

**SANRISE2000/SANRISE9900/
SANRISE Universal Storage Platform/
Hitachi Universal Storage Platform V/
Hitachi Virtual Storage Platform/
Hitachi Virtual Storage Platform G1000/G1500/F1500
Hitachi Virtual Storage Platform 5000**

**FAL for AIX
FAL for HP-UX
FAL for Solaris
FAL for Windows
FAL for NCR UNIX
FAL for HI-UX/WE2
FAL for Tru64 UNIX
FAL for LINUX**

ユーザーズガイド

株式会社 日立製作所

ご注意

- ① 本書の内容の一部または全部を無断で使用、複製することはできません。
- ② 本書の内容については、将来予告なしに変更することがあります。
- ③ 本書を運用した結果の影響については、一切責任を負いかねますのでご了承下さい。
- ④ 本書の内容については、万全を期していますが、ご不審な点や誤り等、お気づきのことがありましたら、弊社までご連絡下さい。

- * UNIXは X/Open Company Limited がライセンスしている米国ならびに他の国における登録商標です。
- * IBMは米国International Business Machines Corporationの登録商標です。
- * MVSは米国International Business Machines Corporationの登録商標です。
- * RISC System/6000は米国International Business Machines Corporationの商標です。
- * POWERstation, POWERserverは米国International Business Machines Corporationの商標です。
- * AIXは米国International Business Machines Corporationの商標です。
- * HACMPは米国International Business Machines Corporationの商標です。
- * HP-UX, HPは米国Hewlett-Packard Corporationの登録商標です。
- * SPARCstation, SPARCserver, SPARCcenterは米国SPARC International, Inc の商標です。
- * Solarisは米国Sun Microsystems, Inc の商標です。
- * WindowsNT/2000Server/Server2003/Server2008/Server2008 R2/Server2012/Server2012 R2/Server2016/Server2019は米国Microsoft Corporationの登録商標です。
- * MSCSは米国Microsoft Corporationの登録商標です。
- * DEC, Digital, Tru64は米国Digital Equipment Corporationの商標です。
- * NCRは米国NCR Corporationの商標です。
- * MSP/XSPは富士通株式会社の商標です。
- * DYNIX/ptx, NUMA-Q, Symmetryは米国Sequent Corporationの登録商標です。
- * LinuxはLinus Torvaldsの米国及びその他の国における登録商標または商標です。
- * VMware は、米国およびその他の地域におけるVMware, Inc. の登録商標または商標です。

その他このマニュアルに記載の固有名詞の中には、製品名など、商標登録されているものがいくつかありますが、本ページでの掲載を省略させていただきます。

なお、本文中では、**®**および[™]は明記しておりません。

2021年8月（第46.8版）

All right reserved, Copyright © 1997, 2021, Hitachi, Ltd.

(注)

本プログラムは各ディスクレイ装置によりプログラム名称が異なります。下記の各ディスクレイ装置に対するプログラム名称の対応表を参照しプログラム名称を読み替えて下さい。

ディスク装置	プログラム名称	
SANRISE2000 シリーズ	FAL	HMDE/KANJI
SANRISE9900 シリーズ	RapidXchange	Code Converter
SANRISE Universal Storage Platform シリーズ (略称:SANRISE USP シリーズ)	File Exchange	
Hitachi Universal Storage Platform V シリーズ (略称: USP V シリーズ)		
Hitachi Virtual Storage Platform シリーズ (略称 VSP シリーズ)		
Hitachi Virtual Storage Platform G1000/G1500 シリーズ (略称 VSP G1000/1500 シリーズ)		
Hitachi Virtual Storage Platform 5000 シリーズ (略称 VSP 5000 シリーズ)		

このマニュアルに記載されているKB(キロバイト)などの単位表記は、以下となります。

1KB or 1Kbyte(キロバイト)、1MB or 1Mbyte(メガバイト)、1GB or 1Gbyte(ギガバイト)、1TB or 1Tbyte(テラバイト)は、それぞれ、1KiB(キビバイト)、1MiB(メガバイト)、1GiB(ギビバイト)、1TiB(テビバイト)と読み替えてください。

1KiB、1MiB、1GiB、1TiBは、それぞれ1,024バイト、1024KiB、1024MiB、1024Gibです。

目 次

1 概要.....	11
1.1 各プログラム機能.....	11
1.1.1 メインフレームーオープンシステム間のデータ交換.....	11
1.1.2 オープンシステムーオープンシステム間データ交換.....	12
1.2 システム構成.....	13
1.2.1 FAL/FCUプログラムの位置付け.....	13
1.2.2 サポート対象.....	16
2 オープンシステムーオープンシステム間のデータ交換時の環境構築.....	33
2.1 環境設定.....	33
2.2 RAWデバイス定義.....	33
2.3 中間ボリュームの生成.....	34
2.3.1 UNIX系OSの場合.....	38
2.3.2 Windowsの場合.....	39
2.4 中間ファイルの生成.....	41
2.4.1 UNIX系OSの場合.....	42
2.4.2 Windowsの場合.....	44
2.5 フォーマット/アロケータのエラーメッセージ.....	47
2.5.1 UNIX系OSの場合.....	47
2.5.2 Windowsの場合.....	48
3 FALのインストール.....	49
3.1 FAL/FCUメモリ使用量.....	49
3.2 FALのインストール.....	49
3.2.1 32bit用FAL/FCUのインストール/アンインストール.....	49
3.2.2 64bit用FAL/FCUのインストール/アンインストール.....	54
3.2.3 ライセンスキーコードについて.....	56
3.2.4 ライセンスキーコードの設定方法.....	57
3.2.5 バージョン確認方法.....	60
3.3 HMDE/KANJI連携時のバージョンアップ/ダウンの注意事項.....	60
4 FALの機能.....	61
4.1 データセットのオープン.....	62
4.2 データの読出し.....	63
4.3 データの書込み.....	64
4.4 データセットのクローズ.....	65
4.5 エラー情報の取得.....	66
4.6 データセット属性情報の取得.....	67
4.7 エラー情報.....	72
4.8 マルチボリュームデータセットによるデータ転送.....	75
4.9 AIX共用オープン機能.....	78
4.10 FALリザーブトライ機能.....	79
5 FALの使用手法.....	83

5.1 環境設定.....	83
5.2 関数の使用例.....	86
6 FALマルチスロット [®] の機能と使用方法.....	88
6.1 仕様.....	88
6.2 コンパイル方法.....	100
6.3 エラー情報.....	100
6.4 関数の使用例.....	101
7 FCUの機能.....	103
7.1 ファイル変換機能.....	103
7.2 パイプ出力機能.....	104
7.3 コード変換機能.....	105
7.4 パディング機能.....	107
7.5 テリミタ付加機能.....	108
7.6 レコード長付加機能.....	110
7.7 メインフレーム-オープンシステム双方向データ転送を行う場合の注意事項.....	110
7.8 ログ強化機能.....	111
7.9 競合時に発生する各OSのシステムエラー.....	115
7.10 FCUリザブリティ機能.....	116
7.11 OTM終了時インターバル機能(Linuxのみ).....	122
8 FCUの使用方法.....	124
8.1 環境設定 (共用ボリュームの定義)	124
8.2 ボリューム定義ファイルの作成.....	126
8.2.1 メインフレーム-オープンシステム間のデータ交換の場合.....	126
8.2.2 オープンシステム-オープンシステム間のデータ交換の場合.....	135
8.3 マルチボリューム定義ファイル.....	136
8.4 パラメタ定義ファイルの作成.....	137
8.5 起動方法.....	144
8.5.1 UNIX系OSの場合.....	144
8.5.2 Windowsの場合.....	145
8.6 画面操作.....	147
8.6.1 UNIX系OSの場合.....	147
8.6.2 Windows の場合.....	157
8.7 操作画面以外の操作及び情報の表示.....	208
8.7.1 UNIX系OSの場合.....	208
8.7.2 Windowsの場合.....	211
8.8 エラー情報.....	214
8.8.1 UNIX系OSの場合.....	214
8.8.2 Windows の場合.....	226
9 使用上の注意事項.....	233
10 トラブル発生時の情報収集.....	239
付録.....	241

はじめに

File Exchange function はメインフレームーオープンシステム間データ交換、オープンシステムーオープンシステム間データ交換を実現するための機能です。

第1章では、メインフレームーオープンシステム間データ交換とオープンシステムーオープンシステム間データ交換とを分けて、それぞれの概要を説明しています。

第2章では、オープンシステムーオープンシステム間データ交換時における固有な操作について説明しています。

第3章以降の FAL/FCU の機能については、メインフレームーオープンシステム間データ交換、オープンシステムーオープンシステム間データ交換では、ほとんど違いがありませんので、メインフレームーオープンシステム間データ交換時を元に説明しています。一部、操作方法などが異なる点については、その部分に注意書きをしています。

第3章以降、オープンシステムーオープンシステム間データ交換時の操作を行いたい場合は、「メインフレーム」、「データセット」という用語を「中間ボリューム」、「中間ファイル」と読み替えて下さい。

尚、本ユーザーガイドの対象プログラムバージョンは以下となります。

SANRISE2000 シリーズ	:	01-02-46 以降
SANRISE9900 シリーズ	:	01-03-53 以降
SANRISE USP シリーズ	:	01-04-61 以降
USP V シリーズ	:	01-05-66 以降
VSP シリーズ	:	01-06-67 以降
VSP G1000/1500 シリーズ	:	01-07-68 以降
VSP 5000 シリーズ	:	01-08-69 以降

※RADI300/200 シリーズについては、01-01-57 以前の FAL/FCU に同梱されるユーザーガイドをご参照下さい。

これ以降のシステムは下記機能を追加しています。追加機能については追加前のバージョンでは未支援ですので注意願います。

No	バージョン	追加機能	備考
1	01-XX-47/00 (UNIX) 01-XX-47/01 (NT) (XX ≥ 02)	ライセンスコード 3/6/9TB 容量帯ワウンドリ方式に加え、1/5/10TB ワウンドリ方式サポート	FAL for Solaris HP-UX AIX WindowsNT Digital UNIX
2	01-XX-48/00	① LUSE/OPEN-E/OPEN-M サポート (ただし、Large ファイル (2GB を超えるファイル/5818 シリンダを超える中間ボリューム) アクセス不可) OPEN-M は Windows2000 のみサポート ②同一 VSN 支援 ③同一ボリューム内、異データセットへの競合可	FAL for Digital UNIX
3	01-XX-48/20 01-XX-48/21	① Large ファイル (2GB を超えるファイル/5818 シリンダを超える中間ボリューム) アクセス可 ②Solaris8 サポート (FAL for Solaris のみ)	FAL for Solaris HP-UX AIX WindowsNT
4	01-XX-49/00	①Tru64 UNIX 5.0 (SCSI のみ) ②製品名変更 Digital → Tru64	FAL for Tru64 UNIX

No	バージョン	追加機能	備考
5	01-XX-49/21	KANJI 連携時 UNICODE サポート	FAL for WindowsNT KANJI 01-05 以降要
6	01-XX-49/00	本バージョン履歴の No.2 と同機能をサポート	FAL for HI-UX/WE2
	01-XX-49/12	2105 DKC エミュレーションサポート	NCR UNIX
	01-XX-49/13	ライセンスキーファイル入力	FAL for HI-UX
7	01-XX-50/20	マルチボリューム (MTO のみ) 機能サポート	FAL for Solaris
	01-XX-50/21	Tru64 UNIX 5.0 Fibre サポート	HP-UX AIX WindowsNT Tru64 UNIX
8	01-XX-51/20	OPEN-L/M サポート	FAL for Solaris
	01-XX-51/21 (XX ≥ 02)	AIX5L サポート	HP-UX AIX WindowsNT
9	01-XX-52/21 (XX ≥ 02)	GUI 画面非表示 FCU サポート	FAL for WindowsNT
10	01-XX-53/20	SANRISE9900 Series サポート	
	01-XX-53/21 (XX ≥ 03)		
11	01-XX-54/20 (XX ≥ 03)	SANRISE9900 Series 64bitFAL サポート	FAL for Solaris HP-UX AIX
12	01-XX-55/20	2105 DKC エミュレーションサポート	FAL for Solaris
	01-XX-55/21	HP-UX 11i サポート PIPE タイムアウト (UNIX 系 OS) のデフォルト 10 分に変更 64bitFAL AIX V5.x サポート (XX ≥ 03)	HP-UX AIX WindowsNT Tru64 UNIX
13	01-XX-56/20	OPEN-V サポート	FAL for Solaris
	01-XX-56/21 (XX ≥ 02)	ライセンスキーファイル入力 Solaris9 (32bit) サポート	HP-UX AIX WindowsNT
14	01-XX-57/20	(I)/(H)シェア、(I)/(F)シェア、(H)/(F)シェア、	FAL for Solaris
	01-XX-57/21 (XX ≥ 02)	MVS ICKDSF (R17) サポート	HP-UX AIX WindowsNT Tru64 UNIX
15	01-XX-58/20	RedHat Linux 7.2 及び Windows2003 サポート	FAL for Linux
	01-XX-58/21 (XX ≥ 03)	RedHat AS2.1 サポート RedHat AS3.0 サポート	WindowsNT
16	01-XX-59/20	VSE2.4 サポート	FAL for Solaris
	01-XX-59/21 (XX ≥ 03)	マルチスロット サポート (AIX 限定) FCU64bit (Solaris, AIX, HP-UX 限定)	HP-UX AIX WindowsNT Tru64 UNIX Linux
17	01-XX-60/20	DKU エミュレーション 3390-9, L サポート	FAL for Solaris
	01-XX-60/21 (XX ≥ 03)	マルチボリュームテープレット OTM サポート AIX 中間ボリューム共有サポート (AIX 限定)	HP-UX AIX WindowsNT Tru64 UNIX Linux

