200	FB
XMIT.XPSHDRV.V300.MACLIB	200,(20)	80	3200	FB

Danach können die nachfolgend genannten Dateien aus dem CD-Rom Unterverzeichnis '**MVS**' vom Client zum Host gesendet werden. Dabei sind die Dateien folgendermaßen umzubenennen:

Clientname	Hostname
XPSH300L.BIN	XMIT.XPSHDRV.V300.LOADLIB
XPSH300M.BIN	XMIT.XPSHDRV.V300.MACLIB

Anschließend sind die TSO-Transfer-Dateien durch folgende TSO-Befehle zu übertragen:

Loadlib

```
RECEIVE INDSN(XMIT.XPSHDRV.V300.LOADLIB)
```

Nach Eingabe des 'RECEIVE'-Befehls erscheint folgender Prompt:

```
INMR901I Dataset XPSHDRV.V300.LOADLIB from XPSSYST on NODENAME
INMR906A Enter restore parameters or 'DELETE' or 'END' +
```

Hier sind der gewünschte Dateiname sowie das Plattenvolumen folgendermaßen anzugeben:

```
DSN(xpshdrv.V300.loadlib) VOL(mvs001)
```

Maclib

```
RECEIVE INDSN(XMIT.XPSHDRV.V300.MACLIB)
```

Nach Eingabe des 'RECEIVE'-Befehls erscheint folgender Prompt:

```
INMR901I Dataset XPSHDRV.V300.MACLIB from XPSSYST on NODENAME
INMR906A Enter restore parameters or 'DELETE' or 'END' +
```

Hier sind der gewünschte Dateiname sowie das Plattenvolume folgendermaßen anzugeben:

```
DSN(xpshdrv.V300.maclib) VOL(mvs001)
```

Definition der XPSHFSWFD-Datei

zZip benötigt für verschiedene Aufgaben die Präsenz einer XPSHFS-Datei. Diese auf VSAM-RRDS basierende Datei implementiert ein hierarchisches Dateisystem, das z. B. zur Speicherung temporärer Informationen während der Batchläufe verwendet wird.

Falls zZip im Zusammenspiel mit einer HostDrive/z Basisinstallation eingesetzt wird, besteht die Möglichkeit, Datentransfers, die zum Zeitpunkt der Jobausführung nicht durchgeführt werden konnten, zu einem späteren Zeitpunkt automatisch nachzuholen. Diese Vorgehensweise wird als wfd-Verarbeitung – 'waiting for delivery' - bezeichnet.

Um die wfd-Verarbeitung zu ermöglichen, speichert zZip die dazu benötigten Informationen in der XPSHFS-Datei ab.

Die XPSHFS-Datei ist vor der ersten Nutzung unter Verwendung des Systemprogramms 'IDCAMS', wie im nachstehenden Beispieljob gezeigt, zu erstellen und zu initialisieren.

```
//JINIT JOB , 'IDCAMS', CLASS=A, MSGCLASS=M
//DELDEF EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE xpshdrv.xpshfs.file CLUSTER PURGE
DEFINE CL(NAME(xpshdrv.xpshfs.file) -
        NUMBERED -
        SHR(2) -
        VOL(volume)) -
        DATA(NAME(xpshdrv.xpshfs.file.data) -
        RECSZ(32761 32761) -
        CISZ(32768) -
        REC(30000))
//*
//INIT EXEC PGM=XPSSPMNT, REGION=32M
//STEPLIB DD DISP=SHR, DSN=XPSHDRV.V300.LOADLIB
//XPSHFS DD DISP=SHR, DSN=xpshdrv.xpshfs.file
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SPOOLLOG DD SYSOUT=*
//SYSIN DD *
LANG D
INIT
//*
```

Abb. 3: Beispieljob zur Erstellung der XPSHFS-Datei - z/OS

Die Satzgröße 'RECSZ' bzw. die Größe eines Kontrollintervalls 'CISZ' und die Anzahl der Sätze können prinzipiell frei gewählt werden, wobei die Satzgröße mindestens 4089 Bytes betragen sollte.

Es ist zu berücksichtigen, dass die XPSHFS-Datei unter Umständen große Datenmengen speichern muss, weswegen ausreichend große Werte sowohl für die Satzgröße als auch für die Anzahl der Sätze zu wählen sind.

Die VSAM-seitigen Beschränkungen für 'RECSZ' und 'CISZ' betragen 32761 bzw. 32768.

zZip - z/VSE

zZip läuft unter VSE/ESA ab Version 2.5., sowie unter z/VSE ab Version 3.1

In Bezug auf die Hardware ist die Installation auf einen IBM System/390 oder kompatiblen Prozessor ausgelegt, auf dem die zuvor genannten Softwarekomponenten lauffähig sind. Für die Installation werden ca. 400 MB Festplattenplatz für die Bibliotheken benötigt. Zum Einspielen des Installationsdatenträgers wird entweder ein Band-/Cartridgelaufwerk oder ein Client-PC mit einem CD-Laufwerk und einer FTP-Anbindung an den Host benötigt.

Einspielen des zZip-Installationsbandes

Beispieljob:

```
// JOB ZZIP   INSTALL ZZIP
// ASSGN SYS006,tape
// EXEC LIBR
  RESTORE SUB=XPS.HDRVV300:lib.sublib -
  TAPE=SYS006 LIST=YES REPLACE=YES
/*
/ &
```

Abb. 4: Installations-Job z/VSE

Definition der XPSHFS/WFD-Datei

zZip benötigt für verschiedene Aufgaben die Präsenz einer XPSHFS-Datei. Diese auf VSAM-RRDS basierende Datei implementiert ein hierarchisches Dateisystem, das z. B. zur Speicherung temporärer Informationen während der Batchläufe verwendet wird.

Falls zZip im Zusammenspiel mit einer HostDrive/z Basisinstallation eingesetzt wird, besteht die Möglichkeit, Datentransfers, die zum Zeitpunkt der Jobausführung nicht durchgeführt werden konnten, zu einem späteren Zeitpunkt automatisch nachzuholen. Diese Vorgehensweise wird als wfd-Verarbeitung – 'waiting for delivery' - bezeichnet.

Um die wfd-Verarbeitung zu ermöglichen, speichert zZip die dazu benötigten Informationen in der XPSHFS-Datei ab.

Die XPSHFS-Datei ist vor der ersten Nutzung unter Verwendung des Systemprogramms 'IDCAMS', wie im nachstehenden Beispieljob gezeigt, zu erstellen und zu initialisieren.

```
* $$ JOB JNM=XPSHFS,DISP=D,PRI=3,CLASS=0
* $$ LST CLASS=A
// JOB XPSSPMNT
// EXEC IDCAMS
DELETE (xpshdrv.xpshfs.file) CL PURGE -
      CATALOG(user.catalog)
```

```

DEFINE CL(NAME(xpshdrv.xpshfs.file)           -
          NUMBERED                           -
          SHR(2)                               -
          VOL(volume)                         -
          RECSZ(27641 27641)                  -
          CISZ(27648)                         -
          REC(50000))                          -
          DATA(NAME(xpshdrv.xpshfs.file.data)) -
          CATALOG(user.catalog)
/*
// LIBDEF PHASE,SEARCH=(XPS.HDRV300)
// DLBL XPSHFS,'xpshdrv.xpshfs.file',,VSAM,CAT=XPSUCAT
// ASSGN SYS005,SYSLST
// EXEC XPSSPMNT,SIZE=AUTO
LANG D
INIT
/*
/&
* $$ EOJ

```

Abb. 5: Beispieljob zur Erstellung der XPSHFS-Datei - z/VSE

Die Satzgröße 'RECSZ' bzw. die Größe eines Kontrollintervalls 'CISZ' und die Anzahl der Sätze können prinzipiell frei gewählt werden, wobei die Satzgröße mindestens 4089 Bytes betragen sollte.

Es ist zu berücksichtigen, dass die XPSHFS-Datei unter Umständen große Datenmengen speichern muss, weswegen ausreichend große Werte sowohl für die Satzgröße als auch für die Anzahl der Sätze zu wählen sind.

Die VSAM-seitigen Beschränkungen für 'RECSZ' und 'CISZ' betragen 32761 bzw. 32768.

Codepages

Im Installationsumfang befindet sich eine Textdatei mit Namen 'CODEPAGE' (z/OS) bzw. 'CODEPAGE.D' (z/VSE), die Tabellen enthält, die zZip für die Übersetzung von Textdaten verwendet.

Die Übersetzung wird erforderlich, wenn Textdateien zwischen verschiedenen Betriebssystemen ausgetauscht werden, die jeweils unterschiedliche Darstellungen für Textdaten verwenden.

Textdateien auf dem Mainframe werden 'EBCDIC'-kodierte gespeichert. 'EBCDIC' steht für 'Extended Binary Coded Decimals Interchange Code'.

Im Gegensatz dazu werden Textdateien auf anderen Betriebssystemen häufig im 'ASCII'-Format gespeichert. 'ASCII' steht für 'American Standard Code for Information Interchange'.

Auf Microsoft Windows und Apple Macintosh Betriebssystemen wird das 'ANSI'-Format als Standard verwendet, das eine Erweiterung des 'ASCII'-Codes darstellt. 'ANSI' ist ein Zeichensatz, der vom 'American National Standards Institute' normiert wurde.

Die nachfolgende Abbildung zeigt einen Ausschnitt aus der Datei 'CODEPAGE', die sich nach der Installation in der zZip Data-Bibliothek befindet:

Abb. 6: Codepage Datei

Die Datei enthält für jede Codepage eine Zeichentabelle, die nach einem festgelegten Format aufgebaut sein muss, das nachfolgend beschrieben wird.

Wenn neue Codepages hinzugefügt werden, ist darauf zu achten, dass das geforderte Format eingehalten wird.

Im Überschriftbereich jeder Tabelle befindet sich die Schlüsselinformation, die mit der Kennung 'CP=' eingeleitet wird. Der sich daran anschließende Text bis zum ersten Leerzeichen wird als Schlüssel für die folgende Tabelle verwendet und muss innerhalb der Datei eindeutig sein. Im Anschluss an den Schlüssel folgt eine kurze Beschreibung des Landes, dessen Sonderzeichen in der Tabelle enthalten sind.

Daran anschließend können beliebig viele Kommentarzeilen folgen, die jeweils mit einem '*' beginnen müssen und beliebigen Inhalt haben können. Die nächste Zeile, die nicht mit einem Kommentarzeichen beginnt, wird von zZip als Tabellenbeginn interpretiert. Bis zum Tabellenende dürfen keine weiteren Kommentarzeilen folgen. Um die Lesbarkeit zu erhöhen enthält die Standarddatei für jede Tabelle eine Überschrift, die das hexadezimale Offset jedes Zeichens zum Zeilenanfang enthält:

```
*0 1 2 3 4 5 6 7 8 9 A B C D E F
```

Im Anschluss daran folgt die Übersetzungstabelle, die aus 16 Zeilen mit jeweils 32 Zeichen besteht. Dabei enthalten jeweils 2 benachbarte Zeichen den Code in hexadezimaler Darstellung, der zur Übersetzung des Zeichens zu verwenden ist, das dem Index des Zeichens in der Tabelle entspricht.

Wie der oben abgebildeten Tabelle – English (US), EBCDIC nach ASCII - zu entnehmen ist, wird z. B. das Leerzeichen in EBCDIC - Index Hex40 (erstes Zeichen in Zeile 8, die der fünften Tabellenzeile entspricht) – nach Hex20, dem Leerzeichen in ASCII, übersetzt.

Durch Veränderungen an bestehenden oder Hinzufügen eigener Codepages können länder- oder regionalspezifische Besonderheiten berücksichtigt werden. Alle zZip Funktionen, die die Verwendung einer Codepage zur Zeichenübersetzung erfordern, ermöglichen die Angabe des Schlüssels der zu verwendenden Codepage – im Beispiel '1140-E2A'.

Die nachfolgende Tabelle zeigt die Liste aller Codepages, die nach der Installation verfügbar sind:

Schlüssel	Beschreibung
1140-E2A	Englisch (US) - EBCDIC nach ASCII
1141-E2A	Österreichisch / Deutsch - EBCDIC nach ASCII
1142-E2A	Dänisch / Norwegisch - EBCDIC nach ASCII
1143-E2A	Finnisch / Schwedisch - EBCDIC nach ASCII
1144-E2A	Italienisch - EBCDIC nach ASCII
1145-E2A	Spanisch - EBCDIC nach ASCII
1146-E2A	Englisch (UK) - EBCDIC nach ASCII
1147-E2A	Französisch - EBCDIC nach ASCII
1148-E2A	Belgisch / Schweizerisch - EBCDIC nach ASCII
1250-E2A	Polnisch - EBCDIC nach ASCII
1140-E2N	Englisch (US) - EBCDIC nach ANSI
1141-E2N	Österreichisch / Deutsch - EBCDIC nach ANSI
1142-E2N	Dänisch / Norwegisch - EBCDIC nach ANSI
1143-E2N	Finnisch / Schwedisch - EBCDIC nach ANSI
1144-E2N	Italienisch - EBCDIC nach ANSI
1145-E2N	Spanisch - EBCDIC nach ANSI
1146-E2N	Englisch (UK) - EBCDIC nach ANSI
1147-E2N	Französisch - EBCDIC nach ANSI
1148-E2N	Belgisch / Schweizerisch - EBCDIC nach ANSI
1250-E2N	Polnisch - EBCDIC nach ANSI
1140-A2E	Englisch (US) - EBCDIC nach ANSI
1141-A2E	Österreichisch / Deutsch – ASCII nach EBCDIC
1142-A2E	Dänisch / Norwegisch - ASCII nach EBCDIC
1143-A2E	Finnisch / Schwedisch - ASCII nach EBCDIC
1144-A2E	Italienisch - ASCII nach EBCDIC
1145-A2E	Spanisch - ASCII nach EBCDIC
1146-A2E	Englisch (UK) - ASCII nach EBCDIC
1147-A2E	Französisch - ASCII nach EBCDIC
1148-A2E	Belgisch / Schweizerisch - ASCII nach EBCDIC
1140-N2E	Englisch (US) – ANSI nach EBCDIC
1141-N2E	Österreichisch / Deutsch – ANSI nach EBCDIC
1142-N2E	Dänisch / Norwegisch - ANSI nach EBCDIC
1143-N2E	Finnisch / Schwedisch - ANSI nach EBCDIC
1144-N2E	Italienisch - ANSI nach EBCDIC
1145-N2E	Spanisch - ANSI nach EBCDIC
1146-N2E	Englisch (UK) - ANSI nach EBCDIC
1147-N2E	Französisch - ANSI nach EBCDIC
1148-N2E	Belgisch / Schweizerisch - ANSI nach EBCDIC

Konfiguration

Datenbibliothek 'XPSDATA'

zZip benötigt zur Ausführung den Zugriff auf einige Dateien wie z. B. auf die Lizenzdatei, auf Codepage-Dateien für die Übersetzung von Textdaten und auf die Initialisierungsdatei.

Dazu ist es erforderlich, zZip über die Bibliothek, die diese Dateien enthält, in Kenntnis zu setzen.

Die Bekanntgabe der zu verwendenden Datenbibliothek erfolgt je nach Betriebssystem auf unterschiedliche Weise.

z/OS

Job-Control

Der nachfolgend abgebildete Jobstream zeigt ein Beispiel für einen zZip Job unter z/OS:

```
//JHDZL2N JOB , 'ZIPPED LIB TO NET', CLASS=E, MSGCLASS=X
//JHDZL2N EXEC PGM=XPSZZIP, REGION=48M
//STEPLIB DD DISP=SHR, DSN=XPSHDRV.V300.LOADLIB
//XPSDATA DD DISP=SHR, DSN=XPSHDRV.V300.MACLIB
//XPSIN DD DISP=SHR, DSN=XPSDAEM.V500.MACLIB
//SYSIN DD *
A ACTION=ZIP PATH=N
I LIB=XPSIN MBR=XPSVP0* EXT=TXT FCOMM='some source code'
F EXCLUDE=XPSVP09
O ARCHIVE=NET='CEETEMP/xpsvp0.zip' CP=1141-E2N CRLF
O IPADDR=192.168.0.9 IPPORT=8840 ROUTE=WS08
//*
```

Abb. 7: Beispieljob z/OS

Die Bekanntgabe der benötigten XPSDATA-Bibliothek erfolgt im abgebildeten Job unter Verwendung eines 'XPSDATA DD'-Statements, das auf die zu verwendende Bibliothek verweist.

Bei dieser Art der Nutzung ist sicherzustellen, dass das Job-Control-Statement in jeden zZip Job aufgenommen wird.

z/OS Parmlibs

Alternativ kann die zu verwendende XPSDATA-Bibliothek über ein Konfigurationsmember in einer der aktiven z/OS Parmlibs bekannt gegeben werden. Diese Methode wird aktiv, wenn der Jobstream kein DD-Statement für die XPSDATA-Bibliothek enthält.

Die aktiven z/OS Parmlibs können unter Verwendung des TSO Kommandos '/d parmlib' auf der z/OS Systemkonsole ausgegeben werden. Die resultierende Anzeige entspricht in etwa der nachfolgend abgebildeten:

```

XPS_zOS_#1 - JProtector
Datei Bearbeiten Ansicht Makro Ohio Skript Verwaltung Einstellungen Verbindung Drucker Sessions Hilfe
-----
Display Filter View Print Options Help
-----
SDSF SYSLOG 2586.106 S0W1 S0W1 11/09/2011 0W 4171 COMMAND ISSUED
COMMAND INPUT ==> _ SCROLL ==> CSR
RESPONSE=S0W1
IEE251I 15.38.53 PARMLIB DISPLAY 251
PARMLIB DATA SETS SPECIFIED
AT IPL
ENTRY FLAGS VOLUME DATA SET
1 S VPMVSD VENDOR.PARMLIB
2 S VTMVSG SVTSC.PARMLIB
3 S VTLVL0 LVL0.PARMLIB
4 D VIMVSB SYS1.PARMLIB
0000 $HASP686 OUTPUT(XPSCOMPR) OUTGRP=2 1 1 CANCELLED

```

Abb. 8: Ausgabe des ISPF Kommandos '/d parmlib'

zZip durchsucht in diesem Fall die Kette der definierten Parmlibs nach einem Member mit dem Namen 'XPSPRM00'.

Die nachfolgende Abbildung zeigt den erwarteten Inhalt des Members 'XPSPRM00':

```

JProtector
Datei Bearbeiten Ansicht Makro Ohio Skript Verwaltung Einstellungen Verbindung Drucker Sessions Hilfe
Kontrollzentrum
TN_zOS_#1
File Edit Edit_Settings Menu Utilities Compilers Text Help
EDIT USER.PARMLIB(XPSPRM00) - 01.02 Columns 00001 00072
Command ==> Scroll ==> CSR
***** Top of Data *****
000001 XPSDATA=XPSMDRV.HDRV.MACLIB
***** Bottom of Data *****

```

Abb. 9: Inhalt des Member XPSPRM00

Die Referenzierung der 'XPSDATA'-Bibliothek über das 'XPSPRMnn' Member erlaubt den Einsatz unterschiedlicher Konfigurationen.

Dazu kann das zweiziffrige Suffix 'nn', im oben besprochenen Standardfall '00', verwendet werden.

Um z. B. die Verwendung einer alternativen 'XPSDATA'-Bibliothek zu erreichen, kann ein Member mit Namen 'XPSPRM01' erstellt und in eine der aktiven Parmlibs kopiert werden.

zZip ist in diesem, vom Standard abweichenden Fall, über die alternative Konfiguration zu informieren, indem in der Jobkarte unter Verwendung der 'PARM=' Option das zu verwendende Suffix mit 'INI=01' angegeben wird.

zZip sucht dann nach einem Member mit Namen 'XPSPRM01' in der Liste der aktiven Parmlibs, um die zu verwendende 'XPSDATA'-Bibliothek zu ermitteln.

Als einziges Statement wird die Angabe der XPSDATA-Bibliothek gemäß dem folgenden Format erwartet:

```
XPSDATA=xpsdata-maclib
```

z/VSE

Job-Control

Der nachfolgend abgebildete Jobstream zeigt ein Beispiel für einen zZip Job unter z/VSE:

```
* $$ LST CLASS=T,JNM=JHDXCMD
// JOB JHDXCMD
// LIBDEF PHASE,SEARCH=(XPS.HDRV300)
// OPTION SYSPARM='00'
// ASSGN SYS005,SYSLST
// DLBL XPSHFS,'XPSHDRV.SPOOL.SAVE',,VSAM,CAT=XPSUCAT
// EXEC XPSZZIP,SIZE=AUTO,PARM='LIB=XPS.HDRV300'
X CMD=J:\HOSTDRIVE\BATCH\COLLECT.BAT WAIT=YES WTO=YES
X     PARM='VSE2'
X     PARM='TEST'
/*
/&
* $$EQJ
```

Abb. 10: Beispieljob z/VSE

Die Bekanntgabe der benötigten XPSDATA-Bibliothek erfolgt im abgebildeten Job unter Verwendung der 'PARM'-Option auf der 'EXEC'-Karte.

Die Angabe für die XPSDATA-Bibliothek wird gemäß dem folgenden Format erwartet:

```
PARM='LIB=xpsdata-library'
```

Eine weitere Möglichkeit zur Angabe der zu verwendenden XPSDATA-Bibliothek besteht darin, diese, am geeignetsten im Rahmen des IPL, mit der folgenden Environment Variablen systemweit bekannt zu geben:

```
// SETPARM SYSTEM,XPSLIB='xpsdata-library'
```

Initialisierungsdatei 'XPSHDINI'

zZip verwendet im Rahmen der Verarbeitung eine Initialisierungsdatei namens 'XPSHDINI' (z/OS) bzw. 'XPSHDINI.D' (z/VSE), in der viele Laufzeitoptionen mit Vorbelegungen initialisiert werden können.

Durch die Verwendung von Vorbelegungen für bestimmte Optionen entfällt das Aufführen der entsprechenden Optionen in den zZip Steuerkarten, falls keine von der Vorbelegung abweichende Angabe erforderlich ist. Damit werden Wartbarkeit und Flexibilität des Systems verbessert.

Die erste Zeile der Initialisierungsdatei muss das Kennzeichen '+SYSTEM+' für den Beginn der Systemeinstellungen enthalten, wie nachfolgend abgebildet:

```

EDIT XPSHDRV.HDRV.MACLIB(XPSHDINI) - 01.99          Columns 00001 00072
Command ===)          Scroll ===) CSR
***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you change
==MSG>          your edit profile using the command RECOVERY ON.
000001 +SYSTEM+
000002 ARC_BLKSIZE=0
000003 ARC_DATACLASS=
000004 ARC_DSORG=PS
000005 ARC_RECFL=U
000006 ARC_LRECL=0
000007 ARC_DIR_BLK5=56

```

Abb. 11: Systemkennzeichen in XPSHDINI

Systemeinstellungen

Die jeweiligen Voreinstellungen sind in den nachfolgenden Zeilen durchzuführen. Im Rahmen der Installation wird eine Initialisierungsdatei mit Standardvorbelegungen kopiert, die angepasst werden kann.

Die nachfolgende Tabelle zeigt alle verfügbaren Voreinstellungen, die über einen entsprechenden Eintrag in der Initialisierungsdatei vorgenommen werden können.

Schlüssel	Erklärung										
ARC_BLKSIZE=	numerischer Wert 'DYNAMIC' 'OPTIMUM' 'MAXIMUM'										
	<p>Mit dieser Option kann die gewünschte Blockgröße festgelegt werden, die zZip zur Erstellung von sequentiellen Dateien bzw. von PO-Datasets verwenden soll, falls diese als Ausgabeziel für einen Komprimierungsvorgang verwendet werden.</p> <p>Die Angabe eines numerischen Wertes wird in Bytes erwartet.</p> <p>Die nachfolgende Tabelle beschreibt die Vorgehensweise bei der Auswahl eines der vordefinierten, konstanten Werte:</p> <table border="1"> <thead> <tr> <th>Option</th> <th>Bedeutung</th> </tr> </thead> <tbody> <tr> <td><i>DYNAMIC</i></td> <td>zZip ermittelt die zu verwendende Blockgröße dynamisch.</td> </tr> <tr> <td><i>OPTIMUM</i></td> <td>zZip ermittelt die zu verwendende Blockgröße dynamisch und optimiert diese für das Zielgerät.</td> </tr> <tr> <td><i>MAXIMUM</i></td> <td>zZip ermittelt die zu verwendende Blockgröße dynamisch und wählt die größtmögliche Blockgröße für das Zielgerät.</td> </tr> <tr> <td>Beispiele:</td> <td>ARC_BLKSIZE=4096 ARC_BLKSIZE=OPTIMUM</td> </tr> </tbody> </table>	Option	Bedeutung	<i>DYNAMIC</i>	zZip ermittelt die zu verwendende Blockgröße dynamisch.	<i>OPTIMUM</i>	zZip ermittelt die zu verwendende Blockgröße dynamisch und optimiert diese für das Zielgerät.	<i>MAXIMUM</i>	zZip ermittelt die zu verwendende Blockgröße dynamisch und wählt die größtmögliche Blockgröße für das Zielgerät.	Beispiele:	ARC_BLKSIZE=4096 ARC_BLKSIZE=OPTIMUM
Option	Bedeutung										
<i>DYNAMIC</i>	zZip ermittelt die zu verwendende Blockgröße dynamisch.										
<i>OPTIMUM</i>	zZip ermittelt die zu verwendende Blockgröße dynamisch und optimiert diese für das Zielgerät.										
<i>MAXIMUM</i>	zZip ermittelt die zu verwendende Blockgröße dynamisch und wählt die größtmögliche Blockgröße für das Zielgerät.										
Beispiele:	ARC_BLKSIZE=4096 ARC_BLKSIZE=OPTIMUM										
ARC_DATACLASS=	maximal 8-stelliger Name einer SMS Data Class										
	<p>Mit dieser Option kann die SMS Data Class festgelegt werden, die für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Komprimierungsvorgangs verwendet wird.</p> <table border="1"> <tbody> <tr> <td>Beispiel</td> <td>ARC_DATACLASS=DC2</td> </tr> </tbody> </table>	Beispiel	ARC_DATACLASS=DC2								
Beispiel	ARC_DATACLASS=DC2										
ARC_DIR_BLK5=	numerischer Wert										
	<p>Mit dieser Option kann die Anzahl der Verzeichnisblöcke angegeben werden, die zZip für eine Bibliothek reservieren soll, die als Ausgabeziel für einen</p>										

	Komprimiervorgang verwendet wird.	
	Beispiel	ARC_DIR_BLKs=56
ARC_DSORG=	'SEQ' 'PS' 'PDS' 'PO'	
	Mit dieser Option ist zZip über den Organisationstyp einer neu anzulegenden Datei zu informieren, die als Ausgabeziel für einen Komprimiervorgang verwendet wird.	
	<i>SEQ</i> <i>PS</i>	Bei Wahl einer dieser Optionen erstellt zZip eine neue Datei mit Organisationstyp 'sequentiell' (sequential / partitioned sequential).
	<i>PDS</i> <i>PO</i>	Bei Wahl einer dieser Optionen erstellt zZip eine neue Bibliothek (partitioned data set / partitioned organized).
ARC_MGMTCLASS=	maximal 8-stelliger Name einer SMS Management Class	
	Mit dieser Option kann die SMS Management Class festgelegt werden, die für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Komprimiervorgangs verwendet wird.	
	Beispiel	ARC_MGMTCLASS=MAG30LIB
ARC_LRECL=	numerischer Wert	
	Mit dieser numerischen Option kann die logische Satzlänge festgelegt werden, die für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Komprimiervorgangs verwendet wird.	
	Beispiel	ARC_LREC=132
ARC_RECFM=	'F' 'FB' 'V' 'VB' 'U'	
	Mit dieser Option kann das Satzformat festgelegt werden, das für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Komprimiervorgangs verwendet wird.	
	<i>F</i>	Die Zieldatei verwendet Sätze mit fester Satzlänge.
	<i>FB</i>	Die Zieldatei verwendet Sätze mit fester Satzlänge, die geblockt verarbeitet werden.
	<i>V</i>	Die Zieldatei verwendet Sätze mit variabler Satzlänge.
	<i>VB</i>	Die Zieldatei verwendet Sätze mit variabler Satzlänge, die geblockt verarbeitet werden.
	<i>U</i>	Die Zieldatei verwendet Sätze mit undefinierter Satzlänge.
ARC_SPACE_PRI=	numerischer Wert	
	Mit dieser Option kann die Anzahl der Allokationseinheiten im Primary Extent für eine neu anzulegende sequentielle Datei bzw. für ein PO-Dataset festgelegt werden, die als Ausgabeziel für einen Komprimiervorgang verwendet werden.	
	Beispiel	ARC_SPACE_PRI=25
ARC_SPACE_RLSE=	'Y' 'N'	

	Mit dieser Option ist festzulegen, ob zZip nicht benötigten und zuvor reservierten Plattenplatz für eine neu erstellte sequentielle Datei bzw. für ein neu erstelltes PO-Dataset freigeben soll, nachdem der Komprimiervorgang beendet wurde.	
	Beispiel	ARC_SPACE_RLSE=Y
ARC_SPACE_SEC=	numerischer Wert	
	Mit dieser Option kann die Anzahl der Allokationseinheiten im Secondary Extent für eine neu anzulegende sequentielle Datei bzw. für ein PO-Dataset festgelegt werden, die als Ausgabeziel für einen Komprimiervorgang verwendet werden.	
	Beispiel	ARC_SPACE_SEC=10
ARC_SPACE_TYPE=	'CYL' 'TRK' 'BLK'	
	Mit dieser Option ist der Typ der Allokationseinheiten für eine neu anzulegende sequentielle Datei bzw. für ein PO-Dataset festzulegen, die als Ausgabeziel für einen Komprimiervorgang verwendet werden. Numerische Angaben für die Optionen 'ARC_SPACE_PRI' und 'ARC_SPACE_SEC' beziehen sich auf den hier festgelegten Typ.	
	Beispiel	ARC_SPACE_TYPE=CYL
ARC_STORCLASS=	maximal 8-stelliger Name einer SMS Storage Class	
	Mit dieser Option kann die SMS Storage Class festgelegt werden, die für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Komprimiervorgangs verwendet wird.	
	Beispiel	ARC_STORCLASS=SMSSTOR
ARC_UNIT=	maximal 8-stelliger Name einer Unit	
	Mit dieser Option ist der Typ der Einheit anzugeben, auf der zZip eine neu anzulegende sequentielle Datei bzw. ein PO-Dataset allokkieren soll, die als Ausgabeziel für einen Komprimiervorgang verwendet werden.	
	Beispiel	ARC_UNIT=SYSDA
ARC_VOLUME=	maximal 8-stelliger Name eines Volumes	
	Mit dieser Option ist der Name des Volumes anzugeben, auf der zZip eine neu anzulegende sequentielle Datei bzw. ein PO-Dataset allokkieren soll, die als Ausgabeziel für einen Komprimiervorgang verwendet werden.	
	Beispiel	ARC_VOLUME=XPS011
ATTR=	'Y' 'N'	
	Mit dieser Option ist festzulegen, wie bei Komprimierung und Entkomprimierung mit Dateiattributen verfahren werden soll.	
	Aktion	Bedeutung von 'ATTR=Y'
	ZIP	Die verfügbaren Dateiattribute werden in das ZIP-Archiv

		inkludiert.
AUTODEL=	numerischer Wert	
	Mit dieser Option kann festgelegt werden, nach wie viel Tagen zZip Dateien, die nicht zugestellt werden konnten, aus der Sicherungsdatei (wfd) löscht. Diese Option findet nur Verwendung, wenn zZip im Verbund mit einer HostDrive/z Basisinstallation betrieben wird.	
	Beispiel	AUTODEL=7
BUFSIZE=	numerischer Wert gefolgt von 'K' 'M'	
	Mit dieser Option kann die Größe der intern genutzten Puffer in Kilobytes ('K') oder in Megabytes ('M') festgelegt werden. zZip benötigt bis zu 4 dieser Puffer gleichzeitig. Bei der Wahl der Puffergröße ist daher darauf zu achten, dass die ausführende Region/Partition groß genug gewählt wird, um, neben dem Grundbedarf an virtuellem Speicher, auch die erforderlichen Arbeitspuffer bereitstellen zu können.	
	Beispiele	BUFSIZE=256K BUFSIZE=2M
COMPLVL=	numerischer Wert zwischen '0' ... '9' 'STORE' 'SUPERFAST' 'FAST' 'NORMAL' 'MAXIMUM'	
	Mit dieser Option ist der Komprimierungslevel bei der Erstellung komprimierter Archive festzulegen. Mit steigendem Wert verringert sich die Größe des erstellten Archivs auf Kosten der benötigten CPU Zeit. Neben der Angabe eines numerischen Wertes kann auch eine der vordefinierten Komprimierungsstufen gewählt werden:	
	vordefinierte Komprimierungsstufe	entspricht numerischem Wert
	STORE	0
	SUPERFAST	1
	FAST	2
	NORMAL	3
	MAXIMUM	4
CONTINUE=	darstellbares Zeichen	
	Mit dieser Option kann ein Zeichen definiert werden, mit dem zZip darüber in Kenntnis gesetzt wird, dass sich die Angabe einer Option über mehr als eine Karte erstreckt, was z. B. bei der Angabe von Dateinamen erforderlich sein kann. Damit die Präsenz einer Fortsetzungskarte erkannt wird, ist das hier definierte Zeichen in Spalte 72 der fortzusetzenden Karte einzutragen.	
	Beispiel	CONTINUE=*
CNVEXT=	'NAMEFILE' 'KEEP' 'DROP' 'SUFFIX' 'MERGE'	

	<p>Mit dieser Option ist festzulegen, wie zZip mit der Erweiterung von Dateinamen verfahren soll, die beim Entpacken von komprimierten Archiven anfallen.</p> <p>Die nachfolgende Tabelle zeigt den resultierenden Name nach Anwendung der möglichen Optionen auf den Dateinamen 'XPSHDRV.TEST.STATS.TXT'.</p>	
	Option	Resultat
	<i>NAMEFILE, KEEP</i>	XPSHDRV.TEST.STATS.TXT
	<i>DROP</i>	XPSHDRV.TEST.STATS
	<i>SUFFIX, MERGE</i>	XPSHDRV.TEST.STATSTXT
COMMCHAR=	ein beliebiges Zeichen	
	<p>Mit dieser Option kann das Kommentarzeichen festgelegt werden. zZip überliest Zeilen komplett, die mit dem hier angegebenen Kommentarzeichen beginnen (Spalte 1).</p>	
	Beispiel	COMMCHAR=*
CONCAT=	'Y' 'N'	
	<p>Mit dieser Option ist festzulegen, wie verfahren werden soll, falls mehrere Dateien als Eingabe für die Erstellung einer GZIP-Datei bereitstehen, da GZIP normalerweise nur eine Datei komprimieren kann.</p> <p>Bei Auswahl der Option 'CONCAT=Y' fügt zZip alle Eingabedateien zur GZIP-Datei hinzu.</p> <p>Bei Auswahl von 'CONCAT=N' wird nur die erste zur Verfügung stehende Eingabedatei in das GZIP-Archiv eingefügt.</p>	
CPA2E=	Schlüssel einer Codepage	
	<p>Mit dieser Option kann festgelegt werden, mit welcher Codepage Textinformationen, wie z. B. Dateinamen und Kommentare beim Lesen aus komprimierten Archiven oder eingehende Textinformationen des FTP- und E-Mail-Protokolls übersetzt werden sollen.</p> <p>Diese Informationen sind aus Gründen der Standardisierung immer in ASCII/ANSI kodiert. Weitere Informationen zu Codepages können im Kapitel 'Codepages' auf Seite '13' nachgelesen werden.</p> <p>Darüber hinaus wird die hier angegebene Codepage als Standard für die Übersetzung von Textdaten von ASCII/ANSI nach EBCDIC verwendet, falls in den Steuerkarten keine spezielle Codepage angegeben wird.</p>	
	Beispiel	CPA2E=1141-N2E
CPE2A=	Schlüssel einer Codepage	
	<p>Mit dieser Option kann festgelegt werden, mit welcher Codepage Textinformationen, wie z. B. Dateinamen und Kommentare beim Schreiben von komprimierten Archiven oder ausgehende Textinformationen des FTP- und E-Mail-Protokolls übersetzt werden sollen.</p> <p>Diese Informationen sind aus Gründen der Standardisierung immer in ASCII/ANSI kodiert. Weitere Informationen zu Codepages können im Kapitel 'Codepages' auf Seite '13' nachgelesen werden.</p> <p>Darüber hinaus wird die hier angegebene Codepage als Standard für die</p>	

	Übersetzung von Textdaten von EBCDIC nach ASCII/ANSI verwendet, falls in den Steuerkarten keine spezielle Codepage angegeben wird.		
	Beispiel	CPOUT=1141-E2N	
CRYPTOFACILITY=	'XPS' 'ICSF' 'CPACF' <i>* nur in der zZipSecure Edition verfügbar</i>		
	<p>Mit dieser Option ist festzulegen, ob kryptographische Funktionen, die im Rahmen der zZipSecure Verarbeitung anfallen, über Software, oder über einen als Hardware installierten Krypto-Coprozessor ausgeführt werden.</p> <p>Bei Angabe von 'XPS' wird die XPS-CryptLib zur softwaretechnischen Ausführung kryptographischer Funktionen verwendet. Unter z/OS können bei Präsenz eines zIIP Prozessors (IBM System z Integrated Information Processor) CPU intensive kryptographische Rechenprozesse auf zIIP ausgelagert werden, was zur Einsparung von Ressourcen und Kosten führt – siehe Option 'ZIIP' auf Seite 35.</p> <p>In den beiden anderen Fällen wird die jeweils installierte Hardwarelösung angesteuert. 'ICSF' steht für 'Integrated Cryptographic Service Facility' unter z/OS und 'CPACF' steht für 'Central Processor Assist for Cryptographic Function' unter z/OS und z/VSE.</p>		
	Beispiel	CRYPTOFACILITY=XPS	
CRYPTOMETHOD=	Algorithmus <i>* nur in der zZipSecure Edition verfügbar</i>		
	<p><u>Definition:</u> Offen gelegtes Rechenverfahren, das festlegt, wie aus Blöcken von Klartext unter Verwendung eines geheimen Schlüssels Chiffre-Blöcke erzeugt werden, die nur bei Kenntnis des geheimen Schlüssels wieder in Klartext-Blöcke zurückverwandelt werden können.</p> <p>Mit dieser Option ist die Voreinstellung für den Algorithmus anzugeben, der zum Ver- bzw. zum Entschlüsseln von Daten verwendet werden soll. Bei der Nutzung eines Crypto-Coprozessors ist sicherzustellen, dass ein von zZipSecure unterstützter Algorithmus gewählt wird.</p> <p>Die nachfolgende Tabelle zeigt die unterstützten Algorithmen sowie die jeweils verwendete Schlüssellänge in Bit:</p>		
	Algorithmus	Beschreibung	Schlüssellänge in Bit
	AES	Advanced Encryption Standard	128
	AES,128	Advanced Encryption Standard	128
	AES,192	Advanced Encryption Standard	192
	AES,256	Advanced Encryption Standard	256
	DES	Data Encryption Standard	56
	DES,56	Data Encryption Standard	56
	DES,112	Data Encryption Standard	112
	DES,168	Data Encryption Standard	168
	3DES	Triple Data Encryption Standard	168
	RC2	Rivest Cipher 2	128

	RC2,40	Rivest Cipher 2	40														
	RC2,64	Rivest Cipher 2	64														
	RC2,128	Rivest Cipher 2	128														
	RC4	Rivest Cipher 4	128														
	RC4,40	Rivest Cipher 4	40														
	RC4,64	Rivest Cipher 4	64														
	RC4,128	Rivest Cipher 4	128														
	BLOWFISH	Blockverschlüsselung / Bruce Schneier	128														
	BLOWFISH,128	Blockverschlüsselung / Bruce Schneier	128														
	Beispiele	CRYPTOMETHOD=3DES CRYPTOMETHOD=RC4,128															
CRYPTOMODE=	'ECB' 'CBC' 'ECB_CTS' 'CBC_CTS' 'OFB' 'CTR' 'CFB' * nur in der zZipSecure Edition verfügbar																
	<p><u>Definition:</u> Ein kryptographischer Modus legt fest, wie sich die Verschlüsselung mehrerer Klartextblöcke vollzieht, indem er definiert, in welcher Art der Verschlüsselungsalgorithmus auf den Datenstrom angewandt wird.</p> <p>Mit dieser Option ist die Voreinstellung für den Modus für die symmetrische Ver- bzw. Entschlüsselung von Daten einzustellen.</p> <p>Es ist sicherzustellen, dass der gewählte Modus von der eingesetzten Cryptofacility unterstützt wird.</p> <p>Der Cryptomode dient u. a. dazu, zu verhindern, dass gleichlautende Abschnitte der zu verschlüsselnden Daten in identischem Chiffretext resultieren, was bei der symmetrischen, blockbasierten Verschlüsselung unter Umständen der Fall wäre.</p> <p>Dies wird normalerweise dadurch erreicht, dass die verschiedenen Modi in jeweils unterschiedlicher Weise das Verschlüsselungsergebnis des zuvor verschlüsselten Blocks in die Berechnung der Verschlüsselung für den nächsten Block mit einbeziehen.</p> <p>Folgende Modi können gewählt werden:</p> <table> <tbody> <tr> <td>ECB</td> <td>Electronic codebook</td> </tr> <tr> <td>CBC</td> <td>Cipher-block chaining</td> </tr> <tr> <td>ECB_CTS</td> <td>ECB ciphertext stealing – nur bei Verwendung der XPS CryptLib</td> </tr> <tr> <td>CBC_CTS</td> <td>CBC ciphertext stealing – nur bei Verwendung der XPS CryptLib</td> </tr> <tr> <td>OFB</td> <td>Output Feedback Mode</td> </tr> <tr> <td>CTR</td> <td>Counter</td> </tr> <tr> <td>CFB</td> <td>Cipher feedback – nur bei Verwendung von CPACF</td> </tr> </tbody> </table>			ECB	Electronic codebook	CBC	Cipher-block chaining	ECB_CTS	ECB ciphertext stealing – nur bei Verwendung der XPS CryptLib	CBC_CTS	CBC ciphertext stealing – nur bei Verwendung der XPS CryptLib	OFB	Output Feedback Mode	CTR	Counter	CFB	Cipher feedback – nur bei Verwendung von CPACF
ECB	Electronic codebook																
CBC	Cipher-block chaining																
ECB_CTS	ECB ciphertext stealing – nur bei Verwendung der XPS CryptLib																
CBC_CTS	CBC ciphertext stealing – nur bei Verwendung der XPS CryptLib																
OFB	Output Feedback Mode																
CTR	Counter																
CFB	Cipher feedback – nur bei Verwendung von CPACF																
	Beispiel	CRYPTOMODE=CBC															
CRYPTOIV=	hexadezimaler Wert (X'...') Zeichenkette * nur in der zZipSecure Edition verfügbar																

	<p>Definition: Meist zufällig bestimmter Startwert, der von manchen Betriebsmodi (z. B. CBC) eines Block-Chiffre vor dem ersten Datenblock bearbeitet wird.</p> <p>Mit dieser Option kann der Initialisierungsvektor vorgelegt werden, der für symmetrische Ver- und Entschlüsselung in den entsprechenden Modi (z. B.: CBC, OFB, CFB) erforderlich ist.</p> <p>Der Initialisierungsvektor muss immer genau so groß wie die Blocklänge des Chiffre sein:</p> <table border="0" data-bbox="446 470 798 660"> <tr> <td>DES</td> <td>8 Bytes</td> </tr> <tr> <td>RC2</td> <td>8 Bytes</td> </tr> <tr> <td>BLOWFISH</td> <td>8 Bytes</td> </tr> <tr> <td>AES</td> <td>16 Bytes</td> </tr> </table> <p>Der Vektor ist entweder hexadezimal mit 2 Stellen pro Zeichen oder als Zeichenkette anzugeben, die bei Bedarf mit Leerzeichen aufgefüllt wird.</p>			DES	8 Bytes	RC2	8 Bytes	BLOWFISH	8 Bytes	AES	16 Bytes	
DES	8 Bytes											
RC2	8 Bytes											
BLOWFISH	8 Bytes											
AES	16 Bytes											
	Beispiel (8 Bytes)	CRYPTOIV=X'0102030405060708' CRYPTOIV=GEHEIM										
CRYPTOKEY=	hexadezimaler Wert (X'...') Zeichenkette <i>* nur in der zZipSecure Edition verfügbar</i>											
	<p>Definition: Schlüssel, der bei symmetrischen Verfahren sowohl zur Verschlüsselung als auch zur Entschlüsselung eingesetzt wird.</p> <p>Mit dieser Option kann der kryptographische Schlüssel angegeben werden, der bei der symmetrischen Verschlüsselung zum Einsatz kommen soll.</p> <p>Die Länge des Schlüssels hängt zum einen vom verwendeten Algorithmus und zum anderen von der gewünschten Verschlüsselungsstärke ab. Weitere Informationen sind der Option 'CRYPTOMETHOD' auf Seite 24 zu entnehmen.</p> <p>Der Schlüssel ist entweder hexadezimal mit 2 Stellen pro Zeichen oder als Zeichenkette anzugeben, die bei Bedarf mit Leerzeichen aufgefüllt wird.</p>											
	Beispiel für DES (56 Bit/7 Bytes)	CRYPTOKEY=X'A827C149DFE623' CRYPTOKEY=GEHEIM										
DATEFMT=	'1' '2'											
	<p>Mit dieser Option kann das Format des Datums festgelegt werden, in dem es von zZip in das Joblog geschrieben wird. Die folgende Tabelle zeigt die verfügbaren Formate.</p> <table border="1" data-bbox="446 1601 957 1747"> <thead> <tr> <th>DATEMFT</th> <th>Beschreibung</th> <th>Resultat</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>amerikanisch</td> <td>MM/TT/YY</td> </tr> <tr> <td>2</td> <td>deutsch</td> <td>TT.MM.YY</td> </tr> </tbody> </table>			DATEMFT	Beschreibung	Resultat	1	amerikanisch	MM/TT/YY	2	deutsch	TT.MM.YY
DATEMFT	Beschreibung	Resultat										
1	amerikanisch	MM/TT/YY										
2	deutsch	TT.MM.YY										
ECHO=	'Y' 'N'											
	Mit dieser Option ist festzulegen, ob zZip alle gelesenen Eingabekarten im Joblog protokollieren soll oder nicht.											
EXTSEP=	ein beliebiges Zeichen											
	Mit dieser Option ist festzulegen, welches Zeichen von zZip als Trennzeichen für eine Dateierweiterung interpretiert werden soll. Diese Angabe ist z. B. im											

	Zusammenhang mit der Option 'CNVEXT' von Interesse, die auf Seite 22 beschrieben ist.	
	Beispiel	EXTSEP=.
FILE_TERM=	hexadezimaler Wert (X'...') vordefinierter Wert	
	<p>Unter Umständen kann es erforderlich oder gewünscht sein, dass das Ende einer Textdatei durch ein bestimmtes Zeichen bzw. durch eine bestimmte Zeichenkombination explizit kenntlich gemacht werden soll.</p> <p>Um zu erreichen, dass zZip ein besonderes Dateiende-Kennzeichen für Textdateien einfügt, ist das gewünschte Dateiende-Kennzeichen mit dieser Option festzulegen.</p>	
	Beispiele	FILE_TERM=X'0D0A0A0D' FILE_TERM=CRLFCZ
	Die nachfolgende Tabelle zeigt eine Auflistung der vordefinierten Werte, die als Dateiende-Kennzeichen verwendet werden können:	
	Bezeichner	Wert
	A-CRLFCZ	X'0D0A1A'
	CRLFCZ	X'0D0A1A'
	A-CRLF	X'0D0A'
	A-LFCR	X'0A0D'
	E-CRLF	X'0D25'
	E-CRNL	X'0D25'
	E-LFCR	X'250D'
	A-NEL	X'85'
	CRLF	X'0D0A'
	LFCR	X'0A0D'
	A-CR	X'0D'
	A-LF	X'0A'
	A-CZ	X'1A'
	E-CR	X'0D'
	E-LF	X'25'
	E-NL	X'15'
	CR	X'0D'
	LF	X'0A'
	CZ	X'1A'
IPADDR=	TCP/IP Adresse – numerisch oder als DNS-Name	
	Mit dieser Option ist anzugeben, welche TCP/IP Adresse von zZip als Standard verwendet werden soll.	
	Beispiele	IPADDR=127.0.0.1

	IPADDR=www.xps-software.de
IPPORT=	numerischer TCP/IP Port <= 65535
	Mit dieser Option ist anzugeben, welcher TCP/IP Port von zZip als Standard verwendet werden soll.
	Beispiel IPPORT=8880
MODE=	'TEXT' 'BIN' 'DETECT'
	Mit dieser Option ist festzulegen, auf welche Weise zZip die zu verarbeitenden Daten interpretieren soll. Bei Angabe von 'TEXT' interpretiert zZip die Daten als Textdaten und verwendet die jeweils angegebene Codepage zur Übersetzung der Textdaten vor dem Komprimieren bzw. nach dem Entkomprimieren. Bei Angabe von 'BIN' werden die Daten als binär interpretiert und es findet keine Übersetzung statt. Falls der Standard 'DETECT' verwendet wird, analysiert zZip die Daten vor dem Komprimieren bzw. nach dem Entkomprimieren und ermittelt den Datentyp unter Berücksichtigung der Angaben für die Optionen 'MODE_DETECT_TAB' und 'MODE_DETECT_LEN'.
	Beispiel MODE=DETECT
MODE_DETECT_LEN=	numerischer Wert gefolgt von 'K' 'M' 'R'
	Falls zZip den Typ der verarbeiteten Daten erkennen soll ('MODE=DETECT'), analysiert zZip die Daten an Hand der mit der Option 'MODE_DETECT_TAB=' angegebenen Tabelle. Mit dieser Option kann festgelegt werden, wie viel Zeichen zZip in diesem Fall maximal überprüfen soll, bis die Entscheidung, ob die Datei ausschließlich Textdaten enthält oder nicht, getroffen werden soll. Die Angabe kann entweder in Kilobytes ('K'), in Megabytes ('M') oder – für Textdateien - als Anzahl von Sätzen ('R') erfolgen.
	Beispiele MODE_DETECT_LEN=8K MODE_DETECT_LEN=2000R
MODE_DETECT_TAB=	Name eines Lademoduls das eine Testtabelle enthält
	Bei Wahl von 'MODE=DETECT' ermittelt zZip den Typ der Daten an Hand einer Tabelle, mit der festgestellt wird, ob die Daten Zeichen enthalten, die nicht als Textzeichen gelten. Im Lieferumfang von zZip befindet sich eine Standardtabelle namens 'DETECTXT', die als gleichnamiges Textmodul in der XPSDATA Bibliothek abgelegt ist. Falls diese Tabelle den Anforderungen nicht entsprechen sollte, können Veränderungen entweder direkt an dieser Tabelle durchgeführt werden oder an einer Kopie, die dann unter Verwendung der 'MODE_DETECT_TAB=' Option bekanntgegeben wird. Damit eine Tabelle verwendet kann, muss diese zur Laufzeit als assembliertes Lademodul verfügbar sein.
	Beispiel MODE_DETECT_TAB=DETECTXT

OBSVADDR=	TCP/IP Adresse – numerisch oder als DNS-Name	
	<p>zZip kann beim Versenden von Daten an einen HostDrive/J Server ('O NET=') bestimmte Informationen an einen HostDrive/J Observer übermitteln, um den Datenfluss transparent zu protokollieren.</p> <p>Wenn dies gewünscht ist, ist mit dieser Option die TCP/IP Adresse anzugeben, die von dem HostDriver/J Server überwacht wird, an den die Observer Informationen zu übermitteln sind.</p>	
	Beispiel	OBSVADDR=192.168.0.131
OBSVPORT=	numerischer TCP/IP Port <= 65535	
	<p>zZip kann beim Versenden von Daten an einen HostDrive/J Server ('O NET=') bestimmte Informationen an einen HostDrive/J Observer übermitteln, um den Datenfluss transparent zu protokollieren.</p> <p>Wenn dies gewünscht ist, ist mit dieser Option der TCP/IP Port anzugeben, der von dem HostDriver/J Server überwacht wird, an den die Observer Informationen zu übermitteln sind.</p>	
	Beispiel	OBSVPORT=8887
OUT_BLKSIZE=	numerischer Wert 'DYNAMIC' 'OPTIMUM' 'MAXIMUM'	
	<p>Mit dieser Option kann die gewünschte Blockgröße festgelegt werden, die zZip zur Erstellung von sequentiellen Dateien bzw. von PO-Datasets verwenden soll, falls diese als Ausgabeziel für einen Entkomprimiervorgang verwendet werden.</p> <p>Die Angabe eines numerischen Wertes wird in Bytes erwartet.</p> <p>Die nachfolgende Tabelle beschreibt die Vorgehensweise bei der Auswahl eines der vordefinierten, konstanten Werte:</p>	
	Option	Bedeutung
	<i>DYNAMIC</i>	zZip ermittelt die zu verwendende Blockgröße dynamisch.
	<i>OPTIMUM</i>	zZip ermittelt die zu verwendende Blockgröße dynamisch und optimiert diese für das Zielgerät.
	<i>MAXIMUM</i>	zZip ermittelt die zu verwendende Blockgröße dynamisch und wählt die größtmögliche Blockgröße für das Zielgerät.
	Beispiele:	OUT_BLKSIZE=2048 OUT_BLKSIZE=DYNAMIC
OUT_DATACLASS=	maximal 8-stelliger Name einer SMS Data Class	
	<p>Mit dieser Option kann die SMS Data Class festgelegt werden, die für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Entkomprimiervorgangs verwendet wird.</p>	
	Beispiel	OUT_DATACLASS=DC03
OUT_DIR_BLKS=	numerischer Wert	
	<p>Mit dieser Option kann die Anzahl der Verzeichnisblöcke angegeben werden, die zZip für eine Bibliothek reservieren soll, die als Ausgabeziel für einen Entkomprimiervorgang verwendet wird.</p>	

	Beispiel	OUT_DIR_BLKS=100
OUT_DSORG=	'SEQ' 'PS' 'PDS' 'PO'	
	Mit dieser Option ist zZip über den Organisationstyp einer neu anzulegenden Datei zu informieren, die als Ausgabeziel für einen Entkomprimierungsvorgang verwendet wird.	
	<i>SEQ</i> <i>PS</i>	Bei Wahl einer dieser Optionen erstellt zZip eine neue Datei mit Organisationstyp 'sequentiell' (sequential / partitioned sequential).
	<i>PDS</i> <i>PO</i>	Bei Wahl einer dieser Optionen erstellt zZip eine neue Bibliothek (partitioned data set / partitioned organized).
OUT_MGMTCLASS=	maximal 8-stelliger Name einer SMS Management Class	
	Mit dieser Option kann die SMS Management Class festgelegt werden, die für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Entkomprimierungsvorgangs verwendet wird.	
	Beispiel	OUT_MGMTCLASS=MGMUZLIB
OUT_LRECL=	numerischer Wert	
	Mit dieser numerischen Option kann die logische Satzlänge festgelegt werden, die für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Entkomprimierungsvorgangs verwendet wird.	
	Beispiel	OUT_LRECL=80
OUT_RECFCM=	'F' 'FB' 'V' 'VB' 'U'	
	Mit dieser Option kann das Satzformat festgelegt werden, das für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Entkomprimierungsvorgangs verwendet wird.	
	<i>F</i>	Die Zieldatei verwendet Sätze mit fester Satzlänge.
	<i>FB</i>	Die Zieldatei verwendet Sätze mit fester Satzlänge, die geblockt verarbeitet werden.
	<i>V</i>	Die Zieldatei verwendet Sätze mit variabler Satzlänge.
	<i>VB</i>	Die Zieldatei verwendet Sätze mit variabler Satzlänge, die geblockt verarbeitet werden.
	<i>U</i>	Die Zieldatei verwendet Sätze mit undefinierter Satzlänge.
OUT_SPACE_PRI=	numerischer Wert	
	Mit dieser Option kann die Anzahl der Allokationseinheiten im Primary Extent für eine neu anzulegende sequentielle Datei bzw. für ein PO-Dataset festgelegt werden, die als Ausgabeziel für einen Entkomprimierungsvorgang verwendet werden.	
	Beispiel	OUT_SPACE_PRI=40
OUT_SPACE_RLSE=	'Y' 'N'	
	Mit dieser Option ist festzulegen, ob zZip nicht benötigten und zuvor reservierten Plattenplatz für eine neu erstellte sequentielle Datei bzw. für ein	

	neu erstelltes PO-Dataset freigeben soll, nachdem der Entkomprimiervorgang beendet wurde.	
	Beispiel	OUT_SPACE_RLSE=N
OUT_SPACE_SEC=	numerischer Wert	
	Mit dieser Option kann die Anzahl der Allokationseinheiten im Secondary Extent für eine neu anzulegende sequentielle Datei bzw. für ein PO-Dataset festgelegt werden, die als Ausgabeziel für einen Entkomprimiervorgang verwendet werden.	
	Beispiel	OUT_SPACE_SEC=30
OUT_SPACE_TYPE=	'CYL' 'TRK' 'BLK'	
	Mit dieser Option ist der Typ der Allokationseinheiten für eine neu anzulegende sequentielle Datei bzw. für ein PO-Dataset festzulegen, die als Ausgabeziel für einen Entkomprimiervorgang verwendet werden. Angaben für die Optionen 'OUT_SPACE_PRI' und 'OUT_SPACE_SEC' beziehen sich auf den hier festgelegten Typ.	
	Beispiel	OUT_SPACE_TYPE=TRK
OUT_STORCLASS=	maximal 8-stelliger Name einer SMS Storage Class	
	Mit dieser Option kann die SMS Storage Class festgelegt werden, die für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Entkomprimiervorgangs verwendet wird.	
	Beispiel	OUT_STORCLASS=UNZSTOR
OUT_UNIT=	maximal 8-stelliger Name einer Unit	
	Mit dieser Option ist der Typ der Einheit anzugeben, auf der zZip eine neu anzulegende sequentielle Datei bzw. ein PO-Dataset allokiert soll, die als Ausgabeziel für einen Entkomprimiervorgang verwendet werden.	
	Beispiel	OUT_UNIT=SYSDA
OUT_VOLUME=	maximal 8-stelliger Name eines Volumes	
	Mit dieser Option ist der Name des Volumes anzugeben, auf der zZip eine neu anzulegende sequentielle Datei bzw. ein PO-Dataset allokiert soll, die als Ausgabeziel für einen Entkomprimiervorgang verwendet werden.	
	Beispiel	OUT_VOLUME=WRKLB2
OVERWRITE=	'Y' 'N'	
	Mit dieser Option ist festzulegen, ob zZip bereits bestehende Zielformate beim Entkomprimieren von Daten überschreiben soll oder nicht. Falls eine Zielformat bereits existiert und die Option 'OVERWRITE=N' gewählt wurde, protokolliert zZip dies mit einer Warnung.	
	Beispiel	OVERWRITE=Y

PADC=	hexadezimaler 2-stelliger EBCDIC Wert (X'...')	
	Mit dieser Option ist festzulegen, mit welchem Zeichen zZip Textzeilen auffüllen soll, die geblockt werden sollen, die aber nicht vollständig gefüllt sind. Das Padding Character ist als EBCDIC Wert in hexadezimaler Darstellung anzugeben.	
	Beispiel (Leerzeichen/Blank)	PADC=X'40'
PARMERR=	'Y' 'N'	
	Mit dieser Option ist festzulegen, wie zZip auf Parameterfehler reagieren soll. Bei Auswahl von 'PARMERR=Y' wird der zZip Jobstep beendet, sobald ein Parameterfehler auftritt und der Jobstep wird mit dem mit Option 'RCERROR' festgelegten Returncode beendet.	
PATH=	'Y' 'N'	
	Mit dieser Option ist festzulegen, ob zZip bei der Erstellung von Zip-Archiven lediglich die Dateinamen ('PATH=N') oder den gesamten Pfad zu den jeweiligen Dateien ('PATH=Y') in das Inhaltsverzeichnis des Archivs einfügen soll.	
PATHSEP=	darstellbares Zeichen	
	Mit dieser Option kann ein Zeichen bestimmt werden, das zZip im Rahmen der Erstellung von Pfadangaben zur Trennung einzelner Pfadbestandteile verwenden soll. Auf Microsoft Windows Plattformen ist dies üblicherweise der Backslash ('\'), wohingegen auf Unix- bzw. Linuxderivaten der Slash (/) verwendet wird.	
	Beispiel	PATHSEP=/'
PROGRESS=	'Y' 'N'	
	Mit dieser Option kann zZip mit der Protokollierung von Dateinamen und weitergehender Informationen, die im Rahmen der Komprimierung und Entkomprimierung von Interesse sind, beauftragt werden.	
	Beispiel	PROGRESS=Y
RCINFO=	numerischer Wert	
	Mit dieser Option ist festzulegen, welchen Returncode zZip zurückgeben soll, falls im Rahmen der Verarbeitung eine besondere Information ausgegeben wurde. zZip Returncodes können zur Steuerung eines Job-Ablaufs mittels 'Conditional JCL' verwendet werden.	
RCWARN=	numerischer Wert	
	Mit dieser Option ist festzulegen, welchen Returncode zZip zurückgeben soll, falls im Rahmen der Verarbeitung eine Warnung ausgegeben wurde. zZip Returncodes können zur Steuerung eines Job-Ablaufs mittels 'Conditional JCL' verwendet werden.	
RCNOTFND=	numerischer Wert	

	Mit dieser Option ist festzulegen, welchen Returncode zZip zurückgeben soll, falls im Rahmen der Verarbeitung eine erforderliche Datei nicht gefunden wurde. zZip Returncodes können zur Steuerung eines Job-Ablaufs mittels 'Conditional JCL' verwendet werden.
<i>RCERROR=</i>	numerischer Wert
	Mit dieser Option ist festzulegen, welchen Returncode zZip zurückgeben soll, falls im Rahmen der Verarbeitung ein Fehler aufgetreten ist. zZip Returncodes können zur Steuerung eines Job-Ablaufs mittels 'Conditional JCL' verwendet werden.
<i>REC_DELIM=</i>	hexadezimaler Wert (X'...') vordefinierter Wert
	Mit dieser Option kann festgelegt werden, welches Zeichen bzw. welche Zeichenkombination zZip als Zeilenende-Kennzeichen bei der Verarbeitung von Textdaten verwenden soll.
Beispiele	REC_DELIM=X'0D0A' REC_DELIM=CR
	Die nachfolgende Tabelle zeigt eine Auflistung der vordefinierten Werte, die als Zeilenende-Kennzeichen verwendet werden können:
Bezeichner	Wert
<i>A-CRLFCZ</i>	X'0D0A1A'
<i>CRLFCZ</i>	X'0D0A1A'
<i>A-CRLF</i>	X'0D0A'
<i>A-LFCR</i>	X'0A0D'
<i>E-CRLF</i>	X'0D25'
<i>E-CRNL</i>	X'0D25'
<i>E-LFCR</i>	X'250D'
<i>A-NEL</i>	X'85'
<i>CRLF</i>	X'0D0A'
<i>LFCR</i>	X'0A0D'
<i>A-CR</i>	X'0D'
<i>A-LF</i>	X'0A'
<i>A-CZ</i>	X'1A'
<i>E-CR</i>	X'0D'
<i>E-LF</i>	X'25'
<i>E-NL</i>	X'15'
<i>CR</i>	X'0D'
<i>LF</i>	X'0A'
<i>CZ</i>	X'1A'
<i>TCPJOB=</i>	Name des TCP/IP Jobs

	<p>Mit dieser Option, die nur unter z/OS Verwendung findet, ist zZip über den Namen der TCP/IP Started Task zu informieren, über die der TCP/IP Datenverkehr erfolgen soll.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>TCPJOB=TCPIP</td> </tr> </table>	Beispiel	TCPJOB=TCPIP
Beispiel	TCPJOB=TCPIP		
WFD=	'Y' 'N'		
	<p>Mit dieser Option ist festzulegen, ob zZip eine Sicherungskopie der Eingabedaten auf der verbundenen XPS HFS-Datei erstellen soll. Diese Option wird nur dann berücksichtigt, wenn die Ausgabedaten über einen HostDrive/z Server zugestellt werden.</p> <p>In diesem Fall kann die wfd-Verarbeitung des HostDrive Queuing Systems genutzt werden. Die wfd-Verarbeitung (waiting for delivery) ermöglicht die automatische Zustellung von Dateien an das gewünschte Ziel zu einem späteren Zeitpunkt, falls dieses zur Ausführungszeit des zZip Jobs nicht erreicht werden kann.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>WFD=Y</td> </tr> </table>	Beispiel	WFD=Y
Beispiel	WFD=Y		
WTO=	'Y' 'N'		
	<p>Mit dieser Option ist festzulegen, ob zZip beim Jobstart eine Nachricht über die verwendete Version und beim Jobende eine Nachricht über die Anzahl der verarbeiteten Sätze auf die Konsole ausgeben soll oder nicht.</p>		
XPSHFS=	Name einer Datei		
	<p>Mit dieser Option ist der Name der HostDrive/z HFS-Datei (VSAM RRDS) bekannt zu geben, die zZip zur Speicherung temporärer Daten verwenden soll.</p> <p>Darüber hinaus wird diese Datei zur Realisierung der wfd-Verarbeitung ('waiting for delivery') verwendet, falls zZip Dateien über eine HostDrive/z Basisinstallation versendet.</p> <p>Die wfd-Verarbeitung eröffnet die Möglichkeit, Dateien, die zum Zeitpunkt der Job-Ausführung nicht zugestellt werden konnten, zu einem späteren Zeitpunkt automatisch zustellen zu lassen.</p>		
	<p><i>z/OS</i></p> <p>Unter z/OS wird diese Datei dynamisch allokiert, weswegen kein DD-Statement im Job Control erforderlich ist.</p> <p>Zur Vermeidung von Reservierungskonflikten beim parallelen Ausführen mehrerer zZip Jobs kann die Angabe für XPSHFS zwei aufeinander folgende Fragezeichen an beliebiger Position enthalten, die als zweistelliger numerischer Platzhalter interpretiert werden wie z. B.</p> <p>XPSHFS=XPS.HDRV.FIL??SAV</p> <p>zZip versucht in diesem Fall zunächst die Datei 'XPS.HDRV.FIL01SAV' für den aktuellen Job zu allokiert. Wenn dies fehlschlägt, wird der numerische Platzhalter erhöht und somit versucht 'XPS.HDRV.FIL02SAV' zu öffnen.</p> <p>Dieses Verfahren wird so lange wiederholt, bis zZip entweder eine XPSHFS Datei erfolgreich öffnen konnte, oder bis der Versuch der dynamischen Allokation einer Datei fehlschlägt, was bedeutet, dass diese Datei nicht definiert ist.</p> <p>Bei Verwendung dieser Methode ist somit darauf zu achten, dass die</p>		

	<p>verschiedenen XPSHFD Dateien bei der Definition ohne Unterbrechung durchnummeriert werden.</p>	
	<p><i>z/VSE</i></p> <p>Unter <i>z/VSE</i> empfiehlt sich die Aufnahme der Datei in die Standardlabels, wodurch auf die Angabe des DLBL-Statements in der JCL verzichtet werden kann.</p> <p>Zur Vermeidung von Reservierungskonflikten beim parallelen Ausführen mehrerer <i>zZip</i> Jobs kann durch das Einfügen eines '%' -Zeichens als erstes Zeichen des DLBL-Statements erreicht werden, dass die ID der Partition, in der der <i>zZip</i> Job ausgeführt wird, als Suffix an den gewählten DLBL-Namen angehängt wird, wie z. B.</p> <pre>// DLBL XPSHFS,'%HDRV.SPOOL.WFD',,VSAM,CAT=XPSUCAT</pre> <p>Vor der Nutzung der verschiedenen XPSHFS Dateien ist es notwendig, einen VSAM Cluster Delete-Define-Init Job in jeder Partition laufen zu lassen, in der auch <i>zZip</i> Jobs ausgeführt werden sollen. Auch im Delete-Define-Init Job ist bei jeder Referenzierung der XPSHFS Datei dem Namen das '%' -Zeichen voranzustellen.</p> <p>Somit werden z. B. bei Nutzung der 'BG' und 'F8' Partition die folgenden XPSHFS-Dateien erstellt:</p> <pre>HDRV.SPOOL.WFD.BG HDRV.SPOOL.WFD.F8</pre> <p>die dann auch beim Ausführen eines <i>zZip</i> Jobs in der jeweiligen Partition allokiert werden.</p>	
<i>ZIIP=</i>	'Y' 'N'	
	<p>Mit dieser Option ist festzulegen, ob <i>zZip</i> unter <i>z/OS</i> geeignete, CPU-intensive Verarbeitungsschritte auf <i>zIIP</i> (IBM System z Integrated Information Processor) auslagern soll.</p> <p>Damit <i>zIIP</i> genutzt werden kann, sind systemseitig die entsprechenden Vorkehrungen zu treffen. <i>zIIPs</i> werden bei der Berechnung der Maschinenleistung (MSU-Rate) nicht berücksichtigt, was zur Reduzierung von Lizenzkosten führt.</p> <p>Falls <i>ZIIP</i> verwendet werden soll, ist es erforderlich, dass die Bibliothek, von der <i>zZip</i> geladen wird, APF-autorisiert ist.</p>	
<i>ZIIPPERC=</i>	numerischer Wert <= 100	
	<p>Mit dieser Option ist festzulegen, bis zu welchem prozentualen Anteil <i>zZip</i> die maximal mögliche Arbeitslast von regulären Prozessoren auf <i>zIIP</i> auslagern soll (nur <i>z/OS</i>).</p>	
	Beispiel	<i>ZIIPPERC=100</i>

Variablen

Die Initialisierungsdatei kann, neben der Vergabe von Vorbelegungen für Systemeinstellungen, für die Definition globaler Variablen für *zZip* verwendet werden.

