

Deutsch



FUJITSU Software BS2000

BS2ZIP V1.2G

Zip-Archivierung in BS2000

Benutzerhandbuch

Ausgabe November 2014

Kritik... Anregungen... Korrekturen...

Die Redaktion ist interessiert an Ihren Kommentaren zu diesem Handbuch. Ihre Rückmeldungen helfen uns, die Dokumentation zu optimieren und auf Ihre Wünsche und Bedürfnisse abzustimmen.

Sie können uns Ihre Kommentare per E-Mail an manuals@ts.fujitsu.com senden.

Zertifizierte Dokumentation nach DIN EN ISO 9001:2008

Um eine gleichbleibend hohe Qualität und Anwenderfreundlichkeit zu gewährleisten, wurde diese Dokumentation nach den Vorgaben eines Qualitätsmanagementsystems erstellt, welches die Forderungen der DIN EN ISO 9001:2008 erfüllt.

cognitas. Gesellschaft für Technik-Dokumentation mbH
www.cognitas.de

Copyright und Handelsmarken

Copyright © Fujitsu Technology Solutions GmbH 2014.

Alle Rechte vorbehalten.

Liefermöglichkeiten und technische Änderungen vorbehalten.

Alle verwendeten Hard- und Softwarenamen sind Handelsnamen und/oder Warenzeichen der jeweiligen Hersteller.

Inhalt

1	Einleitung	5
1.1	Zielsetzung und Zielgruppen des Handbuchs	5
1.2	Konzept des Handbuchs	6
1.3	Änderungen gegenüber dem Vorgänger-Handbuch	7
1.4	Darstellungsmittel	8
2	Überblick über die Anwendung BS2ZIP	9
2.1	ZIP-Archive	9
2.2	Dateien übertragen	15
2.3	Daten verschlüsseln	16
2.4	Unterstützte Dateitypen	16
2.5	Datei-Extents und benutzerspezifische Ressourcen	18
2.6	Unterstützung der K2-Taste	18
2.7	Wiederherstellung im Fehlerfall	19
3	Kommandos und Anweisungen	21
3.1	Starten und Beenden von BS2ZIP	21
3.1.1	Kommando zum Starten von BS2ZIP	21
3.1.2	Beenden von BS2ZIP	22
3.1.3	Meldungen	22

3.2	Beschreibung der Anweisungen	23
3.2.1	ADD-FILE	24
3.2.2	CONVERT-ZIP-CONTAINER	33
3.2.3	DELETE-FILE	36
3.2.4	END	37
3.2.5	EXTRACT-FILE	38
3.2.6	MODIFY-ZIP-OPTIONS	44
3.2.7	OPEN-ZIP-CONTAINER	46
3.2.8	REORGANIZE-ZIP-CONTAINER	49
3.2.9	SHOW-FILE-ATTRIBUTES	50
3.2.10	START-TRACE	56
3.2.11	STOP-TRACE	57
3.3	Beispiele	58
4	Interoperabilität	63
<hr/>		
4.1	Interoperabilität mit Windows	63
4.2	Interoperabilität mit Unix-Systemen	65
4.3	Interoperabilität mit Linux	65
5	Programmschnittstelle	67
<hr/>		
5.1	SZPZIP.H	67
5.2	SZPZOUT.H	80
5.3	BS2ZIPPR LLM	81
5.4	Programmbeispiel	82
	Literatur	83
<hr/>		
	Stichwörter	85
<hr/>		

1 Einleitung

BS2ZIP ist eine BS2000-TU-Anwendung mit einer einfachen und bedienerfreundlichen Schnittstelle für SDF-Anweisungen. Die Version V1.2 kann ab BS2000/OSD-BC V7.0 eingesetzt werden.

1.1 Zielsetzung und Zielgruppen des Handbuchs

Dieses Handbuch wendet sich sowohl an den privilegierten als auch an den nicht-privilegierten Benutzer von BS2000.

1.2 Konzept des Handbuchs

BS2ZIP stellt die folgenden Funktionen bereit:

- Erstellen eines BS2ZIP-Archivs kompatibel zu WinZip® (PKZIP 4.5).
- Hinzufügen von BS2000-Dateien beliebiger Art und Organisation (PAM, ISAM, SAM und PLAM, wobei auch separate Bibliothekselemente möglich sind) in dem im RFC-1950 definierten kompatiblen Zlib-Format. Copyright © 1995-2003 Jean-loup Gailly and Mark Adler:
jloup@gzip.org / madler@alumni.caltech.edu
- Extrahieren und Wiederherstellen der BS2000-Dateien mit ihren ursprünglichen DMS-Attributen.
- Anzeige der in einem BS2ZIP-Archiv enthaltenen verfügbaren Dateien.
- Unterstützung von mehreren Dateien in einem BS2ZIP-Archiv.
- Unterstützung der Standard-Zip-2.0-Verschlüsselung beim Hinzufügen oder Extrahieren von Dateien
- Interoperabilität mit dem Betriebssystem Windows:
 - Unter BS2000:
Öffnen von mit WinZip erstellten Archivdateien, die als BS2ZIP-Archiv in BS2000 transferiert wurden (Transfer mit ftp).
Extrahieren von Dateien als Textdateien (SAM-Dateien) und wahlweise Konvertierung von ISO8859F/WINANSI in EDF04F.
Wahlweises Extrahieren von Dateien als RAW-Dateien (Binär-Format, PAM-Dateien).
 - Unter Windows:
Öffnen von transferierten BS2ZIP-Archiven mit WinZip (Transfer mit ftp).
Die Anwendung stellt die Konvertierung von EBCDIC-Textdateien (wie z.B. SAM/ISAM) sicher (Konvertierung von EDF0x in ISO8859F/WINANSI).
Alle anderen Elemente werden als „binäre“ Datei angesehen.
 - Unterstützung des Transfers mit openFT < V11.0 durch Konvertierung des Dateiformats von BS2ZIP-Archiven (von SAM mit RECORD-FORMAT=U nach PAM und umgekehrt).
- Interoperabilität mit Linux- und Unix-Systemen:
Dateien eines GZIP-Archivs können gelesen oder extrahiert werden.

Außerdem steht eine Programm-Schnittstelle in C++ zur Verfügung, die die gesamte Funktionalität der TU-Anwendung bietet.

Readme-Datei

Funktionelle Änderungen der aktuellen Produktversion und Nachträge zu diesem Handbuch entnehmen Sie bitte ggf. der produktspezifischen Readme-Datei.

Readme-Dateien stehen Ihnen online bei dem jeweiligen Produkt zusätzlich zu den Produkthandbüchern unter <http://manuals.ts.fujitsu.com> zur Verfügung. Alternativ finden Sie Readme-Dateien auch auf der Softbook-DVD.

Informationen unter BS2000

Wenn für eine Produktversion eine Readme-Datei existiert, finden Sie im BS2000-System die folgende Datei:

```
SYSRME.<product>.<version>.<lang>
```

Diese Datei enthält eine kurze Information zur Readme-Datei in deutscher oder englischer Sprache (<lang>=D/E). Die Information können Sie am Bildschirm mit dem Kommando `/SHOW-FILE` oder mit einem Editor ansehen.

Das Kommando `/SHOW-INSTALLATION-PATH INSTALLATION-UNIT=<product>` zeigt, unter welcher Benutzerkennung die Dateien des Produkts abgelegt sind.

Ergänzende Produkt-Informationen

Aktuelle Informationen, Versions-, Hardware-Abhängigkeiten und Hinweise für Installation und Einsatz einer Produktversion enthält die zugehörige Freigabemitteilung. Solche Freigabemitteilungen finden Sie online unter <http://manuals.ts.fujitsu.com>.

1.3 Änderungen gegenüber dem Vorgänger-Handbuch

Für BS2ZIP V1.2G wurden folgende Änderungen implementiert:

- Anweisung ADD-FILE:
Bei den Operanden FROM-FILE und EXCEPT-FILE-NAME kann eine Datei angegeben werden, die eine Liste derjenigen Dateien enthält, die zum ZIP-Archiv hinzugefügt bzw. die nicht ins ZIP-Archiv übernommen werden sollen.
- Anweisung ADD-FILE und EXTRACT-FILE:
Der Operand LOGGING legt den Umfang der Meldungs Ausgabe fest.

1.4 Darstellungsmittel

In diesem Handbuch werden folgende Darstellungsmittel verwendet:



Dieses Zeichen kennzeichnet Hinweise auf wichtige Informationen



Dieses Zeichen kennzeichnet einen Warnhinweis, der auf die Möglichkeit des Datenverlustes oder anderer ernsthafter Schäden an Daten hinweist.

[]

Literaturhinweise werden im Text in Kurztiteln angegeben. Der vollständige Titel jeder Druckschrift, auf die durch eine Nummer verwiesen wird, ist im Literaturverzeichnis hinter der entsprechenden Nummer aufgeführt.

Eingabe

In Anwendungsbeispielen sind Eingaben an das System und Ausgaben des Systems in Schreibmaschinenschrift dargestellt.

2 Überblick über die Anwendung BS2ZIP

2.1 ZIP-Archive

Mit BS2ZIP können in BS2000-Systemen ZIP-Archive erstellt sowie Dateien zu diesen Archiven hinzugefügt bzw. aus ihnen extrahiert werden. Die ZIP-Datenkomprimierung erfolgt im Zlib-Format.

Diese Anwendung gestattet ferner das Lesen von ZIP-Archiven, die auf anderen Plattformen (Windows, UNIX-Systeme, Linux) erstellt wurden, sofern die Zlib-Datenkomprimierungsmethode verwendet wurde und sofern sie kompatibel mit PKZIP 4.5 oder GZIP sind.

Die in BS2000 erstellten Archive können auch auf fremden Plattformen geöffnet werden, wenn sie im WinZip-kompatiblen Format erstellt wurden.

Das Kommando zum Starten der Anwendung BS2ZIP lautet /START-ZIP-MANAGER bzw. /START-ZIP.

Nach dem Start der Anwendung kann man mit der Anweisung OPEN-ZIP-CONTAINER ein vorhandenes Archiv öffnen oder ein neues Archiv erstellen. Im Falle eines neuen Archivs muss der Benutzer das geeignete Archivformat in Abhängigkeit von dem geplanten Gebrauch des zu erstellenden Archivs (Operand FORMAT der Anweisung) wählen.

Bei der Wahl des richtigen Formats sind die folgenden Regeln zu beachten:

1. **Archiv erstellen:**

- Falls man beabsichtigt, das Archiv nur auf BS2000-Plattformen zu verwenden, kann man das Archiv im BS2000-Format erstellen.
- Beabsichtigt man, das Archiv auf fremde Plattformen zu exportieren, muss man das Archiv im Format *WINZIP-COMPATIBLE (Standardformat) erstellen.

2. **Archiv öffnen:**

Das Archiv-Format muss beim Öffnen eines bestehenden Archivs nicht spezifiziert werden. Das Programm findet das entsprechende Format selbst. Wenn das Format jedoch spezifiziert wird, muss es dem tatsächlichen Format des Archivs entsprechen, sonst wird die Anweisung OPEN-ZIP-CONTAINER abgelehnt. Nachdem das Archiv geöffnet wurde, können Dateien hinzugefügt oder extrahiert werden.

Die nachstehende Tabelle bietet einen Überblick über die standardmäßig vorgegebene Behandlung der verschiedenen Dateitypen durch BS2ZIP in Abhängigkeit vom Format, in dem das Archiv geöffnet wird:

ZIP-Format	Dateizugriffsmethode					
	SAM		ISAM		PAM/PLAM	
WINZIP-COMPATIBLE	A:	Nur die Daten werden gespeichert. WIN-ANSI-Konvertierung von EBCDIC-codierten Dateien; als Zeilentrenner wird CRLF (0D0A) eingefügt.	A:	Daten und Schlüssel werden gespeichert. WIN-ANSI-Konvertierung von EBCDIC-codierten Dateien; als Zeilentrenner wird CRLF (0D0A) eingefügt.	A:	Speicherung im Binärformat ohne Konvertierung.
	X:	Die Dateien werden mit denselben Eigenschaften wie die Originaldatei extrahiert. EBCDIC-Konvertierung bei Bedarf. Die Sätze werden in Einklang mit dem Zeilentrenner 0D0A wiederhergestellt. <i>Standardverarbeitung, falls der Datei keine Organisationsinformation zugeordnet wurde.</i>	X:	Die Dateien werden mit denselben Eigenschaften wie die Originaldatei extrahiert. EBCDIC-Konvertierung bei Bedarf. Die Sätze werden in Einklang mit dem Zeilentrenner 0D0A wiederhergestellt.	X:	Die Dateien werden mit denselben Eigenschaften wie die Originaldatei extrahiert. Neuerstellung im Binärformat.
BS2000	A:	Die Satzlänge wird mit den Daten gespeichert. Keine Konvertierung.	A:	Die Satzlänge wird mit den Daten gespeichert. Keine Konvertierung.	A:	Speicherung im Binärformat.
	X:	Die Dateien werden mit denselben Eigenschaften wie die Originaldatei extrahiert. Keine Konvertierung.	X:	Die Dateien werden mit denselben Eigenschaften wie die Originaldatei extrahiert. Keine Konvertierung.	X:	Die Dateien werden mit denselben Eigenschaften wie die Originaldatei extrahiert. Neuerstellung im Binärformat.

A: (add) Hinzufügen

X: (extract) Extrahieren

Bei den Anweisungen ADD-FILE und EXTRACT-FILE kann man dieses Standardverhalten auch ändern. Für weitere Einzelheiten siehe die Beschreibung dieser Anweisungen.

Hinweis

Bei einem WinZip-kompatiblen Archiv wird angenommen, dass CRLF (0D0A) das Ende eines Datensatzes kennzeichnet. Die Originaldatei darf keine solchen binären Werte enthalten. Anderenfalls wird die Datei beim Extrahieren „verdorben“. Ein WinZip-kompatibles Archiv sollte keine nicht abdruckbaren SAM- oder ISAM-Dateien enthalten.

ADD-FILE

CHARACTER-CONVERSION	Bedeutung
*BY-CONTAINER-FORMAT	Standardwert. Verhalten abhängig vom Format, mit dem das ZIP-Archiv geöffnet wird.
*NO	Es wird keine Konvertierung durchgeführt.
*TO-EBCDIC	EBCDIC-Konvertierung wird durchgeführt (nur bei ASCII-codierten SAM/ISAM-Dateien).
*TO-WIN-ANSI	WIN-ANSI-Konvertierung wird durchgeführt (nur bei EBCDIC-codierten SAM/ISAM-Dateien).

EXTRACT-FILE

Das durch das Format des ZIP-Archivs festgelegte Format zum Öffnen kann geändert werden:

DATA-TYPE	Archiv-Format	
	WINZIP-COMPATIBLE	BS2000
*NOT-SPECIFIED	Wenn im ZIP-Archiv keine Datei-Info gefunden wird, dann wird *CHARACTER angenommen, andernfalls wird die Datei-Info verwendet. EBCDIC-Konvertierung erfolgt gemäß Dateityp.	Wenn keine Datei-Info gefunden wird, dann führt dies zu einem Fehler, andernfalls wird die Datei-Info verwendet - keine Konvertierung
*CHARACTER	Die Datei wird als SAM-Datei extrahiert. Es wird davon ausgegangen, dass die Sätze durch 0D0A getrennt sind. EBCDIC-Konvertierung wird durchgeführt.	Anweisung wird zurückgewiesen.
*BINARY	Die Datei wird als PAM-Datei extrahiert. Es wird keine Konvertierung durchgeführt.	Anweisung wird zurückgewiesen.
*SAM-BINARY	Die Datei wird als SAM-Datei mit RECORD=U extrahiert, . Es wird keine Konvertierung durchgeführt.	Anweisung wird zurückgewiesen.

CHARACTER-CONVERSION	Bedeutung
*BY-CONTAINER-FORMAT	Standardwert. Das Verhalten ist abhängig vom Format, mit dem das ZIP-Archiv geöffnet wird.
*NO	Es wird keine Konvertierung durchgeführt.
*TO-WIN-ANSI	WIN-ANSI-Konvertierung wird durchgeführt (nur beim Entpacken in SAM/ISAM-Datei).
*TO-EBCDIC	EBCDIC-Konvertierung wird durchgeführt (nur beim Entpacken in SAM/ISAM-Datei).

Blockformat beim Extrahieren auf verschiedene Plattenformate

BS2ZIP berücksichtigt beim Extrahieren einer Datei das Format der Platte, auf der die Datei abgelegt wird, das im Archiv hinterlegte Blockformat (BLOCK-CONTROL-INFO) der Originaldatei und die Angabe im Operanden BLOCK-CONTROL-INFO, die steuert, ob das Blockformat beibehalten werden muss. Dies führt dazu, dass die Datei in einigen Fällen nicht extrahiert werden kann oder dass für die extrahierte Datei ein von der Originaldatei abweichendes Blockformat gewählt werden muss.

Extrahieren auf K-Platten

Die extrahierten Dateien erhalten das Blockformat der Originaldatei.

Extrahieren auf NK2-Platten

Dateien mit dem Blockformat PAMKEY können mit der Angabe BLOCK-CONTROL=*KEEP nicht extrahiert werden.

Abhängig vom Archivformat wird mit der Angabe BLOCK-CONTROL=*IGNORE das Blockformat wie folgt gesetzt:

Eigenschaften der Originaldatei		Angabe im Operanden BLOCK-CONTROL			
		(Format BS2000)		(Format WINZIP-COMPATIBLE)	
FCB-Type	BLK-CTRL	*KEEP	*IGNORE	*KEEP	*IGNORE
PAM	PAMKEY	nicht extrahiert	NO	nicht extrahiert	NO
PAM	DATA	wie Original	wie Original	wie Original	wie Original
PAM	NO	wie Original	wie Original	wie Original	wie Original
SAM	PAMKEY	nicht extrahiert	DATA	nicht extrahiert	DATA
SAM	DATA	wie Original	wie Original	wie Original	wie Original
ISAM	PAMKEY	nicht extrahiert	DATA2K	nicht extrahiert	DATA2K
ISAM	DATA	wie Original	wie Original	wie Original	wie Original

Extrahieren auf NK4-Platten, Datei mit geradem Blockungsfaktor

Dateien mit dem Blockformat PAMKEY können mit der Angabe BLOCK-CONTROL=*KEEP nicht extrahiert werden.