No	バージョン	追加機能	備考
18	01-XX-61/20 01-XX-61/21 (XX \geq 04)	SANRISE Universal Storage Platform シリーズ (略称: SANRISE USP) サポート 3380-3 サポート	FAL for Solaris HP-UX AIX Windows Tru64 UNIX Linux
19	01-XX-62/20 (XX \geq 03)	Linux AS3.0 IA64 サポート AIX リザーブ/リトライサポート	FAL for Linux(64bit) AIX
20	01-XX-63/20 01-XX-63/21 (XX \geq 03)	リザーブ/リトライ機能サポート ログ強化機能サポート HP-UX(64bit) IA64 サポート Linux(64bit) AS2.1 IA64 サポート DKU エミュレーション 6588-9, L サポート	FAL for Solaris HP-UX AIX Windows Tru64 UNIX Linux
21	01-XX-64/20 01-XX-64/21 (XX \geq 03)	DKU エミュレーション 3390-M サポート	FAL for HP-UX AIX Windows Tru64 UNIX Linux
	01-XX-64/20 (XX \geq 03)	Linux(32bit) SLES9 x86, AMD64/EM64T サポート	FAL for Linux
22	01-XX-64/22 (XX \geq 03)	HP-UX リザーブ/リトライ機能サポート	FAL for HP-UX
23	01-XX-64/23 (XX \geq 03)	Windows リザーブ/リトライ機能サポート Windows2003 x64 Edition サポート	FAL for Windows
24	01-XX-64/24 (XX \geq 03)	Linux(64bit) SLES9 IA64 サポート	FAL for Linux
25	01-XX-64/25 (XX \geq 03)	HP-UX リザーブ/リトライ機能サポート 2	FAL for HP-UX
26	01-XX-64/26 (XX \geq 03)	solaris10(32bit/64bit) SPARC サポート	FAL for solaris
27	01-XX-65/20 01-XX-65/21 (XX \geq 03)	FCU リザーブ/リトライ機能	FAL for Solaris HP-UX AIX Windows Tru64 UNIX Linux
28	01-XX-66/20 01-XX-66/21 (XX \geq 05)	USP V サポート マルチボリュームデータセット 61 ボリュームサポート HP-UX(32bit) Itanium 版サポート WindowsServer2003(64bit) Itanium/x64 サポート Windows 版 listvol サポート OracleVM Solaris10(32bit/64bit) SPARC サポート	FAL for Solaris HP-UX AIX Windows Tru64 UNIX Linux
29	01-XX-65/26 (XX \geq 03)	OTM 終了時インターバル機能	FAL for Linux

No	バージョン	追加機能	備考
30	01-XX-65/25 (XX \geq 03) 01-XX-66/25 (XX \geq 05) 01-XX-66/26 (XX \geq 05)	AIX6.1 サポート Linux(32bit) AS4.0 U5(Update 5) AS5.1 x86,AMD64/Intel64※サポート Linux(64bit) AS4.0 U5(Update 5) AS5.1 IA64 サポート ※ (旧名称 EM64T)	FAL for AIX FAL for Linux
31	01-XX-66/24 (XX \geq 05)	WindowsServer2008(32bit) x86,AMD64/Intel64 サポート WindowsServer2008(64bit) Itanium,AMD64/Intel64 サポート	FAL for Windows
32	01-XX-66/26 (XX \geq 05)	Linux(32bit) AS5.2 x86,AMD64/Intel64 サポート Linux(64bit) AS5.2 IA64 サポート	FAL for Linux
33	01-XX-66/26 (XX \geq 05)	Linux(32bit) AS5.3 x86,AMD64/Intel64 サポート Linux(64bit) AS5.3 IA64 サポート	FAL for Linux
34	01-XX-66/24 (XX \geq 05)	WindowsServer2008 R2 Itanium,AMD64/Intel64 サポート	FAL for Windows
35	01-XX-66/26 (XX \geq 05)	Linux(32bit) SLES10 x86,AMD64/Intel64 サポート Linux(64bit) SLES10 IA64 サポート Linux(32bit) SLES11 x86,AMD64/Intel64 サポート Linux(64bit) SLES11 IA64 サポート	FAL for Linux
36	01-XX-66/26 (XX \geq 05)	Linux(32bit) AS5.4 x86,AMD64/Intel64 サポート Linux(64bit) AS5.4 IA64 サポート	FAL for Linux
37	01-XX-67/21 01-XX-67/22 (XX \geq 06)	VSP サポート	FAL for Solaris HP-UX AIX Windows Linux
38	01-XX-67/23	2 バイトデリミタサポート (UNIX)	FAL for Solaris HP-UX AIX Linux
39	01-XX-68/00	VSP G1000 シリーズ サポート	FAL for Solaris HP-UX AIX Windows Linux

No	バージョン	追加機能	備考
40	01-XX-66/26 (XX \geq 05) 01-XX-67/24 (XX \geq 06) 01-XX-68/01 (XX \geq 07)	Windows Server2012 サポート Windows Server2012 R2 サポート	FAL for Windows
41	01-XX-66/27 (XX \geq 05) 01-XX-67/26 (XX \geq 06) 01-XX-68/01 (XX \geq 07)	AIX(32bit) AIX7.1/7.2 サポート AIX(64bit) AIX7.1/7.2 サポート VMware RHEL6.5 サポート	FAL for AIX Linux
42	01-XX-67/23 (XX \geq 06) 01-XX-68/00 (XX \geq 07)	RHEL7.1/7.2 サポート	FAL for Linux
43	01-XX-67/26 (XX \geq 06) 01-XX-68/01 (XX \geq 07)	AIX(32bit)7.2+PowerHA+VIOS2.2.6.23 サポート AIX(64bit)7.2+PowerHA+VIOS2.2.6.23 サポート	FAL for AIX
44	01-XX-67/23 (XX \geq 06) 01-XX-68/00 (XX \geq 07) 01-XX-67/26 (XX \geq 06) 01-XX-68/01 (XX \geq 07)	RHEL6.9/7.4/7.5 サポート FBA/FBM/VBA/VBM サポート Solaris11.3 サポート	FAL for Linux FAL for Solaris
45	01-XX-67/23 (XX \geq 06) 01-XX-68/00 (XX \geq 07) 01-XX-67/26 (XX \geq 06) 01-XX-68/01 (XX \geq 07)	RHEL6.7/7.6 サポート Windows2016 サポート	FAL for Linux FAL for Windows
46	01-XX-69/00 (XX=08)	VSP 5000 シリーズ サポート	FAL for Solaris HP-UX AIX Windows Linux
47	01-XX-67/23 (XX \geq 06) 01-XX-68/00 (XX \geq 07) 01-XX-69/00 (XX=08) 01-XX-68/01 (XX \geq 07) 01-XX-69/00 (XX=08)	RHEL7.7/8.0 サポート Windows2019 サポート	FAL for Linux FAL for Windows

No	バージョン	追加機能	備考
48	01-XX-67/23 (XX \geq 06) 01-XX-68/00 (XX \geq 07) 01-XX-69/00 (XX=08)	RHEL8.1 サポート	FAL for Linux
49	01-XX-67/23 (XX \geq 06) 01-XX-68/00 (XX \geq 07) 01-XX-69/00 (XX=08)	VMware ESXi5.5, ESXi6.7+RHEL6.5 サポート	FAL for Linux
50	01-XX-69/00 (XX=08)	VMware ESXi7.0+Windows2016 サポート VMware ESXi7.0+Windows2019 サポート	FAL for Windows
51	01-XX-69/00 (XX=08)	VSP 5200/5600 サポート	FAL for Solaris HP-UX AIX Windows Linux

注：(1) FAL/FCUにおけるバージョンの意味づけ 01-XX-YY/ZZ

XX: 01=RAID300/200, 02=SANRISE2000 シリーズ, 03=SANRISE9900 シリーズ

04=SANRISE USP シリーズ, 05=USP V シリーズ, 06=VSP シリーズ, 07=VSP G1000 シリーズ,
08=VSP 5000 シリーズ

YY: バージョン

ZZ: レビジョン(各プラットフォームごとの支援/未支援の差異を示す)

1 概要

Hitachi File Exchange function はメインフレームーオープンシステム間データ交換、オープンシステムーオープンシステム間データ交換を実現するための機能です。

1.1 各プログラムの機能

本機能は、以下のプログラムにより実現されます。

1.1.1 メインフレームーオープンシステム間のデータ交換

(1) ファイルアクセスライブラリ (FAL)

本機能の中で、FALは、メインフレームとオープンシステムとの間でファイル・データの共用を実現するための手段として、マルチプラットフォーム機能を持つ大型ディスクアレイ装置 (A-6592/H-6592、A-6593/H-6593、A-65A1/H-65A1型ディスク制御装置及びディスク装置) 上にメインフレーム側で作成したファイルのデータをオープンシステム側から読出し/書き込みを行う機能を提供するライブラリです。

本ライブラリには以下の特徴があります。

- (A) AP (アプリケーションプログラム) は、メインフレームデータ特有の「CKD」フォーマット等を意識することなくデータの読出し/書き込みを行うことができます。
- (B) メインフレームのデータを直接読み出す/書き込むため、ネットワークを経由する転送方式に比較して高速なデータ転送ができます。
- (C) 本ライブラリの機能は、C言語のライブラリ (関数) として提供するので、C言語で記述するAPの中から容易に使用できます。
- (D) 32bit用ライブラリと64bit用ライブラリがあります。

(2) ファイル変換ユーティリティ (FCU)

本機能の中で、FCU は、マルチプラットフォーム機能を持つ大型ディスクアレイ装置上にメインフレーム側で作成したデータセットをオープンシステム側のファイルに変換する機能、及びオープンシステムで作成したファイルをメインフレーム側のデータセットに変換する機能を提供するユーティリティプログラムです。

FCUには以下の特徴があります。

- (A) FAL (32bit) を経由してメインフレームデータを直接読み出す/書き込むため、ネットワークを経由する転送方式に比較してメインフレームーオープンシステム間の高速なデータ交換ができます。
- (B) 必要に応じて、データ転送時にデータ実体のコード変換 (例: EBCDIC ⇒ ASCII) を指示することができます。
- (C) パイプ出力機能を使用することにより、メインフレームデータを直接OS上のアプリケーションプログラム/ユーティリティプログラムへ転送することができます。 (UNIX系OSの場合)

(3) 補助機能

FAL/FCUを補助する機能として、以下があります。

(A) LISTVOL (UNIX系OSのみ)

LISTVOL/LISTVOL64は指定されたボリューム上にあるデータセット名の一覧を出力します。

* : Windows版では、FCUのMF-Information機能によりデータセット名の一覧を参照出来ます。

1.1.2 オープンシステムーオープンシステム間データ交換

(1) フォーマッタ

ファイルアクセスライブラリ (FAL) /ファイル変換ユーティリティ (FCU) が中間ファイルを作成するために使用する中間ボリュームをフォーマットするプログラムです。

中間ボリュームは、マルチプラットフォーム機能を持つ大型ディスクレイ装置が提供するデバイスエミュレーションタイプ = OPEN-3/OPEN-8/OPEN-9/OPEN-K/OPEN-E/OPEN-L/OPEN-M/OPEN-Vのボリュームを使用します。

(2) アロケータ

フォーマッタがフォーマットした中間ボリュームに、FAL/FCUが使用する中間ファイルを生成(ファイルエリアの割付、ファイル属性の定義等)を行うプログラムです。

(3) ファイルアクセスライブラリ (FAL)

異機種オープンシステムとオープンシステムとの間でデータの交換を実現する為の手段として、オープンシステムが提供するファイルシステムとは独立のファイルシステムを中間ボリューム上に構築し、データの読出し/書込みを行う機能を提供するライブラリです。オープンシステムの機種に依存しないファイルシステムを使用することにより、異機種間のデータ交換を実現しています。

本プログラムには以下の特徴があります。

- (A) ファイルを経由するデータ転送であるため、ネットワークを経由する転送方式に比較して高速なデータ転送が期待できます。
- (B) 本ライブラリの機能は、C言語のライブラリ(関数)として提供するので、C言語で記述するAPの中から容易に使用できます。
- (C) 32bit用ライブラリと64bit用ライブラリがあります。

(4) ファイル変換ユーティリティ (FCU)

中間ファイルとオープンシステムのファイルとの間のデータ転送を行うユーティリティプログラムです。

中間ファイルへのアクセスはFAL(32bit)を使用し、オープンシステムのファイルへのアクセスはOSが提供している通常のファイルアクセス機能を使用します。

本プログラムには以下の特徴があります。

- (A) GUIを使用した簡便な操作により、本プログラムが提供する機能を使用することができます。

(5) 補助機能

FAL/FCUを補助する機能として、以下があります。

(A) LISTVOL (UNIX系OS, Windows系OS※)