Die Verwendung von Variablen bietet zum Einen die Möglichkeit, Informationen, die sich eventuell mitunter ändern oder deren explizite Angabe in den *zZip* Jobs zu aufwändig wäre, durch einen symbolischen Namen anzusprechen.

Zum Anderen kann durch die Verwendung entsprechender Variablen darauf verzichtet werden, sensible Informationen wie z. B. Passworte für FTP-Zugänge oder symmetrische Schlüssel direkt im Job-Control anzugeben.

Der Abschnittsbeginn für die Definition von Variablen ist durch einen entsprechenden Eintrag ('+VAR+') in der Initialisierungsdatei, wie nachfolgend abgebildet, anzuzeigen:

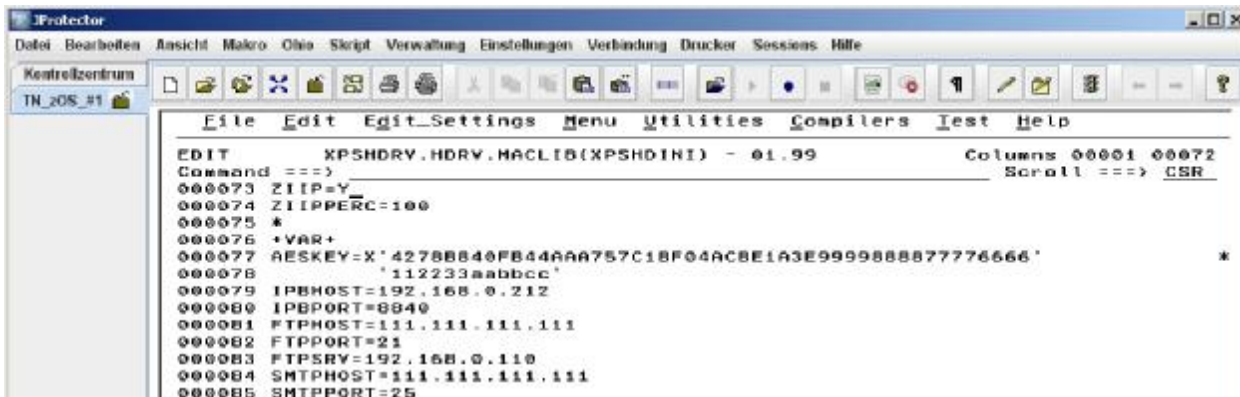


Abb. 12: Variablenkennzeichen in XPSHDINI

Im Abschnitt, der mit '+VAR+' überschrieben ist, können beliebig viele Variablen definiert werden, die jeweils mit einem eindeutigen Namen beginnen müssen und denen, unter Verwendung des Gleichheitszeichens '=', ein Wert zuzuweisen ist. Der links vom Gleichheitszeichen stehende Bezeichner darf maximal 16 Zeichen lang sein.

Im Rahmen der Job-Control kann dann auf vordefinierte Variablen Bezug genommen werden, indem an den entsprechenden Stellen in den zZip Kontrollkarten der Variablenname mit führendem 'kaufmännischen Und' ('&') eingefügt wird, wie nachfolgend gezeigt:

```
...
O ARCHIVE=NET='CEETEMP/xpsvp0.zip' CP=1141-E2N CRLF
O IPADDR=&IPBHOST PORT=&IPBPORT ROUTE=WS08
...
```

Abb. 13: Variablen in Jobkarten

Zur Laufzeit des Jobs ersetzt zZip eingefügte Variablen direkt mit dem Text, der in der Vorbelegung vergeben wurde.

Systemvariablen

Neben der Möglichkeit zur Definition installationsspezifischer Variablen bietet zZip auch Standardvariablen an, die in den Kontrollkarten verwendet werden können.

Variable - allgemein	Bedeutung
&SYSUID.	Platzhalter für den Namen des ausführenden Systems.
&JOBNAME.	Platzhalter für den Namen des aktiven Jobs.
&STEPNAME.	Platzhalter für den Namen des aktiven Job-Steps.
&STEPPROCNAME.	Platzhalter für den Namen der aktiven Job-Step-Prozedur.

<i>&LPAR.</i>	Platzhalter für den Namen des aktiven LPARs – nur z/OS.
<i>&VM.</i>	Platzhalter für den Namen der aktiven VM.
Variable – Datum/Zeit	Bedeutung
<i>&YYYYMMDD.</i>	Aktuelles Datum: 4-stelliges Jahr, 2-stelliger Monat, 2-stelliger Tag
<i>&YYMMDD.</i>	Aktuelles Datum: 2-stelliges Jahr, 2-stelliger Monat, 2-stelliger Tag
<i>&YYYYDDD.</i>	Aktuelles Datum: 4-stelliges Jahr, 3-stelliger laufender Tag im Jahr
<i>&YYDDD.</i>	Aktuelles Datum: 2-stelliges Jahr, 3-stelliger laufender Tag im Jahr
<i>&YYMM.</i>	Aktuelles Datum: 2-stelliges Jahr, 2-stelliger Monat
<i>&YYYY.</i>	Aktuelles Datum: 4-stelliges Jahr
<i>&YY.</i>	Aktuelles Datum: 2-stelliges Jahr
<i>&MM.</i>	Aktuelles Datum: 2-stelliger Monat
<i>&DDD.</i>	Aktuelles Datum: 3-stelliger laufender Tag im Jahr
<i>&HHMMSS.</i>	Aktuelle Zeit: 2-stellige Stunden, 2-stellige Minuten, 2-stellige Sekunden
<i>&HHMN.</i>	Aktuelle Zeit: 2-stellige Stunden, 2-stellige Minuten
<i>&HH.</i>	Aktuelle Zeit: 2-stellige Stunden
<i>&MN.</i>	Aktuelle Zeit: 2-stellige Minuten
<i>&SS.</i>	Aktuelle Zeit: 2-stellige Sekunden
Weitere	Bedeutung
<i>&&FILE</i>	Mit dieser Variablen kann auf Dateinamen Bezug genommen werden, die z. B. in einem komprimierten Archiv gespeichert sind.

Alle zZip Funktionen sind im Programm 'XPSHDFT' bzw. in dessen Alias 'XPSZZIP' implementiert.

zZip liest und analysiert die bereitgestellten Kontrollkarten in der Job-Control und führt entsprechend den gegebenen Anweisungen verschiedene Aktionen und Datentransfers aus.

Hinweis: Die in diesem Kapitel gelisteten Beispieljobs dienen in erster Linie zur Demonstration der Verwendung der verschiedenen Steuerkarten und deren Optionen. In manchen Fällen werden daher eventuell Jobstreams gezeigt, die Steps oder Abläufe enthalten, die in einer produktiven Umgebung so nicht unbedingt erforderlich bzw. sinnvoll wären.

Kontrollkarten

Die Identifizierung der unterschiedlichen Typen von Kontrollkarten erfolgt über ein Typkennzeichen, das an der ersten Position der Kontrollkarte stehen muss. Die nachfolgende Tabelle enthält die möglichen Typkennzeichen sowie eine kurze Beschreibung:

Typ	Beschreibung
'A'	Aktion Mit Karten dieses Typs wird zZip darüber informiert, dass eine Aktion wie z. B. Komprimierung oder Entschlüsselung auf den bereitgestellten Daten ausgeführt werden soll.
'F'	Filter Mit Karten dieses Typs können Include- bzw. Exclude-Filter definiert werden, die es ermöglichen, nur eine Auswahl an Dateien aus den bereitgestellten Datenquellen wie z. B. einer Bibliothek oder einem ZIP-Archiv zu verarbeiten.
'I'	Input Mit Karten dieses Typs werden Ressourcen definiert, die im Rahmen der Verarbeitung als Eingabedaten verwendet werden sollen. Mögliche Eingabetypen sind u. a. ZIP-Archive, VSAM KSDS-Dateien und FTP Quellen, von denen Daten zu lesen sind.
'O'	Output Mit Karten dieses Typs werden Ressourcen definiert, die im Rahmen der Verarbeitung als Datenempfänger verwendet werden sollen. Mögliche Ausgabetyper sind z. B. Netzwerkdateien und E-Mail Server, über die anfallende Dateien als Anhang per E-Mail zu versenden sind.
'R'	Record Mit Karten dieses Typs können bestimmte Felder aus einer Satzstruktur zur weiteren Verarbeitung

	ausgewählt werden. Damit kann z. B. der Datenumfang beim Export einer VSAM-KSDS-Tabelle auf einen benötigten Mindestumfang reduziert werden.
'X'	<p>Execute</p> <p>Mit Karten dieses Typs werden ausführbare Aktionen definiert, die nicht direkt auf einem bereitgestellten Datenbestand operieren. Beispiele sind Anweisungen zur Abarbeitung eventuell vorhandener, nicht zugestellter Datenbestände (waiting for delivery) und Anweisungen zur Ausführung von Programmen auf Servern, die über das Netzwerk erreichbar sind.</p>

Wildcards und Filter

Die Ausführung verschiedener Aktionen durch zZip ermöglicht die Angabe von Dateinamen für die Verarbeitung.

Beispielsweise ist dies der Fall, wenn Dateien verarbeitet werden, die sich in einer Bibliothek oder in einem Zip-Archiv befinden.

In vielen Fällen ist es jedoch ausreichend oder sogar erforderlich, dass nur eine gewisse Auswahl der verfügbaren Dateien auch tatsächlich verarbeitet wird.

Um diese Vorgehensweise zu unterstützen, ermöglicht zZip zum Einen die Verwendung von Platzhaltern zur Bildung von Wildcards bei den Angaben für Dateinamen und zum Anderen die Definition von Filtern zur weiteren Spezifikation der zu verarbeitenden Dateien.

Die nachfolgende Abbildung zeigt ein Beispiel für den Einsatz von Wildcards und Exclude-Filtern.

```

...
//XPSIN DD DISP=SHR,DSN=XPSDAEM.V500.MACLIB
//SYSIN DD *
A ACTION=ZIP PATH=N
I LIB=XPSIN MBR=XPSVP0% EXT=TXT
F EXCLUDE=XPSVP08
F EXCLUDE=XPSVP09
O ARCHIVE=NET='CEETEMP/zipout2.zip' CP=1141-E2N CRLF
...

```

Abb. 14: Beispiel für Wildcard und Exclude-Filter

Mit der 'I LIB=' Karte wird als Eingabedatenbestand der Inhalt einer z/OS Maclib definiert, die durch das DD-Statement 'XPSIN' identifiziert wird.

Die 'MBR=' Option dient im Weiteren dazu, den Umfang der betroffenen Dateien durch die Verwendung der Wildcard 'XPSVP0%' auf Dateinamenebene einzuschränken.

Mit den anschließenden Exclude-Filtern, die durch das Kennzeichen 'F' zu identifizieren sind, werden aus dem zuvor eingeschränkten Dateiumfang weitere Dateien durch Angabe des Dateinamens explizit ausgeschlossen.

Die nachfolgende Abbildung zeigt ein Beispiel für die Verwendung von Include-Filtern:

```

...
//XPSIN DD DISP=SHR,DSN=XPSDAEM.V500.MACLIB
//SYSIN DD *
A ACTION=ZIP PATH=N
I LIB=XPSIN EXT=TXT
F INCLUDE=XPSVP33

```

```
F INCLUDE=XPSVP66
F INCLUDE=XPSVP99
O ARCHIVE=NET='CEETEMP/zipout2.zip' CP=1141-E2N CRLF
...
```

Abb. 15: Beispiel für Include-Filter

In diesem Fall enthält die 'I LIB=' Karte keine 'MBR=' Option, was zur Folge hat, dass die zu verarbeitenden Dateien explizit durch die Verwendung von Include-Filtern zu definieren sind.

Die nachfolgende Tabelle enthält eine Beschreibung der zulässigen Zeichen für die für die Bildung von Wildcards, die von zZip erkannt werden.

Bei der Verwendung von Wildcards bzw. Dateinamen ist immer zu berücksichtigen, dass systembedingt eventuell gewisse Einschränkungen bzgl. des Dateinamens existieren, wie z. B. die maximale Länge von 8 Zeichen für z/OS Maclib- und z/VSE Library Member.

zZip unterstützt die Verwendung mehrerer Platzhalter sowie deren Kombination innerhalb einer Wildcard.

Platzhalter	Beschreibung	
'%'	Mit diesem Platzhalter werden Wildcards definiert, bei denen ein bestimmtes Zeichen des Dateinamens variabel sein soll.	
	Beispiel	Erklärung
	XPSVP0%	Diese Wildcard schließt alle Dateien ein, deren Name mit 'XPSVP0' beginnt und eine Länge von genau 7 Zeichen hat.
'**'	Mit diesem Platzhalter werden Wildcards definiert, bei denen eine zusammenhängende Folge von Zeichen des Dateinamens variabel sein soll.	
	Beispiel	Erklärung
	XPSVP*	Diese Wildcard schließt alle Dateien ein, deren Name mit 'XPSVP' beginnt. Die Anzahl der folgenden Zeichen kann beliebig, insbesondere auch null, sein.
	SAP*X%.*	Diese Wildcard schließt alle Dateien ein, deren Name mit 'SAP' beginnt. Daran kann sich eine beliebige Anzahl von Zeichen anschließen, auf die ein 'X' gefolgt von genau einem beliebigen Zeichen, einem '.' und einer beliebigen Endung folgen muss.

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Eingabekarten des Typs 'F'.

F	
Mit dieser Steuerkarte können Filter definiert werden, um bestimmte Dateien aus einer Auswahl dem zu verarbeitenden Datenbestand hinzuzufügen bzw. um diese davon auszuschließen.	
Optionen	Beschreibung
EXCLUDE	Dateiname
	Mit einem Exclude-Filter können Dateien durch Angabe ihres Namens explizit von der nachfolgenden Verarbeitung ausgeschlossen werden. Pro Exclude-Karte ist genau eine Angabe zu machen, die Wildcards enthalten kann.

	Beispiel	F EXCLUDE XPSVPC#
<i>INCLUDE</i>	Dateiname	
	Mit einem Include-Filter können Dateien durch Angabe ihres Namens explizit in die nachfolgende Verarbeitung aufgenommen werden. Pro Include-Karte ist genau eine Angabe zu machen, die Wildcards enthalten kann.	
	Beispiel	F INCLUDE SAPPI*

Aktionen

zZip kann zur Ausführung verschiedener Operationen wie Komprimierung und Entkomprimierung auf Daten eingesetzt werden.

Die verfügbaren Aktionen werden nachfolgend beschrieben.

GZIP/GUNZIP – Komprimieren bzw. Entkomprimieren einzelner Dateien

Mit diesen Aktionen können Daten gemäß dem GZIP Standard komprimiert bzw. entkomprimiert werden.

Das GZIP Verfahren ist speziell für die Verarbeitung einzelner Dateien geeignet.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JGZIPS2S JOB , 'GZIP TEST', CLASS=A, MSGCLASS=M
//STEP1 EXEC PGM=XPSZZIP, REGION=32M
//STEPLIB DD DSN=SERVICE.HDRV300D.LOADLIB, DISP=SHR
//XPSSEQ02 DD DSN=XPSHDRV.GZIPTST.SEQ02, DISP=(NEW,CATLG),
// SPACE=(TRK,(50,30),RLSE), UNIT=3390,
// DCB=(BLKSIZE=27998,LRECL=27998,RECFM=U,DSORG=PS),
// STORCLAS=DBCLASS
//SYSIN DD *
A ACTION=GZIP ATTR=Y PATH=Y MODE=DETECT COMPLVL=MAXIMUM
I SEQ=XPSHDRV.GZIPTST.SEQ01
O SEQ=XPSSEQ02 CP=1141-E2N
//*
//STEP2 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE XPSHDRV.GZIPTST.SEQ01
//*
//STEP3 EXEC PGM=XPSZZIP, REGION=32M
//STEPLIB DD DSN=SERVICE.HDRV300D.LOADLIB, DISP=SHR
//SYSIN DD *
A ACTION=GUNZIP
I ARCHIVE=SEQ=XPSHDRV.GZIPTST.SEQ02
/* OUT=XPSHDRV.GZIPTST.SEQ01 WIRD AUS GZIP FILE ATTRS HERGESTELLT
/*
```

Abb. 16: Beispieljob für die Aktionen GZIP und GUNZIP z/OS

Im Beispiel wird der Inhalt einer sequentiellen Datei zunächst unter Verwendung der Aktion 'ACTION=GZIP' mit dem GZIP-Verfahren in eine weitere sequentielle Datei komprimiert.

Nach dem Löschen der ursprünglichen sequentiellen Datei im 2. Step wird die zuvor erstellte komprimierte Datei unter Verwendung der Aktion 'ACTION=GUNZIP' entkomprimiert und die ursprüngliche sequentielle Datei wird anhand der in der GZIP-Datei gespeicherten Attribute wieder hergestellt.

Im Rahmen der Entkomprimierung kommt darüber hinaus implizit die in den Systemeinstellungen definierte Codepage zur Übersetzung von Textdaten von ASCII/ANSI nach EBCDIC zum Einsatz (Option 'CPA2E'), da der Umstand, dass Textdaten entkomprimiert werden, durch ein Attribut in der GZIP-Datei kenntlich gemacht wird.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE mit vergleichbarem Ablauf.

```

* $$ JOB JNM=JGZIPS2S,DISP=D,PRI=3,CLASS=0
* $$ LST CLASS=T,JNM=JGZIPS2S
// JOB JGZIPS2S
// LIBDEF PHASE,SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005,SYSLST
// DLBL XPSHFS,'XPSHDRV.SPOOL.SAVE',,VSAM,CAT=XPSUCAT
// EXEC XPSZZIP,SIZE=AUTO,PARM='LIB=XPS.DAEMH600'
A ACTION=GZIP ATTR=Y PATH=Y MODE=DETECT COMPRLVL=MAXIMUM
I SEQ=XPSHDRV.GZIPTST.SEQ01
O SEQ=XPSSEQ02 CP=1141-E2N
/*
// EXEC IDCAMS
DELETE (XPSHDRV.GZIPTST.SEQ01) CL PURGE -
    CATALOG(XPSUCAT)
/*
// LIBDEF PHASE,SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005,SYSLST
// DLBL XPSHFS,'XPSHDRV.SPOOL.SAVE',,VSAM,CAT=XPSUCAT
// EXEC XPSZZIP,SIZE=AUTO,PARM='LIB=XPS.DAEMH600'
A ACTION=GUNZIP
I ARCHIVE=SEQ=XPSHDRV.GZIPTST.SEQ02
/* OUT=XPSHDRV.GZIPTST.SEQ01 WIRD AUS GZIP FILE ATTRS HERGESTELLT
/*
/&
* $$EOJ
    
```

Abb. 17: Beispieljob für die Aktionen GZIP und GUNZIP z/VSE

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Karten des Typs 'A ACTION=GZIP'.

A ACTION=GZIP											
Komprimieren von Eingabedaten gemäß dem GZIP-Verfahren.											
Optionen	Beschreibung										
ARC_BLKSIZE=	numerischer Wert 'DYNAMIC' 'OPTIMUM' 'MAXIMUM'										
	<p>Mit dieser Option kann die gewünschte Blockgröße festgelegt werden, die zZip zur Erstellung von sequentiellen Dateien bzw. von PO-Datasets verwenden soll, falls diese als Ausgabebziel für einen Komprimiervorgang verwendet werden.</p> <p>Die Angabe eines numerischen Wertes wird in Bytes erwartet.</p> <p>Die nachfolgende Tabelle beschreibt die Vorgehensweise bei der Auswahl eines der vordefinierten, konstanten Werte:</p> <table border="1"> <thead> <tr> <th>Option</th> <th>Bedeutung</th> </tr> </thead> <tbody> <tr> <td>DYNAMIC</td> <td>zZip ermittelt die zu verwendende Blockgröße dynamisch.</td> </tr> <tr> <td>OPTIMUM</td> <td>zZip ermittelt die zu verwendende Blockgröße dynamisch und optimiert diese für das Zielgerät.</td> </tr> <tr> <td>MAXIMUM</td> <td>zZip ermittelt die zu verwendende Blockgröße dynamisch und wählt die größtmögliche Blockgröße für das Zielgerät.</td> </tr> <tr> <td>Beispiele:</td> <td>ARC_BLKSIZE=4096 ARC_BLKSIZE=OPTIMUM</td> </tr> </tbody> </table>	Option	Bedeutung	DYNAMIC	zZip ermittelt die zu verwendende Blockgröße dynamisch.	OPTIMUM	zZip ermittelt die zu verwendende Blockgröße dynamisch und optimiert diese für das Zielgerät.	MAXIMUM	zZip ermittelt die zu verwendende Blockgröße dynamisch und wählt die größtmögliche Blockgröße für das Zielgerät.	Beispiele:	ARC_BLKSIZE=4096 ARC_BLKSIZE=OPTIMUM
Option	Bedeutung										
DYNAMIC	zZip ermittelt die zu verwendende Blockgröße dynamisch.										
OPTIMUM	zZip ermittelt die zu verwendende Blockgröße dynamisch und optimiert diese für das Zielgerät.										
MAXIMUM	zZip ermittelt die zu verwendende Blockgröße dynamisch und wählt die größtmögliche Blockgröße für das Zielgerät.										
Beispiele:	ARC_BLKSIZE=4096 ARC_BLKSIZE=OPTIMUM										

ARC_DATACLASS=	maximal 8-stelliger Name einer SMS Data Class	
	Mit dieser Option kann die SMS Data Class festgelegt werden, die für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Komprimiervorgangs verwendet wird.	
	Beispiel	ARC_DATACLASS=DC2
ARC_DIR_BLKs=	numerischer Wert	
	Mit dieser Option kann die Anzahl der Verzeichnisblöcke angegeben werden, die zZip für eine Bibliothek reservieren soll, die als Ausgabeziel für einen Komprimiervorgang verwendet wird.	
	Beispiel	ARC_DIR_BLKs=56
ARC_DSORG=	'SEQ' 'PS' 'PDS' 'PO'	
	Mit dieser Option ist zZip über den Organisationstyp einer neu anzulegenden Datei zu informieren, die als Ausgabeziel für einen Komprimiervorgang verwendet wird.	
	<i>SEQ</i> <i>PS</i>	Bei Wahl einer dieser Optionen erstellt zZip eine neue Datei mit Organisationstyp 'sequentiell' (sequential / partitioned sequential).
	<i>PDS</i> <i>PO</i>	Bei Wahl einer dieser Optionen erstellt zZip eine neue Bibliothek (partitioned data set / partitioned organized).
ARC_MGMTCLASS=	maximal 8-stelliger Name einer SMS Management Class	
	Mit dieser Option kann die SMS Management Class festgelegt werden, die für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Komprimiervorgangs verwendet wird.	
	Beispiel	ARC_MGMTCLASS=MAG30LIB
ARC_LRECL=	numerischer Wert	
	Mit dieser numerischen Option kann die logische Satzlänge festgelegt werden, die für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Komprimiervorgangs verwendet wird.	
	Beispiel	ARC_LREC=132
ARC_RECfm=	'F' 'FB' 'V' 'VB' 'U'	
	Mit dieser Option kann das Satzformat festgelegt werden, das für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Komprimiervorgangs verwendet wird.	
	<i>F</i>	Die Zieldatei verwendet Sätze mit fester Satzlänge.
	<i>FB</i>	Die Zieldatei verwendet Sätze mit fester Satzlänge, die geblockt verarbeitet werden.
	<i>V</i>	Die Zieldatei verwendet Sätze mit variabler Satzlänge.
	<i>VB</i>	Die Zieldatei verwendet Sätze mit variabler Satzlänge, die geblockt verarbeitet werden.

	<i>U</i>	Die Zieldatei verwendet Sätze mit undefinierter Satzlänge.
ARC_SPACE_PRI=	numerischer Wert	
	Mit dieser Option kann die Anzahl der Allokationseinheiten im Primary Extent für eine neu anzulegende sequentielle Datei bzw. für ein PO-Dataset festgelegt werden, die als Ausgabeziel für einen Komprimiervorgang verwendet werden.	
	Beispiel	ARC_SPACE_PRI=25
ARC_SPACE_RLSE=	'Y' 'N'	
	Mit dieser Option ist festzulegen, ob zZip nicht benötigten und zuvor reservierten Plattenplatz für eine neu erstellte sequentielle Datei bzw. für ein neu erstelltes PO-Dataset freigeben soll, nachdem der Komprimiervorgang beendet wurde.	
	Beispiel	ARC_SPACE_RLSE=Y
ARC_SPACE_SEC=	numerischer Wert	
	Mit dieser Option kann die Anzahl der Allokationseinheiten im Secondary Extent für eine neu anzulegende sequentielle Datei bzw. für ein PO-Dataset festgelegt werden, die als Ausgabeziel für einen Komprimiervorgang verwendet werden.	
	Beispiel	ARC_SPACE_SEC=10
ARC_SPACE_TYPE=	'CYL' 'TRK' 'BLK'	
	Mit dieser Option ist der Typ der Allokationseinheiten für eine neu anzulegende sequentielle Datei bzw. für ein PO-Dataset festzulegen, die als Ausgabeziel für einen Komprimiervorgang verwendet werden. Numerische Angaben für die Optionen 'ARC_SPACE_PRI' und 'ARC_SPACE_SEC' beziehen sich auf den hier festgelegten Typ.	
	Beispiel	ARC_SPACE_TYPE=CYL
ARC_STORCLASS=	maximal 8-stelliger Name einer SMS Storage Class	
	Mit dieser Option kann die SMS Storage Class festgelegt werden, die für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Komprimiervorgangs verwendet wird.	
	Beispiel	ARC_STORCLASS=SMSSTOR
ARC_UNIT=	maximal 8-stelliger Name einer Unit	
	Mit dieser Option ist der Typ der Einheit anzugeben, auf der zZip eine neu anzulegende sequentielle Datei bzw. ein PO-Dataset allokiert werden soll, die als Ausgabeziel für einen Komprimiervorgang verwendet werden.	
	Beispiel	ARC_UNIT=SYSDA
ARC_VOLUME=	maximal 8-stelliger Name eines Volumes	
	Mit dieser Option ist der Name des Volumes anzugeben, auf der zZip eine neu anzulegende sequentielle Datei bzw. ein PO-Dataset allokiert werden soll, die als Ausgabeziel für einen Komprimiervorgang verwendet werden.	

	Beispiel	ARC_VOLUME=XPS011												
ATTR=	'Y' 'N'													
	<p>Mit dieser Option ist festzulegen, wie im Rahmen der Komprimierung von Dateien mit Dateiattributen verfahren werden soll.</p> <p>Bei Auswahl von 'Y' werden verfügbare Attribute in die komprimierte Datei inkludiert und können bei Bedarf von Programmen ausgelesen werden. Es ist zu beachten, dass die Verarbeitung von Dateiattributen in GZIP-Dateien von manchen Programmen nicht unterstützt wird.</p>													
COMPLVL=	numerischer Wert zwischen '0' ... '9' 'STORE' 'SUPERFAST' 'FAST' 'NORMAL' 'MAXIMUM'													
	<p>Mit dieser Option ist der Komprimierungslevel festzulegen. Mit steigendem Wert verringert sich die Größe des erstellten Archivs auf Kosten der benötigten CPU Zeit. Neben der Angabe eines numerischen Wertes kann auch eine der folgenden vordefinierten Komprimierungsstufen gewählt werden:</p> <table border="1" data-bbox="399 817 1386 1146"> <thead> <tr> <th>vordefinierte Komprimierungsstufe</th> <th>entspricht numerischem Wert</th> </tr> </thead> <tbody> <tr> <td>STORE</td> <td>0</td> </tr> <tr> <td>SUPERFAST</td> <td>1</td> </tr> <tr> <td>FAST</td> <td>2</td> </tr> <tr> <td>NORMAL</td> <td>3</td> </tr> <tr> <td>MAXIMUM</td> <td>4</td> </tr> </tbody> </table>		vordefinierte Komprimierungsstufe	entspricht numerischem Wert	STORE	0	SUPERFAST	1	FAST	2	NORMAL	3	MAXIMUM	4
vordefinierte Komprimierungsstufe	entspricht numerischem Wert													
STORE	0													
SUPERFAST	1													
FAST	2													
NORMAL	3													
MAXIMUM	4													
CONCAT=	'Y' 'N'													
	<p>Das GZIP Format ist in erster Linie zur Komprimierung einzelner Dateien gedacht. Das bedeutet, dass z. B. bei Angabe einer Bibliothek als Eingabe nur die erste Datei aus der Bibliothek komprimiert wird.</p> <p>Es besteht dennoch die Möglichkeit, mehrere Eingabedateien in eine einzige Zielformatdatei zu komprimieren.</p> <p>Damit dies geschieht, ist die Option 'CONCAT=Y' zu setzen.</p> <p>Es ist allerdings zu bedenken, dass die einzelnen Eingabedateien in der Zielformatdatei nicht mehr unterschieden werden können.</p> <p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 23 beschrieben, gesetzt werden.</p>													
FILE_TERM=	hexadezimaler Wert (X'...') vordefinierter Wert													
	<p>Unter Umständen kann es erforderlich oder gewünscht sein, dass das Ende einer Textdatei durch ein bestimmtes Zeichen bzw. durch eine bestimmte Zeichenkombination explizit kenntlich gemacht werden soll.</p> <p>Um zu erreichen, dass zZip ein besonderes Dateiende-Kennzeichen für Textdateien einfügt, ist das gewünschte Dateiende-Kennzeichen mit dieser Option festzulegen.</p> <table border="1" data-bbox="399 1921 1386 2033"> <tr> <td>Beispiele</td> <td>FILE_TERM=X'0D0A0A0D'</td> </tr> <tr> <td></td> <td>FILE_TERM=A-CRLF</td> </tr> </table> <p>Die nachfolgende Tabelle zeigt eine Auflistung der vordefinierten Werte, die als</p>		Beispiele	FILE_TERM=X'0D0A0A0D'		FILE_TERM=A-CRLF								
Beispiele	FILE_TERM=X'0D0A0A0D'													
	FILE_TERM=A-CRLF													

	Dateiende-Kennzeichen verwendet werden können:	
	Bezeichner	Wert
	<i>A-CRLFCZ</i>	X'0D0A1A'
	<i>CRLFCZ</i>	X'0D0A1A'
	<i>A-CRLF</i>	X'0D0A'
	<i>A-LFCR</i>	X'0A0D'
	<i>E-CRLF</i>	X'0D25'
	<i>E-CRNL</i>	X'0D25'
	<i>E-LFCR</i>	X'250D'
	<i>A-NEL</i>	X'85'
	<i>CRLF</i>	X'0D0A'
	<i>LFCR</i>	X'0A0D'
	<i>A-CR</i>	X'0D'
	<i>A-LF</i>	X'0A'
	<i>A-CZ</i>	X'1A'
	<i>E-CR</i>	X'0D'
	<i>E-LF</i>	X'25'
	<i>E-NL</i>	X'15'
	<i>CR</i>	X'0D'
	<i>LF</i>	X'0A'
	<i>CZ</i>	X'1A'
MODE=	'TEXT' 'BIN' 'DETECT'	
	<p>Mit dieser Option ist festzulegen, auf welche Weise zZip die zu komprimierenden Daten interpretieren soll.</p> <p>Bei Angabe von 'TEXT' interpretiert zZip die Daten als Textdaten und verwendet die jeweils angegebene Codepage zur Übersetzung der Textdaten vor dem Komprimieren.</p> <p>Bei Angabe von 'BIN' werden die Daten als binär interpretiert und es findet keine Übersetzung statt.</p> <p>Falls der Standard 'DETECT' verwendet wird, analysiert zZip die Daten vor dem Komprimieren und ermittelt den Datentyp unter Berücksichtigung der Angaben für die Optionen 'MODE_DETECT_TAB' und 'MODE_DETECT_LEN'.</p>	
	Beispiel	MODE=TEXT
MODE_DETECT_LEN=	numerischer Wert gefolgt von 'K' 'M' 'R'	

	<p>Falls zZip den Typ der verarbeiteten Daten erkennen soll ('MODE=DETECT'), analysiert zZip die Daten an Hand der mit der Option 'MODE_DETECT_TAB=' angegebenen Tabelle.</p> <p>Mit dieser Option kann festgelegt werden, wie viel Zeichen zZip in diesem Fall maximal überprüfen soll, bis die Entscheidung, ob die Datei ausschließlich Textdaten enthält oder nicht, getroffen werden soll.</p> <p>Die Angabe kann entweder in Kilobytes ('K'), in Megabytes ('M') oder – für Textdateien - als Anzahl von Sätzen ('R') erfolgen.</p>	
	Beispiele	<p>MODE_DETECT_LEN=1M MODE_DETECT_LEN=1000R</p>
MODE_DETECT_TAB=	Name eines Lademoduls das eine Testtabelle enthält	
	<p>Bei Wahl von 'MODE=DETECT' ermittelt zZip den Typ der Daten an Hand einer Tabelle, mit der festgestellt wird, ob die Daten Zeichen enthalten, die nicht als Textzeichen gelten.</p> <p>Im Lieferumfang von zZip befindet sich eine Standardtabelle namens 'DETECTXT', die als gleichnamiges Textmodul in der XPSDATA Bibliothek abgelegt ist.</p> <p>Falls diese Tabelle den Anforderungen nicht entsprechen sollte, können Veränderungen entweder direkt an dieser Tabelle durchgeführt werden oder an einer Kopie, die dann unter Verwendung der 'MODE_DETECT_TAB=' Option bekanntgegeben wird. Damit eine Tabelle verwendet kann, muss diese zur Laufzeit als assembliertes Lademodul verfügbar sein.</p>	
	Beispiel	MODE_DETECT_TAB=DETECTXT
NIA=	MVS-Name,Archivname	
	<p>Mit dieser Option kann bestimmt werden, auf welche Weise MVS Dateinamen vor dem Einfügen in das Archiv umbenannt werden sollen. Unter Verwendung des Wildcard Zeichens '*' ist es möglich, eine Umbenennregel für mehrere Dateinamen festzulegen.</p>	
	<i>MVS-Name</i>	<p>Maximal 44-stelliger MVS-Dateiname wie z. B. 'MY.CLIENTS.DATA'.</p> <p>Durch Verwendung des Wildcard Zeichens können mehrere Dateien über eine Schablone angesprochen werden.</p> <p>Mit der Schablone 'MY.CLIENTS.DATA*' würden z. B. die Dateien 'MY.CLIENTS.DATA', 'MY.CLIENTS.DATA1' und 'MY.CLIENTS.DATA2' usw. ausgewählt.</p>
	<i>Archiv-Name</i>	<p>Maximal 80-stelliges Format des zu erstellenden Dateinamens für das Archiv.</p> <p>Das Format enthält den Namen, wie er in der ZIP-Datei erscheinen soll und kann Wildcard Zeichen '*' und Löschkennzeichen '+' enthalten.</p> <p>Für die Verwendung von Wildcard und Löschkennzeichen gilt die folgende Regel:</p> <p>Für jedes Wildcard Zeichen im Archivnamen wird der positionell entsprechende Wildcard Teil des MVS Namens (1., 2., 3., etc.) in den zu erstellenden Archivnamen kopiert.</p> <p>Für jedes Löschkennzeichen im Archivnamen wird der positionell entsprechende Wildcard Teil des MVS Namens (1., 2., 3., etc.)</p>

		entfernt, d. h. nicht zur Bildung des zu erstellenden Archivnamens verwendet.	
	Die nachfolgende Tabelle enthält einige Beispiele zur Verdeutlichung der beschriebenen Regeln:		
	MVS Dateiname	MVS-Name,Archivname	ZIP-Dateiname
	XPS.SEQ.FILE	NIA=XPS.SEQ.FILE,XPS/ZIP	XPS/ZIP
	XPS.SEQ.FILE	NIA=XPS.SEQ.FILE,XPS.TXT	XPS.TXT
	XPS.SEQ.FILE	NIA=*. *.FILE, *+.ZIP	XPS.ZIP
	XPS.SEQ.FILE	NIA=*. *.FILE, *.*	XPS.SEQ
	XPS.SEQ.FILE	NIA=*. *.FILE, +*.INF	SEQ.INF
	XPS.LIB(MEM)	NIA=XPS.LIB(*),ZIP/LIB/*	ZIP/LIB/MEM
	XPS.LIB(MEM)	NIA=*. *(MEM), */*.DAT	XPS/LIB.DAT
	XPS.LIB(MEM)	NIA=*. *(*), */*/*.DAT	XPS/LIB/MEM.DAT
	XPS.LIB(MEM)	NIA=*. LIB(*), */*/INFO	XPS/MEM/INFO
	XPS.LIB(MEM)	NIA=*(*), +*.INF	MEM.INF
	<p>Jeder 'XPSZZIP'-Jobstep kann mehrere 'NIA'-Karten enthalten, wodurch die Definition mehrerer Übersetzungsregeln ermöglicht wird. 'NIA'-Karten werden in der Reihenfolge der Auflistung verarbeitet, so dass die jeweils erste passende 'NIA'-Karte für die Übersetzung eines MVS Dateinamens verwendet wird.</p> <p>Bei der Wahl der Übersetzungsregeln ist darauf zu achten, dass keine ungültigen Dateinamen wie z. B. XPS.LIB(MEM).TXT generiert werden. zZip überprüft generierte Archivnamen nicht auf Gültigkeit.</p> <p>Falls für einen MVS Dateinamen keine passende 'NIA'-Regel für die Übersetzung gefunden wird, wird der Archivname gebildet, indem die Separatorpunkte '.' und die öffnende Klammer '(' jeweils durch das definierte Pfadtrennzeichen - siehe Parameter 'PATHSEP' - ersetzt werden. Schließende Klammern fallen weg.</p>		
PATH=	'Y' 'N'		
	Mit dieser Option ist festzulegen, ob zZip lediglich den Dateinamen ('PATH=N') oder den gesamten Pfad ('PATH=Y') als Attribut in das Archiv einfügen soll.		
PATHSEP=	darstellbares Zeichen		
	<p>Mit dieser Option kann ein Zeichen bestimmt werden, das zZip im Rahmen der Erstellung von Pfadangaben zur Trennung einzelner Pfadbestandteile verwenden soll.</p> <p>Auf Microsoft Windows Plattformen ist dies üblicherweise der Backslash ('\'), wohingegen auf Unix- bzw. Linuxderivaten der Slash ('/') verwendet wird.</p>		
PKCP=	Name einer Codepage		
	Mit dieser Option kann der Name einer assemblierten Codepage in PKWARE kompatiblen Format angegeben werden, die zur Übersetzung von Textdaten verwendet werden soll.		
REC_DELIM	hexadezimaler Wert (X'...') vordefinierter Wert		
	Mit dieser Option kann festgelegt werden, welches Zeichen bzw. welche Zeichenkombination zZip als Zeilenende-Kennzeichen bei der Verarbeitung von Textdaten verwenden soll.		

Beispiele	REC_DELIM=X'0D' REC_DELIM=A-LFCR
Die nachfolgende Tabelle zeigt eine Auflistung der vordefinierten Werte, die als Zeilenende-Kennzeichen verwendet werden können:	
Bezeichner	Wert
<i>A-CRLFCZ</i>	X'0D0A1A'
<i>CRLFCZ</i>	X'0D0A1A'
<i>A-CRLF</i>	X'0D0A'
<i>A-LFCR</i>	X'0A0D'
<i>E-CRLF</i>	X'0D25'
<i>E-CRNL</i>	X'0D25'
<i>E-LFCR</i>	X'250D'
<i>A-NEL</i>	X'85'
<i>CRLF</i>	X'0D0A'
<i>LFCR</i>	X'0A0D'
<i>A-CR</i>	X'0D'
<i>A-LF</i>	X'0A'
<i>A-CZ</i>	X'1A'
<i>E-CR</i>	X'0D'
<i>E-LF</i>	X'25'
<i>E-NL</i>	X'15'
<i>CR</i>	X'0D'
<i>LF</i>	X'0A'
<i>CZ</i>	X'1A'

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Karten des Typs 'A ACTION=GUNZIP'.

A ACTION=GUNZIP																													
<p>Entkomprimieren von Eingabedaten gemäß dem GZIP-Verfahren.</p> <p>Wenn ein Archiv entkomprimiert wird, das Attribute der ursprünglich komprimierten Datei enthält, können die verschiedenen 'OUT_' Optionen dazu verwendet werden, die entsprechenden Attribute der Originaldatei zu überschreiben.</p>																													
Optionen	Beschreibung																												
<i>CNVEXT=</i>	'NAMEFILE' 'KEEP' 'DROP' 'SUFFIX' 'MERGE'																												
	<p>Mit dieser Option ist festzulegen, wie zZip mit der Erweiterung von Dateinamen verfahren soll, die beim Entpacken von komprimierten Archiven anfallen.</p> <p>Die nachfolgende Tabelle zeigt den resultierenden Name nach Anwendung der möglichen Optionen auf den Dateinamen 'COMPANY.STATS.TXT'.</p> <table border="1"> <thead> <tr> <th>Option</th> <th>Resultat</th> </tr> </thead> <tbody> <tr> <td><i>NAMEFILE, KEEP</i></td> <td>COMPANY.STATS.TXT</td> </tr> <tr> <td><i>DROP</i></td> <td>COMPANY.STATS</td> </tr> <tr> <td><i>SUFFIX, MERGE</i></td> <td>COMPANY.STATSTXT</td> </tr> </tbody> </table>	Option	Resultat	<i>NAMEFILE, KEEP</i>	COMPANY.STATS.TXT	<i>DROP</i>	COMPANY.STATS	<i>SUFFIX, MERGE</i>	COMPANY.STATSTXT																				
Option	Resultat																												
<i>NAMEFILE, KEEP</i>	COMPANY.STATS.TXT																												
<i>DROP</i>	COMPANY.STATS																												
<i>SUFFIX, MERGE</i>	COMPANY.STATSTXT																												
<i>FILE_TERM=</i>	hexadezimaler Wert (X'...') vordefinierter Wert																												
	<p>Beim Erstellen komprimierter Archive kann eine besondere Zeichensequenz definiert werden, die als Dateiende-Kennzeichen an jede Daten angefügt werden soll.</p> <p>Falls ein Archiv entkomprimiert wird, für das ein spezielles Dateiende-Kennzeichen definiert wurde, kann die dazu verwendete Zeichensequenz mit dieser Option bekannt gegeben werden.</p> <p>Im Rahmen des Entkomprimierens entfernt zZip diese zuvor beim Komprimieren hinzugefügte Zeichensequenz und transferiert die Daten damit wieder in denselben Zustand, den diese vor dem Komprimieren hatten.</p> <table border="1"> <thead> <tr> <th>Beispiele</th> <th></th> </tr> </thead> <tbody> <tr> <td></td> <td>FILE_TERM=X'0D0AAABB'</td> </tr> <tr> <td></td> <td>FILE_TERM= <i>LFCR</i></td> </tr> </tbody> </table> <p>Die nachfolgende Tabelle zeigt eine Auflistung der vordefinierten Werte, die als Dateiende-Kennzeichen verwendet werden können:</p> <table border="1"> <thead> <tr> <th>Bezeichner</th> <th>Wert</th> </tr> </thead> <tbody> <tr> <td><i>A-CRLFCZ</i></td> <td>X'0D0A1A'</td> </tr> <tr> <td><i>CRLFCZ</i></td> <td>X'0D0A1A'</td> </tr> <tr> <td><i>A-CRLF</i></td> <td>X'0D0A'</td> </tr> <tr> <td><i>A-LFCR</i></td> <td>X'0A0D'</td> </tr> <tr> <td><i>E-CRLF</i></td> <td>X'0D25'</td> </tr> <tr> <td><i>E-CRNL</i></td> <td>X'0D25'</td> </tr> <tr> <td><i>E-LFCR</i></td> <td>X'250D'</td> </tr> <tr> <td><i>A-NEL</i></td> <td>X'85'</td> </tr> <tr> <td><i>CRLF</i></td> <td>X'0D0A'</td> </tr> <tr> <td><i>LFCR</i></td> <td>X'0A0D'</td> </tr> </tbody> </table>	Beispiele			FILE_TERM=X'0D0AAABB'		FILE_TERM= <i>LFCR</i>	Bezeichner	Wert	<i>A-CRLFCZ</i>	X'0D0A1A'	<i>CRLFCZ</i>	X'0D0A1A'	<i>A-CRLF</i>	X'0D0A'	<i>A-LFCR</i>	X'0A0D'	<i>E-CRLF</i>	X'0D25'	<i>E-CRNL</i>	X'0D25'	<i>E-LFCR</i>	X'250D'	<i>A-NEL</i>	X'85'	<i>CRLF</i>	X'0D0A'	<i>LFCR</i>	X'0A0D'
Beispiele																													
	FILE_TERM=X'0D0AAABB'																												
	FILE_TERM= <i>LFCR</i>																												
Bezeichner	Wert																												
<i>A-CRLFCZ</i>	X'0D0A1A'																												
<i>CRLFCZ</i>	X'0D0A1A'																												
<i>A-CRLF</i>	X'0D0A'																												
<i>A-LFCR</i>	X'0A0D'																												
<i>E-CRLF</i>	X'0D25'																												
<i>E-CRNL</i>	X'0D25'																												
<i>E-LFCR</i>	X'250D'																												
<i>A-NEL</i>	X'85'																												
<i>CRLF</i>	X'0D0A'																												
<i>LFCR</i>	X'0A0D'																												

	<i>A-CR</i>	X'0D'
	<i>A-LF</i>	X'0A'
	<i>A-CZ</i>	X'1A'
	<i>E-CR</i>	X'0D'
	<i>E-LF</i>	X'25'
	<i>E-NL</i>	X'15'
	<i>CR</i>	X'0D'
	<i>LF</i>	X'0A'
	<i>CZ</i>	X'1A'
<i>MODE=</i>	'TEXT' 'BIN'	
	<p>Das Format (Text bzw. binär) komprimierter Daten, wird normalerweise im Archiv als Attribut gespeichert.</p> <p>Falls die dort gespeicherte Information überschrieben werden soll, kann dies mit dieser Option erfolgen.</p>	
<i>NOA=</i>	[Archivpfad],[MVS-HLQ]	
	<p>Mit dieser Option kann ein Schema definiert werden, das zur Veränderung des High Level Qualifiers für einen Dateinamen im Rahmen der Entkomprimierung verwendet werden soll.</p> <p>Das Schema findet Anwendung bei der Transformation vom Archivnamen zum MVS Namen. Dabei werden Archivnamen, die dem spezifizierten Archivpfad entsprechen durch den angegebenen MVS-HLQ ersetzt. Die Verwendung von Wildcard Zeichen '*' zur Anwendung auf mehrere Archivpfade ist möglich.</p> <p>Das Weglassen entweder des Archivpfades oder des MVS-HLQ ist zulässig.</p> <p>Jeder 'XPSZZIP'-Jobstep kann mehrere 'NOA'-Karten enthalten, wodurch die Definition mehrerer Ersetzungsregeln ermöglicht wird. 'NOA'-Karten werden in der Reihenfolge der Auflistung verarbeitet, so dass die jeweils erste passende 'NOA'-Karte für die Ersetzung eines Archivnamens verwendet wird.</p>	
	<i>Archivpfad</i>	<p>An dieser Stelle ist der High Level Qualifier anzugeben, der mit den im Zip Archiv gespeicherten Namen auf Übereinstimmung hin zu überprüfen ist, und der im Falle der Übereinstimmung durch den angegebenen MVS-HLQ ersetzt werden soll.</p> <p>Die Angabe darf nicht länger als 80 Zeichen sein.</p> <p>Das Wildcard Zeichen '*' kann dazu verwendet werden, Bezug auf genau einen Level des Archivnamens zu nehmen.</p> <p>Das Wildcard Zeichen '?' kann dazu verwendet werden, Bezug auf genau ein Zeichen des Archivnamens – mit Ausnahme der Separatorpunkte '.' - zu nehmen.</p> <p>Durch die Verwendung des doppelten Wildcard Zeichens '**' kann der komplette Archivpfad ersetzt werden.</p>
	<i>MVS-HLQ</i>	<p>An dieser Stelle ist anzugeben, mit welchem HLQ die durch den ersten Operand spezifizierten Teile des Archivnamens ersetzt und dem restlichen Teil des Archivnamens vorangestellt werden sollen.</p> <p>Die Angabe darf nicht länger als 54 Zeichen sein und muss den MVS</p>

	Namenskonventionen entsprechen.	
	Die nachfolgende Tabelle enthält einige Beispiele zur Verdeutlichung der beschriebenen Regeln:	
Archivname	[Archivpfad],[MVS-HLQ]	MVS Dateiname
XPS/MEMBER/INFO	NOA=XPS.M,NOV. { ersetze 'XPS.M' durch 'NOV.' }	NOV.EMBER.INFO
XPS/MEMBER/INFO	NOA=*,NOV { ersetze den ersten Level durch 'NOV' }	NOV.MEMBER.INFO
XPS/MEMBER/INFO	NOA=XPS.?EMBER,NOV { ersetze 'XPS.?EMBER' durch 'NOV' }	NOV.INFO
XPS/MEMBER/INFO	NOA=XPS.?EMBER.,NOV { ersetze 'XPS.?EMBER.' durch 'NOV' }	NOVINFO
XPS/MEMBER/INFO	NOA=,NEW. { stelle 'NEW.' voran }	NEW.XPS.MEMBER.INFO
XPS/MEMBER/INFO	NOA=,NEW { stelle 'NEW' voran }	NEWXPS.MEMBER.INFO
XPS/MEMBER/INFO	NOA=XP, { entferne ein führendes 'XP' }	S.MEMBER.INFO
XPS/MEMBER/INFO	NOA=**,NEW.FILE.TYPE { ersetze den kompletten Namen }	NEW.FILE.TYPE
	Bevor die jeweils erste passende 'NOA'-Regel auf einen Archivnamen angewendet wird, wird zunächst die Einstellung für die Option 'CNVEXT=' auf den Archivnamen angewendet.	
Archivname	CNVEXT [Archivpfad],[MVS-HLQ]	MVS Dateiname
XPS/MEMBER/README.TXT	CNVEXT=DROP NOA=XPS.MEMBER.,MYHLQ.	MYHLQ.README
XPS/MEMBER/README.TXT	CNVEXT=NAMEFILE NOA=XPS.MEMBER.,MYHLQ.	MYHLQ.README.TXT
OUT_BLKSIZE=	numerischer Wert 'DYNAMIC' 'OPTIMUM' 'MAXIMUM'	
	<p>Mit dieser Option kann die gewünschte Blockgröße festgelegt werden, die zZip zur Erstellung von sequentiellen Dateien bzw. von PO-Datasets verwenden soll, falls diese als Ausgabeziel für einen Entkomprimiervorgang verwendet werden.</p> <p>Die Angabe eines numerischen Wertes wird in Bytes erwartet.</p> <p>Die nachfolgende Tabelle beschreibt die Vorgehensweise bei der Auswahl eines der vordefinierten, konstanten Werte:</p>	
	Option	Bedeutung
	<i>DYNAMIC</i>	zZip ermittelt die zu verwendende Blockgröße dynamisch.
	<i>OPTIMUM</i>	zZip ermittelt die zu verwendende Blockgröße dynamisch und optimiert diese für das Zielgerät.
	<i>MAXIMUM</i>	zZip ermittelt die zu verwendende Blockgröße dynamisch und wählt die größtmögliche Blockgröße für das Zielgerät.
	Beispiele:	OUT_BLKSIZE=2048 OUT_BLKSIZE=DYNAMIC
OUT_DATACLASS=	maximal 8-stelliger Name einer SMS Data Class	

	Mit dieser Option kann die SMS Data Class festgelegt werden, die für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Entkomprimiervorgangs verwendet wird.	
	Beispiel	OUT_DATACLASS=DC03
OUT_DIR_BLKs=	numerischer Wert	
	Mit dieser Option kann die Anzahl der Verzeichnisblöcke angegeben werden, die zZip für eine Bibliothek reservieren soll, die als Ausgabeziel für einen Entkomprimiervorgang verwendet wird.	
	Beispiel	OUT_DIR_BLKs=100
OUT_DSORG=	'SEQ' 'PS' 'PDS' 'PO'	
	Mit dieser Option ist zZip über den Organisationstyp einer neu anzulegenden Datei zu informieren, die als Ausgabeziel für einen Entkomprimiervorgang verwendet wird.	
	<i>SEQ</i> <i>PS</i>	Bei Wahl einer dieser Optionen erstellt zZip eine neue Datei mit Organisationstyp 'sequentiell' (sequential / partitioned sequential).
	<i>PDS</i> <i>PO</i>	Bei Wahl einer dieser Optionen erstellt zZip eine neue Bibliothek (partitioned data set / partitioned organized).
OUT_MGMTCLASS=	maximal 8-stelliger Name einer SMS Management Class	
	Mit dieser Option kann die SMS Management Class festgelegt werden, die für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Entkomprimiervorgangs verwendet wird.	
	Beispiel	OUT_MGMTCLASS=MGMUZLIB
OUT_LRECL=	numerischer Wert	
	Mit dieser numerischen Option kann die logische Satzlänge festgelegt werden, die für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Entkomprimiervorgangs verwendet wird.	
	Beispiel	OUT_LRECL=80
OUT_RECFM=	'F' 'FB' 'V' 'VB' 'U'	
	Mit dieser Option kann das Satzformat festgelegt werden, das für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Entkomprimiervorgangs verwendet wird.	
	<i>F</i>	Die Zieldatei verwendet Sätze mit fester Satzlänge.
	<i>FB</i>	Die Zieldatei verwendet Sätze mit fester Satzlänge, die geblockt verarbeitet werden.
	<i>V</i>	Die Zieldatei verwendet Sätze mit variabler Satzlänge.
	<i>VB</i>	Die Zieldatei verwendet Sätze mit variabler Satzlänge, die geblockt verarbeitet werden.
	<i>U</i>	Die Zieldatei verwendet Sätze mit undefinierter Satzlänge.
OUT_SPACE_PRI=	numerischer Wert	

	Mit dieser Option kann die Anzahl der Allokationseinheiten im Primary Extent für eine neu anzulegende sequentielle Datei bzw. für ein PO-Dataset festgelegt werden, die als Ausgabeziel für einen Entkomprimiervorgang verwendet werden.	
	Beispiel	OUT_SPACE_PRI=40
OUT_SPACE_RLSE=	'Y' 'N'	
	Mit dieser Option ist festzulegen, ob zZip nicht benötigten und zuvor reservierten Plattenplatz für eine neu erstellte sequentielle Datei bzw. für ein neu erstelltes PO-Dataset freigeben soll, nachdem der Entkomprimiervorgang beendet wurde.	
	Beispiel	OUT_SPACE_RLSE=N
OUT_SPACE_SEC=	numerischer Wert	
	Mit dieser Option kann die Anzahl der Allokationseinheiten im Secondary Extent für eine neu anzulegende sequentielle Datei bzw. für ein PO-Dataset festgelegt werden, die als Ausgabeziel für einen Entkomprimiervorgang verwendet werden.	
	Beispiel	OUT_SPACE_SEC=30
OUT_SPACE_TYPE=	'CYL' 'TRK' 'BLK'	
	Mit dieser Option ist der Typ der Allokationseinheiten für eine neu anzulegende sequentielle Datei bzw. für ein PO-Dataset festzulegen, die als Ausgabeziel für einen Entkomprimiervorgang verwendet werden. Angaben für die Optionen 'OUT_SPACE_PRI' und 'OUT_SPACE_SEC' beziehen sich auf den hier festgelegten Typ.	
	Beispiel	OUT_SPACE_TYPE=TRK
OUT_STORCLASS=	maximal 8-stelliger Name einer SMS Storage Class	
	Mit dieser Option kann die SMS Storage Class festgelegt werden, die für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Entkomprimiervorgangs verwendet wird.	
	Beispiel	OUT_STORCLASS=UNZSTOR
OUT_UNIT=	maximal 8-stelliger Name einer Unit	
	Mit dieser Option ist der Typ der Einheit anzugeben, auf der zZip eine neu anzulegende sequentielle Datei bzw. ein PO-Dataset allokiert soll, die als Ausgabeziel für einen Entkomprimiervorgang verwendet werden.	
	Beispiel	OUT_UNIT=SYSDA
OUT_VOLUME=	maximal 8-stelliger Name eines Volumes	
	Mit dieser Option ist der Name des Volumes anzugeben, auf der zZip eine neu anzulegende sequentielle Datei bzw. ein PO-Dataset allokiert soll, die als Ausgabeziel für einen Entkomprimiervorgang verwendet werden.	
	Beispiel	OUT_VOLUME=WRKLB2

OVERWRITE=	'Y' 'N'	
	<p>Mit dieser Option ist festzulegen, ob zZip bereits bestehende Zieldateien beim Entkomprimieren von Daten überschreiben soll oder nicht.</p> <p>Falls eine Zieldatei bereits existiert und die Option 'OVERWRITE=N' gewählt wurde, protokolliert zZip dies mit einer Warnung.</p>	
	Beispiel	OVERWRITE=N
PATHSEP=	darstellbares Zeichen	
	<p>Mit dieser Option ist zZip darüber zu informieren, welches Zeichen zur Trennung der Pfadbestandteile der Dateinamen im Archiv verwendet wurde.</p>	
REC_DELIM	hexadezimaler Wert (X'...') vordefinierter Wert	
	<p>Mit dieser Option kann zZip darüber informiert werden, welches Zeichen bzw. welche Zeichenkombination als Zeilenende-Kennzeichen bei der Verarbeitung von Textdaten verwendet wurde.</p>	
	Beispiele	REC_DELIM=X'0D' REC_DELIM=A-LFCR
	<p>Die nachfolgende Tabelle zeigt eine Auflistung der vordefinierten Werte, die als Zeilenende-Kennzeichen verwendet werden können:</p>	
	Bezeichner	Wert
	A-CRLFCZ	X'0D0A1A'
	CRLFCZ	X'0D0A1A'
	A-CRLF	X'0D0A'
	A-LFCR	X'0A0D'
	E-CRLF	X'0D25'
	E-CRNL	X'0D25'
	E-LFCR	X'250D'
	A-NEL	X'85'
	CRLF	X'0D0A'
	LFCR	X'0A0D'
	A-CR	X'0D'
	A-LF	X'0A'
	A-CZ	X'1A'
	E-CR	X'0D'
	E-LF	X'25'
	E-NL	X'15'
	CR	X'0D'
	LF	X'0A'
	CZ	X'1A'

ZIP/UNZIP – Komprimieren/Entkomprimieren von Archiven

Mit diesen Aktionen können mehrere Dateien in ein Archiv komprimiert bzw. aus einem Archiv entkomprimiert werden.

Im Rahmen der Komprimierung kann außerdem eine Verschlüsselung der komprimierten Daten erfolgen. Zur Entschlüsselung im Rahmen der Entkomprimierung ist lediglich die Angabe des Passwortes erforderlich, mit dem die Verschlüsselung durchgeführt wurde.

Damit eine Komprimierung erfolgt, ist eine 'ACTION'-Karte mit einer Passwort Angabe ('PWD=') einzufügen. In Abhängigkeit von der Angabe für 'ENCRYPT=' werden die komprimierten Daten auf verschiedene Weise verschlüsselt, wie die nachfolgende Tabelle zeigt:

ENCRYPT=	Beschreibung
keine Angabe	Wenn keine Angabe erfolgt, kommt das schwache PKWARE Verschlüsselungsverfahren zum Einsatz. Dieses Verfahren gilt als veraltet.
AE1/AE2	Bei dieser Angabe kommt das starke WinZip Verschlüsselungsverfahren zum Einsatz.
AES/3DES	Bei dieser Angabe kommt das starke PKWARE Verschlüsselungsverfahren zum Einsatz.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JZILI2NE JOB , 'ZIP LIB TO NET', CLASS=A, MSGCLASS=M
//ZILI2NE EXEC PGM=XPSZZIP, REGION=96M
//STEPLIB DD DISP=SHR, DSN=XPSHDRV.V300.LOADLIB
//ZIPIN DD DISP=SHR, DSN=XPSHDRV.TEST.MACLIB
//SYSIN DD *
A ACTION=ZIP COMMENT='CREATED WITH XPS-HOSTDRIVE'
A PWD=RUBINSTEIN ENCRYPT=AE2 CRYPTOMODE=CTR
I LIB=ZIPIN MBR=*
O ARCHIVE=NET='ZIPSTORE/HDRV-TEST-MACLIB.ZIP' CP=1141-E2N CRLF
O IPADDR=&IPA_MIKNO IPPORT=&IPP_MIKNO
//*
//UNNE2LI EXEC PGM=XPSZZIP, REGION=96M
//STEPLIB DD DSN=XPSHDRV.V300.LOADLIB, DISP=SHR
//XPSDATA DD DISP=SHR, DSN=XPSHDRV.V300.MACLIB
//ZIPOUT DD DISP=(NEW,CATLG), DSN=HDRV.TEST.BACKUP, DSNTYPE=PDS,
// SPACE=(6160,(5000,500,300)), VOL=SER=XPS009,
// UNIT=3390, DCB=(RECFM=FB, BLKSIZE=6160, LRECL=80)
//SYSIN DD *
A ACTION=UNZIP
A PWD=RUBINSTEIN
I ARCHIVE=NET='ZIPSTORE/HDRV-TEST-MACLIB.ZIP'
I IPADDR=&IPA_MIKNO IPPORT=&IPP_MIKNO
O LIB=ZIPOUT MBR=&&FILE CP=1141-N2E
//*
```

Abb. 18: Beispieljob für die Aktionen ZIP und UNZIP z/OS

Im Beispiel werden im ersten Jobstep unter Verwendung der Aktion 'ACTION=ZIP' alle Dateien, die sich in der durch das DD-Statement 'ZIPIN' identifizierten Bibliothek befinden, in ein ZIP-Archiv komprimiert und dann an einen HostDrive/J Server zur Speicherung im Netzwerk gesendet.