Abhängig vom Archivformat wird mit der Angabe BLOCK-CONTROL=*IGNORE das Blockformat wie folgt gesetzt:

Eigenschaften der Originaldatei		Angabe im Operanden BLOCK-CONTROL			
		(Format BS2000)		(Format WINZIP-COMPATIBLE)	
FCB-Type	BLK-CTRL	*KEEP	*IGNORE	*KEEP	*IGNORE
PAM	PAMKEY	nicht extrahiert	NO	nicht extrahiert	NO
PAM	DATA	wie Original	wie Original	wie Original	wie Original
PAM	NO	wie Original	wie Original	wie Original	wie Original
SAM	PAMKEY	nicht extrahiert	NO	nicht extrahiert	NO
SAM	DATA	wie Original	wie Original	wie Original	wie Original
ISAM	PAMKEY	nicht extrahiert	DATA4K	nicht extrahiert	DATA4K
ISAM	DATA	DATA4K ¹	DATA4K	Original	Original

¹ Nur ISAM-Dateien mit BLOCK-CONTROL-INFO=DATA4K und BUF-LEN=*STD(n) mit n=4, 8, 12 oder 16 können extrahiert werden.

Extrahieren auf NK4-Platten, Datei mit ungeradem Blockungsfaktor

Dateien mit ungeradem Blockungsfaktor können nicht auf NK4-Platten extrahiert werden.

2.2 Dateien übertragen

Im BS2000 sind ZIP-Archive grundsätzlich PAM-Dateien. Sie können übertragen werden in andere BS2000-Systeme oder zu Systemen mit anderen Betriebssystemen (MS/Windows, Unix, Linux, zOs). In Nicht-BS2000-Systemen können nur ZIP-Archive im WinZip-kompatiblen Format verwendet werden.

Mit ftp und openFT ab V11.0 kann ein ZIP-Archiv im Dateiformat PAM in ein anderes System übertragen werden. Für die Übertragung muss der binäre Modus eingestellt sein.

Wenn eine Datei mit openFT in oder von einem Nicht-BS2000-System übertragen werden soll, akzeptiert openFT < V11.0 nur SAM-Dateien mit RECORD-FORMAT=U. Mit der Anweisung //CONVERT-ZIP-CONTAINER kann ein ZIP-Archiv vom Dateiformat PAM in eine SAM-Datei mit RECORD-FORMAT=U konvertiert werden und umgekehrt. Die konvertierte Datei wird dabei als Kopie in der Ausgabedatei abgelegt.

Falls für diese Ausgabedateien kein Name angegeben wird, generiert BS2ZIP Standard-Dateinamen nach dem folgenden Schema:

<name>.SAM[.ZIP] bzw. <name>.PAM[.ZIP]

Dabei ist <name> ein Basisdateiname mit begrenzter Länge.

Beispiele

Name des ZIP-Archivs	Dateiformat des ZIP-Archivs	Name der Kopie
XYZ	PAM	XYZ.SAM
XYZ	SAM (mit RECORD-FORMAT=U)	XYZ.PAM
XYZ.PAM	PAM	XYZ.SAM
XYZ.SAM	SAM (mit RECORD-FORMAT=U)	XYZ.PAM
XYZ.ZIP	PAM	XYZ.SAM.ZIP
XYZ.ZIP	SAM (mit RECORD-FORMAT=U)	XYZ.PAM.ZIP
XYZ.PAM.ZIP	PAM	XYZ.SAM.ZIP
XYZ.SAM.ZIP	SAM (mit RECORD-FORMAT=U)	XYZ.PAM.ZIP

Hinweise

- Wenn der resultierende Dateiname einschließlich Katalog- und Benutzerkennung 54 Zeichen überschreitet, verkürzt BS2ZIP den Basisdateinamen (im Beispiel XYZ).
- Die konvertierte Datei wird unter der Benutzerkennung und der Default-Katalogkennung des Aufrufers angelegt, auch wenn die Originaldatei unter einer anderen Benutzer- oder Katalogkennung liegt.

2.3 Daten verschlüsseln

Bei der Aufnahme in ein ZIP-Archiv kann eine Datei verschlüsselt werden. Diese Verschlüsselung bietet für diese Datei im ZIP-Archiv einen gewissen Schutz gegen unberechtigten Zugriff. Wer Daten der Datei lesen will, muss die Datei beim Extrahieren wieder entschlüsseln und benötigt dazu den passenden Schlüssel (das Crypto-Kennwort). Das ZIP-Archiv selbst ist durch die Verschlüsselung nicht geschützt.

BS2ZIP unterstützt den Zip 2.0-Verschlüsselungsmechanismus und ist somit kompatibel zu WinZip und vielen anderen Zip-Tools. Die Zip 2.0-Verschlüsselung ist relativ schwach und bietet keinen ausreichenden Schutz gegen spezielle Kennwort Recovery Tools.

BS2ZIP verschlüsselt Dateien beim Hinzufügen zum Archiv bzw. entschlüsselt sie beim Extrahieren wieder, wenn die Verschlüsselung mit der Anweisung MODIFY-ZIP-OPTIONS eingeschaltet wurde. Beim Einschalten der Funktion muss auch das passende Crypto-Kennwort angegeben werden.

Enthält ein ZIP-Archiv Dateien, die mit verschiedenen Kennwörtern verschlüsselt wurden, muss die Verschlüsselung mit dem jeweils benötigten Kennwort vor dem Extrahieren wieder neu eingestellt werden.

Beim Extrahieren auf Nicht-BS2000-Systemen müsste in diesen Fällen WinZip bzw. das Zip-Tool für jedes Kennwort wieder neu gestartet werden.

2.4 Unterstützte Dateitypen

BS2ZIP kann SAM-, ISAM- und PAM-Dateien einschließlich PLAM-Bibliotheken verarbeiten.

ISAM-Dateien

ISAM-Dateien mit Sekundärschlüsseln werden nicht unterstützt.

Index und Daten der nach dem Extrahieren resultierenden ISAM-Datei werden nie getrennt. Die resultierende Datei kann nach der Extraktion größer als die Originaldatei sein.

Einer extrahierten ISAM-Datei wird immer der Standard-Blockfüllungsfaktor (PADDING-FACTOR) zugeordnet (DMS-Einschränkung).

PLAM-Bibliotheken

BS2ZIP verarbeitet PLAM-Bibliotheken als ganze Dateien.

Die separate Verarbeitung von Bibliothekselementen ist ebenfalls möglich. Die Dateiinformationen, die für das Element in der PLAM-Bibliothek hinterlegt sind (falls nicht vorhanden die PLAM-Default-Einstellungen), werden in das ZIP-Archiv übernommen und beim Extrahieren ausgewertet.

Temporäre Dateien

Mit BS2ZIP können temporäre Dateien hinzugefügt und extrahiert werden. Diese Dateien werden mit ihrem katalogisierten Dateinamen, d.h. inklusive Tempfile-Präfix aufgezeichnet.

Da der Tempfile-Präfix vom System taskspezifisch vergeben wird, können in einer Task keine Dateien mit einem anderen Tempfile-Präfix angelegt werden. Das Extrahieren einer Datei mit Tempfile-Präfix ohne Umbenennung ist nur möglich, wenn die Datei in derselben Task hinzugefügt wurde. Anderenfalls muss die Datei beim Extrahieren explizit umbenannt werden.

Lade-Module von K-Platten

Komprimierte Lade-Module von K-Platten können nicht auf einer NK-Platte extrahiert werden, auch wenn BLOCK-CONTROL-INFO=*IGNORE gesetzt ist, da das generierte NK-Lade-Modul nicht alle notwendigen Informationen enthält, um korrekt auf eine NK-Platte geladen zu werden.

Wenn der Operand BLOCK-CONTROL-INFO=*KEEP gesetzt ist, erhalten Sie folgende DMS-Fehlermeldung

```
DMS0D80 ALLOCATED DISK SPACE DOES NOT MATCH WITH THE REQUESTED FILE FORMAT.
```

2.5 Datei-Extents und benutzerspezifische Ressourcen

Es kann vorkommen, dass die maximale Anzahl an Extents des ZIP-Archivs erreicht wird (310). Wenn die Menge an zu komprimierenden Daten eine große Rolle spielt, dann wird empfohlen, die endgültige Größe des Archivs abzuschätzen, um o.g. Effekt zu vermeiden. Details und Beispiele siehe Abschnitt „Hinweise“ auf Seite 47.

Stellen Sie sicher, dass unter der Benutzerkennung, unter der das ZIP-Archiv eingerichtet wird, ausreichend Speicherplatz vorhanden ist. Ist nicht genug Speicherplatz vorhanden, kann das Archiv unzusammenhängend und damit unbrauchbar werden. Der Inhalt ist dann nicht mehr zugänglich.

Wenn Sie viele Dateien mit BS2ZIP verarbeiten wollen, dann sollte Ihrer Benutzerkennung ausreichend Benutzeradressraum zur Verfügung stehen (siehe ADDRESS-SPACE-LIMIT im Benutzereintrag).

2.6 Unterstützung der K2-Taste

Die K2-Taste hat einen begrenzten Funktionsumfang. Sie erlaubt das Unterbrechen des Programms im Anweisungsmodus (*//*). Nach der Unterbrechung mit K2 befindet sich der Benutzer im Kommandomodus (*/*).

Wenn die K2-Taste während der Ausführung einer ADD-FILE-, EXTRACT-FILE- oder SHOW-FILE-ATTRIBUTES-Anweisung gedrückt wird, unterbricht BS2ZIP die Verarbeitung bzw. Ausgabe mit der Fragemeldung SZP0208. Der Benutzer hat folgende Möglichkeiten:

- Er kann die aktuelle Verarbeitung einfach fortsetzen.
- Er kann die aktuelle Verarbeitung abbrechen. Damit kehrt er zurück in den Anweisungsmodus (*//*). Für die Dateien, die bis zum Unterbrechungszeitpunkt nicht verarbeitet worden sind, muss die Anweisung ggf. erneut aufgerufen werden.

2.7 Wiederherstellung im Fehlerfall

Eine abnormale Beendigung der Verarbeitung der ADD-FILE-Anweisung (z. B. bei Systemabsturz oder Stromausfall) kann dazu führen, dass ein Zip-Archiv inkonsistent ist. Diese Inkonsistenz wird dadurch verursacht, dass das Inhaltsverzeichnis des Archivs überschrieben wurde.

Die Wiederherstellung solcher inkonsistenter Zip-Archive wird durch folgenden Mechanismus ermöglicht: Vor der Verarbeitung der ersten ADD-FILE-Anweisung werden eine Kopie des Archiv-Verzeichnisses und der Name des Archivs in einer Backup-Datei mit dem Namen BS2ZIP.YYYY-MM-DD.HHMMSS.BAK gesichert. Diese Datei wird bei normaler Beendigung von BS2ZIP wieder gelöscht.

Bei abnormaler Beendigung wird diese Datei nicht gelöscht. Beim nächsten Öffnen des inkonsistenten Archivs ermittelt das System automatisch die zugehörige Backup-Datei und stellt das Archiv wieder in dem Zustand her, den es vor dem Start der abgebrochenen BS2ZIP-Session hatte. Allerdings sind die Daten, die während dieser Session in das Archiv aufgenommen wurden, noch im Archiv enthalten, aber es kann nicht auf sie zugegriffen werden. Das wiederhergestellte Archiv ist daher größer als vor dem Abbruch. Die unbrauchbaren Daten können mit der Anweisung REORGANIZE-ZIP-CONTAINER gelöscht werden.

Bei Zip-Archiven, auf die von mehreren Benutzerkennungen zugegriffen wird, ist folgende Einschränkung zu beachten:

Die Backup-Datei wird immer unter der aktuellen Benutzerkennung abgelegt. Wenn das inkonsistente Zip-Archiv unter einer anderen Benutzerkennung geöffnet wird, kann die Backup-Datei nicht automatisch gefunden werden. Das Öffnen schlägt daher fehl. In diesem Fall muss die Systemverwaltung die Benutzerkennung ermitteln, unter der die Backup-Datei abgelegt ist. Das Archiv muss dann unter dieser Kennung geöffnet und wiederhergestellt werden.

3 Kommandos und Anweisungen

3.1 Starten und Beenden von BS2ZIP

Die Anwendung BS2ZIP hat eine SDF-Schnittstelle. Bei Eingabe eines Fragezeichens werden alle auf der betreffenden Ebene möglichen Eingaben auf dem Bildschirm angezeigt. Die Schnittstelle SDF wird im Handbuch „Dialogschnittstelle SDF“ [1] beschrieben. Die SDF-Syntaxbeschreibung finden Sie im Handbuch „Kommandos“ [2].

3.1.1 Kommando zum Starten von BS2ZIP

Die Anwendung BS2ZIP wird mit dem Kommando /START-ZIP-MANAGER gestartet:

START-ZIP-MANAGER	Kurzform: ZIP-MANAGER
VERSION = <u>*STD</u> / <product-version> ,MONJV = <u>*NONE</u> / <filename 1..54 without-gen-vers> ,CPU-LIMIT = <u>*JOB-REST</u> / <integer 1..32767 seconds>	

VERSION = *STD / <product-version>

Angabe der BS2ZIP-Version, die verwendet werden soll.

Voreingestellt ist *STD: Die mit dem Kommando SELECT-PRODUCT-VERSION eingestellte Version wird geladen. Wurde keine Version eingestellt, so wird die höchste verfügbare Version geladen.

MONJV =

Angabe einer Monitor-Jobvariablen zur Überwachung des Programmablaufs.

MONJV = *NONE

Es wird keine Monitor-Jobvariable verwendet.

MONJV = <filename 1..54 without-gen-vers>

Angabe der Monitor-Jobvariablen, die den Programmablaufs überwachen soll. Diese Angabe steht nur mit dem Softwareprodukt JV zur Verfügung (siehe hierzu auch „Programmüberwachung“ im Handbuch „Jobvariablen“ [3]).

CPU-LIMIT = *JOB-REST / <integer 1..32767 seconds>

Maximale CPU-Zeit in Sekunden, die das Programm beim Ablauf verbrauchen darf. Überschreitet das Programm beim Ablauf die angegebene Zeit, wird im Dialog das Programm unterbrochen und die Meldung EXC0075 ausgegeben. Der Benutzer kann einen Dump anfordern, das Programm abbrechen oder fortsetzen. Im Stapelbetrieb wird das Programm beendet.

Voreingestellt ist *JOB-REST: Es soll maximal die verbleibende CPU-Zeit der Task verwendet werden. Nähere Informationen zur „Zeitlimitierung im BS2000“ finden Sie im Handbuch „Kommandos“ [2].

3.1.2 Beenden von BS2ZIP

Die Anwendung BS2ZIP wird mit der Anweisung END beendet.

3.1.3 Meldungen

Informationen über den Programmablauf von BS2ZIP, die Verarbeitung der Anweisungen sowie aufgetretene Fehler erhalten Sie über Meldungen aus der Meldungsklasse SZP (Meldungsschlüssel SZPxxxx).

Mit dem Kommando bzw. der Standardanweisung HELP-MSG-INFORMATION können Sie weitere Informationen über die Bedeutung abfragen.

Alle Meldungen von BS2ZIP finden Sie auch über die HTML-Anwendung „Systemmeldungen“ auf dem Manual-Server (URL: <http://manuals.ts.fujitsu.com>) und auf der DVD „BS2000 SoftBooks“.

3.2 Beschreibung der Anweisungen

Die Anwendung BS2ZIP bietet die folgende Funktionen:

Anweisungen	Bedeutung
ADD-FILE	Fügt eine oder mehrere BS2000-Dateien dem aktuellen ZIP-Archiv hinzu.
CONVERT-ZIP-CONTAINER	Konvertiert ein ZIP-Archiv vom Datenformat PAM nach SAM mit RECFORM=U und umgekehrt (für den Datentransfer mit openFT < V11.0).
DELETE-FILE	Löscht eine oder mehrere Dateien aus dem aktuellen ZIP-Archiv.
END	Schließt die aktuelle ZIP-Datei und beendet das Programm.
EXTRACT-FILE	Extrahiert eine oder mehrere Dateien aus dem aktuellen ZIP-Archiv.
MODIFY-ZIP-OPTIONS	Vereinbart Voreinstellungen für die BS2ZIP-Verarbeitung, z.B. zur Verschlüsselung
OPEN-ZIP-CONTAINER	Öffnet ein ZIP-Archiv. Wenn das Archiv noch nicht existiert, wird es ggf. neu erzeugt.
REORGANIZE-ZIP-CONTAINER	Reorganisiert ein ZIP-Archiv.
SHOW-FILE-ATTRIBUTES	Listet den Inhalt des aktuell geöffneten ZIP-Archivs auf.
START-TRACE	Aktiviert einen Trace, der in eine Datei geschrieben wird.
STOP-TRACE	Beendet den Trace.

Neben diesen Anweisungen stehen Ihnen auch die SDF-Standardanweisungen zur Verfügung. Diese werden Ihnen in der Anweisungsliste, die Sie nach Eingabe eines Fragezeichens erhalten, mit angezeigt. Die SDF-Standardanweisungen sind im Handbuch „Dialogschnittstelle SDF“ [1] ausführlich beschrieben.

3.2.1 ADD-FILE

Mit dieser Anweisung fügen Sie eine oder mehrere Dateien dem aktuell geöffneten ZIP-Archiv hinzu. Wenn das ZIP-Archiv im Lesemodus geöffnet ist, wird die Anweisung abgewiesen.

Es werden SAM-, ISAM- und PAM-Dateien einschließlich PLAM-Bibliotheken unterstützt. Wahlweise können auch Elemente einer PLAM-Bibliothek dem ZIP-Archiv hinzugefügt werden.