LISTVOL/LISTVOL64は指定されたボリューム上にあるデータセット名の一覧を出力します。

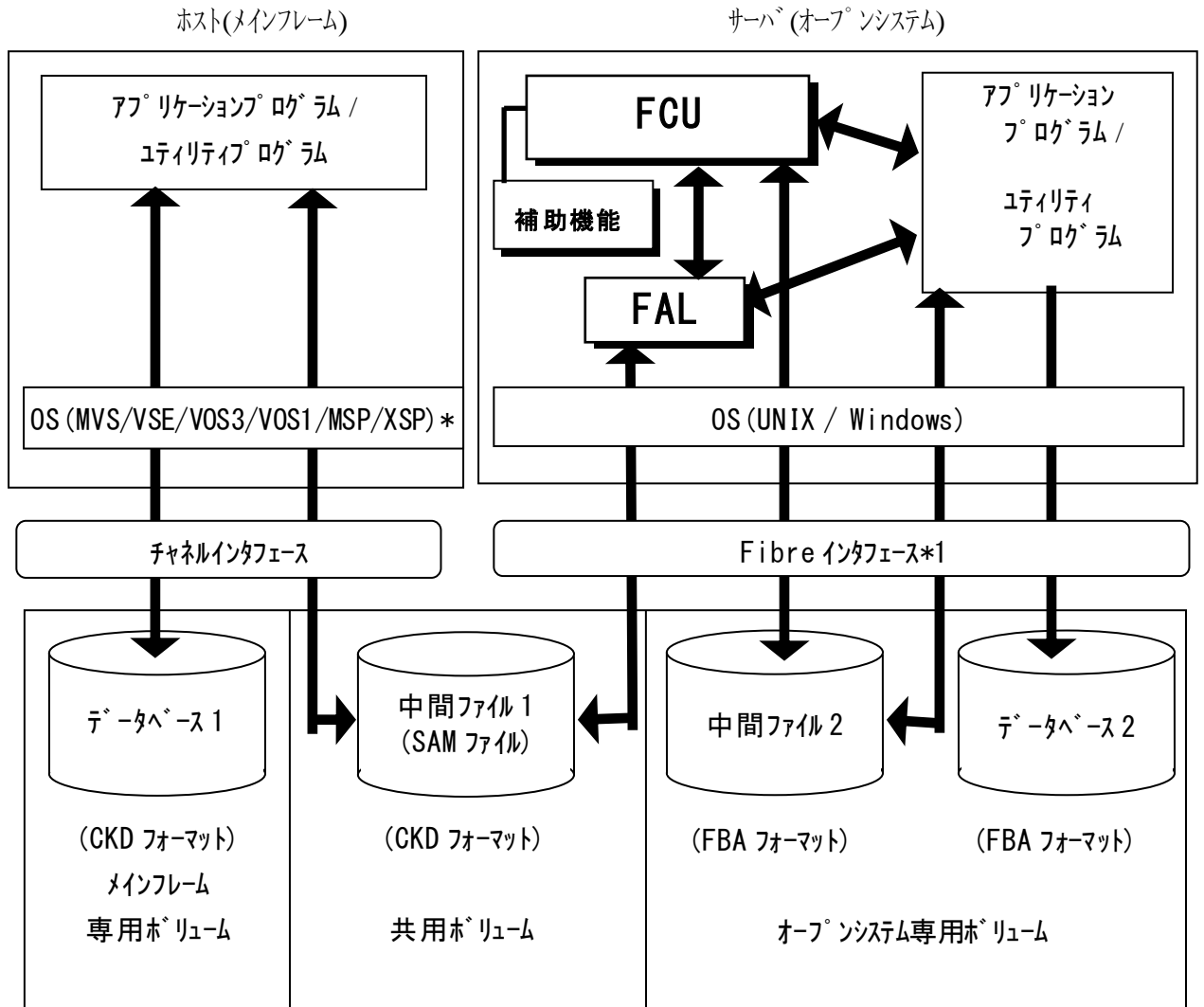
※: Windows版では01-XX-66以降 (XX≥05) でLISTVOL/LISTVOL64が使用出来ます。

また、FCUのMF-Information機能を使用してもデータセット名の一覧を参照出来ます。

1.2 システム構成

1.2.1 FAL/FCUプログラムの位置付け

FAL/FCUが動作するシステムの構成及びプログラムの位置付けを示します。



磁気ディスク装置(マルチプラットフォーム)

* VSEはVSE2.4以降がサポート対象です。

図1. 1-1 メインフレームーオープンシステム間のデータ交換におけるFAL/FCUの位置付け

[用語の説明]

- ① メインフレーム専用ボリューム：メインフレームだけで使用するファイル(データセット)を格納するためのボリュームです。オープンシステム側からはアクセスできません。
- ② 共有ボリューム：メインフレームとオープンシステムの間で共用するファイル(データセット)を格納するためのボリュームです。詳細は後節で説明します。
- ③ オープンシステム専用ボリューム：オープンシステムだけで使用するファイル及びメインフレーム側でバックアップするファイルを格納するためのボリュームです。
- ④ CKDフォーマット：メインフレームでファイルを作成する際の磁気ディスク上の記録フォーマットです。FAL/FCUを使用するときには本フォーマットを意識する必要はありません。
- ⑤ FBAフォーマット：オープンシステムでファイルを作成する際の磁気ディスク上の記録フォーマットです。通常、アプリケーションプログラムでは本フォーマットを意識する必要はありません。
- ⑥ 中間ファイル1：メインフレームで作成し、FCUの入出力となるファイル(データセット)です。データセットの編成種別は順編成(SAM)です。
- ⑦ 中間ファイル2：FCUで入出力するオープンシステムのファイルです。
- ⑧ MVS/VSE：IBM社が提供しているメインフレーム用のOS(オペレーティングシステム)です。
- ⑨ VOS3/VOS1：日立が提供しているメインフレーム用のOS(オペレーティングシステム)です。
- ⑩ MSP/XSP：富士通株式会社が提供しているメインフレーム用のOS(オペレーティングシステム)です。
- ⑪ UNIX：IBM社のAIX、Hewlett-Packard社のHP-UX、Sun Microsystems社のSolaris等があります。
- ⑫ Windows NT/2000Server/Server2003/Server2008/Server2008 R2/Server2012/Server2012 R2 /Server2016/Server2019：Microsoft社が提供しているOS(オペレーティングシステム)です。

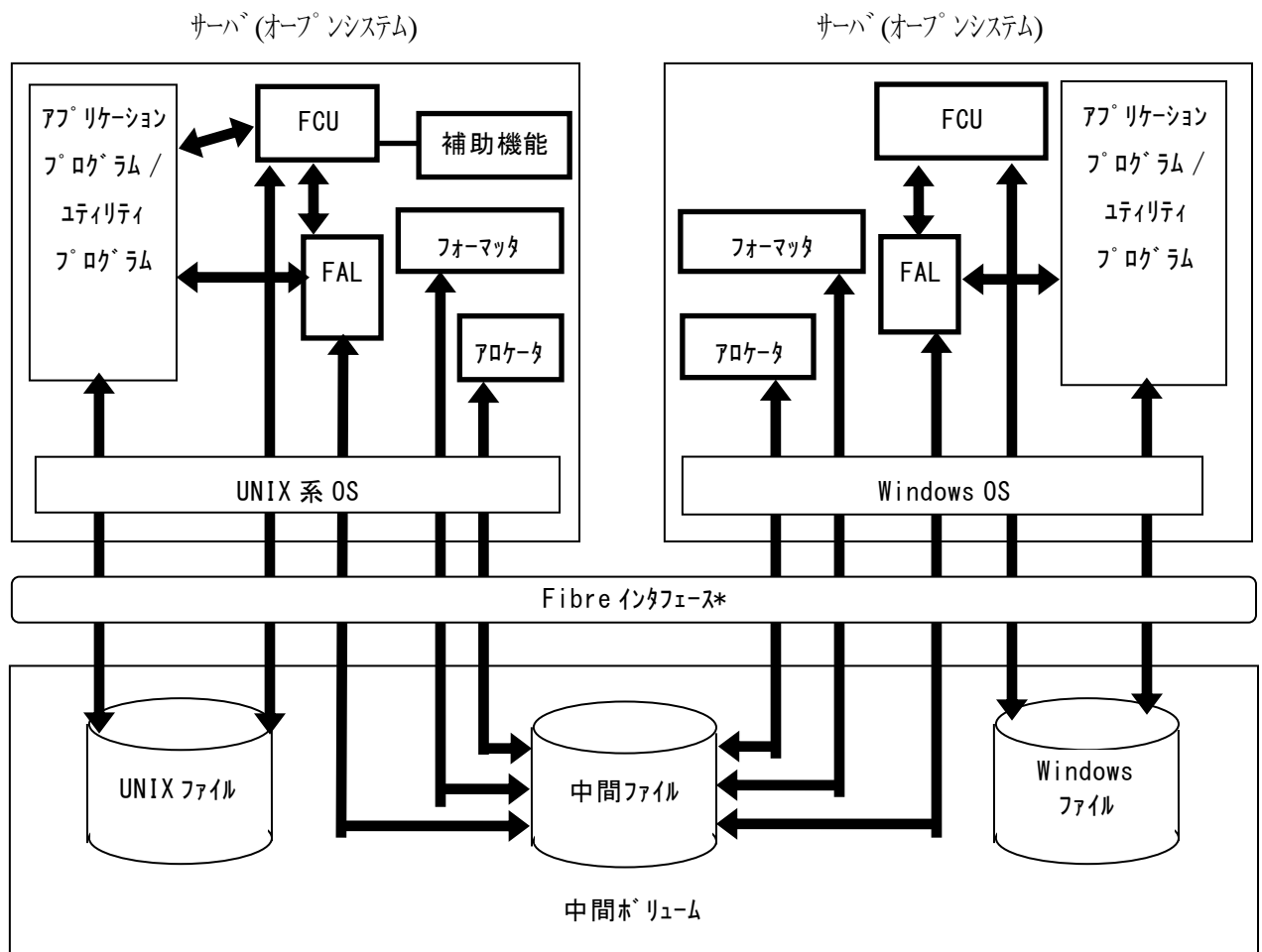
メインフレーム上のデータベースファイルをオープンシステム上のデータベースファイルに変換する場合、以下のような手順になります。

- (1) メインフレームで作成したデータベースファイルを、メインフレーム上のアプリケーションプログラム/ユーティリティプログラムを使用して、共有ボリューム上のSAM(順編成)ファイルに変換する。
- (2) オープンシステム上でFCUを実行して、共有ボリューム上のSAMファイルをオープンシステム専用ボリューム上のオープンシステムファイルに変換する。
このとき、コード変換が必要であればFCUにコード変換を指示する。
- (3) オープンシステム上でアプリケーションプログラム/ユーティリティプログラムを実行してオープンシステムファイルをデータベースファイルに変換する。

(注) パイプ機能を使用する場合(2)(3)の手順は次のようになります。

- (2) オープンシステム上で、FCUを実行して、共有ボリューム上のSAMファイルを取込み、パイプ(Named pipe)により、アプリケーションプログラム/ユーティリティプログラムへデータ転送を行う。
- (3) オープンシステム上のアプリケーションプログラム/ユーティリティプログラムのデータポートモジュールが、パイプにより転送されてきたデータを取込み、データベースファイルに変換する。

オープンシステムーオープンシステム間データ交換の場合のシステム構成及びプログラムの位置付けは下図のようになります



磁気ディスク装置(マルチプラットフォーム)

* Fibreは SANRISE SERIES 及びUSP Vシリーズ, VSPシリーズ, VSP G1000シリーズ, VSP 5000シリーズのみ

図1. 1-2 オープンシステムーオープンシステム間のデータ交換におけるFAL/FCUの位置付け

[用語の説明]

- ・ 中間ファイル：異機種のオープンシステム間でデータ交換する際に介在するファイル。
メインフレームのデータセットと同様のフォーマットになっています。
- ・ 中間ボリューム：中間ファイルを格納するボリューム。
メインフレーム専用ボリューム/共用ボリュームと同様のフォーマットになっています。

その他の用語については、前ページを参照して下さい。

1.2.2 サポート対象

1.2.2.1 メインフレームオープンシステム間のデータ交換の場合

(1) サポートするホスト(メインフレーム)のデータセット

MVS/VSE/VOS3/VOS1/MSP/XSP上で作成した以下の属性で定義されているデータセットをFAL/FCUで処理することができます。(VOS1/XSPをサポートしているプラットフォームは、Solaris/HP-UX/AIX/Windows NT/2000Server/Server2003/Server2008/Server2008 R2/Server2012/Server2012 R2/Server2016/Server2019です。VSEサポートバージョン対象はVSE2.4以降です。)

- ① データセット編成種別：順編成 (SAM)
- ② レコード形式：固定長レコード及び可変長レコード (VSE/VOS1の場合、(注)の13参照)
- ③ ブロック長：OSでサポートしている範囲のブロック長 (VSE/VOS1の場合、(注)の13参照)
- ④ レコード長：OSでサポートしている範囲のレコード長 (VSE/VOS1の場合、(注)の13参照)

(注) 1. 本ユーティリティで処理するデータセットは標準VTOCで管理されていることを前提としています。

索引(論理)VTOCを利用したデータセットサーチのような処理は行いません。ただし、索引VTOCの場合でも索引を利用した効率的な処理を行わないだけでデータセット管理情報を参照することはできます。従って索引VTOCで管理されているデータセットにもアクセスは可能です。