Im zweiten Jobstep wird das zuvor erstellte ZIP-Archiv wieder vom Netzwerk eingelesen und unter Verwendung der Aktion 'ACTION=UNZIP' in ein Partitioned Dataset eingespielt.

Im Rahmen des Komprimierens findet jeweils Übersetzung der Zeichendaten von EBCDIC nach ANSI und im Rahmen des Entkomprimierens eine Übersetzung der Zeichendaten von ANSI nach EBCDIC statt.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE mit vergleichbarem Ablauf.

```
* $$ JOB JNM=JZILI2NE,DISP=D,PRI=3,CLASS=0
* $$ LST CLASS=T,JNM=JZILI2NE
// JOB JZILI2NE
// LIBDEF PHASE,SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005,SYSLST
// DLBL XPSHFS,'XPSHDRV.SPOOL.SAVE',,VSAM,CAT=XPSUCAT
// EXEC XPSZZIP,SIZE=AUTO,PARM='LIB=XPS.DAEMH600'
A ACTION=ZIP COMMENT='ZVSE ARCHIV'
A PWD=RUBINSTEIN ENCRYPT=AE1 CRYPTOMODE=CTR
I LIB=XPS.HDRV300,MBR=*.D
O ARCHIVE=NET='ZIPSTORE/HDRV-TEST-ZVSE.ZIP' CP=1141-E2N
O IPADDR=&IPA_GEORG IPPORT=&IPP_GEORG
/*
// LIBDEF PHASE,SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005,SYSLST
// DLBL XPSHFS,'XPSHDRV.SPOOL.SAVE',,VSAM,CAT=XPSUCAT
// EXEC XPSZZIP,SIZE=AUTO,PARM='LIB=XPS.DAEMH600'
A ACTION=UNZIP
A PWD=RUBINSTEIN ENCRYPT=AE1 CRYPTOMODE=CTR
I ARCHIVE=NET='ZIPSTORE/HDRV-TEST-ZVSE.ZIP'
I IPADDR=&IPA_GEORG IPPORT=&IPP_GEORG
O LIB=XPS.BACKUP MBR=&&FILE CP=1141-N2E
/*
/&
* $$EOJ
```

Abb. 19: Beispieljob für die Aktionen ZIP und UNZIP z/VSE

Um zZip über die Verwendung von ZIP-Archiven in Kenntnis zu setzen, ist den jeweiligen Resourcentypen auf den Eingabe- und Ausgabekarten die Angabe 'ARCHIVE=' voranzustellen.

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Karten des Typs 'A ACTION=ZIP'.

A ACTION=ZIP					
Komprimieren von Eingabedateien und erstellen eines PKZIP-kompatiblen Archivs.					
Optionen	Beschreibung				
ARC_BLKSIZE=	numerischer Wert 'DYNAMIC' 'OPTIMUM' 'MAXIMUM'				
	Mit dieser Option kann die gewünschte Blockgröße festgelegt werden, die zZip zur Erstellung von sequentiellen Dateien bzw. von PO-Datasets verwenden soll, falls diese als Ausgabeziel für einen Komprimiervorgang verwendet werden. Die Angabe eines numerischen Wertes wird in Bytes erwartet. Die nachfolgende Tabelle beschreibt die Vorgehensweise bei der Auswahl eines der vordefinierten, konstanten Werte:				
	<table border="1"> <thead> <tr> <th>Option</th> <th>Bedeutung</th> </tr> </thead> <tbody> <tr> <td>DYNAMIC</td> <td>zZip ermittelt die zu verwendende Blockgröße dynamisch.</td> </tr> </tbody> </table>	Option	Bedeutung	DYNAMIC	zZip ermittelt die zu verwendende Blockgröße dynamisch.
Option	Bedeutung				
DYNAMIC	zZip ermittelt die zu verwendende Blockgröße dynamisch.				

	<i>OPTIMUM</i>	zZip ermittelt die zu verwendende Blockgröße dynamisch und optimiert diese für das Zielgerät.
	<i>MAXIMUM</i>	zZip ermittelt die zu verwendende Blockgröße dynamisch und wählt die größtmögliche Blockgröße für das Zielgerät.
	Beispiele:	ARC_BLKSIZE=4096 ARC_BLKSIZE=OPTIMUM
ARC_DATACLASS=	maximal 8-stelliger Name einer SMS Data Class	
	Mit dieser Option kann die SMS Data Class festgelegt werden, die für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Komprimiervorgangs verwendet wird.	
	Beispiel	ARC_DATACLASS=DC2
ARC_DIR_BLKS=	numerischer Wert	
	Mit dieser Option kann die Anzahl der Verzeichnisblöcke angegeben werden, die zZip für eine Bibliothek reservieren soll, die als Ausgabeziel für einen Komprimiervorgang verwendet wird.	
	Beispiel	ARC_DIR_BLKS=56
ARC_DSORG=	'SEQ' 'PS' 'PDS' 'PO'	
	Mit dieser Option ist zZip über den Organisationstyp einer neu anzulegenden Datei zu informieren, die als Ausgabeziel für einen Komprimiervorgang verwendet wird.	
	<i>SEQ</i> <i>PS</i>	Bei Wahl einer dieser Optionen erstellt zZip eine neue Datei mit Organisationstyp 'sequentiell' (sequential / partitioned sequential).
	<i>PDS</i> <i>PO</i>	Bei Wahl einer dieser Optionen erstellt zZip eine neue Bibliothek (partitioned data set / partitioned organized).
ARC_MGMTCLASS=	maximal 8-stelliger Name einer SMS Management Class	
	Mit dieser Option kann die SMS Management Class festgelegt werden, die für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Komprimiervorgangs verwendet wird.	
	Beispiel	ARC_MGMTCLASS=MAG30LIB
ARC_LRECL=	numerischer Wert	
	Mit dieser numerischen Option kann die logische Satzlänge festgelegt werden, die für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Komprimiervorgangs verwendet wird.	
	Beispiel	ARC_LREC=132
ARC_RECFCM=	'F' 'FB' 'V' 'VB' 'U'	
	Mit dieser Option kann das Satzformat festgelegt werden, das für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Komprimiervorgangs verwendet wird.	

	<i>F</i>	Die Zieldatei verwendet Sätze mit fester Satzlänge.
	<i>FB</i>	Die Zieldatei verwendet Sätze mit fester Satzlänge, die geblockt verarbeitet werden.
	<i>V</i>	Die Zieldatei verwendet Sätze mit variabler Satzlänge.
	<i>VB</i>	Die Zieldatei verwendet Sätze mit variabler Satzlänge, die geblockt verarbeitet werden.
	<i>U</i>	Die Zieldatei verwendet Sätze mit undefinierter Satzlänge.
ARC_SPACE_PRI=	numerischer Wert	
	Mit dieser Option kann die Anzahl der Allokationseinheiten im Primary Extent für eine neu anzulegende sequentielle Datei bzw. für ein PO-Dataset festgelegt werden, die als Ausgabeziel für einen Komprimiervorgang verwendet werden.	
	Beispiel	ARC_SPACE_PRI=25
ARC_SPACE_RLSE=	'Y' 'N'	
	Mit dieser Option ist festzulegen, ob zZip nicht benötigten und zuvor reservierten Plattenplatz für eine neu erstellte sequentielle Datei bzw. für ein neu erstelltes PO-Dataset freigeben soll, nachdem der Komprimiervorgang beendet wurde.	
	Beispiel	ARC_SPACE_RLSE=Y
ARC_SPACE_SEC=	numerischer Wert	
	Mit dieser Option kann die Anzahl der Allokationseinheiten im Secondary Extent für eine neu anzulegende sequentielle Datei bzw. für ein PO-Dataset festgelegt werden, die als Ausgabeziel für einen Komprimiervorgang verwendet werden.	
	Beispiel	ARC_SPACE_SEC=10
ARC_SPACE_TYPE=	'CYL' 'TRK' 'BLK'	
	Mit dieser Option ist der Typ der Allokationseinheiten für eine neu anzulegende sequentielle Datei bzw. für ein PO-Dataset festzulegen, die als Ausgabeziel für einen Komprimiervorgang verwendet werden. Numerische Angaben für die Optionen 'ARC_SPACE_PRI' und 'ARC_SPACE_SEC' beziehen sich auf den hier festgelegten Typ.	
	Beispiel	ARC_SPACE_TYPE=CYL
ARC_STORCLASS=	maximal 8-stelliger Name einer SMS Storage Class	
	Mit dieser Option kann die SMS Storage Class festgelegt werden, die für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Komprimiervorgangs verwendet wird.	
	Beispiel	ARC_STORCLASS=SMSSTOR
ARC_UNIT=	maximal 8-stelliger Name einer Unit	
	Mit dieser Option ist der Typ der Einheit anzugeben, auf der zZip eine neu anzulegende sequentielle Datei bzw. ein PO-Dataset allokalieren soll, die als Ausgabeziel für einen Komprimiervorgang verwendet werden.	

	Beispiel	ARC_UNIT=SYSDA
ARC_VOLUME=	maximal 8-stelliger Name eines Volumes	
	Mit dieser Option ist der Name des Volumes anzugeben, auf der zZip eine neu anzulegende sequentielle Datei bzw. ein PO-Dataset allokiert werden soll, die als Ausgabeziel für einen Komprimierungsvorgang verwendet werden.	
	Beispiel	ARC_VOLUME=XPS011
ATTR=	'Y' ' <u>N</u> '	
	<p>Bei der Erstellung von ZIP-Archiven können bestimmte Dateiattribute gespeichert werden, die von Programmen, die das Archiv zu einem späteren Zeitpunkt entkomprimieren, zur Wiederherstellung der originalen Dateien verwendet werden können.</p> <p>Wenn gewünscht wird, dass zZip Dateiattribute im ZIP-Archiv speichert, ist die Option 'ATTR=Y' zu wählen.</p> <p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 21 beschrieben, global vorgelegt werden.</p> <p>Bezüglich der Nutzung von Attributen unterstützt zZip momentan nur den Dateityp 'sequentielle Datei' unter z/OS.</p> <p>Das bedeutet, dass die Attribute, die zZip beim Erstellen eines ZIP-Archivs einer sequentiellen Datei speichert, beim Entkomprimieren des Archivs von zZip verwendet werden können, um die sequentielle Datei mit denselben Attributen, u. a. für SPACE, RECL, RECFM etc., wiederherzustellen. Siehe hierzu auch die Erläuterungen zur Option 'SEQ=' auf Seite 150.</p>	
COMMENT=	Phrase	
	Mit dieser Option kann ein Kommentar für das ZIP-Archiv angegeben werden. Der Kommentar kann mit geeigneten Programmen zur Anzeige von ZIP-Archiven, wie z. B. WinZip, angezeigt werden.	
	Beispiel	COMMENT='Backup der Umsätze für 2011/KW 21'
COMPLVL=	numerischer Wert zwischen '0' ... '9' 'STORE' 'SUPERFAST' 'FAST' ' <u>NORMAL</u> ' 'MAXIMUM'	
	Mit dieser Option ist der Komprimierungslevel bei der Erstellung komprimierter Archive festzulegen. Mit steigendem Wert verringert sich die Größe des erstellten Archivs auf Kosten der benötigten CPU Zeit. Neben der Angabe eines numerischen Wertes kann auch eine der vordefinierten Komprimierungsstufen gewählt werden:	
	vordefinierte Komprimierungsstufe	entspricht numerischem Wert
	STORE	0
	SUPERFAST	1
	FAST	2
	NORMAL	3
	MAXIMUM	4

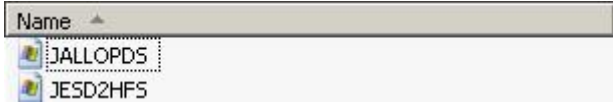
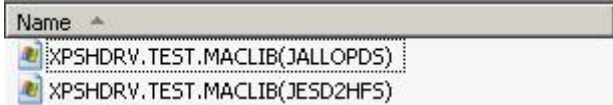
<p>CRYPTOFACILITY=</p>	<p>'XPS' 'ICSF' 'CPACF' <i>* nur in der zZipSecure Edition verfügbar</i></p>
	<p>Mit dieser Option ist festzulegen, auf welche Weise kryptographische Funktionen im Rahmen der Erstellung eines komprimierten Archivs ausgeführt werden sollen.</p> <p>Folgende Option können gewählt werden:</p> <p><u>XPS</u> Kryptographische Funktionen werden unter Verwendung der XPS CryptLib für IBM z-Systeme ausgeführt.</p> <p>Bei Nutzung der XPS CryptLib unter z/OS können rechenintensive kryptographische Berechnungen auf kostengünstige zIIP Prozessoren ausgelagert werden, falls diese präsent und aktiviert sind. Dazu ist die Option 'ZIIP=Y' in den allgemeinen Einstellungen für zZip zu aktivieren – siehe Seite 35.</p> <p>ICSF Kryptographische Funktionen werden unter z/OS zur Ausführung an die installierte 'Integrated Cryptographic Service Facility' übermittelt.</p> <p>Damit wird erreicht, dass eine installierte Hardwarelösung genutzt wird.</p> <p>CPACF Kryptographische Funktionen werden unter z/OS und unter z/VSE zur Ausführung an den installierten 'Central Processor Assist for Cryptographic Function' übermittelt.</p> <p>Damit wird erreicht, dass eine installierte Hardwarelösung genutzt wird.</p> <p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 24 beschrieben, global vorgebelegt werden.</p> <p>Es ist zu beachten, dass einige der unterstützten Kryptomodi nur bei Verwendung einer bestimmten Cryptofacility genutzt werden können.</p>
<p>Beispiel</p>	<p>CRYPTOFACILITY=CPACF</p>
<p>CRYPTOMODE=</p>	<p>'ECB' 'CBC' 'ECB_CTS' 'CBC_CTS' 'OFB' 'CTR' 'CFB' <i>* nur in der zZipSecure Edition verfügbar</i></p>
	<p><u>Definition:</u> Ein kryptographischer Modus legt fest, wie sich die Verschlüsselung mehrerer Klartextblöcke vollzieht, indem er definiert, in welcher Art der gewählte Verschlüsselungsalgorithmus auf den Datenstrom angewandt wird.</p> <p>Mit dieser Option ist der Mode of Operation festzulegen, der bei der Verschlüsselung angewendet werden soll.</p> <p>Folgende Modi werden unterstützt:</p> <p>ECB Electronic codebook</p> <p>CBC Cipher-block chaining</p> <p>ECB_CTS ECB ciphertext stealing – nur bei Verwendung der XPS CryptLib</p> <p>CBC_CTS CBC ciphertext stealing – nur bei Verwendung der XPS CryptLib</p> <p>OFB Output Feedback Mode</p> <p>CTR Counter</p> <p>CFB Cipher feedback – nur bei Verwendung von CPACF</p> <p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite</p>

	25 beschrieben, global vorbelegt werden.	
	Beispiel	CRYPTOMODE=OFB
ENCRYPT=	Verschlüsselungsalgorithmus * nur in der zZipSecure Edition verfügbar	
	<p><u>Definition:</u> Offen gelegtes Rechenverfahren, das festlegt, wie aus Blöcken von Klartext unter Verwendung eines geheimen Schlüssels Chiffre-Blöcke erzeugt werden, die nur bei Kenntnis des geheimen Schlüssels wieder in Klartext-Blöcke zurückverwandelt werden können.</p> <p>Mit dieser Option ist der Algorithmus zur Verschlüsselung der komprimierten Dateien zu wählen. Diese Option findet nur Berücksichtigung, wenn unter Verwendung der zuvor beschriebenen Option 'PWD=' ein Passwort für die Verschlüsselung angegeben wurde.</p> <p>Die nachfolgende Tabelle zeigt die unterstützten Algorithmen sowie die jeweils verwendete Schlüssellänge in Bit:</p>	
	Algorithmus	Beschreibung
	AE1	Advanced Encryption Standard / WinZip
	AE1,128	Advanced Encryption Standard / WinZip
	AE1,192	Advanced Encryption Standard / WinZip
	AE1,256	Advanced Encryption Standard / WinZip
	AE2	Advanced Encryption Standard / WinZip
	AE2,128	Advanced Encryption Standard / WinZip
	AE2,192	Advanced Encryption Standard / WinZip
	AE2,256	Advanced Encryption Standard / WinZip
	AES	Advanced Encryption Standard
	AES,128	Advanced Encryption Standard
	AES,192	Advanced Encryption Standard
	AES,256	Advanced Encryption Standard
	DES	Data Encryption Standard
	DES,56	Data Encryption Standard
	DES,112	Data Encryption Standard
	DES,168	Data Encryption Standard
	3DES	Triple Data Encryption Standard
	RC2	Rivest Cipher 2
	RC2,40	Rivest Cipher 2
	RC2,64	Rivest Cipher 2
	RC2,128	Rivest Cipher 2
	RC4	Rivest Cipher 4
	RC4,40	Rivest Cipher 4
	RC4,64	Rivest Cipher 4
	RC4,128	Rivest Cipher 4
	BLOWFISH	Blockverschlüsselung / Bruce Schneier

	BLOWFISH, 128	Blockverschlüsselung / Bruce Schneier	128
	Falls keine Angabe für den Algorithmus gemacht wird, werden die Daten gemäß dem traditionellen PKZIP Verfahren verschlüsselt.		
	Beispiele:	ENCRYPT=AES,256 ENCRYPT=BLOWFISH	
EXTSEP=	ein beliebiges Zeichen		
	Mit dieser Option ist festzulegen, welches Zeichen von zZip als Trennzeichen für eine Dateierweiterung interpretiert werden soll. Diese Angabe ist z. B. im Zusammenhang mit der Option 'CNVEXT' von Interesse, die auf Seite 22 beschrieben ist.		
	Beispiel	EXTSEP=.	
FILE_TERM=	hexadezimaler Wert (X'...') vordefinierter Wert		
	Unter Umständen kann es erforderlich oder gewünscht sein, dass das Ende einer Textdatei durch ein bestimmtes Zeichen bzw. durch eine bestimmte Zeichenkombination explizit kenntlich gemacht werden soll. Um zu erreichen, dass zZip ein besonderes Dateiende-Kennzeichen für Textdateien einfügt, ist das gewünschte Dateiende-Kennzeichen mit dieser Option festzulegen.		
	Beispiele	FILE_TERM=X'0D0A0A0D' FILE_TERM=A-CRLF	
	Die nachfolgende Tabelle zeigt eine Auflistung der vordefinierten Werte, die als Dateiende-Kennzeichen verwendet werden können:		
	Bezeichner	Wert	
	A-CRLFCZ	X'0D0A1A'	
	CRLFCZ	X'0D0A1A'	
	A-CRLF	X'0D0A'	
	A-LFCR	X'0A0D'	
	E-CRLF	X'0D25'	
	E-CRNL	X'0D25'	
	E-LFCR	X'250D'	
	A-NEL	X'85'	
	CRLF	X'0D0A'	
	LFCR	X'0A0D'	
	A-CR	X'0D'	
	A-LF	X'0A'	
	A-CZ	X'1A'	
	E-CR	X'0D'	
	E-LF	X'25'	
	E-NL	X'15'	

	<i>CR</i>	X'0D'
	<i>LF</i>	X'0A'
	<i>CZ</i>	X'1A'
MODE=	'TEXT' 'BIN' 'DETECT'	
	<p>Mit dieser Option ist festzulegen, auf welche Weise zZip die zu komprimierenden Daten interpretieren soll.</p> <p>Bei Angabe von 'TEXT' interpretiert zZip die Daten als Textdaten und verwendet die jeweils angegebene Codepage zur Übersetzung der Textdaten vor dem Komprimieren.</p> <p>Bei Angabe von 'BIN' werden die Daten als binär interpretiert und es findet keine Übersetzung statt.</p> <p>Falls der Standard 'DETECT' verwendet wird, analysiert zZip die Daten vor dem Komprimieren und ermittelt den Datentyp unter Berücksichtigung der Angaben für die Optionen 'MODE_DETECT_TAB=' und 'MODE_DETECT_LEN='.</p>	
	Beispiel	MODE=BIN
MODE_DETECT_LEN=	numerischer Wert gefolgt von 'K' 'M' 'R'	
	<p>Falls zZip den Typ der verarbeiteten Daten erkennen soll ('MODE=DETECT'), analysiert zZip die Daten an Hand der mit der Option 'MODE_DETECT_TAB=' angegebenen Tabelle.</p> <p>Mit dieser Option kann festgelegt werden, wie viel Zeichen zZip in diesem Fall maximal überprüfen soll, bis die Entscheidung, ob die Datei ausschließlich Textdaten enthält oder nicht, getroffen werden soll.</p> <p>Die Angabe kann entweder in Kilobytes ('K'), in Megabytes ('M') oder – für Textdateien - als Anzahl von Sätzen ('R') erfolgen.</p>	
	Beispiele	MODE_DETECT_LEN=330K MODE_DETECT_LEN=100R
MODE_DETECT_TAB=	Name eines Lademoduls das eine Testtabelle enthält	
	<p>Bei Wahl von 'MODE=DETECT' ermittelt zZip den Typ der Daten an Hand einer Tabelle, mit der festgestellt wird, ob die Daten Zeichen enthalten, die nicht als Textzeichen gelten.</p> <p>Im Lieferumfang von zZip befindet sich eine Standardtabelle namens 'DETECTXT', die als gleichnamiges Textmodul in der XPSDATA Bibliothek abgelegt ist.</p> <p>Falls diese Tabelle den Anforderungen nicht entsprechen sollte, können Veränderungen entweder direkt an dieser Tabelle durchgeführt werden oder an einer Kopie, die dann unter Verwendung der 'MODE_DETECT_TAB=' Option bekanntgegeben wird. Damit eine Tabelle verwendet kann, muss diese zur Laufzeit als assembliertes Lademodul verfügbar sein.</p>	
	Beispiel	MODE_DETECT_TAB=DETECTXT
NIA=	MVS-Name,Archivname	
	<p>Mit dieser Option kann bestimmt werden, auf welche Weise MVS Dateinamen vor dem Einfügen in das Archiv umbenannt werden sollen. Unter Verwendung des Wildcard Zeichens '*' ist es möglich, eine Umbenennregel für mehrere Dateinamen</p>	

festzulegen.		
<i>MVS-Name</i>	<p>Maximal 54-stelliger MVS-Dateiname wie z. B. 'MY.CLIENTS.DATA'.</p> <p>Durch Verwendung des Wildcard Zeichens können mehrere Dateien über eine Schablone angesprochen werden.</p> <p>Mit der Schablone 'MY.CLIENTS.DATA*' würden z. B. die Dateien 'MY.CLIENTS.DATA', 'MY.CLIENTS.DATA1' und 'MY.CLIENTS.DATA2' usw. ausgewählt.</p>	
<i>Archiv-Name</i>	<p>Maximal 80-stelliges Format des zu erstellenden Dateinamens für das Archiv.</p> <p>Das Format enthält den Namen, wie er in der ZIP-Datei erscheinen soll und kann Wildcard Zeichen '*' und Löschkennzeichen '+' enthalten.</p> <p>Für die Verwendung von Wildcard und Löschkennzeichen gilt die folgende Regel:</p> <p>Für jedes Wildcard Zeichen im Archivnamen wird der positionell entsprechende Wildcard Teil des MVS Namens (1., 2., 3., etc.) in den zu erstellenden Archivnamen kopiert.</p> <p>Für jedes Löschkennzeichen im Archivnamen wird der positionell entsprechende Wildcard Teil des MVS Namens (1., 2., 3., etc.) entfernt, d. h. nicht zur Bildung des zu erstellenden Archivnamens verwendet.</p>	
Die nachfolgende Tabelle enthält einige Beispiele zur Verdeutlichung der beschriebenen Regeln:		
MVS Dateiname	MVS-Name,Archivname	ZIP-Dateiname
XPS.SEQ.FILE	NIA=XPS.SEQ.FILE,XPS/ZIP	XPS/ZIP
XPS.SEQ.FILE	NIA=XPS.SEQ.FILE,XPS.TXT	XPS.TXT
XPS.SEQ.FILE	NIA=*. *.FILE,+*.ZIP	XPS.ZIP
XPS.SEQ.FILE	NIA=*. *.FILE,*. *	XPS.SEQ
XPS.SEQ.FILE	NIA=*. *.FILE,+*.INF	SEQ.INF
XPS.LIB(MEM)	NIA=XPS.LIB(*),ZIP/LIB/*	ZIP/LIB/MEM
XPS.LIB(MEM)	NIA=*. *(MEM),*/*.DAT	XPS/LIB.DAT
XPS.LIB(MEM)	NIA=*. *(*),*/*/*.DAT	XPS/LIB/MEM.DAT
XPS.LIB(MEM)	NIA=*. LIB(*),*/*/INFO	XPS/MEM/INFO
XPS.LIB(MEM)	NIA=*(*),+*.INF	MEM.INF
<p>Jeder 'XPSZZIP'-Jobstep kann mehrere 'NIA'-Karten enthalten, wodurch die Definition mehrerer Übersetzungsregeln ermöglicht wird. 'NIA'-Karten werden in der Reihenfolge der Auflistung verarbeitet, so dass die jeweils erste passende 'NIA'-Karte für die Übersetzung eines MVS Dateinamens verwendet wird.</p> <p>Bei der Wahl der Übersetzungsregeln ist darauf zu achten, dass keine ungültigen Dateinamen wie z. B. XPS.LIB(MEM).TXT generiert werden. zZip überprüft generierte Archivnamen nicht auf Gültigkeit.</p> <p>Falls für einen MVS Dateinamen keine passende 'NIA'-Regel für die Übersetzung gefunden wird, wird der Archivname gebildet, indem die Separatorpunkte '.' und die öffnende Klammer '(' jeweils durch das definierte Pfadtrennzeichen - siehe Parameter 'PATHSEP' - ersetzt werden. Schließende Klammern fallen weg.</p>		
PATH=	'Y' 'N'	

	<p>Bei der Erstellung von ZIP-Archiven aus Bibliotheken kann mit dieser Option darüber entschieden werden, ob zZip beim Erstellen des Archivs die Dateinamen voll qualifiziert in das Archiv aufnehmen soll oder nicht.</p> <p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 32 beschrieben, global vorgelegt werden.</p> <p>Die beiden nachfolgenden Abbildungen zeigen die verschiedenen Ergebnisse, je nach Wahl der Einstellung für diese Option. 'PATH=N' führt zu folgenden Dateinamen im Archiv:</p>  <p>'PATH=Y' führt zu folgendem Ergebnis:</p> 
PATHSEP=	darstellbares Zeichen
	<p>Mit dieser Option kann ein Zeichen bestimmt werden, das zZip im Rahmen der Erstellung von Pfadangaben zur Trennung einzelner Pfadbestandteile verwenden soll.</p> <p>Auf Microsoft Windows Plattformen ist dies üblicherweise der Backslash ('\'), wohingegen auf Unix- bzw. Linuxderivaten der Slash (/) verwendet wird.</p>
PKCP=	Name einer Codepage
	<p>Mit dieser Option kann der Name einer assemblierten Codepage in PKWARE kompatiblen Format angegeben werden, die zur Übersetzung von Textdaten verwendet werden soll.</p>
PROGRESS=	'Y' 'N'
	<p>Falls diese Option mit 'PROGRESS=Y' aktiviert wird, schreibt zZipSecure pro Datei, die einem komprimierten Archiv hinzugefügt wird, einen Logeintrag.</p>
PWD=	<p>Passwort * nur in der zZipSecure Edition verfügbar</p>
	<p>zZipSecure unterstützt das Standardverfahren zur Erstellung verschlüsselter ZIP-Archive.</p> <p>Bei diesem Verfahren werden die Daten auf Dateiebene, nachdem sie komprimiert wurden, zusätzlich verschlüsselt.</p> <p>Mit dieser Angabe ist das Passwort anzugeben, mit der die einzelnen Dateien des Archivs verschlüsselt werden sollen.</p> <p>Das hier angegebene Passwort muss einem Programm, das das Archiv zu einem späteren Zeitpunkt entkomprimieren soll, bekannt sein.</p> <p>Wie bei anderen sensiblen Informationen auch, empfiehlt sich die Verwendung einer Variablen für das Passwort, deren Wert in der Initialisierungsdatei 'XPSHDINI' hinterlegt ist. Damit erscheint das gewählte Passwort nicht im Jobprotokoll.</p> <p>Die Vorgehensweise zur Definition von Variablen ist auf Seite 35 beschrieben.</p>
Beispiel	PWD=&ZIP_PWD

REC_DELIM	hexadezimaler Wert (X'...') vordefinierter Wert	
	<p>Mit dieser Option kann festgelegt werden, welches Zeichen bzw. welche Zeichenkombination zZip als Zeilenende-Kennzeichen bei der Verarbeitung von Textdaten verwenden soll.</p>	
	Beispiele	REC_DELIM=X'0D' REC_DELIM=A-LFCR
	<p>Die nachfolgende Tabelle zeigt eine Auflistung der vordefinierten Werte, die als Zeilenende-Kennzeichen verwendet werden können:</p>	
	Bezeichner	Wert
	A-CRLFCZ	X'0D0A1A'
	CRLFCZ	X'0D0A1A'
	A-CRLF	X'0D0A'
	A-LFCR	X'0A0D'
	E-CRLF	X'0D25'
	E-CRNL	X'0D25'
	E-LFCR	X'250D'
	A-NEL	X'85'
	CRLF	X'0D0A'
	LFCR	X'0A0D'
	A-CR	X'0D'
	A-LF	X'0A'
	A-CZ	X'1A'
	E-CR	X'0D'
	E-LF	X'25'
	E-NL	X'15'
	CR	X'0D'
	LF	X'0A'
	CZ	X'1A'
SAVE_LRECL=	'Y' ' <u>N</u> '	
	<p>Mit dieser Option kann zZip dazu angewiesen werden, bei der Verarbeitung binärer Host Dateien, deren Satzlänge variabel ist, die Satzlänge mit zu speichern.</p> <p>Damit besteht die Möglichkeit, Dateien, die Sätze variabler Länge enthalten, im Rahmen des Entkomprimierens auf dem Host korrekt wieder herzustellen.</p> <p>Diese Option wird auf anderen Plattformen eventuell nicht unterstützt.</p> <p>Bei Verwendung von 'SAVE_LRECL=Y' wird ein entsprechendes Attribut in das resultierende Archiv eingefügt, das die Verwendung dieser Option protokolliert.</p>	
SIMULATE=	'Y' ' <u>N</u> '	
	<p>Bei Auswahl von 'SIMULATE=Y' simuliert zZip den Komprimiervorgang lediglich,</p>	

	<p>anstatt diesen tatsächlich auszuführen.</p> <p>Die Simulation des Komprimiervorgangs ist hilfreich, um festzustellen, ob die vorgegebenen 'NIA'-Regeln zur Generierung von Datei- bzw. Pfadnamen im resultierenden Archiv korrekt sind und das gewünschte Ergebnis liefern.</p>
--	--

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Karten des Typs 'A ACTION=UNZIP'.

A ACTION=UNZIP									
Entkomprimieren eines PKZIP-kompatiblen Archivs. Wenn ein Archiv entkomprimiert wird, das Attribute der ursprünglich komprimierten Dateien enthält, können die verschiedenen 'OUT_' Optionen dazu verwendet werden, die entsprechenden Attribute der Originaldateien zu überschreiben.									
Optionen	Beschreibung								
CNVEXT=	'NAMEFILE' 'KEEP' 'DROP' 'SUFFIX' 'MERGE'								
	Mit dieser Option ist festzulegen, wie zZip mit der Erweiterung von Dateinamen verfahren soll, die beim Entpacken von komprimierten Archiven anfallen. Die nachfolgende Tabelle zeigt den resultierenden Name nach Anwendung der möglichen Optionen auf den Dateinamen 'COMPANY.STATS.TXT'.								
	<table border="1"> <thead> <tr> <th>Option</th> <th>Resultat</th> </tr> </thead> <tbody> <tr> <td>NAMEFILE, KEEP</td> <td>COMPANY.STATS.TXT</td> </tr> <tr> <td>DROP</td> <td>COMPANY.STATS</td> </tr> <tr> <td>SUFFIX, MERGE</td> <td>COMPANY.STATSTXT</td> </tr> </tbody> </table>	Option	Resultat	NAMEFILE, KEEP	COMPANY.STATS.TXT	DROP	COMPANY.STATS	SUFFIX, MERGE	COMPANY.STATSTXT
Option	Resultat								
NAMEFILE, KEEP	COMPANY.STATS.TXT								
DROP	COMPANY.STATS								
SUFFIX, MERGE	COMPANY.STATSTXT								
CRYPTOFACILITY=	'XPS' 'ICSF' 'CPACF' * nur in der zZipSecure Edition verfügbar								
	<p>Mit dieser Option ist festzulegen, auf welche Weise kryptographische Funktionen im Rahmen der Entkomprimierung des Archivs ausgeführt werden sollen.</p> <p>Folgende Option können gewählt werden:</p> <p>XPS Kryptographische Funktionen werden unter Verwendung der XPS CryptLib für IBM z-Systeme ausgeführt. Bei Nutzung der XPS CryptLib unter z/OS können rechenintensive kryptographische Berechnungen auf kostengünstige zIIP Prozessoren ausgelagert werden, falls diese präsent und aktiviert sind. Dazu ist die Option 'ZIIP=Y' in den allgemeinen Einstellungen für zZip zu aktivieren – siehe Seite 35.</p> <p>ICSF Kryptographische Funktionen werden unter z/OS zur Ausführung an die installierte 'Integrated Cryptographic Service Facility' übermittlelt. Damit wird erreicht, dass eine installierte Hardwarelösung genutzt wird.</p> <p>CPACF Kryptographische Funktionen werden unter z/OS und unter z/VSE zur Ausführung an den installierten 'Central Processor Assist for Cryptographic Function' übermittlelt. Damit wird erreicht, dass eine installierte Hardwarelösung genutzt wird.</p> <p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 24 beschrieben, global vorbelegt werden.</p> <p>Es ist zu beachten, dass einige der unterstützten Kryptomodi nur bei Verwendung einer bestimmten Cryptofacility genutzt werden können.</p>								
	<table border="1"> <tr> <td>Beispiel</td> <td>CRYPTOFACILITY=CPACF</td> </tr> </table>	Beispiel	CRYPTOFACILITY=CPACF						
Beispiel	CRYPTOFACILITY=CPACF								
FILE_TERM=	hexadezimaler Wert (X'...') vordefinierter Wert								

<p>Beim Erstellen komprimierter Archive kann eine besondere Zeichensequenz definiert werden, die als Dateiende-Kennzeichen an jede Daten angefügt werden soll.</p> <p>Falls ein Archiv entkomprimiert wird, für das ein spezielles Dateiende-Kennzeichen definiert wurde, kann die dazu verwendete Zeichensequenz mit dieser Option bekannt gegeben werden.</p> <p>Im Rahmen des Entkomprimierens entfernt zZip diese zuvor beim Komprimieren hinzugefügte Zeichensequenz und transferiert die Daten damit wieder in denselben Zustand, den diese vor dem Komprimieren hatten.</p>	
Beispiele	<pre>FILE_TERM=X'0D0AAABB'</pre> <pre>FILE_TERM= LFCR</pre>
<p>Die nachfolgende Tabelle zeigt eine Auflistung der vordefinierten Werte, die als Dateiende-Kennzeichen verwendet werden können:</p>	
Bezeichner	Wert
<i>A-CRLFCZ</i>	X'0D0A1A'
<i>CRLFCZ</i>	X'0D0A1A'
<i>A-CRLF</i>	X'0D0A'
<i>A-LFCR</i>	X'0A0D'
<i>E-CRLF</i>	X'0D25'
<i>E-CRNL</i>	X'0D25'
<i>E-LFCR</i>	X'250D'
<i>A-NEL</i>	X'85'
<i>CRLF</i>	X'0D0A'
<i>LFCR</i>	X'0A0D'
<i>A-CR</i>	X'0D'
<i>A-LF</i>	X'0A'
<i>A-CZ</i>	X'1A'
<i>E-CR</i>	X'0D'
<i>E-LF</i>	X'25'
<i>E-NL</i>	X'15'
<i>CR</i>	X'0D'
<i>LF</i>	X'0A'
<i>CZ</i>	X'1A'
NOA=	[Archivpfad],[MVS-HLQ]
<p>Mit dieser Option kann ein Schema definiert werden, das zur Veränderung des High Level Qualifiers für einen Dateinamen im Rahmen der Entkomprimierung verwendet werden soll.</p> <p>Das Schema findet Anwendung bei der Transformation vom Archivnamen zum MVS Namen. Dabei werden Archivnamen, die dem spezifizierten Archivpfad entsprechen durch den angegebenen MVS-HLQ ersetzt. Die Verwendung von Wildcard Zeichen '*' zur Anwendung auf mehrere Archivpfade ist möglich.</p> <p>Das Weglassen entweder des Archivpfades oder des MVS-HLQ ist zulässig.</p> <p>Jeder 'XPSZZIP'-Jobstep kann mehrere 'NOA'-Karten enthalten, wodurch die Definition</p>	

<p>mehrerer Ersetzungsregeln ermöglicht wird. 'NOA'-Karten werden in der Reihenfolge der Auflistung verarbeitet, so dass die jeweils erste passende 'NOA'-Karte für die Ersetzung eines Archivnamens verwendet wird.</p>		
<i>Archivpfad</i>	<p>An dieser Stelle ist der High Level Qualifier anzugeben, der mit den im Zip Archiv gespeicherten Namen auf Übereinstimmung hin zu überprüfen ist, und der im Falle der Übereinstimmung durch den angegebenen MVS-HLQ ersetzt werden soll.</p> <p>Die Angabe darf nicht länger als 80 Zeichen sein.</p> <p>Das Wildcard Zeichen '*' kann dazu verwendet werden, Bezug auf genau einen Level des Archivnamens zu nehmen.</p> <p>Das Wildcard Zeichen '?' kann dazu verwendet werden, Bezug auf genau ein Zeichen des Archivnamens – mit Ausnahme der Separatorpunkte '.' - zu nehmen.</p> <p>Durch die Verwendung des doppelten Wildcard Zeichens '**' kann der komplette Archivpfad ersetzt werden.</p>	
<i>MVS-HLQ</i>	<p>An dieser Stelle ist anzugeben, mit welchem HLQ die durch den ersten Operand spezifizierten Teile des Archivnamens ersetzt und dem restlichen Teil des Archivnamens vorangestellt werden sollen.</p> <p>Die Angabe darf nicht länger als 54 Zeichen sein und muss den MVS Namenskonventionen entsprechen.</p>	
<p>Die nachfolgende Tabelle enthält einige Beispiele zur Verdeutlichung der beschriebenen Regeln:</p>		
Archivname	[Archivpfad],[MVS-HLQ]	MVS Dateiname
XPS/MEMBER/INFO	NOA=XPS.M,NOV. { ersetze 'XPS.M' durch 'NOV.' }	NOV.EMBER.INFO
XPS/MEMBER/INFO	NOA=*,NOV { ersetze den ersten Level durch 'NOV' }	NOV.MEMBER.INFO
XPS/MEMBER/INFO	NOA=XPS.?EMBER,NOV { ersetze 'XPS.?EMBER' durch 'NOV' }	NOV.INFO
XPS/MEMBER/INFO	NOA=XPS.?EMBER.,NOV { ersetze 'XPS.?EMBER.' durch 'NOV' }	NOVINFO
XPS/MEMBER/INFO	NOA=,NEW. { stelle 'NEW.' voran }	NEW.XPS.MEMBER.INFO
XPS/MEMBER/INFO	NOA=,NEW { stelle 'NEW' voran }	NEWXPS.MEMBER.INFO
XPS/MEMBER/INFO	NOA=XP, { entferne ein führendes 'XP' }	S.MEMBER.INFO
XPS/MEMBER/INFO	NOA=**,NEW.FILE.TYPE { ersetze den kompletten Namen }	NEW.FILE.TYPE
<p>Bevor die jeweils erste passende 'NOA'-Regel auf einen Archivnamen angewendet wird, wird zunächst die Einstellung für die Option 'CNVEXT=' auf den Archivnamen angewendet.</p>		
Archivname	CNVEXT [Archivpfad],[MVS-HLQ]	MVS Dateiname
XPS/MEMBER/README.TXT	CNVEXT=DROP NOA=XPS.MEMBER.,MYHLQ.	MYHLQ.README
XPS/MEMBER/README.TXT	CNVEXT=NAMEFILE NOA=XPS.MEMBER.,MYHLQ.	MYHLQ.README.TXT

OUT_BLKSIZE=	numerischer Wert 'DYNAMIC' 'OPTIMUM' 'MAXIMUM'	
	<p>Mit dieser Option kann die gewünschte Blockgröße festgelegt werden, die zZip zur Erstellung von sequentiellen Dateien bzw. von PO-Datasets verwenden soll, falls diese als Ausgabeziel für einen Entkomprimierungsvorgang verwendet werden.</p> <p>Die Angabe eines numerischen Wertes wird in Bytes erwartet.</p> <p>Die nachfolgende Tabelle beschreibt die Vorgehensweise bei der Auswahl eines der vordefinierten, konstanten Werte:</p>	
	Option	Bedeutung
	<i>DYNAMIC</i>	zZip ermittelt die zu verwendende Blockgröße dynamisch.
	<i>OPTIMUM</i>	zZip ermittelt die zu verwendende Blockgröße dynamisch und optimiert diese für das Zielgerät.
	<i>MAXIMUM</i>	zZip ermittelt die zu verwendende Blockgröße dynamisch und wählt die größtmögliche Blockgröße für das Zielgerät.
	Beispiele:	OUT_BLKSIZE=2048 OUT_BLKSIZE=DYNAMIC
OUT_DATACLASS=	maximal 8-stelliger Name einer SMS Data Class	
	<p>Mit dieser Option kann die SMS Data Class festgelegt werden, die für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Entkomprimierungsvorgangs verwendet wird.</p>	
	Beispiel	OUT_DATACLASS=DC03
OUT_DIR_BLKs=	numerischer Wert	
	<p>Mit dieser Option kann die Anzahl der Verzeichnisblöcke angegeben werden, die zZip für eine Bibliothek reservieren soll, die als Ausgabeziel für einen Entkomprimierungsvorgang verwendet wird.</p>	
	Beispiel	OUT_DIR_BLKs=100
OUT_DSORG=	'SEQ' 'PS' 'PDS' 'PO'	
	<p>Mit dieser Option ist zZip über den Organisationstyp einer neu anzulegenden Datei zu informieren, die als Ausgabeziel für einen Entkomprimierungsvorgang verwendet wird.</p>	
	<i>SEQ</i> <i>PS</i>	Bei Wahl einer dieser Optionen erstellt zZip eine neue Datei mit Organisationstyp 'sequentiell' (sequential / partitioned sequential).
	<i>PDS</i> <i>PO</i>	Bei Wahl einer dieser Optionen erstellt zZip eine neue Bibliothek (partitioned data set / partitioned organized).
OUT_MGMTCLASS=	maximal 8-stelliger Name einer SMS Management Class	
	<p>Mit dieser Option kann die SMS Management Class festgelegt werden, die für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Entkomprimierungsvorgangs verwendet wird.</p>	
	Beispiel	OUT_MGMTCLASS=MGMUZLIB
OUT_LRECL=	numerischer Wert	

	Mit dieser numerischen Option kann die logische Satzlänge festgelegt werden, die für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Entkomprimiervorgangs verwendet wird.	
	Beispiel	OUT_LRECL=80
OUT_RECFCM=	'F' 'FB' 'V' 'VB' 'U'	
	Mit dieser Option kann das Satzformat festgelegt werden, das für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Entkomprimiervorgangs verwendet wird.	
	F	Die Zielfeile verwendet Sätze mit fester Satzlänge.
	FB	Die Zielfeile verwendet Sätze mit fester Satzlänge, die geblockt verarbeitet werden.
	V	Die Zielfeile verwendet Sätze mit variabler Satzlänge.
	VB	Die Zielfeile verwendet Sätze mit variabler Satzlänge, die geblockt verarbeitet werden.
	U	Die Zielfeile verwendet Sätze mit undefinierter Satzlänge.
OUT_SPACE_PRI=	numerischer Wert	
	Mit dieser Option kann die Anzahl der Allokationseinheiten im Primary Extent für eine neu anzulegende sequentielle Datei bzw. für ein PO-Dataset festgelegt werden, die als Ausgabeziel für einen Entkomprimiervorgang verwendet werden.	
	Beispiel	OUT_SPACE_PRI=40
OUT_SPACE_RLSE=	'Y' 'N'	
	Mit dieser Option ist festzulegen, ob zZip nicht benötigten und zuvor reservierten Plattenplatz für eine neu erstellte sequentielle Datei bzw. für ein neu erstelltes PO-Dataset freigeben soll, nachdem der Entkomprimiervorgang beendet wurde.	
	Beispiel	OUT_SPACE_RLSE=N
OUT_SPACE_SEC=	numerischer Wert	
	Mit dieser Option kann die Anzahl der Allokationseinheiten im Secondary Extent für eine neu anzulegende sequentielle Datei bzw. für ein PO-Dataset festgelegt werden, die als Ausgabeziel für einen Entkomprimiervorgang verwendet werden.	
	Beispiel	OUT_SPACE_SEC=30
OUT_SPACE_TYPE=	'CYL' 'TRK' 'BLK'	
	Mit dieser Option ist der Typ der Allokationseinheiten für eine neu anzulegende sequentielle Datei bzw. für ein PO-Dataset festzulegen, die als Ausgabeziel für einen Entkomprimiervorgang verwendet werden. Angaben für die Optionen 'OUT_SPACE_PRI' und 'OUT_SPACE_SEC' beziehen sich auf den hier festgelegten Typ.	
	Beispiel	OUT_SPACE_TYPE=TRK
OUT_STORCLASS=	maximal 8-stelliger Name einer SMS Storage Class	

	Mit dieser Option kann die SMS Storage Class festgelegt werden, die für die Erstellung einer neuen sequentiellen Datei bzw. eines PO-Datasets für das Ausgabeziel eines Entkomprimiervorgangs verwendet wird.	
	Beispiel	OUT_STORCLASS=UNZSTOR
OUT_UNIT=	maximal 8-stelliger Name einer Unit	
	Mit dieser Option ist der Typ der Einheit anzugeben, auf der zZip eine neu anzulegende sequentielle Datei bzw. ein PO-Dataset allokalieren soll, die als Ausgabeziel für einen Entkomprimiervorgang verwendet werden.	
	Beispiel	OUT_UNIT=SYSDA
OUT_VOLUME=	maximal 8-stelliger Name eines Volumes	
	Mit dieser Option ist der Name des Volumes anzugeben, auf der zZip eine neu anzulegende sequentielle Datei bzw. ein PO-Dataset allokalieren soll, die als Ausgabeziel für einen Entkomprimiervorgang verwendet werden.	
	Beispiel	OUT_VOLUME=WRKLB2
OVERWRITE=	'Y' 'N'	
	Mit dieser Option ist festzulegen, ob zZip bereits bestehende Zieldateien beim Entkomprimieren von Daten überschreiben soll oder nicht. Falls eine Zieldatei bereits existiert und die Option 'OVERWRITE=N' gewählt wurde, protokolliert zZip dies mit einer Warnung.	
PATHSEP=	darstellbares Zeichen	
	Mit dieser Option kann ein Zeichen bestimmt werden, das im Rahmen der Erstellung von Pfadangaben zur Trennung einzelner Pfadbestandteile verwendet wurde. Auf Microsoft Windows Plattformen ist dies üblicherweise der Backslash ('\'), wohingegen auf Unix- bzw. Linuxderivaten der Slash (/) verwendet wird.	
PKCP=	Name einer Codepage	
	Mit dieser Option kann der Name einer assemblierten Codepage in PKWARE kompatiblen Format angegeben werden, die zur Übersetzung von Textdaten verwendet werden soll.	
PROGRESS=	'Y' 'N'	
	Falls diese Option mit 'PROGRESS=Y' aktiviert wird, schreibt zZipSecure pro Datei, die entkomprimiert wird, einen Logeintrag.	
PWD=	Passwort <i>* nur in der zZipSecure Edition verfügbar</i>	
	Falls zZipSecure ein verschlüsseltes Archiv entkomprimieren soll, ist mit dieser Option das Passwort bekannt zu geben, das zur Verschlüsselung der Dateien des Archivs verwendet wurde. Wie bei anderen sensiblen Informationen auch, empfiehlt sich die Verwendung einer Variablen für das Passwort, deren Wert in der Initialisierungsdatei 'XPSHDINI' hinterlegt	

	<p>ist. Damit erscheint das gewählte Passwort nicht im Jobprotokoll. Die Vorgehensweise zur Definition von Variablen ist auf Seite 35 beschrieben.</p>	
	Beispiel	PWD=&ZIP_PWD
REC_DELIM	hexadezimaler Wert (X'...') vordefinierter Wert	
	<p>Mit dieser Option kann zZip darüber informiert werden, welches Zeichen bzw. welche Zeichenkombination als Zeilenende-Kennzeichen bei der Verarbeitung von Textdaten verwendet wurde.</p>	
	Beispiele	<p>REC_DELIM=X'0D' REC_DELIM=A-LFCR</p>
	<p>Die nachfolgende Tabelle zeigt eine Auflistung der vordefinierten Werte, die als Zeilenende-Kennzeichen verwendet werden können:</p>	
	Bezeichner	Wert
	A-CRLFCZ	X'0D0A1A'
	CRLFCZ	X'0D0A1A'
	A-CRLF	X'0D0A'
	A-LFCR	X'0A0D'
	E-CRLF	X'0D25'
	E-CRNL	X'0D25'
	E-LFCR	X'250D'
	A-NEL	X'85'
	CRLF	X'0D0A'
	LFCR	X'0A0D'
	A-CR	X'0D'
	A-LF	X'0A'
	A-CZ	X'1A'
	E-CR	X'0D'
	E-LF	X'25'
	E-NL	X'15'
	CR	X'0D'
	LF	X'0A'
	CZ	X'1A'
SIMULATE=	'Y' 'N'	
	<p>Bei Auswahl von 'SIMULATE=Y' simuliert zZip den Entkomprimiervorgang lediglich, anstatt diesen tatsächlich auszuführen. Die Simulation des Entkomprimiervorgangs ist hilfreich, um festzustellen, ob die vorgegebenen 'NOA'-Regeln zur Generierung von MVS Dateinamen das gewünschte</p>	

	Ergebnis liefern.
--	-------------------

Managementfunktionen für ZIP-Archive

Neben dem Erstellen und Verarbeiten ganzer ZIP-Archive stellt zZip noch eine Reihe weiterer Aktionen zur Verwaltung bzw. Änderung von ZIP-Archiven bereit.

Die folgende Tabelle zeigt eine kurze Übersicht der verfügbaren Funktionen für bestehende ZIP-Archive:

Funktion	Beschreibung
ZIP,ADD	Hinzufügen von Dateien zu einem ZIP-Archiv.
ZIP,COPY	Kopieren von Dateien aus einem ZIP-Archiv in ein anderes.
ZIP,DEL	Löschen von Dateien aus einem ZIP-Archiv.
ZIP,UPD	Ersetzen von Dateien in einem ZIP-Archiv.
ZIP,VIEW	Erstellen des Inhaltsverzeichnisses eines ZIP-Archivs.

ZIP,ADD - Dateien zu einem ZIP-Archiv hinzufügen

Mit der Aktion 'ZIP,ADD' können Dateien zu einem bestehenden ZIP-Archiv hinzugefügt werden.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JZIPADD JOB , 'ADD TO ZIP ARCHIV',CLASS=A,MSGCLASS=M
//ZIPADD EXEC PGM=XPSZZIP,REGION=96M
//STEPLIB DD DISP=SHR,DSN=XPSHDRV.V300.LOADLIB
//LIBTOADD DD DISP=SHR,DSN=XPSHDRV.V300.MACLIB
//SYSIN DD *
A ACTION=ZIP,ADD PATH=Y PROGRESS=YES
I ARCHIVE=NET= 'ZIPSTORE/ZIP-MANAGEMENT.ZIP '
I IPADDR=&IPA_MIKNO IPPORT=&IPP_MIKNO
I LIB=LIBTOADD MBR=*
O ARCHIVE=NET= 'ZIPSTORE/ZIP-MANAGEMENT-ADD.ZIP '
O IPADDR=&IPA_MIKNO IPPORT=&IPP_MIKNO CP=1141-E2N
//*
```

Abb. 20: Beispieljob für die Aktion ZIP,ADD z/OS

Im Beispiel werden unter Verwendung der Aktion 'ACTION=ZIP,ADD' neue Dateien zu einem bereits bestehenden ZIP-Archiv hinzugefügt.

Das zu erweiternde ZIP-Archiv wird zunächst mit den ersten beiden 'I'-Karten definiert.

Mit der dritten 'I'-Karte wird zZip darüber informiert, dass alle Dateien aus der durch das DD-Statement identifizierten Bibliothek zum ZIP-Archiv hinzugefügt werden sollen.

Die beiden abschließenden 'O'-Karten legen fest, dass das resultierende ZIP-Archiv zur Speicherung im Netzwerk an einen HostDrive/J Server gesendet werden soll.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE mit vergleichbarem Ablauf.

```
* $$ JOB JNM=JZIPADD,DISP=D,PRI=3,CLASS=0
* $$ LST CLASS=T,JNM=JZIPADD
// JOB JZIPADD
// LIBDEF PHASE,SEARCH=(XPS.DAEMH600)
```

```
// OPTION SYSPARM='00'
// ASSGN SYS005,SYSLST
// DLBL XPSHFS,'XPSHDRV.SPOOL.SAVE',,VSAM,CAT=XPSUCAT
// EXEC XPSZZIP,SIZE=AUTO,PARM='LIB=XPS.DAEMH600'
A ACTION=ZIP,ADD PATH=Y PROGRESS=YES
I ARCHIVE=NET='ZIPSTORE/HDRV-TEST-ZVSE.ZIP'
I IPADDR=&IPA_GEORG IPPORT=&IPP_GEORG
I LIB=XPS.DAEMV500 MBR=*
F INCLUDE=IN*.*
O ARCHIVE=NET='ZIPSTORE/HDRV-TEST-ZVSE-ADD.ZIP'
O IPADDR=&IPA_GEORG IPPORT=&IPP_GEORG CP=1141-E2N
/*
/&
* $$EQJ
```

Abb. 21: Beispieljob für die Aktion ZIP,ADD z/VSE

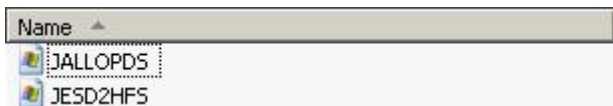
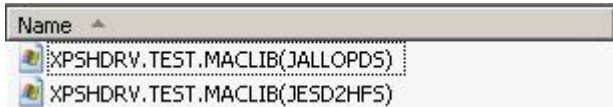
Um zZip über die Verwendung von ZIP-Archiven in Kenntnis zu setzen, ist den jeweiligen Resourcentypen auf den Eingabe- und Ausgabekarten die Angabe 'ARCHIVE=' voranzustellen.

Die folgende Tabelle zeigt die zulässigen Optionen für Karten des Typs 'A ACTION=ZIP,ADD'.

A ACTION=ZIP,ADD													
Hinzufügen von Dateien zu einem PKZIP-kompatiblen Archiv.													
Optionen	Beschreibung												
COMMENT=	Phrase												
	Mit dieser Option kann ein Kommentar für das ZIP-Archiv angegeben werden. Der Kommentar kann mit geeigneten Programmen zur Anzeige von ZIP-Archiven, wie z. B. WinZip, angezeigt werden.												
	Beispiel COMMENT='Backup der Umsätze für 2011/KW 21'												
COMPRLVL=	numerischer Wert zwischen '0' ... '9' 'STORE' 'SUPERFAST' 'FAST' 'NORMAL' 'MAXIMUM'												
	Mit dieser Option ist der Komprimierungslevel bei der Erstellung komprimierter Archive festzulegen. Mit steigendem Wert verringert sich die Größe des erstellten Archivs auf Kosten der benötigten CPU Zeit. Neben der Angabe eines numerischen Wertes kann auch eine der vordefinierten Komprimierungsstufen gewählt werden:												
	<table border="1"> <thead> <tr> <th>vordefinierte Komprimierungsstufe</th> <th>entspricht numerischem Wert</th> </tr> </thead> <tbody> <tr> <td>STORE</td> <td>0</td> </tr> <tr> <td>SUPERFAST</td> <td>1</td> </tr> <tr> <td>FAST</td> <td>2</td> </tr> <tr> <td>NORMAL</td> <td>3</td> </tr> <tr> <td>MAXIMUM</td> <td>4</td> </tr> </tbody> </table>	vordefinierte Komprimierungsstufe	entspricht numerischem Wert	STORE	0	SUPERFAST	1	FAST	2	NORMAL	3	MAXIMUM	4
vordefinierte Komprimierungsstufe	entspricht numerischem Wert												
STORE	0												
SUPERFAST	1												
FAST	2												
NORMAL	3												
MAXIMUM	4												
CRYPTOFACILITY=	'XPS' 'ICSF' 'CPACF' * nur in der zZipSecure Edition verfügbar												
	Mit dieser Option ist festzulegen, auf welche Weise kryptographische Funktionen im Rahmen der Erstellung eines komprimierten Archivs ausgeführt werden sollen.												

	<p>Folgende Option können gewählt werden:</p> <p>XPS Kryptographische Funktionen werden unter Verwendung der XPS CryptLib für IBM z-Systeme ausgeführt.</p> <p>Bei Nutzung der XPS CryptLib unter z/OS können rechenintensive kryptographische Berechnungen auf kostengünstige zIIP Prozessoren ausgelagert werden, falls diese präsent und aktiviert sind. Dazu ist die Option 'ZIIP=Y' in den allgemeinen Einstellungen für zZip zu aktivieren – siehe Seite 35.</p> <p>ICSF Kryptographische Funktionen werden unter z/OS zur Ausführung an die installierte 'Integrated Cryptographic Service Facility' übermittlelt.</p> <p>Damit wird erreicht, dass eine installierte Hardwarelösung genutzt wird.</p> <p>CPACF Kryptographische Funktionen werden unter z/OS und unter z/VSE zur Ausführung an den installierten 'Central Processor Assist for Cryptographic Function' übermittlelt.</p> <p>Damit wird erreicht, dass eine installierte Hardwarelösung genutzt wird.</p> <p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 24 beschrieben, global vorbelegt werden.</p> <p>Es ist zu beachten, dass einige der unterstützten Kryptomodi nur bei Verwendung einer bestimmten Cryptofacility genutzt werden können.</p>		
	<table border="1"> <tr> <td data-bbox="359 943 475 1008">Beispiel</td> <td data-bbox="475 943 1383 1008">CRYPTOFACILITY=CPACF</td> </tr> </table>	Beispiel	CRYPTOFACILITY=CPACF
Beispiel	CRYPTOFACILITY=CPACF		
<p>CRYPTOMODE=</p>	<p>'ECB' 'CBC' 'ECB_CTS' 'CBC_CTS' 'OFB' 'CTR' 'CFB' <i>* nur in der zZipSecure Edition verfügbar</i></p>		
	<p>Definition: Ein kryptographischer Modus legt fest, wie sich die Verschlüsselung mehrerer Klartextblöcke vollzieht, indem er definiert, in welcher Art der gewählte Verschlüsselungsalgorithmus auf den Datenstrom angewandt wird.</p> <p>Mit dieser Option ist der Mode of Operation festzulegen, der bei der Verschlüsselung angewendet werden soll.</p> <p>Folgende Modi werden unterstützt:</p> <p>ECB Electronic codebook</p> <p>CBC Cipher-block chaining</p> <p>ECB_CTS ECB ciphertext stealing – nur bei Verwendung der XPS CryptLib</p> <p>CBC_CTS CBC ciphertext stealing – nur bei Verwendung der XPS CryptLib</p> <p>OFB Output Feedback Mode</p> <p>CTR Counter</p> <p>CFB Cipher feedback – nur bei Verwendung von CPACF</p> <p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 25 beschrieben, global vorbelegt werden.</p>		
<p>ENCRYPT=</p>	<p>Algorithmus <i>* nur in der zZipSecure Edition verfügbar</i></p>		

<p>Definition: Offen gelegtes Rechenverfahren, das festlegt, wie aus Blöcken von Klartext unter Verwendung eines geheimen Schlüssels Chiffre-Blöcke erzeugt werden, die nur bei Kenntnis des geheimen Schlüssels wieder in Klartext-Blöcke zurückverwandelt werden können.</p> <p>Mit dieser Option ist der Algorithmus zur Verschlüsselung der komprimierten Dateien zu wählen. Diese Option findet nur Berücksichtigung, wenn unter Verwendung der zuvor beschriebenen Option 'PWD=' ein Passwort für die Verschlüsselung angegeben wurde.</p> <p>Die nachfolgende Tabelle zeigt die unterstützten Algorithmen sowie die jeweils verwendete Schlüssellänge in Bit:</p>		
Algorithmus	Beschreibung	Schlüssellänge in Bit
AE1	Advanced Encryption Standard / WinZip	128
AE1,128	Advanced Encryption Standard / WinZip	128
AE1,192	Advanced Encryption Standard / WinZip	192
AE1,256	Advanced Encryption Standard / WinZip	256
AE2	Advanced Encryption Standard / WinZip	128
AE2,128	Advanced Encryption Standard / WinZip	128
AE2,192	Advanced Encryption Standard / WinZip	192
AE2,256	Advanced Encryption Standard / WinZip	256
AES	Advanced Encryption Standard	128
AES,128	Advanced Encryption Standard	128
AES,192	Advanced Encryption Standard	192
AES,256	Advanced Encryption Standard	256
DES	Data Encryption Standard	56
DES,56	Data Encryption Standard	56
DES,112	Data Encryption Standard	112
DES,168	Data Encryption Standard	168
3DES	Triple Data Encryption Standard	168
RC2	Rivest Cipher 2	128
RC2,40	Rivest Cipher 2	40
RC2,64	Rivest Cipher 2	64
RC2,128	Rivest Cipher 2	128
RC4	Rivest Cipher 4	128
RC4,40	Rivest Cipher 4	40
RC4,64	Rivest Cipher 4	64
RC4,128	Rivest Cipher 4	128
BLOWFISH	Blockverschlüsselung / Bruce Schneier	128
BLOWFISH,128	Blockverschlüsselung / Bruce Schneier	128
<p>Falls keine Angabe für den Algorithmus gemacht wird, werden die Daten gemäß dem traditionellen PKZIP Verfahren verschlüsselt.</p>		
Beispiele:	<p>ENCRYPT=AES,256 ENCRYPT=BLOWFISH</p>	
OPTION=	'MERGE' 'REPLACE'	

	<p>Mit dieser Option kann gesteuert werden, wie zZip verfahren soll, falls dem ZIP-Archiv Dateien hinzugefügt werden sollen, die bereits darin enthalten sind.</p> <p>MERGE In der Standardeinstellung werden nur Dateien zum Archiv hinzugefügt, die noch nicht darin enthalten sind.</p> <p>REPLACE Bei Wahl dieser Einstellung werden Dateien, die bereits im Archiv enthalten sind, durch Dateien mit gleichem Namen ersetzt, falls diese in der Liste der hinzuzufügenden Dateien enthalten sind.</p>		
PATH=	'Y' 'N'		
	<p>Bei der Erstellung von ZIP-Archiven aus Bibliotheken kann mit dieser Option darüber entschieden werden, ob zZip beim Erstellen des Archivs die Dateinamen voll qualifiziert in das Archiv aufnehmen soll oder nicht.</p> <p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 32 beschrieben, global vorgelegt werden.</p> <p>Die beiden nachfolgenden Abbildungen zeigen die verschiedenen Ergebnisse, je nach Wahl der Einstellung für diese Option. 'PATH=N' führt zu folgenden Dateinamen im Archiv:</p>  <p>'PATH=Y' führt zu folgendem Ergebnis:</p> 		
PROGRESS=	'Y' 'N'		
	Falls diese Option mit 'PROGRESS=Y' aktiviert wird, schreibt zZip pro Datei, die einem komprimierten Archiv hinzugefügt wird, einen Logeintrag.		
PWD=	<p>Passwort</p> <p><i>* nur in der zZipSecure Edition verfügbar</i></p>		
	<p>zZipSecure unterstützt das Standardverfahren zur Erstellung verschlüsselter ZIP-Archive.</p> <p>Bei diesem Verfahren werden die Daten auf Dateiebene, nachdem sie komprimiert wurden, zusätzlich verschlüsselt.</p> <p>Mit dieser Angabe ist das Passwort anzugeben, mit der die einzelnen Dateien des Archivs verschlüsselt werden sollen.</p> <p>Das hier angegebene Passwort muss einem Programm, das das Archiv zu einem späteren Zeitpunkt entkomprimieren soll, bekannt sein.</p> <p>Wie bei anderen sensiblen Informationen auch, empfiehlt sich die Verwendung einer Variablen für das Passwort, deren Wert in der Initialisierungsdatei 'XPSHDINI' hinterlegt ist. Damit erscheint das gewählte Passwort nicht im Jobprotokoll.</p> <p>Die Vorgehensweise zur Definition von Variablen ist auf Seite 35 beschrieben.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>PWD=&ZIP_PWD</td> </tr> </table>	Beispiel	PWD=&ZIP_PWD
Beispiel	PWD=&ZIP_PWD		

ZIP,COPY - Dateien von einem ZIP-Archiv in ein anderes kopieren

Mit der Aktion 'ZIP,COPY' können Dateien aus einem ZIP-Archiv in ein anderes ZIP-Archiv kopiert werden.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JZIPCOPY JOB , 'COPY ZIP TO ZIP',CLASS=A,MSGCLASS=M
//ZIPCOPY EXEC PGM=XPSZZIP,REGION=96M
//STEPLIB DD DISP=SHR,DSN=XPSHDRV.V300.LOADLIB
//SYSIN DD *
A ACTION=ZIP,COPY PROGRESS=Y OPTION=REPLACE
I ARCHIVE=NET='ZIPSTORE/ZIP-MANAGEMENT.ZIP'
I IPADDR=&IPA_MIKNO IPPORT=&IPP_MIKNO
I COPYARC=NET='ZIPSTORE/ZIP-MANAGEMENT-ADD.ZIP'
I IPADDR=&IPA_MIKNO IPPORT=&IPP_MIKNO
F INCLUDE=XPSHDRV.V300.MACLIB(COMP*)
O ARCHIVE=NET='ZIPSTORE/ZIP-MANAGEMENT-COPY.ZIP'
O IPADDR=&IPA_MIKNO IPPORT=&IPP_MIKNO CP=1141-E2N
//*
```

Abb. 22: Beispieljob für die Aktion ZIP,COPY z/OS

Im Beispiel werden unter Verwendung der Aktion 'ACTION=ZIP,COPY' Dateien aus einem ZIP-Archiv in ein anderes ZIP-Archiv kopiert.