ADD-FILE

```

FROM-FILE = *ALL / *FROM-FILE(...) / *FROM-LIBRARY-ELEMENT(...) / *LIBRARY-ELEMENT(...) /
list-poss(20): <filename 1..54 without-gen-vers with-wild (80)>

*FROM-FILE(...)
| LIST-FILE-NAME = <filename 1..54 without-gen-vers>

*FROM-LIBRARY-ELEMENT(...)
| LIBRARY = <filename 1..54 without-gen-vers>
| ELEMENT = <composed-name 1..64 with-under>

*LIBRARY-ELEMENT(...)
| LIBRARY = <filename 1..54 without-vers>
| ELEMENT = <composed-name 1..64 with-under with-wild(132)>(…)
| <composed-name 1..64 with-under with-wild(132)>(…)
| | VERSION = *HIGHEST-EXISTING / *UPPER-LIMIT /
| | <composed-name 1..24 with-under with-wild(40)>
| | BASE = *STD / <composed-name 1..24 with-under with-wild(40)>
| TYPE = <alphanum-name 1..8 with-wild(12)>

,TO-FILE = *BY-SOURCE / <composed-name 1..98 with-underscore with-wild-constr(132)>(…)
| <composed-name 1..98 with-underscore with-wild-constr(132)>(…)
| WITH-VERSION = *STD / *YES / *NO
| WITH-TYPE = *STD / *YES / *NO

,CHARACTER-CONVERSION = *BY-CONTAINER-FORMAT / *NO / *TO-EBCDIC / *TO-WIN-ANSI

,COMPRESSION-LEVEL = *STD / *NONE / *BEST-SPEED / *BEST-COMPRESSION

,DATA-TYPE = *NOT-SPECIFIED / *CHARACTER / *BINARY

,EXCEPT-FILE-NAME = *NONE / *FROM-FILE(...) / *FROM-LIBRARY-ELEMENT(...) /
list-poss(20): <filename 1..54 without-vers with-wild (80)>

*FROM-FILE(...)
| LIST-FILE-NAME = <filename 1..54 without-gen-vers>

*FROM-LIBRARY-ELEMENT(...)
| LIBRARY = <filename 1..54 without-gen-vers>
| ELEMENT = <composed-name 1..64 with-under>

,DELETE-SOURCE = *NO / *YES

,LOGGING = *MINIMUM / *MAXIMUM

```

FROM-FILE =

Gibt an, welche Dateien bzw. welche Bibliothekselemente zum ZIP-Archiv hinzugefügt werden sollen. Die Auswahl lässt sich durch den Operanden EXCEPT-FILE-NAME noch einschränken.

FROM-FILE = *ALL

Es werden alle unterstützten Dateien der aktuellen Benutzerkennung zum ZIP-Archiv hinzugefügt.

FROM-FILE = *FROM-FILE(...)

Die Pfadnamen der Dateien, die zum ZIP-Archiv hinzugefügt werden sollen, werden einer Datei entnommen. Der nicht-privilegierte Aufrufer muss Eigentümer oder Miteigentümer dieser Datei sein. Die Datei muss eine SAM-Datei mit variabler Satzlänge sein, die pro Satz einen Pfadnamen enthält. Kleinbuchstaben dürfen verwendet werden, sie werden jedoch in Großbuchstaben umgewandelt. Sie müssen die Dateinamen vollqualifiziert mit oder ohne Katalog- oder Benutzerkennung angeben.

Die Datei können Sie z.B. mit dem DVS-Kommando SHOW-FILE-ATTRIBUTES erstellen.

LIST-FILE-NAME = <filename 1..54 without-gen-vers>

Pfadname der Datei, die die Liste mit Dateinamen enthält.

FROM-FILE = *FROM-LIBRARY-ELEMENT(...)

Die Pfadnamen der Dateien, die zum ZIP-Archiv hinzugefügt werden sollen, werden einem PLAM-Bibliothekselement (Typ S) entnommen. Das Bibliothekselement besteht aus Sätzen variabler Länge und enthält pro Satz einen Pfadnamen. Kleinbuchstaben dürfen verwendet werden, sie werden jedoch in Großbuchstaben umgewandelt. Sie müssen die Dateinamen vollqualifiziert mit oder ohne Katalog- oder Benutzerkennung angeben.

LIBRARY = <filename 1..54 without-gen-vers>

Name der PLAM-Bibliothek.

ELEMENT = <composed-name 1..64 with-under>

Name des Elements vom Typ S, das die Liste mit Dateinamen enthält. Das Element mit der höchsten existierenden Version wird verwendet.

FROM-FILE = *LIBRARY-ELEMENT(...)

Elemente einer PLAM-Bibliothek sollen zum ZIP-Archiv hinzugefügt werden. Ein Element ist vollständig definiert durch seinen Namen, seinen Typ und die Versionsnummer.

LIBRARY = <filename 1..54 without-vers>

Name der PLAM-Bibliothek, aus der Elemente hinzugefügt werden sollen.

ELEMENT = <composed-name 1..64 with-under with-wild(132)>(…)

Das angegebene Element wird zum ZIP-Archiv hinzugefügt, falls der Elementtyp unterstützt wird. Mit Musterzeichen können mehrere Elemente angegeben werden.

VERSION = *HIGHEST-EXISTING / *UPPER-LIMIT / <composed-name 1..24 with-under with-wild(40)>

Version des Elements, das ausgegeben werden soll. Voreingestellt ist *HIGHEST-EXISTING, d.h. das letzte Element in alphabetischer Reihenfolge.

*UPPER-LIMIT bezeichnet die höchst mögliche Version 'X'FF' (angezeigt mit dem Zeichen '@').

Wird die Version mit Wildcards angegeben und existieren gleichnamige Bibliothekselemente in Versionen, die durch die Wildcard-Angabe betroffen sind, werden alle diese Bibliothekselemente ausgegeben.

BASE = *STD / <composed-name 1..24 with-under with-wild(40)>

Angabe der Basis, falls mehrere Delta-Elemente zu der angegebenen Version existieren.

TYPE = <alphanum-name 1..8 with-wild (12)>

Typ des Bibliothekselements. Unterstützt werden die Typen D, J, M, P, S und X, sowie die darauf basierenden benutzer-definierten Typen.

Der Name besteht bei Angabe des Typs mit Wildcards aus maximal 12 alphanumerischen Zeichen.

FROM-FILE = list-poss(20): <filename 1..54 without-vers with-wild(80)>

Die Pfadnamen der Dateien, die zum ZIP-Archiv hinzugefügt werden sollen, werden direkt eingegeben. Sie können maximal 20 Namen in einer Liste eingeben.

Sie können die Dateinamen voll- oder teilqualifiziert, mit oder ohne Katalog- oder Benutzerkennung angeben. Der Dateiname wird gegebenenfalls mit der Benutzerkennung des Auftrags und der Standard-Katalogkennung ergänzt.

Sie können die Dateien auch unter Verwendung der Wildcard-Syntax auswählen. In diesem Fall werden alle unterstützten Dateien, die der Musterzeichenfolge entsprechen, zum ZIP-Archiv hinzugefügt.

TO-FILE = *BY-SOURCE

Legt fest, dass die eingegebenen Dateien mit ihrem aktuellen Namen ohne Katalog-/Benutzerkennung abgelegt werden.

TO-FILE = <composed-name 1..98 with-underscore with-wild-constr(132)>(…)

Die Datei wird mit dem angegebenen Namen im Archiv abgelegt.

Wenn im Operanden FROM-FILE Wildcard-Syntax verwendet wurde, kann mit Musterzeichen in einer Konstruktionszeichenfolge angegeben werden, wie die neuen Namen im Archiv gebildet werden sollen (siehe SDF-Regeln für Konstruktionsangaben, Handbuch „Kommandos“ [2]).

Wenn im Operanden FROM-FILE eine Liste der Dateinamen verwendet wurde, wird nur TO-FILES = *BY-SOURCE akzeptiert oder ein Wert ohne Wildcards, der jedoch mit einem Punkt abgeschlossen wurde. In diesem Fall wird der TO-FILE-Wert als Präfix verwendet.

Beispiel:

ADD-FILE FROM-FILE=MYFILE,TO-FILE=XXX. wird akzeptiert.
Alle in MYFILE enthaltenen Dateinamen erhalten das Präfix XXX.

Für Bibliothekselemente ist voreingestellt, dass der angegebene Name zusätzlich einen Suffix mit dem Typ und der Version des Elements erhält (Format: <to-file>.<type>.<version>). In den folgenden Operanden lässt sich die Erstellung dieses Suffix optional steuern:

WITH-VERSION = *STD / *YES / *NO

Wird nur für Bibliothekselemente ausgewertet:

Gibt an, ob der Suffix die Elementversion enthalten soll.

Mit *STD ist *YES voreingestellt.



Ein Element mit VERSION=*UPPER-LIMIT erhält anstelle des Zeichens '@' ein 'U' im Suffix (z.B. TO-FILE=myelem1 wird bei Typ S zu MYELEM1.U.S).

WITH-TYPE = *STD / *YES / *NO

Wird nur für Bibliothekselemente ausgewertet:

Gibt an, ob der Suffix den Elementtyp enthalten soll.

Mit *STD ist *YES voreingestellt.

CHARACTER-CONVERSION = *BY-CONTAINER-FORMAT

Die Eingabedateien werden gemäß dem Format der ZIP-Datei konvertiert.

CHARACTER-CONVERSION = *NO

Die Eingabedateien werden nicht konvertiert.

CHARACTER-CONVERSION = *TO-EBCDIC

Die ASCII-kodierten Eingabedateien werden vor der Komprimierung in EBCDIC konvertiert. Es werden nur SAM/ISAM-Dateien konvertiert.

CHARACTER-CONVERSION = *TO-WIN-ANSI

Die EBCDIC-kodierten Eingabedateien werden vor der Komprimierung in WIN-ANSI konvertiert. Es werden nur SAM/ISAM-Dateien konvertiert.

COMPRESSION-LEVEL =

Bestimmt den Komprimierungsmodus, der das Verhältnis von Komprimierungsrate und Geschwindigkeit bestimmt.

COMPRESSION-LEVEL = *STD

Dieser Komprimierungsmodus wählt einen Kompromiss zwischen Geschwindigkeit und Komprimierungsrate.

COMPRESSION-LEVEL = *BEST-SPEED

Bei diesem Komprimierungsmodus wird die schnellste Komprimierung gewählt. Das Archiv wird dadurch größer.

COMPRESSION-LEVEL = *BEST-COMPRESSION

Bei diesem Komprimierungsmodus wird die beste Komprimierung gewählt. Das Archiv wird dadurch kleiner, die Verarbeitung dauert jedoch länger.

COMPRESSION-LEVEL = *NONE

Die Datei wird hinzugefügt, ohne komprimiert zu werden.

DATA-TYPE =

Der Operand DATA-TYPE steuert die Einlagerung der Datei. Diese Option ist nur relevant für SAM-Dateien, die in ein WinZip-kompatibles Archiv aufgenommen werden.

Operand DATA-TYPE=	Archivierungs-Format	
	WINZIP-COMPATIBLE	BS2000
<u>*NOT-SPECIFIED</u>	<ul style="list-style-type: none"> – SAM/ISAM-Dateien: werden als Text-Dateien übernommen. Ein Datensatz = eine Textzeile – PAM-Dateien: werden als binäre Dateien übernommen. 	<ul style="list-style-type: none"> – SAM/ISAM-Dateien: werden als Text-Dateien übernommen. Ein Datensatz = eine Textzeile – PAM-Dateien: werden als binäre Dateien übernommen.
*CHARACTER	Nicht relevant. Bearbeitet als DATA-TYPE=*NOT-SPECIFIED	Nicht relevant. Bearbeitet als DATA-TYPE=*NOT-SPECIFIED
*BINARY	<ul style="list-style-type: none"> – SAM-Dateien: werden als binäre Dateien übernommen. Die Datei kann nur als PAM-Datei extrahiert werden. – ISAM-Dateien: Nicht relevant. Werden als Text-Dateien übernommen. Ein Datensatz = eine Textzeile – PAM-Dateien: werden als binäre Dateien verarbeitet. 	Nicht relevant. Bearbeitet als DATA-TYPE=*NOT-SPECIFIED

EXCEPT-FILE-NAME =

Sie können Dateien angeben, die nicht zum ZIP-Archiv hinzugefügt werden sollen.

EXCEPT-FILE-NAME = *NONE

Es werden alle Dateien zum ZIP-Archiv hinzugefügt, die beim Operanden FROM-FILE angegeben sind.

EXCEPT-FILE-NAME = *FROM-FILE(...)

Die Pfadnamen der Dateien, die nicht zum ZIP-Archiv hinzugefügt werden sollen, werden einer Datei entnommen. Der nicht-privilegierte Aufrufer muss Eigentümer oder Miteigentümer dieser Datei sein. Die Datei muss eine SAM-Datei mit variabler Satzlänge sein, die pro Satz einen Pfadnamen enthält. Kleinbuchstaben dürfen verwendet werden, sie werden jedoch in Großbuchstaben umgewandelt. Sie müssen die Dateinamen vollqualifiziert mit oder ohne Katalog- oder Benutzerkennung angeben.

Sie können die Datei z.B. mit dem DVS-Kommando SHOW-FILE-ATTRIBUTES erstellen.

LIST-FILE-NAME = <filename 1..54 without-gen-vers>

Pfadname der Datei, die die Liste mit Dateinamen enthält.

EXCEPT-FILE-NAME = *FROM-LIBRARY-ELEMENT(...)

Die Pfadnamen der Dateien, die nicht zum ZIP-Archiv hinzugefügt werden sollen, werden einem PLAM-Bibliothekselement (Typ S) entnommen. Das Bibliothekselement besteht aus Sätzen variabler Länge und enthält pro Satz einen Pfadnamen. Kleinbuchstaben dürfen verwendet werden, sie werden jedoch in Großbuchstaben umgewandelt. Sie müssen die Dateinamen vollqualifiziert mit oder ohne Katalog- oder Benutzerkennung angeben.

LIBRARY = <filename 1..54 without-gen-vers>

Name der PLAM-Bibliothek.

ELEMENT = <composed-name 1..64 with-under>

Name des Elements vom Typ S, das die Liste mit Dateinamen enthält. Das Element mit der höchsten existierenden Version wird verwendet.

EXCEPT-FILE-NAME = list-poss(20): <filename 1..54 without-vers with-wild(80)>

Die Pfadnamen der Dateien, die nicht zum ZIP-Archiv hinzugefügt werden sollen, werden direkt eingegeben. Sie können maximal 20 Namen in einer Liste eingeben.

Die Dateinamen dürfen nicht mit einem Bindestrich beginnen. Sie können die Dateinamen voll- oder teilqualifiziert, mit oder ohne Katalog- oder Benutzerkennung angeben. Der Dateiname wird gegebenenfalls mit der Benutzerkennung des Auftrags und der Standard-Katalogkennung ergänzt.

Sie können die Dateien auch unter Verwendung der Wildcard-Syntax auswählen.

DELETE-SOURCE = *NO / *YES

Gibt an, ob die ursprünglichen Dateien bzw. Bibliothekselemente, nachdem sie dem ZIP-Archiv hinzugefügt wurden, gelöscht werden sollen.

DELETE-SOURCE = *NO

Die ursprünglichen Dateien bzw. Bibliothekselemente bleiben erhalten.

DELETE-SOURCE = *YES

Die ursprünglichen Dateien bzw. Bibliothekselemente werden gelöscht. Wenn das Löschen wegen der Schutzattribute nicht möglich ist, wird eine Meldung ausgegeben.



- Wenn die ursprüngliche Datei schreibgeschützt ist, wird die Datei zum ZIP-Archiv hinzugefügt und eine Fehlermeldung an den Benutzer ausgegeben.
- Wenn das ursprüngliche Bibliothekselement schreibgeschützt ist, wird das Bibliothekselement zum ZIP-Archiv hinzugefügt und eine Fehlermeldung an den Benutzer ausgegeben.
- Wenn die ursprüngliche Bibliothek schreibgeschützt ist, wird das Bibliothekselement zum ZIP-Archiv hinzugefügt und eine Fehlermeldung an den Benutzer ausgegeben.

Hinweise

- Komprimierungsmodi (Beispiel)

Komprimierungsmodus	Datenmenge komprimiert (ohne Komprimierung 280630 PP)	Komprimierungsrate	CPU-Zeit
*BEST-SPEED	54096 PP	80,7%	448,5 s
*STD	44496 PP	84,2%	970,2 s
*BEST-COMPRESSION	43488 PP	84,5%	8229,6 s

- Wenn während der Ausführung der Anweisung ADD-FILE die K2-Taste gedrückt wird, wird die Verarbeitung mit der Fragemeldung SZP0208 unterbrochen:
 - Der Benutzer kann die Verarbeitung einfach fortsetzen.
 - Der Benutzer kann die Verarbeitung beenden und kehrt zurück in den Anweisungsmodus (/). Die Dateien, die bis zum Unterbrechungszeitpunkt nicht hinzugefügt sind, werden nicht mehr hinzugefügt. Sie müssen ggf. erneut hinzugefügt werden.
- Mit der Voreinstellung TO-FILE=*BY-SOURCE werden Dateinamen im ZIP-Archiv immer ohne Katalog-/Benutzerkennung abgelegt. Katalog-/Benutzerkennung werden nur übernommen, wenn diese im Operanden TO-FILE im Dateinamen mit angegeben sind.

Angabe bei ADD-FILE	Registriert als
FROM-FILE = MYFILE	MYFILE
FROM-FILE = \$UID.MYFILE	MYFILE
FROM-FILE = :CAT1:\$UID.MYFILE	MYFILE
FROM-FILE = MYFILE, TO-FILE=\$UID.MYFILE	\$UID.MYFILE
FROM-FILE = MY*, TO-FILE=NEW-*	NEW-FILE
FROM-FILE = #MYFILE.1	S.163.TSN1.MYFILE.1
FROM-FILE = #MYFILE.1, TO-FILE=MYFILE	MYFILE
FROM-FILE = #MYFILE.*, TO-FILE=MYFILE-*	MYFILE-1

- Wenn mit der Anweisung **MODIFY-ZIP-OPTIONS** Datenverschlüsselung eingestellt wurde, werden die Dateien beim Hinzufügen verschlüsselt. Die dabei verwendete Standard-Zip-2.0-Verschlüsselung ist kompatibel zu WinZip auf Windows-basierten Systemen.

LOGGING = *MINIMUM / *MAXIMUM

Legt den Umfang der Meldungs Ausgabe fest.

LOGGING = *MINIMUM

Nur Fehlermeldungen werden ausgegeben

LOGGING =*MAXIMUM

Alle Meldungen werden ausgegeben. Derzeit werden nach dem Hinzufügen der Datei bzw. des Bibliothekselements die (garantierten) Meldungen SZP0116 und SZP0117 ausgegeben; in Zukunft werden gegebenenfalls weitere Meldungen ausgegeben.

3.2.2 CONVERT-ZIP-CONTAINER

Ein ZIP-Archiv besitzt im Standardfall das Dateiformat PAM.