2. マルチボリュームデータセットは01-XX-50/YY(YY \geq 20)以降のバージョンでMTOのみ、01-XX-60/ZZ(XX \geq 03)以降からはMTOとOTM転送をサポートします。

(サポートするホストは、MVS/MSP/VOS3です)

それ以前のバージョンでは、データセットがマルチボリュームになっても1ボリューム内のデータセットについてだけ処理を行います。ただし、メインフレームがMSPまたはVOS3の場合マルチボリュームの最終ボリューム以外のボリュームにあるデータセットからのMTOは行えません。LISTVOL/LISTVOL64のTT R/MF-File/MF-file information/MF-File selectのDSは、0で表示されます。(最終ブロックアドレスが0の為)

3. データベースファイルへの直接アクセスはサポートしていません。

データベースファイルを処理する場合、一旦SAMファイルに変換する必要があります。

4. キー付きのレコードはサポートしていません。キー部があるとエラーになります。

5. 標準R0により管理されているトラックフォーマットを前提としています。

同一ボリューム内に、非標準R0を持つトラックが存在する場合FAL/FCUの処理は保証できません。

6. 順編成以外のデータセット(DAM, PAM, ISAM, VSAM等)はサポートしていません。

順編成以外のデータセットを指定するとエラーになります。

7. 不定長レコード、スパントレコードはサポートしていません。

8. 異常なブロック長、レコード長を持つデータを検出した場合、エラーを報告し、処理を中断します。

9. バージョン01-XX-49以前では、メインフレーム上の1ボリュームを越える大きさのデータセットを処理することはできません。バージョン01-XX-50ではMTOのみ可能となります。

10. データセット編成種別、レコード形式、ブロック長、レコード長、データセットサイズはFCUまたはLISTVOL/LISTVOL64で確認できます。

11. データセット名の中に「空白」/「漢字」/「英小文字」を含むデータセットは処理できません。

12. オープンシステムからメインフレームへデータを転送する場合、対象となるデータセットはVTOCに登録され、エクステントの割当、属性(データセット編成種別・レコード形式・ブロック長・レコード長)の定義が済んでいる必要があります。また、エクステントの自動拡張は行いませんので、オープンシステムからのデータを格納できる容量をあらかじめ用意しておく必要があります。

13. メインフレームのOSがVSE2.3または、メインフレームOSがVOS1で可変長レコードの場合、レコード形式・レコード長・ブロック長がVTOC中に定義されていないので、FCUのパラメータとしてレコード形式・レコード長・ブロック長を指定する必要があります(7.4節参照)。但し、VSE2.4をバージョン01-03-59で使用する場合は、FCUのパラメータとしてレコード形式・レコード長・ブロック長を指定する必要はありません。また、ユーザプログラムからFALをコールする使用方法では、VSE/VOS1のデータセットをFALで処理することはできません。

14. メインフレームのOSがXSPの場合、下記制限があります。

- ・サポートしているプラットフォームは、Solaris/HP-UX/AIX/Windows NT/2000Server/Server2003/Server2008/Server2008 R2/Server2012/Server2012 R2/Server2016/Server2019です。
- ・拡張VTOCはサポートしません。
- ・パスポートのついていないボリューム/データセットに対してもチェックなしで、OTM/MTO可能となります。
- ・LISTVOL/FCUにおいて、レポート形式の表示は、F/FB/V/VBのみです。
- ・LISTVOL/FCUにおいて、データセット編成種別がSAM以外の時、TT R EX(Cyl:Trk)の表示内容は保証できません。

15. メインフレームのOSがVOS1の場合、下記制限があります。

- ・サポートしているプラットフォームは、Solaris/HP-UX/AIX/Windows NT/2000Server/Server2003/Server2008/Server2008 R2/Server2012/Server2012 R2/Server2016/Server2019です。
- ・中間ボリュームにOTMしたデータは、FACYで別ボリュームにコピーするか、データ管理を用いて作成したUPにてデータの読み込みを行ってください。中間ボリューム上のデータセットは直接拡張処理を行えません。(FDUMPによるバックアップ/コピー後のデータも拡張処理は行えません。)

16. メインフレームOSのVSEサポートバージョンは、VSE2.4以降です。

(2) サポートするデバイスエミュレーションタイプ

マルチプラットフォーム機能を持つ大型ディスプレイ装置上に作成する中間ファイルを格納するボリュームとして、以下のデバイスエミュレーションタイプのボリュームをサポートします。

尚、AIXサーバでHDLN連携を行う場合は、AIXサーバに現在適用されている日立ディスプレイシステム向けHTC_ODMパッケージが、中間ボリュームに使用するデバイスエミュレーションタイプに対応していることを、ご確認願います。

(A) メインフレームのOSがMVS/VSEの場合

3390-3A	メインフレーム側からはAを除いた各々のエミュレーションタイプ（例えば3390-3Aの場合は3390-3）と同様のアクセスができます。 また、オープンシステム側からはデータの読出し・書込みができます
3380-KA	
3390-9A	
3390-LA	
3390-MA *	
3380-3A *	
3390-3B	メインフレーム側からはBを除いた各々のエミュレーションタイプ（例えば3390-3Bの場合は3390-3）と同様のアクセスができます。 オープンシステム側からはデータの読出しだけができます。
3380-KB	
3390-9B	
3390-LB	
3390-MB *	
3380-3B *	
3390-3C	メインフレーム側からはデータの読出しだけができます。 オープンシステム側からはデータの読出し・書込みができます。
3380-KC	
3390-9C	
3390-LC	
3390-MC *	
3380-3C *	

* : 3390-M, 3380-3はSANRISE USPシリーズ及びUSP VシリーズとVSPシリーズ、VSP G1000シリーズ、VSP 5000シリーズのサポート対象となります。

* : 3390-Mは、solarisのみ未サポートとなります。

(B) メインフレームのOSがMSPの場合

3380-KA	メインフレーム側からはAを除いた各々のエミュレーションタイプ [°] （例えば3380-KAの場合は3380-K）と同様のアクセスができます。 また、ホブシステム側からはデータの読出し・書込みができます
3380-KB	メインフレーム側からはBを除いた各々のエミュレーションタイプ [°] （例えば3380-KBの場合は3380-K）と同様のアクセスができます。 ホブシステム側からはデータの読出しだけができます。
3380-KC	メインフレーム側からはデータの読出しだけができます。 ホブシステム側からはデータの読出し・書込みができます。

(C) メインフレームのOSがVOS3の場合

6588-3A	メインフレーム側からはAを除いた各々のエミュレーションタイプ [°] （例えば6588-3Aの場合は6588-3）と同様のアクセスができます。 また、ホブシステム側からはデータの読出し・書込みができます
6588-9A	
6588-LA	
6586-KA	
6588-3B	メインフレーム側からはBを除いた各々のエミュレーションタイプ [°] （例えば6588-3Bの場合は6588-3）と同様のアクセスができます。 ホブシステム側からはデータの読出しだけができます。
6588-9B	
6588-LB	
6586-KB	
6588-3C	メインフレーム側からはデータの読出しだけができます。 ホブシステム側からはデータの読出し・書込みができます。
6588-9C	
6588-LC	
6586-KC	

(D) メインフレームのOSがVOS1/XSPの場合

6586-KA	メインフレーム側からはAを除いた各々のエミュレーションタイプ [°] （例えば6586-KAの場合は6586-K）と同様のアクセスができます。 また、ホブシステム側からはデータの読出し・書込みができます
3380-KA	
6586-KB	メインフレーム側からはBを除いた各々のエミュレーションタイプ [°] （例えば6586-KBの場合は6586-K）と同様のアクセスができます。 ホブシステム側からはデータの読出しだけができます。
3380-KB	
6586-KC	メインフレーム側からはデータの読出しだけができます。 ホブシステム側からはデータの読出し・書込みができます。
3380-KC	

(注) (A)～(D)のデバ イエミュレーションタイプ[°]について、*Cとなっているものはメインフレーム側からデータの読み出しだけが出来るものであるため、メインフレーム側より初期化等の更新処理を行った場合はエラーとなります。ホブシステム側からのデータ読み出し・書き込みはできます。
また、*Cの初期化(イニシャライズ)をメインフレーム側から実施する場合は、必ずQUICKイニシャライズ(先頭1シリンダ分のみをイニシャライズする)を指定して実施してください。

(2)-1. デバイスエミュレーションタイプ の制約

下記の表(2)-1. に記されるデバイスエミュレーションタイプを使用する場合には、SANRISE側及びUSP VまたはVSP, VSP G1000, VSP 5000側で使用するDKCメインマイクロプログラムに制約があります。

表(2)-1. 各デバイスエミュレーションタイプをサポートするDKCメインマイクロプログラムのバージョン

	SANRISE 9900 シリーズ	SANRISE, USP, VSP, VSP G1000 シリーズ VSP 5000 シリーズ
3390-9A/9B/9C	21-10-02-00/00 以上	50-01-03-00/00 以上
3390-LA/LB/LC		
3390-MA/MB/MC	未サポート	50-04-05-00/00 以上
6588-9A/9B/9C	21-14-07-00/00 以上	50-03-51-00/00 以上
6588-LA/LB/LC		

(2)-2. デバイスエミュレーションタイプに対応しないエミュレーションタイプをボリューム定義ファイルに記述した場合

SANRISE側及びUSP VまたはVSP, VSP G1000, VSP 5000側のデバイスエミュレーションタイプに対応しないエミュレーションタイプをボリューム定義ファイルに記述して転送を行った場合、最終的な転送結果が正常終了となった場合であっても、その転送データについては保障しません。SANRISE側及びUSP VまたはVSP, VSP G1000, VSP 5000側のデバイスエミュレーションタイプとボリューム定義ファイルに定義するエミュレーションタイプを合わせてから再度転送を行ってください。

例として、SANRISE側及びUSP VまたはVSP, VSP G1000, VSP 5000側のデバイスエミュレーションタイプが3390-3A/3B/3Cに対し、ボリューム定義ファイルのエミュレーションタイプを合わせずに3390-9A/9B/9C、3390-LA/LB/LC、3390-MA/MB/MCと定義して転送した場合、下記の表(2)-2. のようになります。

(ボリューム定義ファイルについては、8.2節を参照して下さい)

表(2)-2. エミュレーションタイプを合わせずに転送した場合

		MTO 転送時	OTM 転送時
ボリューム定義ファイルに定義するエミュレーションタイプ	3390-9A/9C	正常終了または FAL エラー(-13)	システムエラー(6) (AIX, HP, Sol, Linux)
	3390-9B		FAL エラー(-27) (Tru64, Win)
	3390-LA/LC	正常終了	FAL エラー(-25)
	3390-LB		正常終了
	3390-MA/MC	正常終了	システムエラー(5) (AIX, Tru64)
	3390-MB		システムエラー(22) (HP-UX) FAL エラー(-27) (Linux, Win)
		FAL エラー(-25)	

(3) サポートするデータ転送

メインフレームからオープンシステムへのデータ転送及びオープンシステムからメインフレームへのデータ転送をサポートします。

(注) オープンシステムからメインフレームへデータを転送する場合、対象となるデータセットはVTOCに登録され、エクステントの割当、属性(データセット編成種別、レコード形式、ブロック長、レコード長)の定義が済んでいる必要があります。また、エクステントの自動拡張は行いません。

(4) パイプ出力機能

パイプ (Named pipe) を使用してデータをOS上のアプリケーションプログラム/ユーティリティプログラムへ転送することができます。(メインフレームからオープンシステム方向のデータ転送時でUNIX系OSのみサポート)

(5) サポートするコード体系

(A) FAL/FCUは、メインフレーム側のコード体系として「EBCDIC/EBCDIK」、オープンシステム側のコード体系として「ASCII/JIS8」を前提としています。

(B) FCUではデータ実体に対して以下のコード変換を指定できます。

(a) FCUが所持するコード変換テーブルを使用したコード変換
EBCDIC ⇔ ASCII 及び EBCDIK ⇔ JIS8

(b) ユーザが所持するコード変換テーブルを使用したコード変換
・メインフレーム-オープンシステム間で任意の1バイトコード変換ができます。

1.2.2.2. FAL/FCUサポートリスト

(1) FAL/FCU(32bit 版)サポートリスト

各サブシステムとオープンシステムに対して FAL/FCU(32bit)が動作するサポートリストを以下に示します。

OS	サブシステム	CPUタイプ	OSビット	バージョン	
AIX	VSP 5000 シリーズ VSP G1000 シリーズ VSP シリーズ USP V シリーズ SANRISE ・ USP シリーズ ・ 9900 シリーズ ・ 2000 シリーズ (*4)	Power PC	32	4.3.3(*5), 5.1, 5.2, 5.3, 6.1(*3), 7.1(*3), 7.2(*2)	
			64	4.3.3(*5), 5.1, 5.2, 5.3, 6.1(*3), 7.1(*3), 7.2(*2)	
	VSP 5000 シリーズ VSP G1000 シリーズ VSP シリーズ USP V シリーズ	Power HA(*7)	32/64	7.2	
Solaris	VSP 5000 シリーズ VSP G1000 シリーズ VSP シリーズ USP V シリーズ SANRISE ・ USP シリーズ ・ 9900 シリーズ ・ 2000 シリーズ	UltraSPARC	32	2.6(*5), 7(*5), 8, 9, 10, 11	
			64	7(*5), 8, 9, 10, 11	
Linux	Redhat	VSP 5000 シリーズ VSP G1000 シリーズ VSP シリーズ USP V シリーズ SANRISE ・ USP シリーズ ・ 9900 シリーズ	x86	32	7.2(*5), AS2.1(*5), AS3.0(*1), AS4.0 U5 以降(*2), AS5.1(*2), AS5.2(*2), AS5.3(*2), AS5.4(*2), AS6.1(*2)
			AMD64/Intel64	64	AS3.0(*1), AS4.0 U5 以降(*2), AS5.1(*2), AS5.2(*2), AS5.3(*2), AS5.4(*2), AS6.1(*2)
		SuSE	x86	32	SLES9, 10(*2), 11(*2)
			AMD64/Intel64	64	
	Windows	VSP 5000 シリーズ VSP G1000 シリーズ VSP シリーズ USP V シリーズ SANRISE ・ USP シリーズ ・ 9900 シリーズ	x86	32	Windows NT4.0 Server (SP6) (*5) Windows 2000Server (SP1 以上) (*5) Windows Server2003 Windows Server2008 Windows Server2008 R2
Itanium AMD64/Intel64			64	Windows Server2003 Windows Server2008 Windows Server2008 R2 Windows Server2012 (*6) Windows Server2012 R2 (*6) Windows Server2016 (*6) Windows Server2019 (*6)	
SANRISE ・ 2000 シリーズ		x86	32	Windows NT4.0 Server (SP6) Windows 2000Server (SP1 以上)	