Dazu wird mit den ersten beiden 'I'-Karten ein bestehendes ZIP-Archiv definiert, dessen kompletter Inhalt in das zu erstellende ZIP-Archiv aufgenommen werden soll.

Mit den sich daran anschließenden beiden 'I'-Karten wird unter Verwendung des vorangestellten Schlüsselwortes 'COPYARC=' ein ZIP-Archiv adressiert, aus dem eine Reihe von Dateien kopiert werden sollen.

Die zu kopierenden Dateien werden zZip mit der Filter Karte bekannt gegeben.

Die beiden 'O'-Karten definieren das zu erstellende, resultierende Archiv, das sowohl die Dateien aus dem ersten Archiv ('I ARCHIVE=') als auch die gefilterte Auswahl aus dem zweiten Archiv ('I COPYARC=') enthält.

Hinweis: Auf die Angabe des ersten ZIP-Archivs ('I ARCHIVE=') kann verzichtet werden. In diesem Falle werden lediglich die gefilterten Dateien aus dem Copy-Archiv ('I COPYARC=') in das neu erstellte ZIP-Archiv kopiert.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE mit vergleichbarem Ablauf.

```
* $$ JOB JNM=JZIPCOPY,DISP=D,PRI=3,CLASS=0
* $$ LST CLASS=T,JNM=JZIPCOPY
// JOB JZIPCOPY
// LIBDEF PHASE,SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005,SYSLST
// DLBL XPSHFS,'XPSHDRV.SPOOL.SAVE',,VSAM,CAT=XPSUCAT
// EXEC XPSZZIP,SIZE=AUTO,PARM='LIB=XPS.DAEMH600'
A ACTION=ZIP,COPY PROGRESS=Y
I ARCHIVE=NET='ZIPSTORE/HDRV-TEST-ZVSE.ZIP'
I IPADDR=&IPA_GEORG IPPORT=&IPP_GEORG
I COPYARC=NET='ZIPSTORE/ZIPOUT2.ZIP'
I IPADDR=&IPA_GEORG IPPORT=&IPP_GEORG
```

```
F INCLUDE=A6*
O ARCHIVE=NET= ' ZIPSTORE/ZIP-MANAGEMENT-COPY.ZIP '
O IPADDR=&IPA_GEORG IPPORT=&IPP_GEORG
/*
/&
* $$EQJ
```

Abb. 23: Beispieljob für die Aktion ZIP,COPY z/VSE

Um zZip über die Verwendung von ZIP-Archiven in Kenntnis zu setzen, ist den jeweiligen Ressourcentypen auf den Eingabe- und Ausgabekarten die Angabe 'ARCHIVE=' bzw. 'COPYRC=' voranzustellen.

Die folgende Tabelle zeigt die zulässigen Optionen für Karten des Typs 'A ACTION=ZIP,COPY'.

A ACTION=ZIP,COPY	
Kopieren von Dateien aus einem PKZIP-kompatiblen Archiv in ein anderes.	
Optionen	Beschreibung
<i>OPTION=</i>	'MERGE' 'REPLACE'
	Mit dieser Option kann gesteuert werden, wie zZip verfahren soll, falls Dateien in das neue ZIP-Archiv kopiert werden sollen, die bereits darin enthalten sind. <u>MERGE</u> In der Standardeinstellung werden nur Dateien in das neue Archiv kopiert, die noch nicht darin enthalten sind. REPLACE Bei Wahl dieser Einstellung werden Dateien, die bereits im neuen Archiv enthalten sind, durch Dateien mit gleichem Namen ersetzt, falls diese in der Liste der zu kopierenden Dateien enthalten sind.
<i>PROGRESS=</i>	'Y' 'N'
	Falls diese Option mit 'PROGRESS=Y' aktiviert wird, schreibt zZip pro Datei, die einem komprimierten Archiv hinzugefügt wird, einen Logeintrag.

ZIP,DEL - Dateien aus einem ZIP-Archiv löschen

Mit der Aktion 'ZIP,DEL' können Dateien aus einem ZIP-Archiv gelöscht werden.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JZIPDEL JOB , 'DEL FROM ZIP ARCHIV', CLASS=A, MSGCLASS=M
//ZIPDEL EXEC PGM=XPSZZIP, REGION=96M
//STEPLIB DD DISP=SHR, DSN=XPSHDRV.V300.LOADLIB
//SYSIN DD *
A ACTION=ZIP, DEL PROGRESS=Y
I ARCHIVE=NET= 'ZIPSTORE/ZIP-MANAGEMENT.ZIP '
I IPADDR=&IPA_MIKNO IPPORT=&IPP_MIKNO
F INCLUDE=XPSHDRV.V300.MACLIB( *)
O ARCHIVE=NET= 'ZIPSTORE/ZIP-MANAGEMENT-DEL.ZIP '
O IPADDR=&IPA_MIKNO IPPORT=&IPP_MIKNO
/*
```

Abb. 24: Beispieljob für die Aktion ZIP,DEL z/OS

Im Beispiel werden unter Verwendung der Aktion 'ACTION=ZIP,DEL' Dateien aus einem ZIP-Archiv gelöscht.

Dazu wird mit den ersten beiden 'I'-Karten ein bestehendes ZIP-Archiv definiert, aus dem einige Dateien gelöscht werden sollen. Die zu löschenden Dateien werden zZip mit der Filter Karte bekannt gegeben. Die beiden 'O'-Karten definieren das zu erstellende Archiv.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE mit vergleichbarem Ablauf.

```
* $$ JOB JNM=JZIPDEL, DISP=D, PRI=3, CLASS=0
* $$ LST CLASS=T, JNM=JZIPDEL
// JOB JZIPDEL
// LIBDEF PHASE, SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM= '00 '
// ASSGN SYS005, SYSLST
// DLBL XPSHFS, 'XPSHDRV.SPOOL.SAVE ', , VSAM, CAT=XPSUCAT
// EXEC XPSZZIP, SIZE=AUTO, PARM= 'LIB=XPS.DAEMH600 '
A ACTION=ZIP, DEL
I ARCHIVE=NET= 'ZIPSTORE/ZIP-MANAGEMENT-COPY.ZIP '
I IPADDR=&IPA_GEORG IPPORT=&IPP_GEORG
F INCLUDE=A69*
F INCLUDE=A6B*
O ARCHIVE=NET= 'ZIPSTORE/ZIP-MANAGEMENT-DEL.ZIP '
O IPADDR=&IPA_GEORG IPPORT=&IPP_GEORG
/*
/&
* $$EQJ
```

Abb. 25: Beispieljob für die Aktion ZIP,DEL z/VSE

Um zZip über die Verwendung von ZIP-Archiven in Kenntnis zu setzen, ist den jeweiligen Ressourcentypen auf den Eingabe- und Ausgabekarten die Angabe 'ARCHIVE=' voranzustellen.

Für Karten des Typs 'A ACTION=ZIP,DEL' stehen keine weiteren Optionen zur Verfügung.

ZIP,UPD - Dateien in einem ZIP-Archiv ändern

Mit der Aktion 'ZIP,UPD' können Dateien zu einem bestehenden ZIP-Archiv geändert werden.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JZIPUPD JOB , 'UPD TO ZIP ARCHIV' ,CLASS=A,MSGCLASS=M
//ZIPUPD EXEC PGM=XPSZZIP,REGION=96M
//STEPLIB DD DISP=SHR,DSN=XPSHDRV.V300.LOADLIB
//UPDSRC DD DISP=SHR,DSN=XPSHDRV.V300.MACLIB
//SYSIN DD *
A ACTION=ZIP,UPD PATH=Y PROGRESS=Y
I ARCHIVE=NET='ZIPSTORE/ZIP-MANAGEMENT.ZIP'
I IPADDR=&IPA_MIKNO IPPORT=&IPP_MIKNO
I LIB=UPDSRC MBR=*
F INCLUDE=ZIPT*
O ARCHIVE=NET='ZIPSTORE/ZIP-MANAGEMENT-UPD.ZIP'
O IPADDR=&IPA_MIKNO IPPORT=&IPP_MIKNO CP=1141-E2N
//*
```

Abb. 26: Beispieljob für die Aktion ZIP,UPD z/OS

Im Beispiel werden unter Verwendung der Aktion 'ACTION=ZIP,UPD' einige Dateien in einem bestehenden ZIP-Archiv verändert.

Zunächst wird das ZIP-Archiv mit den ersten beiden T-Karten definiert.

Mit der dritten T-Karte wird zZip darüber informiert, dass die veränderten Dateien, aus einer Bibliothek entnommen werden sollen, die durch das DD-Statement 'UPDSRC' identifiziert wird.

Die Dateien, die verändert werden sollen, werden zZip mit der Filterkarte 'F INCLUDE' bekannt gegeben.

Die beiden abschließenden 'O'-Karten legen fest, dass das resultierende ZIP-Archiv zur Speicherung im Netzwerk an einen HostDrive/J Server gesendet werden soll.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE mit vergleichbarem Ablauf.

```
* $$ JOB JNM=JZIPUPD,DISP=D,PRI=3,CLASS=0
* $$ LST CLASS=T,JNM=JZIPUPD
// JOB JZIPUPD
// LIBDEF PHASE,SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005,SYSLST
// DLBL XPSHFS,'XPSHDRV.SPOOL.SAVE',,VSAM,CAT=XPSUCAT
// EXEC XPSZZIP,SIZE=AUTO,PARM='LIB=XPS.DAEMH600'
A ACTION=ZIP,UPD PATH=N PROGRESS=N
I ARCHIVE=NET='ZIPSTORE/ZIP-MANAGEMENT-COPY.ZIP' TRACE=ALL
I IPADDR=&IPA_GEORG IPPORT=&IPP_GEORG
I LIB=XPS.BACKUP MBR=*
F INCLUDE=*RIND*
O ARCHIVE=NET='ZIPSTORE/ZIP-MANAGEMENT-UPD.ZIP'
O IPADDR=&IPA_GEORG IPPORT=&IPP_GEORG CP=1141-E2N
/*
/&
* $$EOJ
```





Abb. 27: Beispieljob für die Aktion ZIP,UPD z/VSE

Um zZip über die Verwendung von ZIP-Archiven in Kenntnis zu setzen, ist den jeweiligen Ressourcentypen auf den Eingabe- und Ausgabekarten die Angabe 'ARCHIVE=' voranzustellen.

Die folgende Tabelle zeigt die zulässigen Optionen für Karten des Typs 'A ACTION=ZIP,UPD'.

A ACTION=ZIP,UPD													
Ändern von Dateien in einem PKZIP-kompatiblen Archiv.													
Optionen	Beschreibung												
<i>COMMENT=</i>	Phrase												
	Mit dieser Option kann ein Kommentar für das ZIP-Archiv angegeben werden. Der Kommentar kann mit geeigneten Programmen zur Anzeige von ZIP-Archiven, wie z. B. WinZip, angezeigt werden.												
	Beispiel COMMENT='Backup der Umsätze für 2011/KW 21'												
<i>COMPRLVL=</i>	numerischer Wert zwischen '0' ... '9' 'STORE' 'SUPERFAST' 'FAST' <u>'NORMAL'</u> 'MAXIMUM'												
	Mit dieser Option ist der Komprimierungslevel bei der Erstellung komprimierter Archive festzulegen. Mit steigendem Wert verringert sich die Größe des erstellten Archivs auf Kosten der benötigten CPU Zeit. Neben der Angabe eines numerischen Wertes kann auch eine der vordefinierten Komprimierungsstufen gewählt werden:												
	<table border="1"> <thead> <tr> <th>vordefinierte Komprimierungsstufe</th> <th>entspricht numerischem Wert</th> </tr> </thead> <tbody> <tr> <td>STORE</td> <td>0</td> </tr> <tr> <td>SUPERFAST</td> <td>1</td> </tr> <tr> <td>FAST</td> <td>2</td> </tr> <tr> <td>NORMAL</td> <td>3</td> </tr> <tr> <td>MAXIMUM</td> <td>4</td> </tr> </tbody> </table>	vordefinierte Komprimierungsstufe	entspricht numerischem Wert	STORE	0	SUPERFAST	1	FAST	2	NORMAL	3	MAXIMUM	4
vordefinierte Komprimierungsstufe	entspricht numerischem Wert												
STORE	0												
SUPERFAST	1												
FAST	2												
NORMAL	3												
MAXIMUM	4												
<i>CRYPTOFACILITY=</i>	<u>'XPS'</u> 'ICSF' 'CPACF' * nur in der zZipSecure Edition verfügbar												
	Mit dieser Option ist festzulegen, auf welche Weise kryptographische Funktionen im Rahmen der Erstellung eines komprimierten Archivs ausgeführt werden sollen. Folgende Option können gewählt werden:												
	<p><u>XPS</u> Kryptographische Funktionen werden unter Verwendung der XPS CryptLib für IBM z-Systeme ausgeführt.</p> <p>Bei Nutzung der XPS CryptLib unter z/OS können rechenintensive kryptographische Berechnungen auf kostengünstige zIIP Prozessoren ausgelagert werden, falls diese präsent und aktiviert sind. Dazu ist die Option 'ZIIP=Y' in den allgemeinen Einstellungen für zZipSecure zu aktivieren – siehe Seite 35.</p> <p>ICSF Kryptographische Funktionen werden unter z/OS zur Ausführung an die installierte 'Integrated Cryptographic Service Facility' übermittelt.</p> <p>Damit wird erreicht, dass eine installierte Hardwarelösung genutzt wird.</p> <p>CPACF Kryptographische Funktionen werden unter z/OS und unter z/VSE zur Ausführung an den installierten 'Central Processor Assist for Cryptographic</p>												

	<p>Function' übermittelt.</p> <p>Damit wird erreicht, dass eine installierte Hardwarelösung genutzt wird.</p> <p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 24 beschrieben, global vorgelegt werden.</p> <p>Es ist zu beachten, dass einige der unterstützten Kryptomodi nur bei Verwendung einer bestimmten Cryptofacility genutzt werden können.</p>	
	Beispiel	CRYPTOFACILITY=CPACF
CRYPTOMODE=	'ECB' 'CBC' 'ECB_CTS' 'CBC_CTS' 'OFB' 'CTR' 'CFB' * nur in der zZipSecure Edition verfügbar	
	<p><u>Definition:</u> Ein kryptographischer Modus legt fest, wie sich die Verschlüsselung mehrerer Klartextblöcke vollzieht, indem er definiert, in welcher Art der gewählte Verschlüsselungsalgorithmus auf den Datenstrom angewandt wird.</p> <p>Mit dieser Option ist der Mode of Operation festzulegen, der bei der Verschlüsselung angewendet werden soll.</p> <p>Folgende Modi werden unterstützt:</p> <p>ECB Electronic codebook</p> <p>CBC Cipher-block chaining</p> <p>ECB_CTS ECB ciphertext stealing – nur bei Verwendung der XPS CryptLib</p> <p>CBC_CTS CBC ciphertext stealing – nur bei Verwendung der XPS CryptLib</p> <p>OFB Output Feedback Mode</p> <p>CTR Counter</p> <p>CFB Cipher feedback – nur bei Verwendung von CPACF</p> <p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 25 beschrieben, global vorgelegt werden.</p>	
	Beispiel	CRYPTOMODE=OFB
ENCRYPT=	Algorithmus * nur in der zZipSecure Edition verfügbar	
	<p><u>Definition:</u> Offen gelegtes Rechenverfahren, das festlegt, wie aus Blöcken von Klartext unter Verwendung eines geheimen Schlüssels Chiffre-Blöcke erzeugt werden, die nur bei Kenntnis des geheimen Schlüssels wieder in Klartext-Blöcke zurückverwandelt werden können.</p> <p>Mit dieser Option ist der Algorithmus zur Verschlüsselung der komprimierten Dateien zu wählen. Diese Option findet nur Berücksichtigung, wenn unter Verwendung der zuvor beschriebenen Option 'PWD=' ein Passwort für die Verschlüsselung angegeben wurde.</p> <p>Die nachfolgende Tabelle zeigt die unterstützten Algorithmen sowie die jeweils verwendete Schlüssellänge in Bit:</p>	
	Algorithmus	Beschreibung
	AE1	Advanced Encryption Standard / WinZip
	AE1,128	Advanced Encryption Standard / WinZip
	AE1,192	Advanced Encryption Standard / WinZip
	AE1,256	Advanced Encryption Standard / WinZip
	AE2	Advanced Encryption Standard / WinZip
		Schlüssellänge in Bit
		128
		128
		192
		256
		128

	AE2,128	Advanced Encryption Standard / WinZip	128
	AE2,192	Advanced Encryption Standard / WinZip	192
	AE2,256	Advanced Encryption Standard / WinZip	256
	AES	Advanced Encryption Standard	128
	AES,128	Advanced Encryption Standard	128
	AES,192	Advanced Encryption Standard	192
	AES,256	Advanced Encryption Standard	256
	DES	Data Encryption Standard	56
	DES,56	Data Encryption Standard	56
	DES,112	Data Encryption Standard	112
	DES,168	Data Encryption Standard	168
	3DES	Triple Data Encryption Standard	168
	RC2	Rivest Cipher 2	128
	RC2,40	Rivest Cipher 2	40
	RC2,64	Rivest Cipher 2	64
	RC2,128	Rivest Cipher 2	128
	RC4	Rivest Cipher 4	128
	RC4,40	Rivest Cipher 4	40
	RC4,64	Rivest Cipher 4	64
	RC4,128	Rivest Cipher 4	128
	BLOWFISH	Blockverschlüsselung / Bruce Schneier	128
	BLOWFISH,128	Blockverschlüsselung / Bruce Schneier	128
<p>Falls keine Angabe für den Algorithmus gemacht wird, werden die Daten gemäß dem traditionellen PKZIP Verfahren verschlüsselt.</p>			
	Beispiele:	ENCRYPT=AES,256 ENCRYPT=BLOWFISH	
PATH=	'Y' 'N'		
	<p>Bei der Erstellung von ZIP-Archiven aus Bibliotheken kann mit dieser Option darüber entschieden werden, ob zZip beim Erstellen des Archivs die Dateinamen voll qualifiziert in das Archiv aufnehmen soll oder nicht.</p> <p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 32 beschrieben, global vorbelegt werden.</p> <p>Die beiden nachfolgenden Abbildungen zeigen die verschiedenen Ergebnisse, je nach Wahl der Einstellung für diese Option. 'PATH=N' führt zu folgenden Dateinamen im Archiv:</p> <div style="border: 1px solid gray; padding: 5px; margin: 5px 0;"> <p>Name ▲</p> <ul style="list-style-type: none">  JALLOPDS  JESD2HFS </div> <p>'PATH=Y' führt zu folgendem Ergebnis:</p> <div style="border: 1px solid gray; padding: 5px; margin: 5px 0;"> <p>Name ▲</p> <ul style="list-style-type: none">  XPSHDRV.TEST.MACLIB(JALLOPDS)  XPSHDRV.TEST.MACLIB(JESD2HFS) </div>		

<i>PROGRESS=</i>	'Y' 'N'
	Falls diese Option mit 'PROGRESS=Y' aktiviert wird, schreibt zZip/zZipSecure pro Datei, die einem komprimierten Archiv hinzugefügt wird, einen Logeintrag.
<i>PWD=</i>	Passwort <i>* nur in der zZipSecure Edition verfügbar</i>
	<p>zZipSecure unterstützt das Standardverfahren zur Erstellung verschlüsselter ZIP-Archive. Bei diesem Verfahren werden die Daten auf Dateiebene, nachdem sie komprimiert wurden, zusätzlich verschlüsselt.</p> <p>Mit dieser Angabe ist das Passwort anzugeben, mit der die einzelnen Dateien des Archivs verschlüsselt werden sollen.</p> <p>Das hier angegebene Passwort muss einem Programm, das das Archiv zu einem späteren Zeitpunkt entkomprimieren soll, bekannt sein.</p> <p>Wie bei anderen sensiblen Informationen auch, empfiehlt sich die Verwendung einer Variablen für das Passwort, deren Wert in der Initialisierungsdatei 'XPSHDINI' hinterlegt ist. Damit erscheint das gewählte Passwort nicht im Jobprotokoll.</p> <p>Die Vorgehensweise zur Definition von Variablen ist auf Seite 35 beschrieben.</p>
Beispiel	PWD=&ZIP_PWD

ZIP,VIEW – Drucken des Inhaltsverzeichnisses eines ZIP-Archivs

Mit der Aktion 'ZIP,VIEW' kann das Inhaltsverzeichnis eines ZIP-Archivs in das Joblisting gedruckt werden.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JZIPVIEW JOB , 'VIEW ZIP ARCHIV',CLASS=A,MSGCLASS=M
//ZIPVIEW EXEC PGM=XPSZZIP,REGION=96M
//STEPLIB DD DISP=SHR,DSN=XPSHDRV.V300.LOADLIB
//SYSIN DD *
A ACTION=ZIP,VIEW
I ARCHIVE=NET= 'ZIPSTORE/ZIP-MANAGEMENT.ZIP'
I IPADDR=&IPA_MIKNO IPPORT=&IPP_MIKNO
//*
```

Abb. 28: Beispieljob für die Aktion ZIP,VIEW z/OS

Im Beispiel wird das Inhaltsverzeichnis eines im Netzwerk abgelegten ZIP-Archivs gedruckt.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE mit vergleichbarem Ablauf.

```
* $$ JOB JNM=JZIPVIEW,DISP=D,PRI=3,CLASS=0
* $$ LST CLASS=T,JNM=JZIPVIEW
// JOB JZIPVIEW
// LIBDEF PHASE,SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005,SYSLST
// DLBL XPSHFS, 'XPSHDRV.SPOOL.SAVE',,VSAM,CAT=XPSUCAT
// EXEC XPSZZIP,SIZE=AUTO,PARM='LIB=XPS.DAEMH600'
A ACTION=ZIP,VIEW
I ARCHIVE=NET='ZIPSTORE/ZIP-MANAGEMENT-COPY.ZIP'
I IPADDR=&IPA_GEORG IPPORT=&IPP_GEORG
/*
/&
* $$EOJ
```

Abb. 29: Beispieljob für die Aktion ZIP,VIEW z/VSE

Für Karten des Typs 'A ACTION=ZIP,VIEW' stehen keine weiteren Optionen zur Verfügung.

ENCRYPT/DECRYPT – Verschlüsseln und Entschlüsseln von Daten

Mit diesen Aktionen können Dateien verschlüsselt oder entschlüsselt werden. Diese Funktionalität steht nur in der zZipSecure Edition zur Verfügung.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JCRLI2NE JOB , 'CRYPT LIB MEM', CLASS=A, MSGCLASS=M
//CRLI2NE EXEC PGM=XPSZZIP, REGION=96M
//STEPLIB DD DISP=SHR, DSN=XPSHDRV.V300.LOADLIB
//XPSIN DD DISP=SHR, DSN=XPSHDRV.TEST.MACLIB
//SYSIN DD *
A ACTION=ENCRYPT
A CRYPTOFACILITY=XPS
A CRYPTOMETHOD=AES,128
A CRYPTOMODE=CBC
A CRYPTOIV=X'000102030405060708090A0B0C0D0E0F'
A CRYPTOKEY=X'000102030405060708090A0B0C0D0E0F'
I LIB=XPSIN MBR=X
O NET='CRYPTSTORE/CRYPTOUT.ENC' CP=1141-E2N
O IPADDR=&IPA_MIKNO IPPORT=&IPP_MIKNO
//*
//DCNE2LI EXEC PGM=XPSZZIP, REGION=96M
//STEPLIB DD DISP=SHR, DSN=XPSHDRV.V300.LOADLIB
//XPSOUT DD DISP=SHR, DSN=XPSHDRV.TEST.MACLIB
//SYSIN DD *
A ACTION=DECRYPT
A CRYPTOFACILITY=XPS
A CRYPTOMETHOD=AES,128
A CRYPTOMODE=CBC
A CRYPTOIV=X'000102030405060708090A0B0C0D0E0F'
A CRYPTOKEY=X'000102030405060708090A0B0C0D0E0F'
I NET='CRYPTSTORE/CRYPTOUT.ENC'
I IPADDR=&IPA_MIKNO IPPORT=&IPP_MIKNO
O LIB=XPSOUT MBR=XDEC CP=1141-N2E
//*
```

Abb. 30: Beispieljob für die Aktionen ENCRYPT und DECRYPT z/OS

Im Beispiel wird im ersten Jobstep ein Bibliotheksmitglied ('X') unter Verwendung der Aktion 'ACTION=ENCRYPT' verschlüsselt und dann an einen HostDrive/J Server zur Speicherung im Netzwerk gesendet.

Im zweiten Jobstep wird die zuvor erstellte verschlüsselte Datei wieder aus dem Netzwerk eingelesen und unter Verwendung der Aktion 'ACTION=DECRYPT' entschlüsselt und anschließend unter dem Namen 'XDEC' in dieselbe Bibliothek zurückgespielt, die auch das zuvor verschlüsselte Member enthielt.

Dabei findet vor der Verschlüsselung eine Übersetzung der Zeichendaten von EBCDIC nach ANSI und nach der Entschlüsselung eine Übersetzung der Zeichendaten von ANSI nach EBCDIC statt.

Es ist zu beachten, dass die Angaben der Zusatzoptionen, die das Verschlüsselungsergebnis beeinflussen, wie z. B. 'CRYPTOMETHOD=', für die Verschlüsselung und die Entschlüsselung exakt übereinstimmen müssen, um korrekte Ergebnisse zu erzielen.

Weiterhin ist es empfehlenswert, vertrauliche Informationen, wie z. B. Schlüssel, als Variablen in den Jobstream einzubinden, damit die tatsächlichen Werte nicht im Joblisting enthalten sind.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE mit vergleichbarem Ablauf.

```

* $$ JOB JNM=JCRLI2NE,DISP=D,PRI=3,CLASS=0
* $$ LST CLASS=T,JNM=JCRLI2NE
// JOB JCRLI2NE
// LIBDEF PHASE,SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005,SYSLST
// DLBL XPSHFS,'XPSHDRV.SPOOL.SAVE',,VSAM,CAT=XPSUCAT
// EXEC XPSZZIP,SIZE=AUTO,PARM='LIB=XPS.DAEMH600'
A ACTION=ENCRYPT
A CRYPTOFACILITY=XPS
A CRYPTOMETHOD=AES,128
A CRYPTOMODE=CBC
A CRYPTOIV=X'000102030405060708090A0B0C0D0E0F'
A CRYPTOKEY=X'000102030405060708090A0B0C0D0E0F'
I LIB=XPS.HDRV300 MBR=X.D
O NET='CRYPTSTORE/CRYPTOUT.ENC' CP=1141-E2N
O IPADDR=127.0.0.1 IPPORT=8840 ROUTE=WS08
/*
// LIBDEF PHASE,SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005,SYSLST
// DLBL XPSHFS,'XPSHDRV.SPOOL.SAVE',,VSAM,CAT=XPSUCAT
// EXEC XPSZZIP,SIZE=AUTO,PARM='LIB=XPS.DAEMH600'
A ACTION=DECRYPT
A CRYPTOFACILITY=XPS
A CRYPTOMETHOD=AES,128
A CRYPTOMODE=CBC
A CRYPTOIV=X'000102030405060708090A0B0C0D0E0F'
A CRYPTOKEY=X'000102030405060708090A0B0C0D0E0F'
I NET='CRYPTSTORE/CRYPTOUT.ENC'
I IPADDR=127.0.0.1 IPPORT=8840 ROUTE=WS08
O LIB=XPS.HDRV300 MBR=XDEC.D CP=1141-N2E TRACE=ALL
/*
/&
* $$EOJ

```

Abb. 31: Beispieljob für die Aktionen ENCRYPT und DECRYPT z/VSE

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Karten des Typs 'A ACTION=ENCRYPT'.

A ACTION=ENCRYPT	
* nur in der zZipSecure Edition verfügbar	
Verschlüsseln von Daten.	
Optionen	Beschreibung
CRYPTOFACILITY=	'XPS' 'ICSF' 'CPACF'
	<p>Mit dieser Option ist festzulegen, auf welche Weise die Verschlüsselung ausgeführt werden sollen.</p> <p>Folgende Option können gewählt werden:</p> <p>XPS Die Verschlüsselung wird unter Verwendung der XPS CryptLib für IBM z-Systeme ausgeführt.</p> <p>Bei Nutzung der XPS CryptLib unter z/OS können rechenintensive kryptographische Berechnungen auf kostengünstige zIIP Prozessoren ausgelagert werden, falls diese präsent und aktiviert sind. Dazu ist die Option 'ZIIP=Y' in den allgemeinen Einstellungen für zZipSecure zu aktivieren – siehe Seite 35.</p>

	<p>ICSF Die Ausführung der Verschlüsselung wird unter z/OS an die installierte 'Integrated Cryptographic Service Facility' übermittelt. Damit wird erreicht, dass eine installierte Hardwarelösung genutzt wird.</p> <p>CPACF Die Ausführung der Verschlüsselung wird unter z/OS und unter z/VSE an den installierten 'Central Processor Assist for Cryptographic Function' übermittelt. Damit wird erreicht, dass eine installierte Hardwarelösung genutzt wird.</p> <p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 24 beschrieben, global vorbelegt werden.</p> <p>Es ist zu beachten, dass einige der unterstützten Kryptomodi nur bei Verwendung einer bestimmten Cryptofacility genutzt werden können.</p>								
	<table border="1"> <tr> <td data-bbox="352 707 475 766">Beispiel</td> <td data-bbox="475 707 1380 766">CRYPTOFACILITY=XPS</td> </tr> </table>	Beispiel	CRYPTOFACILITY=XPS						
Beispiel	CRYPTOFACILITY=XPS								
CRYPTOIV=	hexadezimaler Wert (X'...') Zeichenkette								
	<p><u>Definition:</u> Meist zufällig bestimmter Startwert, der von manchen Betriebsmodi (z. B. CBC) eines Block-Chiffre vor dem ersten Datenblock bearbeitet wird.</p> <p>Mit dieser Option kann der Initialisierungsvektor vorbelegt werden, der für symmetrische Verschlüsselung in den entsprechenden Modi (z. B. CBC, OFB, CFB) erforderlich ist.</p> <p>Der Initialisierungsvektor muss immer genau so groß wie die Blocklänge des Chiffre sein:</p> <table border="0"> <tr> <td>DES</td> <td>8 Bytes</td> </tr> <tr> <td>RC2</td> <td>8 Bytes</td> </tr> <tr> <td>BLOWFISH</td> <td>8 Bytes</td> </tr> <tr> <td>AES</td> <td>16 Bytes</td> </tr> </table> <p>Der Vektor ist entweder hexadezimal mit 2 Stellen pro Zeichen oder als Zeichenkette anzugeben, die bei Bedarf mit Leerzeichen aufgefüllt wird.</p> <p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 25 beschrieben, global vorbelegt werden.</p> <p>Bei der Wahl des Initialisierungsvektors ist zu berücksichtigen, dass die Sicherheit kryptographischer Funktionen in unterschiedlicher Weise von der Zufälligkeit des Initialisierungsvektors abhängt.</p>	DES	8 Bytes	RC2	8 Bytes	BLOWFISH	8 Bytes	AES	16 Bytes
DES	8 Bytes								
RC2	8 Bytes								
BLOWFISH	8 Bytes								
AES	16 Bytes								
	<table border="1"> <tr> <td data-bbox="352 1559 608 1617">Beispiel (8 Bytes)</td> <td data-bbox="608 1559 1380 1617">CRYPTOIV=X'32FC06A51DB94E87' CRYPTOIV=LEZDANCE</td> </tr> </table>	Beispiel (8 Bytes)	CRYPTOIV=X'32FC06A51DB94E87' CRYPTOIV=LEZDANCE						
Beispiel (8 Bytes)	CRYPTOIV=X'32FC06A51DB94E87' CRYPTOIV=LEZDANCE								
CRYPTOKEY=	hexadezimaler Wert (X'...') Zeichenkette								
	<p><u>Definition:</u> Schlüssel, der bei symmetrischen Verfahren sowohl zur Verschlüsselung als auch zur Entschlüsselung eingesetzt wird.</p> <p>Mit dieser Option kann der kryptographische Schlüssel angegeben werden, der bei der symmetrischen Verschlüsselung zum Einsatz kommen soll.</p> <p>Die Länge des Schlüssels hängt zum einen vom verwendeten Algorithmus und zum anderen von der gewünschten Verschlüsselungsstärke ab. Weitere Informationen sind der Option 'CRYPTOMETHOD=' auf Seite 95 zu entnehmen.</p> <p>Der Schlüssel ist entweder hexadezimal mit 2 Stellen pro Zeichen oder als Zeichenkette anzugeben, die bei Bedarf mit Leerzeichen aufgefüllt wird.</p> <p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 26 beschrieben, global vorbelegt werden.</p>								

	Beispiel für DES (56 Bit/7 Bytes)	CRYPTOKEY=X'A827C149DFE623' CRYPTOKEY=GEHEIM																																																												
CRYPTOMETHOD=	Algorithmus																																																													
	<p><u>Definition:</u> Offen gelegtes Rechenverfahren, das festlegt, wie aus Blöcken von Klartext unter Verwendung eines geheimen Schlüssels Chiffre-Blöcke erzeugt werden, die nur bei Kenntnis des geheimen Schlüssels wieder in Klartext-Blöcke zurückverwandelt werden können.</p> <p>Mit dieser Option ist der Algorithmus zu wählen, mit dem die Daten verschlüsselt werden sollen.</p> <p>Die nachfolgende Tabelle zeigt die unterstützten Algorithmen sowie die jeweils verwendete Schlüssellänge in Bit:</p> <table border="1"> <thead> <tr> <th>Algorithmus</th> <th>Beschreibung</th> <th>Schlüssellänge in Bit</th> </tr> </thead> <tbody> <tr><td>AES</td><td>Advanced Encryption Standard</td><td>128</td></tr> <tr><td>AES,128</td><td>Advanced Encryption Standard</td><td>128</td></tr> <tr><td>AES,192</td><td>Advanced Encryption Standard</td><td>192</td></tr> <tr><td>AES,256</td><td>Advanced Encryption Standard</td><td>256</td></tr> <tr><td>DES</td><td>Data Encryption Standard</td><td>56</td></tr> <tr><td>DES,56</td><td>Data Encryption Standard</td><td>56</td></tr> <tr><td>DES,112</td><td>Data Encryption Standard</td><td>112</td></tr> <tr><td>DES,168</td><td>Data Encryption Standard</td><td>168</td></tr> <tr><td>3DES</td><td>Triple Data Encryption Standard</td><td>168</td></tr> <tr><td>RC2</td><td>Rivest Cipher 2</td><td>128</td></tr> <tr><td>RC2,40</td><td>Rivest Cipher 2</td><td>40</td></tr> <tr><td>RC2,64</td><td>Rivest Cipher 2</td><td>64</td></tr> <tr><td>RC2,128</td><td>Rivest Cipher 2</td><td>128</td></tr> <tr><td>RC4</td><td>Rivest Cipher 4</td><td>128</td></tr> <tr><td>RC4,40</td><td>Rivest Cipher 4</td><td>40</td></tr> <tr><td>RC4,64</td><td>Rivest Cipher 4</td><td>64</td></tr> <tr><td>RC4,128</td><td>Rivest Cipher 4</td><td>128</td></tr> <tr><td>BLOWFISH</td><td>Blockverschlüsselung / Bruce Schneier</td><td>128</td></tr> <tr><td>BLOWFISH,128</td><td>Blockverschlüsselung / Bruce Schneier</td><td>128</td></tr> </tbody> </table> <p>Falls keine Angabe für den Algorithmus gemacht wird, wird die in der Initialisierungsdatei festgelegte Voreinstellung für die Option 'CRYPTOMETHOD' verwendet – siehe Seite 24.</p> <p>Beispiele: CRYPTOMETHOD=3DES CRYPTOMETHOD=RC4,128</p>		Algorithmus	Beschreibung	Schlüssellänge in Bit	AES	Advanced Encryption Standard	128	AES,128	Advanced Encryption Standard	128	AES,192	Advanced Encryption Standard	192	AES,256	Advanced Encryption Standard	256	DES	Data Encryption Standard	56	DES,56	Data Encryption Standard	56	DES,112	Data Encryption Standard	112	DES,168	Data Encryption Standard	168	3DES	Triple Data Encryption Standard	168	RC2	Rivest Cipher 2	128	RC2,40	Rivest Cipher 2	40	RC2,64	Rivest Cipher 2	64	RC2,128	Rivest Cipher 2	128	RC4	Rivest Cipher 4	128	RC4,40	Rivest Cipher 4	40	RC4,64	Rivest Cipher 4	64	RC4,128	Rivest Cipher 4	128	BLOWFISH	Blockverschlüsselung / Bruce Schneier	128	BLOWFISH,128	Blockverschlüsselung / Bruce Schneier	128
Algorithmus	Beschreibung	Schlüssellänge in Bit																																																												
AES	Advanced Encryption Standard	128																																																												
AES,128	Advanced Encryption Standard	128																																																												
AES,192	Advanced Encryption Standard	192																																																												
AES,256	Advanced Encryption Standard	256																																																												
DES	Data Encryption Standard	56																																																												
DES,56	Data Encryption Standard	56																																																												
DES,112	Data Encryption Standard	112																																																												
DES,168	Data Encryption Standard	168																																																												
3DES	Triple Data Encryption Standard	168																																																												
RC2	Rivest Cipher 2	128																																																												
RC2,40	Rivest Cipher 2	40																																																												
RC2,64	Rivest Cipher 2	64																																																												
RC2,128	Rivest Cipher 2	128																																																												
RC4	Rivest Cipher 4	128																																																												
RC4,40	Rivest Cipher 4	40																																																												
RC4,64	Rivest Cipher 4	64																																																												
RC4,128	Rivest Cipher 4	128																																																												
BLOWFISH	Blockverschlüsselung / Bruce Schneier	128																																																												
BLOWFISH,128	Blockverschlüsselung / Bruce Schneier	128																																																												
CRYPTOMODE=	'ECB' 'CBC' 'ECB_CTS' 'CBC_CTS' 'OFB' 'CTR' 'CFB'																																																													
	<p><u>Definition:</u> Ein kryptographischer Modus legt fest, wie sich die Verschlüsselung mehrerer Klartextblöcke vollzieht, indem er definiert, in welcher Art der gewählte Verschlüsselungsalgorithmus auf den Datenstrom angewandt wird.</p> <p>Mit dieser Option ist der Mode of Operation festzulegen, der bei der Verschlüsselung angewendet werden soll.</p> <p>Folgende Modi werden unterstützt:</p>																																																													

ECB	Electronic codebook
CBC	Cipher-block chaining
ECB_CTS	ECB ciphertext stealing – nur bei Verwendung der XPS CryptLib
CBC_CTS	CBC ciphertext stealing – nur bei Verwendung der XPS CryptLib
OFB	Output Feedback Mode
CTR	Counter
CFB	Cipher feedback – nur bei Verwendung von CPACF
Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 25 beschrieben, global vorgelegt werden.	
Beispiel	CRYPTOMODE=CFB (nur bei CRYPTOFACILITY=CPACF) CRYPTOMODE=ECB_CTS (nur bei CRYPTOFACILITY=XPS) CRYPTOMODE=ECB

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Karten des Typs 'A ACTION=DECRYPT'.

A ACTION=DECRYPT <i>* nur in der zZipSecure Edition verfügbar</i>	
Entschlüsseln von Daten.	
Optionen	Beschreibung
CRYPTOFACILITY=	'XPS' 'ICSF' 'CPACF'
	<p>Mit dieser Option ist festzulegen, auf welche Weise die Entschlüsselung ausgeführt werden sollen.</p> <p>Folgende Option können gewählt werden:</p> <p>XPS Die Entschlüsselung wird unter Verwendung der XPS CryptLib für IBM z-Systeme ausgeführt.</p> <p>Bei Nutzung der XPS CryptLib unter z/OS können rechenintensive kryptographische Berechnungen auf kostengünstige zIIP Prozessoren ausgelagert werden, falls diese präsent und aktiviert sind. Dazu ist die Option 'ZIIP=Y' in den allgemeinen Einstellungen für zZipSecure zu aktivieren – siehe Seite 35.</p> <p>ICSF Die Ausführung der Entschlüsselung wird unter z/OS an die installierte 'Integrated Cryptographic Service Facility' übermittelt.</p> <p>Damit wird erreicht, dass eine installierte Hardwarelösung genutzt wird.</p> <p>CPACF Die Ausführung der Entschlüsselung wird unter z/OS und unter z/VSE an den installierten 'Central Processor Assist for Cryptographic Function' übermittelt.</p> <p>Damit wird erreicht, dass eine installierte Hardwarelösung genutzt wird.</p> <p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 24 beschrieben, global vorgelegt werden.</p>
Beispiel	CRYPTOFACILITY=XPS
CRYPTOIV=	hexadezimaler Wert (X'....') Zeichenkette

	<p>Definition: Meist zufällig bestimmter Startwert, der von manchen Betriebsmodi (z. B. CBC) eines Block-Chiffre vor dem ersten Datenblock bearbeitet wird.</p> <p>Mit dieser Option kann der Initialisierungsvektor vorbelegt werden, der für symmetrische Entschlüsselung in den entsprechenden Modi (z. B. CBC, OFB, CFB) erforderlich ist.</p> <p>Der Initialisierungsvektor muss immer genau so groß wie die Blocklänge des Chiffre sein:</p> <table border="0" data-bbox="475 416 810 595"> <tr> <td>DES</td> <td>8 Bytes</td> </tr> <tr> <td>RC2</td> <td>8 Bytes</td> </tr> <tr> <td>BLOWFISH</td> <td>8 Bytes</td> </tr> <tr> <td>AES</td> <td>16 Bytes</td> </tr> </table> <p>Der Vektor ist entweder hexadezimal mit 2 Stellen pro Zeichen oder als Zeichenkette anzugeben, die bei Bedarf mit Leerzeichen aufgefüllt wird.</p> <p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 25 beschrieben, global vorbelegt werden.</p> <p>Der zum Entschlüsseln verwendete Initialisierungsvektor muss exakt mit dem Vektor übereinstimmen, der bei der Verschlüsselung der Daten zum Einsatz kam.</p>		DES	8 Bytes	RC2	8 Bytes	BLOWFISH	8 Bytes	AES	16 Bytes
DES	8 Bytes									
RC2	8 Bytes									
BLOWFISH	8 Bytes									
AES	16 Bytes									
	<p>Beispiel (8 Bytes)</p>	<p>CRYPTOIV=X'32FC06A51DB94E87' CRYPTOIV=LETSDANZ</p>								
<p>CRYPTOKEY=</p>	<p>hexadezimaler Wert (X'...') Zeichenkette</p>									
	<p>Definition: Schlüssel, der bei symmetrischen Verfahren sowohl zur Verschlüsselung als auch zur Entschlüsselung eingesetzt wird.</p> <p>Mit dieser Option kann der kryptographische Schlüssel angegeben werden, der bei der symmetrischen Entschlüsselung zum Einsatz kommen soll.</p> <p>Die Länge des Schlüssels hängt zum einen vom verwendeten Algorithmus und zum anderen von der gewünschten Verschlüsselungsstärke ab. Weitere Informationen sind der Option '<i>CRYPTOMETHOD</i>' auf Seite 97 zu entnehmen.</p> <p>Der Schlüssel ist entweder hexadezimal mit 2 Stellen pro Zeichen oder als Zeichenkette anzugeben, die bei Bedarf mit Leerzeichen aufgefüllt wird.</p> <p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 26 beschrieben, global vorbelegt werden.</p> <p>Der hier angegebene Wert muss exakt dem Schlüssel entsprechen, der verwendet wurde, um die Daten zu verschlüsseln.</p>									
	<p>Beispiel für DES (56 Bit/7 Bytes)</p>	<p>CRYPTOKEY=X'A827C149DFE623' CRYPTOKEY=BEKANNT</p>								
<p>CRYPTOMETHOD=</p>	<p>Algorithmus</p>									
	<p>Definition: Offen gelegtes Rechenverfahren, das festlegt, wie aus Blöcken von Klartext unter Verwendung eines geheimen Schlüssels Chiffre-Blöcke erzeugt werden, die nur bei Kenntnis des geheimen Schlüssels wieder in Klartext-Blöcke zurückverwandelt werden können.</p> <p>Mit dieser Option ist der Algorithmus anzugeben, unter dessen Verwendung die Daten entschlüsselt werden sollen.</p> <p>Damit die verschlüsselten Daten korrekt entschlüsselt werden können, ist exakt derselbe Algorithmus zu wählen, der auch bei der Verschlüsselung der Daten angewendet wurde.</p> <p>Die nachfolgende Tabelle zeigt die unterstützten Algorithmen sowie die jeweils</p>									

	verwendete Schlüssellänge in Bit:		
	Algorithmus	Beschreibung	Schlüssellänge in Bit
	AES	Advanced Encryption Standard	128
	AES,128	Advanced Encryption Standard	128
	AES,192	Advanced Encryption Standard	192
	AES,256	Advanced Encryption Standard	256
	DES	Data Encryption Standard	56
	DES,56	Data Encryption Standard	56
	DES,112	Data Encryption Standard	112
	DES,168	Data Encryption Standard	168
	3DES	Triple Data Encryption Standard	168
	RC2	Rivest Cipher 2	128
	RC2,40	Rivest Cipher 2	40
	RC2,64	Rivest Cipher 2	64
	RC2,128	Rivest Cipher 2	128
	RC4	Rivest Cipher 4	128
	RC4,40	Rivest Cipher 4	40
	RC4,64	Rivest Cipher 4	64
	RC4,128	Rivest Cipher 4	128
	BLOWFISH	Blockverschlüsselung / Bruce Schneier	128
BLOWFISH,128	Blockverschlüsselung / Bruce Schneier	128	
Falls keine Angabe für den Algorithmus gemacht wird, wird die in der Initialisierungsdatei festgelegte Voreinstellung für die Option ' <i>CRYPTOMETHOD</i> ' verwendet – siehe Seite 24.			
Beispiele:	CRYPTOMETHOD=AES CRYPTOMETHOD=DES,168		
<i>CRYPTOMODE=</i>	'ECB' 'CBC' 'ECB_CTS' 'CBC_CTS' 'OFB' 'CTR' 'CFB'		
	<p>Definition: Ein kryptographischer Modus legt fest, wie sich die Verschlüsselung mehrerer Klartextblöcke vollzieht, indem er definiert, in welcher Art der gewählte Verschlüsselungsalgorithmus auf den Datenstrom angewandt wird.</p> <p>Mit dieser Option ist der Mode of Operation festzulegen, der bei der Entschlüsselung angewendet werden soll. Um ein korrektes Ergebnis zu erzielen, ist exakt derselbe Modus zu wählen, der auch bei der Verschlüsselung der Daten angewendet wurde.</p> <p>Folgende Modi werden unterstützt:</p> <p>ECB Electronic codebook</p> <p>CBC Cipher-block chaining</p> <p>ECB_CTS ECB ciphertext stealing – nur bei Verwendung der XPS CryptLib</p> <p>CBC_CTS CBC ciphertext stealing – nur bei Verwendung der XPS CryptLib</p> <p>OFB Output Feedback Mode</p> <p>CTR Counter</p>		

	CFB Cipher feedback – nur bei Verwendung von CPACF	
	Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 25 beschrieben, global vorgelegt werden.	
Beispiel	CRYPTOMODE=CFB	(nur bei CRYPTOFACILITY=CPACF)
	CRYPTOMODE=ECB_CTS	(nur bei CRYPTOFACILITY=XPS)
	CRYPTOMODE=ECB	

Programmausführungen

Neben der Nutzung für File Transfers kann zZipSecure auch zur Ausführung verschiedener Programme eingesetzt werden.

Die verfügbaren Möglichkeiten werden nachfolgend beschrieben.

CMD – Ausführen eines Programmes unter der Kontrolle von HostDrive/J

Mit dieser Aktion kann die Ausführung eines Programmes unter der Kontrolle eines HostDrive/J Servers veranlasst werden.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JCMDXBAT JOB , 'EXEC WINDOWS BAT', CLASS=E, MSGCLASS=M
//CMDXBAT EXEC PGM=XPSZZIP, REGION=96M
//STEPLIB DD DSN=XPSHDRV.V300.LOADLIB, DISP=SHR
//SYSOUT DD SYSOUT=*
//SYSIN DD *
X CMD=J:\HOSTDRIVE\UMSATZ.BAT
X PARM=UMSATZ
X PARM=DEZEMBER
X PARM=2011
X PARM=C2UST84
X IPADDR=127.0.0.1 IPPORT=8840 ROUTE=MIKNO
//*
```

Abb. 32: Beispieljob für die Ausführung eines Netzwerkprogrammes - z/OS

Im Beispiel wird die Ausführung des Windows Batchprogrammes 'J:\HOSTDRIVE\UMSATZ.BAT' auf einem HostDrive/J Server veranlasst. Dem Batchprogramm werden 4 Parameter übergeben.

Der Start des Programmes erfolgt unter Zwischenschaltung einer HostDrive/z Installation, die den Port 8840 auf dem lokalen Host überwacht, und die Anfrage an den mit der Route-Angabe definierten HostDrive/J Server weiterleitet.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE mit vergleichbarem Ablauf.

```
* $$ JOB JNM=JCMDXBAT, DISP=D, PRI=3, CLASS=0
* $$ LST CLASS=T
// JOB JCMDXBAT
// LIBDEF PHASE, SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005, SYSLST
// DLBL XPSHFS, 'XPSHDRV.SPOOL.SAVE', , VSAM, CAT=XPSUCAT
// EXEC XPSZZIP, SIZE=AUTO, PARM='LIB=XPS.DAEMH600'
X CMD=D:\HOSTDRIVE\ROUTEPARMS.BAT WAIT=YES
X PARM=UMSATZ
```

```
X PARM=DEZEMBER
X PARM=2011
X PARM=C2UST84
X IPADDR=127.0.0.1 IPPORT=8840 ROUTE=MIKNO
/*
/&
* $$EOJ
```

Abb. 33: Beispieljob für die Ausführung eines Netzwerkprogrammes - z/VSE

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Karten des Typs 'X CMD=PFAD'.

X CMD=											
Ausführen eines Programmes unter der Kontrolle eines HostDrive/J Servers.											
Optionen	Beschreibung										
<i>CMD=</i>	Pfadangabe zum ausführbaren Programm										
	Die Angabe muss den voll qualifizierten Pfad zum auszuführenden Programm enthalten. Bezüglich des Typs des auszuführenden Programmes bestehen keine Einschränkungen. Falls die Pfadangabe eines oder mehrere Leerzeichen enthalten sollte, ist die Angabe in einfache Anführungszeichen zu setzen.										
	Beispiel <code>CMD='C:\Programme\Microsoft Office\Office12\EXCEL.EXE'</code>										
<i>DISP=</i>	'L' 'D' '(C,CHANNEL[,PATTERN])' '(M,CHANNEL[,PATTERN])'										
	Mit dieser Angabe dann der HostDrive/J Server darüber informiert werden, wie mit Dateien verfahren werden soll, die vom gestarteten Programm zur weiteren Verarbeitung an den HostDrive/J Server übergeben werden. Die nachfolgende Tabelle zeigt die gültigen Dispositionen:										
	<table border="1"> <thead> <tr> <th>Disposition</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>L – LEAVE</td> <td>HostDrive/J unternimmt keine Aktion für die verarbeiteten Dateien.</td> </tr> <tr> <td>D – DELETE</td> <td>Die Dateien werden nach der Weiterverarbeitung durch HostDrive/J gelöscht.</td> </tr> <tr> <td>C – COPY</td> <td>Die Dateien werden nach der Weiterverarbeitung durch HostDrive/J kopiert.</td> </tr> <tr> <td>M – MOVE</td> <td>Die Dateien werden nach der Weiterverarbeitung durch HostDrive/J verschoben.</td> </tr> </tbody> </table>	Disposition	Beschreibung	L – LEAVE	HostDrive/J unternimmt keine Aktion für die verarbeiteten Dateien.	D – DELETE	Die Dateien werden nach der Weiterverarbeitung durch HostDrive/J gelöscht.	C – COPY	Die Dateien werden nach der Weiterverarbeitung durch HostDrive/J kopiert.	M – MOVE	Die Dateien werden nach der Weiterverarbeitung durch HostDrive/J verschoben.
Disposition	Beschreibung										
L – LEAVE	HostDrive/J unternimmt keine Aktion für die verarbeiteten Dateien.										
D – DELETE	Die Dateien werden nach der Weiterverarbeitung durch HostDrive/J gelöscht.										
C – COPY	Die Dateien werden nach der Weiterverarbeitung durch HostDrive/J kopiert.										
M – MOVE	Die Dateien werden nach der Weiterverarbeitung durch HostDrive/J verschoben.										
	Bei Wahl einer der Dispositionen Kopieren oder Verschieben ist als zweiter Parameter die Angabe des Namens eines Mapping-Kanals erforderlich, in den die Kopie der Datei erstellt werden soll. Als optionale Angabe kann bei diesen beiden Dispositionen noch ein Umbenennschema angegeben werden, mit dem der Name der Dateikopie bestimmt werden kann. Nähere Angaben hierzu sind dem HostDrive/J Handbuch zu entnehmen.										
<i>IPADDR=</i>	TCP/IP Adresse – numerisch oder als DNS-Name										

	<p>Mit dieser Option ist die TCP/IP Adresse anzugeben, über die zZip die Ausführung des Programmes veranlassen soll.</p> <p>Diese Angabe kann sich entweder direkt auf einen HostDrive/J beziehen oder auf eine HostDrive/z Basisinstallation. Im zweiten Fall ist die Angabe einer 'ROUTE=' Option erforderlich, mit der dem HostDrive/z Server mitgeteilt wird, über welche TCP/IP Adresse der HostDrive/J Server erreichbar ist.</p> <p>Falls keine Angabe für diese Option erfolgt, verwendet zZip die TCP/IP Adresse, die als Vorbelegung in der Initialisierungsdatei – siehe Seite 27 - hinterlegt ist.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>IPADDR=127.0.0.1</td> </tr> </table>	Beispiel	IPADDR=127.0.0.1
Beispiel	IPADDR=127.0.0.1		
IPPORT=	numerischer TCP/IP Port <= 65535		
	<p>Mit dieser Option ist der TCP/IP Port anzugeben, der von dem - mit 'IPADDR=' adressierten - HostDrive Server für eingehende Anfragen überwacht wird.</p> <p>Falls keine Angabe für diese Option erfolgt, verwendet zZip den TCP/IP Port, der als Vorbelegung in der Initialisierungsdatei – siehe Seite 28 - hinterlegt ist.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>IPPORT=8841</td> </tr> </table>	Beispiel	IPPORT=8841
Beispiel	IPPORT=8841		
PARM=	Zeichenkette		
	<p>Mit dieser Karte, die beliebig oft angegeben werden kann, können Parameter spezifiziert werden, die an das auszuführende Programm übergeben werden sollen.</p> <p>Falls die Angabe für einen Parameter eines oder mehrere Leerzeichen enthalten sollte, ist die Angabe in einfache Anführungszeichen zu setzen.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>PARM='PRODUKTGRUPPE 4711'</td> </tr> </table>	Beispiel	PARM='PRODUKTGRUPPE 4711'
Beispiel	PARM='PRODUKTGRUPPE 4711'		
ROUTE=	Schlüssel einer HostDrive/z Route Definition		
	<p>Mit dieser Option ist der Name einer HostDrive/z Route Definition anzugeben, in der weitere TCP/IP Informationen über einen angeschlossenen HostDrive/J Server hinterlegt sind.</p> <p>Diese Angabe ist nur erforderlich, wenn zZip die Programmausführung auf dem HostDrive/J Server nicht direkt veranlassen soll, sondern indirekt über einen HostDrive/z Server.</p> <p>Definition und Verwaltung von HostDrive/z Routen sind im HostDrive/z Handbuch beschrieben.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>ROUTE=WS08</td> </tr> </table>	Beispiel	ROUTE=WS08
Beispiel	ROUTE=WS08		
WAIT=	'Y' 'N'		
	<p>Mit dieser Angabe ist zZip darüber zu informieren, ob auf die Übermittlung des Ausführungsergebnisses des zu startenden Programmes gewartet werden soll oder nicht.</p> <p>Falls die Angabe 'WAIT=Y' erfolgt, steht der zZip Job so lange, bis das gestartete Programm beendet wird.</p> <p>Damit kann auf einfache Weise eine Synchronisierung erreicht werden, so dass z. B. Datenbestände, die vom gestarteten Programm bereitgestellt und von nachfolgenden Jobsteps verarbeitet werden, zur Verfügung stehen, sobald zZip die Verarbeitung fortsetzt.</p>		

WFD – Verarbeitung von Dateien mit dem Status 'waiting for delivery'

Mit dieser Aktion kann die Verarbeitung von Dateien mit dem WFD-Status (waiting for delivery) veranlasst werden.

Dazu überprüft zZip alle Dateien, die sich auf der angebotenen XPS HFS-Datei befinden und versucht alle Dateien, die den WFD-Status besitzen, erneut zuzustellen.

Die Möglichkeit zur Speicherung und späteren erneuten Zustellung von WFD-Dateien ist nur beim Versenden der Dateien über eine angebotene HostDrive/z Basisinstallation möglich.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JCMDXWFD JOB , 'EXEC WFD' ,CLASS=A,MSGCLASS=M
//CMDXWFD EXEC PGM=XPSZZIP,REGION=48M
//STEPLIB DD DISP=SHR,DSN=XPSHDRV.V300.LOADLIB
//XPSDATA DD DSN=XPSHDRV.V300.MACLIB,DISP=SHR
//XPSIN DD DISP=SHR,DSN=XPSHDRV.V300.MACLIB
//SYSIN DD *
X WFD
/*
```

Abb. 34: Beispieljob für die Veranlassung der WFD-Verarbeitung - z/OS

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE mit vergleichbarem Ablauf.

```
* $$ JOB JNM=JCMDXWFD,DISP=D,PRI=3,CLASS=0
* $$ LST CLASS=T
// JOB JCMDXWFD
// LIBDEF PHASE,SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005,SYSLST
// DLBL XPSHFS,'XPSHDRV.SPOOL.SAVE',,VSAM,CAT=XPSUCAT
// EXEC XPSZZIP,SIZE=AUTO,PARM='LIB=XPS.DAEMH600'
X WFD
/*
/&
* $$EQJ
```

Abb. 35: Beispieljob für die Veranlassung der WFD-Verarbeitung - z/VSE

Für die Verarbeitung von WFD-Dateien stehen keine weiteren Optionen zur Verfügung.

Eingabedaten

Unter Verwendung einer oder mehrerer 'I'-Karten werden die Eingabedaten beschrieben, die von zZip verarbeitet werden sollen.

Pro Eingabedatei bzw. Eingabequelle ist eine eigene 'I'-Karte anzugeben, die sich, bei Bedarf oder falls gewünscht, über mehrere Zeilen erstrecken kann.

Das Eintragen von Fortsetzungszeichen auf Stelle 72 ist nicht erforderlich, jedoch ist auf jeder Karte das 'I'-Kennzeichen auf der ersten Stelle einzufügen, wie der nachfolgenden Abbildung zu entnehmen ist:

```
...
I NET='CEETEMP/SAPDK.DAT' CP=EBC-CRLF FNAME=FILE1
I FTP=' /ARCHIVE/SAPDK.BAK '
I IPADDR=192.168.0.110 IPPORT=21
I USER='user' PWD='password'
O ...
...
```

Abb. 36: Beispiel für Eingabekarten

Wie das Beispiel zeigt, werden zwei Eingabedateien definiert, was durch die beiden Schlüsselworte 'NET=' und 'FTP=' angezeigt wird.