Alternativ kann ein ZIP-Archiv auch als binäre SAM-Datei (mit RECORD-FORMAT=U) vorliegen. Dieses Format tritt in folgenden Fällen auf (siehe auch [Abschnitt „Interoperabilität mit Windows“ auf Seite 63](#)):

- Wenn ein ZIP-Archiv mit openFT < V11.0 in das BS2000 übertragen wurde, wurde es mit diesem Format angelegt. Für die weitere Bearbeitung muss das ZIP-Archiv im Dateiformat PAM vorliegen.
- Um ein ZIP-Archiv mit openFT < V11.0 übertragen zu können, musste es in diesem Format vorliegen.

Die Anweisung CONVERT-ZIP-CONTAINER konvertiert ein ZIP-Archiv in das jeweils andere Dateiformat (von PAM nach SAM mit RECORD-FORMAT=U und umgekehrt). Das ZIP-Archiv bleibt unverändert bestehen und das konvertierte ZIP-Archiv wird in der angegebenen Ausgabedatei abgelegt.

CONVERT-ZIP-CONTAINER

```

FROM-FILE = *CURRENT / <filename 1..54 without-gen-vers> / *LINK(...)
    *LINK(...)
      | LINK-NAME = <structured-name 1..8>
,TO-FILE = *STD / <filename 1..54 without-gen-vers> / *LINK(...)
    *LINK(...)
      | LINK-NAME = <structured-name 1..8>
,WRITE-MODE = *CREATE / *REPLACE-ONLY / *ANY

```

FROM-FILE = *CURRENT / <filename 1..54 without-gen-vers> / *LINK(...)

Bestimmt das ZIP-Archiv, das konvertiert werden soll.

FROM-FILE = *CURRENT

Es soll das aktuell geöffnete ZIP-Archiv konvertiert werden.

FROM-FILE = <filename 1..54 without-gen-vers>

Konvertiert das angegebene ZIP-Archiv.

FROM-FILE = *LINK(...)

Das zu konvertierende ZIP-Archiv wird über einen Linknamen angegeben.

LINK-NAME = <structured-name 1..8>

Linkname, der dem ZIP-Archiv zugewiesen ist.

TO-FILE = *STD / <filename 1..54 without-gen-vers> / *LINK(...)

Bestimmt die Ausgabedatei, in der das konvertierte ZIP-Archiv abgelegt wird.

TO-FILE = *STD

Die Ausgabedatei erhält einen Standardnamen, der den Dateinamen des zu konvertierenden ZIP-Archivs und einen Suffix für das vorliegende Dateiformat (SAM bzw. PAM) enthält. Die Ausgabedatei wird unter der Benutzerkennung und der Default-Katalogkennung des Aufrufers angelegt. Eine im Operanden FROM-FILE angegebene Benutzer- und/oder Katalogkennung wird für die Ausgabedatei ignoriert.

Der Standardname wird nach folgenden Regeln gebildet:

- Das ZIP-Archiv liegt im Dateiformat PAM vor.

Name des ZIP-Archivs	Name der Ausgabedatei
teilname.PAM	teilname.SAM
teilname.PAM.ZIP	teilname.SAM.ZIP
teilname.ZIP	teilname.SAM.ZIP
dateiname	dateiname.SAM

- Das ZIP-Archiv liegt im Dateiformat SAM mit RECORD-FORMAT=U vor.

Name des ZIP-Archivs	Name der Ausgabedatei
teilname.SAM	teilname.PAM
teilname.SAM.ZIP	teilname.PAM.ZIP
teilname.ZIP	teilname.PAM.ZIP
dateiname	dateiname.PAM

Bei einem ZIP-Archiv *teilname.ZIP* bzw. *dateiname* kann für die Ausgabedatei durch das Anhängen des Suffix ein Pfadname entstehen, der länger als 54 Zeichen ist. In diesem Fall wird *teilname* bzw. *dateiname* vor dem Suffix um eine entsprechende Anzahl von Zeichen gekürzt.

TO-FILE = <filename 1..54 without-gen-vers>

Das konvertierte ZIP-Archiv wird in der angegebenen Datei abgelegt.

TO-FILE = *LINK(...)

Die Ausgabedatei wird über einen Linknamen angegeben.

LINK-NAME = <structured-name 1..8>

Linkname, der der Ausgabedatei zugewiesen ist.

WRITE-MODE =

Gibt an, ob die Ausgabedatei neu angelegt oder nur überschrieben werden soll.

WRITE-MODE = *CREATE

Die Ausgabedatei wird neu angelegt. Wenn sie bereits existiert, wird die Anweisung abgewiesen.

WRITE-MODE = *REPLACE-ONLY

Die Ausgabedatei existiert und wird überschrieben. Wenn sie nicht existiert, wird die Anweisung abgewiesen.

WRITE-MODE = *ANY

Existiert die Ausgabedatei noch nicht, wird sie neu angelegt. Anderenfalls wird sie nur überschrieben.

3.2.3 DELETE-FILE

Mit dieser Anweisung löschen Sie eine oder mehrere Dateien aus dem aktuell geöffneten ZIP-Archiv. Wenn das ZIP-Archiv im Lesemodus geöffnet ist, wird die Anweisung abgewiesen.

Die Größe des ZIP-Archivs ändert sich durch Löschen darin enthaltener Dateien nicht. Nicht mehr benötigter Speicherplatz wird erst beim Reorganisieren des ZIP-Archivs freigegeben (siehe Anweisung [REORGANIZE-ZIP-CONTAINER](#)).

DELETE-FILE
FILE-NAME = *ALL / <composed-name 1..98 with-under with-wild(132)> / <c-string 1..1024 with-low>

FILE-NAME = *ALL

Es werden alle Dateien aus dem ZIP-Archiv gelöscht.

FILE-NAME = <composed-name 1..98 with-under with-wild(132)>

Die angegebene Datei wird aus dem ZIP-Archiv gelöscht. Bei Verwendung von Musterzeichen werden alle Dateien, die der Musterzeichenfolge entsprechen, aus dem ZIP-Archiv gelöscht.

FILE-NAME = <c-string 1..1024 with-low>

Alle Dateien, die der angegebenen Zeichenfolge entsprechen (Musterzeichen entsprechend den SDF-Regeln für Auswahlangaben sind erlaubt (siehe SDF-Metasyntax, Handbuch „Kommandos“ [2]), werden aus dem ZIP-Archiv gelöscht. Die Angabe als C-String ist zu verwenden, wenn das Archiv nicht in einem BS2000-System erstellt wurde und die betroffenen Dateinamen nicht BS2000-konform sind (z.B. Groß-/Kleinschreibung).

3.2.4 END

Diese Anweisung schließt die aktuell geöffnete ZIP-Datei und beendet das Programm.

END

3.2.5 EXTRACT-FILE

Mit dieser Anweisung extrahieren Sie eine oder mehrere Dateien aus dem aktuell geöffneten ZIP-Archiv.

EXTRACT-FILE

```

FILE-NAME = *ALL / <composed-name 1..98 with-under with-wild(132)> / <c-string 1..1024 with-low>
,TO-FILE = *BY-SOURCE / <filename 1..54 without-gen-vers with-wild-constr(80)>
,WRITE-MODE = *CREATE / *REPLACE-ONLY / *ANY
,DATA-TYPE = *NOT-SPECIFIED / *CHARACTER / *BINARY / *SAM-BINARY
,CHARACTER-CONVERSION = *BY-CONTAINER-FORMAT / *NO / *TO-WIN-ANSI / *TO-EBCDIC
,BLOCK-CONTROL-INFO = *KEEP / *IGNORE
,LOGGING = *MINIMUM / *MAXIMUM

```

FILE-NAME = *ALL

Alle im Archiv enthaltenen Dateien werden extrahiert.

FILE-NAME = <composed-name 1..98 with-under with-wild(132)>

Die angegebene Datei wird aus dem ZIP-Archiv extrahiert. Bei Verwendung von Musterzeichen werden alle Dateien, die der Musterzeichenfolge entsprechen, aus dem ZIP-Archiv extrahiert.

FILE-NAME = <c-string 1..1024 with-low>

Alle Dateien, die der angegebenen Zeichenfolge entsprechen (Musterzeichen entsprechend den SDF-Regeln für Auswahlangaben sind erlaubt (siehe SDF-Metasyntax, Handbuch „Kommandos“ [2]), werden aus dem ZIP-Archiv extrahiert. Die Angabe als C-String ist zu verwenden, wenn das Archiv nicht in einem BS2000-System erstellt wurde und die betroffenen Dateinamen nicht BS2000-konform sind (z.B. Groß-/Kleinschreibung).

TO-FILE = *BY-SOURCE / <filename 1..54 without-gen-vers with-wild-constr(80)>

Entsprechend der Herkunft der komprimierten Dateien gilt für den Ausgabenamen Folgendes:

- Bei BS2000-Dateien wird der Name der Ausgabedatei entsprechend den SDF-Regeln für Konstruktionsangaben gebildet (siehe SDF-Metasyntax, Handbuch „Kommandos“ [2]).

- Bei Nicht-BS2000-Dateien (PC, UNIX-System) wird der Name der Ausgabedatei so gebildet, dass das Zeichen * im Operanden TO-FILE durch den im Archiv enthaltenen Dateinamen ersetzt wird. Der mögliche Pfad der Datei wird bei Erstellung des Dateinamens ignoriert.

TO-FILE=	BS2000-Datei	Nicht-BS2000-Datei
<u>*BY-SOURCE</u>	Der Ausgabedateiname ist der Dateiname, wie er im Archiv registriert wurde. <ul style="list-style-type: none"> – wurde er mit Katalog- und Benutzerkennung eingetragen, wird er mit dieser Katalog- und Benutzerkennung extrahiert – wurde er ohne Katalog- und Benutzerkennung eingetragen, wird er unter der aktuellen Katalog- und Benutzerkennung extrahiert 	Der Ausgabedateiname ist der Dateiname ohne Zugriffspfad mit der aktuellen Katalog-/Benutzerkennung als Präfix. Dateinamen, die nicht der Syntax im BS2000 entsprechen, werden von BS2ZIP umbenannt (s.u).
<filename 1..54 without-gen-vers with-wild- constr(80)>	Gültiges Format ist: <ul style="list-style-type: none"> – Dateiname ohne Platzhalter – Dateiname mit Platzhalter: der Name der Ausgabedatei ist entsprechend den SDF-Regeln für Konstruktionsangaben gebildet. Entspricht der resultierende Dateiname nicht der Syntax im BS2000, wird die Entnahme der Datei abgelehnt. 	Gültiges Format ist: <ul style="list-style-type: none"> – ein Pfadname ohne Platzhalter – [<PREFIX>]*[<SUFFIX>]: Der Name der Ausgabedatei ist dann [<PREFIX>]filename[<SUFFIX>]. filename ist der eingetragene Dateiname ohne Zugriffspfad. Entspricht der resultierende Dateiname nicht der Syntax im BS2000, wird er von BS2ZIP umbenannt (s.u).

Wenn einige Namen der komprimierten Dateien nicht mit BS2000 kompatibel sind, muss man sie einzeln extrahieren und jeweils einen gültigen Ausgabenamen angeben. Wenn sich beim Extrahieren ein ungültiger Dateiname ergibt, erhält die die betroffene Datei ersatzweise den folgenden Dateinamen:

FILExxxx.yyyymmdd.hhmmss

Dabei ist xxxx eine fortlaufende Nummer und yyyymmdd.hhmmss steht für das aktuelle Datum und die aktuelle Uhrzeit. Die vorgenommene Umbenennung zeigt BS2ZIP mit der Meldung SZP0090 an. Beispiel:

```
% SZP0090 Warning. File name 'TEST_KDO_1.HTM' is not BS2000 compliant.
The file will be extracted under the name 'FILE0001.20111108.161442'
```

Die Katalog-/Benutzerkennung dieser Dateien wird vom <PREFIX> übernommen. Falls sie nicht angegeben sind, wird standardmäßig die Katalog- und Benutzerkennung des aktuellen Benutzers verwendet.

WRITE-MODE = *CREATE

Die Ausgabedatei wird neu erzeugt, sie darf nicht existieren. Die Anweisung wird zurückgewiesen, wenn die Datei schon existiert.

WRITE-MODE = *REPLACE-ONLY

Die Ausgabedatei muss existieren und wird ersetzt. Die Anweisung wird zurückgewiesen, wenn die Datei nicht existiert.

WRITE-MODE = *ANY

Die Ausgabedatei wird überschrieben, wenn sie schon existiert, bzw. wird erstellt, falls sie noch nicht existiert.

DATA-TYPE =

Der Operand DATA-TYPE steuert die Datensatzstruktur der zu extrahierenden Dateien.

DATA-TYPE	Archiv-Format WINZIP-COMPATIBLE	Archiv-Format BS2000
<u>*NOT-SPECIFIED</u>	Wenn im ZIP-Archiv keine Datei-Info gefunden wird, dann wird *CHARACTER angenommen, andernfalls wird die Datei-Info verwendet. Die Konvertierung erfolgt gemäß Dateityp und Angabe im Operanden CHARACTER-CONVERSION.	Wenn keine Datei-Info gefunden wird, dann führt dies zu einem Fehler, andernfalls wird die Datei-Info verwendet - keine Konvertierung
*CHARACTER	Die Datei wird als SAM-Datei extrahiert. Es wird davon ausgegangen, dass die Sätze durch ODOA getrennt sind. Die Konvertierung wird gemäß der Angabe im Operanden CHARACTER-CONVERSION durchgeführt.	Die Anweisung wird zurückgewiesen.
*BINARY	Die Datei wird als PAM-Datei extrahiert. Es wird keine Konvertierung durchgeführt. Im Fall von Original-PAM-Dateien wird die Anweisung mit der Fehlermeldung SZP0121 zurückgewiesen.	Die Anweisung wird zurückgewiesen.

DATA-TYPE	Archiv-Format WINZIP-COMPATIBLE	Archiv-Format BS2000
*SAM-BINARY	Die Datei wird als binäre SAM-Datei (REC-FORM=U) extrahiert. Es wird keine Konvertierung durchgeführt.	Die Anweisung wird zurückgewiesen.

CHARACTER-CONVERSION =

Dieser Operand steuert die WIN-ANSI/EBCDIC-Konvertierung. Die Konvertierung wird nur unterstützt für WinZip-kompatible Archive und für folgende Dateien:

- Datei extrahiert mit DATA-TYPE=*CHARACTER
- Datei extrahiert mit DATA-TYPE=*NOT-SPECIFIED und die Datei ist in der Datei-Info nicht als PAM-Datei ausgewiesen.

CHARACTER-CONVERSION	Konvertierungsverhalten
*BY-CONTAINER-FORMAT	Standardwert. Das Verhalten ist abhängig vom Format, mit dem das ZIP-Archiv geöffnet wird.
*NO	Es wird keine Konvertierung durchgeführt.
*TO-WIN-ANSI	WIN-ANSI-Konvertierung wird durchgeführt (nur beim Entpacken in SAM/ISAM-Datei).
*TO-EBCDIC	EBCDIC-Konvertierung wird durchgeführt (nur beim Entpacken in SAM/ISAM-Datei).

BLOCK-CONTROL-INFO =

Steuert die Blockkontrolleigenschaft der resultierenden Datei. Dies erlaubt insbesondere das Extrahieren von Original-PAMKEY-Dateien auf einer NK-Platte.

BLOCK-CONTROL-INFO = *KEEP

Die resultierende Datei behält die Blockkontrolleigenschaft der ursprünglichen Datei.

BLOCK-CONTROL-INFO = *IGNORE

Die resultierende Datei wird mit der Standard-Blockkontrolle der Platte erstellt, auf der sie gespeichert wird.

Hinweise

- Wenn mit der Anweisung **MODIFY-ZIP-OPTIONS** Datenverschlüsselung eingestellt wurde, werden verschlüsselte Dateien beim Extrahieren wieder entschlüsselt. Die dabei verwendete Standard-Zip-2.0-Verschlüsselung ist kompatibel zu WinZip auf Windows-basierten Systemen.

- Dateien, die aus einem in BS2000 erstellten Archiv extrahiert werden, werden mit denselben Organisationsmerkmalen wie die Originaldatei erstellt, außer Blockfüllungsfaktor (PADDING-FACTOR) und Blockformat. Der Blockfüllungsfaktor ist der Standard-DMS-Blockfüllungsfaktor. Das hat zur Folge, dass die Größe der extrahierten SAM- und ISAM-Datei differieren kann von der Größe der Originaldateien.
- Dateien, die aus einem in einer fremden Umgebung erstellten Archiv extrahiert werden, werden erstellt
 - als SAM-Dateien mit BUF-LEN=STD(16), falls DATA-TYPE=*NOT-SPECIFIED oder *CHARACTER,
 - als PAM-Dateien mit BUF-LEN=STD(16), falls DATA-TYPE=*BINARY ist, und
 - als SAM-Dateien mit REC-FORM=U, falls DATA-TYPE=*SAM-BINARY ist.
- K-/NK-Platten
 - Wenn eine komprimierte Datei mit PAMKEY-Blockkontrolle auf einer NK-Platte extrahiert werden muss, dann muss man den Operanden BLOCK-CONTROL-INFO=*IGNORE verwenden. Die Datei wird in das NK-Format konvertiert. Bei SAM- oder ISAM-Dateien, deren Sätze den gesamten im Block verfügbaren Platz einnehmen, werden jedoch Daten abgeschnitten. In diesem Fall wird ein Fehler gefunden und der Extraktionsvorgang für die aktuelle Datei abgebrochen. Die Ausgabe-Datei wird gelöscht.
 - Um komprimierte Dateien (insbesondere Lade-Module) von NK-Platten auf einer K-Platte zu extrahieren, müssen Sie die BLOCK-CONTROL-INFO=*KEEP angeben.
- Wenn während der Ausführung der Anweisung EXTRACT-FILE die K2-Taste gedrückt wird, wird die Verarbeitung mit der Fragemeldung SZP0208 unterbrochen:
 - Der Benutzer kann die Verarbeitung einfach fortsetzen.
 - Der Benutzer kann die Verarbeitung beenden und kehrt zurück in den Anweisungsmodus (//). Die Dateien, die bis zum Unterbrechungszeitpunkt nicht extrahiert sind, werden nicht mehr extrahiert. Sie müssen ggf. erneut extrahiert werden.
- Bei der Auswahl von Dateien, die von fremden Plattformen kommen, ist zu berücksichtigen, dass bei den registrierten Dateinamen zwischen Groß- und Kleinschreibung unterschieden wird. Daher muss man für die Auswahl von Dateinamen mit Kleinbuchstaben das C-String-Format verwenden.