*1 : 01-03-60以降サポート

*2 : 01-XX-66/26 以降 (XX ≥ 05) サポート

*3 : 01-XX-65/25 以降 (XX ≥ 03) 及び 01-XX-66/25 以降 (XX ≥ 05) サポート

*4 : AIX6.1 では未対応

*5 : VSP シリーズ 対応バージョン以降では未対応

*6 : Itanium は未サポート

*7 : VIOS 2.2.6.23 でサポート

(2) FAL/FCU(32bit 版)対象バージョン

各サブシステムに対する FAL/FCU (32bit 版)のバージョンを示します。

サブシステム	FAL/FCU 対応バージョン
SANRISE2000 シリーズ (*1)	01-02-46 以降
SANRISE9900 シリーズ	01-03-53 以降
SANRISE USP シリーズ	01-04-61 以降
USP V シリーズ	01-05-66 以降
VSP シリーズ	01-06-67 以降
VSP G1000 シリーズ	01-07-68 以降
VSP 5000 シリーズ	01-08-69 以降

*1 : AIX6.1 では未対応

(3) FAL/FCU(64bit 版)サポートリスト

各サブシステムとオペレーティングシステムに対して FAL/FCU(64bit 版)が動作するサポートリストを以下に示します。

OS	サブシステム	CPUタイプ	OSビット	バージョン
HP-UX	VSP 5000 シリーズ	Itanium	64	11.23, 11.31
Solaris	VSP G1000 シリーズ	Ultra SPARC		7(*3), 8, 9, 10, 11
AIX	VSP シリーズ USP V シリーズ	Power PC		4.3.3(*3), 5.1, 5.2, 5.3, 6.1(*2), 7.1(*2), 7.2(*1)
		Power HA(*5)		7.2
Linux	Redhat SuSE	AMD64/Intel64		AS3.0, AS4.0 U5 以降(*1), AS5.1(*1), AS5.2(*1), AS5.3(*1), AS5.4(*1), AS6.1(*1)
				SLES9, 10(*1), 11(*1)
Windows		Itanium		Windows Server2003 Windows Server2008 Windows Server2008 R2 Windows Server2012(*4) Windows Server2012 R2(*4) Windows Server2016(*4) Windows Server2019(*4)
		AMD64/Intel64		

*1 : 01-XX-66/26 以降 (XX ≥ 05) サポート

*2 : 01-XX-65/25 以降 (XX ≥ 03) 及び 01-XX-66/25 以降 (XX ≥ 05) サポート

*3 : VSP シリーズ 対応バージョン以降では未対応

*4 : Itanium は未サポート

*5 : USP V シリーズ 以降且つ VIOS 2.2.6.23 でサポート

(4) FAL/FCU(64bit)対象バージョン

各サブシステムに対する FAL/FCU(64bit 版)のバージョンを示します。

サブシステム	FAL/FCU 対応バージョン
SANRISE2000 シリーズ	未対応
SANRISE9900 シリーズ	01-03-59 以降
SANRISE USP シリーズ	01-04-61 以降
USP V シリーズ	01-05-66 以降
VSP シリーズ	01-06-67 以降
VSP G1000 シリーズ	01-07-68 以降
VSP 5000 シリーズ	01-08-69 以降

1.2.2.3. FALライブラリサポートリスト

32ビット版または64ビット版のFALライブラリはUP(ユーザプログラム)に組み込みエークに使用することができます。

(1) FALライブラリ(32bit版)開発環境

FALライブラリ(32bit版)の開発環境を以下に示します(下記表にあるOSバージョンまたはコンパイラバージョンより新しい環境でUPを作成する場合には必要な環境下で動作検証を行ってください)。

OSに対する各コンパイラバージョンはメカが保証するバージョンをご使用下さい。

OS	CPUタイプ	開発 OS バージョン	開発コンパイラバージョン
Solaris	Ultra SPARC	2.6	WorkShop Compilers 4.2.1 04 Feb 1997 C 4.2.1
AIX	Power PC	4.3.3	C for AIX Compiler Version 4.4.0.0
Linux	x86	Redhat 7.2	gcc version 2.96 20000731 (Red Hat Linux 7.1 2.96-98)
Windows	x86	Windows NT4.0 Server (SP6)	Visual C++ 6.0 SP6

(2) FALライブラリ(64bit版)サポートリスト

FALライブラリ(64bit版)の開発環境を以下に示します(下記表にあるOSバージョンまたはコンパイラバージョンより新しい環境でUPを作成する場合には必要な環境下で動作検証を行ってください)。

OSに対する各コンパイラバージョンはメカが保証するバージョンをご使用下さい。

OS	CPUタイプ	開発 OS バージョン	開発コンパイラバージョン
HP-UX	Itanium	11.23	HP aC++/ANSI C B3910B A.05.52 [Sep 05 2003] ※ B3901BA C.05.50 HP C/ANSI C Developer's Bundle (S800)に下記パッチを適用 ・ PHSS_29656 HP C Compiler (A.05.52) ・ PHCO_29957 libc cumulative patch
Solaris	Ultra SPARC	7	WorkShop Compilers 5.0 98/12/15 C 5.0
AIX	Power PC	4.3.3	C for AIX Compiler Version 4.4.0.0
		5.1	C for AIX Compiler Version 5.0.2.8
Linux	Itanium	AS2.1	gcc version 2.96 20000731 (Red Hat Linux 7.2 2.96-118.7.2)
		AS3.0	gcc version 3.2.3 20030502 (Red Hat Linux 3.2.3-20)
Windows (01-XX-66以降)	Itanium	WindowsServer	Microsoft Platform SDK for Windows Server 2003 SP1
	x64	2003	Visual C++ 2005

1.2.2.4. HDLM(JP1/HiCommand Dynamic Link Manager) サポート

FAL/FCUは中間ボリュームに対する交替パス指定(HDLM:JP1/HiCommand Dynamic Link Manager)をサポートしています。FAL/FCUとHDLM連携をする場合、FAL/FCUの対象バージョンは01-XX-56以降です。FAL/FCUはOS(オペレーションシステム)上で動作するソフトウェアです。接続した中間ボリュームを使用可能なデバイスとしてOS上で検出している事がFAL/FCUの動作条件となります。

以下にFAL/FCUが動作実績のあるHDLMバージョンを示します。

尚、HDLMに関する詳細は、HDLMのユーザズガイドを参照してください。

OS(オペレーションシステム)	HDLM Version
Windows 2000/NT	04-01, 05-00-/B
AIX	05-00-/A, 05-01
HP-UX	04-00-/A
Solaris	04-01, 05-00

AIXサーバ使用時の注意事項

HDLM連携を行う場合は、現在適用されている日立ディスクレイアウトシステム向けHTC_ODMパッケージが、中間ボリュームに使用するデバイスエミュレーションタイプに対応していることを、ご確認願います。

1.2.2.5. (I)/(H)シェア、(I)/(F)シェア、(H)/(F)シェア、MVS ICKDSF(R17)サポート

FAL/FCUは、以下に説明する(I)/(H)シェア、(I)/(F)シェア、(H)/(F)シェア、MVSのICKDSF(R17)をサポートしています。それぞれの環境でFAL/FCUを使用する場合、FAL/FCUの対象バージョンは01-XX-57以降です。(XX≥02)

- (I)/(H)シェア: DKUエミュレーション3390-3Aまたは3380-KAをMVS及びVOS3で共用して使用する
- (I)/(F)シェア: DKUエミュレーション3380-KAをMVS及びMSPで共用して使用する
- (H)/(F)シェア: DKUエミュレーション3380-KAをVOS3及びMSPで共用して使用する
- MVS ICKDSF(R17): DKUエミュレーション3390-3または3380-KをMVSのICKDSF(R17)でフォーマットして使用する。

1.2.2.6. VSEサポートと転送条件

FAL/FCUは、メインフレームOSのVSEをサポートしています。

以下に、サポート条件とMTO及びOTM転送時の条件を示します。

(a) サポート条件

サポートするVSEのバージョンとFAL/FCUのバージョン及びサブシステムを示します。

(下記表の組合せ以外では、FAL/FCUの正常動作を保証できません。)

OSバージョン	サブシステム	FAL/FCUバージョン
VSE2.4	SANRAISE9900 シリーズ	01-03-59 以降
	SANRISE USP シリーズ	01-04-61 以降
	USP V シリーズ	01-05-66 以降
	VSP シリーズ	01-06-67 以降
	VSP G1000 シリーズ	01-07-68 以降
	VSP 5000 シリーズ	01-08-69 以降

(b) MTO 及び OTM 転送時の条件

VSE2.3(FAL/FCU のバージョンは 01-xx-58 以下)でアクションしたデータセットに MTO 及び OTM 転送を行なう場合は VSE2.3 のデータ形式上、VSE パラメタにより RF(レコードフォーマット、以下 RF とする)、RL(レコード長、以下 RL とする)、BL(ブロック長、以下 BL とする)の指定が必須となります。

VSE2.4(FAL/FCU のバージョンは 01-xx-59 以降)でアクションしたデータセットに MTO 及び OTM 転送を行なう場合は、VSE パラメタの指定無しでも転送を可能としています。以下に VSE パラメタの指定と MTO/OTM 転送の対応表と転送時のデータセット属性(VSE パラメタ)の条件を示します。

MTO/OTM 転送時の VSE 指定転送対応表

OS バージョン (FAL/FCU バージョン)	VSE 指定	データセット属性			
		F	FB※5	V	VB※5
VSE2.3 (01-xx-58 以前)	有	○	○	○	○
	無	×	×	×	×
VSE2.4 (01-xx-59 以降)	有	○	○	○※1	○※2
	無	○	○	○※3	○※4

○：転送可能 ×：転送不可

F：固定長非ブロック化 FB：固定長ブロック化 V：可変長非ブロック化 VB：可変長ブロック化

※1：正常なデータセット属性($5 \leq RL \leq BL - 4$ が成り立つデータセット属性)では転送可能ですが、以下に示す VSE パラメタの指定条件に一致しない場合は転送が無効になります。

- $RL \leq 32756$
- $BL \leq 32760$
- $BL = RL + 4$

また、VTOC 上データの RL 及び BL の値が $RL > 32756$ 、 $BL > 32760$ の場合は、上記の条件に一致する様に VSE パラメタを指定しなければ、転送が無効になります。

更にまた、VTOC 上データの RL 及び BL の値が $RL \leq 32756$ 、 $BL \leq 32760$ の場合で、 $RL = BL$ の場合は、

- $RL(\text{VSE パラメタに入力する値}) = RL(\text{VTOC 上のデータの値}) + 4 \leq 32756$
- $BL(\text{VSE パラメタに入力する値}) = BL(\text{VTOC 上のデータの値}) + 8 \leq 32760$

上記の指定条件で VSE パラメタを指定しなければ、転送が無効になります。

※2：正常なデータセット属性($5 \leq RL \leq BL - 4$ が成り立つデータセット属性)では転送可能ですが、以下に示す VSE パラメタの指定条件に一致しない場合は転送が無効になります。

- $RL(\text{VSE パラメタに入力する値}) = RL(\text{VTOC 上のデータの値}) + 4 \leq 32756$
- $BL(\text{VSE パラメタに入力する値}) = BL(\text{VTOC 上のデータの値}) + 8 \leq 32760$

※ 3 : 正常なデータセット属性($BL=RL+4 \leq 32760$ が成り立つデータセット属性)では転送可能ですが、VTOC 上のデータの RL, BL の値が以下に示す条件に一致しない場合は転送が無効になります。