Weiterhin ist ersichtlich, dass sich die Beschreibung der zweiten Eingabequelle über drei aufeinanderfolgende Zeilen erstreckt, jeweils beginnend mit 'I', jedoch ohne die Verwendung eines Fortsetzungszeichens.

Pro zZip Jobstep können mehrere Eingabedateien definiert werden.

Im Rahmen der Verarbeitung analysiert zZip zunächst die Definition aller Eingabedateien und stellt diese den nachfolgenden Funktionen zur Erstellung von Ausgabedaten zur Verfügung.

Die folgende Tabelle zeigt eine kurze Übersicht der verfügbaren Quellen für die Bereitstellung von Eingabedaten:

Quelle	Beschreibung
ARCHIVE	Lesen der Daten aus einem PKZIP-kompatiblen Archiv.
IPT	Bereitstellung der Daten im Job-Control.
LIB	Lesen der Daten aus einer Bibliothek.
NET	Lesen der Daten von einem Netzwerklaufwerk.
SEQ	Lesen der Daten aus einer sequentiellen Datei.
TAPE	Lesen der Daten von Band.
ESDS	Lesen der Daten aus einer VSAM ESDS Datei.
KSDS	Lesen der Daten aus einer VSAM KSDS Datei.
XPSHFS	Lesen der Daten vom XPS HFS-Dateisystem.

FTP	Lesen der Daten von einem FTP-Server.
HFS	Lesen der Daten aus dem OpenMVS HFS-Dateisystem - nur z/OS.
JES	Lesen der Daten aus dem z/OS Spoolsystem.
POWER	Lesen der Daten aus dem z/VSE Spoolsystem.

Es folgt eine Beschreibung der unterstützten Quellen für Eingabedaten.

ARCHIVE - Lesen der Daten aus einem PKZIP-kompatiblen Archiv

zZip kann PKZIP-kompatible Archive verarbeiten.

Um zZip darüber zu informieren, dass ein PKZIP-kompatibles Archiv verarbeitet werden soll, ist dem Typschlüssel auf der jeweiligen 'I'-Karte das Schlüsselwort 'ARCHIVE=' voranzustellen.

Beispiel:

Der nachfolgende Auszug aus den Kontrollkarten zeigt die Verwendung einer Textdatei aus dem Netzwerk – Typschlüssel 'NET=':

```
...
I NET= 'BACKUP/RECHNUNGEN/MUSTER20111222.TXT'
...
```

Falls anstatt der Textdatei ein PKZIP-kompatibles Archiv verarbeitet werden soll, wäre die 'I'-Karte folgendermaßen anzupassen (normalerweise ändert sich der Dateiname):

```
...
I ARCHIVE=NET= 'BACKUP/RECHNUNGEN/ALLE20111222.ZIP'
...
```

An Hand der Angabe von 'I ARCHIVE=NET=' anstelle von 'I NET=' erkennt zZip nun, dass ein komprimiertes ZIP-Archiv verarbeitet werden soll und führt daher zusätzliche erforderliche Arbeitsschritte, wie z. B. bei Bedarf das Entkomprimieren der Daten, aus.

Die tatsächlich durchzuführende Aktion wird unter Verwendung einer Karte vom Typ 'ACTION=UNZIP' festgelegt. Eine Beschreibung erfolgt in Kapitel 'ZIP/UNZIP – Komprimieren/Entkomprimieren von Archiven' auf Seite 57.

Die nachfolgende Tabelle zeigt die zusätzlichen Optionen für Eingabekarten an, die bei der Verarbeitung von ZIP-Archiven verwendet werden können.

I ARCHIVE=...=	
Verarbeiten von PKZIP-kompatiblen Archiven.	
Optionen	Beschreibung
CPA2E=	Schlüssel einer Codepage

	<p>Ergänzende Textinformationen wie z. B. Kommentare oder Dateinamen werden in ZIP-Archiven auf Grund von Portabilitätsgründen immer in ASCII/ANSI kodiert.</p> <p>Mit dieser Option kann festgelegt werden, welche Codepage zur Übersetzung dieser Textdaten beim Lesen von ZIP-Archiven verwendet werden soll.</p> <p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 23 beschrieben, global vorbelegt werden.</p> <p>Falls unter z/OS auf eine Angabe für diese Option verzichtet wird, werden die ergänzenden Textdaten unter Verwendung der im System eingestellten Codepage übersetzt, die bei Ausführung des Systemmakros 'XLATE' für die Übersetzung von ASCII/ANSI nach EBCDIC zum Einsatz kommt.</p>	
	Beispiel	CPA2E=1141-N2E
CPE2A=	Schlüssel einer Codepage	
	<p>Ergänzende Textinformationen wie z. B. Kommentare oder Dateinamen werden in ZIP-Archiven auf Grund von Portabilitätsgründen immer in ASCII/ANSI kodiert.</p> <p>Mit dieser Option kann festgelegt werden, welche Codepage zur Übersetzung dieser Textdaten beim Schreiben von ZIP-Archiven verwendet werden soll.</p> <p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 23 beschrieben, global vorbelegt werden.</p> <p>Falls unter z/OS auf eine Angabe für diese Option verzichtet wird, werden die ergänzenden Textdaten unter Verwendung der im System eingestellten Codepage übersetzt, die bei Ausführung des Systemmakros 'XLATE' für die Übersetzung von EBCDIC nach ASCII/ANSI zum Einsatz kommt.</p>	
	Beispiel	CPE2A=1141-E2N

IPT - Bereitstellung der Daten im Job-Control

Mit diesem Steuerkartentyp wird zZip darüber informiert, dass Eingabedaten direkt im Job-Control bereitgestellt werden.

Dazu sind die Eingabedaten im Anschluss an die Kontrollkarten für zZip in einem eigenen SYSIN (z/OS) bzw. SYSIPT (z/VSE) aufzulisten.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JIPT2NET JOB , 'IPT TO NET',CLASS=A,MSGCLASS=M
//JIPT2NET EXEC PGM=XPSZZIP,REGION=48M
//STEPLIB DD DSN=XPSHDRV.V300.LOADLIB,DISP=SHR
//XPSDATA DD DSN=XPSHDRV.V300.MACLIB,DISP=SHR
//SYSIN DD *
I IPT RECORD=ARTISTEN/KOMPONISTEN
O NET='CEETEMP/komponisten.csv' CP=1141-E2N CRLF WTO=Y
O IPADDR=192.168.0.131 IPPORT=8841 CSV=; CSVHDR=Y
//SYSIN2 DD *
JOHANN SEBASTIAN BACH EISENACH 1685-1750
WOLFGANG AMADEUS MOZART SALZBURG 1756-1791
LUDWIG VAN BEETHOVEN BONN 1770-1827
FRANZ SCHUBERT WIEN 1797-1828
//*
```

Abb. 37: Beispieljob für IPT Eingabe z/OS

Die Auflistung der Eingabedaten erfolgt im Anschluss an die '//SYSIN2 DD *' Karte, mit der der neue SYSIN Bereich eröffnet wird.

Im gezeigten Beispiel wird zZip über die zusätzlichen Optionen auf der I-Karte darüber informiert, dass die zu verarbeitenden Daten als Tabelle zu interpretieren ist, die gemäß einer definierten Satzstruktur aufgebaut ist.

Eine Beschreibung der Verwendung von Satzstrukturen ist Kapitel 'Satzstrukturen' auf Seite 186 zu entnehmen.

Die Ausgabe der IPT-Daten erfolgt in eine Datei auf einem Netzwerkverzeichnis, das von einem HostDrive/J Server verwaltet wird, der durch die TCP/IP Definition auf der O-Karte adressiert wird.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE:

```
* $$ JOB JNM=JIPT2NET,DISP=D,PRI=3,CLASS=0
* $$ LST CLASS=T,JNM=JIPT2NET
// JOB JIPT2NET
// LIBDEF PHASE,SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005,SYSLST
// DLBL XPSHFS,'XPSHDRV.SPOOL.SAVE',,VSAM,CAT=XPSUCAT
// EXEC XPSZZIP,SIZE=AUTO,PARAM='LIB=XPS.DAEMH600'
I IPT
O NET='CEETEMP/KOMPONISTEN.TXT' CP=1141-E2N CRLF WTO=Y
O IPADDR=192.168.0.110 IPPORT=8880
/*
JOHANN SEBASTIAN BACH EISENACH 1685-1750
WOLFGANG AMADEUS MOZART SALZBURG 1756-1791
LUDWIG VAN BEETHOVEN BONN 1770-1827
FRANZ SCHUBERT WIEN 1797-1828
/*
```

```

/&
* $$EOJ

```

Abb. 38: Beispieljob für IPT Eingabe z/VSE

Die Auflistung der Eingabedaten erfolgt im Anschluss an die erste '/'* Karte, mit der der aktuell verwendete SYSIPT Bereich geschlossen wird.

Die Ausgabe der IPT-Daten erfolgt in eine Datei auf einem Netzwerkverzeichnis, das von einem HostDrive/J Server verwaltet wird, der durch die TCP/IP Definition auf der O-Karte adressiert wird.

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Eingabekarten des Typs 'I IPT'.

I IPT	
Lesen von 80-stelligen Daten aus dem Job Control.	
Optionen	Beschreibung
<i>RECORD=</i>	Verweis auf eine Satzstruktur auf der XPS HFS-Datei
	Mit dieser Option kann bekannt gegeben werden, dass das Format der Eingabedaten durch die angegebene Struktur beschrieben wird. Damit wird die Verarbeitung der Daten auf Feldebene ermöglicht.
Beispiel	RECORD=ARTISTS/KOMPONISTEN
<i>TRACE=</i>	'ALL' 'D' 'T' 'S' 'Z' oder eine, jeweils durch Komma getrennte, Kombination
	Mit dieser Option können Traceoptionen für den Job festgelegt werden. Der Trace dient zur Ausgabe von Informationen, die bei Bedarf die Analyse des internen Ablaufs ermöglichen. Traceausgaben erfolgen direkt in das Job-Listing. Die folgende Tabelle enthält die verfügbaren Tracetypen:
Typ	Beschreibung
<i>ALL</i>	Alle Tracetypen sind aktiviert.
<i>D</i>	Der Datentrace wird aktiviert.
<i>T</i>	Der TCP/IP Trace wird aktiviert.
<i>S</i>	Der Speichertrace wird aktiviert.
<i>Z</i>	Der ZIP/UNZIP Trace wird aktiviert.
	Mehrere Tracetypen können aktiviert werden, in dem diese, durch Komma getrennt, aufgelistet werden.
Beispiel	TRACE=D,T
<i>WFD=</i>	'Y' 'N'
	Mit dieser Option ist festzulegen, ob zZip eine Sicherungskopie der Eingabedaten auf der verbundenen XPS HFS-Datei erstellen soll. Diese Datei wird in der JCL durch das DD/DLBL Statement 'XPSHFS' identifiziert. Falls diese Angabe in der JCL nicht erfolgt, wird die Vorbelegung für 'XPSHFS' aus der Initialisierungsdatei, siehe Seite 34, übernommen. Diese Option wird nur dann berücksichtigt, wenn die Ausgabedaten über einen HostDrive/z Server zugestellt werden. In diesem Fall kann die wfd-Verarbeitung des HostDrive Queuing Systems genutzt werden. Die wfd-Verarbeitung (waiting for delivery) ermöglicht die automatische Zustellung von Dateien an das gewünschte Ziel zu einem späteren Zeitpunkt, falls dieses zur Ausführungszeit des zZip Jobs nicht erreicht werden kann.

WTO=	'Y' 'N'
	Mit dieser Option ist festzulegen, ob zZip beim Jobstart eine Nachricht über die verwendete Version und beim Jobende eine Nachricht über die Anzahl der verarbeiteten Sätze auf die Konsole ausgeben soll oder nicht. Mit einer Angabe kann die globale Voreinstellung – siehe Seite 34 - überschrieben werden.

LIB - Lesen der Daten aus einer Bibliothek

Mit diesem Steuerkartentyp wird zZip darüber informiert, dass Eingabedaten, die in Bibliotheksmitgliedern gespeichert sind (z/OS: Maclib, z/VSE: Library), verarbeitet werden sollen.

Bibliotheken, die zu verarbeitende Member enthalten, können zum Einen über die JCL als DD-Statement (z/OS) bzw. als DLBL-Statement (z/VSE) bereitgestellt werden. In diesem Fall ist das entsprechende JCL Statement in der 'I LIB=' Karte zu referenzieren. Zum Anderen besteht die Möglichkeit, die Bibliothek voll qualifiziert direkt in der 'I LIB=' Karte anzugeben.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JLIB2ESD JOB , 'LIB TO ESDS' , CLASS=A, MSGCLASS=M
//JESDS EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE (XPSHDRV.ESDS) CL PURGE
DEFINE CL(NAME(XPSHDRV.ESDS) -
        NONINDEXED -
        REUSE -
        SHR(2) -
        VOL(XPS006) -
        RECSZ(80 80) -
        CISZ(4096) -
        REC(1000))
//*
//LIB2ESD EXEC PGM=XPSZZIP, REGION=48M
//STEPLIB DD DISP=SHR, DSN=XPSHDRV.V300.LOADLIB
//XPSDATA DD DSN=XPSHDRV.V300.MACLIB, DISP=SHR
//XPSIN DD DISP=SHR, DSN=XPS.ARTS.MACLIB
//ESDSOUT DD DISP=SHR, DSN=XPSHDRV.ESDS
//SYSIN DD *
I LIB=XPSIN MBR=PAINTERS
O ESDS=ESDSOUT
//*
```

Abb. 39: Beispieljob für LIB Eingabe z/OS

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE:

```
* $$ JOB JNM=JLIB2ESD, DISP=D, PRI=3, CLASS=0
* $$ LST CLASS=T, PROGR=IDCAMS-DEFINE
// JOB JLIB2ESD
// EXEC IDCAMS, SIZE=AUTO
DELETE (XPS.HDRV.ESDS) CL PURGE -
        CATALOG(XPS.USER.CATALOG)
DEFINE CL(NAME(XPS.HDRV.ESDS) -
        VOL(SYSWK2) -
        RECSZ(80 80) -
        CISZ(4096) -
        NONINDEXED -
        REUSE -
        SHR(2) -
        REC(5000)) -
        DATA(NAME(XPS.HDRV.ESDS.DATA)) -
        CATALOG(XPS.USER.CATALOG)
/*
// LIBDEF PHASE, SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005, FEE
```


	<i>ALL</i>	Alle Tracetypen sind aktiviert.
	<i>D</i>	Der Datentrace wird aktiviert.
	<i>T</i>	Der TCP/IP Trace wird aktiviert.
	<i>S</i>	Der Speichertrace wird aktiviert.
	<i>Z</i>	Der ZIP/UNZIP Trace wird aktiviert.
	Mehrere Tracetypen können aktiviert werden, in dem diese, durch Komma getrennt, aufgelistet werden.	
	Beispiel	TRACE=D,T
<i>WFD=</i>	'Y' 'N'	
	<p>Mit dieser Option ist festzulegen, ob zZip eine Sicherungskopie der Eingabedaten auf der verbundenen XPS HFS-Datei erstellen soll.</p> <p>Diese Datei wird in der JCL durch das DD/DLBL Statement 'XPSHFS' identifiziert. Falls diese Angabe in der JCL nicht erfolgt, wird die Vorbelegung für 'XPSHFS' aus der Initialisierungsdatei, siehe Seite 34, übernommen.</p> <p>Diese Option wird nur dann berücksichtigt, wenn die Ausgabedaten über einen HostDrive/z Server zugestellt werden.</p> <p>In diesem Fall kann die wfd-Verarbeitung des HostDrive Queuing Systems genutzt werden. Die wfd-Verarbeitung (waiting for delivery) ermöglicht die automatische Zustellung von Dateien an das gewünschte Ziel zu einem späteren Zeitpunkt, falls dieses zur Ausführungszeit des zZip Jobs nicht erreicht werden kann.</p>	
<i>WTO=</i>	'Y' 'N'	
	<p>Mit dieser Option ist festzulegen, ob zZip beim Jobstart eine Nachricht über die verwendete Version und beim Jobende eine Nachricht über die Anzahl der verarbeiteten Sätze auf die Konsole ausgeben soll oder nicht. Mit einer Angabe kann die globale Voreinstellung – siehe Seite 34 - überschrieben werden.</p>	

NET – Lesen der Daten von einem Netzwerklaufwerk

Mit diesem Steuerkartentyp wird zZip darüber informiert, dass Eingabedaten von einem Netzwerklaufwerk gelesen werden sollen.

Damit zZip auf Netzwerkdateien zugreifen kann, ist es erforderlich, dass die Netzwerkkomponente des HostDrive Systems, der HostDrive/J Server, installiert ist und auf die gewünschten Daten zugreifen kann. Die Installation und Konfiguration des HostDrive/J Servers ist in einem eigenen Handbuch dokumentiert.

Für die Kommunikation mit dem HostDrive/J Server stehen zZip zwei verschiedene Möglichkeiten zur Verfügung:

1. Direkte Kommunikation.
2. Indirekte Kommunikation über eine HostDrive/z Basisinstallation.

Die Unterscheidung erfolgt über die Angabe einer 'ROUTE=' Option auf der 'I NET=' Karte. Wenn diese Angabe fehlt, kommuniziert zZip direkt mit dem HostDrive/J Server.

Wenn die 'ROUTE=' Option angegeben ist, erfolgt die Kommunikation indirekt über eine HostDrive/z Basisinstallation. In diesem Fall verwendet die HostDrive/z Basisinstallation die angegebene Route zur Identifikation des HostDrive/J Servers, der die Eingabedaten liefern soll.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JNET2HFS JOB , 'NET TO HFS', CLASS=A, MSGCLASS=M
//NET2HFS EXEC PGM=XPSZZIP, REGION=48M
//STEPLIB DD DISP=SHR, DSN=XPSHDRV.V300.LOADLIB
//XPSDATA DD DSN=XPSHDRV.V300.MACLIB, DISP=SHR
//SYSIN DD *
I NET='GRAPHICS/overlay.jpg' IPADDR=192.168.0.131 IPPORT=8841
O HFS=' /usr/common/'
/*
```

Abb. 41: Beispieljob für NET Eingabe z/OS

Im Beispiel wird die Datei 'overlay.jpg' direkt aus dem Netzwerk gelesen (keine 'ROUTE=' Option) und unter Verwendung der 'O HFS=' Karte in das OpenMVS HFS Dateisystem in das Verzeichnis 'usr/common/' mit identischem Namen kopiert.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE:

```
* $$ JOB JNM=JNET2LIB, DISP=D, PRI=3, CLASS=0
* $$ LST CLASS=T
// JOB JNET2LIB
// LIBDEF PHASE, SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005, SYSLST
// DLBL XPSHFS, 'XPSHDRV.SPOOL.SAVE', , VSAM, CAT=XPSUCAT
// EXEC XPSZZIP, SIZE=AUTO, PARM='LIB=XPS.DAEMH600'
I NET='CEETEMP/KOMPONISTEN.TXT' CP=1141-N2E RECL=80
I IPADDR=127.0.0.1 IPPORT=8840 ROUTE=WS08
O LIB=XPS.BACKUP MBR=KOMP.TXT
/*
/&
* $$EQJ
```


Abb. 42: Beispieljob für NET Eingabe zVSE

In diesem Beispiel wird die Datei 'OVERLAY.JPG' indirekt über den HostDrive/z Server mit der angegebenen TCP/IP Definition gelesen, was durch die Angabe der 'ROUTE=WS08' Option ersichtlich ist. Die Ausgabe der Graphik erfolgt in eine VSE Bibliothek, die mit der Karte 'O LIB=' näher definiert ist.

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Eingabekarten des Typs 'I NET='.

I NET=	
Lesen von Daten aus einer Netzwerkdatei.	
Optionen	Beschreibung
CP=	Codepage
	Mit dieser Option ist zZip über die zu verwendende Codepage zu informieren, falls Textdaten verarbeitet werden sollen. Weitere Informationen zu Codepages sind dem Kapitel 'Codepages' auf Seite 13 zu entnehmen. Eine Tabelle der verfügbaren Codepages befindet sich auf Seite 15.
Beispiel	CP=1141-N2E
IPADDR=	TCP/IP Adresse – numerisch oder als DNS-Name
	Mit dieser Option ist die TCP/IP Adresse anzugeben, von der zZip die Eingabedatei beziehen soll. Diese Angabe kann sich entweder direkt auf einen HostDrive/J beziehen oder auf eine HostDrive/z Basisinstallation. Im zweiten Fall ist die Angabe einer 'ROUTE=' Option erforderlich, mit der dem HostDrive/z Server mitgeteilt wird, über welche TCP/IP Adresse der HostDrive/J Server erreichbar ist. Falls keine Angabe für diese Option erfolgt, verwendet zZip die TCP/IP Adresse, die als Vorbelegung in der Initialisierungsdatei – siehe Seite 27 - hinterlegt ist.
Beispiel	IPADDR=127.0.0.1
IPPORT=	numerischer TCP/IP Port <= 65535
	Mit dieser Option ist der TCP/IP Port anzugeben, der von dem - mit 'IPADDR=' adressierten – HostDrive Server für eingehende Anfragen überwacht wird. Falls keine Angabe für diese Option erfolgt, verwendet zZip den TCP/IP Port, der als Vorbelegung in der Initialisierungsdatei – siehe Seite 28 - hinterlegt ist.
Beispiel	IPPORT=8841
NET=	Pfadangabe zur Datei im HostDrive/J Format
	Die Pfadangabe muss mit der Angabe des Namens eines gültigen Mapping-Kanals auf dem adressierten HostDrive/J Server beginnen. Mapping-Kanäle dienen als symbolische Platzhalter für physische Pfade im Netzwerk. Nähere Informationen hierzu sind dem HostDrive Handbuch zu entnehmen.
Beispiel	NET=MAPPINGCHANNEL/SUBDIR/FILE.EXT
RECL=	numerischer Wert

	<p>Mit dieser Option kann festgelegt werden, dass zZip aus Text-Eingabedaten Sätze mit festgelegter Länge generieren soll.</p> <p>Dazu werden die Eingabedaten zeilenweise gelesen und Zeilen mit kleinerer Länge als die hier angegebene werden mit dem in den Vorbelegungen definierten Fülleichen – siehe 'PADC=' auf Seite 32 – bis zur gewünschten Länge aufgefüllt.</p> <p>Zeilen, die länger als der hier angegebene Wert sind, werden in mehrere Zeilen der gewünschten Länge umgebrochen.</p>												
	<table border="1"> <tr> <td>Beispiel</td> <td>RECL=132</td> </tr> </table>	Beispiel	RECL=132										
Beispiel	RECL=132												
RECORD=	Verweis auf eine Satzstruktur auf der XPS HFS-Datei												
	<p>Mit dieser Option kann bekannt gegeben werden, dass das Format der Eingabedaten durch die angegebene Struktur beschrieben wird. Damit wird die Verarbeitung der Daten auf Feldebene ermöglicht.</p>												
	<table border="1"> <tr> <td>Beispiel</td> <td>RECORD=ARTISTS/KOMPONISTEN</td> </tr> </table>	Beispiel	RECORD=ARTISTS/KOMPONISTEN										
Beispiel	RECORD=ARTISTS/KOMPONISTEN												
ROUTE=	Schlüssel einer HostDrive/z Route Definition												
	<p>Mit dieser Option ist der Name einer HostDrive/z Route Definition anzugeben, in der weitere TCP/IP Informationen über einen angeschlossenen HostDrive/J Server hinterlegt sind.</p> <p>Diese Angabe ist nur erforderlich, wenn zZip die Netzwerkdatei nicht direkt von einem HostDrive/J Server lesen soll, sondern indirekt über einen HostDrive/z Server.</p> <p>Definition und Verwaltung von HostDrive/z Routen sind im HostDrive/z Handbuch beschrieben.</p>												
	<table border="1"> <tr> <td>Beispiel</td> <td>ROUTE=WS08</td> </tr> </table>	Beispiel	ROUTE=WS08										
Beispiel	ROUTE=WS08												
TRACE=	'ALL' 'D' 'T' 'S' 'Z' oder eine, jeweils durch Komma getrennte, Kombination												
	<p>Mit dieser Option können Traceoptionen für den Job festgelegt werden. Der Trace dient zur Ausgabe von Informationen, die bei Bedarf die Analyse des internen Ablaufs ermöglichen. Traceausgaben erfolgen direkt in das Job-Listing. Die folgende Tabelle enthält die verfügbaren Tracetypen:</p> <table border="1"> <thead> <tr> <th>Typ</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>ALL</td> <td>Alle Tracetypen sind aktiviert.</td> </tr> <tr> <td>D</td> <td>Der Datentrace wird aktiviert.</td> </tr> <tr> <td>T</td> <td>Der TCP/IP Trace wird aktiviert.</td> </tr> <tr> <td>S</td> <td>Der Speichertrace wird aktiviert.</td> </tr> <tr> <td>Z</td> <td>Der ZIP/UNZIP Trace wird aktiviert.</td> </tr> </tbody> </table> <p>Mehrere Tracetypen können aktiviert werden, in dem diese, durch Komma getrennt, aufgelistet werden.</p>	Typ	Beschreibung	ALL	Alle Tracetypen sind aktiviert.	D	Der Datentrace wird aktiviert.	T	Der TCP/IP Trace wird aktiviert.	S	Der Speichertrace wird aktiviert.	Z	Der ZIP/UNZIP Trace wird aktiviert.
Typ	Beschreibung												
ALL	Alle Tracetypen sind aktiviert.												
D	Der Datentrace wird aktiviert.												
T	Der TCP/IP Trace wird aktiviert.												
S	Der Speichertrace wird aktiviert.												
Z	Der ZIP/UNZIP Trace wird aktiviert.												
	<table border="1"> <tr> <td>Beispiel</td> <td>TRACE=D,T</td> </tr> </table>	Beispiel	TRACE=D,T										
Beispiel	TRACE=D,T												
WTO=	'Y' 'N'												
	<p>Mit dieser Option ist festzulegen, ob zZip beim Jobstart eine Nachricht über die verwendete Version und beim Jobende eine Nachricht über die Anzahl der verarbeiteten Sätze auf die Konsole ausgeben soll oder nicht. Mit einer Angabe kann die globale Voreinstellung – siehe Seite 34 - überschrieben werden.</p>												

SEQ – Lesen der Daten aus einer sequentiellen Datei

Mit diesem Steuerkartentyp wird zZip darüber informiert, dass Eingabedaten aus einer sequentiellen Datei gelesen werden sollen.

Sequentielle Dateien, die zu verarbeitende Daten enthalten, können zum Einen über die JCL als DD-Statement (z/OS) bzw. als DLBL-Statement (z/VSE) bereitgestellt werden. In diesem Fall ist das entsprechende JCL Statement in der 'I SEQ=' Karte zu referenzieren. Zum Anderen besteht die Möglichkeit, die sequentielle Datei voll qualifiziert direkt in der 'I SEQ=' Karte anzugeben.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JSEQ2FTP JOB , 'SEQ TO FTP' , CLASS=A, MSGCLASS=M
//LIB2SEQ EXEC PGM=XPSZZIP, REGION=48M
//STEPLIB DD DSN=XPSHDRV.V300.LOADLIB, DISP=SHR
//XPSIN DD DISP=SHR, DSN=XPSDAEM.V500.MACLIB
//XPSSEQ DD DSN=XPSHDRV.SEQ, DISP=(NEW,CATLG),
// SPACE=(TRK,(5,3),RLSE), UNIT=3390, VOL=SER=XPS005,
// DCB=(LRECL=100, BLKSIZE=1000, RECFM=FB, DSORG=PS)
//SYSIN DD *
I LIB=XPSIN MBR=XPSVP01
O SEQ=XPSSEQ
//*
//SEQ2FTP EXEC PGM=XPSZZIP, REGION=48M
//STEPLIB DD DISP=SHR, DSN=XPSHDRV.V300.LOADLIB
//XPSDATA DD DSN=XPSHDRV.V300.MACLIB, DISP=SHR
//XPSIN DD DSN=XPSHDRV.SEQ, DISP=(OLD,DELETE)
//SYSIN DD *
I SEQ=XPSIN RECL=100
O FTP='TEST/XPSVP01.TXT' CP=1141-E2N
O IPADDR=192.168.0.131 IPPORT=21
O USER='user' PWD='pwd'
/*
```

Abb. 43: Beispieljob für SEQ Eingabe z/OS

Im Beispiel wird zunächst eine sequentielle Datei erstellt, in die dann unter Verwendung der Steuerkarten 'I LIB=' und 'O SEQ=' ein Maclib-Member kopiert wird. Im letzten Schritt dient die erstellte sequentielle Datei dann als Eingabemedium und wird über FTP versendet.

Hinweis: Die Aufgabe 'Versenden eines Bibliotheksmembers über FTP' kann auch direkt, d. h. ohne die Verwendung einer sequentiellen Datei als Zwischenspeicher, erfolgen. Die Verwendung der sequentiellen Datei dient lediglich Anschauungszwecken.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE, der denselben Verarbeitungsprozess durchläuft, wie der zuvor abgebildete Beispieljob für z/OS:

```
* $$ JOB JNM=JSEQ2FTP, DISP=D, PRI=3, CLASS=0
* $$ LST CLASS=T
// JOB JSEQ2FTP
// LIBDEF PHASE, SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005, FEE
// DLBL XPSSEQ, 'HDRV.SEQ.FILE', , VSAM, DISP=(NEW,KEEP), X
// RECORDS=(500,50), RECSIZE=80
// DLBL XPSHFS, 'XPSHDRV.SPOOL.SAVE', , VSAM, CAT=XPSUCAT
// EXEC XPSZZIP, SIZE=AUTO, PARM='LIB=XPS.DAEMH600'
I LIB=XPS.DAEMV500, MBR=INSTALL.PROC
O SEQ=XPSSEQ RECL=80
```

```

/*
// LIBDEF PHASE,SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005,FEE
// DLBL XPSSEQ,'HDRV.SEQ.FILE',,VSAM,DISP=(OLD,DELETE)
// DLBL XPSHFS,'XPSHDRV.SPOOL.SAVE',,VSAM,CAT=XPSUCAT
// EXEC XPSZZIP,SIZE=AUTO,PARM='LIB=XPS.DAEMH600'
I SEQ=XPSSEQ RECL=80
O FTP='/TEST/XPSVP01.TXT' CP=1141-E2N
O IPADDR=192.168.0.110 IPPORT=21
O USER=&USERID PWD=&PASSWORD
/*
/&
* $$SEQ
    
```

Abb. 44: Beispieljob für SEQ Eingabe z/VSE

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Eingabekarten des Typs 'I SEQ='.

I SEQ=	
Lesen von Daten aus einer sequentiellen Datei.	
Optionen	Beschreibung
BLKSIZE=	numerischer Wert
	Mit dieser Option ist zZip über die Blocksize zu informieren, mit der die sequentielle Datei erstellt wurde. Diese Angabe ist unter z/VSE immer erforderlich und unter z/OS nur dann, wenn die Information nicht über das DD-Statement für die sequentielle Datei ermittelbar ist.
	Beispiel BLKSIZE=8200
RECFM=	'F' 'FB' 'U' 'V' 'VB'
	Mit dieser Angabe ist das Recordformat der sequentiellen Datei zu bestimmen. Diese Angabe ist unter z/VSE immer erforderlich, wenn sie nicht dem Standard 'RECFM=FB' entspricht. Unter z/OS ist die Angabe nur dann erforderlich, wenn zZip das Recordformat nicht automatisch ermitteln kann. Die anzugebenden Werte entsprechen den Originalwerten der Job Control: 'F' fixed 'FB' fixed blocked 'U' undefined 'V' variable 'VB' variable blocked
RECL=	numerischer Wert
	Mit dieser Option ist die Satzlänge der Daten auf der sequentiellen Datei anzugeben. Diese Angabe ist unter z/VSE immer und unter z/OS nur dann erforderlich, wenn zZip die Satzlänge nicht automatisch ermitteln kann.

	Beispiel	RECL=80
RECORD=	Verweis auf eine Satzstruktur auf der XPS HFS-Datei	
	Mit dieser Option kann bekannt gegeben werden, dass das Format der Eingabedaten durch die angegebene Struktur beschrieben wird. Damit wird die Verarbeitung der Daten auf Feldebene ermöglicht.	
	Beispiel	RECORD=ARTISTS/KOMPONISTEN
SEQ=	z/OS: Verweis auf das JCL DD-Statement der sequentiellen Datei oder voll qualifizierter Name der sequentiellen Datei z/VSE: Verweis auf das JCL DD-Statement der sequentiellen Datei	
	Die Angabe muss unter z/OS auf ein DD-Statement und unter z/VSE auf ein DLBL-Statement verweisen, mit dem die sequentielle Datei definiert wird. Unter z/OS kann alternativ der voll qualifizierte Name der sequentiellen Datei angegeben werden.	
	Beispiel	SEQ=XPSSEQ
TRACE=	'ALL' 'D' 'T' 'S' 'Z' oder eine, jeweils durch Komma getrennte, Kombination	
	Mit dieser Option können Traceoptionen für den Job festgelegt werden. Der Trace dient zur Ausgabe von Informationen, die bei Bedarf die Analyse des internen Ablaufs ermöglichen. Traceausgaben erfolgen direkt in das Job-Listing. Die folgende Tabelle enthält die verfügbaren Tracetypen:	
	Typ	Beschreibung
	ALL	Alle Tracetypen sind aktiviert.
	D	Der Datentrace wird aktiviert.
	T	Der TCP/IP Trace wird aktiviert.
	S	Der Speichertrace wird aktiviert.
	Z	Der ZIP/UNZIP Trace wird aktiviert.
	Mehrere Tracetypen können aktiviert werden, indem diese, durch Komma getrennt, aufgelistet werden.	
	Beispiel	TRACE=D,T
WFD=	'Y' 'N'	
	Mit dieser Option ist festzulegen, ob zZip eine Sicherungskopie der Eingabedaten auf der verbundenen XPS HFS-Datei erstellen soll. Diese Datei wird in der JCL durch das DD/DLBL Statement 'XPSHFS' identifiziert. Falls diese Angabe in der JCL nicht erfolgt, wird die Vorbelegung für 'XPSHFS' aus der Initialisierungsdatei, siehe Seite 34, übernommen. Diese Option wird nur dann berücksichtigt, wenn die Ausgabedaten über einen HostDrive/z Server zugestellt werden. In diesem Fall kann die wfd-Verarbeitung des HostDrive Queuing Systems genutzt werden. Die wfd-Verarbeitung (waiting for delivery) ermöglicht die automatische Zustellung von Dateien an das gewünschte Ziel zu einem späteren Zeitpunkt, falls dieses zur Ausführungszeit des zZip Jobs nicht erreicht werden kann.	
WTO=	'Y' 'N'	
	Mit dieser Option ist festzulegen, ob zZip beim Jobstart eine Nachricht über die verwendete Version und beim Jobende eine Nachricht über die Anzahl der verarbeiteten Sätze auf die Konsole ausgeben soll oder nicht. Mit einer Angabe kann die globale Voreinstellung – siehe Seite 34 - überschrieben werden.	

TAPE - Lesen der Daten von Band

Mit diesem Steuerkartentyp wird zZip darüber informiert, dass Eingabedaten von einem Band (virtuell oder physisch) gelesen werden sollen.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JTAP2EML JOB , 'TAP TO EMAIL', CLASS=A, MSGCLASS=M
//DELBKUP EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE XPS.HFS.FILE.TAPE.BACKUP
/*
//BACKUP EXEC PGM=XPSSPMNT, REGION=32M
//STEPLIB DD DISP=SHR, DSN=XPSHDRV.V300.LOADLIB
//XPSHFS DD DISP=SHR, DSN=XPSDAEM.TEST.HFS
//TAP DD DISP=(NEW,CATLG), DSN=XPS.HFS.FILE.TAPE.BACKUP,
// SPACE=(65000,(10000,200)), VOL=SER=XPS009,
// UNIT=3390, DCB=(RECFM=V, BLKSIZE=8200, LRECL=8196)
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SPOOLLOG DD SYSOUT=*
//SYSIN DD *
LANG D
LISTDIR
BACKUP
/*
//TAP2EML EXEC PGM=XPSZZIP, REGION=96M
//STEPLIB DD DSN=XPSHDRV.V300.LOADLIB, DISP=SHR
//TAP DD DISP=(OLD,KEEP), DSN=XPS.HFS.FILE.TAPE.BACKUP,
// SPACE=(65000,(10000,200)), VOL=SER=XPS006,
// UNIT=3390, DCB=(RECFM=V, BLKSIZE=8200, LRECL=8196)
//SYSIN DD *
A ACTION=ZIP
I TAPE=TAP
O ARCHIVE=EMAIL=XPSHFS.ZIP
O IPADDR=192.168.0.110 IPPORT=25
O USER=&EMAILUSER PWD=&PASSWORD
O FROM=XPSHOST@ECHING.DE
O TO=&HFS_BKUP_REC
O SUBJECT='XPSHFS BACKUP'
INCLUDE=BKUPBODY
/*
```

Abb. 45: Beispieljob für TAPE Eingabe z/OS

Im Beispiel wird zunächst unter Verwendung des XPS HFS-Batchprozessors 'XPSSPMNT' ein Backup einer XPS HFS-Datei auf ein virtuelles Band erstellt.

Im zweiten Jobstep wird das zuvor erstellte virtuelle Band als Eingabemedium verwendet, was durch die Steuerkarte 'I TAPE=' erreicht wird.

Aus dem Inhalt des virtuellen Bandes wird ein ZIP-Archiv erstellt, das schließlich als E-Mail Anhang versendet wird.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE, der einen vergleichbaren Verarbeitungsprozess durchläuft, wie der zuvor abgebildete Beispieljob für z/OS. Das virtuelle Band wird in diesem Fall durch einen VTAPE-Server realisiert, der auf einem Rechner läuft, der über TCP/IP adressiert werden kann.

Durch die Verwendung einer virtuellen Banddatei mit der Endung '.ZAWS' wird automatisch ein komprimiertes Archiv erstellt.

```
* $$ JOB JNM=JTAP2EML,DISP=D,PRI=3,CLASS=0
* $$ LST CLASS=T
// JOB JTAP2EML
// VTAPE STOP,UNIT=590
// VTAPE START,UNIT=590,LOC=192.168.0.110:8880,FILE='C:/VTAPE/HFS.ZAWS'
// LIBDEF PHASE,SEARCH=(XPS.DAEMH600)
// DLBL XPSHFS,'XPS.SPOOL.FILE',,VSAM,CAT=XPSUCAT
// ASSGN SYS005,FEE
// TLBL TAP
// ASSGN SYS010,590
// OPTION NOSYSDUMP
// EXEC XPSSPMNT,SIZE=AUTO
LANG D
LISTDIR
BACKUP
/*
// OPTION SYSPARM='00'
// ASSGN SYS005,SYSLST
// ASSGN SYS010,590
// TLBL TAP
// DLBL XPSHFS,'XPSHDRV.SPOOL.SAVE',,VSAM,CAT=XPSUCAT
// EXEC XPSZZIP,SIZE=AUTO,PARM='LIB=XPS.DAEMH600'
I TAPE=TAP RECFM=V BLKSIZE=8200 RECL=8196 LABEL=N
O EMAIL=SPOOL.BACKUP
O IPADDR=192.168.0.110 IPPORT=25
O USER=&EMAIL_GEOG PWD=&EMAIL_GEOG_PWD
O FROM=XPSHOST@ECHING.DE
O TO=&EMAIL_RECIPIENT
O SUBJECT='XPSHFS BACKUP'
O BODY:
O THIS IS THE BODY.
O THIS IS LINE TWO(2).
O THIS IS LINE THREE(40).
O END:
/*
/&
// VTAPE STOP,UNIT=590
* $$ EOJ
```

Abb. 46: Beispieljob für TAPE Eingabe z/VSE

Unter z/VSE ist das Band, wenn es als Eingabemedium eingesetzt wird, immer über ein Assignment an 'SYS010' anzubinden.

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Eingabekarten des Typs 'I TAPE='.

I TAPE=	
Lesen von Daten von einem Band – virtuell oder physisch.	
Optionen	Beschreibung
BLKSIZE=	numerischer Wert
	Mit dieser Option ist zZip über die Blocksize zu informieren, mit der die Daten auf das Band geschrieben wurden.
	Diese Angabe ist unter z/VSE immer erforderlich und unter z/OS nur dann, wenn die

	Information nicht über das DD-Statement für das Band ermittelbar ist.	
	Beispiel	BLKSIZE=8200
LABEL=	'N' 'S'	
	<p>Mit dieser Option ist zZip darüber zu informieren, ob bei der Erstellung des Bandes ein Label geschrieben wurde.</p> <p>Folgende Angaben sind möglich:</p> <p>'N' das Band enthält keinen Label / Voreinstellung</p> <p>'S' das Band enthält einen Standardlabel</p>	
RECFM=	'F' ' <u>F</u> B' 'V' 'VB'	
	<p>Mit dieser Angabe ist das Recordformat des Bandes zu bestimmen.</p> <p>Diese Angabe ist unter z/VSE immer erforderlich, wenn sie nicht dem Standard 'RECFM=FB' entspricht.</p> <p>Unter z/OS ist die Angabe nur dann erforderlich, wenn zZip das Recordformat nicht automatisch ermitteln kann.</p> <p>Die anzugebenden Werte entsprechen den Originalwerten der Job Control:</p> <p>'F' fixed</p> <p>'<u>F</u>B' fixed blocked</p> <p>'V' variable</p> <p>'VB' variable blocked</p>	
RECL=	numerischer Wert	
	<p>Mit dieser Option ist die Satzlänge der Daten auf dem Band anzugeben.</p> <p>Diese Angabe ist unter z/VSE immer und unter z/OS nur dann erforderlich, wenn zZip die Satzlänge nicht automatisch aus der Job Control ermitteln kann.</p>	
	Beispiel	RECL=8196
RECORD=	Verweis auf eine Satzstruktur auf der XPS HFS-Datei	
	<p>Mit dieser Option kann bekannt gegeben werden, dass das Format der Eingabedaten durch die angegebene Struktur beschrieben wird. Damit wird die Verarbeitung der Daten auf Feldebene ermöglicht.</p>	
	Beispiel	RECORD=ARTISTS/KOMPONISTEN
REW=	'N' ' <u>R</u> ' 'U'	
	<p>Mit dieser Option ist anzugeben, wie mit dem Band nach Abschluss des Lesens verfahren werden soll.</p> <p>Folgende Angaben sind möglich:</p> <p>'N' das Band wird nach Abschluss des Lesens nicht zurückgespult</p> <p>'<u>R</u>' das Band wird nach Abschluss des Lesens zurückgespult / Voreinstellung</p> <p>'U' das Band wird nach Abschluss des Lesens entladen</p>	

<i>TAPE=</i>	Verweis auf das JCL-Statement des Bandes	
	Die Angabe muss unter z/OS auf ein DD-Statement und unter z/VSE auf ein TLBL-Statement verweisen, mit dem das Band definiert wird.	
	Beispiel	TAPE=TAP
<i>TRACE=</i>	'ALL' 'D' 'T' 'S' 'Z' oder eine, jeweils durch Komma getrennte, Kombination	
	Mit dieser Option können Traceoptionen für den Job festgelegt werden. Der Trace dient zur Ausgabe von Informationen, die bei Bedarf die Analyse des internen Ablaufs ermöglichen. Traceausgaben erfolgen direkt in das Job-Listing. Die folgende Tabelle enthält die verfügbaren Tracetypen:	
	Typ	Beschreibung
	<i>ALL</i>	Alle Tracetypen sind aktiviert.
	<i>D</i>	Der Datentrace wird aktiviert.
	<i>T</i>	Der TCP/IP Trace wird aktiviert.
	<i>S</i>	Der Speichertrace wird aktiviert.
	<i>Z</i>	Der ZIP/UNZIP Trace wird aktiviert.
	Mehrere Tracetypen können aktiviert werden, indem diese, durch Komma getrennt, aufgelistet werden.	
	Beispiel	TRACE=D,T
<i>WFD=</i>	'Y' 'N'	
	Mit dieser Option ist festzulegen, ob zZip eine Sicherungskopie der Eingabedaten auf der verbundenen XPS HFS-Datei erstellen soll. Diese Datei wird in der JCL durch das DD/DLBL Statement 'XPSHFS' identifiziert. Falls diese Angabe in der JCL nicht erfolgt, wird die Vorbelegung für 'XPSHFS' aus der Initialisierungsdatei, siehe Seite 34, übernommen. Diese Option wird nur dann berücksichtigt, wenn die Ausgabedaten über einen HostDrive/z Server zugestellt werden. In diesem Fall kann die wfd-Verarbeitung des HostDrive Queuing Systems genutzt werden. Die wfd-Verarbeitung (waiting for delivery) ermöglicht die automatische Zustellung von Dateien an das gewünschte Ziel zu einem späteren Zeitpunkt, falls dieses zur Ausführungszeit des zZip Jobs nicht erreicht werden kann.	
<i>WTO=</i>	'Y' 'N'	
	Mit dieser Option ist festzulegen, ob zZip beim Jobstart eine Nachricht über die verwendete Version und beim Jobende eine Nachricht über die Anzahl der verarbeiteten Sätze auf die Konsole ausgeben soll oder nicht. Mit einer Angabe kann die globale Voreinstellung – siehe Seite 34 - überschrieben werden.	

ESDS - Lesen der Daten aus einer VSAM ESDS Datei

Mit diesem Steuerkartentyp wird der z/OS darüber informiert, dass Eingabedaten aus einer VSAM ESDS Datei gelesen werden sollen.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JESD2XHF JOB , 'ESDS TO XPSHFS',CLASS=A,MSGCLASS=M
//ESD2XHF EXEC PGM=XPSZZIP,REGION=48M
//STEPLIB DD DSN=XPSHDRV.V300.LOADLIB,DISP=SHR
//XPSDATA DD DSN=XPSHDRV.V300.MACLIB,DISP=SHR
//ESDSIN DD DISP=SHR,DSN=XPSHDRV.ESDS
//SYSIN DD *
I ESDS=ESDSIN
O XPSHFS='TEST/KOMPS.TXT'
//*
```

Abb. 47: Beispieljob für ESDS Eingabe z/OS

Im Beispiel wird der Inhalt der VSAM ESDS Datei 'XPSHDRV.ESDS', referenziert durch die Angabe 'ESDSIN' auf der 'I ESDS=' Karte, binär in die Datei 'TEST/KOMPS.TXT' auf der angehängten XPS HFS Datei kopiert.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE, der denselben Verarbeitungsprozess durchläuft, wie der zuvor abgebildete Beispieljob für z/OS:

```
* $$ JOB JNM=JESD2XHF,DISP=D,PRI=3,CLASS=0
* $$ LST CLASS=T
// JOB JESD2XHF
// LIBDEF PHASE,SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005,FEE
// DLBL XPSESDS,'XPS.HDRV.ESDS',,VSAM,CAT=XPSUCAT
// DLBL XPSHFS,'XPSHDRV.SPOOL.SAVE',,VSAM,CAT=XPSUCAT
// EXEC XPSZZIP,SIZE=AUTO,PARM='LIB=XPS.DAEMH600'
I ESDS=XPSESDS
O XPSHFS='TEST/KOMPS.TXT'
/*
/&
* $$EQJ
```

Abb. 48: Beispieljob für ESDS Eingabe z/VSE

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Eingabekarten des Typs 'I SEQ='.

I ESDS=	
Lesen von Daten aus einer VSAM ESDS Datei.	
Optionen	Beschreibung
ESDS=	z/OS: Verweis auf das JCL DD-Statement der VSAM ESDS Datei oder voll qualifizierter Name der VSAM ESDS Datei z/VSE: Verweis auf das JCL DD-Statement der VSAM ESDS Datei
	Die Angabe muss unter z/OS auf ein DD-Statement und unter z/VSE auf ein DLBL-Statement verweisen, mit dem die VSAM ESDS Datei definiert wird. Unter z/OS kann alternativ der voll qualifizierte Name der VSAM ESDS Datei angegeben

	werden.												
	Beispiel ESDS=XPSESDS												
RECORD=	Verweis auf eine Satzstruktur auf der XPS HFS-Datei												
	Mit dieser Option kann bekannt gegeben werden, dass das Format der Eingabedaten durch die angegebene Struktur beschrieben wird. Damit wird die Verarbeitung der Daten auf Feldebene ermöglicht.												
	Beispiel RECORD=ARTISTS/KOMPONISTEN												
TRACE=	'ALL' 'D' 'T' 'S' 'Z' oder eine, jeweils durch Komma getrennte, Kombination												
	Mit dieser Option können Traceoptionen für den Job festgelegt werden. Der Trace dient zur Ausgabe von Informationen, die bei Bedarf die Analyse des internen Ablaufs ermöglichen. Traceausgaben erfolgen direkt in das Job-Listing. Die folgende Tabelle enthält die verfügbaren Tracetypen:												
	<table border="1"> <thead> <tr> <th>Typ</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>ALL</td> <td>Alle Tracetypen sind aktiviert.</td> </tr> <tr> <td>D</td> <td>Der Datentrace wird aktiviert.</td> </tr> <tr> <td>T</td> <td>Der TCP/IP Trace wird aktiviert.</td> </tr> <tr> <td>S</td> <td>Der Speichertrace wird aktiviert.</td> </tr> <tr> <td>Z</td> <td>Der ZIP/UNZIP Trace wird aktiviert.</td> </tr> </tbody> </table>	Typ	Beschreibung	ALL	Alle Tracetypen sind aktiviert.	D	Der Datentrace wird aktiviert.	T	Der TCP/IP Trace wird aktiviert.	S	Der Speichertrace wird aktiviert.	Z	Der ZIP/UNZIP Trace wird aktiviert.
Typ	Beschreibung												
ALL	Alle Tracetypen sind aktiviert.												
D	Der Datentrace wird aktiviert.												
T	Der TCP/IP Trace wird aktiviert.												
S	Der Speichertrace wird aktiviert.												
Z	Der ZIP/UNZIP Trace wird aktiviert.												
	Mehrere Tracetypen können aktiviert werden, in dem diese, durch Komma getrennt, aufgelistet werden.												
	Beispiel TRACE=D,T												
WFD=	'Y' 'N'												
	<p>Mit dieser Option ist festzulegen, ob zZip eine Sicherungskopie der Eingabedaten auf der verbundenen XPS HFS-Datei erstellen soll.</p> <p>Diese Datei wird in der JCL durch das DD/DLBL Statement 'XPSHFS' identifiziert. Falls diese Angabe in der JCL nicht erfolgt, wird die Vorbelegung für 'XPSHFS' aus der Initialisierungsdatei, siehe Seite 34, übernommen.</p> <p>Diese Option wird nur dann berücksichtigt, wenn die Ausgabedaten über einen HostDrive/z Server zugestellt werden.</p> <p>In diesem Fall kann die wfd-Verarbeitung des HostDrive Queuing Systems genutzt werden. Die wfd-Verarbeitung (waiting for delivery) ermöglicht die automatische Zustellung von Dateien an das gewünschte Ziel zu einem späteren Zeitpunkt, falls dieses zur Ausführungszeit des zZip Jobs nicht erreicht werden kann.</p>												
WTO=	'Y' 'N'												
	Mit dieser Option ist festzulegen, ob zZip beim Jobstart eine Nachricht über die verwendete Version und beim Jobende eine Nachricht über die Anzahl der verarbeiteten Sätze auf die Konsole ausgeben soll oder nicht. Mit einer Angabe kann die globale Voreinstellung – siehe Seite 34 - überschrieben werden.												

KSDS – Lesen der Daten aus einer VSAM KSDS Datei

Mit diesem Steuerkartentyp wird zZip darüber informiert, dass Eingabedaten aus einer VSAM KSDS Datei gelesen werden sollen.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JKSD2JES JOB , 'KSDS TO JES',CLASS=A,MSGCLASS=M
//KSD2JES EXEC PGM=XPSZZIP,REGION=96M
//STEPLIB DD DSN=XPSHDRV.V300.LOADLIB,DISP=SHR
//XPSADVW DD DISP=SHR,DSN=VSAM.ADVW.V500.FILE
//SYSIN DD *
I KSDS=XPSADVW RECORD=XPS/ADVW2
R LAST-NAME FIRST-NAME ZIP CITY
O JES=OUT CLA=M DEST=XPS FORM=ADVW
O CSV= CSVHDR=R
//*
```

Abb. 49: Beispieljob für KSDS Eingabe z/OS

Im Beispiel wird der gesamte Inhalt der VSAM KSDS Datei 'VSAM.ADVW.V500.FILE', identifiziert durch die Angabe 'XPSADVW' auf der 'I KSDS=' Karte, in die JES Output Klasse 'M' gespielt.

Über die Record-Karte 'R' wird die Ausgabe auf bestimmte Felder aus der referenzierten Struktur beschränkt.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE:

```
* $$ JOB JNM=JKSD2POW,DISP=D,PRI=3,CLASS=0
* $$ LST CLASS=T
// JOB JKSD2POW
// LIBDEF PHASE,SEARCH=(XPS.HDRV300,XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005,FEE
// DLBL XPSADVW,'ADVW.V500.FILE',,VSAM,CAT=VSESPUC
// DLBL XPSHFS,'XPSHDRV.SPOOL.SAVE',,VSAM,CAT=XPSUCAT
// EXEC XPSZZIP,SIZE=AUTO,PARM='LIB=XPS.DAEMH600'
I KSDS=XPSADVW RECORD=XPS/ADVW
O POWER=LST JNAME=HDRV1 PUSER=MIKE CLA=C DISP=D CTLCH=09 RECL=100
O CSV= CSVHDR=Y CSVFMT=T
/*
/&
* $$EOJ
```

Abb. 50: Beispieljob für KSDS Eingabe z/VSE

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Eingabekarten des Typs 'I KSDS='.

I KSDS=	
Lesen von Daten aus einer VSAM KSDS Datei.	
Optionen	Beschreibung
KEY=	(Generischer) Schlüssel
	Mit dieser Option kann ein (generischer) Schlüssel angegeben werden, mit dem der Umfang der von der KSDS Datei gelesenen Records eingeschränkt werden kann. Diese Angabe ist direkt mit der nachfolgenden Option 'OPT=' verknüpft.

	Falls die gesamte KSDS Datei verarbeitet werden soll, ist diese Option wegzulassen.												
	Beispiel KEY=MUSTER*												
KSDS=	z/OS: Verweis auf das JCL DD-Statement der VSAM KSDS Datei oder voll qualifizierter Name der VSAM KSDS Datei z/VSE: Verweis auf das JCL DD-Statement der VSAM KSDS Datei												
	Die Angabe muss unter z/OS auf ein DD-Statement und unter z/VSE auf ein DLBL-Statement verweisen, mit dem die VSAM KSDS Datei definiert wird. Unter z/OS kann alternativ der voll qualifizierte Name der VSAM KSDS Datei angegeben werden.												
	Beispiel KSDS=XPSKSDS												
OPT=	'EQ' 'LE' 'GE'												
	Mit dieser Option ist festzulegen, wie der bei der vorherigen Option 'KEY=' angegebene Schlüssel im Rahmen des Browsens der Datei zu interpretieren ist. Folgende Angaben sind möglich: 'EQ' Lediglich der Satz mit dem angegebenen Schlüssel wird verarbeitet. Dieser Wert entspricht der Voreinstellung. 'LE' Die Verarbeitung beginnt am Dateianfang und endet, nachdem der angegebene Schlüssel verarbeitet wurde. 'GE' Die Verarbeitung beginnt mit dem angegebenen Schlüssel und endet am Dateende.												
	Beispiel OPT=LE												
RECORD=	Verweis auf eine Satzstruktur auf der XPS HFS-Datei												
	Mit dieser Option kann bekannt gegeben werden, dass das Format der Eingabedaten durch die angegebene Struktur beschrieben wird. Damit wird die Verarbeitung der Daten auf Feldebene ermöglicht.												
	Beispiel RECORD=ARTISTS/KOMPONISTEN												
TRACE=	'ALL' 'D' 'T' 'S' 'Z' oder eine, jeweils durch Komma getrennte, Kombination												
	Mit dieser Option können Traceoptionen für den Job festgelegt werden. Der Trace dient zur Ausgabe von Informationen, die bei Bedarf die Analyse des internen Ablaufs ermöglichen. Traceausgaben erfolgen direkt in das Job-Listing. Die folgende Tabelle enthält die verfügbaren Tracetypen:												
	<table border="1"> <thead> <tr> <th>Typ</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>ALL</td> <td>Alle Tracetypen sind aktiviert.</td> </tr> <tr> <td>D</td> <td>Der Datentrace wird aktiviert.</td> </tr> <tr> <td>T</td> <td>Der TCP/IP Trace wird aktiviert.</td> </tr> <tr> <td>S</td> <td>Der Speichertrace wird aktiviert.</td> </tr> <tr> <td>Z</td> <td>Der ZIP/UNZIP Trace wird aktiviert.</td> </tr> </tbody> </table>	Typ	Beschreibung	ALL	Alle Tracetypen sind aktiviert.	D	Der Datentrace wird aktiviert.	T	Der TCP/IP Trace wird aktiviert.	S	Der Speichertrace wird aktiviert.	Z	Der ZIP/UNZIP Trace wird aktiviert.
Typ	Beschreibung												
ALL	Alle Tracetypen sind aktiviert.												
D	Der Datentrace wird aktiviert.												
T	Der TCP/IP Trace wird aktiviert.												
S	Der Speichertrace wird aktiviert.												
Z	Der ZIP/UNZIP Trace wird aktiviert.												
	Mehrere Tracetypen können aktiviert werden, in dem diese, durch Komma getrennt, aufgelistet werden.												
	Beispiel TRACE=D,T												
WFD=	'Y' 'N'												

	<p>Mit dieser Option ist festzulegen, ob zZip eine Sicherungskopie der Eingabedaten auf der verbundenen XPS HFS-Datei erstellen soll.</p> <p>Diese Datei wird in der JCL durch das DD/DLBL Statement 'XPSHFS' identifiziert. Falls diese Angabe in der JCL nicht erfolgt, wird die Vorbelegung für 'XPSHFS' aus der Initialisierungsdatei, siehe Seite 34, übernommen.</p> <p>Diese Option wird nur dann berücksichtigt, wenn die Ausgabedaten über einen HostDrive/z Server zugestellt werden.</p> <p>In diesem Fall kann die wfd-Verarbeitung des HostDrive Queuing Systems genutzt werden. Die wfd-Verarbeitung (waiting for delivery) ermöglicht die automatische Zustellung von Dateien an das gewünschte Ziel zu einem späteren Zeitpunkt, falls dieses zur Ausführungszeit des zZip Jobs nicht erreicht werden kann.</p>
<i>WTO=</i>	'Y' ' <u>N</u> '
	<p>Mit dieser Option ist festzulegen, ob zZip beim Jobstart eine Nachricht über die verwendete Version und beim Jobende eine Nachricht über die Anzahl der verarbeiteten Sätze auf die Konsole ausgeben soll oder nicht. Mit einer Angabe kann die globale Voreinstellung – siehe Seite 34 - überschrieben werden.</p>

XPSHFS – Lesen der Daten vom XPS HFS-Dateisystem

Mit diesem Steuerkartentyp wird zZip darüber informiert, dass Eingabedaten aus dem XPS HFS-Dateisystem gelesen werden sollen.

Das XPS HFS-Dateisystem ist ein hierarchisches Dateisystem auf Basis einer VSAM RRDS Datei.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JXHF2LIB JOB , 'XPSHFS TO LIB',CLASS=A,MSGCLASS=M
//XHF2LIB EXEC PGM=XPSZZIP,REGION=48M
//STEPLIB DD DISP=SHR,DSN=XPSHDRV.V300.LOADLIB
//XPSDATA DD DSN=XPSHDRV.V300.MACLIB,DISP=SHR
//XPSOUT DD DISP=SHR,DSN=XPSHDRV.TEST.MACLIB
//SYSIN DD *
I XPSHFS='TEST/KOMPS.TXT' RECL=80
O LIB=XPSOUT MBR=KOMPS
//*
```

Abb. 51: Beispieljob für XPSHFS Eingabe z/OS

Im Beispiel wird die Datei 'KOMPS.TXT' aus dem Verzeichnis 'TEST' auf der XPS HFS Datei in ein Member mit Namen 'KOMPS' auf der durch 'LIB=XPSOUT' identifizierten Bibliothek kopiert.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE:

```
* $$ JOB JNM=JXHF2LIB,DISP=D,PRI=3,CLASS=0
* $$ LST CLASS=T
// LIBDEF PHASE,SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005,FEE
// DLBL XPSHFS,'XPSHDRV.SPOOL.SAVE',,VSAM,CAT=XPSUCAT
// EXEC XPSZZIP,SIZE=AUTO,PARM='LIB=XPS.DAEMH600'
I XPSHFS='TEST/KOMPS.TXT'
O LIB=XPS.HDRV300,MBR=KOMPSTST.D
/*
/&
* $$EOJ
```

Abb. 52: Beispieljob für XPSHFS Eingabe z/VSE

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Eingabekarten des Typs 'I XPSHFS='.