- Regeln für die Benennung extrahierter Dateien

BS2000-Dateien (beachten Sie die SDF-Regeln für Konstruktionsangaben, Handbuch „Kommandos“ [2])

FILE-NAME	TO-FILE	Eingetragener Dateiname	Resultierender Dateiname ¹
MYFILE1	*BY-SOURCE	MYFILE1	:ccid:\$cuid.MYFILE1
*	*BY-SOURCE	MYFILE1 MYFILE2	:ccid:\$cuid.MYFILE1 :ccid:\$cuid.MYFILE2
MY*	EXT-*	MYFILE1 MYFILE2	:ccid:\$cuid.EXT-FILE1 :ccid:\$cuid.EXT-FILE2
MYFILE1	*BY-SOURCE	:XXXX:\$UID.MYFILE1	keine Datei gefunden
:XXXX:\$UID.	*BY-SOURCE	:XXXX:\$UID.MYFILE1	:XXXX:\$UID.MYFILE1
\$UID.	*BY-SOURCE	\$UID.MYFILE1	:ccid:\$UID.MYFILE1

¹ Dabei ist \$cuid = aktuelle Benutzerkennung und :ccid = Katalogkennung der Benutzerkennung

Nicht-BS2000-Dateien:

FILE-NAME	TO-FILE	Eingetragener Dateiname	Resultierender Dateiname ¹
MYFILE1	*BY-SOURCE	/temp/data/myfile.txt	keine Datei gefunden
*	*BY-SOURCE	/temp/data/myfile1.txt /temp/data/myfile2.txt	:ccid:\$cuid.MYFILE1.TXT :ccid:\$cuid.MYFILE2.TXT
myfile	EXT-*	/temp/data/myfile1.txt /temp/data/myfile2.txt	:ccid:\$cuid.EXT-MYFILE1.TXT :ccid:\$cuid.EXT-MYFILE2.TXT

¹ Dabei ist \$cuid = aktuelle Benutzerkennung und :ccid = Katalogkennung der Benutzerkennung

LOGGING = *MINIMUM / *MAXIMUM

Legt den Umfang der Meldungsangabe fest.

LOGGING = *MINIMUM

Nur Fehlermeldungen werden ausgegeben

LOGGING =*MAXIMUM

Alle Meldungen werden ausgegeben. Derzeit wird nach jedem Extrahieren einer Datei die (garantierte) Meldung SZP0122 ausgegeben; in Zukunft werden gegebenenfalls weitere Meldungen ausgegeben.

3.2.6 MODIFY-ZIP-OPTIONS

Mit dieser Anweisung legen Sie Voreinstellungen für den aktuellen BS2ZIP-Programmlauf fest:

Sie können einstellen, dass BS2ZIP Dateien beim Hinzufügen verschlüsselt bzw. Extrahieren entschlüsselt. Die benötigten Schlüssel bildet BS2ZIP aus dem angegebenen Crypto-Kennwort.

MODIFY-ZIP-OPTIONS

ENCRYPTION = *UNCHANGED / *NO / *YES(...)

*YES(...)

CRYPTO-PASSWORD = <c-string 1..100 with-low> / <x-string 1..200> / *SECRET

, **CONFIRM-PASSWORD** = *NOT-SPECIFIED / <c-string 1..100 with-low> / <x-string 1..200> / *SECRET

ENCRYPTION = *UNCHANGED / *NO / *YES(...)

Gibt an, ob Dateien beim Hinzufügen verschlüsselt bzw. beim Extrahieren entschlüsselt werden sollen.

ENCRYPTION = *UNCHANGED

Die bisherige Einstellung bleibt bestehen. Beim Start von BS2ZIP ist ENCRYPTION=*NO eingestellt.

ENCRYPTION = *NO

Dateien sollen unverschlüsselt verarbeitet werden. Die Schlüssel für ein bereits angegebenes Crypto-Kennwort werden gelöscht.

ENCRYPTION = *YES(...)

Dateien sollen beim Hinzufügen verschlüsselt bzw. beim Extrahieren entschlüsselt werden.

CRYPTO-PASSWORD = <c-string 1..100> / <x-string 1..200> / *SECRET

Crypto-Kennwort, das BS2ZIP zum Verschlüsseln bzw. Entschlüsseln der Daten verwenden soll.

Der Operand hat folgende Besonderheiten:

- Der eingegebene Wert wird nicht protokolliert.
- Im geführten Dialog ist das Eingabefeld automatisch dunkelgesteuert.
- Bei Angabe von *SECRET oder ^ stellt SDF im ungeführten Dialog und in Vordergrundprozeduren ein dunkelgesteuertes Eingabefeld zur verdeckten Eingabe des Kennwortes zur Verfügung.

CONFIRM-PASSWORD = *NOT-SPECIFIED / <c-string 1..100> / <x-string 1..200> / *SECRET

Ermöglicht die Kontrolleingabe für das CRYPTO-Kennwort. Die zweimalige Eingabe des Kennworts soll verhindern, dass bei dunkelgesteuerter Eingabe ein versehentlich durch Tippfehler entstandenes Kennwort vergeben wird.

Sofern statt des Wertes *NOT-SPECIFIED (Voreinstellung) ein Kennwort angegeben wird, muss dieses mit der Angabe im Operanden CRYPTO-PASSWORD übereinstimmen. Andernfalls wird die Anweisung abgewiesen.

Der Operand hat folgende Besonderheiten:

- Der eingegebene Wert wird nicht protokolliert.
- Im geführten Dialog ist das Eingabefeld automatisch dunkelgesteuert.
- Bei Angabe von *SECRET oder ^ stellt SDF im ungeführten Dialog und in Vordergrundprozeduren ein dunkelgesteuertes Eingabefeld zur verdeckten Eingabe des Kennwortes zur Verfügung.

Hinweis

Das angegebene Crypto-Kennwort selbst wird nicht gespeichert. BS2ZIP bildet aus dem Crypto-Kennwort entsprechende Schlüssel und speichert nur diese.

3.2.7 OPEN-ZIP-CONTAINER

Mit dieser Anweisung öffnen Sie ein ZIP-Archiv. Wenn bereits ein ZIP-Archiv geöffnet war, wird es vorher geschlossen.

Voreingestellt wird das ZIP-Archiv im Lesemodus geöffnet. Sollen Dateien hinzugefügt oder daraus gelöscht werden, muss es im Schreibmodus geöffnet werden.

OPEN-ZIP-CONTAINER

```

CONTAINER = <filename 1..54 without-gen-vers> / *LINK(...)
    *LINK(...)
        |   LINK-NAME = <structured-name 1..8>
, MODE = *READ / *UPDATE(...)
    *UPDATE(...)
        |   STATE = *ANY / *NEW
, FORMAT = *STD / *WINZIP-COMPATIBLE / *BS2000

```

CONTAINER = <filename 1..54 without-gen-vers>

Name des zu öffnenden ZIP-Archivs.

CONTAINER = *LINK(...)

Das zu öffnende ZIP-Archiv wird durch einen Linknamen angesprochen.

LINK-NAME = <structured-name 1..8>

Linkname, der dem zu öffnenden ZIP-Archiv zugewiesen ist.

MODE = *READ

Das ZIP-Archiv wird im Lesemodus geöffnet. Die Datei muss existieren.

MODE = *UPDATE(...)

Das ZIP-Archiv wird im Schreibmodus geöffnet.

STATE = *ANY

Das ZIP-Archiv wird im Update-Modus geöffnet. Falls die angegebene Datei nicht existiert, wird sie vorher neu erstellt.

STATE = *NEW

Das ZIP-Archiv wird neu erstellt und im Update-Modus geöffnet. Wenn es die Datei schon gibt, wird die Anweisung mit Fehler abgewiesen.

FORMAT = *STD

Wenn das Archiv nicht existiert und neu erstellt werden muss, d.h. bei MODE=*UPDATE, dann wird die Datei im BS2000-Format erstellt.

Wenn das Archiv existiert, d.h. bei MODE=*READ bzw. *UPDATE(STATE=*ANY), dann wird die Datei in dem Format geöffnet, in dem sie erstellt wurde.

FORMAT = *BS2000

Das ZIP-Archiv wird im BS2000-Format geöffnet. ZIP-Archive in diesem Format können nur in einer BS2000-Umgebung verwendet werden.

FORMAT = *WINZIP-COMPATIBLE

Das ZIP-Archiv wird im WinZip-kompatiblen Format geöffnet. ZIP-Archive in diesem Format können in der offenen Welt verwendet werden, da sie kompatibel mit PKZIP V4.5 sind.

Hinweise

- Wenn ein ZIP-Archiv mit openFT < V11.0 nach BS2000 übertragen wurde, besitzt es das Dateiformat SAM mit RECORD-FORMAT=U. Bevor dieses Archiv in BS2ZIP geöffnet wird, muss es mit der Anweisung CONVERT-ZIP-CONTAINER in das Dateiformat PAM konvertiert werden.
Umgekehrt muss ein ZIP-Archiv von PAM in das Dateiformat SAM mit RECORD-FORMAT=U konvertiert werden, bevor es mit openFT < V11.0 in ein anderes System transferiert werden kann. Siehe „[CONVERT-ZIP-CONTAINER](#)“ auf Seite 33.
- Wenn die Datei nicht als gültiges ZIP-Archiv erkannt wird, dann wird eine Fehlermeldung ausgegeben.
- Ein im Format WINZIP-COMPATIBLE erstelltes ZIP-Archiv kann nicht im BS2000-Format geöffnet werden und ein im BS2000-Format erstelltes ZIP-Archiv kann nicht im Format WINZIP-COMPATIBLE geöffnet werden.
- Ein unter Windows- oder Unix-Systemen erstelltes ZIP-Archiv kann nur im Format WINZIP-COMPATIBLE geöffnet werden.
- GZIP-Archive können nur mit MODE=*READ geöffnet werden. Somit ist nur Auflisten der Dateien und das Extrahieren möglich. Als Format darf nur die Voreinstellung FORMAT=*STD angegeben werden.
- Öffnet man ein Archiv, während ein anderes Archiv geöffnet ist, wird dieses aktuell geöffnete Archiv geschlossen. Das aktuelle Archiv wird auch geschlossen, wenn das Öffnen des zweiten Archivs fehlschlägt.
- Die Anweisung OPEN-ZIP-CONTAINER kann nicht mit K2 unterbrochen werden.

- Erreichen der maximalen Anzahl an Extents des ZIP-Archivs

Die Speicherplatzbelegung eines ZIP-Archivs kann der Benutzer explizit durch eine entsprechende Primär- und Sekundärzuweisung beim Erstellen der Datei mit dem CREATE-FILE-Kommando beeinflussen. Wenn die Datei noch nicht existiert, erstellt BS2ZIP das ZIP-Archiv mit einer Ausgangsgröße von 1920 PAM-Seiten (Primärzuweisung) und einer Sekundärzuweisung von 192 PAM-Seiten.

Wenn das Archiv voraussichtlich groß wird, sollte der Benutzer eine höhere Sekundärzuweisung festlegen. Andernfalls besteht die Gefahr, dass das Archiv durch ständiges Erweitern die maximal mögliche Anzahl Extents erhält und dann nicht mehr erweitert werden kann. Eine mit BS2ZIP-Standardwerten angelegte Datei kann maximal etwa 120 MB groß werden.

Beispiel für eine Abschätzung

```
DATAFILE1 100.000pp
DATAFILE2  50.000pp
Total =   150.000pp
```

Mit einer Komprimierungsrate von 60% kann die Größe des resultierenden ZIP-Archivs auf 60.000 PAM-Seiten geschätzt werden. Sie können weniger als diesen Wert schätzen. Dann wird BS2ZIP die Datei erweitern, falls notwendig.

Führen Sie folgendes CREATE-FILE-Kommando aus:

```
/CREATE-FILE MYCONT.ZIP,SUPPORT=*PUBLIC(SPACE=*RELA(60000,100))
/START-ZIP
//OPEN-ZIP-CONTAINER CONTAINER=MYCONT.ZIP,MODE=*UPDATE(STATE=*ANY)
//ADD DATAFILE*
//END
```

Umgekehrt kann die Ausgangsgröße von 1920 PAM-Seiten überhöht sein, auch wenn sie zeitlich begrenzt ist. Um dieses Problem zu vermeiden, sollte der Benutzer die endgültige Größe abschätzen, und im SPACE-Operanden des CREATE-FILE-Kommandos eine entsprechende Primärzuweisung angeben. Für sehr kleine ZIP-Archive kann z.B. eine Primärzuweisung von nur 6 PAM-Seiten ausreichen.

- Wenn der Benutzer weniger als 1920 PAM-Seiten freier Speicherplatz zur Verfügung stehen, ist es notwendig das Archiv anzulegen, bevor man es mit BS2ZIP öffnet.
- Die Größe eines Archivs für die selben Inhalte ist variabel, abhängig davon, wie es eingerichtet und bearbeitet wurde (einschließlich Reorganisation mit REORGANIZE-ZIP-CONTAINER).

3.2.8 REORGANIZE-ZIP-CONTAINER

Mit dieser Anweisung reduzieren Sie den Speicherbedarf eines ZIP-Archivs. Beim Reorganisieren wird Speicherplatz für Dateien, die aus dem ZIP-Archiv gelöscht wurden, entfernt und der reservierte Speicherplatz des ZIP-Archivs so weit wie möglich reduziert.

REORGANIZE-ZIP-CONTAINER

CONTAINER = *STD / <filename 1..54 without-gen-vers> / *LINK(...)

*LINK(...)

| **LINK-NAME** = <structured-name 1..8>

CONTAINER = *STD

Es wird das geöffnete ZIP-Archiv reorganisiert. Ist noch kein ZIP-Archiv geöffnet, wird die Anweisung zurückgewiesen.

CONTAINER = <filename 1..54 without-gen-vers>

Name des zu reorganisierenden ZIP-Archivs.

CONTAINER = *LINK(...)

Das zu reorganisierende ZIP-Archiv wird über einen Linknamen angesprochen.

LINK-NAME = <structured-name 1..8>

Linkname, der dem zu reorganisierenden ZIP-Archiv zugewiesen ist.

Hinweise

- Während des Reorganisierens ist kein Zugriff auf das ZIP-Archiv möglich.
- Wenn das ZIP-Archiv bereits von einer anderen Task geöffnet ist, kann es nicht reorganisiert werden.

3.2.9 SHOW-FILE-ATTRIBUTES

Mit dieser Anweisung listen Sie die Dateien des aktuell geöffneten ZIP-Archivs auf.

Die Anweisung unterstützt die strukturierte Ausgabe in S-Variable.

SHOW-FILE-ATTRIBUTES

```

FILE-NAME = *ALL / <composed-name 1..98 with-under with-wild(132)> / <c-string 1..1024 with-low>
,INFORMATION = *SUMMARY / *ALL
,TEXT-OUTPUT = *SYSOUT / *SYSLST(...) / *NONE
    *SYSLST(...)
        | SYSLST-NUMBER = *STD / <integer 1..99>
,STRUCTURE-OUTPUT = *SYSINF / *NONE / <composed-name 1..255>(…)
    <composed-name 1..255>(…)
        | WRITE-MODE = *REPLACE / *EXTEND

```

FILE-NAME = *ALL

Alle im Archiv enthaltenen Dateien werden aufgelistet.

FILE-NAME = <composed-name 1..98 with-under with-wild(132)>

Alle dem Muster entsprechenden Dateien werden aufgelistet.

FILE-NAME = <c-string 1..1024 with-low>

Alle Dateien, die der angegebenen Zeichenfolge entsprechen (Musterzeichen nach den SDF-Regeln für Auswahlangaben sind erlaubt (siehe SDF-Metasyntax, Handbuch „Kommandos“ [2]), werden aus dem ZIP-Archiv aufgelistet. Die Angabe als C-String ist zu verwenden, wenn das Archiv nicht in einem BS2000-System erstellt wurde und die betroffenen Dateinamen nicht BS2000-konform sind (z.B. Groß-/Kleinschreibung).

INFORMATION = *SUMMARY

Nur der Name der archivierten Datei sowie ihre Herkunft werden angezeigt. Die Ausgabe endet mit der Meldung SZP0087, die die Anzahl der insgesamt aufgelisteten Dateien anzeigt. Siehe auch „[Layout der Ausgabeinformation](#)“ auf Seite 52.

INFORMATION = *ALL

Es wird die vollständige Information für die archivierte Datei angezeigt. Die Ausgabe endet mit der Meldung SZP0087, die die Anzahl der insgesamt aufgelisteten Dateien anzeigt. Siehe auch „[Layout der Ausgabeinformation](#)“ auf Seite 52.

TEXT-OUTPUT = *SYSOUT / *SYSLST(...) /*NONE

Gibt an, wohin die Information auszugeben ist (Ausgabe als Textstring).

TEXT-OUTPUT = *SYSOUT

Die Ausgabe erfolgt in die Systemdatei SYSOUT.

TEXT-OUTPUT = *NONE

Mit Ausnahme von Fehlermeldungen erfolgt keine Informationsausgabe.

TEXT-OUTPUT = *SYSLST(...)

Die Ausgabe erfolgt in die Systemdatei SYSLST.

SYSLST-NUMBER = *STD / <integer 1..99>

Bestimmt, ob die Ausgabe in die Systemdatei SYSLST oder in eine SYSLST-Datei aus der Menge SYSLST01 bis SYSLST99 erfolgen soll.

Voreinstellt ist *STD, d.h. die Ausgabe erfolgt in die Systemdatei SYSLST.

STRUCTURE-OUTPUT = *SYSINF / *NONE / <composed-name 1..255>(...)

Steuert die strukturierte Ausgabe in S-Variable (siehe auch „[Strukturierte Ausgabe in S-Variable](#)“ auf Seite 54).

STRUCTURE-OUTPUT = *SYSINF

Die strukturierte Ausgabe wird in den S-Variablenstrom SYSINF gelenkt. Die Informationen sind zugreifbar, wenn SYSINF mit dem Kommando ASSIGN-STREAM vorher einer S-Variablen zugewiesen wurde (siehe „Ausgabe in S-Variablen“, Handbuch „Kommandos“ [2]).