- $BL=RL+4 \leq 32760$

また、VTOC 上データの RL 及び BL の値が $RL > 32756$ 、 $BL > 32760$ の場合は、

RapidXchange として $RL=32756$ 、 $BL=32760$ とみなして転送を行います。

更にまた、VTOC 上データの RL 及び BL の値が $RL \leq 32756$ 、 $BL \leq 32760$ の場合で、

$RL=BL$ の場合は、

- $RL(\text{RapidXchange として扱う値})=RL(\text{VTOC 上のデータの値})+4 \leq 32756$

- $BL(\text{RapidXchange として扱う値})=BL(\text{VTOC 上のデータの値})+8 \leq 32760$

上記の条件で RapidXchange は、それぞれの RL, BL の値をみなして転送を行います。

※ 4 : 正常なデータセット属性($BL=RL+4 \leq 32760$ が成り立つデータセット属性)では転送可能ですが、VTOC 上のデータの RL, BL の値が以下に示す条件に一致しない場合は転送が無効になります。

- $BL=RL+4 \leq 32760$

また、VTOC 上データの RL 及び BL の値が $RL=BL \leq 32752$ の場合は、RapidXchange として

$RL(\text{VTOC 上のデータの値})+4$ 、 $BL(\text{VTOC 上のデータの値})+8$ とみなして転送を行います。

※ 5 : データセット属性が FBA/FBM は FB として、VBA/VBM は VB として扱います。

尚、データセット属性の変換規則をユーザプログラムにより作成する場合、外部関数は以下が対象となります。

- datasetGetFileInformation ()
- datasetGetFileInformationEx ()

1. 2. 2. 7. 開発ツール対応 OS

開発ツールとそれぞれの対応 OS の関係を示します。これは、FAL/FCU で開発、正常動作を確認したことにより動作を保証した開発ツールと対応 OS の情報です。以下に示したバージョンよりも低いレベルの環境下でユーザプログラムを開発した場合はユーザの責任により必要な環境化でテストして頂く必要があります。

RedHat Linux 7.2(カーネルバージョン 2.4.7-10)の対応開発ツール

- gcc(Ver. 2.96 20000731 (Red Hat Linux 7.1 2.96-98))
- glibc(Ver.2.2.4-13)

RedHat Linux AS2.1(64bit)(カーネルバージョン 2.4.18-e.31)の対応開発ツール

- gcc(Ver. 2.96 20000731 (Red Hat Linux 7.2 2.96-118.7.2))
- glibc(Ver. 2.2.4-32.3)

RedHat Linux AS3.0(64bit)(カーネルバージョン 2.4.21-9.0.1.EL)の対応開発ツール

- gcc(Ver. 3.2.3 20030502 (Red Hat Linux 3.2.3-20))
- glibc(Ver. 2.3.2-95.3)

Solaris2.6 の対応開発ツール

- Work Shop Compilers 4.2.1

Solaris7(64bit)の対応開発ツール

- Work Shop Compilers 5.0

Windows NT 4.0 service Pack 6a の対応開発ツール

- Visual C++ 6.0 SP4

WindowsServer2003(64bit) Itanium 版の対応開発ツール

- Microsoft Platform SDK for Windows Server 2003 SP1

WindowsServer2003(64bit) x64 版の対応開発ツール

- Visual C++ 2005

HP-UX11iv2(64bit) Itanium 版の対応開発ツール

- HP aC++/ANSI C B3910B A.05.52 [Sep 05 2003]

AIX4.3 の対応開発ツール

- C for AIX 4.4

AIX5L(64bit)の対応開発ツール

- C for AIX, Version 5.0

Tru64 の対応開発ツール

<バージョン 01-XX-59 以前>

- Digital UNIX Compiler Driver 3.11
- DEC C V5.6-075 on Digital UNIX V4.0 (Rev. 878)

<バージョン 01-XX-60 以降>

- Digital UNIX Compiler Driver 3.11
- DEC C V5.9-005 on Digital UNIX V4.0 (Rev. 1229)

HI-UX の対応開発ツール

- P-1632-111+C0 01-10-/A 最適化C

1.2.2.8. FCU64bitサポート

FAL/FCUは64bit版対応のFCUをサポートしています。使用方法は32bit版のFCUと同様です。但し、64bit版のFCUは、GUIには対応していません(Windows版はGUI対応)。尚、FAL/FCUの64bit版の対応バージョンと対応OSは以下の通りです。

FAL/FCUバージョン(64bit対応)	対応OS
01-03-59以降 01-04-61以降	Solaris, HP-UX (PA-RISC), AIX4.3, AIX5L
01-XX-62以降 (XX ≥ 03)	Redhat Linux AS3.0
01-XX-63以降 (XX ≥ 03)	Redhat Linux AS2.1, HP-UX11iv2(Itanium)
01-XX-66以降 (XX ≥ 05)	Windows Server2003(Itanium/x64), Windows Server2008(Itanium/x64) Windows Server2008 R2 (Itanium/x64) Windows Server2012(x64) Windows Server2012 R2 (x64) Windows Server2016(x64) Windows Server2019(x64)
01-XX-65以降 (XX ≥ 03) 01-XX-66以降 (XX ≥ 05)	AIX6.1 AIX7.1 AIX7.2
01-XX-66以降 (XX ≥ 05)	Redhat Linux AS4.0(Itanium)※1 Redhat Linux AS5.1(Itanium) Redhat Linux AS5.2(Itanium) Redhat Linux AS5.3(Itanium) Redhat Linux AS5.4(Itanium) Redhat Linux AS6.1(Itanium) SUSE Linux ES10(Itanium) SUSE Linux ES11(Itanium)

※1 Linux AS4.0についてはUpdate5以降がサポート対象となります。

1.2.2.9. オープンシステムーオープンシステム間のデータ交換の場合

(1) サポートするデバイスエミュレーションタイプ

大型ディスクレイ装置上に作成する中間ファイルを格納するボリュームとして、デバイスエミュレーションタイプ = OPEN-3 / OPEN-8 / OPEN-9 / OPEN-K / OPEN-E / OPEN-L / OPEN-M / OPEN-V のボリュームをサポートします。

(2) サポートするデータ転送方向

中間ファイルからオープンシステムファイル及びオープンシステムファイルから中間ファイルへのデータ転送をサポートします。

(3) バイブ出力機能

バイブ (Named pipe) を使用してデータをOS上のアプリケーションプログラム/ユーティリティプログラムへ転送することができます。(中間ファイルからオープンシステムファイル方向のデータ転送時でUNIX系OSのみサポート)

(4) コード体系

オープンシステムー中間ファイル間のデータ転送においては、コード変換をサポートしていません。

(5) オープンシステムのハードウェア・ソフトウェア

(i) フォーマット/アスキー/FAL/FCUが動作するオープンシステム、OS、GUI については前ページを参照して下さい。

(ii) データ交換可能なマシンの組み合わせは以下の通りです。

後続マシン	AIX	HP-UX	Solaris	Windows	Linux
ラベルを書くマシン					
AIX	○	○	○	○	○
HP-UX	○	○	○	○	○
Solaris	○	○	○	○	○
Windows	○	○	△	△	○
Linux	○	○	○	○	○

(凡例) ○ : 組み合わせ可能

△ : ラベルの書込みを行った場合不可

1. 2. 2. 10. メインフレームボリューム/オープンボリューム データ格納容量算出方法

(1) エミュレーションタイプ 3390-3/3390-9/3390-L/3390-M/6588-3/6588-9/6588-L

の場合は、表1. 2. 2. 10-1 ブロック長と1トラック当りの書込み可能データ量の関係を参照して下さい

①表1. 2. 2. 10-1 を参考に、1トラックあたりのブロック長がいくつ入るか調べます。

②指定されたブロック長の中に、レコード長がいくつ入るか調べます。

例1 : エミュレーションタイプ 3390-3の場合

データセット レコード長:80 ブロック長:800 Cyl:10

①39 ②10

1 Trkあたり : $39 \times (800/80) \times 80\text{rec} = 31,200(\text{byte})$

1 Cylあたり : $31,200 \times 15\text{Trk} = 468,000(\text{byte})$

10CYLあたり : $468,000 \times 10\text{Cyl} = 4,680,000(\text{byte})$ 格納できます。

例2 : エミュレーションタイプ 3390-3の場合

データセット レコード長:80 ブロック長:80 Cyl:10

①78 ②1

1 Trkあたり : $80 \times (80/80) \times 80\text{rec} = 6,400(\text{byte})$

1 Cylあたり : $6,400 \times 15\text{Trk} = 96,000(\text{byte})$

10CYLあたり : $96,000 \times 10\text{Cyl} = 960,000(\text{byte})$ 格納できます。

表 1. 2. 2. 10-1 ブロック長と1トラック当りの書込み可能データ量の関係

アロケータで指定するブロック長 (バイト)	1トラックに書込み可能なデータ量 (バイト)	アロケータで指定するブロック長 (バイト)	1トラックに書込み可能なデータ量 (バイト)	アロケータで指定するブロック長 (バイト)	1トラックに書込み可能なデータ量 (バイト)
27999~32760	(ブロック長)×1	1851~1946	(ブロック長)×22	657~690	(ブロック長)×43
18453~27998	(ブロック長)×2	1749~1850	(ブロック長)×23	623~656	(ブロック長)×44
13683~18452	(ブロック長)×3	1647~1748	(ブロック長)×24	589~622	(ブロック長)×45
10797~13682	(ブロック長)×4	1551~1646	(ブロック長)×25	555~588	(ブロック長)×46
8907~10796	(ブロック長)×5	1483~1550	(ブロック長)×26	521~554	(ブロック長)×48
7549~8906	(ブロック長)×6	1387~1482	(ブロック長)×27	487~520	(ブロック長)×49
6519~7548	(ブロック長)×7	1319~1386	(ブロック長)×28	459~486	(ブロック長)×50
5727~6518	(ブロック長)×8	1251~1318	(ブロック長)×29	425~458	(ブロック長)×52
5065~5726	(ブロック長)×9	1183~1250	(ブロック長)×30	391~424	(ブロック長)×54
4567~5064	(ブロック長)×10	1155~1182	(ブロック長)×31	357~390	(ブロック長)×55
4137~4566	(ブロック長)×11	1087~1154	(ブロック長)×32	323~356	(ブロック長)×57
3769~4136	(ブロック長)×12	1019~1086	(ブロック長)×33	289~322	(ブロック長)×59
3441~3768	(ブロック長)×13	985~1018	(ブロック長)×34	255~288	(ブロック長)×61
3175~3440	(ブロック長)×14	951~984	(ブロック長)×35	227~254	(ブロック長)×64
2943~3174	(ブロック長)×15	889~950	(ブロック長)×36	193~226	(ブロック長)×66
2711~2942	(ブロック長)×16	855~888	(ブロック長)×37	159~192	(ブロック長)×69
2547~2710	(ブロック長)×17	821~854	(ブロック長)×38	125~158	(ブロック長)×72
2377~2546	(ブロック長)×18	787~820	(ブロック長)×39	91~124	(ブロック長)×75
2213~2376	(ブロック長)×19	753~786	(ブロック長)×40	57~90	(ブロック長)×78
2083~2212	(ブロック長)×20	719~752	(ブロック長)×41	23~56	(ブロック長)×82
1947~2082	(ブロック長)×21	691~718	(ブロック長)×42	1~22	(ブロック長)×86

(1) エミュレーションタイプ 3380-K/3380-3/6586-K

OPEN-3/OPEN-8/OPEN-9/OPEN-K/OPEN-E/OPEN-L/OPEN-M/OPEN-V

の場合は、表2.4-1 ブロック長と1トラック当りの書込み可能データ量の関係参照して下さい
(格納容量は、OPEN系と3380-Xは同一となりますのでOPENを3380-Xと読み替えて使用して下さい)

①表2.4-1 を参考に、1トラックあたりのブロック長がいくつ入るか調べます。

②指定されたブロック長の中に、レコード長がいくつ入るか調べます。

例3：エミュレーションタイプ 3380-Kの場合

データセット レコード長:80 ブロック長:800 Cyl:10

①36 ②10

1 Trkあたり：36×(800÷80)×80rec = 28,800(byte)

1 Cylあたり：28,800×15Trk = 432,000(byte)

10CYLあたり：468,000×10Cyl = 4,320,000(byte) 格納できます。

例4：エミュレーションタイプ OPEN-3の場合

データセット レコード長:80 ブロック長:80 Cyl:10

①83 ②1

1 Trkあたり：83×(80÷80)×80rec = 6,640(byte)

1 Cylあたり：6,640×15Trk = 99,600(byte)

10CYLあたり：99,600×10Cyl = 996,000(byte) 格納できます。

LargeファイルサポートバージョンではOPENボリュームをLUSEでつなげた場合、使用できるデータセットの最大シリンダ数は65533シリンダとなります。

メインフレームボリュームで使用できるデータセットのシリンダ数は最大で4369シリンダとなります。

マルチボリュームデータセットで最大30ボリューム(*1)のデータセットが転送可能です。

*1 01-XX-66以降からは最大61ボリュームをサポート

1.2.2.11. インストール時のディスク容量を 表1.2.2.11-1 インストールディスク容量一覧をに示す

表1.2.2.11-1 インストールディスク容量一覧

OS	CPUタイプ	FAL/FCU(32bit)	FAL/FCU(64bit)
HP-UX	Itanium		約 2.0M
Solaris	Ultra SPARC	約 1.5M	約 1.0M
AIX	Power PC	約 1.5M	約 1.0M (AIX4, AIX5L, AIX6.1)
Linux	x86 AMD64/Intel64	約 1.0M	約 1.0M
Windows	x86	約 2.0M	
Windows	Itanium		約 8.2M
Windows	x64		約 4.3M