I XPSHFS=	
Lesen von Daten aus dem XPS HFS-Dateisystem.	
Optionen	Beschreibung
RECORD=	Verweis auf eine Satzstruktur auf der XPS HFS-Datei
	Mit dieser Option kann bekannt gegeben werden, dass das Format der Eingabedaten durch die angegebene Struktur beschrieben wird. Damit wird die Verarbeitung der Daten auf Feldebene ermöglicht.
	Beispiel RECORD=ARTISTS/KOMPONISTEN
TRACE=	'ALL' 'D' 'T' 'S' 'Z' oder eine, jeweils durch Komma getrennte, Kombination
	Mit dieser Option können Traceoptionen für den Job festgelegt werden. Der Trace dient zur Ausgabe von Informationen, die bei Bedarf die Analyse des internen Ablaufs

	ermöglichen. Traceausgaben erfolgen direkt in das Job-Listing. Die folgende Tabelle enthält die verfügbaren Tracetypen:	
	Typ	Beschreibung
	<i>ALL</i>	Alle Tracetypen sind aktiviert.
	<i>D</i>	Der Datentrace wird aktiviert.
	<i>T</i>	Der TCP/IP Trace wird aktiviert.
	<i>S</i>	Der Speichertrace wird aktiviert.
	<i>Z</i>	Der ZIP/UNZIP Trace wird aktiviert.
	Mehrere Tracetypen können aktiviert werden, in dem diese, durch Komma getrennt, aufgelistet werden.	
	Beispiel	TRACE=D,T
<i>WFD=</i>	'Y' 'N'	
	<p>Mit dieser Option ist festzulegen, ob zZip eine Sicherungskopie der Eingabedaten auf der verbundenen XPS HFS-Datei erstellen soll.</p> <p>Diese Datei wird in der JCL durch das DD/DLBL Statement 'XPSHFS' identifiziert. Falls diese Angabe in der JCL nicht erfolgt, wird die Vorbelegung für 'XPSHFS' aus der Initialisierungsdatei, siehe Seite 34, übernommen.</p> <p>Diese Option wird nur dann berücksichtigt, wenn die Ausgabedaten über einen HostDrive/z Server zugestellt werden.</p> <p>In diesem Fall kann die wfd-Verarbeitung des HostDrive Queuing Systems genutzt werden. Die wfd-Verarbeitung (waiting for delivery) ermöglicht die automatische Zustellung von Dateien an das gewünschte Ziel zu einem späteren Zeitpunkt, falls dieses zur Ausführungszeit des zZip Jobs nicht erreicht werden kann.</p>	
<i>WTO=</i>	'Y' 'N'	
	<p>Mit dieser Option ist festzulegen, ob zZip beim Jobstart eine Nachricht über die verwendete Version und beim Jobende eine Nachricht über die Anzahl der verarbeiteten Sätze auf die Konsole ausgeben soll oder nicht. Mit einer Angabe kann die globale Voreinstellung – siehe Seite 34 - überschrieben werden.</p>	
<i>XPSHFS=</i>	voll qualifizierte Pfadangabe	
	Mit dieser Option wird die zu verarbeitende Datei definiert.	
	Beispiel	XPSHFS='TEST/SUBTEST/USERS.TXT'

FTP – Lesen der Daten von einem FTP-Server

Mit diesem Steuerkartentyp wird zZip darüber informiert, dass Eingabedaten von einem FTP-Server gelesen werden sollen.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JFTP2SEQ JOB , 'FTP TO SEQ', CLASS=A, MSGCLASS=M
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE XPSHDRV.SEQ
//FTP2SEQ EXEC PGM=XPSZZIP, REGION=48M
//STEPLIB DD DSN=XPSHDRV.V300.LOADLIB, DISP=SHR
//XPSSEQ DD DSN=XPSHDRV.SEQ, DISP=(NEW,CATLG),
// SPACE=(TRK,(5,3),RLSE), UNIT=3390, VOL=SER=XPS005,
// DCB=(LRECL=100, BLKSIZE=1000, RECFM=FB, DSORG=PS)
//SYSIN DD *
I FTP='/TEST/XPSVP01.TXT' CP=1141-N2E
I IPADDR=192.168.0.131 IPPORT=21
I USER='user' PWD='pwd'
O SEQ=XPSSEQ
//*
//PRSEQ EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SEQIN DD DISP=SHR, DSN=XPSHDRV.SEQ
//SYSIN DD *
PRINT -
INFILE(SEQIN) -
HEX
//*
```

Abb. 53: Beispieljob für FTP Eingabe z/OS

Im Beispiel wird die Datei 'KOMPS.TXT' aus dem Verzeichnis 'TEST' auf der XPS HFS Datei in ein Member mit Namen 'KOMPS' auf der durch 'LIB=XPSOUT' identifizierten Bibliothek kopiert.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE:

```
* $$ JOB JNM=JFTP2SEQ, DISP=D, PRI=3, CLASS=0
* $$ LST CLASS=T
// JOB JFTP2SEQ
// LIBDEF PHASE, SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005, FEE
// DLBL XPSSEQ, 'HDRV.SEQ.FILE', , VSAM, DISP=(NEW,KEEP), X
// RECORDS=(500,50), RECSIZE=80
// DLBL XPSHFS, 'XPSHDRV.SPOOL.SAVE', , VSAM, CAT=XPSUCAT
// EXEC XPSZZIP, SIZE=AUTO, PARM='LIB=XPS.DAEMH600'
I FTP='/TEST/XPSVP01.TXT' CP=1141-N2E
I IPADDR=192.168.0.110 IPPORT=21
I USER=&USERID PWD=&PASSWORD
O SEQ=XPSSEQ RECL=80
/*
/&
* $$EQJ
```

Abb. 54: Beispieljob für FTP Eingabe z/VSE

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Eingabekarten des Typs 'I FTP='.

I FTP=			
Lesen von Daten von einem FTP-Server.			
Optionen	Beschreibung		
CP=	Codepage		
	<p>Mit dieser Option ist zZip darüber zu informieren, dass Textdaten verarbeitet werden sollen. Falls keine Angabe für eine Codepage gemacht wird, werden die Daten binär übertragen.</p> <p>Weitere Informationen zu Codepages sind dem Kapitel 'Codepages' auf Seite 13 zu entnehmen. Eine Tabelle der verfügbaren Codepages befindet sich auf Seite 15.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>CP=1141-N2E</td> </tr> </table>	Beispiel	CP=1141-N2E
Beispiel	CP=1141-N2E		
CPA2E=	Schlüssel einer Codepage		
	<p>Textinformationen, die im Rahmen des FTP-Protokolls zwischen beiden Parteien ausgetauscht werden, sind aus Gründen der Standardisierung immer in ASCII/ANSI kodiert.</p> <p>Mit dieser Option kann festgelegt werden, welche Codepage zur Übersetzung dieser Textdaten beim Lesen von Protokollinformationen verwendet werden soll.</p> <p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 23 beschrieben, global vorbelegt werden.</p> <p>Falls unter z/OS auf eine Angabe für diese Option verzichtet wird, werden die textbasierten Protokollinformationen unter Verwendung der im System eingestellten Codepage übersetzt, die bei Ausführung des Systemmakros 'XLATE' für die Übersetzung von ASCII/ANSI nach EBCDIC zum Einsatz kommt.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>CPA2E=1141-N2E</td> </tr> </table>	Beispiel	CPA2E=1141-N2E
Beispiel	CPA2E=1141-N2E		
CPE2A=	Schlüssel einer Codepage		
	<p>Textinformationen, die im Rahmen des FTP-Protokolls zwischen beiden Parteien ausgetauscht werden, sind aus Gründen der Standardisierung immer in ASCII/ANSI kodiert.</p> <p>Mit dieser Option kann festgelegt werden, welche Codepage zur Übersetzung dieser Textdaten beim Schreiben von Protokollinformationen verwendet werden soll.</p> <p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 23 beschrieben, global vorbelegt werden.</p> <p>Falls unter z/OS auf eine Angabe für diese Option verzichtet wird, werden die textbasierten Protokollinformationen unter Verwendung der im System eingestellten Codepage übersetzt, die bei Ausführung des Systemmakros 'XLATE' für die Übersetzung von EBCDIC nach ASCII/ANSI zum Einsatz kommt.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>CPE2A=1141-E2N</td> </tr> </table>	Beispiel	CPE2A=1141-E2N
Beispiel	CPE2A=1141-E2N		
FTP=	voll qualifizierte Pfadangabe		
	<p>Mit dieser Option wird die zu lesende Datei definiert auf dem FTP-Server definiert.</p> <p>Es ist zu beachten, dass die Angabe relativ zu einem eventuell eingerichteten Home Verzeichnis auf dem Server für den ausführenden User interpretiert wird.</p> <p>Unter Verwendung eines oder mehrerer FTP Kommandos – siehe hierzu die nachfolgend</p>		

	beschriebene Option 'FTPCMD=' – kann das Verzeichnis auf dem Server gewechselt werden, falls dies für den angemeldeten User zulässig ist.	
	Beispiel	FTP='TEST/XSPVP01.TXT'
FTPCMD=	gültiges FTP Kommando	
	Mit dieser Option können FTP Kommandos angegeben werden, deren Ausführung zZip vor der Übertragung von Daten auf den FTP Server veranlassen soll. zZip erlaubt die Angabe beliebig vieler FTP Kommandos. Es ist jedoch zu beachten, dass jedes FTP Kommando in einer eigene I-Steuerkarte anzugeben ist, und dass es in einfache Anführungszeichen einzuschließen ist, falls Leerzeichen enthalten sind.	
	Beispiel	I FTPCMD='CWD ..'
IPADDR=	TCP/IP Adresse – numerisch oder als DNS-Name	
	Mit dieser Option ist TCP/IP Adresse des FTP-Servers anzugeben.	
	Beispiele	IPADDR=192.168.0.131 IPADDR=ftp.mozilla.org
IPPORT=	numerischer TCP/IP Port <= 65535 '21'	
	Mit dieser Option ist der TCP/IP Port Anzugeben, den der FTP-Server auf eingehende Anfragen hin überwacht. Falls für diese Option keine Angabe erfolgt, wird Port 21 verwendet, was dem gängigen Standard entspricht.	
PWD=	<u>Anonymus</u> Passwort	
	Mit dieser Option ist das Passwort des Benutzers für die Anmeldung am FTP-Server anzugeben.	
	Beispiel	PWD=schubert
RECORD=	Verweis auf eine Satzstruktur auf der XPS HFS-Datei	
	Mit dieser Option kann bekannt gegeben werden, dass das Format der Eingabedaten durch die angegebene Struktur beschrieben wird. Damit wird die Verarbeitung der Daten auf Feldebene ermöglicht.	
	Beispiel	RECORD=ARTISTS/KOMPONISTEN
TRACE=	'ALL' 'D' 'T' 'S' 'Z' oder eine, jeweils durch Komma getrennte, Kombination	
	Mit dieser Option können Traceoptionen für den Job festgelegt werde. Der Trace dient zur Ausgabe von Informationen, die bei Bedarf die Analyse des internen Ablaufs ermöglichen. Traceausgaben erfolgen direkt in das Job-Listing. Die folgende Tabelle enthält die verfügbaren Tracetypen:	
	Typ	Beschreibung
	ALL	Alle Tracetypen sind aktiviert.
	D	Der Datentrace wird aktiviert.
	T	Der TCP/IP Trace wird aktiviert.
	S	Der Speichertrace wird aktiviert.
	Z	Der ZIP/UNZIP Trace wird aktiviert.

	Mehrere Tracetypen können aktiviert werden, in dem diese, durch Komma getrennt, aufgelistet werden.	
	Beispiel	TRACE=D,T
USER=	<u>Anonymus</u> Benutzername	
	Mit dieser Option ist der Name des Benutzers für die Anmeldung am FTP-Server anzugeben.	
	Beispiel	USER=franz
WFD=	'Y' 'N'	
	<p>Mit dieser Option ist festzulegen, ob zZip eine Sicherungskopie der Eingabedaten auf der verbundenen XPS HFS-Datei erstellen soll.</p> <p>Diese Datei wird in der JCL durch das DD/DLBL Statement 'XPSHFS' identifiziert. Falls diese Angabe in der JCL nicht erfolgt, wird die Vorbelegung für 'XPSHFS' aus der Initialisierungsdatei, siehe Seite 34, übernommen.</p> <p>Diese Option wird nur dann berücksichtigt, wenn die Ausgabedaten über einen HostDrive/z Server zugestellt werden.</p> <p>In diesem Fall kann die wfd-Verarbeitung des HostDrive Queuing Systems genutzt werden. Die wfd-Verarbeitung (waiting for delivery) ermöglicht die automatische Zustellung von Dateien an das gewünschte Ziel zu einem späteren Zeitpunkt, falls dieses zur Ausführungszeit des zZip Jobs nicht erreicht werden kann.</p>	
WTO=	'Y' ' <u>N</u> '	
	Mit dieser Option ist festzulegen, ob zZip beim Jobstart eine Nachricht über die verwendete Version und beim Jobende eine Nachricht über die Anzahl der verarbeiteten Sätze auf die Konsole ausgeben soll oder nicht. Mit einer Angabe kann die globale Voreinstellung – siehe Seite 34 - überschrieben werden.	

HFS – Lesen der Daten aus dem OpenMVS HFS-Dateisystem

Mit diesem Steuerkartentyp wird der zZip darüber informiert, dass Eingabedaten aus dem OpenMVS HFS-Dateisystem gelesen werden sollen. Diese Option kann nur unter z/OS verwendet werden.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JHFS2TAP JOB , 'HFS TO TAPE' ,CLASS=A,MSGCLASS=M
//DELTAPE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE XPS.HFS.FILE.TAPE.BACKUP
/*
//HFS2TAP EXEC PGM=XPSZZIP,REGION=96M
//STEPLIB DD DSN=XPSHDRV.V300.LOADLIB,DISP=SHR
//TAP DD DISP=(NEW,CATLG),DSN=XPS.HFS.FILE.TRANSFER,
// SPACE=(65000,(10000,200)),VOL=SER=XPS009,
// UNIT=3390,DCB=(RECFM=VB,BLKSIZE=81964,LRECL=8196)
//SYSIN DD *
I HFS='/usr/common/overlay.jpg'
O TAPE=TAP
/*
```

Abb. 55: Beispieljob für HFS Eingabe z/OS

Im Beispiel wird die Datei 'overlay.jpg' aus dem Verzeichnis '/usr/common' aus dem OpenMVS HFS-Dateisystem gelesen und auf ein virtuelles Band geschrieben.

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Eingabekarten des Typs 'I HFS='.

I HFS=	
Lesen von Daten aus dem OpenMVS HFS-Dateisystem – nur z/OS.	
Optionen	Beschreibung
<i>HFS=</i>	Pfadangabe zur Datei im HFS-Dateisystem
	Es ist zu beachten, dass bei Dateinamen im OpenMVS HFS-Dateisystem zwischen Groß- und Kleinschreibung unterschieden wird.
	Beispiel <i>HFS='/usr/common/overlay.jpg'</i>
<i>RECORD=</i>	Verweis auf eine Satzstruktur auf der XPS HFS-Datei
	Mit dieser Option kann bekannt gegeben werden, dass das Format der Eingabedaten durch die angegebene Struktur beschrieben wird. Damit wird die Verarbeitung der Daten auf Feldebene ermöglicht.
	Beispiel <i>RECORD=ARTISTS/KOMPONISTEN</i>
<i>RECL=</i>	numerischer Wert
	Mit dieser Option kann festgelegt werden, dass zZip aus Text-Eingabedaten Sätze mit festgelegter Länge generieren soll. Dazu werden die Eingabedaten zeilenweise gelesen und Zeilen mit kleinerer Länge als die hier angegebene werden mit dem in den Vorbelegungen definierten Fülleichen – siehe 'PADC=' auf Seite 32 – bis zur gewünschten Länge aufgefüllt. Zeilen, die länger als der hier angegebene Wert sind, werden in mehrere Zeilen der

	gewünschten Länge umgebrochen.	
	Beispiel	RECL=132
TRACE=	'ALL' 'D' 'T' 'S' 'Z' oder eine, jeweils durch Komma getrennte, Kombination	
	Mit dieser Option können Traceoptionen für den Job festgelegt werden. Der Trace dient zur Ausgabe von Informationen, die bei Bedarf die Analyse des internen Ablaufs ermöglichen. Traceausgaben erfolgen direkt in das Job-Listing. Die folgende Tabelle enthält die verfügbaren Tracetypen:	
	Typ	Beschreibung
	ALL	Alle Tracetypen sind aktiviert.
	D	Der Datentrace wird aktiviert.
	T	Der TCP/IP Trace wird aktiviert.
	S	Der Speichertrace wird aktiviert.
	Z	Der ZIP/UNZIP Trace wird aktiviert.
	Mehrere Tracetypen können aktiviert werden, in dem diese, durch Komma getrennt, aufgelistet werden.	
	Beispiel	TRACE=D,T
WFD=	'Y' 'N'	
	Mit dieser Option ist festzulegen, ob zZip eine Sicherungskopie der Eingabedaten auf der verbundenen XPS HFS-Datei erstellen soll. Diese Datei wird in der JCL durch das DD/DLBL Statement 'XPSHFS' identifiziert. Falls diese Angabe in der JCL nicht erfolgt, wird die Vorbelegung für 'XPSHFS' aus der Initialisierungsdatei, siehe Seite 34, übernommen. Diese Option wird nur dann berücksichtigt, wenn die Ausgabedaten über einen HostDrive/z Server zugestellt werden. In diesem Fall kann die wfd-Verarbeitung des HostDrive Queuing Systems genutzt werden. Die wfd-Verarbeitung (waiting for delivery) ermöglicht die automatische Zustellung von Dateien an das gewünschte Ziel zu einem späteren Zeitpunkt, falls dieses zur Ausführungszeit des zZip Jobs nicht erreicht werden kann.	
WTO=	'Y' 'N'	
	Mit dieser Option ist festzulegen, ob zZip beim Jobstart eine Nachricht über die verwendete Version und beim Jobende eine Nachricht über die Anzahl der verarbeiteten Sätze auf die Konsole ausgeben soll oder nicht. Mit einer Angabe kann die globale Voreinstellung – siehe Seite 34 - überschrieben werden.	

JES – Lesen der Daten aus dem z/OS Spoolsystem

Mit diesem Steuerkartentyp wird zZip darüber informiert, dass Eingabedaten aus JES, dem z/OS Spoolsystem, gelesen werden sollen. Diese Option kann nur unter z/OS verwendet werden.

Damit zZip JES Eingabedaten verarbeiten kann, ist die Präsenz einer XPS-PrintEx Basisinstallation erforderlich. PrintEx von XPS ist ein Managementsystem zur Erweiterung der Druckmöglichkeiten unter z/OS. Die Beschreibung der Installation und Verwaltung von PrintEx ist Gegenstand eines eigenen Handbuchs.

Durch die Verwendung von PrintEx zum Verarbeiten von JES Dateien steht zZip ein großer Funktionsumfang bzgl. der Verarbeitung von JES Ausgabedaten zur Verfügung wie z. B. Formulardruck oder die Konvertierung der Daten nach PDF oder Postscript.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JJES2NET JOB , 'JES TO NET', CLASS=A, MSGCLASS=M
//JJES2NET EXEC PGM=XPSZZIP, REGION=48M
//STEPLIB DD DISP=SHR, DSN=XPSHDRV.V300.LOADLIB
//XPDATA DD DSN=XPSHDRV.V300.MACLIB, DISP=SHR
//SYSIN DD *
I JES=J001
I IPADDR=127.0.0.1 IPPORT=8840
O NET='JESOUT/J001/&&FILE'
O IPADDR=192.168.0.131 IPPORT=8841
/**
```

Abb. 56: Beispieljob für JES Eingabe z/OS

Im Beispiel werden alle JES Ausgabedateien, die für das PrintEx Selektionskriterium 'J001' zur Verfügung stehen, gelesen und unter dem von PrintEx generierten Namen im Netzwerk abgelegt.

Im Rahmen der Konfiguration der PrintEx Selektionskriterien ist kenntlich zu machen, dass das gewählte Kriterium – im Beispiel 'J001' – von zZip genutzt wird. Außerdem kann dort definiert werden, dass ein bestimmter Batchjob gestartet werden soll, sobald Daten für das definierte Selektionskriterium bereit stehen. Bei Bedarf kann der Batchjob auch periodisch gestartet werden. Nähere Informationen sind dem PrintEx Handbuch zu entnehmen.

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Eingabekarten des Typs 'I JES='.

I JES=	
Lesen von Daten aus dem JES – nur z/OS.	
Optionen	Beschreibung
JES=	PrintEx Selektionskriterium
	<p>An dieser Stelle ist der Schlüssel des PrintEx Selektionskriteriums anzugeben, das die JES Ausgabedaten, die gelesen werden sollen, näher spezifiziert.</p> <p>Da die Druckdaten von PrintEx verwaltet und aufbereitet werden, stehen für diesen Eingabetyp keine weiteren Optionen zur Verfügung.</p> <p>Die nachfolgende Abbildung zeigt eine mögliche Definition eines PrintEx Selektionskriteriums:</p>

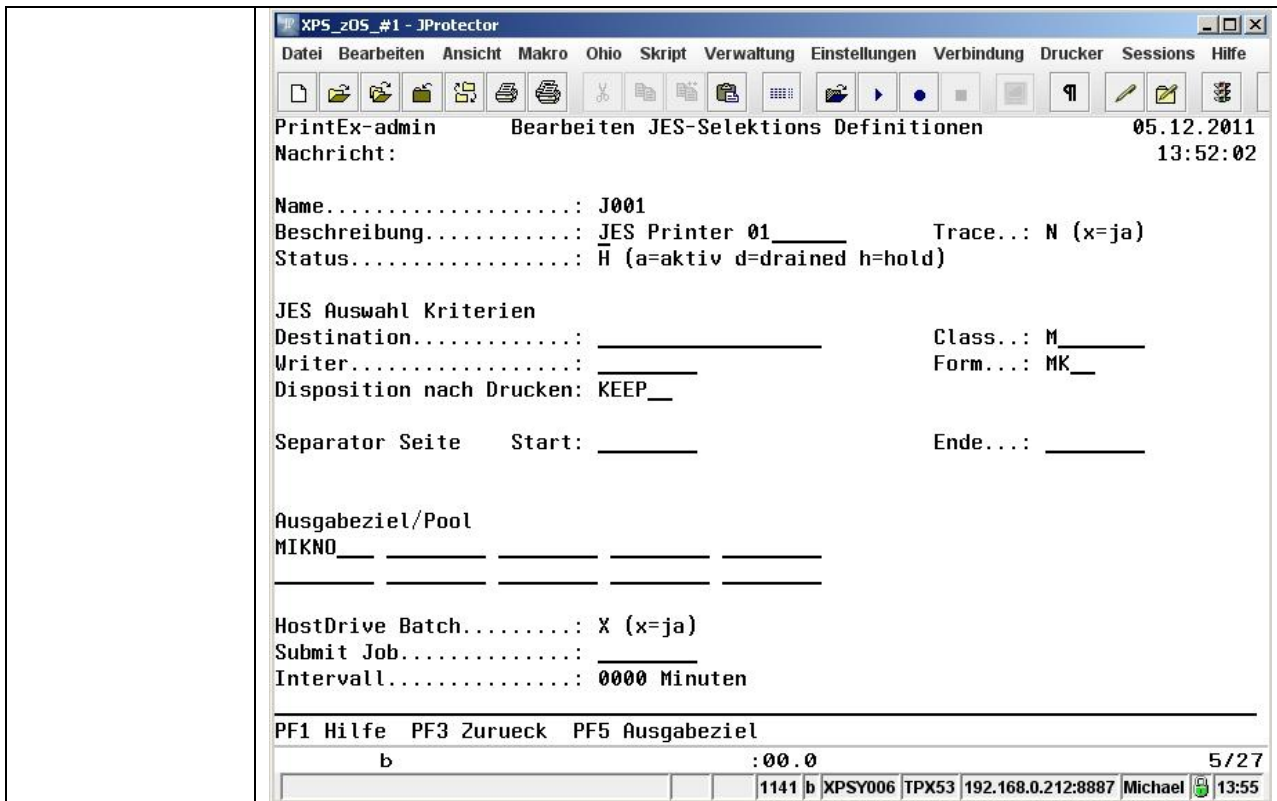


Abb. 57: Definition JES Selektionskriterium

Beispiel | JES=J001

TRACE=	'ALL' 'D' 'T' 'S' 'Z' oder eine, jeweils durch Komma getrennte, Kombination												
	Mit dieser Option können Traceoptionen für den Job festgelegt werden. Der Trace dient zur Ausgabe von Informationen, die bei Bedarf die Analyse des internen Ablaufs ermöglichen. Traceausgaben erfolgen direkt in das Job-Listing. Die folgende Tabelle enthält die verfügbaren Tracetypen:												
	<table border="1"> <thead> <tr> <th>Typ</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>ALL</td> <td>Alle Tracetypen sind aktiviert.</td> </tr> <tr> <td>D</td> <td>Der Datentrace wird aktiviert.</td> </tr> <tr> <td>T</td> <td>Der TCP/IP Trace wird aktiviert.</td> </tr> <tr> <td>S</td> <td>Der Speichertrace wird aktiviert.</td> </tr> <tr> <td>Z</td> <td>Der ZIP/UNZIP Trace wird aktiviert.</td> </tr> </tbody> </table>	Typ	Beschreibung	ALL	Alle Tracetypen sind aktiviert.	D	Der Datentrace wird aktiviert.	T	Der TCP/IP Trace wird aktiviert.	S	Der Speichertrace wird aktiviert.	Z	Der ZIP/UNZIP Trace wird aktiviert.
Typ	Beschreibung												
ALL	Alle Tracetypen sind aktiviert.												
D	Der Datentrace wird aktiviert.												
T	Der TCP/IP Trace wird aktiviert.												
S	Der Speichertrace wird aktiviert.												
Z	Der ZIP/UNZIP Trace wird aktiviert.												
	Mehrere Tracetypen können aktiviert werden, in dem diese, durch Komma getrennt, aufgelistet werden.												
	Beispiel TRACE=D,T												
WTO=	'Y' 'N'												
	Mit dieser Option ist festzulegen, ob zZip beim Jobstart eine Nachricht über die verwendete Version und beim Jobende eine Nachricht über die Anzahl der verarbeiteten Sätze auf die Konsole ausgeben soll oder nicht. Mit einer Angabe kann die globale Voreinstellung – siehe Seite 34 - überschrieben werden.												

POWER – Lesen der Daten aus dem z/VSE Spoolsystem

Mit diesem Steuerkartentyp wird zZip darüber informiert, dass Eingabedaten aus POWER, dem z/VSE Spoolsystem, gelesen werden sollen. Diese Option kann nur unter z/VSE verwendet werden.

Damit zZip POWER Eingabedaten verarbeiten kann, ist die Präsenz einer XPS-PrintEx Basisinstallation erforderlich. PrintEx von XPS ist ein Managementsystem zur Erweiterung der Druckmöglichkeiten unter z/VSE. Die Beschreibung der Installation und Verwaltung von PrintEx ist Gegenstand eines eigenen Handbuchs.

Durch die Verwendung von PrintEx zum Verarbeiten von POWER Dateien steht zZip ein großer Funktionsumfang bzgl. der Verarbeitung von POWER Ausgabedaten zur Verfügung wie z. B. Formulardruck oder die Konvertierung der Daten nach PDF oder Postscript.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE:

```
* $$ JOB JNM=JPOW2NET,DISP=D,PRI=3,CLASS=0
* $$ LST CLASS=T,JNM=JPOW2NET
// JOB JPOW2NET
// LIBDEF PHASE,SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005,SYSLST
// DLBL XPSHFS,'XPSHDRV.SPOOL.SAVE',,VSAM,CAT=XPSUCAT
// EXEC XPSZZIP,SIZE=AUTO,PARM='LIB=XPS.DAEMH600'
I POWER=J001 IPADDR=127.0.0.1 IPPORT=8840
O NET='CEETEMP/%%FILE'
O IPADDR=192.168.0.110 IPPORT=8880
/*
/&
* $$EQJ
```

Abb. 58: Beispieljob für POWER Eingabe z/VSE

Im Beispiel werden alle Power Ausgabedateien, die für das PrintEx Selektionskriterium 'J001' zur Verfügung stehen, gelesen und unter dem von PrintEx generierten Namen im Netzwerk abgelegt.

Im Rahmen der Konfiguration der PrintEx Selektionskriterien ist kenntlich zu machen, dass das gewählte Kriterium – im Beispiel 'J001' – von zZip genutzt wird. Außerdem kann dort definiert werden, dass ein bestimmter Batchjob gestartet werden soll, sobald Daten für das definierte Selektionskriterium bereit stehen. Bei Bedarf kann der Batchjob auch periodisch gestartet werden. Nähere Informationen sind dem PrintEx Handbuch zu entnehmen.

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Eingabekarten des Typs 'I POWER='.

I POWER=	
Lesen von Daten aus dem POWER – nur z/VSE.	
Optionen	Beschreibung
POWER=	PrintEx Selektionskriterium
	An dieser Stelle ist der Schlüssel des PrintEx Selektionskriteriums anzugeben, das die POWER Ausgabedaten, die gelesen werden sollen, näher spezifiziert. Da die Druckdaten von PrintEx verwaltet und aufbereitet werden, stehen für diesen Eingabetyp keine weiteren Optionen zur Verfügung. Nachfolgend eine Abbildung einer mögliche Definition eines PrintEx Selektionskriteriums:

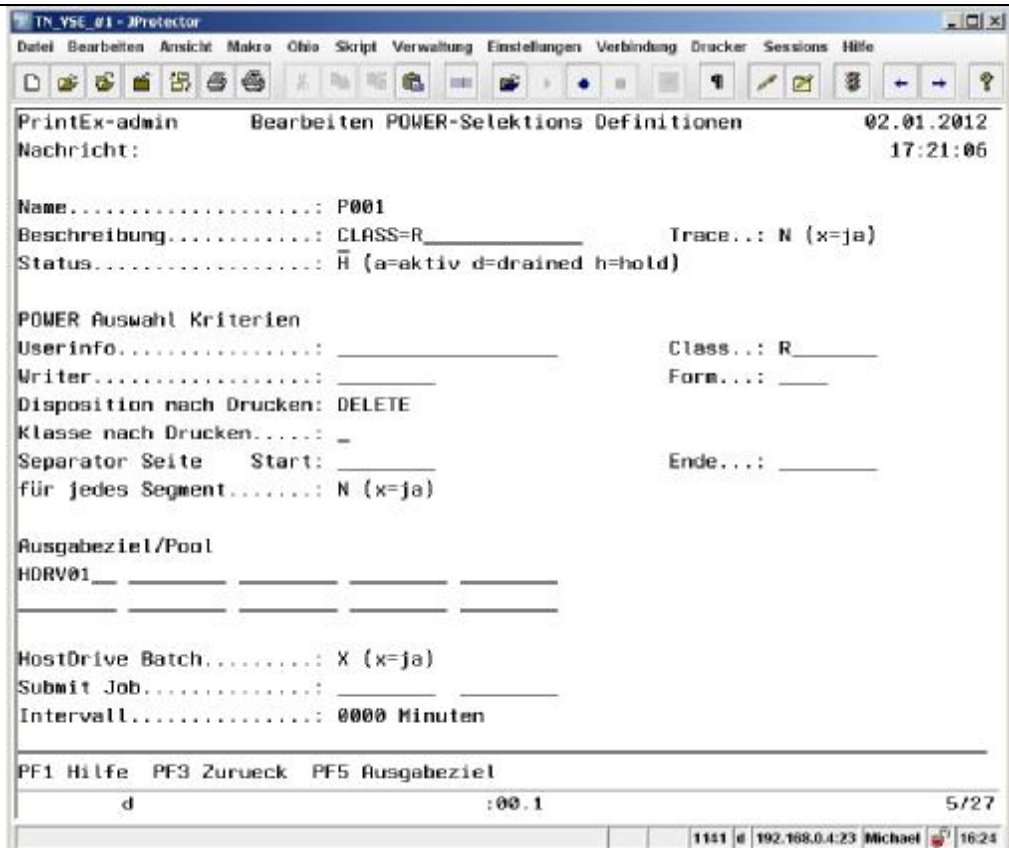


Abb. 59: Definition POWER Selektionskriterium

Beispiel POWER=P001

TRACE=	'ALL' 'D' 'T' 'S' 'Z' oder eine, jeweils durch Komma getrennte, Kombination												
	Mit dieser Option können Traceoptionen für den Job festgelegt werden. Der Trace dient zur Ausgabe von Informationen, die bei Bedarf die Analyse des internen Ablaufs ermöglichen. Traceausgaben erfolgen direkt in das Job-Listing. Die folgende Tabelle enthält die verfügbaren Tracetypen:												
	<table border="1"> <thead> <tr> <th>Typ</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>ALL</td> <td>Alle Tracetypen sind aktiviert.</td> </tr> <tr> <td>D</td> <td>Der Datentrace wird aktiviert.</td> </tr> <tr> <td>T</td> <td>Der TCP/IP Trace wird aktiviert.</td> </tr> <tr> <td>S</td> <td>Der Speichertrace wird aktiviert.</td> </tr> <tr> <td>Z</td> <td>Der ZIP/UNZIP Trace wird aktiviert.</td> </tr> </tbody> </table>	Typ	Beschreibung	ALL	Alle Tracetypen sind aktiviert.	D	Der Datentrace wird aktiviert.	T	Der TCP/IP Trace wird aktiviert.	S	Der Speichertrace wird aktiviert.	Z	Der ZIP/UNZIP Trace wird aktiviert.
Typ	Beschreibung												
ALL	Alle Tracetypen sind aktiviert.												
D	Der Datentrace wird aktiviert.												
T	Der TCP/IP Trace wird aktiviert.												
S	Der Speichertrace wird aktiviert.												
Z	Der ZIP/UNZIP Trace wird aktiviert.												
	Mehrere Tracetypen können aktiviert werden, in dem diese, durch Komma getrennt, aufgelistet werden.												
	Beispiel TRACE=D,T												
WTO=	'Y' 'N'												
	Mit dieser Option ist festzulegen, ob zZip beim Jobstart eine Nachricht über die verwendete Version und beim Jobende eine Nachricht über die Anzahl der verarbeiteten Sätze auf die Konsole ausgeben soll oder nicht. Mit einer Angabe kann die globale Voreinstellung – siehe Seite 34 - überschrieben werden.												

Ausgabedaten

Unter Verwendung einer oder mehrerer 'O'-Karten werden die Ausgabedaten beschrieben, die von zZip erstellt werden sollen.

Pro Ausgabedatei bzw. Ausgabeziel ist eine eigene 'O'-Karte anzugeben, die sich, bei Bedarf oder falls gewünscht, über mehrere Zeilen erstrecken kann.

Das Eintragen von Fortsetzungszeichen auf Stelle 72 ist nicht erforderlich. Jedoch ist auf jeder Karte das 'O'-Kennzeichen auf der ersten Stelle einzufügen, wie der nachfolgenden Abbildung zu entnehmen ist:

```

...
I NET='CEETEMP/SAPDK.DAT' CP=EBC-CRLF FNAME=FILE1
O LIB=XPS.TEST,MBR=OVERLAY.JPG
O FTP='/TEST/rechnung.jpg'
O IPADDR=192.168.0.131 IPPORT=21
O USER='user' PWD='pwd'
...

```

Abb. 60: Beispiel für Ausgabekarten

Wie das Beispiel zeigt, werden zwei Ausgabedateien definiert, was durch die beiden Schlüsselworte 'LIB=' und 'FTP=' angezeigt wird.

Weiterhin ist ersichtlich, dass sich die Beschreibung des zweiten Ausgabeziels über drei aufeinanderfolgende Zeilen erstreckt, jeweils beginnend mit 'O', jedoch ohne die Verwendung eines Fortsetzungszeichens.

Pro zZip Jobstep können mehrere Ausgabedateien erstellt werden.

Nachdem zZip im Rahmen der Verarbeitung zunächst alle Eingabedateien gelesen hat, stellt der diese den nachfolgenden Funktionen zur Erstellung von Ausgabedaten zur Verfügung.

Die folgende Tabelle zeigt eine kurze Übersicht der verfügbaren Ziele für die Ausgabe von Daten:

Ziel	Beschreibung
ARCHIVE	Schreiben der Daten in ein PKZIP-kompatibles Archiv.
LIB	Schreiben der Daten in eine Bibliothek.
NET	Schreiben der Daten auf ein Netzwerklaufwerk.
SEQ	Schreiben der Daten in eine sequentielle Datei.
TAPE	Schreiben der Daten auf Band.
ESDS	Schreiben der Daten in eine VSAM ESDS Datei.
KSDS	Schreiben der Daten in eine VSAM KSDS Datei.
XPSHFS	Schreiben der Daten in das XPS HFS-Dateisystem.
FTP	Versenden der Daten an einen FTP-Server.
HFS	Schreiben der Daten in das OpenMVS HFS-Dateisystem - nur z/OS.

JES	Schreiben der Daten in das z/OS Spoolsystem.
POWER	Schreiben der Daten in das z/VSE Spoolsystem.
EMAIL	Versenden der Daten als E-Mail Anhang.

Es folgt eine Beschreibung der unterstützten Ziele für Ausgabedaten.

ARCHIVE – Schreiben der Daten in ein PKZIP-kompatibles Archiv

zZip kann PKZIP-kompatible Archive erstellen.

Um zZip darüber zu informieren, dass ein PKZIP-kompatibles Archiv erstellt werden soll, ist dem Typschlüssel auf der jeweiligen 'O'-Karte das Schlüsselwort 'ARCHIVE=' voranzustellen.

Beispiel:

Der nachfolgende Auszug aus den Kontrollkarten zeigt die Erstellung einer Textdatei im Netzwerk – Typschlüssel 'NET=':

```
...
O NET= 'BACKUP/RECHNUNGEN/MUSTER20111222.TXT'
...
```

Falls anstatt der Textdatei ein PKZIP-kompatibles Archiv erstellt werden soll, wäre die 'O'-Karte folgendermaßen anzupassen (normalerweise ist ein anderer Dateiname zu wählen):

```
...
O ARCHIVE=NET= 'BACKUP/RECHNUNGEN/ALLE20111222.ZIP'
...
```

An Hand der Angabe von 'O ARCHIVE=NET=' anstelle von 'O NET=' wird zZip mitgeteilt, dass ein komprimiertes ZIP-Archiv erstellt werden soll, was zur Ausführung zusätzlicher Funktionen wie z. B. dem Komprimieren einzelner Dateien, führt.

Die tatsächlich durchzuführende Aktion wird unter Verwendung einer Karte vom Typ 'ACTION=ZIP' festgelegt. Eine Beschreibung erfolgt in Kapitel 'ZIP/UNZIP – Komprimieren/Entkomprimieren von Archiven' auf Seite 57.

Die nachfolgende Tabelle zeigt die zusätzlichen Optionen für Ausgabekarten an, die bei der Verarbeitung von ZIP-Archiven verwendet werden können.

O ARCHIVE=...=	
Erstellen von PKZIP-kompatiblen Archiven.	
Optionen	Beschreibung
CPA2E=	Schlüssel einer Codepage
	Ergänzende Textinformationen wie z. B. Kommentare oder Dateinamen werden in ZIP-Archiven auf Grund von Portabilitätsgründen immer in ASCII/ANSI kodiert. Mit dieser Option kann festgelegt werden, welche Codepage zur Übersetzung dieser Textdaten beim Lesen von ZIP-Archiven verwendet werden soll.

	<p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 23 beschrieben, global vorbelegt werden.</p> <p>Falls unter z/OS auf eine Angabe für diese Option verzichtet wird, werden die ergänzenden Textdaten unter Verwendung der im System eingestellten Codepage übersetzt, die bei Ausführung des Systemmakros 'XLATE' für die Übersetzung von ASCII/ANSI nach EBCDIC zum Einsatz kommt.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>CPA2E=1141-N2E</td> </tr> </table>	Beispiel	CPA2E=1141-N2E
Beispiel	CPA2E=1141-N2E		
CPE2A=	Schlüssel einer Codepage		
	<p>Ergänzende Textinformationen wie z. B. Kommentare oder Dateinamen werden in ZIP-Archiven auf Grund von Portabilitätsgründen immer in ASCII/ANSI kodiert.</p> <p>Mit dieser Option kann festgelegt werden, welche Codepage zur Übersetzung dieser Textdaten beim Schreiben von ZIP-Archiven verwendet werden soll.</p> <p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 23 beschrieben, global vorbelegt werden.</p> <p>Falls unter z/OS auf eine Angabe für diese Option verzichtet wird, werden die ergänzenden Textdaten unter Verwendung der im System eingestellten Codepage übersetzt, die bei Ausführung des Systemmakros 'XLATE' für die Übersetzung von EBCDIC nach ASCII/ANSI zum Einsatz kommt.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>CPE2A=1141-E2N</td> </tr> </table>	Beispiel	CPE2A=1141-E2N
Beispiel	CPE2A=1141-E2N		

LIB - Schreiben der Daten in eine Bibliothek

Mit diesem Steuerkartentyp wird zZip darüber informiert, dass Daten in Bibliotheksmitgliedern (z/OS: Maclib, z/VSE: Library), gespeichert werden sollen.

Bibliotheken, die die Dateien aufnehmen sollen, können zum Einen über die JCL als DD-Statement (z/OS) bzw. als DLBL-Statement (z/VSE) bereitgestellt werden. In diesem Fall ist das entsprechende JCL Statement in der 'O LIB=' Karte zu referenzieren. Zum Anderen besteht die Möglichkeit, die Bibliothek voll qualifiziert direkt in der 'O LIB=' Karte anzugeben.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JFTP2LIB JOB , 'FTP TO LIB', CLASS=A, MSGCLASS=M
//FTP2LIB EXEC PGM=XPSZ ZIP, REGION=48M
//STEPLIB DD DSN=XPSHDRV.V300.LOADLIB, DISP=SHR
//LIBOUT DD DISP=SHR, DSN=ERHARD.TEST.MACLIB
//SYSIN DD *
I FTP='/TEST/RINDE.TXT' CP=1141-N2E
I IPADDR=192.168.0.131
I USER='user' PWD='password'
O LIB=LIBOUT MBR=RINDE
//*
```

Abb. 61: Beispieljob für LIB Ausgabe z/OS

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE:

```
* $$ JOB JNM=JFTP2LIB, DISP=D, PRI=3, CLASS=0
* $$ LST CLASS=T, JNM=JFTP2LIB
// JOB JFTP2LIB
// LIBDEF PHASE, SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
```

```

ASSGN SYS005,SYSLST
// DLBL XPSHFS,'XPSHDRV.SPOOL.SAVE',,VSAM,CAT=XPSUCAT
// EXEC XPSZZIP,SIZE=AUTO,PARM='LIB=XPS.DAEMH600'
I FTP='/TEST1/RINDE.TXT' CP=1141-N2E
I IPADDR=192.168.0.110 IPPORT=21
I USER=&USERID PWD=&PASSWORD RECL=80
O LIB=XPS.HDRV300 MBR=RINDE.D TRACE=ALL
/*
/&
* $$EOJ
    
```

Abb. 62: Beispieljob für LIB Ausgabe z/VSE

Im Job wird eine Datei über FTP eingelesen und in einem Bibliotheksmitglied abgespeichert.

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Karten des Typs 'O LIB='.

O LIB=			
<p>Schreiben von Daten in eine Bibliothek (z/OS: Maclib, z/VSE: Library). Wenn eine Bibliothek als Ausgabeziel für die Verarbeitung eines ZIP- bzw. GZIP-Archivs verwendet wird, kann in einigen Fällen auf die Angabe einer 'O LIB=' Karte verzichtet werden. Dies ist dann der Fall, wenn zur Erstellung des komprimierten Archivs eine Bibliothek verwendet wurde, deren Attribute, d. h. Dateidefinition, unter Verwendung der Option 'ATTR=Y' im Archiv gespeichert wurden, und wenn gewünscht wird, dass diese Bibliothek im Rahmen des Entkomprimiervorgangs von zZip automatisch wieder hergestellt wird.</p>			
Optionen	Beschreibung		
CSV=	beliebiges Zeichen, Standard: <u>Leerzeichen</u>		
	<p>Mit dieser Option ist das Zeichen festzulegen, dass zZip zur Trennung der einzelnen Felder bei der Aufbereitung der Ausgabedaten verwenden soll. Die daraus resultierende csv-Datei (character separated values) kann, falls sie ins Netzwerk versendet wird, mit Standardprogrammen wie z. B. Microsoft Excel angezeigt werden. Damit diese Option verwendet werden kann, ist bei der Definition der Eingabedaten ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich, damit zZip den Satzaufbau kennt.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>CSV=;</td> </tr> </table>	Beispiel	CSV=;
Beispiel	CSV=;		
CSVFMT=	'C' 'T'		
	<p>Mit dieser Option ist das Format der erstellten csv-Daten zu bestimmen. Damit zZip den Satzaufbau kennt, ist bei der Definition der Eingabedatei ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich. zZip unterstützt die Verarbeitung gepackter und hexadezimaler Daten. Folgende Optionen sind verfügbar:</p> <p>'C' Die Inhalte der einzelnen Satzfelder werden in der Weise komprimiert, dass abschließende Leerzeichen entfernt und durch das angegebene csv-Trennzeichen ersetzt werden.</p> <p>'T' Die Inhalte der einzelnen Satzfelder werden tabellarisch aufbereitet, so dass jedes Feld genau so lang dargestellt wird, wie der Definition der Satzstruktur festgelegt. Bei Bedarf werden führende oder abschließende Leerzeichen eingefügt. Schließlich wird jedes Feld mit dem definierten csv-Trennzeichen abgeschlossen.</p>		

<i>CSVHDR=</i>	'Y' 'N'														
	<p>Mit dieser Option ist festzulegen, ob zZip bei der Erstellung einer csv-Datei basierend auf einer Satzstruktur die Namen der Felder als erste Ausgabezeile einfügen soll.</p> <p>Damit zZip den Satzaufbau kennt, ist bei der Definition der Eingabedatei ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich.</p> <p>Folgende Optionen sind verfügbar:</p> <p>'Y' zZip fügt eine Überschriftzeile ein, die die Namen der einzelnen Felder der Satzstruktur, getrennt durch das mit 'CSV=' angegebene Zeichen, enthält.</p> <p>'N' zZip fügt keine Überschriftzeile ein.</p>														
<i>LIB=</i>	<p>z/OS: Verweis auf das JCL DD-Statement der Bibliothek oder voll qualifizierter Name der Bibliothek</p> <p>z/VSE: Name der VSE Library</p>														
	<p>Mit dieser Option ist die Bibliothek festzulegen, in der das neue Member erstellt werden soll.</p> <table border="1"> <tr> <td>Beispiele</td> <td>z/OS</td> <td>LIB=LIBOUT LIB=XPS.HDRV.OUT</td> </tr> <tr> <td></td> <td>z/VSE</td> <td>LIB=ERHARD.TEST</td> </tr> </table>	Beispiele	z/OS	LIB=LIBOUT LIB=XPS.HDRV.OUT		z/VSE	LIB=ERHARD.TEST								
Beispiele	z/OS	LIB=LIBOUT LIB=XPS.HDRV.OUT													
	z/VSE	LIB=ERHARD.TEST													
<i>MBR=</i>	Name														
	Mit dieser Option ist der Name des zu erstellenden Members anzugeben.														
<i>TRACE=</i>	'ALL' 'D' 'T' 'S' 'Z' oder eine, jeweils durch Komma getrennte, Kombination														
	<p>Mit dieser Option können Traceoptionen für den Job festgelegt werden. Der Trace dient zur Ausgabe von Informationen, die bei Bedarf die Analyse des internen Ablaufs ermöglichen. Traceausgaben erfolgen direkt in das Job-Listing. Die folgende Tabelle enthält die verfügbaren Tracetypen:</p> <table border="1"> <thead> <tr> <th>Typ</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td><i>ALL</i></td> <td>Alle Tracetypen sind aktiviert.</td> </tr> <tr> <td><i>D</i></td> <td>Der Datentrace wird aktiviert.</td> </tr> <tr> <td><i>T</i></td> <td>Der TCP/IP Trace wird aktiviert.</td> </tr> <tr> <td><i>S</i></td> <td>Der Speichertrace wird aktiviert.</td> </tr> <tr> <td><i>Z</i></td> <td>Der ZIP/UNZIP Trace wird aktiviert.</td> </tr> </tbody> </table> <p>Mehrere Tracetypen können aktiviert werden, in dem diese, durch Komma getrennt, aufgelistet werden.</p> <table border="1"> <tr> <td>Beispiel</td> <td>TRACE=D,T</td> </tr> </table>	Typ	Beschreibung	<i>ALL</i>	Alle Tracetypen sind aktiviert.	<i>D</i>	Der Datentrace wird aktiviert.	<i>T</i>	Der TCP/IP Trace wird aktiviert.	<i>S</i>	Der Speichertrace wird aktiviert.	<i>Z</i>	Der ZIP/UNZIP Trace wird aktiviert.	Beispiel	TRACE=D,T
Typ	Beschreibung														
<i>ALL</i>	Alle Tracetypen sind aktiviert.														
<i>D</i>	Der Datentrace wird aktiviert.														
<i>T</i>	Der TCP/IP Trace wird aktiviert.														
<i>S</i>	Der Speichertrace wird aktiviert.														
<i>Z</i>	Der ZIP/UNZIP Trace wird aktiviert.														
Beispiel	TRACE=D,T														
<i>WTO=</i>	'Y' 'N'														
	Mit dieser Option ist festzulegen, ob zZip beim Jobstart eine Nachricht über die verwendete Version und beim Jobende eine Nachricht über die Anzahl der verarbeiteten Sätze auf die Konsole ausgeben soll oder nicht. Mit einer Angabe kann die globale Voreinstellung – siehe Seite 34 - überschrieben werden.														

NET - Schreiben der Daten auf ein Netzwerklaufwerk

Mit diesem Steuerkartentyp wird zZip darüber informiert, dass Daten in eine Datei auf einem Netzwerklaufwerk gespeichert werden sollen.

Damit zZip Netzwerkdateien erstellen kann, ist es erforderlich, dass die Netzwerkkomponente des HostDrive Systems, der HostDrive/J Server, installiert ist und auf das gewünschte Laufwerk zugreifen kann. Die Installation und Konfiguration des HostDrive/J Servers ist in einem eigenen Handbuch dokumentiert.

Für die Kommunikation mit dem HostDrive/J Server stehen zZip zwei verschiedene Möglichkeiten zur Verfügung:

1. Direkte Kommunikation.
2. Indirekte Kommunikation über eine HostDrive/z Basisinstallation.

Die Unterscheidung erfolgt über die Angabe einer 'ROUTE=' Option auf der 'O NET=' Karte. Wenn diese Angabe fehlt, kommuniziert zZip direkt mit dem HostDrive/J Server.

Wenn die 'ROUTE=' Option angegeben ist, erfolgt die Kommunikation indirekt über eine HostDrive/z Basisinstallation. In diesem Fall verwendet die HostDrive/z Basisinstallation die angegebene Route zur Identifikation des HostDrive/J Servers, der die Eingabedaten liefern soll.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JKSD2NET JOB , 'KSDS TO NET', CLASS=A, MSGCLASS=M
//KSD2NET EXEC PGM=XPSZZIP, REGION=96M
//STEPLIB DD DSN=XPSHDRV.V300.LOADLIB, DISP=SHR
//XPSADVW DD DISP=SHR, DSN=VSAM.ADVW.V500.FILE
//SYSIN DD *
I KSDS=XPSADVW RECORD=XPS/ADVW2
R LAST-NAME FIRST-NAME ZIP CITY
O NET='ZOSSOUT/ADRESSEN.CSV' CP=1141-E2N WTO=Y CRLF
O IPADDR=192.168.0.131 IPPORT=8841 CSV=; CSVHDR=Y CSVFMT=T
/*
```

Abb. 63: Beispieljob für NET Ausgabe z/OS

Im Beispiel werden Sätze aus der KSDS Datei 'XPSADVW' gelesen und als Report direkt an einen HostDrive/J Server zur Speicherung in einer Netzwerkdatei gesendet.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE:

```
* $$ JOB JNM=JKSD2NET, DISP=D, PRI=3, CLASS=0
* $$ LST CLASS=T
// JOB JKSD2NET
// LIBDEF PHASE, SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005, FEE
// DLBL XPSADVW, 'ADVW.V500.FILE', , VSAM, CAT=VSESPUC
// DLBL XPSHFS, 'XPSHDRV.SPOOL.SAVE', , VSAM, CAT=XPSUCAT
// EXEC XPSZZIP, SIZE=AUTO, PARM='LIB=XPS.DAEMH600'
I KSDS=XPSADVW KEY=MEIER OPT=EQ RECORD=XPS/ADVW
R LAST-NAME FIRST-NAME ZIP CITY
O NET='CEETEMP/ADRESSEN.CSV' CP=1141-E2N WTO=Y CRLF
O IPADDR=192.168.0.110 IPPORT=8880 CSV=; CSVHDR=Y CSVFMT=N
/*
```



```

/&
* $$EOJ

```

Abb. 64: Beispieljob für NET Ausgabe z/VSE

Im Beispiel werden ebenfalls alle Sätze aus der KSDS Datei 'XPSADVW' gelesen und als Report an einen HostDrive/J Server zur Speicherung in einer Netzwerkdatei gesendet. Allerdings erfolgt das Versenden in diesem Fall indirekt über eine HostDrive/z Basisinstallation, was an der Angabe für 'ROUTE=' ersichtlich ist.

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Karten des Typs 'O NET='.

O NET=			
Schreiben von Daten in eine Netzwerkdatei.			
Optionen	Beschreibung		
CSV=	beliebiges Zeichen, Standard: <u>Leerzeichen</u>		
	<p>Mit dieser Option ist das Zeichen festzulegen, das zZip zur Trennung der einzelnen Felder bei der Aufbereitung der Ausgabedaten verwenden soll. Die daraus resultierende csv-Datei (character separated values) kann, falls sie ins Netzwerk versendet wird, mit Standardprogrammen wie z. B. Microsoft Excel angezeigt werden.</p> <p>Damit diese Option verwendet werden kann, ist bei der Definition der Eingabedaten ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich, damit zZip den Satzaufbau kennt.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>CSV=;</td> </tr> </table>	Beispiel	CSV=;
Beispiel	CSV=;		
CSVFMT=	'C' 'T'		
	<p>Mit dieser Option ist das Format der erstellten csv-Daten zu bestimmen.</p> <p>Damit zZip den Satzaufbau kennt, ist bei der Definition der Eingabedatei ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich.</p> <p>zZip unterstützt die Verarbeitung gepackter und hexadezimaler Daten.</p> <p>Folgende Optionen sind verfügbar:</p> <p>'C' Die Inhalte der einzelnen Satzfelder werden in der Weise komprimiert, dass abschließende Leerzeichen entfernt und durch das angegebene csv-Trennzeichen ersetzt werden.</p> <p>'T' Die Inhalte der einzelnen Satzfelder werden tabellarisch aufbereitet, so dass jedes Feld genau so lang dargestellt wird, wie der Definition der Satzstruktur festgelegt. Bei Bedarf werden führende oder abschließende Leerzeichen eingefügt. Schließlich wird jedes Feld mit dem definierten csv-Trennzeichen abgeschlossen.</p>		
CSVHDR=	'Y' 'N'		
	<p>Mit dieser Option ist festzulegen, ob zZip bei der Erstellung einer csv-Datei basierend auf einer Satzstruktur die Namen der Felder als erste Ausgabezeile einfügen soll.</p> <p>Damit zZip den Satzaufbau kennt, ist bei der Definition der Eingabedatei ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich.</p> <p>Folgende Optionen sind verfügbar:</p> <p>'Y' zZip fügt eine Überschriftzeile ein, die die Namen der einzelnen Felder der Satzstruktur, getrennt durch das mit 'CSV=' angegebene Zeichen, enthält.</p>		

	'N' zZip fügt keine Überschriftzeile ein.	
IPADDR=	TCP/IP Adresse – numerisch oder als DNS-Name	
	<p>Mit dieser Option ist die TCP/IP Adresse anzugeben, an die zZip die Daten ausliefern soll.</p> <p>Diese Angabe kann sich entweder direkt auf einen HostDrive/J beziehen oder auf eine HostDrive/z Basisinstallation. Im zweiten Fall ist die Angabe einer 'ROUTE=' Option erforderlich, mit der dem HostDrive/z Server mitgeteilt wird, über welche TCP/IP Adresse der HostDrive/J Server erreichbar ist.</p> <p>Falls keine Angabe für diese Option erfolgt, verwendet zZip die TCP/IP Adresse, die als Vorbelegung in der Initialisierungsdatei – siehe Seite 27 - hinterlegt ist.</p>	
	Beispiel	IPADDR=127.0.0.1
IPPORT=	numerischer TCP/IP Port <= 65535	
	<p>Mit dieser Option ist der TCP/IP Port anzugeben, der von dem - mit 'IPADDR=' adressierten - HostDrive Server für eingehende Anfragen überwacht wird.</p> <p>Falls keine Angabe für diese Option erfolgt, verwendet zZip den TCP/IP Port, der als Vorbelegung in der Initialisierungsdatei – siehe Seite 28 - hinterlegt ist.</p>	
	Beispiel	IPPORT=8841
NET=	Pfadangabe zur Datei im HostDrive/J Format	
	<p>Die Pfadangabe muss mit der Angabe des Namens eines gültigen Mapping-Kanals auf dem adressierten HostDrive/J Server beginnen. Mapping-Kanäle dienen als symbolische Platzhalter für physische Pfade im Netzwerk. Nähere Informationen hierzu sind dem HostDrive Handbuch zu entnehmen.</p>	
	Beispiel	NET=ZOUT/SUBDIR/FILE.EXT
ROUTE=	Schlüssel einer HostDrive/z Route Definition	
	<p>Mit dieser Option ist der Name einer HostDrive/z Route Definition anzugeben, in der weitere TCP/IP Informationen über einen angeschlossenen HostDrive/J Server hinterlegt sind.</p> <p>Diese Angabe ist nur erforderlich, wenn zZip die Netzwerkdatei nicht direkt von einem HostDrive/J Server lesen soll, sondern indirekt über einen HostDrive/z Server.</p> <p>Definition und Verwaltung von HostDrive/z Routen sind im HostDrive/z Handbuch beschrieben.</p>	
	Beispiel	ROUTE=WS08
TRACE=	'ALL' 'D' 'T' 'S' 'Z' oder eine, jeweils durch Komma getrennte, Kombination	
	<p>Mit dieser Option können Traceoptionen für den Job festgelegt werden. Der Trace dient zur Ausgabe von Informationen, die bei Bedarf die Analyse des internen Ablaufs ermöglichen. Traceausgaben erfolgen direkt in das Job-Listing. Die folgende Tabelle enthält die verfügbaren Tracetypen:</p>	
	Typ	Beschreibung
	ALL	Alle Tracetypen sind aktiviert.
	D	Der Datentrace wird aktiviert.
	T	Der TCP/IP Trace wird aktiviert.
	S	Der Speichertrace wird aktiviert.

	Z	Der ZIP/UNZIP Trace wird aktiviert.
		Mehrere Tracetypen können aktiviert werden, in dem diese, durch Komma getrennt, aufgelistet werden.
	Beispiel	TRACE=D,T
WTO=	'Y' 'N'	
		Mit dieser Option ist festzulegen, ob zZip beim Jobstart eine Nachricht über die verwendete Version und beim Jobende eine Nachricht über die Anzahl der verarbeiteten Sätze auf die Konsole ausgeben soll oder nicht. Mit einer Angabe kann die globale Voreinstellung – siehe Seite 34 - überschrieben werden.

SEQ - Schreiben der Daten in eine sequentielle Datei

Mit diesem Steuerkartentyp wird zZip darüber informiert, dass Daten in einer sequentiellen Datei gespeichert werden sollen.

Sequentielle Dateien, die Daten aufnehmen sollen, können zum Einen über die JCL als DD-Statement (z/OS) bzw. als DLBL-Statement (z/VSE) bereitgestellt werden. In diesem Fall ist das entsprechende JCL Statement in der 'O SEQ=' Karte zu referenzieren. Zum Anderen besteht die Möglichkeit, die sequentielle Datei voll qualifiziert direkt in der 'O SEQ=' Karte anzugeben.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JNET2SEQ JOB , 'NET TO SEQ' ,CLASS=A,MSGCLASS=M
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE XPSHDRV.SEQ
//NET2SEQ EXEC PGM=XPSZZIP,REGION=48M
//STEPLIB DD DSN=XPSHDRV.V300.LOADLIB,DISP=SHR
//XPSSEQ DD DSN=XPSHDRV.SEQ,DISP=(NEW,CATLG),
// SPACE=(TRK,(5,3),RLSE),UNIT=3390,VOL=SER=XPS005,
// DCB=(LRECL=100,BLKSIZE=1000,RECFM=FB,DSORG=PS)
//SYSIN DD *
I NET= '/TEST/KOMPONISTEN.TXT' CP=1141-N2E
I IPADDR=192.168.0.131 IPPORT=8841
O SEQ=XPSSEQ
//*
```

Abb. 65: Beispieljob für SEQ Ausgabe z/OS

Im Beispiel wird die Datei 'KOMPONISTEN.TXT' von einem HostDrive/J Server gelesen und in der sequentiellen Datei, die durch das 'XPSSEQ' DD-Statement definiert wird, gespeichert.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE mit identischem Ablauf.

```
* $$ JOB JNM=JNET2SEQ,DISP=D,PRI=3,CLASS=0
* $$ LST CLASS=T
// JOB JNET2SEQ
// LIBDEF PHASE,SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005,FEE
// DLBL XPSSEQ, 'HDRV.SEQ.FILE2', ,VSAM,DISP=(NEW,KEEP), X
RECORDS=(500,50),RECSIZE=80
// DLBL XPSHFS, 'XPSHDRV.SPOOL.SAVE', ,VSAM,CAT=XPSUCAT
// EXEC XPSZZIP,SIZE=AUTO,PARAM='LIB=XPS.DAEMH600'
I NET='CEETEMP/KOMPONISTEN.TXT' CP=1141-N2E
I IPADDR=192.168.0.110 IPPORT=8880
O SEQ=XPSSEQ RECL=80
/*
/&
* $$EQJ
```

Abb. 66: Beispieljob für SEQ Ausgabe z/VSE

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Karten des Typs 'O SEQ='.

O SEQ=

Schreiben von Daten in eine sequentielle Datei.

<p>Wenn eine sequentielle Datei als Ausgabeziel für die Verarbeitung eines ZIP- bzw. GZIP-Archivs verwendet wird, kann in einigen Fällen auf die Angabe einer 'O SEQ=' Karte verzichtet werden.</p> <p>Dies ist dann der Fall, wenn zur Erstellung des komprimierten Archivs eine sequentielle Datei verwendet wurde, deren Attribute, d. h. Dateidefinition, unter Verwendung der Option 'ATTR=Y' im Archiv gespeichert wurden, und wenn gewünscht wird, dass diese sequentielle Datei im Rahmen des Entkomprimiervorgangs von zZip automatisch wieder hergestellt wird.</p>			
Optionen	Beschreibung		
CSV=	beliebiges Zeichen, Standard: <u>Leerzeichen</u>		
	<p>Mit dieser Option ist das Zeichen festzulegen, das zZip zur Trennung der einzelnen Felder bei der Aufbereitung der Ausgabedaten verwenden soll. Die daraus resultierende csv-Datei (character separated values) kann, falls sie ins Netzwerk versendet wird, mit Standardprogrammen wie z. B. Microsoft Excel angezeigt werden.</p> <p>Damit diese Option verwendet werden kann, ist bei der Definition der Eingabedaten ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich, damit zZip den Satzaufbau kennt.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>CSV=;</td> </tr> </table>	Beispiel	CSV=;
Beispiel	CSV=;		
CSVFMT=	'C' 'T'		
	<p>Mit dieser Option ist das Format der erstellten csv-Daten zu bestimmen.</p> <p>Damit zZip den Satzaufbau kennt, ist bei der Definition der Eingabedatei ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich.</p> <p>zZip unterstützt die Verarbeitung gepackter und hexadezimaler Daten.</p> <p>Folgende Optionen sind verfügbar:</p> <p>'C' Die Inhalte der einzelnen Satzfelder werden in der Weise komprimiert, dass abschließende Leerzeichen entfernt und durch das angegebene csv-Trennzeichen ersetzt werden.</p> <p>'T' Die Inhalte der einzelnen Satzfelder werden tabellarisch aufbereitet, so dass jedes Feld genau so lang dargestellt wird, wie der Definition der Satzstruktur festgelegt. Bei Bedarf werden führende oder abschließende Leerzeichen eingefügt. Schließlich wird jedes Feld mit dem definierten csv-Trennzeichen abgeschlossen.</p>		
CSVHDR=	'Y' 'N'		
	<p>Mit dieser Option ist festzulegen, ob zZip bei der Erstellung einer csv-Datei basierend auf einer Satzstruktur die Namen der Felder als erste Ausgabezeile einfügen soll.</p> <p>Damit zZip den Satzaufbau kennt, ist bei der Definition der Eingabedatei ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich.</p> <p>Folgende Optionen sind verfügbar:</p> <p>'Y' zZip fügt eine Überschriftzeile ein, die die Namen der einzelnen Felder der Satzstruktur, getrennt durch das mit 'CSV=' angegebene Zeichen, enthält.</p> <p>'N' zZip fügt keine Überschriftzeile ein.</p>		
RECFM=	'F' 'FB' 'U' 'V' 'VB'		
	<p>Mit dieser Angabe ist das Recordformat der sequentiellen Datei zu bestimmen.</p> <p>Diese Angabe ist unter z/VSE immer erforderlich, wenn sie nicht dem Standard 'RECFM=FB' entspricht.</p> <p>Unter z/OS ist die Angabe nur dann erforderlich, wenn zZip das Recordformat aus der Job</p>		

	<p>Control nicht automatisch ermitteln kann.</p> <p>Die anzugebenden Werte entsprechen den Originalwerten der Job Control:</p> <p>'F' fixed</p> <p>'FB' fixed blocked</p> <p>'U' undefined</p> <p>'V' variable</p> <p>'VB' variable blocked</p>	
RECL=	numerischer Wert	
	<p>Mit dieser Option ist die Satzlänge der Daten auf der sequentiellen Datei anzugeben.</p> <p>Diese Angabe ist unter z/VSE immer und unter z/OS nur dann erforderlich, wenn zZip die Satzlänge nicht automatisch aus der Job Control ermitteln kann.</p>	
	Beispiel	RECL=80
SEQ=	<p>z/OS: Verweis auf das JCL DD-Statement der sequentiellen Datei oder voll qualifizierter Name der sequentiellen Datei</p> <p>z/VSE: Verweis auf das JCL DD-Statement der sequentiellen Datei</p>	
	<p>Die Angabe muss unter z/OS auf ein DD-Statement und unter z/VSE auf ein DLBL-Statement verweisen, mit dem die sequentielle Datei definiert wird. Alternativ kann unter z/OS der voll qualifizierte Name der sequentiellen Datei angegeben werden.</p> <p><u>Ausnahme unter z/OS:</u></p> <p>Falls die sequentielle Datei unter z/OS aus einem zuvor von zZip erstellten komprimierten Archiv (ACTION=ZIP bzw. ACTION=GZIP) einer sequentiellen Datei wiederhergestellt wird, können die im Archiv gespeicherten Attribute zur Erstellung der sequentiellen Datei mit den originalen Werten z. B. für SPACE oder RECFM verwendet werden.</p> <p>In diesem Fall ist auf die Angabe einer Output-Karte zu verzichten.</p> <p>Eine Beschreibung der Vorgehensweise zur Erstellung von ZIP-Archiven erfolgt auf Seite 57.</p>	
	Beispiel	SEQ=XPSSEQ
TRACE=	'ALL' 'D' 'T' 'S' 'Z' oder eine, jeweils durch Komma getrennte, Kombination	
	<p>Mit dieser Option können Traceoptionen für den Job festgelegt werden. Der Trace dient zur Ausgabe von Informationen, die bei Bedarf die Analyse des internen Ablaufs ermöglichen. Traceausgaben erfolgen direkt in das Job-Listing. Die folgende Tabelle enthält die verfügbaren Tracetypen:</p>	
	Typ	Beschreibung
	ALL	Alle Tracetypen sind aktiviert.
	D	Der Datentrace wird aktiviert.
	T	Der TCP/IP Trace wird aktiviert.
	S	Der Speichertrace wird aktiviert.
	Z	Der ZIP/UNZIP Trace wird aktiviert.
	<p>Mehrere Tracetypen können aktiviert werden, in dem diese, durch Komma getrennt, aufgelistet werden.</p>	
	Beispiel	TRACE=S

<i>TRUNC=</i>	'Y' ' <u>N</u> '
	<p>Mit dieser Angabe kann bestimmt werden, wie mit Zeilen verfahren werden soll, die länger als die mit der Option 'RECL=' festgelegte Satzlänge sind.</p> <p>Bei Auswahl von 'TRUNC=Y' werden überschüssige Zeichen abgeschnitten und gehen verloren.</p> <p>Bei der Standardeinstellung 'TRUNC=N' werden überschüssige Zeichen in die nächste Zeile umgebrochen und bei Bedarf bis zur festgelegten Satzlänge mit dem Füllzeichen aufgefüllt.</p>
<i>WTO=</i>	'Y' ' <u>N</u> '
	<p>Mit dieser Option ist festzulegen, ob zZip beim Jobstart eine Nachricht über die verwendete Version und beim Jobende eine Nachricht über die Anzahl der verarbeiteten Sätze auf die Konsole ausgeben soll oder nicht. Mit einer Angabe kann die globale Voreinstellung – siehe Seite 34 - überschrieben werden.</p>

TAPE - Schreiben der Daten auf Band

Mit diesem Steuerkartentyp wird zZip darüber informiert, dass Daten auf einem virtuellen oder physischen Band gespeichert werden sollen.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JLIB2TAP JOB , 'LIB TO TAPE' ,CLASS=A,MSGCLASS=M
//DELTAPE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE XPS.LIB.TAPE.BACKUP
/*
//LIB2TAP EXEC PGM=XPSZZIP,REGION=96M
//STEPLIB DD DSN=XPSHDRV.V300.LOADLIB,DISP=SHR
//TAP DD DISP=(NEW,CATLG),DSN=XPS.LIB.TAPE.BACKUP,
// SPACE=(65000,(10000,200)),VOL=SER=XPS009,
// UNIT=3390,DCB=(RECFM=VB,BLKSIZE=81964,LRECL=8196)
//XPSLIB DD DISP=SHR,DSN=XPSHDRV.TEST.MACLIB
//SYSIN DD *
I LIB=XPSLIB MBR=*
O TAPE=TAP CP=1141-E2N
/*
```

Abb. 67: Beispieljob für TAPE Ausgabe z/OS

Im Beispiel werden alle Member der z/OS Maclib 'XPSHDRV.TEST.MACLIB' sequentiell und ohne Trenner hintereinander auf das Band geschrieben. Dabei findet eine Umsetzung der Zeichen nach ANSI statt.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE mit vergleichbarem Ablauf.