STRUCTURE-OUTPUT = *NONE

Es erfolgt keine strukturierte Ausgabe.

STRUCTURE-OUTPUT = <composed-name 1..255>(...)

Name der Variablen, in welche die strukturierte Ausgabe erfolgen soll. Die Variable muss als Listenvariable deklariert sein:

```
/DECLARE-VARIABLE VAR-NAME=<var>(TYPE=*STRUCTURE),MULTIPLE-ELEMENTS=*LIST
```

WRITE-MODE = *REPLACE / *EXTEND

Gibt an, ob die strukturierte Ausgabe eine bestehende Liste überschreiben oder erweitern soll. Mit *REPLACE ersetzt die Ausgabe bereits existierende Listenelemente.

Im Fall von Nicht-BS2000-Dateien hat dieser Name das Format, mit dem er gespeichert wurde. Zum Beispiel /dir1/dir2/Myfile.txt. Groß- und Kleinschreibung sind berücksichtigt.

In einer Zeile werden maximal 64 Zeichen des Dateinamens angezeigt. Bei Dateinamen, die länger als 64 Zeichen sind, werden die restlichen Zeichen in Folgezeilen angezeigt. Maximal sind 1024 Zeichen möglich.

BS2000	Gibt die Herkunft der Datei an.
YES	bedeutet, dass die Datei im BS2000 mit dem Programm BS2ZIP in das Archiv aufgenommen wurde.
NO	steht für alle anderen Fälle.
MODIFIED	steht für das Datum und die Uhrzeit der letzten Änderung an der Datei.
SIZE	ist die Größe der Originaldatei in Byte.
PACKED	ist die Größe der komprimierten Datei in Byte.
RATIO	ist die Komprimierungsrate. Die Berechnung erfolgt nach der Formel: (SIZE – PACKED) * 100 / SIZE.
	Es kann vorkommen, dass die komprimierte Datei größer ist als die Originaldatei, wenn in der Anweisung ADD-FILE keine Komprimierung verlangt wurde. In diesem Fall hat RATIO den Wert 0.
ENCRYPTED	Gibt an, ob die Datei beim Aufnehmen in das Archiv verschlüsselt wurde.
YES	zeigt eine verschlüsselte Datei an.
NO	zeigt eine unverschlüsselte Datei an.
COMMENTS	steht für die einer Datei zugeordneten Kommentare. Bei BS2000-Dateien beginnt der Kommentar mit der Zeichenfolge *BS2: und listet danach die DMS-Eigenschaften der Originaldatei als Operanden eines FILE-Makroaufrufs auf (siehe Handbuch „DMS Makroaufrufe“ [4]).

Hinweis

Das Format des Archivs ist auch in dessen Katalog-Eintrag im Feld USER-INFO (Abschnitt Organization) abgelegt:

USER-INFO = BS2ZIP-B für BS2000-Format

USER-INFO = BS2ZIP-W für WinZip-kompatibles Format

Strukturierte Ausgabe in S-Variable

Mit dem Operanden INFORMATION=*SUMMARY werden nur die S-Variablen für den Dateinamen und die Herkunft der Datei (FILENAME und BS2000) mit Werten versorgt. Mit INFORMATION=*ALL werden alle aufgelisteten S-Variablen mit Werten versorgt.

Ausgabe-Information	Name der S-Variablen	T	Inhalt
Indikator für Sekundär-Index (bei ISAM-Datei)	var(*LIST).ALTERNATE-INDEX	S	" *YES
Blockkontrollinformation	var(*LIST).BLKCTRL	S	NO PAMKEY DATA2K DATA4K DATA
Puffergröße	var(*LIST).BLKSIZE	S	" <integer> STD(n)
Herkunft der Datei	var(*LIST).BS2000	S	*YES *NO
Name der Code-Tabelle	var(*LIST).CODED-CHAR-SET	S	" <ccs>
Dateiverschlüsselung	var(*LIST).ENCRYPTED		*YES *NO
Dateistruktur (Zugriffsmethode, mit der die Datei erstellt wurde)	var(*LIST).FCBTYPE	S	SAM ISAM PAM
Name der Datei; bei Nicht-BS2000-Dateien ggf. als Pfadname	var(*LIST).FILENAME	S	<filename> <pathname 1..1024>
Indikator für den Dateityp PLAM-Bibliothek	var(*LIST).FILETYPE	S	" PLAM-LIB
Performance-Eigenschaft bzgl. Ein-/Ausgabeoperationen	var(*LIST).IOPERF	S	" HIGH STD VERY-HIGH
Operation(en), auf die sich die Performance-Eigenschaft bezieht	var(*LIST).IOUSAGE	S	" READ RDWRT WRITE
Länge des ISAM-Schlüssels	var(*LIST).KEYLEN	I	<integer>
Position des ISAM-Schlüssels im Datensatz	var(*LIST).KEYPOS	I	<integer>
Länge der logischen Markierung im ISAM-Index	var(*LIST).LOGLEN	I	<integer>
Datum des letzten ändernden Dateizugriffs	var(*LIST).MODIFIED-DATE	S	<yyyy-mm-dd>

Ausgabe-Information	Name der S-Variablen	T	Inhalt
Zeitpunkt (Uhrzeit) des letzten ändernden Dateizugriffs	var(*LIST).MODIFIED-TIME	S	<hh:mm:ss>
Dateigröße im komprimierten Zustand (in Byte)	var(*LIST).PACKED	I	<integer>
Komprimierungsrate	var(*LIST).RATIO	S	nn.n
Satzformat der Datei	var(*LIST).RECFORM	S	F U V
Satzlänge der Datei	var(*LIST).RECSIZE	I	<integer>
Sekundärzuweisung für Dateierweiterung	var(*LIST).SEC-ALLOC	I	<integer>
Original-Dateigröße (in Byte)	var(*LIST).SIZE	I	<integer>
Länge der Wertmarkierung im ISAM-Index	var(*LIST).VALLEN	I	<integer>
Behandlung der Wertmarkierung innerhalb eines Daten- oder Indexblocks (für K-ISAM-Dateien)	var(*LIST).VALPROP	S	" MAX MIN

Beispiel

```

/START-ZIP-MANAGER
//OPEN-ZIP-CONTAINER MYCONT.ZIP
//SHOW-FILE-ATTRIBUTES
CURRENT CONTAINER : MYCONT.ZIP
$DIAGDUMP.A0478578.SLED.S210.VM2.110611                BS2000 : YES
% SZP0087 '1' file(s) are matching your request
//SHOW-FILE-ATTRIBUTES *ALL,INFORMATION=ALL
CURRENT CONTAINER : MYCONT.ZIP
----- FILE INFORMATION -----
FILENAME      : $DIAGDUMP.A0478578.SLED.S210.VM2.110611
BS2000       : YES
MODIFIED     : 2011-06-11 18:49:32
SIZE        : 227454976
PACKED      : 35012077
RATIO       : 84.6 %
ENCRYPTED    : NO
----- COMMENTS -----
*BS2: ,FCBTYPE=PAM,BLKSIZE=(STD,1),CODED-CHAR-SET=EDF03IRV,BLKCTRL=NO
% SZP0087 '1' file(s) are matching your request

```

3.2.10 START-TRACE

Mit dieser Anweisung aktivieren Sie einen Trace.

START-TRACE
TRACE-FILE = *<u>STD</u> / <filename 1..54 without-gen-vers>

TRACE-FILE = *STD

Die Trace-Datei erhält den Standardnamen SYSTRC.BS2ZIP.<yyyy-mm-dd>.<hhmmss> und wird unter der Benutzerkennung abgelegt, die das Programm BS2ZIP ausführt.

TRACE-FILE = <filename 1..54 without-gen-vers>

Eine Trace-Datei mit benutzerspezifischem Dateinamen wird erstellt. Existiert die Datei bereits, wird sie überschrieben.

3.2.11 STOP-TRACE

Mit dieser Anweisung deaktivieren Sie den Trace. Die Trace-Datei wird geschlossen.

STOP-TRACE

3.3 Beispiele

BS2000-Beispiel

```

/show-file-attr test.
%          3 :10SN:$BS2ZIP.TEST.ISAM-STD1-PAMKEY-VN0-EDF03IRV-8-5
%          18 :10SN:$BS2ZIP.TEST.ISAM2
%           6 :10SN:$BS2ZIP.TEST.IS088591
%           3 :10SN:$BS2ZIP.TEST.NKEY
%          12 :10SN:$BS2ZIP.TEST.NKFILE
%          12 :10SN:$BS2ZIP.TEST.PAM
%          51 :10SN:$BS2ZIP.TEST.PAM.16VP-32768
%          36 :10SN:$BS2ZIP.TEST.PAM16-MAX.2
%          21 :10SN:$BS2ZIP.TEST.PAM2
%          48 :10SN:$BS2ZIP.TEST.RECFORMU
%         192 :10SN:$BS2ZIP.TEST.SAM.16VP-32768
%          18 :10SN:$BS2ZIP.TEST.SAM1-MAX
/start-zip-manager
% SZPLOAD Program 'BS2ZIP',Version 'V01.2G15' of '2014-09-15' loaded
% SZPCOPY Copyright (C) 2014 Fujitsu Technology Solutions GmbH All Rights
Reserved
//open-zip-container container=mycont,mode=*update(state=*new)
//add-file test.
//show-file-attr
CURRENT CONTAINER : MYCONT
TEST.ISAM-STD1-PAMKEY-VN0-EDF03IRV-8-5                BS2000 : YES
TEST.ISAM2                                              BS2000 : YES
TEST.IS088591                                           BS2000 : YES
TEST.NKEY                                               BS2000 : YES
TEST.NKFILE                                             BS2000 : YES
TEST.PAM                                                BS2000 : YES
TEST.PAM.16VP-32768                                     BS2000 : YES
TEST.PAM16-MAX.2                                       BS2000 : YES
TEST.PAM2                                              BS2000 : YES
TEST.RECFORMU                                          BS2000 : YES
TEST.SAM.16VP-32768                                    BS2000 : YES
TEST.SAM1-MAX                                          BS2000 : YES
% SZP0087 '12' file(s) are matching your request

```

```
//show-file-attr file-name=test.isam-std.*,inf=*all
CURRENT CONTAINER : MYCONT
----- FILE INFORMATION -----
FILENAME      : TEST.ISAM-STD1-PAMKEY-VN0-EDF03IRV-8-5
BS2000       : YES
MODIFIED     : 2014-11-25 10:21:22
SIZE        : 231
PACKED      : 56
RATIO      : 75.8 %
ENCRYPTED   : NO
----- COMMENTS -----
*BS2:  ,FCBTYPE=ISAM,BLKSIZE=(STD,1),KEYPOS=5,KEYLEN=8,RECFORM=(V,N),
IOPERF=STD,IOUSAGE=RDWRT,CODED-CHAR-SET=EDF03IRV,BLKCTRL=PAMKEY
% SZP0087 '1' file(s) are matching your request
//extract-files file-name=test.pam.*,to-file=ext-test.pam.*
//end
/show-file-attr ext-*
%          51 :10SN:$BS2ZIP.EXT-TEST.PAM.16VP-32768
%          33 :10SN:$BS2ZIP.EXT-TEST.PAM16-MAX.2
%          21 :10SN:$BS2ZIP.EXT-TEST.PAM2
```

Beispiel mit WinZip-Archiv

```

/start-ftp
.
.
.
ftp> dir
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
05-15-14 03:19PM          978691 server_1.zip
226 Transfer complete.
53 bytes received in 0.08 seconds (0.62 Kbytes/s)
ftp> bin
200 Type set to I.
ftp> get server_1.zip SERVER-1.PC.ZIP
200 PORT command successful.
150 Opening BINARY mode data connection for server_1.zip(978691 bytes).
226 Transfer complete.
978691 bytes received in 6.90 seconds (1.4e+02 Kbytes/s)
ftp> bye
221
% CCM0998 CPU TIME USED: 0.0925 SECONDS
/show-file-attr server-1.pc.zip,inf=all
%0000000480 :20SG:$USER1.SERVER-1.PC.ZIP
% ----- HISTORY -----
% CRE-DATE   = 2014-05-15  ACC-DATE   = 2014-05-15  CHANG-DATE = 2014-05-15
% CRE-TIME   = 15:46:32   ACC-TIME   = 15:47:45  CHANG-TIME = 15:46:39
% ACC-COUNT  = 3          S-ALLO-NUM = 0
% ----- SECURITY -----
% READ-PASS  = NONE       WRITE-PASS  = NONE     EXEC-PASS  = NONE
% USER-ACC   = OWNER-ONLY ACCESS        = WRITE     ACL        = NO
% AUDIT      = NONE       FREE-DEL-D  = *NONE     EXPIR-DATE = 2014-05-15
% DESTROY    = NO         FREE-DEL-T  = *NONE     EXPIR-TIME = 00:00:00
% SP-REL-LOCK= NO        ENCRYPTION  = *NONE
% ----- BACKUP -----
% BACK-CLASS = A          SAVED-PAG  = COMPL-FILE VERSION    = 1
% MIGRATE    = ALLOWED
% ----- ORGANIZATION -----
% FILE-STRUC = PAM        BUF-LEN    = STD(1)    BLK-CONTR  = PAMKEY
% IO(USAGE)  = READ-WRITE IO(PERF)    = STD        DISK-WRITE = IMMEDIATE
% AVAIL      = *STD
% WORK-FILE  = *NO        F-PREFORM  = *K        SO-MIGR    = *ALLOWED
% ----- ALLOCATION -----
% SUPPORT    = PUB        S-ALLOC    = 16        HIGH-US-PA = 478
% EXTENTS    VOLUME      DEVICE-TYPE EXTENTS    VOLUME     DEVICE-TYPE
% 1          GVS2.5      D3435
% NUM-OF-EXT = 1
%:20SG: PUBLIC:          1 FILE RES=          480 FRE=          2 REL=          0 PAGES

```

```

/start-zip
% SZPLOAD Program 'BS2ZIP',Version 'V01.2G15' of '2014-09-15' loaded
% SZPCOPY Copyright (C) 2014 Fujitsu Technology Solutions GmbH All Rights
Reserved
//open-zip-container server-1.pc.zip
//show-file-attr '*\<ADD,Glos,Ind>*.htm'
CURRENT CONTAINER : SERVER-1.PC.ZIP
server_1/osd_allg/projekte/test_kdo_1/Output/new/MyWebHelp/Conte BS2000 : NO
nt/test_kdo_1/ADD_CJC-ACTION.htm
server_1/osd_allg/projekte/test_kdo_1/Output/new/MyWebHelp/Conte BS2000 : NO
nt/test_kdo_1/ADD_FILE_LINK__TFT_Eintr.htm
server_1/osd_allg/projekte/test_kdo_1/Output/new/MyWebHelp/Conte BS2000 : NO
nt/test_kdo_2/ADD_CJC_ACTION.htm
server_1/osd_allg/projekte/test_kdo_1/Output/new/MyWebHelp/Conte BS2000 : NO
nt/test_kdo_2/ADD_FILE_LINK.htm
server_1/osd_allg/projekte/test_kdo_1/Output/new/MyWebHelp/Conte BS2000 : NO
nt/Glossary.htm
server_1/osd_allg/projekte/test_kdo_1/Output/new/MyWebHelp/Skin/ BS2000 : NO
Index.htm
% SZP0087 '6' file(s) are matching your request
//show-file-attr *ADD/CJC*
CURRENT CONTAINER : SERVER-1.PC.ZIP
server_1/osd_allg/projekte/test_kdo_1/Output/new/MyWebHelp/Conte BS2000 : NO
nt/test_kdo_1/ADD_CJC-ACTION.htm
server_1/osd_allg/projekte/test_kdo_1/Output/new/MyWebHelp/Conte BS2000 : NO
nt/test_kdo_2/ADD_CJC_ACTION.htm
% SZP0087 '2' file(s) are matching your request
//show-file-attr '*.css'
server_1/osd_allg/projekte/test_kdo_1/Output/new/MyWebHelp/Conte BS2000 : NO
nt/Resources/Stylesheets/test_kdo_1.css
server_1/osd_allg/projekte/test_kdo_1/Output/new/MyWebHelp/Conte BS2000 : NO
nt/SkinSupport/MadCap.css
% SZP0087 '2' file(s) are matching your request
//extract-file file-name=*ADD/CJC*,to-file=webhelp.*.copy
% SZP0090 Warning. File name 'WEBHELP.ADD_CJC-ACTION.HTM.COPY' is not
BS2000 compliant. The file will be extracted under the name
'FILE0001.20140515.155645'
% SZP0090 Warning. File name 'WEBHELP.ADD_CJC_ACTION.HTM.COPY' is not
BS2000 compliant. The file will be extracted under the name
'FILE0002.20140515.155645'
//extract-file file-name='*.css',to-file=webhelp.*.copy
% SZP0090 Warning. File name 'WEBHELP.TEST_KDO_1.CSS.COPY' is not BS2000
compliant. The file will be extracted under the name
'FILE0001.20140515.155759'
//end

```

```
/show-file-attr <file,webhelp>*
%      33 :20SG:$USER1.FILE0001.20140515.155645
%      48 :20SG:$USER1.FILE0001.20140515.155759
%      33 :20SG:$USER1.FILE0002.20140515.155645
%      33 :20SG:$USER1.WEBHELP.MADCAP.CSS.COPY
%:20SG: PUBLIC:      4 FILES RES=      147 FRE=      19 REL=      0 PAGES
```

4 Interoperabilität

4.1 Interoperabilität mit Windows

Windows-Programme, die das ZIP-Format unterstützen (WinZip®), können BS2ZIP-Archive lesen, wenn diese im WinZip-kompatiblen Modus erstellt wurden.

BS2ZIP kann Windows-ZIP-Archive lesen, wenn diese im WinZip-kompatiblen Modus geöffnet werden und für BS2ZIP lesbar sind.

Die Archive müssen im Binärformat übertragen werden. Das folgende Beispiel zeigt eine solche Übertragung.

Eingabe an der DOS-Eingabeaufforderung:

```
D:\>ftp
ftp>open 999.999.999.999
connected to 999.999.999.999
user : bs2zip
password : *****
account : xxxx
ftp>binary
ftp>quote ftyp binary
ftp>get bs2000.zip
ftp>put pc.zip
ftp>bye
```

D:\bs2000.zip kann von WinZip geöffnet werden und \$BS2ZIP.PC.ZIP kann durch BS2ZIP geöffnet werden.