2 オープンシステム-オープンシステム間のデータ交換時の環境構築

2.1 環境設定

オープンシステム-オープンシステム間データ交換を行うためには、以下の手順により動作環境を設定します。

- ① 中間ボリュームとして使用するRAWデバイス¹の定義
- ② フォーマット/アロケータ/FAL/FCUのインストール
- ③ ボリューム定義ファイルの作成
- ④ フォーマットによる中間ボリュームの生成
- ⑤ アロケータによる中間ファイルの生成
- ⑥ パラメータ定義ファイルの作成 (FCUを使用する場合)
FALを使用するプログラムのコーディング・コンパイル (FALを使用する場合)
- ⑦ データ交換の元データ/ファイルの作成



FCU/FALを使用するプログラムの実行

2.2 RAWデバイス¹の定義

中間ボリュームとして使用するボリューム(デバイスエミュレーションタイプ = OPEN-X)をRAWデバイス¹として定義して下さい。(中間ボリュームとして使用するOPEN-Xと他の目的に使用するOPEN-Xとを識別する情報はありませんので使い分けに十分注意して下さい) (X: 3, 8, 9, K, E, L, M, V)

なお、中間ボリュームにファイルシステムを作成する以下ような処理を実行してはなりません。

(情報が破壊され、中間ボリュームとして使用できなくなります)

- (1) Solarisの場合
 - ・ newfsコマンド²の実行
- (2) HP-UXの場合
 - ・ pvcreateコマンド²の実行
- (3) AIXの場合
 - ・ ボリュームグループ³の生成/PVIDの設定
- (4) Digital UNIX/Tru64 UNIXの場合
 - ・ newfsコマンド²の実行
- (5) DYNIX/ptxの場合
 - ・ スライスの設定/ファイルシステムの生成(ラベル書込み)
- (6) NCR UNIXの場合
 - ・ スライスの設定/ファイルシステムの生成
- (7) WindowsNT/2000Server/Server2003/Server2008/Server2008 R2/Server2012/Server2012 R2/Server2016/Server2019の場合
 - ・ パーティションの設定/ファイルシステムの生成
- (8) Linuxの場合
 - ・ mkfsコマンド²の実行

2.3 中間ボリュームの生成

フォーマットを起動して、raw デバイスとして定義されたボリューム上に、FAL/FCU が使用する中間ボリュームのフォーマットを作成します。

(注 1) フォーマット/アロケータには UNIX 系 OS 用と Windows 用とがあります。

(注 2) 中間ボリュームのサイズを表すシリンダの大きさは以下の通りですが、実際に格納できるデータ量は以下の値より小さくなります。詳細は2.4節を参照して下さい。

・ 1 シリンダ = 15 トラック、1 トラック = 96 サブブロック、1 サブブロック = 512 バイト

(注 3) フォーマットで指定できるシリンダ容量の最大値は、エミュレーションタイプにより異なります。

また Solaris では、使用できるシリンダ数が他と異なる場合がありますので注意して下さい。(*4)

(YY ≥ 48, x = 数字, ZZ < 20)

エミュレーションタイプ	OPEN-3	OPEN-8	OPEN-9	OPEN-K	OPEN-E(*3)	OPEN-L(*3)	OPEN-M(*3)
01-XX-47 以前	3331	5818(*1)	5818(*1)	2536	未サポート	未サポート	未サポート
01-XX-YY/ZZ LUSE なし	3331	5818(*1)	5818(*1)	2536	5818(*1)	5818(*1)	5818(*1)
01-XX-YY/ZZ LUSE あり	5818(*1)	5818(*1)	5818(*1)	5818(*1)	5818(*1)	5818(*1)	5818(*1)
01-XX-YY/2x LUSE なし	3331	9959	10009	2536	19752	49429(*4)	63976(*4)
01-XX-YY/2x LUSE あり n: 連結ボリューム数	3338 × n - 7 (*2) (*4)	9966 × n - 7 (*2) (*4)	10016 × n - 7 (*2) (*4)	2543 × n - 7 (*2) (*4)	19759 × n - 7 (*2) (*4)	49439 × n - 7 (*2) (*4)	63999 × n - 7 (*2) (*4)

エミュレーションタイプ	OPEN-V 作成時 (1 トラック = 128 サブブロック換算)	OPEN-V フォーマット指定 (1 トラック = 96 サブブロック換算) (*3)
01-XX-55 以前		未サポート
01-XX-56/2x LUSE なし n: 連結ボリューム数	49160 Solaris の場合 32768 以上使用する場合(*4) (*5) 参照してください。	65534 (*2) (*4) (*5)
01-XX-56/2x LUSE あり n: 連結ボリューム数	LUSE で連結して 49160 シリンダの場合 フォーマットの最大値 65534 シリンダ設定可能	フォーマットで指定可能なシリンダ数(整数) ≤ (OPEN-V 作成時のシリンダ値 × n × 128 ÷ 96) - 7 <u>(注) 割り切れない数値は使用しないで下さい。</u> (*2) (*4) (*5)

(*1) :バージョン 01-XX-YY/ZZ (YY = 数字, ZZ < 20) (Large ファイル未サポート版) では最大シリンダまで扱えません。

(*2) :ボリュームを連結しても、最大 65534 シリンダしかフォーマットできません。

(*3) :OPEN-E/L/M/V は AIX/HP-UX/Solaris/Windows 版のみのサポートです。

(*4) :Solaris では、32768 シリンダ以上のボリュームに対するジオメトリ設定値が変わる為それ以上のボリューム (LUSE 含む) では、指定できるシリンダ数が小さくなります。
(FAL で指定するシリンダ数は、Head = 15、Block/Track = 96 として扱います。)

Solaris を使用する場合、指定シリンダ数は次を参考に算出して下さい。

ジオメトリパラメータで指定する Head : A

ジオメトリパラメータで指定する Block/Track : B

ジオメトリパラメータで指定する Cylinder : C とすると

$$\text{指定シリンダ数 (整数)} \leq (A \times B \times C) \div (15 \times 96) - 5$$

(ジオメトリ情報に関しては、SCSI/Fibre Installation Manual を参照願います。)

上記計算式の結果がフォーマット指定シリンダ数以上の場合に使用して下さい。

32768 シリンダ以上のボリュームで Solaris と他 OS でオープンシステム-オープンシステム間データ交換を実施する場合 Solaris でフォーマットを行ない運用して下さい。

- (*5) OPEN-V は、1 シリンダ = 15 トラック、1 トラック = 128 サブブロック、1 サブブロック = 512 バイトです。
 しかし中間ボリュームをフォーマットでフォーマットを行なう際は 1 シリンダ = 15 トラック、1 トラック = 96 サブブロック
 1 サブブロック = 512 バイトで計算したシリンダ数を指定する必要があります。
 また、OPEN-V はデフォルト値が決まっていない為、ボリューム毎にフォーマット可能シリンダ数を算出
 する必要があります。
 Solaris で 65534 シリンダを使用する場合は、下記の値で割り当ててください。(推奨値)

(ジオメトリ情報に関しては、SCSI/Fibre Installation Manual を参照願います。)

OPEN-V 49160 シリンダのジオメトリ設定例

#	Data Cylinders- Partition Size - (1)	Heads - (2)	Blocks/Track-(3)	Usable Blocks {(1) + 2} × (2) × (3)
1	24578	30	128	94387200

- (注 4) CVS (Customizable Volume Size) 適用ボリュームの場合、フォーマットにて指定できるシリンダ容量は
 CV 設定サイズをシリンダ数に換算した値 - 7 シリンダとなります。(換算方法に関しては、SCSI/Fibre
 Installation Manual を参照願います。)
 (例) バージョン 01-XX-YY/ZZ (YY = 数字, ZZ < 20) (Large ファイル未サポート版) では OPEN-8/OPEN-9
 /OPEN-E/OPEN-M/OPEN-L の場合 CVS の設定により 4096Mbyte を指定すると、
 ボリュームを効率良く利用できます。
- (注 5) 間違ったボリュームに対しフォーマットを実行すると、そのボリュームは使用できなくなります。
 フォーマットの実行時はパラメータ指定に十分注意して下さい。
- (注 6) フォーマットとアロケータを同時に起動しないで下さい。フォーマット/アロケータが終了 (NT 版の場合は、操作
 画面を閉じる) してから、アロケータ/フォーマットを起動して下さい。
- (注 7) フォーマット実行後の中間ボリュームに対し、ボリューム容量を変更しないで下さい。アロケータが正常に
 動作出来なくなります。もしボリューム容量を変更した場合は、フォーマットからやり直して下さい。
- (注 8) UNIX 系 OS のフォーマット/アロケータで、ボリュームリアルタイムデータセット名に "¥" や "\ " 等の文字を使用
 する場合は、使用するシェルの特許文字の扱いに注意して指定して下さい。
 例えば、Bourne シェルでは "¥" や "\ " はエスケープ文字として扱われる為、"¥¥" 又は "\ \ " と指定
 する必要があります。
- (注 9) フォーマット/アロケータを使用する前に、ボリューム定義ファイルを作成しておく必要があります。
 (ボリューム定義ファイルについては、8.2 節を参照して下さい)
- (注 10) バージョン 01-XX-YY/2x (YY ≥ 48, x = 数字) (Large ファイルサポート版) でフォーマットした中間ボリュームは
 バージョン 01-XX-YY/ZZ (YY = 数字, ZZ < 20) (Large ファイル未サポート版) からはアクセスできません。
 混在環境で OTO を行う場合は、必ずバージョン 01-XX-YY/ZZ (YY = 数字, ZZ < 20) (Large ファイル
 未サポート版) のシステムを用いて、フォーマットして下さい。(ただし、このボリュームでは、4369 シリンダ
 を超えるデータセットは作成できません。) ≧ ≧

バージョンごとの機能マトリクス(01-XX-YY/ZZ)

アクセス フォーマット	YY ≥ 48 かつ ZZ ≥ 20(Large ファイルサポート版)	YY ≥ 48 かつ ZZ < 20 (Large ファイル未サポート版)	旧バージョン(YY ≤ 47)
YY ≥ 48 かつ ZZ ≥ 20(Large ファイルサポート版)	○ アロケートは1データセットあたり、65533 シリンドラまで	×アロケータ/LISTVOL/OTM/MTO 全不可	×アロケータ/LISTVOL/OTM/MTO 全不可
YY ≥ 48 かつ ZZ < 20 (Large ファイル未サポート版)で LUSE にて最大シリンドラ数以上フォーマット	○ アロケートは1データセットあたり、4369 シリンドラまで	○ 同左	×
YY ≥ 48 かつ ZZ < 20 (Large ファイル未サポート版)で OPEN-3 を 3331 シリンドラ以下にフォーマット	○	○ アロケートは1データセットあたり、4369 シリンドラまで	○
YY ≥ 48 かつ ZZ < 20 (Large ファイル未サポート版)で OPEN-K を 2536 シリンドラ以下にフォーマット	○	○ アロケートは1データセットあたり、4369 シリンドラまで	01-XX-44 以降は可能 その他×
YY ≥ 48 かつ ZZ < 20 (Large ファイル未サポート版)で OPEN-8/9 を 5818 シリンドラ以下にフォーマット	○ アロケートは1データセットあたり、4369 シリンドラまで	○ アロケートは1データセットあたり、4369 シリンドラまで	01-XX-44 以降は可能 その他×
YY ≥ 48 かつ ZZ < 20 (Large ファイル未サポート版)で、OPEN-E/L/M を 5818 シリンドラ以下にフォーマット	○ アロケートは1データセットあたり、4369 シリンドラまで	○ アロケートは1データセットあたり、4369 シリンドラまで	×

バージョン 01-XX-YY/ZZ (YY \geq 48, ZZ \geq 20) (Large ファイルポ ート版)での最大ファイルサイズを
以下に示します。中間ボリュームのレコード長/ブロック長により大きく異なります。

	ボリュームエミュレーションタイプ	LUSE なし	LUSE あり	備考
1	OPEN-3	約 2.1GB	約 42.8 GB	
2	OPEN-8	約 6.5GB	約 42.8 GB	
3	OPEN-9	約 6.5GB	約 42.8 GB	
4	OPEN-K	約 1.6GB	約 42.8 GB	
5	OPEN-E	約 12.9GB	約 42.8 GB	
6	OPEN-L	約 32.3GB	約 42.8 GB	
7	OPEN-M	約 41.9GB	約 42.8 GB	
8	OPEN-V	最大約 42.8 GB	約 42.8 GB	
9	3390-3X	約 2.6 GB		
10	3380-KX/6586-KX	約 1.7 GB		
11	3380-3	約 2.1GB		
12	3390-9/6588-9	約 8.3GB		
13	3390-L/6588-L	約 27.1GB		
14	3390-M	約 54.2GB		
15	6588-3X	約 2.6 GB		

2.3.1 UNIX系OSの場合

(1) 実行方法

以下のコマンドとパラメータによりフォーマットを実行させます。

- 32bit の場合 : # mformat -d devname -v VOLSER-name [-p primary_cylinders]
- 64bit の場合 : # mformat64 -d devname -v VOLSER-name [-p primary_cylinders]

-d devname : 中間ボリュームのフォーマットを作成する raw デバイスの名称を指定します。
本指定は省略できません。
(FAL/FCU 用のボリューム定義ファイル中の記述形式と同様です。)