```
* $$ JOB JNM=JLIB2TAP,DISP=D,PRI=3,CLASS=0
* $$ LST CLASS=T
// JOB JTAP2EML
// VTAPE STOP,UNIT=590
// VTAPE START,UNIT=590,LOC=192.168.0.110:8880,FILE='C:/VTAPE/TEST.AWS'
// LIBDEF PHASE,SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// DLBL XPSHFS,'XPSHDRV.SPOOL.SAVE',,VSAM,CAT=XPSUCAT
// TLBL TAP
// ASSGN SYS005,SYSLST
// ASSGN SYS010,590
// EXEC XPSZZIP,SIZE=AUTO,PARM='LIB=XPS.DAEMH600'
I LIB=XPS.HDRV300 MBR=INSTALL.PROC
O TAPE=TAP RECFM=FB BLKSIZE=800 RECL=80 LABEL=N
/*
/&
// VTAPE STOP,UNIT=590
* $$ EQJ
```

Abb. 68: Beispieljob für TAPE Ausgabe z/VSE

Unter z/VSE ist das Band, wenn es als Ausgabemedium eingesetzt wird, immer über ein Assignment an 'SYS011' anzubinden.

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Karten des Typs 'O TAPE='.

O TAPE=

Schreiben von Daten auf Band.			
Optionen	Beschreibung		
<i>BLKSIZE=</i>	numerischer Wert		
	<p>Mit dieser Option ist zZip über die Blocksize zu informieren, mit der die Daten auf das Band geschrieben werden sollen.</p> <p>Diese Angabe ist unter z/VSE immer erforderlich und unter z/OS nur dann, wenn die Information nicht über das DD-Statement für das Band ermittelbar ist.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>BLKSIZE=8200</td> </tr> </table>	Beispiel	BLKSIZE=8200
Beispiel	BLKSIZE=8200		
<i>CSV=</i>	beliebiges Zeichen, Standard: <u>Leerzeichen</u>		
	<p>Mit dieser Option ist das Zeichen festzulegen, das zZip zur Trennung der einzelnen Felder bei der Aufbereitung der Ausgabedaten verwenden soll. Die daraus resultierende csv-Datei (character separated values) kann, falls sie ins Netzwerk versendet wird, mit Standardprogrammen wie z. B. Microsoft Excel angezeigt werden.</p> <p>Damit diese Option verwendet werden kann, ist bei der Definition der Eingabedaten ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich, damit zZip den Satzaufbau kennt.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>CSV=;</td> </tr> </table>	Beispiel	CSV=;
Beispiel	CSV=;		
<i>CSVFMT=</i>	' <u>C</u> ' ' <u>T</u> '		
	<p>Mit dieser Option ist das Format der erstellten csv-Daten zu bestimmen.</p> <p>Damit zZip den Satzaufbau kennt, ist bei der Definition der Eingabedatei ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich.</p> <p>zZip unterstützt die Verarbeitung gepackter und hexadezimaler Daten.</p> <p>Folgende Optionen sind verfügbar:</p> <p>'<u>C</u>' Die Inhalte der einzelnen Satzfelder werden in der Weise komprimiert, dass abschließende Leerzeichen entfernt und durch das angegebene csv-Trennzeichen ersetzt werden.</p> <p>'<u>T</u>' Die Inhalte der einzelnen Satzfelder werden tabellarisch aufbereitet, so dass jedes Feld genau so lang dargestellt wird, wie der Definition der Satzstruktur festgelegt. Bei Bedarf werden führende oder abschließende Leerzeichen eingefügt. Schließlich wird jedes Feld mit dem definierten csv-Trennzeichen abgeschlossen.</p>		
<i>CSVHDR=</i>	' <u>Y</u> ' ' <u>N</u> '		
	<p>Mit dieser Option ist festzulegen, ob zZip bei der Erstellung einer csv-Datei basierend auf einer Satzstruktur die Namen der Felder als erste Ausgabezeile einfügen soll.</p> <p>Damit zZip den Satzaufbau kennt, ist bei der Definition der Eingabedatei ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich.</p> <p>Folgende Optionen sind verfügbar:</p> <p>'<u>Y</u>' zZip fügt eine Überschriftzeile ein, die die Namen der einzelnen Felder der Satzstruktur, getrennt durch das mit 'CSV=' angegebene Zeichen, enthält.</p> <p>'<u>N</u>' zZip fügt keine Überschriftzeile ein.</p>		
<i>LABEL=</i>	' <u>N</u> ' ' <u>S</u> '		

	<p>Mit dieser Option ist zZip darüber zu informieren, ob bei der Erstellung des Bandes ein Label geschrieben werden soll.</p> <p>Folgende Angaben sind möglich:</p> <p>'N' das Band erhält keinen Label / Voreinstellung</p> <p>'S' das Band erhält einen Standardlabel</p>		
RECFM=	'F' 'FB' 'V' 'VB'		
	<p>Mit dieser Angabe ist das Recordformat des erstellten Bandes zu bestimmen.</p> <p>Diese Angabe ist unter z/VSE immer erforderlich, wenn sie nicht dem Standard 'RECFM=FB' entspricht.</p> <p>Unter z/OS ist die Angabe nur dann erforderlich, wenn zZip das Recordformat aus der Job Control nicht automatisch ermitteln kann.</p> <p>Die anzugebenden Werte entsprechen den Originalwerten der Job Control:</p> <p>'F' fixed</p> <p>'FB' fixed blocked</p> <p>'V' variable</p> <p>'VB' variable blocked</p>		
RECL=	numerischer Wert		
	<p>Mit dieser Option ist die Satzlänge anzugeben, mit der die Daten auf Band geschrieben werden sollen.</p> <p>Diese Angabe ist unter z/VSE immer und unter z/OS nur dann erforderlich, wenn zZip die Satzlänge nicht automatisch aus der Job Control ermitteln kann.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>RECL=8196</td> </tr> </table>	Beispiel	RECL=8196
Beispiel	RECL=8196		
REW=	'N' 'R' 'U'		
	<p>Mit dieser Option ist anzugeben, wie mit dem Band nach Abschluss des Schreibens verfahren werden soll.</p> <p>Folgende Angaben sind möglich:</p> <p>'N' das Band wird nach Abschluss des Lesens nicht zurückgespult</p> <p>'R' das Band wird nach Abschluss des Lesens zurückgespult / Voreinstellung</p> <p>'U' das Band wird nach Abschluss des Lesens entladen</p>		
TAPE=	Verweis auf das JCL-Statement des Bandes		
	<p>Die Angabe muss unter z/OS auf ein DD-Statement und unter z/VSE auf ein TLBL-Statement verweisen, mit dem das Band definiert wird.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>TAPE=TAP</td> </tr> </table>	Beispiel	TAPE=TAP
Beispiel	TAPE=TAP		
TRACE=	'ALL' 'D' 'T' 'S' 'Z' oder eine, jeweils durch Komma getrennte, Kombination		
	<p>Mit dieser Option können Traceoptionen für den Job festgelegt werden. Der Trace dient zur Ausgabe von Informationen, die bei Bedarf die Analyse des internen Ablaufs ermöglichen. Traceausgaben erfolgen direkt in das Job-Listing. Die folgende Tabelle enthält die verfügbaren Tracetypen:</p> <table border="1"> <thead> <tr> <th>Typ</th> <th>Beschreibung</th> </tr> </thead> </table>	Typ	Beschreibung
Typ	Beschreibung		

	<i>ALL</i>	Alle Tracetypen sind aktiviert.
	<i>D</i>	Der Datentrace wird aktiviert.
	<i>T</i>	Der TCP/IP Trace wird aktiviert.
	<i>S</i>	Der Speichertrace wird aktiviert.
	<i>Z</i>	Der ZIP/UNZIP Trace wird aktiviert.
	Mehrere Tracetypen können aktiviert werden, in dem diese, durch Komma getrennt, aufgelistet werden.	
	Beispiel	TRACE=S
<i>TRUNC=</i>	'Y' 'N'	
	<p>Mit dieser Angabe kann bestimmt werden, wie mit Zeilen verfahren werden soll, die länger als die mit der Option 'RECL=' festgelegte Satzlänge sind, wenn mit einem fixen Recordformat gearbeitet wird.</p> <p>Bei Auswahl von 'TRUNC=Y' werden überschüssige Zeichen abgeschnitten und gehen verloren.</p> <p>Bei der Standardeinstellung 'TRUNC=N' werden überschüssige Zeichen in die nächste Zeile umgebrochen und bei Bedarf bis zur festgelegten Satzlänge mit dem Füllzeichen aufgefüllt.</p>	
<i>WTO=</i>	'Y' 'N'	
	<p>Mit dieser Option ist festzulegen, ob zZip beim Jobstart eine Nachricht über die verwendete Version und beim Jobende eine Nachricht über die Anzahl der verarbeiteten Sätze auf die Konsole ausgeben soll oder nicht. Mit einer Angabe kann die globale Voreinstellung – siehe Seite 34 - überschrieben werden.</p>	

ESDS - Schreiben der Daten in eine VSAM ESDS Datei

Mit diesem Steuerkartentyp wird zZip darüber informiert, dass Daten in einer VSAM ESDS Datei gespeichert werden sollen.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JXHF2ESD JOB , 'XPS HFS TO ESDS', CLASS=A, MSGCLASS=M
//MAKEESDS EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE (XPSHDRV.ESDS) CL PURGE
DEFINE CL(NAME(XPSHDRV.ESDS)           -
          NONINDEXED                   -
          REUSE                         -
          SHR(2)                        -
          VOL(XPS006)                   -
          RECSZ(62 62)                  -
          CISZ(6200)                    -
          REC(1000))
//*
//XHF2ESD EXEC PGM=XPSZZIP, REGION=48M
//STEPLIB DD DSN=XPSHDRV.V300.LOADLIB, DISP=SHR
//XPSDATA DD DISP=SHR, DSN=XPSHDRV.V300.MACLIB
//ESDSOUT DD DISP=SHR, DSN=XPSHDRV.ESDS
//SYSIN DD *
I XPSHFS='TEST/KOMPS.TXT'
O ESDS=ESDSOUT
//*
//PRINT EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//ESDSOUT DD DISP=SHR, DSN=XPSHDRV.ESDS
//SYSIN DD *
PRINT -
  INFILE(ESDSOUT) -
  DUMP
/*
```

Abb. 69: Beispieljob für ESDS Ausgabe z/OS

Im Beispiel wird zunächst eine ESDS Datei erstellt, in die dann der Inhalt der Datei 'KOMPS.TXT' aus dem Unterverzeichnis 'TEST' aus dem XPS HFS-Dateisystem kopiert wird. Schließlich wird der Inhalt der ESDS Datei unter Verwendung des Systemprogrammes 'IDCAMS' ausgedruckt.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE mit vergleichbarem Ablauf.

```
* $$ JOB JNM=JXHF2ESD, DISP=D, PRI=3, CLASS=0
* $$ LST CLASS=T
// JOB JXHF2ESD
// EXEC IDCAMS, SIZE=AUTO
DELETE (XPS.HDRV.ESDS) CL PURGE -
      CATALOG(XPS.USER.CATALOG)
DEFINE CL(NAME(XPS.HDRV.ESDS)           -
          VOL(SYSWK2)                   -
          RECSZ(80 80)                  -
          CISZ(6200)                    -
          NONINDEXED                    -
          REUSE                         -
          SHR(2)                        -
          REC(1000))
```

```

DATA(NAME(XPS.HDRV.ESDS.DATA)) -
CATALOG(XPS.USER.CATALOG)

/*
// LIBDEF PHASE,SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005,FEE
// DLBL XPSESDS,'XPS.HDRV.ESDS',,VSAM,CAT=XPSUCAT
// DLBL XPSHFS,'XPSHDRV.SPOOL.SAVE',,VSAM,CAT=XPSUCAT
// EXEC XPSZZIP,SIZE=AUTO,PARM='LIB=XPS.DAEMH600'
I XPSHFS='TEST/KOMPS.TXT'
O ESDS=XPSESDS
/*
// DLBL XPSESDS,'XPS.HDRV.ESDS',,VSAM,CAT=XPSUCAT
// EXEC IDCAMS,SIZE=AUTO
PRINT INFILE (XPSESDS) -
DUMP
/*
/&
* $$$EOJ
    
```

Abb. 70: Beispieljob für ESDS Ausgabe zVSE

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Karten des Typs 'O ESDS='.

O ESDS=			
Schreiben von Daten in eine VSAM ESDS Datei.			
Optionen	Beschreibung		
CSV=	beliebiges Zeichen, Standard: <u>Leerzeichen</u>		
	<p>Mit dieser Option ist das Zeichen festzulegen, das zZip zur Trennung der einzelnen Felder bei der Aufbereitung der Ausgabedaten verwenden soll. Die daraus resultierende csv-Datei (character separated values) kann, falls sie ins Netzwerk versendet wird, mit Standardprogrammen wie z. B. Microsoft Excel angezeigt werden.</p> <p>Damit diese Option verwendet werden kann, ist bei der Definition der Eingabedaten ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich, damit zZip den Satzaufbau kennt.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>CSV=;</td> </tr> </table>	Beispiel	CSV=;
Beispiel	CSV=;		
CSVFMT=	'C' 'T'		
	<p>Mit dieser Option ist das Format der erstellten csv-Daten zu bestimmen.</p> <p>Damit zZip den Satzaufbau kennt, ist bei der Definition der Eingabedatei ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich.</p> <p>zZip unterstützt die Verarbeitung gepackter und hexadezimaler Daten.</p> <p>Folgende Optionen sind verfügbar:</p> <p>'C' Die Inhalte der einzelnen Satzfelder werden in der Weise komprimiert, dass abschließende Leerzeichen entfernt und durch das angegebene csv-Trennzeichen ersetzt werden.</p> <p>'T' Die Inhalte der einzelnen Satzfelder werden tabellarisch aufbereitet, so dass jedes Feld genau so lang dargestellt wird, wie der Definition der Satzstruktur festgelegt. Bei Bedarf werden führende oder abschließende Leerzeichen eingefügt. Schließlich wird jedes Feld mit dem definierten csv-Trennzeichen abgeschlossen.</p>		

CSVHDR=	'Y' ' <u>N</u> '												
	<p>Mit dieser Option ist festzulegen, ob zZip bei der Erstellung einer csv-Datei basierend auf einer Satzstruktur die Namen der Felder als erste Ausgabezeile einfügen soll.</p> <p>Damit zZip den Satzaufbau kennt, ist bei der Definition der Eingabedatei ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich.</p> <p>Folgende Optionen sind verfügbar:</p> <p>'Y' zZip fügt eine Überschriftzeile ein, die die Namen der einzelnen Felder der Satzstruktur, getrennt durch das mit 'CSV=' angegebene Zeichen, enthält.</p> <p>'<u>N</u>' zZip fügt keine Überschriftzeile ein.</p>												
ESDS=	<p>z/OS: Verweis auf das JCL DD-Statement der VSAM ESDS Datei oder voll qualifizierter Name der VSAM ESDS Datei</p> <p>z/VSE: Verweis auf das JCL DD-Statement der VSAM ESDS Datei</p>												
	<p>Die Angabe muss unter z/OS auf ein DD-Statement und unter z/VSE auf ein DLBL-Statement verweisen, mit dem die VSAM ESDS Datei definiert wird.</p> <p>Unter z/OS kann alternativ der voll qualifizierte Name der VSAM ESDS Datei angegeben werden.</p>												
	<table border="1"> <tr> <td>Beispiel</td> <td>ESDS=XPSESDS</td> </tr> </table>	Beispiel	ESDS=XPSESDS										
Beispiel	ESDS=XPSESDS												
OPT=	'APPEND'												
	<p>Mit dieser Angabe können weitere Optionen für die Dateiverarbeitung angegeben werden. Folgende Angaben sind möglich:</p> <p>'APPEND' Wenn diese Option gesetzt ist, werden neue Daten am Ende der Datei angefügt. Ein bereits bestehender Inhalt der Datei bleibt erhalten.</p>												
	<table border="1"> <tr> <td>Beispiel</td> <td>OPT=APPEND</td> </tr> </table>	Beispiel	OPT=APPEND										
Beispiel	OPT=APPEND												
TRACE=	'ALL' 'D' 'T' 'S' 'Z' oder eine, jeweils durch Komma getrennte, Kombination												
	<p>Mit dieser Option können Traceoptionen für den Job festgelegt werden. Der Trace dient zur Ausgabe von Informationen, die bei Bedarf die Analyse des internen Ablaufs ermöglichen. Traceausgaben erfolgen direkt in das Job-Listing. Die folgende Tabelle enthält die verfügbaren Tracetypen:</p> <table border="1"> <thead> <tr> <th>Typ</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>ALL</td> <td>Alle Tracetypen sind aktiviert.</td> </tr> <tr> <td>D</td> <td>Der Datentrace wird aktiviert.</td> </tr> <tr> <td>T</td> <td>Der TCP/IP Trace wird aktiviert.</td> </tr> <tr> <td>S</td> <td>Der Speichertrace wird aktiviert.</td> </tr> <tr> <td>Z</td> <td>Der ZIP/UNZIP Trace wird aktiviert.</td> </tr> </tbody> </table> <p>Mehrere Tracetypen können aktiviert werden, in dem diese, durch Komma getrennt, aufgelistet werden.</p>	Typ	Beschreibung	ALL	Alle Tracetypen sind aktiviert.	D	Der Datentrace wird aktiviert.	T	Der TCP/IP Trace wird aktiviert.	S	Der Speichertrace wird aktiviert.	Z	Der ZIP/UNZIP Trace wird aktiviert.
Typ	Beschreibung												
ALL	Alle Tracetypen sind aktiviert.												
D	Der Datentrace wird aktiviert.												
T	Der TCP/IP Trace wird aktiviert.												
S	Der Speichertrace wird aktiviert.												
Z	Der ZIP/UNZIP Trace wird aktiviert.												
	<table border="1"> <tr> <td>Beispiel</td> <td>TRACE=S</td> </tr> </table>	Beispiel	TRACE=S										
Beispiel	TRACE=S												
TRUNC=	'Y' ' <u>N</u> '												
	<p>Mit dieser Angabe kann bestimmt werden, wie mit Zeilen verfahren werden soll, die länger als die Satzlänge der ESDS Datei sind.</p> <p>Bei Auswahl von 'TRUNC=Y' werden überschüssige Zeichen abgeschnitten und gehen</p>												

	<p>verloren.</p> <p>Bei der Standardeinstellung 'TRUNC=N' werden überschüssige Zeichen in die nächste Zeile umgebrochen und bei Bedarf bis zur festgelegten Satzlänge mit dem Füllzeichen aufgefüllt.</p>
WTO=	'Y' ' <u>N</u> '
	<p>Mit dieser Option ist festzulegen, ob zZip beim Jobstart eine Nachricht über die verwendete Version und beim Jobende eine Nachricht über die Anzahl der verarbeiteten Sätze auf die Konsole ausgeben soll oder nicht. Mit einer Angabe kann die globale Voreinstellung – siehe Seite 34 - überschrieben werden.</p>

KSDS – Schreiben der Daten in eine VSAM KSDS Datei

Mit diesem Steuerkartentyp wird zZip darüber informiert, dass Daten in einer VSAM KSDS Datei gespeichert werden sollen.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JIPT2KSD JOB , 'IPT TO KSDS' ,CLASS=A,MSGCLASS=M
//DELDEF EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE (VSAM.HDRV.TEST.FILE) CL PURGE
DEFINE CL(NAME(VSAM.HDRV.TEST.FILE) -
INDEXED RECOVERY -
VOLUMES(XPS009) -
KEYS(36,0) -
RECORDS(2000 200) -
RECSZ(60 60) -
FREESPACE(10 10) -
CISZ(2048) -
SHR(2) -
DATA(NAME(VSAM.HDRV.TEST.FILE.DATA))
//*
//IPT2KSD EXEC PGM=XPSZ ZIP,REGION=48M
//STEPLIB DD DISP=SHR,DSN=XPSHDRV.V300.LOADLIB
//XPSDATA DD DISP=SHR,DSN=XPSHDRV.V300.MACLIB
//KSDSOUT DD DISP=SHR,DSN=VSAM.HDRV.TEST.FILE
//SYSIN DD *
I IPT
O KSDS=KSDSOUT OPT=ALL TRUNC=YES
//SYSIN2 DD *
JOHANN SEBASTIAN BACH EISENACH 1685-1750
WOLFGANG AMADEUS MOZART SALZBURG 1756-1791
LUDWIG VAN BEETHOVEN BONN 1770-1827
FRANZ SCHUBERT WIEN 1797-1828
//*
//PRINT EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//KSDSOUT DD DISP=SHR,DSN=VSAM.HDRV.TEST.FILE
//SYSIN DD *
PRINT -
INFILE(KSDSOUT) -
DUMP
//*
```

Abb. 71: Beispieljob für KSDS Ausgabe z/OS

Im Beispiel wird zunächst eine KSDS Datei erstellt, in die dann Datensätze aus dem Bereich 'SYSIN2' aus der JCL eingefügt werden.

Da die Satzlänge der SYSIN Daten immer 80 beträgt, ist die Option 'TRUNC=YES' zu setzen, damit die Daten nur in der Satzlänge der KSDS Datei – 60 stellig – verarbeitet werden.

zZip berücksichtigt den Umstand, dass der erste Satz, der in eine VSAM KSDS Datei eingefügt wird, besonders zu behandeln ist. Es besteht somit die Möglichkeit, zZip eine neu initialisierte Datei bereitzustellen, wie im Beispiel gezeigt.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE mit vergleichbarem Ablauf.

```
* $$ JOB JNM=JIPT2KSD,DISP=D,PRI=3,CLASS=0
```



```

* $$ LST CLASS=T,JNM=JIPT2KSD
// JOB JIPT2KSD
// EXEC IDCAMS,SIZE=AUTO
DELETE (XPS.HDRV.TEST) CL PURGE -
      CATALOG(XPS.USER.CATALOG)
DEFINE CL(NAME(XPS.HDRV.TEST)           -
          INDEXED RECOVERY              -
          VOL(SYSWK2)                   -
          RECSZ(60 60)                   -
          KEYS(36,0)                     -
          CISZ(4096)                     -
          REUSE                           -
          SHR(2)                           -
          REC(5000))                       -
          DATA(NAME(XPS.HDRV.TEST.DATA) -
          CATALOG(XPS.USER.CATALOG)
/*
// LIBDEF PHASE,SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005,SYSLST
// DLBL XPSHFS,'XPSHDRV.SPOOL.SAVE',,VSAM,CAT=XPSUCAT
// DLBL KSDSOUT,'XPS.HDRV.TEST',,VSAM,CAT=XPSUCAT
// EXEC XPSZZIP,SIZE=AUTO,PARM='LIB=XPS.DAEMH600'
I IPT
O KSDS=KSDSOUT OPT=ALL TRUNC=Y
/*
JOHANN SEBASTIAN   BACH                EISENACH        1685-1750
WOLFGANG AMADEUS MOZART                SALZBURG        1756-1791
LUDWIG            VAN BEETHOVEN         BONN             1770-1827
FRANZ            SCHUBERT                WIEN            1797-1828
/*
// DLBL XPSKSDS,'XPS.HDRV.TEST',,VSAM,CAT=XPSUCAT
// EXEC IDCAMS,SIZE=AUTO
PRINT INFILE (XPSKSDS) -
      DUMP
/*
/&
* $$EQJ

```

Abb. 72: Beispieljob für KSDS Ausgabe z/VSE

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Karten des Typs 'O KSDS='.

O KSDS=	
Schreiben von Daten in eine VSAM KSDS Datei.	
Optionen	Beschreibung
KSDS=	z/OS: Verweis auf das JCL DD-Statement der VSAM KSDS Datei oder voll qualifizierter Name der VSAM KSDS Datei z/VSE: Verweis auf das JCL DD-Statement der VSAM KSDS Datei
	Die Angabe muss unter z/OS auf ein DD-Statement und unter z/VSE auf ein DLBL-Statement verweisen, mit dem die VSAM KSDS Datei definiert wird. Unter z/OS kann alternativ der voll qualifizierte Name der VSAM KSDS Datei angegeben werden.
	Beispiel KSDS=XPSKSDS

OPT=	'ALL' 'ADD' 'UPD'												
	<p>Mit dieser Angabe können weitere Optionen für die Dateiverarbeitung angegeben werden. Folgende Angaben sind möglich:</p> <p>'ALL' Alle bereitgestellten Sätze werden verarbeitet. Sätze, die sich noch nicht auf der KSDS Datei befinden, werden hinzugefügt, für bereits existierende Sätze wird ein Update durchgeführt.</p> <p>'ADD' Bei dieser Auswahl werden nur neue Sätze hinzugefügt. Sätze, die sich bereits auf der Datei befinden, bleiben unverändert. Das Verhalten von zZip beim Auftreten bereits vorhandener Sätze kann mit der Option 'WARN=' gesteuert werden.</p> <p>'UPD' Bei dieser Auswahl wird nur für Sätze, die sich bereits auf der Datei befinden, ein Update durchgeführt. Neue Sätze werden nicht hinzugefügt. Das Verhalten von zZip beim Auftreten noch nicht vorhandener Sätze kann mit der Option 'WARN=' gesteuert werden.</p>												
	<table border="1"> <tr> <td>Beispiel</td> <td>OPT=ALL</td> </tr> </table>	Beispiel	OPT=ALL										
Beispiel	OPT=ALL												
TRACE=	'ALL' 'D' 'T' 'S' 'Z' oder eine, jeweils durch Komma getrennte, Kombination												
	<p>Mit dieser Option können Traceoptionen für den Job festgelegt werden. Der Trace dient zur Ausgabe von Informationen, die bei Bedarf die Analyse des internen Ablaufs ermöglichen. Traceausgaben erfolgen direkt in das Job-Listing. Die folgende Tabelle enthält die verfügbaren Tracetypen:</p> <table border="1"> <thead> <tr> <th>Typ</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>ALL</td> <td>Alle Tracetypen sind aktiviert.</td> </tr> <tr> <td>D</td> <td>Der Datentrace wird aktiviert.</td> </tr> <tr> <td>T</td> <td>Der TCP/IP Trace wird aktiviert.</td> </tr> <tr> <td>S</td> <td>Der Speichertrace wird aktiviert.</td> </tr> <tr> <td>Z</td> <td>Der ZIP/UNZIP Trace wird aktiviert.</td> </tr> </tbody> </table> <p>Mehrere Tracetypen können aktiviert werden, in dem diese, durch Komma getrennt, aufgelistet werden.</p>	Typ	Beschreibung	ALL	Alle Tracetypen sind aktiviert.	D	Der Datentrace wird aktiviert.	T	Der TCP/IP Trace wird aktiviert.	S	Der Speichertrace wird aktiviert.	Z	Der ZIP/UNZIP Trace wird aktiviert.
Typ	Beschreibung												
ALL	Alle Tracetypen sind aktiviert.												
D	Der Datentrace wird aktiviert.												
T	Der TCP/IP Trace wird aktiviert.												
S	Der Speichertrace wird aktiviert.												
Z	Der ZIP/UNZIP Trace wird aktiviert.												
	<table border="1"> <tr> <td>Beispiel</td> <td>TRACE=S</td> </tr> </table>	Beispiel	TRACE=S										
Beispiel	TRACE=S												
TRUNC=	'Y' 'N'												
	<p>Mit dieser Angabe kann bestimmt werden, wie mit Zeilen verfahren werden soll, die länger als die Satzlänge der KSDS Datei sind.</p> <p>Bei Auswahl von 'TRUNC=Y' werden überschüssige Zeichen abgeschnitten und gehen verloren.</p> <p>Bei der Standardeinstellung 'TRUNC=N' werden überschüssige Zeichen als eigener Record geschrieben und bei Bedarf bis zur festgelegten Satzlänge mit dem Füllzeichen aufgefüllt.</p>												
WARN=	'Y' 'N'												
	<p>Mit dieser Angabe kann bestimmt werden, wie sich zZip verhalten soll, wenn entweder bereits bestehende Sätze für 'OPT=ADD' oder neue Sätze für 'OPT=UPD' verarbeitet werden.</p> <p>Bei Auswahl der Standardoption 'WARN=Y' führt dies zur Ausgabe einer Nachricht für jeden nicht verarbeiteten Satz und zum Setzen des Returncodes, der mit der Standardeinstellung 'RCNOTFND=' (siehe Seite 32) festgelegt wurde.</p> <p>Bei Auswahl von 'WARN=N' werden die entsprechenden Sätze nicht protokolliert und es</p>												

	wird kein Returncode gesetzt.
<i>WTO=</i>	'Y' ' <u>N</u> '
	Mit dieser Option ist festzulegen, ob zZip beim Jobstart eine Nachricht über die verwendete Version und beim Jobende eine Nachricht über die Anzahl der verarbeiteten Sätze auf die Konsole ausgeben soll oder nicht. Mit einer Angabe kann die globale Voreinstellung – siehe Seite 34 - überschrieben werden.

XPSHFS - Schreiben der Daten in das XPS HFS Dateisystem

Mit diesem Steuerkartentyp wird zZip darüber informiert, dass Daten im XPS HFS Dateisystem gespeichert werden sollen.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JSEQ2XHF JOB , 'SEQ TO XPS HFS', CLASS=A, MSGCLASS=M
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE XPSHDRV.SEQ
//LIB2SEQ EXEC PGM=XPSZZIP, REGION=48M
//STEPLIB DD DSN=XPSHDRV.V300.LOADLIB, DISP=SHR
//XPSIN DD DISP=SHR, DSN=XPSHDRV.V300.MACLIB
//XPSSEQ DD DSN=XPSHDRV.SEQ, DISP=(NEW,CATLG),
// SPACE=(TRK,(5,3),RLSE), UNIT=3390, VOL=SER=XPS005,
// DCB=(LRECL=100, BLKSIZE=1000, RECFM=FB, DSORG=PS)
//SYSIN DD *
I LIB=XPSIN MBR=XPSZIIP
O SEQ=XPSSEQ
//*
//SEQ2XHF EXEC PGM=XPSZZIP, REGION=48M
//STEPLIB DD DSN=XPSHDRV.V300.LOADLIB, DISP=SHR
//XPSSEQ DD DSN=XPSHDRV.SEQ, DISP=(OLD,KEEP),
// SPACE=(TRK,(5,3),RLSE), UNIT=3390, VOL=SER=XPS005,
// DCB=(LRECL=100, BLKSIZE=1000, RECFM=FB, DSORG=PS)
//SYSIN DD *
I SEQ=XPSSEQ
O XPSHFS= 'XPS/HDRV/TEST/XPSZIIP.SRC'
//*
//SPMAINT EXEC PGM=XPSSPMNT, REGION=32M
//STEPLIB DD DSN=XPSHDRV.V300.LOADLIB, DISP=SHR
//XPSHFS DD DISP=SHR, DSN=XPSHDRV.SPOOL.SAVE
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SPOOLLOG DD SYSOUT=*
//SYSIN DD *
LANG D
LISTDIR
DUMPFIL 'XPS/HDRV/TEST/XPSZIIP.SRC'
/*
```

Abb. 73: Beispieljob für XPSHFS Ausgabe z/OS

Im Beispiel wird zunächst ein Bibliotheksmember in eine sequentielle Datei kopiert, deren Inhalt anschließend in das XPS HFS Dateisystem kopiert wird. Im letzten Step wird der Inhalt der HFS Datei unter Verwendung des XPS HFS Batchprozessors 'XPSSPMNT' ausgedruckt.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE mit vergleichbarem Ablauf.

```
* $$ JOB JNM=JSEQ2XHF, DISP=D, PRI=3, CLASS=0
* $$ LST CLASS=T
// JOB JSEQ2XHF
// LIBDEF PHASE, SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005, FEE
// DLBL XPSSEQ, 'HDRV.SEQ.FILE', , VSAM, DISP=(NEW,KEEP), X
RECORDS=(1000,200), RECSIZE=80, CAT=XPSUCAT
// EXTENT , SYSWK5, 1
```

```

// DLBL XPSHFS, 'XPSHDRV.SPOOL.SAVE', ,VSAM,CAT=XPSUCAT
// EXEC XPSZZIP,SIZE=AUTO,PARM='LIB=XPS.DAEMH600'
I LIB=XPS.BACKUP,MBR=RINDE.D
O SEQ=XPSSEQ RECL=80
/*
// LIBDEF PHASE,SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005,FEE
// DLBL XPSSEQ, 'HDRV.SEQ.FILE', ,VSAM,DISP=(OLD,DELETE),CAT=XPSUCAT
// DLBL XPSHFS, 'XPSHDRV.SPOOL.SAVE', ,VSAM,CAT=XPSUCAT
// EXEC XPSZZIP,SIZE=AUTO,PARM='LIB=XPS.DAEMH600'
I SEQ=XPSSEQ RECL=80
O XPSHFS='XPS/HDRV/TEST/XPSCRYPT.D'
/*
// JOB PRINTHFS
// LIBDEF PHASE,SEARCH=(XPS.DAEMH600)
// DLBL XPSHFS, 'XPSHDRV.SPOOL.SAVE', ,VSAM,CAT=XPSUCAT
// ASSGN SYS005,X'FEE'
// EXEC XPSSPMNT,SIZE=AUTO
LANG D
DUMPFIL 'XPS/HDRV/TEST/XPSCRYPT.D'
/*
/&
* $$SEQ

```

Abb. 74: Beispieljob für XPSHFS Ausgabe z/VSE

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Karten des Typs 'O XPSHFS='.

O XPSHFS=			
Schreiben von Daten in das XPS HFS Dateisystem.			
Optionen	Beschreibung		
CSV=	beliebiges Zeichen, Standard: <u>Leerzeichen</u>		
	<p>Mit dieser Option ist das Zeichen festzulegen, das zZip zur Trennung der einzelnen Felder bei der Aufbereitung der Ausgabedaten verwenden soll. Die daraus resultierende csv-Datei (character separated values) kann, falls sie ins Netzwerk versendet wird, mit Standardprogrammen wie z. B. Microsoft Excel angezeigt werden.</p> <p>Damit diese Option verwendet werden kann, ist bei der Definition der Eingabedaten ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich, damit zZip den Satzaufbau kennt.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>CSV=;</td> </tr> </table>	Beispiel	CSV=;
Beispiel	CSV=;		
CSVFMT=	'C' 'T'		
	<p>Mit dieser Option ist das Format der erstellten csv-Daten zu bestimmen.</p> <p>Damit zZip den Satzaufbau kennt, ist bei der Definition der Eingabedatei ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich.</p> <p>zZip unterstützt die Verarbeitung gepackter und hexadezimaler Daten.</p> <p>Folgende Optionen sind verfügbar:</p> <p>'C' Die Inhalte der einzelnen Satzfelder werden in der Weise komprimiert, dass abschließende Leerzeichen entfernt und durch das angegebene csv-Trennzeichen ersetzt werden.</p> <p>'T' Die Inhalte der einzelnen Satzfelder werden tabellarisch aufbereitet, so dass jedes Feld genau so lang dargestellt wird, wie der Definition der</p>		

	Satzstruktur festgelegt. Bei Bedarf werden führende oder abschließende Leerzeichen eingefügt. Schließlich wird jedes Feld mit dem definierten csv-Trennzeichen abgeschlossen.													
CSVHDR=	'Y' ' <u>N</u> '													
	<p>Mit dieser Option ist festzulegen, ob zZip bei der Erstellung einer csv-Datei basierend auf einer Satzstruktur die Namen der Felder als erste Ausgabezeile einfügen soll.</p> <p>Damit zZip den Satzaufbau kennt, ist bei der Definition der Eingabedatei ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich.</p> <p>Folgende Optionen sind verfügbar:</p> <p>'Y' zZip fügt eine Überschriftzeile ein, die die Namen der einzelnen Felder der Satzstruktur, getrennt durch das mit 'CSV=' angegebene Zeichen, enthält.</p> <p>'<u>N</u>' zZip fügt keine Überschriftzeile ein.</p>													
OPT=	'APPEND'													
	<p>Mit dieser Angabe können weitere Optionen für die Dateiverarbeitung angegeben werden. Folgende Angaben sind möglich:</p> <p>'APPEND' Wenn diese Option gesetzt ist, werden neue Daten am Ende der Datei angefügt. Ein bereits bestehender Inhalt der Datei bleibt erhalten.</p>													
	Beispiel	OPT=APPEND												
TRACE=	'ALL' 'D' 'T' 'S' 'Z' oder eine, jeweils durch Komma getrennte, Kombination													
	<p>Mit dieser Option können Traceoptionen für den Job festgelegt werden. Der Trace dient zur Ausgabe von Informationen, die bei Bedarf die Analyse des internen Ablaufs ermöglichen. Traceausgaben erfolgen direkt in das Job-Listing. Die folgende Tabelle enthält die verfügbaren Tracetypen:</p> <table border="1"> <thead> <tr> <th>Typ</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>ALL</td> <td>Alle Tracetypen sind aktiviert.</td> </tr> <tr> <td>D</td> <td>Der Datentrace wird aktiviert.</td> </tr> <tr> <td>T</td> <td>Der TCP/IP Trace wird aktiviert.</td> </tr> <tr> <td>S</td> <td>Der Speichertrace wird aktiviert.</td> </tr> <tr> <td>Z</td> <td>Der ZIP/UNZIP Trace wird aktiviert.</td> </tr> </tbody> </table> <p>Mehrere Tracetypen können aktiviert werden, in dem diese, durch Komma getrennt, aufgelistet werden.</p>		Typ	Beschreibung	ALL	Alle Tracetypen sind aktiviert.	D	Der Datentrace wird aktiviert.	T	Der TCP/IP Trace wird aktiviert.	S	Der Speichertrace wird aktiviert.	Z	Der ZIP/UNZIP Trace wird aktiviert.
Typ	Beschreibung													
ALL	Alle Tracetypen sind aktiviert.													
D	Der Datentrace wird aktiviert.													
T	Der TCP/IP Trace wird aktiviert.													
S	Der Speichertrace wird aktiviert.													
Z	Der ZIP/UNZIP Trace wird aktiviert.													
	Beispiel	TRACE=S												
WTO=	'Y' ' <u>N</u> '													
	<p>Mit dieser Option ist festzulegen, ob zZip beim Jobstart eine Nachricht über die verwendete Version und beim Jobende eine Nachricht über die Anzahl der verarbeiteten Sätze auf die Konsole ausgeben soll oder nicht. Mit einer Angabe kann die globale Voreinstellung – siehe Seite 34 - überschrieben werden.</p>													
XPSHFS=	voll qualifizierte Pfadangabe													
	<p>Mit dieser Option wird die zu speichernde Datei definiert.</p>													
	Beispiel	XPSHFS=' XPS/HDRV/TEST/XPSCRIPT.D '												

FTP - Versenden von Daten an einen FTP-Server

Mit diesem Steuerkartentyp wird zZip darüber informiert, dass Daten an einen FTP-Server geliefert werden sollen.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JJES2FTP JOB , 'JES TO NET', CLASS=A, MSGCLASS=M
//JJES2FTP EXEC PGM=XPSZZIP, REGION=48M
//STEPLIB DD DISP=SHR, DSN=XPSHDRV.V300.LOADLIB
//XPSDATA DD DSN=XPSHDRV.V300.MACLIB, DISP=SHR
//SYSIN DD *
I JES=J001 IPADDR=192.168.0.212 IPPORT=8840
O FTP= '/TEST/&&DATE_&&TIME_J001.TXT' CP=1141-E2N
O IPADDR=192.168.0.131 IPPORT=21
O USER='name' PWD='password'
//*
```

Abb. 75: Beispieljob für FTP Ausgabe z/OS

Im Beispiel wird über den XPS Mainframe Print Services Extender 'PrintEx' eine Datei aus dem JES eingelesen und an einen FTP-Server versendet.

Bei der Bildung des Dateinamens, der an den FTP-Server übermittelt wird, werden die Systemvariablen '&DATE' und '&TIME' – siehe Seite 36 - verwendet, um einen eindeutigen Namen zu generieren.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE mit vergleichbarem Ablauf.

```
* $$ JOB JNM=JPOW2FTP, DISP=D, PRI=3, CLASS=0
* $$ LST CLASS=T, JNM=JPOW2FTP
// JOB JPOW2FTP
// LIBDEF PHASE, SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005, SYSLST
// DLBL XPSHFS, 'XPSHDRV.SPOOL.SAVE', , VSAM, CAT=XPSUCAT
// EXEC XPSZZIP, SIZE=AUTO, PARM='LIB=XPS.DAEMH600'
I POWER=J001 IPADDR=127.0.0.1 IPPORT=8840
O FTP= '/TEST/&&FILE'
O IPADDR=192.168.0.110 IPPORT=21
O USER=&USERID PWD=&PASSWORD
/*
/&
* $$EOJ
```

Abb. 76: Beispieljob für FTP Ausgabe z/VSE

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Karten des Typs 'O FTP='.

O FTP=	
Senden von Daten an einen FTP-Server.	
Optionen	Beschreibung
CP=	Codepage
	Mit dieser Option ist zZip darüber zu informieren, dass Textdaten verarbeitet werden sollen. Falls keine Angabe für eine Codepage gemacht wird, werden die Daten binär übertragen.

	<p>Weitere Informationen zu Codepages sind dem Kapitel 'Codepages' auf Seite 13 zu entnehmen. Eine Tabelle der verfügbaren Codepages befindet sich auf Seite 15.</p>	
	Beispiel	CP=1141-E2N
CPA2E=	Schlüssel einer Codepage	
	<p>Textinformationen, die im Rahmen des FTP-Protokolls zwischen beiden Parteien ausgetauscht werden, sind aus Gründen der Standardisierung immer in ASCII/ANSI kodiert.</p> <p>Mit dieser Option kann festgelegt werden, welche Codepage zur Übersetzung dieser Textdaten beim Lesen von Protokollinformationen verwendet werden soll.</p> <p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 23 beschrieben, global vorbelegt werden.</p> <p>Falls unter z/OS auf eine Angabe für diese Option verzichtet wird, werden die textbasierten Protokollinformationen unter Verwendung der im System eingestellten Codepage übersetzt, die bei Ausführung des Systemmakros 'XLATE' für die Übersetzung von ASCII/ANSI nach EBCDIC zum Einsatz kommt.</p>	
	Beispiel	CPA2E=1141-N2E
CPE2A=	Schlüssel einer Codepage	
	<p>Textinformationen, die im Rahmen des FTP-Protokolls zwischen beiden Parteien ausgetauscht werden, sind aus Gründen der Standardisierung immer in ASCII/ANSI kodiert.</p> <p>Mit dieser Option kann festgelegt werden, welche Codepage zur Übersetzung dieser Textdaten beim Schreiben von Protokollinformationen verwendet werden soll.</p> <p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 23 beschrieben, global vorbelegt werden.</p> <p>Falls unter z/OS auf eine Angabe für diese Option verzichtet wird, werden die textbasierten Protokollinformationen unter Verwendung der im System eingestellten Codepage übersetzt, die bei Ausführung des Systemmakros 'XLATE' für die Übersetzung von EBCDIC nach ASCII/ANSI zum Einsatz kommt.</p>	
	Beispiel	CPE2A=1141-E2N
CSV=	beliebiges Zeichen, Standard: <u>Leerzeichen</u>	
	<p>Mit dieser Option ist das Zeichen festzulegen, das zZip zur Trennung der einzelnen Felder bei der Aufbereitung der Ausgabedaten verwenden soll. Die daraus resultierende csv-Datei (character separated values) kann, falls sie ins Netzwerk versendet wird, mit Standardprogrammen wie z. B. Microsoft Excel angezeigt werden.</p> <p>Damit diese Option verwendet werden kann, ist bei der Definition der Eingabedaten ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich, damit zZip den Satzaufbau kennt.</p>	
	Beispiel	CSV=;
CSVFMT=	'C' 'T'	
	<p>Mit dieser Option ist das Format der erstellten csv-Daten zu bestimmen.</p> <p>Damit zZip den Satzaufbau kennt, ist bei der Definition der Eingabedatei ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich.</p>	

	<p>zZip unterstützt die Verarbeitung gepackter und hexadezimaler Daten. Folgende Optionen sind verfügbar:</p> <p>'<u>C</u>' Die Inhalte der einzelnen Satzfelder werden in der Weise komprimiert, dass abschließende Leerzeichen entfernt und durch das angegebene csv-Trennzeichen ersetzt werden.</p> <p>'<u>T</u>' Die Inhalte der einzelnen Satzfelder werden tabellarisch aufbereitet, so dass jedes Feld genau so lang dargestellt wird, wie der Definition der Satzstruktur festgelegt. Bei Bedarf werden führende oder abschließende Leerzeichen eingefügt. Schließlich wird jedes Feld mit dem definierten csv-Trennzeichen abgeschlossen.</p>		
CSVHDR=	' <u>Y</u> ' ' <u>N</u> '		
	<p>Mit dieser Option ist festzulegen, ob zZip bei der Erstellung einer csv-Datei basierend auf einer Satzstruktur die Namen der Felder als erste Ausgabezeile einfügen soll. Damit zZip den Satzaufbau kennt, ist bei der Definition der Eingabedatei ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich. Folgende Optionen sind verfügbar:</p> <p>'<u>Y</u>' zZip fügt eine Überschriftzeile ein, die die Namen der einzelnen Felder der Satzstruktur, getrennt durch das mit 'CSV=' angegebene Zeichen, enthält.</p> <p>'<u>N</u>' zZip fügt keine Überschriftzeile ein.</p>		
FTP=	voll qualifizierte Pfadangabe		
	<p>Mit dieser Option werden Name und Pfad der zu schreibenden Datei auf dem FTP-Server angegeben. Es ist zu beachten, dass die Angabe relativ zu einem eventuell eingerichteten Home Verzeichnis auf dem Server für den ausführenden User interpretiert wird. Unter Verwendung eines oder mehrerer FTP Kommandos – siehe hierzu die nachfolgend beschriebene Option 'FTPCMD=' – kann das Verzeichnis auf dem Server gewechselt werden, falls dies für den angemeldeten User zulässig ist.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>FTP='/TEST/DATE_&&TIME_J001.TXT '</td> </tr> </table>	Beispiel	FTP='/TEST/DATE_&&TIME_J001.TXT '
Beispiel	FTP='/TEST/DATE_&&TIME_J001.TXT '		
FTPCMD=	gültiges FTP Kommando		
	<p>Mit dieser Option können FTP Kommandos angegeben werden, deren Ausführung zZip vor der Übertragung von Daten auf den FTP Server veranlassen soll. zZip erlaubt die Angabe beliebig vieler FTP Kommandos. Es ist jedoch zu beachten, dass jedes FTP Kommando in einer eigene I-Steuerkarte anzugeben ist, und dass es in einfache Anführungszeichen einzuschließen ist, falls das Kommando Leerzeichen enthält.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>I FTPCMD='SITE UNIX ON'</td> </tr> </table>	Beispiel	I FTPCMD='SITE UNIX ON'
Beispiel	I FTPCMD='SITE UNIX ON'		
IPADDR=	TCP/IP Adresse – numerisch oder als DNS-Name		
	<p>Mit dieser Option ist TCP/IP Adresse des FTP-Servers anzugeben.</p>		
	<table border="1"> <tr> <td>Beispiele</td> <td>IPADDR=192.168.0.131 IPADDR=ftp.mozilla.org</td> </tr> </table>	Beispiele	IPADDR=192.168.0.131 IPADDR= ftp.mozilla.org
Beispiele	IPADDR=192.168.0.131 IPADDR= ftp.mozilla.org		
IPPORT=	numerischer TCP/IP Port <= 65535 ' <u>21</u> '		

	<p>Mit dieser Option ist der TCP/IP Port Anzugeben, den der FTP-Server auf eingehende Anfragen hin überwacht.</p> <p>Falls für diese Option keine Angabe erfolgt, wird Port 21 verwendet, was dem gängigen Standard entspricht.</p>												
PWD=	<u>Anonymus</u> Passwort												
	<p>Mit dieser Option ist das Passwort des Benutzers für die Anmeldung am FTP-Server anzugeben.</p>												
	<table border="1"> <tr> <td>Beispiel</td> <td>PWD=mozart</td> </tr> </table>	Beispiel	PWD=mozart										
Beispiel	PWD=mozart												
TRACE=	'ALL' 'D' 'T' 'S' 'Z' oder eine, jeweils durch Komma getrennte, Kombination												
	<p>Mit dieser Option können Traceoptionen für den Job festgelegt werden. Der Trace dient zur Ausgabe von Informationen, die bei Bedarf die Analyse des internen Ablaufs ermöglichen. Traceausgaben erfolgen direkt in das Job-Listing. Die folgende Tabelle enthält die verfügbaren Tracetypen:</p>												
	<table border="1"> <thead> <tr> <th>Typ</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>ALL</td> <td>Alle Tracetypen sind aktiviert.</td> </tr> <tr> <td>D</td> <td>Der Datentrace wird aktiviert.</td> </tr> <tr> <td>T</td> <td>Der TCP/IP Trace wird aktiviert.</td> </tr> <tr> <td>S</td> <td>Der Speichertrace wird aktiviert.</td> </tr> <tr> <td>Z</td> <td>Der ZIP/UNZIP Trace wird aktiviert.</td> </tr> </tbody> </table>	Typ	Beschreibung	ALL	Alle Tracetypen sind aktiviert.	D	Der Datentrace wird aktiviert.	T	Der TCP/IP Trace wird aktiviert.	S	Der Speichertrace wird aktiviert.	Z	Der ZIP/UNZIP Trace wird aktiviert.
Typ	Beschreibung												
ALL	Alle Tracetypen sind aktiviert.												
D	Der Datentrace wird aktiviert.												
T	Der TCP/IP Trace wird aktiviert.												
S	Der Speichertrace wird aktiviert.												
Z	Der ZIP/UNZIP Trace wird aktiviert.												
	<p>Mehrere Tracetypen können aktiviert werden, in dem diese, durch Komma getrennt, aufgelistet werden.</p>												
	<table border="1"> <tr> <td>Beispiel</td> <td>TRACE=S</td> </tr> </table>	Beispiel	TRACE=S										
Beispiel	TRACE=S												
USER=	<u>Anonymus</u> Benutzername												
	<p>Mit dieser Option ist der Name des Benutzers für die Anmeldung am FTP-Server anzugeben.</p>												
	<table border="1"> <tr> <td>Beispiel</td> <td>USER=nannerl</td> </tr> </table>	Beispiel	USER=nannerl										
Beispiel	USER=nannerl												
WTO=	'Y' ' <u>N</u> '												
	<p>Mit dieser Option ist festzulegen, ob zZip beim Jobstart eine Nachricht über die verwendete Version und beim Jobende eine Nachricht über die Anzahl der verarbeiteten Sätze auf die Konsole ausgeben soll oder nicht. Mit einer Angabe kann die globale Voreinstellung – siehe Seite 34 - überschrieben werden.</p>												

HFS - Schreiben der Daten in das OpenMVS HFS-Dateisystem

Mit diesem Steuerkartentyp wird zZip darüber informiert, dass Daten in eine Datei in das OpenMVS HFS-Dateisystem geschrieben werden sollen. Diese Option kann nur unter z/OS verwendet werden.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JESD2HFS JOB , 'ESDS TO HFS' ,CLASS=A,MSGCLASS=M
//DELDEF EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE (XPSHDRV.ESDS) CL PURGE
DEFINE CL(NAME(XPSHDRV.ESDS) -
        NONINDEXED -
        REUSE -
        SHR(2) -
        VOL(XPS006) -
        RECSZ(80 80) -
        CISZ(4096) -
        REC(1000))
//*
//IPT2ESD EXEC PGM=XPSZZIP,REGION=48M
//STEPLIB DD DSN=XPSHDRV.V300.LOADLIB,DISP=SHR
//XPDATA DD DSN=XPSHDRV.V300.MACLIB,DISP=SHR
//ESDSOUT DD DISP=SHR,DSN=XPSHDRV.ESDS
//SYSIN DD *
I IPT RECORD=ARTISTS/KOMPONISTEN
O ESDS=ESDSOUT CSV= CSVHDR=Y CSVFMT=T TRUNC=Y
//SYSIN2 DD *
GEORG FRIEDRICH HAENDEL HALLE 1685-1759
JOHANN SEBASTIAN BACH EISENACH 1685-1750
WOLFGANG AMADEUS MOZART SALZBURG 1756-1791
LUDWIG VAN BEETHOVEN BONN 1770-1827
FRANZ SCHUBERT WIEN 1797-1828
FREDERIC CHOPIN ZELAZOWA WOLA 1810-1849
GEORGE GERSHWIN NEW YORK CITY 1898-1937
//*
//ESD2HFS EXEC PGM=XPSZZIP,REGION=48M
//STEPLIB DD DSN=XPSHDRV.V300.LOADLIB,DISP=SHR
//XPDATA DD DSN=XPSHDRV.V300.MACLIB,DISP=SHR
//ESDSIN DD DISP=SHR,DSN=XPSHDRV.ESDS
//SYSIN DD *
I ESDS=ESDSIN
O HFS='/usr/common/komponisten.txt'
//*
```

Abb. 77: Beispieljob für HFS Ausgabe z/OS

Im Beispiel wird zunächst eine VSAM ESDS Datei erstellt, in die im zweiten Jobstep einige Daten aus SYSIN eingefügt werden.

Der Inhalt der so erstellten ESDS Datei wird schließlich in eine Datei auf dem OpenMVS HFS-Dateisystem kopiert.

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Karten des Typs 'O HFS='.

O HFS=
Schreiben von Daten in das OpenMVS HFS-Dateisystem.