Die Länge der SAM- und ISAM-Datei-Datensätze ist begrenzt durch die logische Blocklänge (BUFFER-LENGTH) der Datei. Da Datensätze von Windows-Textdateien keiner Längenbeschränkung unterliegen, versucht BS2ZIP das Risiko von zu langen Datensätzen zu minimieren und legt die logische Blocklänge immer mit BUF-LEN=*STD(16) fest, die 32 KB - 8 Byte entspricht. Dies muss von den Anwendungen, die solche Dateien lesen, berücksichtigt werden.

Wenn längere Datensätze gefunden werden, werden sie in so viele Datensätze unterteilt, wie notwendig.

Übertragen von ZIP-Dateien mit openFT

Wenn Sie ein ZIP-Archiv mit openFT < V11.0 von/nach Windows übertragen, dann müssen Sie das ZIP-Archiv entweder vor dem Transfer (bei Transfer nach Windows) oder nach dem Transfer (bei Transfer von Windows) in das richtige Dateiformat konvertieren.

Es gilt Folgendes:

1. BS2ZIP verarbeitet nur PAM-Dateien mit BUFFER-LENGTH=STD(x),
2. PAM-Dateien dürfen nicht mit openFT < V11.0 nach Windows übertragen werden.
3. Die Dateien müssen im Binärmodus übertragen werden (verbindlich für gezippte Dateien).
4. Die Ergebnis-Datei, die (von Windows) mit openFT < V11.0 übertragen wird, ist eine SAM-Datei mit BUFFER-LENGTH=STD(x) und RECORD-FORMAT=U.

SAM/PAM-Konvertierung

Die Konvertierung können Sie in BS2ZIP mit der Anweisung CONVERT-ZIP-CONTAINER vornehmen. Siehe Beschreibung von „[CONVERT-ZIP-CONTAINER](#)“ auf [Seite 33](#).

Aus Kompatibilitätsgründen kann der Konverter auch über das Kommando /START-SAM-PAM-CONVERTER aufgerufen werden:

Der Konverter fordert Sie auf, die Eingabe-Datei (erster Parameter) und die Ausgabe-Datei (zweiter Parameter) anzugeben. Der Konverter wandelt die Eingabedatei wie folgt um:

- eine PAM-Datei mit BUF-LEN=STD(x) in eine SAM-Datei mit BUF-LEN=STD(8)
- eine SAM-Datei mit BUF-LEN=STD(x) in eine PAM-Datei mit BUF-LEN=STD(16)

4.2 Interoperabilität mit Unix-Systemen

GZIP- oder GZ-Werkzeuge auf Unix-Systemen generieren GZIP-Archive. Diese müssen binär an das BS2000 gesendet werden und können dann von BS2ZIP gelesen werden. Sie können nicht von BS2ZIP geändert werden.

Standardmäßig extrahiert BS2ZIP die archivierte Datei in eine SAM-Datei mit BUF-LEN=STD(16). Es wird dabei angenommen, dass das Zeilentrennzeichen ein Zeilenvorschub (LF) ist. Dateien können auch binär extrahiert werden, dabei werden PAM-Dateien mit BUF-LEN=STD(16) erzeugt.



Nur gz-Dateien werden unterstützt; tar.gz-Dateien werden nicht unterstützt.

4.3 Interoperabilität mit Linux

Archive, die von einem Linux-ZIP-Werkzeug erzeugt wurden, sind von BS2ZIP lesbar. Diese Archive müssen im Binärformat an das BS2000 übertragen werden. Standardmäßig extrahiert BS2ZIP die archivierte Datei in eine SAM-Datei mit BUF-LEN=STD(16). Es wird dabei angenommen, dass das Zeilentrennzeichen ein Wagenrücklauf mit Zeilenvorschub (CRLF) ist. Dateien können auch binär extrahiert werden. Dabei werden PAM-Dateien mit BUF-LEN=STD(16) erzeugt.

Auf der anderen Seite können BS2ZIP-Archive, die im Kompatibilitätsmodus erzeugt wurden, binär auf eine Linux-Plattform übertragen und dann vom Linux-Tool UNZIP weiterverarbeitet werden.

5 Programmschnittstelle

BS2ZIP kann als Unterprogramm in TU-Anwendungen verwendet werden. Zu diesem Zweck wird in der Bibliothek SYSLNK.BS2ZIP.<version>.RTE eine Programmschnittstelle (API) und ein Laufzeitmodul zur Verfügung gestellt:

- Der C++ Datei-Header SZPZIP.H: API von BS2ZIP
- Der C++ Datei-Header SZPZOUT.H: Layout für den Ausgabe-Puffer von *ListFiles()*
- Das Laufzeitmodul BS2ZIPPR

5.1 SZPZIP.H

```
#ifndef __SzpZip_h__
#define __SzpZip_h__
/*****
 * BEGIN-INTERFACE  SZPZIP.H
 *
 * TITLE    (/ zip program interface /)
 * Classes      CSzpZip
 * NAME         szpzip.h
 * DOMAIN       BS2ZIP
 * LANGUAGE     CXX
 * Copyright    FUJITSU TECHNOLOGY SOLUTIONS 2014
 *              All rights reserved
 * COMPILATION-SCOPE USER
 * INTERFACE-TYPE  CALL
 * RUN-CONTEXT    TU
 *
 * VERSION      133
 * CRDATE       2014-04-08
 * AUTHOR       (/ J. Beaume, OSL41 /)
 * UPDATE       (/ BS2ZIP v1.2G10 /)
 *
 * END-INTERFACE
 *****/
/*
 * Version      132
```

```
* Date          2013-02-01
* Author        P. Louis - OSL EPS
* Update        BS2ZIP v1.2E05
*
* Version       130
* Date          2011-05-18
* Author        J. Beaume - OSL EPS
* Update        BS2ZIP v1.2D05
*
* Version       129
* Date          2009-09-30
* Author        P. Louis - OSL EPS
* Update        BS2ZIP v1.2C05
*
* Version       128
* Date          2008-11-27
* Author        P. Louis - OSL EPS
* Update        BS2ZIP v1.2B05
*               Introduction CpuExhauster (A0571814)
*
* Version       126
* Date          2008-10-14
* Author        Ph/Dumont- OSL EPS
* Update        BS2ZIP v1.2A25
*               Introduction CpuExhauster (A0570787)
*
* Version       121
* Date          2008-04-18
* Author        P.Louis - OSL EPS
* Update        BS2ZIP v1.2A10
*
* Version       118
* Date          2007-12-01
* Author        P.Louis - OSL EPS
* Update        BS2ZIP v1.2A
*
* Version       116
* Date          2006-12-01
* Author        Ph.Dumont - OSL EPS
* Update        BS2ZIP v1.1C
*
* Version       115
* Date          2005-06-23
* Author        L. Tambour - OSL EPS
* Update        BS2ZIP v1.1B
*
* Version       110
* Date          2004-09-30
```

```

* Author      L. Tambour - OSL EPS
* Update     BS2ZIP v1.1A
*
* Version    100
* Date       2003-06-02
* Author     J. Beaume - OSL EPS
* Update     BS2ZIP v1.0B
*****/
#include "szpzout.h"
// SzpZip return codes:
#define CSZPZIP_LIB_EXCEPT_ERROR    114    //133
#define CSZPZIP_FILE_EXCEPT_ERROR  113    //133
#define CSZPZIP_LIB_SELECTION_ERROR  112    //133
#define CSZPZIP_FILE_SELECTION_ERROR 111    //133
#define CSZPZIP_DELETE_ORILIBEL_ERROR 110
#define CSZPZIP_DELETE_ORIGFILE_DMS  109
#define CSZPZIP_DELETE_ORIGFILE_ERROR 108
#define CSZPZIP_LMS_ERROR              101
#define CSZPZIP_PARAMETER_ERROR        100
#define CSZPZIP_INCOHERENT_FORMAT      99
#define CSZPZIP_OPEN_ERROR              98
#define CSZPZIP_CONTAINER_EXISTS        97
#define CSZPZIP_NO_CONTAINER_EXISTS    96
#define CSZPZIP_LINKNAME                95
#define CSZPZIP_NO_FILE                 94
#define CSZPZIP_FILE_EXISTS             93
#define CSZPZIP_NO_FILE_EXISTS          92
#define CSZPZIP_INTERNAL_ERROR          91
#define CSZPZIP_RSVM                    90
#define CSZPZIP_EXTRACT_ERROR           89
#define CSZPZIP_CATALOG                 88
#define CSZPZIP_NO_CONTAINER_OPENED     86
#define CSZPZIP_READMODE                85
#define CSZPZIP_ALREADY_ZIPPED         84
#define CSZPZIP_ALREADY_ZIPPED_WLDC    83
#define CSZPZIP_ADD_ERROR               82
#define CSZPZIP_ILLEGAL_NEW_NAME        81
#define CSZPZIP_ILLEGAL_LINK           80
#define CSZPZIP_INVALID_RENAMING        79
#define CSZPZIP_IMPOSSIBLE_RENAMING     39
#define CSZPZIP_REORG_ERROR             78
#define CSZPZIP_DELETE_ERROR           77
#define CSZPZIP_INCON_FORMAT_ERROR      76
#define CSZPZIP_OUCON_DMS_ERROR         75
#define CSZPZIP_INCON_DMS_ERROR         74
#define CSZPZIP_OUCON_EXIST_ERROR       73
#define CSZPZIP_OUCON_NOEXIST_ERROR     72
#define CSZPZIP_INCON_NOEXIST_ERROR     71

```

```
#define CSZPZIP_OUCON_LINK_ERROR      70
#define CSZPZIP_INCON_LINK_ERROR     69
#define CSZPZIP_OUCON_FLINK_ERROR    68
#define CSZPZIP_INCON_FLINK_ERROR    67
#define CSZPZIP_INCON_OPEN_ERROR     66
#define CSZPZIP_PSWORDCHECK_ERROR    42
#define CSZPZIP_PSWORDNOTCOR_ERROR   41
#define CSZPZIP_PSWORDNOTGIV_ERROR   40
#define CSZPZIP_DATATYPE_BS2000     37
#define CSZPZIP_DMS_ERROR             31
#define CSZPZIP_INT_ERROR            19
#define CSZPZIP_STD_NAME              2
#define CSZPZIP_SHORT_STD_NAME       1
// Message id
#define MSG_K2                        "SZP0207"
#define MSG_LIB_EXCEPT_ERROR        "SZP0114" //133
#define MSG_FILE_EXCEPT_ERROR      "SZP0113" //133
#define MSG_LIB_SELECTION_ERROR      "SZP0112" //133
#define MSG_FILE_SELECTION_ERROR     "SZP0111" //133
#define MSG_DELETE_ORILEL_FILE      "SZP0110" //133
#define MSG_DELETE_ORIDMS_FILE      "SZP0109"
#define MSG_DELETE_ORIGIN_FILE      "SZP0108"
#define MSG_LMS_ERROR               "SZP0101"
#define MSG_PARAMETER_ERROR          "SZP0100"
#define MSG_INCOHERENT_FORMAT        "SZP0099"
#define MSG_OPEN_ERROR               "SZP0098"
#define MSG_CONTAINER_EXISTS         "SZP0097"
#define MSG_NO_CONTAINER_EXISTS      "SZP0096"
#define MSG_LINKNAME                 "SZP0095"
#define MSG_NO_FILE                  "SZP0094"
#define MSG_FILE_EXISTS              "SZP0093"
#define MSG_NO_FILE_EXISTS           "SZP0092"
#define MSG_INTERNAL_ERROR           "SZP0091"
#define MSG_WARNING1                 "SZP0090"
#define MSG_EXTRACT_ERROR            "SZP0089"
#define MSG_CATALOG                  "SZP0088"
#define MSG_NO_CONTAINER_OPENED      "SZP0086"
#define MSG_READMODE                 "SZP0085"
#define MSG_ALREADY_ZIPPED           "SZP0084"
#define MSG_ALREADY_ZIPPED_WLDC     "SZP0083"
#define MSG_ADD_ERROR                "SZP0082"
#define MSG_ILLEGAL_NEW_NAME        "SZP0081"
#define MSG_ILLEGAL_LINK             "SZP0080"
#define MSG_INVALID_RENAMING         "SZP0079"
#define MSG_IMPOSSIBLE_RENAMING     "SZP0039"
#define MSG_REORG_ERROR              "SZP0078"
#define MSG_DELETE_ERROR             "SZP0077"
#define MSG_INCON_FORMAT_ERROR       "SZP0076"
```

```
#define MSG_OUCON_DMS_ERROR      "SZP0075"  
#define MSG_INCON_DMS_ERROR     "SZP0074"  
#define MSG_OUCON_EXIST_ERROR   "SZP0073"  
#define MSG_OUCON_NOEXIST_ERROR "SZP0072"  
#define MSG_INCON_NOEXIST_ERROR "SZP0071"  
#define MSG_OUCON_LINK_ERROR    "SZP0070"  
#define MSG_INCON_LINK_ERROR    "SZP0069"  
#define MSG_OUCON_FLINK_ERROR   "SZP0068"  
#define MSG_INCON_FLINK_ERROR   "SZP0067"  
#define MSG_INCON_OPEN_ERROR    "SZP0066"  
#define MSG_PASSWORDCHECK_ERROR "SZP0042"  
#define MSG_PASSWORDNOTCOR_ERROR "SZP0041"  
#define MSG_PASSWORDNOTGIV_ERROR "SZP0040"  
#define MSG_DATATYPE_BS2000    "SZP0037"  
#define MSG_DMS_ERROR          "SZP0031"  
#define MSG_INT_ERROR          "SZP0019"  
/**
```

This class is the C++ api for managing a BS2ZIP container file.
This class is able to handle the K2 but not to intercept it.
**/

```
class CSzpZip {  
public:  
    /**  
    **/  
    CSzpZip();  
    /**  
    **/  
    CSzpZip(bool bOutputMsg);  
    /**  
    **/  
    virtual ~CSzpZip();  
    enum szpOpenMode {  
        read,  
        updateAny,  
        updateNew  
    };  
    enum szpFormat {  
        defaut,  
        compatible,  
        bs2000  
    };  
    enum szpConvertMode {  
        std,  
        none,  
        ascii,  
        ebcdic  
    };  
    enum szpWriteMode {
```

```
        create,
        replace,
        any
};
enum szpLink {
    no,
    yes
};
enum szpDataType {
    notSpecified,
    character,
    binary,
    sambinary
};
enum szpBlkCtrlInfo {
    keep,
    ignore
};
enum szpLevel {
    noCompression = 0,
    bestSpeed = 1,
    bestCompression = 9,
    defaultCompression = -1
};
enum szpInfo {
    infoNone,
    infoAll,
    infoSummary
};
// 132
enum szpDelete {
    DeleteOption_No,
    DeleteOption_Yes
};
enum szpToFileOption {
    TofileOptionVersStd_TypeStd = 0,
    TofileOptionVersStd_TypeYes = 1,
    TofileOptionVersStd_TypeNo = 2,
    TofileOptionVersYes_TypeStd = 10,
    TofileOptionVersYes_TypeYes = 11,
    TofileOptionVersYes_TypeNo = 12,
    TofileOptionVersNo_TypeStd = 20,
    TofileOptionVersNo_TypeYes = 21,
    TofileOptionVersNo_TypeNo = 22
};
/**
Open a zip container.
It returns 0 if ok.
```



```

char *pContainer = name of the zip container file
this file name must be valid
if "link=linkname", the container is opened
using this link name
int iOpenMode      = read, update, create
int iFormat        = bs2000 or compatible
**/
int OpenZip(char *pContainer, int iOpenMode, int iFormat);
/**
Close the zip container and release eventually file link.
It returns 0 if ok.
**/
int CloseZip();
/**
Add files in the zip container. The filename may contain wildcards.
By this way several files can be zipped in a single operation. Eventual
data conversion is required as well as the compression level.
Files are compressed one by one. If an error occurred, the data already
written in the container remain in it but the entry is not visible. If
several files have to be written, the processing goes on even in case of
error on a previous file. The same behavior is implemented if K2 is
    handled
during the processing of a file but if there are still files to be
    included,
those ones are not processed.
To keep the container coherent, the central header is rewritten after each
added files and then reread to rebuild the structure in memory.
It returns 0 if ok.
V01.0B: introduction of the parameter data-type
This operand is only significant for adding sam files in compatible
    format,
otherwise, it is ignored. In char mode, each record of a sam file is
interpreted as a line. In binary mode, data are interpreted as a simple
byte stream. Such a file is extracted as a pam std 16.
char *pFileNames = file name (incl. wild cards ev.)
int iConvertMode = convert mode
int iLevel       = compression level
int iDataType    = data type (not specified, character, binary)
char *pToFiles   = target construction file name
int iDeleteOption = source file delete option
**/
int AddFiles(char *pFileNames, int iConvertMode, int iLevel, int
    iDataType, char *pToFiles, int iDeleteOption);
/**
Add files in the zip container. Old format for program compatibility.
**/
int AddFiles(char *pFileNames, int iConvertMode, int iLevel, int
    iDataType, char *pToFiles);

```

132

```

/**
Add PLAM elements in the zip container. The element names, version, base
and type
may contain wildcards. Library name may not.
By this way several elements can be zipped in a single operation.
    Eventual
data conversion is required as well as the compression level.
Elements are compressed one by one. If an error occurred, the data
    already
written in the container remain in it but the entry is not visible. If
several elements have to be written, the processing goes on even in case
    of
error on a previous file. The same behavior is implemented if K2 is
    handled
during the processing of a file but if there are still files to be
    included,
those ones are not processed.
To keep the container coherent, the central header is rewritten after each
added elements and then reread to rebuild the structure in memory.
It returns 0 if ok.
V01.0B: introduction of the parameter data-type
This operand is only significant for adding sam files in compatible
    format,
otherwise, it is ignored. In char mode, each record of a sam file is
interpreted as a line. In binary mode, data are interpreted as a simple
byte stream. Such a file is extracted as a pam std 16.
char *pLibName = library name (excl. wild cards)
char *pElements = element name selector (incl. wild cards ev.)
char *pVersion = element version selector (incl. wild cards ev.)
char *pType = element type selector (incl. wild cards ev.)
char *pBase = version base selector
int iConvertMode = convert mode
int iLevel      = compression level
int iDataType   = data type (not specified, character, binary)
char *pToFiles  = target construction file name
int iTofileOption = specifies if libr element version and/or type are
    saved in the name
int iDeleteOption = library element delete option          132
**/
int AddLibElements(char *pLibName, char *pElements, char *pVersion,
    char *pType, char *pBase, int iConvertMode, int iLevel, int
    iDataType,
    char *pToFiles, int iTofileOption, int iDeleteOption);
/**
Add PLAM elements in the zip container. Old format for program
compatibility.
**/
int AddLibElements(char *pLibName, char *pElements, char *pVersion,

```

```

        char *pType,char *pBase, int iConvertMode, int iLevel, int
        iDataType,
        char *pToFiles, int iTofileOption);
/**
Extract one or several files selected by the pFileNames patterns into
BS2000 files named according the rule specified by the pToFiles pattern.
If an error occurred during the extract of a file, the extract of this
file is interrupted but the other files are processed.
If K2 is handled, the current file process is stopped, the output file
is deleted and the other extract files are not processed if any.
char *pFileNames = file pattern used to select the files that must
be extracted from the container
char *pToFiles   = pattern specifying the output name of the extracted
files
int iWriteMode   = new (default) or replace
int iDataType    = data type of the file (not specified (default), char or
binary)
int iConvertMode = convert mode (std (default), none, ascii or ebcdic)
int iBlkCtrlInfo = specifies if the original block control must be used
or not
**/
int ExtractFiles(char *pFileNames, char *pToFiles,
int iWriteMode = 0, int iDataType = 0, int iConvertMode = 0,
int iBlkCtrlInfo = 0);
/**
Returns the information about the zipped files selected by the pFileNames
pattern that may contain wild cards. Information summary or full is
possible.
The layout of the output is described in the header file szpzout.h.
Data are returned in a buffer requested by this method. Its address and
size
are returned to the caller who is in charge to release it after use.
The number of matching files is returned. -1 is returned in case of
error.
char *pFileNames = file selection pattern
char **pBuf      = address where the output buffer address is returned
int *iSize       = address of an int where the output buffer size is
returned
int iInfo        = information type selection (infoAll default,
infoSummary)
**/
int ListFiles(char *pFileNames, char **pBuf, int *iSize = 0, int iInfo =
infoAll, bool ListFirst = true);
/**
Delete files from a zip container. The filename may contain wildcards.
By this way several files can be deleted in a single operation.
If several files have to be deleted, the processing goes on even in case
of

```

error on a previous file. The same behavior is implemented if K2 is handled during the processing of a file but if there are still files to be deleted, those ones are not processed.

It returns 0 if ok.

```
char *pFileNames = file name (incl. wild cards ev.)
```

```
*/
```

```
int DeleteFiles(char *pFileNames);
```

```
/**
```

```
CONVERT *****
```

It returns 0 if ok.

```
char *pFromContainer = Output file name
```

```
char *pToContainer = Input file name
```

```
int iHuser = Home userid length
```

```
int iHcat = Home catid length
```

```
int iWriteMode = new (default) or replace
```

```
*/
```

```
int CnvZip(char *pFromContainer, char *pToContainer, int iHuser, int iHcat, int iWriteMode = 0);
```

```
/**
```

```
MODIFY-ZIP-OPTIONS *****
```

It returns 0 if ok.

```
char *pCrypto = crypto password
```

```
int iLen = length of crypto password
```

```
int iEncrypt = encryption or not
```

```
*/
```

```
int ModZipopt(char *pCrypto, int iLen, int iEncrypt);
```

```
/**
```

```
// reorganize the file by rewriting only the files having a header in the central directory
```

It returns 0 if ok.

```
char *pContainer = name of the zip container file
```

```
this file name must be valid
```

```
if "link=linkname", the container is opened
```

```
using this link name
```

```
*/
```

```
int ReorganizeZip(char *pContainer);
```

```
/**
```

Activate the trace processing.

It returns 0 if ok.

```
char *pFileName = name of the trace file. If NULL, default name is used
```

```
*/
```

```
int ActivateTrace(char *pFileName);
```

```

/**
Deactivate the trace
**/
int DeactivateTrace();
/**
Return the zip container comments into the input buffer with the specified
input size. If this size is not sufficient, the data are truncated.
char *Comments    = buffer address
int iCommentsSize = buffer size
**/
int GetComments(char *Comments, int iCommentsSize);
/**
Allows to set or reset that K2 has been intercepted at interface level.
bool b = indicates that K2 indicator must be set or reset
**/
void K2given(bool b);
/**
Activated if the 'Cpu Exhausted' event occurs.
**/
void CpuExhausted();
/**
Activated if the 'Term' event occurs.
**/
void Term();
/**
Global return code
**/
//void *pRcErr; // CRcErr m_RcErr;
//#define RcErr (*(CRcErr*)pRcErr)
/**
Return the last error maincode. Use for LIST problem when -1 is returned
**/
int GetLastError();
// private:
/**
Get the file name according to the link name. The file name
is returned in the ContainerName variable.
ReadTFT rc is returned.
char *pLink = link name
**/
long GetFileFromLink(char *pLink, char *pContainer);
/**
Check if the file exists.
It returns 1 if the file exists, -1 if it exists but is empty,
0 if it does not exist.
char *pFileName = file name
**/
int FileExists(char* pFileName);

```

```
/**
Build a file name for output: :catid:$uid.FILEyyyyymmdd.hhmmss
**/
void BuildFileName(char *fn, int i, char *catid=0, char *userid=0);
/**
Validate if the input file name is a valid file name
Return true is bs2000 file name compliant.
const char *fn = file name
**/
bool IsBS2000FileName(const char *fn);
/**
Not implemented
**/
static void fsRout(const char* fn, int i);
/**
Convert input string in upper case
char *s = string to convert
**/
void ToUpperCase(char *s);
/**
Catalog a container (PAM, STD16)
char *cont = container name
char *link = link name
int format = container format
bool space = true (define space in prg)
**/
int CatalogContainer(char *cont, char *link, int format, bool space);
/**
Get new file name using SDF wildcard construction
char *selection
char *construction
char *srcname
char *newname
**/
int GetNewFileName(char *selection, char *construction, char *srcname,
char *newname);
/**
Get new file name using SDF wildcard construction
char *selection
char *construction
void *src = fileitem object
char *newname
**/
int GetNewFileName(char *selection, char *construction, void *src, char
*newname);
/**
Common function adding files or lib elements in ZIP container
char *pFileNames = files or lib file name
```

```

char *pElements = null ptr or element selector
char *pVersion = null ptr or version selector
char *pType = null ptr or type selector
char *pBase = null ptr or base version base selector
int iConvertMode = convert mode
int iLevel       = compression level
int iDataType    = data type (not specified, character, binary)
char *pToFiles   = target construction file name
int iTofileOption = specifies if libr element version and/or type are
    saved in the name
int iDeleteOption = library element delete option           132
int iLogOption    = logging option                         133
long nbr = number of added files                           133
**/
int AddItems(char *pFileNames, char *pElements, char *pVersion, char
    *pType,
    char *pBase, int iConvertMode, int iLevel, int iDataType,
    char *pToFiles, int iTofileOption, int iDeleteOption, int iLogOption,
    unsigned long &nbr, void *flist);

/* methods added for selection enhancement 1.2G */

/* internal method
   pre-requisite CAddParam object has been created with all the parameters
   of the
   add statement
*/
int AddToZip(void *paddparam);

void SetExtractLogging(bool log);

char    ContainerName[55];
int     iCFormat;
void *pZip; //CZipArch m_Zip;
#define Zip (*(CZipArch*)pZip)
bool bRelLink;
int m_OpenMode;
bool k2Pressed;
void UpdateUserInfo();
bool bLib;
int m_TofileOption;
void *pAddParam; // CAddParam instance
bool bExtractLoggingMax; // 134
};
// extern bool glb_sysout; // msg on sysout by default
#endif // __Szip_h__

```

5.2 SZPZOUT.H

Diese C++-Headerdatei beschreibt den Aufbau des Puffers, der bei der SHOW-FILE-ATTRIBUTES-Operation zurückgegeben wird (Funktion *ListFiles()*). Der Puffer muss durch den Aufrufer freigegeben werden.

```
#ifndef __SzpZout_h__
#define __SzpZout_h__
/*****
 * Classes      --
 * File         szpzout.h
 *
 * Copyright    (c) 'FUJITSU TECHNOLOGY SOLUTIONS' '2009'
 *
 * Description   Output Find structure layouts
 *
 * Version      120
 * Date         2008-03-02
 * Author       Ph. Dumont - OSL EPS
 * Update       BS2ZIP v1.2A
 *
 * Version      110
 * Date         2004-09-30
 * Author       L. Tambour - OSL EPS
 * Update       BS2ZIP v1.1A
 *
 * Version      100
 * Date         2003-06-02
 * Author       J. Beaume - OSL EPS
 * Update       BS2ZIP v1.0B
 *****/
struct szpOutSummary {
    unsigned int uRecSize;        // total size of a returned item
    void *pHeader;               // pointer to CFileHdr object
    unsigned short uFileNameSize; // file name size
    unsigned short BS2Flag;      // 1 = BS2000 file, 0 = others
    unsigned short ENCFflag;     // 1 = Encrypt. file, 0 = others (V120)
    // File name
};
struct szpOutFull {
    unsigned int uRecSize;        // total size of a returned item
    void *pHeader;               // pointer to CFileHdr object
    unsigned short uFileNameSize; // file name size
    unsigned short iBS2Flag;     // 1 = BS2000 file, 0 = others
    unsigned short iENCFflag;    // 1 = Encrypt. file, 0 = others (V120)
    char uModTime[8];            // last mod file time
    char uModDate[10];           // last mod file date
};
```



```

//unsigned short uFiller;      // unused   V120
long long uComprSize;        // compressed size
long long uUncomprSize;     // uncompressed size
unsigned short uCommentSize; // file comment size
unsigned short uExtraFieldSize; // extra field length
unsigned int uFiller2;      // unused

// Filename
// Extrafield
// Comment
};
#endif // __SzpZout_h__

```

Beispiele für Ausgabe-Puffer

1. INFORMATION=*SUMMARY

szpOutSummary structure	File name 1	szpOutSummary structure	File name 2
SzpOutSummary structure	File name 3		

2. INFORMATION=*ALL

szpOutFull structure	File name 1	Extra Fields 1	Comments 1
szpOutFull structure	File name 3	Comments 2	

5.3 BS2ZIPPR LLM

Das Laufzeitmodul BS2ZIPPR muss mit der TU-Anwendung gebunden werden, welche BS2ZIP als Unterprogramm benutzen soll.

5.4 Programmbeispiel

```
#include "tstzip.h"
#include "SzpZout.h"
#include <string.h>
#include <stdio.h>

void main() {
    // create a CSzpZip object with error reported in rc and log file.
    CSzpZip *zip = new CSzpZip(false);

    // create a new container (BS2000 format by default)
    int rc = zip->OpenZip("MYCONT.ZIP", CSzpZip::updateNew, CSzpZip::default);

    // add a file to the container
    rc = zip->AddFiles("MYFILE.TXT", CSzpZip::std,
                     CSzpZip::defaultCompression);

    // list the contents of the container
    char *pBuf = 0;
    int iSize = 0;
    int iNumber = 0;
    // Get first element and loop while rc = 1;
    rc = zip->ListFiles("*", &pBuf, &iSize, CSzpZip::infoSummary);
    while(rc == 1) {
        iNumber++;

        // do something with buffer
        ...
        delete [] pBuf;

        rc = zip->ListFiles("*", &pBuf, &iSize, CSzpZip::infoSummary);
    }
    printf("Number of matching files = %d\n", iNumber);

    // extract file from the container - extracted file = EXT-MYFILE.TXT
    rc = zip->ExtractFiles("*", "EXT-*", CSzpZip::any, CSzpZip::notSpecified,
                          CSzpZip::std, CSzpZip::keep);

    // close zip container
    rc = zip->CloseZip();
    delete zip;
}
```

Literatur

Die Handbücher finden Sie im Internet unter <http://manuals.ts.fujitsu.com>. Handbücher, die mit einer Bestellnummer angezeigt werden, können Sie in auch gedruckter Form bestellen.

- [1] **SDF (BS2000)**
Dialogschnittstelle SDF
Benutzerhandbuch
- [2] **BS2000/OSD-BC**
Kommandos
Benutzerhandbuch
- [3] **JV (BS2000)**
Jobvariablen
Benutzerhandbuch
- [4] **BS2000/OSD-BC**
DMS Makroaufrufe
Benutzerhandbuch

Stichwörter

A

ADD-FILE [11](#), [23](#), [24](#)
Aktivieren eines Trace [56](#)
Änderungen in BS2ZIP V1.2A [7](#)
Anweisungen [21](#)
 ADD-FILE [11](#), [23](#), [24](#)
 Beschreibung [23](#)
 CONVERT-ZIP-CONTAINER [33](#)
 DELETE-FILE [23](#), [36](#)
 END [22](#), [23](#), [37](#)
 EXTRACT-FILE [12](#), [23](#), [38](#)
 MODIFY-ZIP-OPTIONS [23](#), [44](#)
 OPEN-ZIP-CONTAINER [23](#), [46](#)
 REORGANIZE-ZIP-CONTAINER [23](#), [49](#)
 SHOW-FILE-ATTRIBUTES [23](#), [50](#)
 START-TRACE [23](#), [56](#)
 START-ZIP-MANAGER [21](#)
 STOP-TRACE [23](#), [57](#)
Anwendung BS2ZIP [9](#)
Archiv (siehe ZIP-Archiv)
Archivformat [9](#), [29](#)
Ausgabe-Puffer, Beispiel [81](#)

B

Beenden von BS2ZIP [21](#), [37](#)
Beispiel
 Ausgabe-Puffer [81](#)
 BS2000 [58](#)
 Programm [82](#)
Bibliothekselement
 Erstellungsdatum [27](#)
Blockformat [13](#), [14](#)
Blockfüllungsfaktor [16](#)
Blockkontrollinformation [54](#)
BS2ZIP [9](#)

Beenden der Anwendung [21](#), [37](#)
Beispiel [58](#), [82](#)
BS2ZIPPR (Laufzeitmodul) [81](#)
Header-Datei SZPZIP.H [67](#)
Header-Datei SZPZOUT.H [80](#)
Starten der Anwendung [9](#), [21](#)
Trace aktivieren [56](#)
Trace deaktivieren [57](#)
BS2ZIPPR Laufzeitmodul [81](#)

C

CONVERT-ZIP-CONTAINER [33](#)

D

Datei
 Größe (komprimiert) [55](#)
 Originalgröße [55](#)
 Pfadname [54](#)
 Satzformat [55](#)
 Satzlänge [55](#)
 Struktur [54](#)
 Typ [54](#)
 Verschlüsselung [54](#)
Dateien
 Archiv übertragen [64](#)
 Extrahieren [38](#)
 Hinzufügen [24](#)
 Liste ausgeben [50](#)
 Löschen [36](#)
Dateiformat ändern [33](#)
Dateitypen [10](#)
Dateizugriffsmethoden [10](#)
Datenkomprimierung [9](#), [31](#)
Deaktivieren eines Trace [57](#)
DELETE-FILE [23](#), [36](#)

E

END [22](#), [23](#), [37](#)
Erstellen eines Archivs [9](#)
EXTRACT-FILE [12](#), [23](#), [38](#)
Extrahieren
 auf K-Platte [13](#)
 auf NK2-Platte [13](#)
 auf NK4-Platte [14](#)
Extrahieren einer Datei [38](#)

G

GZIP-Archive [65](#)

H

Header-Datei
 SZPZIP.H [67](#)
 SZPZOUT.H [80](#)
Hinzufügen einer Datei [24](#)

I

Interoperabilität [63](#)
 mit Linux [65](#)
 mit Unix-Systemen [65](#)
 mit Windows [63](#)
ISAM-Schlüssel [54](#)

K

Kennwort, Kontrolleingabe [45](#)
Kommandos (siehe Anweisungen)
Kompatibilität zu WinZip [10](#)
Komprimierung [31](#)
Komprimierungsrate [55](#)
Konverter SAM/PAM [64](#)

L

Liste, Dateien [50](#)
Löschen einer Datei [36](#)

M

MODIFY-ZIP-OPTIONS [23](#), [44](#)

O

Öffnen eines Archivs [9](#), [46](#)
OPEN-ZIP-CONTAINER [23](#), [46](#)
openFT [64](#)

P

Performance-Eigenschaft [54](#)
Plattenformat [13](#)
Programmbeispiel [82](#)
Programmschnittstelle [67](#)
Puffergröße [54](#)

R

Readme-Datei [7](#)
Reorganisieren eines Archivs [49](#)
REORGANIZE-ZIP-CONTAINER [23](#), [49](#)

S

SAM/PAM-Konverter [64](#)
Satzformat einer Datei [55](#)
Satzlänge einer Datei [55](#)
Schließen eines Archivs [37](#)
Schnittstelle, Programm [67](#)
Schnittstelle, SDF [21](#)
SDF-Schnittstelle [21](#)
Sekundärzuweisung [55](#)
SHOW-FILE-ATTRIBUTES [23](#), [50](#)
START-SAM-PAM-CONVERTER
 (Kommando) [64](#)
START-TRACE [23](#), [56](#)
START-ZIP-MANAGER [21](#)
Starten von BS2ZIP [9](#), [21](#)
STOP-TRACE [23](#), [57](#)
SZPZIP.H (Header-Datei) [67](#)
SZPZOUT.H (Header-Datei) [80](#)

T

temporäre Dateien [17](#)
Trace
 aktivieren [56](#)
 deaktivieren [57](#)

U

Übertragen von ZIP-Dateien [64](#)
Unterstützung
 ISAM-Dateien [16](#)
 PLAM-Bibliotheken [17](#)
 temporärer Dateien [17](#)

V

Verschlüsselung 54
einstellen 44

W

Wertmarkierung 55
WinZip-kompatibel 10

Z

Zip 2.0-Verschlüsselung 16
ZIP-Archiv
Dateien extrahieren 38
Dateien hinzufügen 24
Dateien löschen 36
erstellen 9
konvertieren 33
Liste der Dateien ausgeben 50
öffnen 9, 46
reorganisieren 49
schließen 37
übertragen 64
ZIP-Optionen, Verschlüsselung einstellen 44
ZIP, Datenkomprimierung 9, 31
Zugriffsmethoden, Dateien 10