Optionen	Beschreibung		
CSV=	beliebiges Zeichen, Standard: <u>Leerzeichen</u>		
	<p>Mit dieser Option ist das Zeichen festzulegen, das zZip zur Trennung der einzelnen Felder bei der Aufbereitung der Ausgabedaten verwenden soll. Die daraus resultierende csv-Datei (character separated values) kann, falls sie ins Netzwerk versendet wird, mit Standardprogrammen wie z. B. Microsoft Excel angezeigt werden.</p> <p>Damit diese Option verwendet werden kann, ist bei der Definition der Eingabedaten ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich, damit zZip den Satzaufbau kennt.</p>		
	<table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Beispiel</td> <td>CSV=;</td> </tr> </table>	Beispiel	CSV=;
Beispiel	CSV=;		
CSVFMT=	'C' 'T'		
	<p>Mit dieser Option ist das Format der erstellten csv-Daten zu bestimmen.</p> <p>Damit zZip den Satzaufbau kennt, ist bei der Definition der Eingabedatei ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich.</p> <p>zZip unterstützt die Verarbeitung gepackter und hexadezimaler Daten.</p> <p>Folgende Optionen sind verfügbar:</p> <p>'C' Die Inhalte der einzelnen Satzfelder werden in der Weise komprimiert, dass abschließende Leerzeichen entfernt und durch das angegebene csv-Trennzeichen ersetzt werden.</p> <p>'T' Die Inhalte der einzelnen Satzfelder werden tabellarisch aufbereitet, so dass jedes Feld genau so lang dargestellt wird, wie der Definition der Satzstruktur festgelegt. Bei Bedarf werden führende oder abschließende Leerzeichen eingefügt. Schließlich wird jedes Feld mit dem definierten csv-Trennzeichen abgeschlossen.</p>		
CSVHDR=	'Y' 'N'		
	<p>Mit dieser Option ist festzulegen, ob zZip bei der Erstellung einer csv-Datei basierend auf einer Satzstruktur die Namen der Felder als erste Ausgabezeile einfügen soll.</p> <p>Damit zZip den Satzaufbau kennt, ist bei der Definition der Eingabedatei ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich.</p> <p>Folgende Optionen sind verfügbar:</p> <p>'Y' zZip fügt eine Überschriftzeile ein, die die Namen der einzelnen Felder der Satzstruktur, getrennt durch das mit 'CSV=' angegebene Zeichen, enthält.</p> <p>'N' zZip fügt keine Überschriftzeile ein.</p>		
HFS=	Pfadangabe zur Datei im HFS-Dateisystem		
	<p>Es ist zu beachten, dass bei Dateinamen im OpenMVS HFS-Dateisystem zwischen Groß- und Kleinschreibung unterschieden wird.</p>		
	<table border="1" style="width: 100%;"> <tr> <td style="width: 20%;">Beispiel</td> <td>HFS='/usr/common/komponisten.txt'</td> </tr> </table>	Beispiel	HFS='/usr/common/komponisten.txt'
Beispiel	HFS='/usr/common/komponisten.txt'		
OPT=	'APPEND'		
	<p>Mit dieser Angabe können weitere Optionen für die Dateiverarbeitung angegeben werden. Folgende Angaben sind möglich:</p> <p>'APPEND' Wenn diese Option gesetzt ist, werden neue Daten am Ende der Datei angefügt. Ein bereits bestehender Inhalt der Datei bleibt erhalten.</p>		

	Beispiel	OPT=APPEND												
<i>TRACE=</i>	'ALL' 'D' 'T' 'S' 'Z' oder eine, jeweils durch Komma getrennte, Kombination													
	<p>Mit dieser Option können Traceoptionen für den Job festgelegt werden. Der Trace dient zur Ausgabe von Informationen, die bei Bedarf die Analyse des internen Ablaufs ermöglichen. Traceausgaben erfolgen direkt in das Job-Listing. Die folgende Tabelle enthält die verfügbaren Tracetypen:</p> <table border="1"> <thead> <tr> <th>Typ</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td><i>ALL</i></td> <td>Alle Tracetypen sind aktiviert.</td> </tr> <tr> <td><i>D</i></td> <td>Der Datentrace wird aktiviert.</td> </tr> <tr> <td><i>T</i></td> <td>Der TCP/IP Trace wird aktiviert.</td> </tr> <tr> <td><i>S</i></td> <td>Der Speichertrace wird aktiviert.</td> </tr> <tr> <td><i>Z</i></td> <td>Der ZIP/UNZIP Trace wird aktiviert.</td> </tr> </tbody> </table> <p>Mehrere Tracetypen können aktiviert werden, indem diese, durch Komma getrennt, aufgelistet werden.</p>		Typ	Beschreibung	<i>ALL</i>	Alle Tracetypen sind aktiviert.	<i>D</i>	Der Datentrace wird aktiviert.	<i>T</i>	Der TCP/IP Trace wird aktiviert.	<i>S</i>	Der Speichertrace wird aktiviert.	<i>Z</i>	Der ZIP/UNZIP Trace wird aktiviert.
Typ	Beschreibung													
<i>ALL</i>	Alle Tracetypen sind aktiviert.													
<i>D</i>	Der Datentrace wird aktiviert.													
<i>T</i>	Der TCP/IP Trace wird aktiviert.													
<i>S</i>	Der Speichertrace wird aktiviert.													
<i>Z</i>	Der ZIP/UNZIP Trace wird aktiviert.													
	Beispiel	TRACE=S												
<i>WTO=</i>	'Y' 'N'													
	<p>Mit dieser Option ist festzulegen, ob zZip beim Jobstart eine Nachricht über die verwendete Version und beim Jobende eine Nachricht über die Anzahl der verarbeiteten Sätze auf die Konsole ausgeben soll oder nicht. Mit einer Angabe kann die globale Voreinstellung – siehe Seite 34 - überschrieben werden.</p>													

JES - Schreiben der Daten in das z/OS Spoolsystem

Mit diesem Steuerkartentyp wird zZip darüber informiert, dass Daten ins JES eingespielt werden sollen. Diese Option kann nur unter z/OS verwendet werden.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JHFS2JES JOB , 'HFS TO JES' ,CLASS=A,MSGCLASS=M
//HFS2JES EXEC PGM=XPSZZIP,REGION=96M
//STEPLIB DD DSN=XPSHDRV.V300.LOADLIB,DISP=SHR
//SYSIN DD *
I HFS=' /usr/common/komponisten.txt' RECL=66
O JES=OUT CLA=M DEST=LOCAL FORM=HFS PDISP=W JNAME=HFSSI
/*
```

Abb. 78: Beispieljob für JES Ausgabe z/OS

Im Beispiel wird der Inhalt einer Datei aus dem OpenMVS HFS-Dateisystem ins JES ausgegeben.

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Karten des Typs 'O JES='.

O JES=	
Schreiben von Daten ins JES – nur z/OS.	
Optionen	Beschreibung
CLA=	'0' ... '9' 'A' ... 'Z'
	Mit dieser Option ist eine gültige JES Output Klasse anzugeben, in die die Datei eingestellt werden soll.
	Beispiel CLA=M
CSV=	beliebiges Zeichen, Standard: <u>Leerzeichen</u>
	Mit dieser Option ist das Zeichen festzulegen, das zZip zur Trennung der einzelnen Felder bei der Aufbereitung der Ausgabedaten verwenden soll. Die daraus resultierende csv-Datei (character separated values) kann, falls sie ins Netzwerk versendet wird, mit Standardprogrammen wie z. B. Microsoft Excel angezeigt werden. Damit diese Option verwendet werden kann, ist bei der Definition der Eingabedaten ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich, damit zZip den Satzaufbau kennt.
	Beispiel CSV=;
CSVFMT=	'C' 'T'
	Mit dieser Option ist das Format der erstellten csv-Daten zu bestimmen. Damit zZip den Satzaufbau kennt, ist bei der Definition der Eingabedatei ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich. zZip unterstützt die Verarbeitung gepackter und hexadezimaler Daten. Folgende Optionen sind verfügbar: 'C' Die Inhalte der einzelnen Satzfelder werden in der Weise komprimiert, dass abschließende Leerzeichen entfernt und durch das angegebene csv-Trennzeichen ersetzt werden. 'T' Die Inhalte der einzelnen Satzfelder werden tabellarisch aufbereitet, so dass jedes Feld genau so lang dargestellt wird, wie der Definition der

	Satzstruktur festgelegt. Bei Bedarf werden führende oder abschließende Leerzeichen eingefügt. Schließlich wird jedes Feld mit dem definierten csv-Trennzeichen abgeschlossen.
<i>CSVHDR=</i>	'Y' 'N'
	<p>Mit dieser Option ist festzulegen, ob zZip bei der Erstellung einer csv-Datei basierend auf einer Satzstruktur die Namen der Felder als erste Ausgabezeile einfügen soll.</p> <p>Damit zZip den Satzaufbau kennt, ist bei der Definition der Eingabedatei ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich.</p> <p>Folgende Optionen sind verfügbar:</p> <p>'Y' zZip fügt eine Überschriftzeile ein, die die Namen der einzelnen Felder der Satzstruktur, getrennt durch das mit 'CSV=' angegebene Zeichen, enthält.</p> <p>'N' zZip fügt keine Überschriftzeile ein.</p>
<i>CTLCH=</i>	hexadezimaler 2-stelliger EBCDIC Wert (X'...')
	<p>Mit dieser Option kann das Drucksteuerzeichen angegeben werden, das zZip jeder Zeile voranstellen soll. Diese Option ist verpflichtend, falls die Ausgabedaten nicht bereits ein Drucksteuerzeichen enthalten.</p> <p>Die folgenden Werte sind möglich:</p> <p>X'09' Der Zeilenvorschub beträgt eine Zeile.</p> <p>X'11' Der Zeilenvorschub beträgt zwei Zeilen.</p> <p>X'19' Der Zeilenvorschub beträgt drei Zeilen.</p>
	Beispiel CTLCH=X'09'
<i>DEST=</i>	Zeichenkette
	<p>Mit dieser Angabe ist eine beliebige, maximal 8 Zeichen lange Angabe für die JES Destination durchzuführen, die der erstellten Liste zugewiesen werden soll.</p> <p>Falls keine Angabe erfolgt, erhält wird der Liste automatisch die JES Standard Destination 'LOCAL' zugewiesen.</p>
	Beispiel DEST=FTP
<i>FORM=</i>	Zeichenkette
	<p>Mit dieser Angabe ist die JES Formularbezeichnung anzugeben, die der erstellten Liste zugewiesen werden soll.</p> <p>Falls keine Angabe erfolgt erhält wird der Liste automatisch die JES Standardforms 'STD' zugewiesen.</p>
	Beispiel FORM=HDRV
<i>JES=</i>	'OUT' 'IN'
	<p>Mit dieser Option wird zZip angewiesen, Daten ins JES zu schreiben.</p> <p>Folgende Optionen sind verfügbar:</p> <p>'OUT' Die Daten werden in die JES Output Queue eingestellt.</p> <p>'IN' Die Daten werden an den JES Internal Reader übermittelt, was zu einer Interpretation der Daten als Job Control sowie einem Submit führt.</p>
	Beispiel JES=OUT

JNAME=	Bezeichnung	
	Die hier angegebene Bezeichnung wird zum DD-Namen der erstellten JES Datei. Falls diese Angabe fehlt, generiert zZip einen eindeutigen DD-Namen der Form 'HDRVDDnn', wobei die beiden mit 'nn' gekennzeichneten Stellen einen Zähler in aufsteigender Form erhalten.	
	Beispiel	JNAME=HFSI
PDISP=	'W' 'K' 'L'	
	Mit dieser Option ist die Output Disposition der erstellten JES Datei festzulegen. Folgende Optionen sind verfügbar: 'W' Die Liste erhält die Disposition 'WRITE'. 'K' Die Liste erhält die Disposition 'KEEP'. 'L' Die Liste erhält die Disposition 'LEAVE'.	
	Beispiel	PDISP=K
TRACE=	'ALL' 'D' 'T' 'S' 'Z' oder eine, jeweils durch Komma getrennte, Kombination	
	Mit dieser Option können Traceoptionen für den Job festgelegt werden. Der Trace dient zur Ausgabe von Informationen, die bei Bedarf die Analyse des internen Ablaufs ermöglichen. Traceausgaben erfolgen direkt in das Job-Listing. Die folgende Tabelle enthält die verfügbaren Tracetypen:	
	Typ	Beschreibung
	ALL	Alle Tracetypen sind aktiviert.
	D	Der Datentrace wird aktiviert.
	T	Der TCP/IP Trace wird aktiviert.
	S	Der Speichertrace wird aktiviert.
	Z	Der ZIP/UNZIP Trace wird aktiviert.
	Mehrere Tracetypen können aktiviert werden, in dem diese, durch Komma getrennt, aufgelistet werden.	
	Beispiel	TRACE=S
WTO=	'Y' 'N'	
	Mit dieser Option ist festzulegen, ob zZip beim Jobstart eine Nachricht über die verwendete Version und beim Jobende eine Nachricht über die Anzahl der verarbeiteten Sätze auf die Konsole ausgeben soll oder nicht. Mit einer Angabe kann die globale Voreinstellung – siehe Seite 34 - überschrieben werden.	

POWER – Schreiben der Daten in das z/VSE Spoolsystem

Mit diesem Steuerkartentyp wird zZip darüber informiert, dass Daten ins POWER eingespielt werden sollen. Diese Option kann nur unter z/VSE verwendet werden.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE:

```
* $$ JOB JNM=JFTP2POW,DISP=D,PRI=3,CLASS=0
* $$ LST CLASS=T,JNM=JFTP2POW
// JOB JFTP2POW
// LIBDEF PHASE,SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005,SYSLST
// DLBL XPSHFS,'XPSHDRV.SPOOL.SAVE',,VSAM,CAT=XPSUCAT
// EXEC XPSZZIP,SIZE=AUTO,PARM='LIB=XPS.DAEMH600'
I FTP='/ABSCHLUSS/UMSATZ.TXT' CP=1141-N2E TRACE=NO
I IPADDR=192.168.0.110 IPPORT=21
I USER='georg' PWD='lombok'
O POWER=LST JNAME=UMSATZ01 CLA=H PUSER=XPSHDRV CTLCH=09
/*
/&
* $$EOJ
```

Abb. 79: Beispieljob für POWER Ausgabe z/VSE

Im Beispiel wird eine Datei über FTP gelesen und in die POWER List Queue eingestellt.

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Karten des Typs 'O POWER='.

O POWER=	
Schreiben von Daten ins POWER – nur z/VSE.	
Optionen	Beschreibung
CLA=	'0' ... '9' 'A' ... 'Z'
	Mit dieser Option ist eine gültige POWER Output Klasse anzugeben, in die die Datei eingestellt werden soll.
	Beispiel CLA=A
CSV=	beliebiges Zeichen, Standard: <u>Leerzeichen</u>
	Mit dieser Option ist das Zeichen festzulegen, das zZip zur Trennung der einzelnen Felder bei der Aufbereitung der Ausgabedaten verwenden soll. Die daraus resultierende csv-Datei (character separated values) kann, falls sie ins Netzwerk versendet wird, mit Standardprogrammen wie z. B. Microsoft Excel angezeigt werden. Damit diese Option verwendet werden kann, ist bei der Definition der Eingabedaten ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich, damit zZip den Satzaufbau kennt.
	Beispiel CSV=;
CSVFMT=	'C' 'T'
	Mit dieser Option ist das Format der erstellten csv-Daten zu bestimmen. Damit zZip den Satzaufbau kennt, ist bei der Definition der Eingabedatei ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich.

	<p>zZip unterstützt die Verarbeitung gepackter und hexadezimaler Daten. Folgende Optionen sind verfügbar:</p> <p>'<u>C</u>' Die Inhalte der einzelnen Satzfelder werden in der Weise komprimiert, dass abschließende Leerzeichen entfernt und durch das angegebene csv-Trennzeichen ersetzt werden.</p> <p>'<u>T</u>' Die Inhalte der einzelnen Satzfelder werden tabellarisch aufbereitet, so dass jedes Feld genau so lang dargestellt wird, wie der Definition der Satzstruktur festgelegt. Bei Bedarf werden führende oder abschließende Leerzeichen eingefügt. Schließlich wird jedes Feld mit dem definierten csv-Trennzeichen abgeschlossen.</p>		
CSVHDR=	'Y' ' <u>N</u> '		
	<p>Mit dieser Option ist festzulegen, ob zZip bei der Erstellung einer csv-Datei basierend auf einer Satzstruktur die Namen der Felder als erste Ausgabezeile einfügen soll. Damit zZip den Satzaufbau kennt, ist bei der Definition der Eingabedatei ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich. Folgende Optionen sind verfügbar:</p> <p>'Y' zZip fügt eine Überschriftzeile ein, die die Namen der einzelnen Felder der Satzstruktur, getrennt durch das mit 'CSV=' angegebene Zeichen, enthält.</p> <p>'<u>N</u>' zZip fügt keine Überschriftzeile ein.</p>		
CTLCH=	hexadezimaler 2-stelliger EBCDIC Wert (X'....')		
	<p>Mit dieser Option kann das Drucksteuerzeichen angegeben werden, das zZip jeder Zeile voranstellen soll. Diese Option ist verpflichtend, falls die Ausgabedaten nicht bereits ein Drucksteuerzeichen enthalten. Die folgenden Werte sind möglich:</p> <p>X'09' Der Zeilenvorschub beträgt eine Zeile. X'11' Der Zeilenvorschub beträgt zwei Zeilen. X'19' Der Zeilenvorschub beträgt drei Zeilen.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>CTLCH=X'09'</td> </tr> </table>	Beispiel	CTLCH=X'09'
Beispiel	CTLCH=X'09'		
DEST=	Zeichenkette		
	<p>Mit dieser Angabe ist, falls gewünscht, eine beliebige, maximal 8 Zeichen lange Angabe für die POWER Destination durchzuführen, die der erstellten Liste zugewiesen werden soll.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>DEST=EMAIL</td> </tr> </table>	Beispiel	DEST=EMAIL
Beispiel	DEST=EMAIL		
FORM=	Zeichenkette		
	<p>Mit dieser Angabe ist, falls gewünscht, die POWER Formularbezeichnung ('FNO') anzugeben, die der erstellten Liste zugewiesen werden soll.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>FORM=HDRV</td> </tr> </table>	Beispiel	FORM=HDRV
Beispiel	FORM=HDRV		
JNAME=	Bezeichnung		
	<p>Mit dieser Option ist festzulegen, wie der Name der erstellten Datei im POWER lauten soll.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>JNAME=HDRVFTP</td> </tr> </table>	Beispiel	JNAME=HDRVFTP
Beispiel	JNAME=HDRVFTP		
PDISP=	' <u>D</u> ' 'K' 'L'		

	<p>Mit dieser ist die Output Disposition der erstellten POWER Datei festzulegen.</p> <p>Folgende Optionen sind verfügbar:</p> <p>'D' Die Liste erhält die Disposition 'DELETE'.</p> <p>'K' Die Liste erhält die Disposition 'KEEP'.</p> <p>'L' Die Liste erhält die Disposition 'LEAVE'.</p>												
	<table border="1"> <tr> <td>Beispiel</td> <td>PDISP=K</td> </tr> </table>	Beispiel	PDISP=K										
Beispiel	PDISP=K												
POWER=	'LST' 'PUN' 'RDR' 'XMT'												
	<p>Mit dieser Option wird zZip angewiesen, Daten ins POWER zu schreiben.</p> <p>Folgende Optionen sind verfügbar:</p> <p>'LST' Die Daten werden in die List Queue eingestellt.</p> <p>'PUN' Die Daten werden in die Punch Queue eingestellt.</p> <p>'RDR' Die Daten werden in die Reader Queue eingestellt.</p> <p>'XMT' Die Daten werden in die XMIT Queue eingestellt.</p>												
	<table border="1"> <tr> <td>Beispiel</td> <td>POWER=LST</td> </tr> </table>	Beispiel	POWER=LST										
Beispiel	POWER=LST												
PUSER=	Zeichenkette												
	<p>Mit dieser Angabe ist der Name des Benutzers anzugeben, dem die erstellte POWER Liste zugeordnet werden soll.</p>												
	<table border="1"> <tr> <td>Beispiel</td> <td>PUSER=XPS</td> </tr> </table>	Beispiel	PUSER=XPS										
Beispiel	PUSER=XPS												
RECL=	numerischer Wert '133'												
	<p>Mit dieser Option kann bei Bedarf für 'POWER=LST' eine vom Standard '133' abweichende Satzlänge festgelegt werden.</p>												
	<table border="1"> <tr> <td>Beispiel</td> <td>RECL=80</td> </tr> </table>	Beispiel	RECL=80										
Beispiel	RECL=80												
TRACE=	'ALL' 'D' 'T' 'S' 'Z' oder eine, jeweils durch Komma getrennte, Kombination												
	<p>Mit dieser Option können Traceoptionen für den Job festgelegt werden. Der Trace dient zur Ausgabe von Informationen, die bei Bedarf die Analyse des internen Ablaufs ermöglichen. Traceausgaben erfolgen direkt in das Job-Listing. Die folgende Tabelle enthält die verfügbaren Tracetypen:</p> <table border="1"> <thead> <tr> <th>Typ</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>ALL</td> <td>Alle Tracetypen sind aktiviert.</td> </tr> <tr> <td>D</td> <td>Der Datentrace wird aktiviert.</td> </tr> <tr> <td>T</td> <td>Der TCP/IP Trace wird aktiviert.</td> </tr> <tr> <td>S</td> <td>Der Speichertrace wird aktiviert.</td> </tr> <tr> <td>Z</td> <td>Der ZIP/UNZIP Trace wird aktiviert.</td> </tr> </tbody> </table> <p>Mehrere Tracetypen können aktiviert werden, in dem diese, durch Komma getrennt, aufgelistet werden.</p>	Typ	Beschreibung	ALL	Alle Tracetypen sind aktiviert.	D	Der Datentrace wird aktiviert.	T	Der TCP/IP Trace wird aktiviert.	S	Der Speichertrace wird aktiviert.	Z	Der ZIP/UNZIP Trace wird aktiviert.
Typ	Beschreibung												
ALL	Alle Tracetypen sind aktiviert.												
D	Der Datentrace wird aktiviert.												
T	Der TCP/IP Trace wird aktiviert.												
S	Der Speichertrace wird aktiviert.												
Z	Der ZIP/UNZIP Trace wird aktiviert.												
	<table border="1"> <tr> <td>Beispiel</td> <td>TRACE=S</td> </tr> </table>	Beispiel	TRACE=S										
Beispiel	TRACE=S												
WTO=	'Y' 'N'												
	<p>Mit dieser Option ist festzulegen, ob zZip beim Jobstart eine Nachricht über die verwendete Version und beim Jobende eine Nachricht über die Anzahl der verarbeiteten Sätze auf die Konsole ausgeben soll oder nicht. Mit einer Angabe kann die globale</p>												

	Voreinstellung – siehe Seite 34 - überschrieben werden.
--	---

EMAIL - Versenden von Daten als E-Mail Anhang

Mit diesem Steuerkartentyp wird zZip darüber informiert, dass Daten als E-Mail Anhang versendet werden sollen.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/OS:

```
//JKSD2EML JOB , 'KSDS TO EMAIL', CLASS=A, MSGCLASS=M
//KSD2EML EXEC PGM=XPSZZIP, REGION=96M
//STEPLIB DD DSN=XPSHDRV.V300.LOADLIB, DISP=SHR
//XPSKSDS DD DISP=SHR, DSN=VSAM.HDRV.TEST.FILE
//SYSIN DD *
I KSDS=XPSKSDS
O EMAIL=KSDS.TXT CP=1141-E2N
O IPADDR=&SMTP_SERVER IPPORT=&SMTP_SERVER_PORT
O USER=&EMAIL_ACCOUNT PWD=&EMAIL_ACCOUNT_PWD
O FROM=&EMAIL_FROM
O TO=recipient@host.de
O SUBJECT= 'KOMPONISTEN'
O BODY:
O Hello Empfänger!
O Anbei die Daten einiger klassischer Komponisten
O END:
/*
```

Abb. 80: Beispieljob für E-Mail Ausgabe z/OS

Im Beispiel werden alle Sätze einer VSAM KSDS-Datei als E-Mail Dateianhang ('.txt') versendet.

Die nachfolgende Abbildung zeigt einen Beispieljob für z/VSE mit vergleichbarem Ablauf.

```
* $$ JOB JNM=JKSD2EML, DISP=D, PRI=3, CLASS=0
* $$ LST CLASS=T, JNM=JKSD2EML
// JOB JKSD2EML
// LIBDEF PHASE, SEARCH=(XPS.DAEMH600)
// OPTION SYSPARM='00'
// ASSGN SYS005, SYSLST
// DLBL XPSHFS, 'XPSHDRV.SPOOL.SAVE', , VSAM, CAT=XPSUCAT
// DLBL XPSKSDS, 'XPS.HDRV.TEST', , VSAM, CAT=XPSUCAT
// EXEC XPSZZIP, SIZE=AUTO, PARM='LIB=XPS.DAEMH600'
I KSDS=XPSKSDS
O EMAIL=KSDS.TXT CP=1141-E2N
O IPADDR=&SMTP_SRVR IPPORT=&SMTP_SRVR_PORT
O USER=&EMAIL_ACCOUNT PWD=&EMAIL_ACCOUNT_PWD
O FROM=&EMAIL_FROM
O TO=&EMAIL_RECIPIENT
O SUBJECT= 'KOMPONISTEN'
O BODY:
O EMAIL FROM ZVSE
O
O MFG
O END:
/*
/&
* $$EQJ
```

Abb. 81: Beispieljob für E-Mail Ausgabe z/VSE

Die nachfolgende Tabelle zeigt die zulässigen Optionen für Karten des Typs 'O EMAIL='.

O EMAIL=	
Versenden von Daten als E-Mail Anhang.	
Optionen	Beschreibung
<i>BCC=</i>	E-Mail Adresse(n)
	<p>Mit dieser Option sind die E-Mail Adressen all der Empfänger anzugeben, die eine blinde Kopie der Mail erhalten sollen. Die Empfänger blinder Kopien erscheinen nicht in der Statusinformation für eine versendete E-Mail und können auf diese Weise geheim gehalten werden. Für jeden Empfänger einer blinden Kopie ist eine eigene 'BCC=...' Angabe erforderlich.</p>
Beispiel	<p>BCC=blindcarboncopyrecipient1@company.com BCC=blindcarboncopyrecipient2@company.com</p>
<i>BODY=</i>	<p>BODY: Text END:</p>
	<p>Mit dieser Option kann eine beliebige Anzahl an Zeilen definiert werden, die als Nachrichtentext erscheinen.</p> <p>Dazu ist zur Anzeige des Beginns des Nachrichtentextes eine Output Karte mit dem Inhalt 'BODY:' anzugeben.</p> <p>Daran anschließend sind alle Textzeilen in Output Karten anzugeben, aus denen der Nachrichtentext bestehen soll.</p> <p>Abschließend ist eine Output Karte mit dem Inhalt 'END:' anzugeben, um das Ende des Nachrichtentextes anzuzeigen.</p> <p>Unter Verwendung der 'INCLUDE' Option kann der Nachrichtentext auch aus einem Bibliotheksmitglied kopiert werden.</p>
Beispiel	<p>BODY: Nicht vergessen: 16:50 Uhr ab Paddington! END:</p>
<i>CC=</i>	E-Mail Adresse(n)
	<p>Mit dieser Option sind die E-Mail Adressen all der Empfänger anzugeben, die eine Kopie der Mail erhalten sollen. Für jeden Empfänger einer Kopie ist eine eigene 'CC=...' Angabe erforderlich.</p>
Beispiel	<p>CC=carboncopyrecipient1@company.com CC=carboncopyrecipient2@company.com</p>
<i>CPA2E=</i>	Schlüssel einer Codepage
	<p>Textinformationen, die im Rahmen des E-Mail-Protokolls zwischen beiden Parteien ausgetauscht werden, sind aus Gründen der Standardisierung immer in ASCII/ANSI kodiert.</p> <p>Mit dieser Option kann festgelegt werden, welche Codepage zur Übersetzung dieser</p>

	<p>Textdaten beim Lesen von Protokollinformationen verwendet werden soll.</p> <p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 23 beschrieben, global vorbelegt werden.</p> <p>Falls unter z/OS auf eine Angabe für diese Option verzichtet wird, werden die textbasierten Protokollinformationen unter Verwendung der im System eingestellten Codepage übersetzt, die bei Ausführung des Systemmakros 'XLATE' für die Übersetzung von ASCII/ANSI nach EBCDIC zum Einsatz kommt.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>CPA2E=1141-N2E</td> </tr> </table>	Beispiel	CPA2E=1141-N2E
Beispiel	CPA2E=1141-N2E		
CPE2A=	Schlüssel einer Codepage		
	<p>Textinformationen, die im Rahmen des E-Mail-Protokolls zwischen beiden Parteien ausgetauscht werden, sind aus Gründen der Standardisierung immer in ASCII/ANSI kodiert.</p> <p>Mit dieser Option kann festgelegt werden, welche Codepage zur Übersetzung dieser Textdaten beim Schreiben von Protokollinformationen verwendet werden soll.</p> <p>Der Wert für diese Option kann in der Initialisierungsdatei 'XPSHDINI', wie auf Seite 23 beschrieben, global vorbelegt werden.</p> <p>Falls unter z/OS auf eine Angabe für diese Option verzichtet wird, werden die textbasierten Protokollinformationen unter Verwendung der im System eingestellten Codepage übersetzt, die bei Ausführung des Systemmakros 'XLATE' für die Übersetzung von EBCDIC nach ASCII/ANSI zum Einsatz kommt.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>CPE2A=1141-E2N</td> </tr> </table>	Beispiel	CPE2A=1141-E2N
Beispiel	CPE2A=1141-E2N		
CSV=	beliebiges Zeichen, Standard: <u>Leerzeichen</u>		
	<p>Mit dieser Option ist das Zeichen festzulegen, das zZip zur Trennung der einzelnen Felder bei der Aufbereitung der Ausgabedaten verwenden soll. Die daraus resultierende csv-Datei (character separated values) kann, falls sie ins Netzwerk versendet wird, mit Standardprogrammen wie z. B. Microsoft Excel angezeigt werden.</p> <p>Damit diese Option verwendet werden kann, ist bei der Definition der Eingabedaten ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich, damit zZip den Satzaufbau kennt.</p>		
	<table border="1"> <tr> <td>Beispiel</td> <td>CSV=;</td> </tr> </table>	Beispiel	CSV=;
Beispiel	CSV=;		
CSVFMT=	'C' 'T'		
	<p>Mit dieser Option ist das Format der erstellten csv-Daten zu bestimmen.</p> <p>Damit zZip den Satzaufbau kennt, ist bei der Definition der Eingabedatei ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich.</p> <p>zZip unterstützt die Verarbeitung gepackter und hexadezimaler Daten.</p> <p>Folgende Optionen sind verfügbar:</p> <p>'C' Die Inhalte der einzelnen Satzfelder werden in der Weise komprimiert, dass abschließende Leerzeichen entfernt und durch das angegebene csv-Trennzeichen ersetzt werden.</p> <p>'T' Die Inhalte der einzelnen Satzfelder werden tabellarisch aufbereitet, so dass jedes Feld genau so lang dargestellt wird, wie der Definition der Satzstruktur festgelegt. Bei Bedarf werden führende oder abschließende Leerzeichen eingefügt. Schließlich wird jedes Feld mit dem definierten csv-Trennzeichen abgeschlossen.</p>		

CSVHDR=	'Y' ' <u>N</u> '		
	<p>Mit dieser Option ist festzulegen, ob zZip bei der Erstellung einer csv-Datei basierend auf einer Satzstruktur die Namen der Felder als erste Ausgabezeile einfügen soll.</p> <p>Damit zZip den Satzaufbau kennt, ist bei der Definition der Eingabedatei ein Verweis auf eine Satzstruktur auf der XPS HFS-Datei unter Verwendung der Option 'RECORD=' erforderlich.</p> <p>Folgende Optionen sind verfügbar:</p> <p>'Y' zZip fügt eine Überschriftzeile ein, die die Namen der einzelnen Felder der Satzstruktur, getrennt durch das mit 'CSV=' angegebene Zeichen, enthält.</p> <p>'<u>N</u>' zZip fügt keine Überschriftzeile ein.</p>		
EMAIL=	Name des Dateianhangs		
	Mit dieser Option ist der Name des Dateianhangs festzulegen.		
	<table border="1"> <tr> <td>Beispiel</td> <td>EMAIL=KSDS.TXT</td> </tr> </table>	Beispiel	EMAIL=KSDS.TXT
Beispiel	EMAIL=KSDS.TXT		
FROM=	E-Mail Adresse		
	Mit dieser Option ist die E-Mail Adresse anzugeben, die als Absenderadresse in der versendeten E-Mail erscheinen soll.		
	<table border="1"> <tr> <td>Beispiel</td> <td>FROM=someone@somewhere.biz</td> </tr> </table>	Beispiel	FROM=someone@somewhere.biz
Beispiel	FROM=someone@somewhere.biz		
IPADDR=	TCP/IP Adresse – numerisch oder als DNS-Name		
	Mit dieser Option ist die TCP/IP Adresse des Postausgangsservers anzugeben.		
	<table border="1"> <tr> <td>Beispiel</td> <td>IPADDR=smtpt.server.com</td> </tr> </table>	Beispiel	IPADDR=smtpt.server.com
Beispiel	IPADDR=smtpt.server.com		
IPPORT=	numerischer TCP/IP Port <= 65535 ' <u>25</u> '		
	Mit dieser Option ist der TCP/IP Port des Postausgangsservers anzugeben, falls dieser vom Standardwert Port 25 abweicht.		
	<table border="1"> <tr> <td>Beispiel</td> <td>IPPORT=4125</td> </tr> </table>	Beispiel	IPPORT=4125
Beispiel	IPPORT=4125		
PWD=	Passwort		
	Mit dieser Option ist, falls erforderlich, das Passwort anzugeben, der bei der Anmeldung an den Postausgangsserver übermittelt werden soll.		
	<table border="1"> <tr> <td>Beispiel</td> <td>PWD=verysecret</td> </tr> </table>	Beispiel	PWD=verysecret
Beispiel	PWD=verysecret		
SUBJECT=	Phrase		
	Mit dieser Option ist anzugeben, welcher Text im 'Betreff' der E-Mail erscheinen soll.		
	Falls kein Betreff angegeben wird, wird es abhängig vom verwendeten Postausgangsserver eventuell ein Standardbetreff generiert.		
	<table border="1"> <tr> <td>Beispiel</td> <td>SUBJECT='Umsätze KW 21'</td> </tr> </table>	Beispiel	SUBJECT='Umsätze KW 21'
Beispiel	SUBJECT='Umsätze KW 21'		
TO=	E-Mail Adresse(n)		
	Mit dieser Option sind die E-Mail Adressen aller Empfänger anzugeben. Für jeden		

	Empfänger ist eine eigene 'TO=...' Angabe erforderlich.	
	Beispiel	TO=recipient1@company.com TO=recipient2@company.com
TRACE=	'ALL' 'D' 'T' 'S' 'Z' oder eine, jeweils durch Komma getrennte, Kombination	
	Mit dieser Option können Traceoptionen für den Job festgelegt werden. Der Trace dient zur Ausgabe von Informationen, die bei Bedarf die Analyse des internen Ablaufs ermöglichen. Traceausgaben erfolgen direkt in das Job-Listing. Die folgende Tabelle enthält die verfügbaren Tracetypen:	
	Typ	Beschreibung
	<i>ALL</i>	Alle Tracetypen sind aktiviert.
	<i>D</i>	Der Datentrace wird aktiviert.
	<i>T</i>	Der TCP/IP Trace wird aktiviert.
	<i>S</i>	Der Speichertrace wird aktiviert.
	<i>Z</i>	Der ZIP/UNZIP Trace wird aktiviert.
	Mehrere Tracetypen können aktiviert werden, in dem diese, durch Komma getrennt, aufgelistet werden.	
	Beispiel	TRACE=S
USER=	Benutzernamen	
	Mit dieser Option ist der Benutzername anzugeben, der bei der Anmeldung an den Postausgangsserver übermittelt werden soll. Eine Angabe ist nur erforderlich, wenn der Postausgangsserver eine Benutzer Authentifizierung verlangt.	
	Beispiel	USER=user%provider.com
WTO=	'Y' ' <u>N</u> '	
	Mit dieser Option ist festzulegen, ob zZip beim Jobstart eine Nachricht über die verwendete Version und beim Jobende eine Nachricht über die Anzahl der verarbeiteten Sätze auf die Konsole ausgeben soll oder nicht. Mit einer Angabe kann die globale Voreinstellung – siehe Seite 34 - überschrieben werden.	

Satzstrukturen

Einige zZip Funktionen erlauben die Verwendung von Satzstrukturen. Satzstrukturen dienen der logischen Unterteilung eines Datensatzes in benannte Elemente (Felder), die jeweils in einem festen Abstand vom Satzanfang beginnen und sowohl eine konstante Länge als auch einen bestimmten Datentyp haben.

Damit eine Satzstruktur verwendet werden kann, muss die Definition der Struktur zuvor in eine XPS HFS-Datei importiert werden, die dem zZip Job zur Laufzeit zur Verfügung gestellt wird.

Zum Importieren einer Datenstruktur ist das Batchprogramm 'XPSSRVXB' auszuführen, das sich im Installationspaket befindet. Die nachfolgenden Abbildungen zeigen Beispieljobs für z/OS und z/VSE:

```
//JSERVEX JOB , 'SERVEX BATCH',CLASS=A,MSGCLASS=G,USER=XPSDAEM
//JSERVEX EXEC PGM=XPSSRVXB,REGION=32M
//STEPLIB DD DSN=XPSHDRV.V300.LOADLIB,DISP=SHR
//XPSDATA DD DSN=XPSADVW.V100.MACLIB,DISP=SHR
//XPSHFS DD DISP=SHR,DSN=XPSHDRV.SPOOL.SAVE
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
ADD_STRUCT NAME=XPS/ADVW2, X
DESC=ADVW-DEFINITION, X
MBR=XPSADVW, X
STRUCT=ADVW
```

Abb. 82: Job zum Strukturimport z/OS

```
* $$ JOB JNM=JSERVEX,DISP=D,PRI=3,CLASS=0
* $$ LST CLASS=A
// JOB JSERVEX
// LIBDEF PHASE,SEARCH=(XPS.HDRV300)
// DLBL XPSHFS,'XPSHDRV.SPOOL.SAVE',,VSAM,CAT=XPSUCAT
// ASSGN SYS005,FEE
// EXEC XPSSRVXB,SIZE=AUTO
ADD_STRUCT NAME=TEST/PL1/TEST1,DESC=PL1_INLINE,PL1=YES
DCL 1 FELDER (10),
      2 LINETBL          BIN FIXED(31),
      2 ZEILEN_NR       BIN (15) FIXED,
      2 KENNZ           CHAR (1) INIT('0');
/*
/&
* $$ EOJ
```

Abb. 83: Job zum Strukturimport z/VSE

Zur Steuerung des Batchprogramms 'XPSSRVXB' sind die folgenden Steuerkarten zu verwenden:

Struktur hinzufügen

ADD_STRUCT	
Mit dieser Steuerkarte können neue Satzstrukturen in eine XPS HFS-Datei importiert werden. Der Import kann entweder durch den Verweis auf eine Strukturdefinition in einem Bibliotheksmitglied (siehe JCL-Beispiel z/OS) oder durch das direkte Einfügen der Satzstruktur in den Job (siehe JCL-Beispiel z/VSE) erfolgen. Das Programm 'XPSSRVXB' kann Strukturen im Cobol und PL/1 Format importieren.	
JCL - z/OS	Beschreibung

<i>XPSDATA DD</i>	Angabe der Bibliothek, die das Member, aus dem die Datenstruktur importiert werden soll, enthält.
<i>XPSHFS DD</i>	Angabe der XPS HFS-Datei, in die die Struktur importiert werden soll.
JCL - z/VSE	Beschreibung
<i>DLBL XPSDATA</i>	Angabe der Bibliothek, die das Member, aus dem die Datenstruktur importiert werden soll, enthält.
<i>DLBL XPSHFS</i>	Angabe der XPS HFS-Datei, in die die Struktur importiert werden soll.
Optionen	Beschreibung
<i>NAME</i>	Mit dieser Option ist der Name anzugeben, unter dem die Struktur in der XPS HFS-Datei abzuspeichern ist. Alle Strukturen werden von 'XPSSRVBX' in das Unterverzeichnis 'STRUCTS' abgelegt. Durch Voranstellen eines Pfades vor den Namen (Pfadtrennzeichen = '/') kann die neue Struktur in ein bestimmtes Unterverzeichnis von 'STRUCTS' abgelegt werden, womit auf einfache Weise eine Gruppierung erfolgen kann.
<i>DESC</i>	Mit dieser Option kann eine Beschreibung für die zu importierende Struktur angegeben werden.
<i>MBR</i>	Falls die Struktur aus einem Member aus der mit 'XPSDATA' referenzierten Bibliothek importiert werden soll, ist mit dieser Option der Name des Bibliotheksmembers anzugeben, das die zu importierende Struktur enthält.
<i>STRUCT</i>	Im Falle des Imports aus einem Bibliotheksmembers ist mit dieser Option der Schlüssel der zu importierenden Struktur anzugeben. Für Cobol-Strukturen ist dies die Bezeichnung der 01-er Stufennummer, für PL/1 die der 'DCL 1' Deklaration.
<i>PL1</i>	Falls eine PL/1 Struktur importiert werden soll, ist die Angabe von 'PL1=YES' erforderlich.

Struktur löschen

DEL_STRUCT	
Mit dieser Steuerkarte können bestehende Satzstrukturen aus einer XPS HFS-Datei gelöscht werden.	
JCL - z/OS	Beschreibung
<i>XPSHFS DD</i>	Angabe der XPS HFS-Datei, die die zu löschende Struktur enthält.
JCL - z/VSE	Beschreibung
<i>DLBL XPSHFS</i>	Angabe der XPS HFS-Datei, die die zu löschende Struktur enthält.
Optionen	Beschreibung
<i>NAME</i>	Mit dieser Option ist der Name der Struktur anzugeben, die gelöscht werden soll.

Beim Einsatz einer HostDrive/z Basisinstallation können die verfügbaren Strukturen, die sich auf der angebotenen XPS HFS-Datei befinden, über die Online Verwaltungsfunktion zur Anzeige der Satzstrukturen eingesehen werden. Die Beschreibung der HostDrive/z Verwaltungsfunktionen ist Gegenstand eines eigenen Handbuchs. Die nachfolgende Abbildung zeigt ein Beispiel für die Anzeige einer Satzstruktur.

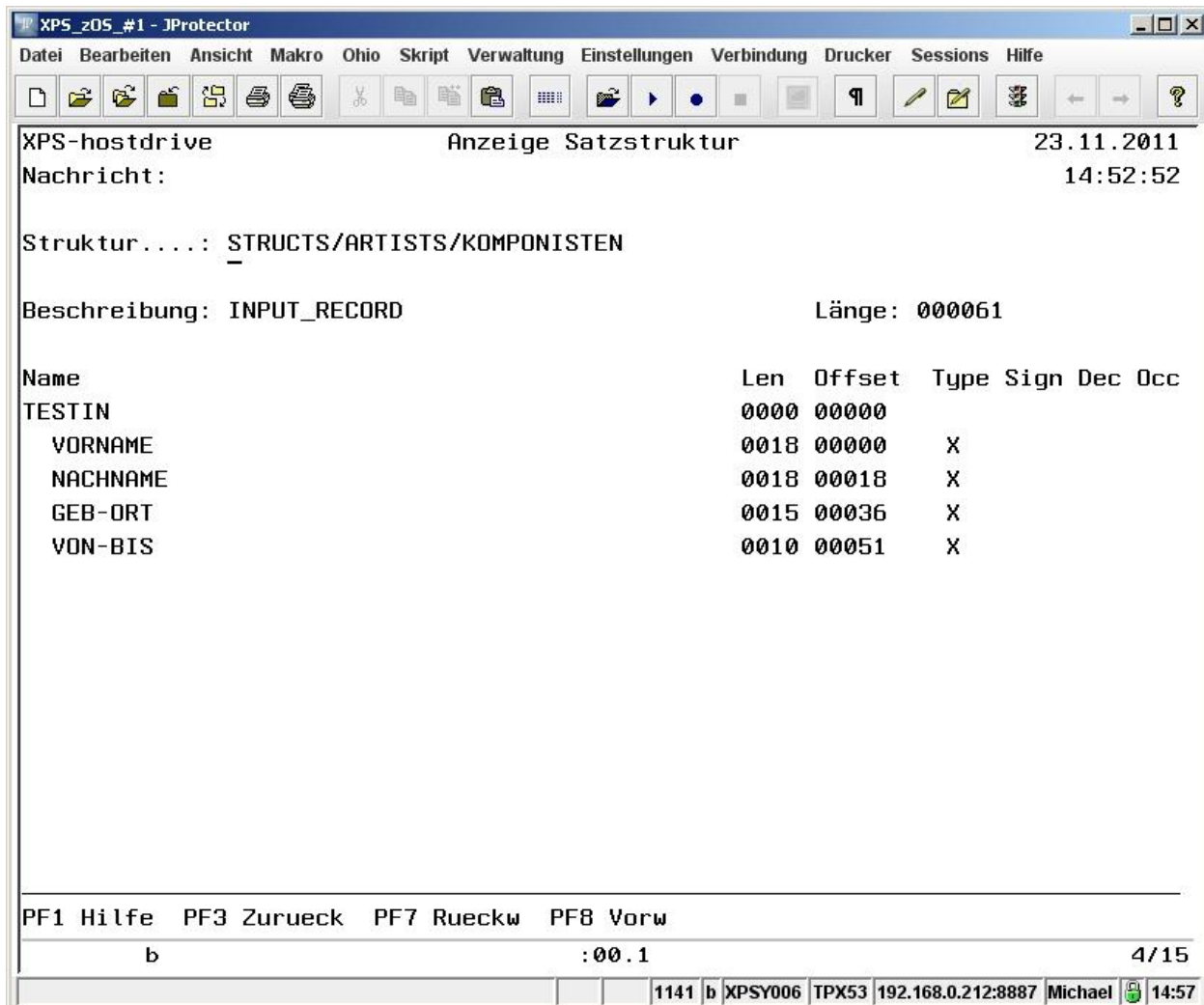


Abb. 84: Satzstrukturanzeige Online HostDrive/z

Records

Um eine Satzstruktur in einem zZip Job zu verwenden, ist ein Verweis auf die gewünschte Struktur im Rahmen der Definition von Eingabedaten erforderlich. Die nachfolgende Abbildung zeigt einen Auszug aus einem Job, der auf eine Datenstruktur verweist:

```

...
//SYSIN DD *
I IPT RECORD=ARTISTS/KOMPONISTEN
O ESDS=ESDSOUT CSV= CSVHDR=Y CSVFMT=T TRUNC=Y
//SYSIN2 DD *
GEORG FRIEDRICH HAENDEL HALLE 1685-1759
JOHANN SEBASTIAN BACH EISENACH 1685-1750
...

```

Abb. 85: Beispiel für die Verwendung einer RECORD-Angabe

Im gezeigten Beispiel wird zZip unter Verwendung der 'RECORD'-Angabe über die Struktur der IPT-Eingabedaten informiert, die im Anschluss an das 'SYSIN2'-Statement folgen.

Der nachfolgend abgebildete Jobauszug zeigt, wie der Umfang der zu verarbeitenden Satzfelder mit Hilfe der 'R'-Steuerkarten eingeschränkt werden kann:

```

...
//SYSIN DD *
I IPT RECORD=ARTISTS/KOMPONISTEN
O ESDS=ESDSOUT CSV= CSVHDR=Y CSVFMT=T TRUNC=Y
R VORNAME NACHNAME
R GEB-ORT
//SYSIN2 DD *
GEORG FRIEDRICH HAENDEL HALLE 1685-1759
JOHANN SEBASTIAN BACH EISENACH 1685-1750
...

```

Abb. 86: Beispiel für die Verwendung von 'R'-Steuerkarten

Unter Verwendung der 'R'-Steuerkarten wird im Beispiel der Umfang der zu berücksichtigenden Felder der mit 'RECORD=' referenzierten Satzstruktur auf 'VORNAME', 'NACHNAME' und 'GEB-ORT' eingeschränkt.

Damit würde die erstellte ESDS-Datei (Steuerkarte 'O ESDS=') die Angabe für das Geburts- und Todesjahr des jeweiligen Komponisten ('VON-BIS') nicht enthalten.

zZip erlaubt die Angabe beliebig vieler 'R'-Steuerkarten, wobei jede Steuerkarte den Namen eines oder mehrerer Felder der Satzstruktur enthalten kann.

Exit

Neben dem Aufruf des zZip Prozessors 'XPSZZIP' über Job Control besteht auch die Möglichkeit des Aufrufs aus einem anderen Programm unter Verwendung der CALL-Methode.

Damit können selbst entwickelte Programme auf einfache und komfortable Weise mit Komprimier- bzw. Entkomprimierfunktionalität ausgestattet werden.

Für den externen Aufruf von XPSZZIP ist zunächst eine Parameterliste aufzubauen, die dem erwarteten Format entspricht. Diese Parameterliste ist dann im Rahmen des CALL-Aufrufs an XPSZZIP zu übergeben, das das Ergebnis nach Beendigung der Verarbeitung an das rufende Programm zurückgibt.

Die Parameterliste muss das folgende Format haben:

Offset	Beschreibung
0-1	Länge aller sich anschließender Parameter exklusive der Länge dieses Feldes.
2-n	Erster Parameter. Falls weitere Parameter folgen, ist das Ende des Parameters mit einem abschließenden Leerzeichen kenntlich zu machen.
(n+1)-m ...	Optional: beliebig viele weitere Parameter, mit Ausnahme des letzten Parameters jeweils mit einem Leerzeichen terminiert.

Das Format der übergebenen Parameter ist identisch mit dem Format, das auch im Rahmen der JCL Input Statements für XPSZZIP Verwendung findet.

Damit steht dem aufrufenden Programm exakt derselbe Funktionsumfang zur Verfügung wie bei Ausführung von XPSZZIP im Rahmen der Ausführung eines Batchjobs.

Der nachfolgend abgebildete Quellcode zeigt ein Beispiel für den Aufruf von XPSZZIP aus einem Cobol Programm.

```
CBL APOST,LIB,LIST,MAP,XREF,SOURCE,DATA(31),RMODE(ANY),DYNAM
*
  ID DIVISION.
  PROGRAM-ID.
    CALLZIP.
*
  ENVIRONMENT DIVISION.
  CONFIGURATION SECTION.
*
  INPUT-OUTPUT SECTION.
*
  FILE-CONTROL.
*
  DATA DIVISION.
  FILE SECTION.
*
  WORKING-STORAGE SECTION.
*
  01 RC PIC 9(4) COMP VALUE 0.
  01 ZIPPARMS.
    05 PARMSTART PIC 9(4) COMP VALUE 131.
    05 FILLER PIC X(25) VALUE 'I LIB=XPSDAEM.V500.MACLIB'.
    05 FILLER PIC X(23) VALUE ' MBR=XPSVP01 OBSERVE=NO'.
    05 FILLER PIC X(1) VALUE ' '.
    05 FILLER PIC X(32) VALUE 'O ARCHIVE=NET=CEETEMP/XPSSEQ.ZIP'.
    05 FILLER PIC X(12) VALUE ' CP=1141-E2N'.
    05 FILLER PIC X(18) VALUE ' IPADDR=&IPA_TEST1'.
    05 FILLER PIC X(1) VALUE ' '.
    05 FILLER PIC X(19) VALUE 'O IPPORT=&IPP_TEST1'.
*
  LINKAGE SECTION.
*
  PROCEDURE DIVISION.
  MAIN SECTION.
*
    DISPLAY ZIPPARMS.
    CALL 'XPSZZIP' USING ZIPPARMS
      RETURNING RC.
*
  STOP RUN.
ENDRUN.
```

Abb. 87: COBOL Beispielprogramm für XPSZZIP Exit

Verzeichnis der Beispieljobs

Die nachfolgenden Tabellen enthalten die verschiedenen Beispieljobs, die in diesem Dokument abgedruckt sind, sortiert nach den verschiedenen Eingabe- und Ausgabetypen.

Beispieljobs für z/OS

Eingabetyp	Jobnamen	Seite
ESDS	JESD2HFS	171
	JESD2XHF	122
FTP	JFTP2LIB	141
	JFTP2SEQ	129
HFS	JHFS2JES	174
	JHFS2TAP	133
IPT	JIPT2KSD	160
	JIPT2NET	106
JES	JJES2FTP	167
	JJES2NET	135
KSDS	JKSD2EML	181
	JKSD2JES	124
	JKSD2NET	144
LIB	JLIB2ESD	109
	JLIB2TAP	152
NET	JNET2HFS	112
	JNET2SEQ	148
SEQ	JSEQ2FTP	115
	JSEQ2XHF	164
TAPE	JTAP2EML	118
XPSHFS	JXHF2ESD	156
	JXHF2LIB	127
Ausgabetypp	Jobnamen	Seite
EMAIL	JKSD2EML	181
	JTAP2EML	118
ESDS	JLIB2ESD	109
	JXHF2ESD	156
FTP	JJES2FTP	167
	JSEQ2FTP	115
HFS	JESD2HFS	171
	JNET2HFS	112
JES	JHFS2JES	174

	JKSD2JES	124
KSDS	JIPT2KSD	160
LIB	JFTP2LIB JXHF2LIB	141 127
NET	JIPT2NET JJES2NET JKSD2NET	106 135 144
SEQ	JFTP2SEQ JNET2SEQ	129 148
TAPE	JHFS2TAP JLIB2TAP	133 152
XPSHFS	JESD2XHF JSEQ2XHF	122 164
Aktion	Jobname	Seite
DECRYPT	JCRLI2NE	92
ENCRYPT	JCRLI2NE	92
GUNZIP	JGZIPS2S	42
GZIP	JGZIPS2S	42
UNZIP	JZILI2NE	57
ZIP	JZILI2NE	57
ZIP,ADD	JZIPADD	78
ZIP,COPY	JZIPCOPY	83
ZIP,DEL	JZIPDEL	85
ZIP,UPD	JZIPUPD	86
ZIP,VIEW	JZIPVIEW	91
Execute	Jobname	Seite
CMD	JCMDXBAT	99
WFD	JCMDXWFD	102

Beispieljobs für z/VSE

Eingabetyp	Jobnamen	Seite
ESDS	JESD2XHF	122
FTP	JFTP2LIB JFTP2POW JFTP2SEQ	141 177 129
IPT	JIPT2KSD JIPT2NET	160 106
KSDS	JKSD2EML JKSD2NET JKSD2POW	181 144 124
LIB	JLIB2ESD JLIB2TAP	109 152
NET	JNET2LIB JNET2SEQ	112 148
POWER	JPOW2FTP JPOW2NET	167 137
SEQ	JSEQ2FTP JSEQ2XHF	115 164
TAPE	JTAP2EML	119
XPSHFS	JXHF2ESD JXHF2LIB	156 127
Ausgabetyt	Jobnamen	Seite
EMAIL	JKSD2EML JTAP2EML	181 119
ESDS	JLIB2ESD JXHF2ESD	109 156
FTP	JPOW2FTP JSEQ2FTP	167 115
KSDS	JIPT2KDS	160
LIB	JFTP2LIB JNET2LIB JXHF2LIB	141 112 127
NET	JIPT2NET JKSD2NET JPOW2NET	106 144 137
POWER	JFTP2POW JKSD2POW	177 124

SEQ	JFTP2SEQ JNET2SEQ	129 148
TAPE	JLIB2TAP	152
XPSHFS	JESD2XHF JSEQ2XHF	122 164
Aktion	Jobname	Seite
DECRYPT	JCRLI2NE	93
ENCRYPT	JCRLI2NE	93
GUNZIP	JGZIPS2S	43
GZIP	JGZIPS2S	43
UNZIP	JZILI2NE	58
ZIP	JZILI2NE	58
ZIP,ADD	JZIPADD	78
ZIP,COPY	JZIPCOPY	83
ZIP,DEL	JZIPDEL	85
ZIP,UPD	JZIPUPD	86
ZIP,VIEW	JZIPVIEW	91
Execute	Jobname	Seite
CMD	JCMDXBAT	99
WFD	JCMDXWFD	102

Index

&			
&&FILE	37	JES	174
&DDD	37	KSDS	160
&HH	37	LIB	141
&HHMN	37	NET	144
&HHMNSS	37	POWER	177
&JOBNAME	36	SEQ	148
&LPAR	37	TAPE	152
&MM	37	XPSHFS	164
&MN	37		
&SS	37	B	
&STEPNAME	36	BCC	182
&STEPPROCNAME	36	BLKSIZE	116, 119, 153
&SYSUID	36	BLOWFISH	25, 26, 63, 81, 89, 94, 95, 97, 98
&VM	37	BODY	182
&YY	37	BUFFSIZE	22
&YYDDD	37		
&YYMM	37	C	
&YYMMDD	37	CBC	25, 62, 80, 88, 95, 98
&YYYY	37	CBC_CTS	25, 62, 80, 88, 95, 98
&YYYYDDD	37	CC	182
&YYYYMMDD	37	CFB	25, 62, 80, 88, 95, 98
		CLA	174, 177
3		CMD	100
3DES	24, 63, 81, 89, 95, 98	CNVEXT	22, 51, 70
		Codepage	13
A		COMMCHAR	23
AE1	63, 81, 88	COMMENT	61, 79, 87
AE2	63, 81, 88	COMPRLVL	22, 46, 61, 79, 87
AES	24, 26, 63, 81, 89, 94, 95, 97, 98	CONCAT	23, 46
Aktionen	42	CONTINUE	22
CMD	99	CP	113, 130, 167
DECRYPT	92	CPA2E	23, 104, 130, 140, 168, 182
ENCRYPT	92	CPE2A	23, 105, 130, 141, 168, 183
GUNZIP	42, 51	CRYPTOFACILITY	24, 62, 70, 79, 87, 93, 96
GZIP	42, 43	CRYPTOIV	25, 94, 96
UNZIP	57, 70	CRYPTOKEY	26, 94, 97
Entschlüsselung	57	CRYPTOMETHOD	24, 95, 97
WFD	102	CRYPTOMODE	25, 62, 80, 88, 95, 98
ZIP	57, 58	CSV	142, 145, 149, 153, 157, 165, 168, 172, 174, 177, 183
ADD	78	CSVFMT	142, 145, 149, 153, 157, 165, 168, 172, 174, 177, 183
COPY	83	CSVHDR	143, 145, 149, 153, 158, 166, 169, 172, 175, 178, 184
DEL	85	CTLCH	175, 178
UPD	86	CTR	25, 62, 80, 88, 95, 98
Verschlüsselung	57		
VIEW	91	D	
ARC_ATTR	21	DATEFMT	26
ARC_AUTODEL	22	DES	24, 26, 63, 81, 89, 94, 95, 97, 98
ARC_BLKSIZE	19, 43, 58	DEST	175, 178
ARC_DATACLASS	19, 44, 59	DISP	100
ARC_DIR_BLKS	19, 44, 59		
ARC_DSORG	20, 44, 59	E	
ARC_LRECL	20, 44, 59	ECB	25, 62, 80, 88, 95, 98
ARC_MGMTCLASS	20, 44, 59	ECB_CTS	25, 62, 80, 88, 95, 98
ARC_RECFCM	20, 44, 59	ECHO	26
ARC_SPACE_PRI	20, 45, 60	Eingabedaten	103
ARC_SPACE_RLSE	20, 45, 60	ARCHIVE	104
ARC_SPACE_SEC	21, 45, 60	ESDS	122
ARC_SPACE_TYPE	21, 45, 60	FTP	129
ARC_STORCLASS	21, 45, 60	HFS	133
ARC_UNIT	21, 45, 60	IPT	106
ARC_VOLUME	21, 45, 61	JES	135
ATTR	46, 61	KSDS	124
Ausgabedaten	139	LIB	109
ARCHIVE	140	NET	112
EMAIL	181	POWER	136
ESDS	156	SEQ	115
FTP	167	TAPE	118
HFS	171	XPSHFS	127

EMAIL..... 184
 ENCRYPT..... 63, 80, 88
 ESDS..... 122, 158
 Exit..... 189
 EXTSEP..... 26, 64

F

FILE_TERM 27, 46, 51, 64, 70
 Filter..... 39
 EXCLUDE..... 40
 INCLUDE..... 41
 FORM..... 175, 178
 FROM..... 184
 FTP..... 130, 169
 FTPCMD..... 131, 169

H

HFS..... 133, 172
 HostDrive/J... 7, 9, 57, 78, 86, 92, 99, 100, 106, 107, 112, 113, 144, 146, 148
 HostDrive/z – z/OS
 Installation..... 9
 HostDrive/z – z/VSE
 Installation..... 12

I

Installation
 HostDrive/z – z/OS..... 9
 HostDrive/z – z/VSE..... 12
 IPADDR..... 27, 100, 113, 131, 146, 169, 184
 IPPORT..... 28, 101, 113, 131, 146, 169, 184

J

JES..... 135, 175
 JNAME..... 176, 178

K

KEY..... 124
 Kontrollkarten..... 38
 Aktion..... 38
 Execute..... 39
 Filter..... 38
 Input..... 38
 Output..... 38
 Record..... 38
 KSDS..... 125, 161

L

LABEL..... 120, 153
 LIB..... 110

M

MBR..... 110, 143
 MODE..... 28, 47, 52, 65
 MODE_DETECT_LEN..... 28, 47, 65
 MODE_DETECT_TAB..... 28, 48, 65

N

NET..... 113, 146
 NIA..... 48, 65
 NOA..... 52, 71

O

O LIB= Karte Verzicht..... 142
 O SEQ= Karte Verzicht..... 148
 OBSVADDR..... 29
 OBSVPORT..... 29
 OFB..... 25, 62, 80, 88, 95, 98
 OPT..... 125, 158, 162, 166, 172

OPTION..... 81, 84

Optionen

Aktionen

DECRYPT

CRYPTOFACILITY..... 96
 CRYPTOIV..... 96
 CRYPTOKEY..... 97
 CRYPTOMETHOD..... 97
 CRYPTOMODE..... 98

ENCRYPT

CRYPTOFACILITY..... 93
 CRYPTOIV..... 94
 CRYPTOKEY..... 94
 CRYPTOMETHOD..... 95
 CRYPTOMODE..... 95

GUNZIP

CNVEXT..... 51
 FILE_TERM..... 51
 MODE..... 52
 NOA..... 52
 OUT_BLKSIZE..... 53
 OUT_DATACLASS..... 53
 OUT_DIR_BLKs..... 54
 OUT_DSORG..... 54
 OUT_LRECL..... 54
 OUT_MGMTCLASS..... 54
 OUT_RECFM..... 54
 OUT_SPACE_PRI..... 54
 OUT_SPACE_RLSE..... 55
 OUT_SPACE_SEC..... 55
 OUT_SPACE_TYPE..... 55
 OUT_STORCLASS..... 55
 OUT_UNIT..... 55
 OUT_VOLUME..... 55
 OVERWRITE..... 56
 PATHSEP..... 56
 REC_DELIM..... 56

GZIP

ARC_BLKSIZE..... 43
 ARC_DATACLASS..... 44
 ARC_DIR_BLKs..... 44
 ARC_DSORG..... 44
 ARC_LRECL..... 44
 ARC_MGMTCLASS..... 44
 ARC_RECF..... 44
 ARC_SPACE_PRI..... 45
 ARC_SPACE_RLSE..... 45
 ARC_SPACE_SEC..... 45
 ARC_SPACE_TYPE..... 45
 ARC_STORCLASS..... 45
 ARC_UNIT..... 45
 ARC_VOLUME..... 45
 ATTR..... 46
 COMPRLVL..... 46
 CONCAT..... 46
 FILE_TERM..... 46
 MODE..... 47
 MODE_DETECT_LEN..... 47
 MODE_DETECT_TAB..... 48
 NIA..... 48
 PATH..... 49
 PATHSEP..... 49
 PKCP..... 49
 REC_DELIM..... 49

UNZIP

CNVEXT..... 70
 CRYPTOFACILITY..... 70
 FILE_TERM..... 70
 NOA..... 71
 OUT_BLKSIZE..... 73
 OUT_DATACLASS..... 73
 OUT_DIR_BLKs..... 73
 OUT_DSORG..... 73
 OUT_LRECL..... 73
 OUT_MGMTCLASS..... 73
 OUT_RECFM..... 74
 OUT_SPACE_PRI..... 74

OUT_SPACE_RLSE	74	CSV	183
OUT_SPACE_SEC	74	CSVFMT	183
OUT_SPACE_TYPE	74	CSVHDR	184
OUT_STORCLASS	74	EMAIL	184
OUT_UNIT	75	FROM	184
OUT_VOLUME	75	IPADDR	184
OVERWRITE	75	IPPORT	184
PATHSEP	75	PWD	184
PKCP	75	SUBJECT	184
PROGRESS	75	TO	184
PWD	75	TRACE	185
REC_DELIM	76	USER	185
SIMULATE	76	WTO	185
ZIP		ESDS	
ARC_BLKSIZE	58	CSV	157
ARC_DATACLASS	59	CSVFMT	157
ARC_DIR_BLKS	59	CSVHDR	158
ARC_DSORG	59	ESDS	158
ARC_LRECL	59	OPT	158
ARC_MGMTCLASS	59	TRACE	158
ARC_RECFCM	59	TRUNC	158
ARC_SPACE_PRI	60	WTO	159
ARC_SPACE_RLSE	60	FTP	
ARC_SPACE_SEC	60	CP	167
ARC_SPACE_TYPE	60	CPA2E	168
ARC_STORCLASS	60	CPE2A	168
ARC_UNIT	60	CSV	168
ARC_VOLUME	61	CSVFMT	168
ATTR	61	CSVHDR	169
COMMENT	61	FTP	169
CRYPTOFACILITY	62	FTPCMD	169
CRYPTOMODE	62	IPADDR	169
ENCRYPT	63	IPPORT	169
EXTSEP	64	PWD	170
FILE_TERM	64	TRACE	170
MODE	65	USER	170
MODE_DETECT_LEN	65	WTO	170
MODE_DETECT_TAB	65	HFS	
NIA	65	CSV	172
PATH	66	CSVFMT	172
PATHSEP	67	CSVHDR	172
PKCP	67	HFS	172
PROGRESS	67	OPT	172
PWD	67	TRACE	173
REC_DELIM	68	WTO	173
SAVE_LRECL	68	JES	
SIMULATE	68	CLA	174
ZIP,ADD		CSV	174
COMMENT	79	CSVFMT	174
CRYPTOFACILITY	79	CSVHDR	175
CRYPTOMODE	80	CTLCH	175
ENCRYPT	80	DEST	175
OPTION	81	FORM	175
PATH	82	JES	175
PROGRESS	82	JNAME	176
PWD	82	PDISP	176
ZIP,COPY		TRACE	176
OPTION	84	WTO	176
PROGRESS	84	KSDS	
ZIP,UPD		KSDS	161
COMMENT	87	OPT	162
CRYPTOFACILITY	87	TRACE	162
CRYPTOMODE	88	TRUNC	162
ENCRYPT	88	WARN	162
PATH	89	WTO	163
PROGRESS	90	LIB	
PWD	90	CSV	142
Ausgabedaten		CSVFMT	142
ARCHIVE		CSVHDR	143
CPA2E	140	MBR	143
CPE2A	141	TRACE	143
EMAIL		WTO	143
BCC	182	NET	
BODY	182	CSV	145
CC	182	CSVFMT	145
CPA2E	182	CSVHDR	145
CPE2A	183	IPADDR	146

IPPORT	146	RECL	133
NET	146	RECORD	133
ROUTE	146	TRACE	134
TRACE	146	WFD	134
WTO	147	WTO	134
POWER		IPT	
CLA	177	RECORD	107
CSV	177	TRACE	107
CSVFMT	177	WFD	107
CSVHDR	178	WTO	108
CTLCH	178	JES	
DEST	178	JES	135
FORM	178	TRACE	136
JNAME	178	WTO	136
PDISP	178	KSDS	
POWER	179	KEY	124
PUSER	179	KSDS	125
RECL	179	OPT	125
TRACE	179	RECORD	125
WTO	179	TRACE	125
SEQ		WFD	125
CSV	149	WTO	126
CSVFMT	149	LIB	
CSVHDR	149	LIB	110
RECFM	149	MBR	110
RECL	150	RECORD	110
SEQ	150	TRACE	110
TRACE	150	WFD	111
TRUNC	151	WTO	111
WTO	151	NET	
TAPE		CP	113
BLKSIZE	153	IPADDR	113
CSV	153	IPPORT	113
CSVFMT	153	NET	113
CSVHDR	153	RECL	113
LABEL	153	RECORD	114
RECFM	154	ROUTE	114
RECL	154	TRACE	114
REW	154	WTO	114
TAPE	154	POWER	
TRACE	154	POWER	137
TRUNC	155	TRACE	138
WTO	155	WTO	138
XPSHFS		SEQ	
CSV	165	BLKSIZE	116
CSVFMT	165	RECFM	116
CSVHDR	166	RECL	116
OPT	166	RECORD	117
TRACE	166	SEQ	117
WTO	166	TRACE	117
XPSHFS	166	WFD	117
Eingabedaten		WTO	117
ARCHIVE		TAPE	
CPA2E	104	BLKSIZE	119
CPE2A	105	LABEL	120
ESDS		RECFM	120
ESDS	122	RECL	120
RECORD	123	RECORD	120
TRACE	123	REW	120
WFD	123	TAPE	121
WTO	123	TRACE	121
FTP		WFD	121
CP	130	WTO	121
CPA2E	130	XPSHFS	
CPE2A	130	RECORD	127
FTP	130	TRACE	127
FTPCMD	131	WFD	128
IPADDR	131	WTO	128
IPPORT	131	XPSHFS	128
PWD	131	Execute	
RECORD	131	CMD	100
TRACE	131	DISP	100
USER	132	IPADDR	100
WFD	132	IPPORT	101
WTO	132	PARM	101
HFS		ROUTE	101
HFS	133		

&YYYYDDD 37
 &YYYYMMDD 37

T

TAPE 121, 154
 TCPJOB 33
 TO 184
 TRACE 107, 110, 114, 117, 121, 123, 125, 127, 131, 134, 136,
 138, 143, 146, 150, 154, 158, 162, 166, 170, 173, 176, 179, 185
 TRUNC 151, 155, 158, 162

U

USER 132, 170, 185

V

Variablen 35
 Verschlüsselung
 PKWARE
 schwach 57
 stark 57
 WinZip
 stark 57
 Verzicht auf O LIB= Karte 142
 Verzicht auf O SEQ= Karte 148

W

WAIT 101
 WARN 162

wfd 22, 107, 111, 121, 123, 125, 128, 132, 134
 Verarbeitung 102
 WFD 34, 107, 111, 117, 121, 123, 125, 128, 132, 134
 wfd-Datei
 Definition
 z/OS 11
 z/VSE 12
 Wildcards 39
 WTO 34, 108, 111, 114, 117, 121, 123, 126, 128, 132, 134, 136,
 138, 143, 147, 151, 155, 159, 163, 166, 170, 173, 176, 179, 185

X

XPSDATA
 z/OS 16
 z/VSE 18
 XPSHDINI 18
 XPSHFS 34, 128, 166
 XPSHFS-Datei
 Definition
 z/OS 11
 z/VSE 12
 XPSZIP
 Exit 189

Z

zIIP 7, 24, 35, 62, 70, 79, 87, 93, 96
 ZIIP 35
 ZIIPPERC 35