-v VOLSER-name : 中間ボリュームを識別するためのボリュームシリアルナンバーを 6 桁以下の
英字 (A-Z、@、#、¥(\))、数字 (0-9) で指定します。
(FAL/FCU 用のボリューム定義ファイル中の記述形式と同様です。)

-p primary_cylinders : プライマリシリンダ数を、2 ~ 65534 の範囲の十進数で指定します。
指定は省略可能で、デフォルト値は各エミュレーションタイプの最大シリンダ数となります
(下表参照)。尚、LUSE 設定時の場合に本パラメータを省略しても LUSE 連結による最大シリンダ数は設定されず、各エミュレーションタイプの
ボリューム単位の最大シリンダ数が設定されます。また、CVS 適用ボリュームの
場合、本パラメータの省略できません。

エミュレーションタイプ	OPEN-3	OPEN-8	OPEN-9	OPEN-K	OPEN-E	OPEN-L	OPEN-M	OPEN-V
01-XX-YY/ZZ (ZZ < 20: LARGE ファイル 未サポート版)	3331	5818	5818	2536	5818	5818	5818	
01-XX-YY/ZZ (ZZ ≥ 20: LARGE ファイル サポート版)	3331	9959	10009	2536	19752	49429	63976	65534

(注) -d/-v/-p の後に付く値は各々一つだけです。

(2) リターン値

実行終了後、以下のリターン値を返します。

- 0 : 正常終了
- 1 : 異常終了

(3) メッセージ

フォーマットが正常終了した場合、完了メッセージを表示します。

処理中に、異常を検出した場合は、エラーメッセージを表示します。

(詳細は、2.5節を参照して下さい)

(4) 注意事項

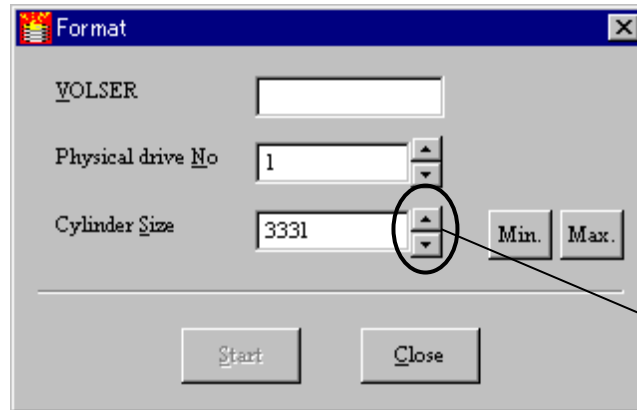
- 次のメッセージが表示された時は、フォーマットは実行されず、ボリューム上にはフォーマット実行前の状態が残ります。

「Format check error」

- 上記のメッセージが表示されない時は、フォーマット実行中(ボリュームへの書き込み中)にエラーが発生していますので、フォーマットを再度実行して下さい。

2.3.2 Windowsの場合

フォーマットのアイコンを選択してフォーマットを起動します
以下の操作画面が表示されます。



バージョン 01-XX-48
以降
スピノボタンはなくな
ります。

図2.3-1 フォーマットの操作画面

(1) **VOLSER**

中間ボリュームを識別するためのボリュームシリアルナンバ (VSN) を1~6桁で指定します。

- ・使用できる文字は、英字(A-Z、@、#、¥(\))、数字(0-9)です。
- ・ここで指定したVSNは以下のように使用します。
 - ・アロケータを実行する際の中間ボリュームを識別する情報として使用します。
 - ・FAL/FCUを実行させる際のボリューム定義ファイル中のVSNとして指定します。

(2) **Physical drive No**

フォーマットするドライブの物理ドライブ番号を指定します。

- ・1~4桁の数字で指定します。
- ・ここで指定した物理ドライブ番号を、FAL/FCUを実行させる際のボリューム定義ファイル中の物理ドライブ番号として指定します。

(3) **Cylinder No**

中間ボリュームのサイズをシリンダ数で指定します。

- ・2~ 65534(バージョンにより異なります)までの値を指定できます。

(4) **Min.**

中間ボリュームのサイズを数値で指定する代わりに最小値(2)を指定するときにクリックします。

(5) **Max.**

中間ボリュームのサイズを数値で指定する代わりにバージョン01-XX-47以前では、最大値(5818)を指定するときにクリックします。バージョン01-XX-48/ZZ以降ではエミュレーションタイプごとに異なります。

エミュレーションタイプ	OPEN-3	OPEN-8	OPEN-9	OPEN-K	OPEN-E	OPEN-L	OPEN-M	OPEN-V
ZZ < 20 (Large ファイル未サポート版)	3331	5818	5818	2536	5818	5818	5818	
ZZ ≥ 20 (Large ファイルサポート版)	3331	9959	10009	2536	19752	49429	63976	65534

(6) **Start**

必要情報設定後、フォーマットを起動するときにクリックします。
本ボタンをクリックすると処理を実行してよいか否かの確認のために以下のダイアログを表示します。
実行させる場合は「OK」を、やめる場合は「キャンセル」をクリックして下さい。

(注) 中間ファイルを作成済みのボリュームに対してフォーマットを実行すると、ファイルは全てなくなります。

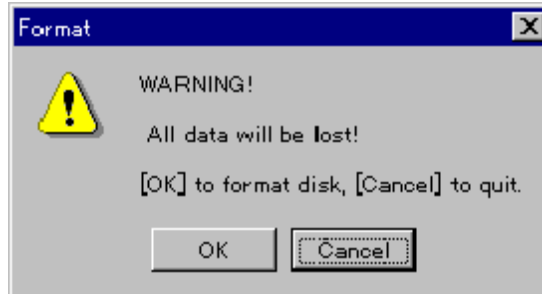


図2. 3-2 実行確認ダイアログ

指定されたVSNが他の中間ボリュームで既に使用されている場合は、
“This VOLSER is already used.”のErrorメッセージを表示します。

フォーマットの処理が正常終了した場合は、“Format complete!”のメッセージを表示します。

フォーマットの処理が異常終了した場合は、“Format Failed”のErrorメッセージを表示します。

本ダイアログで報告するエラーメッセージについては2. 5節を参照して下さい。

(注1) ダイアログ中次のメッセージが表示された時は、フォーマットは実行されず、ボリューム上にはフォーマット実行前の状態が残ります。

「Format check error」

上記のメッセージが表示されない時は、フォーマット実行中(ボリュームへの書込み中)にエラーが発生していますので、フォーマットを再度実行して下さい。

(注2) 最終シリンダ数を超えた値を指定してフォーマットを実行すると、下記のエラーが表示されます。

「Format Failed... Format check error」

Windows NT/2000Server/Server2008/Server2008 R2/Server2012/Server2012 R2/Server2016/Server2019では、上記メッセージを即時に出力しますが、Windows Server2003では、上記メッセージを約1分後に出力しますのでフォーマット完了と誤認しないで下さい。

(7) **Close**

フォーマットを終了させるときにクリックします。

2.4 中間ファイルの生成

アロケータを起動して、フォーマットでフォーマットしたボリュームに中間ファイルを作成します。
ただし、データ実体は作成しません。

(注 1) フォーマットでフォーマットしていない中間ボリュームに対して、アロケータを使用することはできません。
エラーメッセージを表示してアロケータの処理を中止します。
(詳細は、2.5 節を参照して下さい)

(注 2) アロケータにて作成可能なデータセットの数はボリュームエミュレーションタイプ / LUSE の有無にかかわらず、最大 740 個です。

(注 3) 中間ファイルに格納できるデータ量は、物理的な記憶容量より小さく、かつブロック長により変動します。
表 2.4-1 により 1トラックに格納できるデータ量を求め、中間ファイルの必要サイズを計算して下さい。
なお、レコード形式が可変長の場合、実際のブロック長が定義したブロック長よりも小さくなる場合がありますので、余裕を持たせた中間ファイルのサイズにして下さい。

表 2.4-1 ブロック長と 1トラック当りの書込み可能データ量の関係

アロケータで指定するブロック長 (バイト)	1トラックに書込み可能なデータ量 (バイト)	アロケータで指定するブロック長 (バイト)	1トラックに書込み可能なデータ量 (バイト)	アロケータで指定するブロック長 (バイト)	1トラックに書込み可能なデータ量 (バイト)
23477~32760	(ブロック長)×1	1589~1684	(ブロック長)×22	565~596	(ブロック長)×44
15477~23476	(ブロック長)×2	1493~1588	(ブロック長)×23	533~564	(ブロック長)×45
11477~15476	(ブロック長)×3	1397~1492	(ブロック長)×24	501~532	(ブロック長)×46
9077~11476	(ブロック長)×4	1333~1396	(ブロック長)×25	469~500	(ブロック長)×48
7477~9076	(ブロック長)×5	1269~1332	(ブロック長)×26	437~468	(ブロック長)×49
6357~7476	(ブロック長)×6	1205~1268	(ブロック長)×27	405~436	(ブロック長)×51
5493~6356	(ブロック長)×7	1141~1204	(ブロック長)×28	373~404	(ブロック長)×53
4821~5492	(ブロック長)×8	1077~1140	(ブロック長)×29	341~372	(ブロック長)×55
4277~4820	(ブロック長)×9	1045~1076	(ブロック長)×30	309~340	(ブロック長)×57
3861~4276	(ブロック長)×10	981~1044	(ブロック長)×31	277~308	(ブロック長)×59
3477~3860	(ブロック長)×11	949~980	(ブロック長)×32	245~276	(ブロック長)×62
3189~3476	(ブロック長)×12	917~948	(ブロック長)×33	213~244	(ブロック長)×65
2933~3188	(ブロック長)×13	853~916	(ブロック長)×34	181~212	(ブロック長)×68
2677~2932	(ブロック長)×14	821~852	(ブロック長)×35	149~180	(ブロック長)×71
2485~2676	(ブロック長)×15	789~820	(ブロック長)×36	117~148	(ブロック長)×74
2325~2484	(ブロック長)×16	757~788	(ブロック長)×37	85~116	(ブロック長)×78
2165~2324	(ブロック長)×17	725~756	(ブロック長)×38	53~84	(ブロック長)×83
2005~2164	(ブロック長)×18	693~724	(ブロック長)×39	21~52	(ブロック長)×88
1877~2004	(ブロック長)×19	661~692	(ブロック長)×40	1~20	(ブロック長)×93
1781~1876	(ブロック長)×20	629~660	(ブロック長)×41		
1685~1780	(ブロック長)×21	597~628	(ブロック長)×42		

(注1) 「書込み可能なデータ量」には、レコード形式が可変長の場合、レコード長を表す情報(レコード毎に 4 バイト)とブロック長を表す情報(ブロックに 4 バイト)を含むバイト数です。レコード形式が可変長の場合、ユーザーに加え、当該情報のエリアが必要となります。

(注 2) 本表の「書込み可能なデータ量」を 15 倍したものがシリンダ当りのデータ量になります。

2.4.1 UNIX系OSの場合

(1) 実行方法

以下のコマンドとパラメータによりアロケータを実行します。

• 32bit 版の場合：

```
# allocds { -d devname | -v VOLSER-name | -d devname -v VOLSER-name } [-n datasetname]  
[-f F/FB/V/VB] [-r reclen] [-b blocklen] [-c cylinders]
```

• 64bit 版の場合：

```
# allocds64 { -d devname | -v VOLSER-name | -d devname -v VOLSER-name } [-n datasetname]  
[-f F/FB/V/VB] [-r reclen] [-b blocklen] [-c cylinders]
```

- d devname** : フォーマットでフォーマットした中間ボリュームの raw デバイス名を指定します。
- v VOLSER-name** : 中間ボリュームを識別するためのボリュームシリアル番号を 6 桁以下の英字 (A-Z、@、#、¥(\))、数字 (0-9) で指定します。
- n datasetname** : 作成するデータセットの名称を 44 桁以下の英字 (A-Z、@、#、¥(\))、数字 (0-9) およびピリオド (.) で指定します。
中間ファイルのアロケートする場合、本指定を省略することはできません。
本指定が省略された場合、当該中間ボリュームの残容量 (シリンダ数) を表示します。
- f F/FB/V/VB** : レコード形式を指定します。
F : 固定長非ブロック化
FB : 固定長ブロック化
V : 可変長非ブロック化
VB : 可変長ブロック化
本指定は省略可能で、デフォルト値は "F" です。
- r reclen** : レコード長をバイト数 (十進数) で、1~32760 の範囲で指定します。
レコード形式=F のとき : レコード長 = ブロック長
レコード形式=FB のとき : レコード長 = (ブロック長) ÷ n n : 整数
レコード形式=V/VB のとき : $5 \leq \text{レコード長} \leq (\text{ブロック長} - 4)$
本指定は省略可能で、デフォルト値は 4096 バイトです。
- b blocklen** : ブロック長をバイト数 (十進数) で、1~32760 の範囲で指定します。
レコード形式=F のとき : ブロック長 = レコード長
レコード形式=FB のとき : ブロック長 = レコード長の整数倍
レコード形式=V/VB のとき : ブロック長 = レコード長 + 4 以上
($9 \leq \text{ブロック長} \leq 32760$)
本指定は省略可能で、デフォルト値は以下の通りです。
レコード形式=F/FB のとき : ブロック長 = レコード長
レコード形式=V/VB のとき : ブロック長 = レコード長 + 4
- c cylinders** : 作成するデータセットのサイズをシリンダ数 (十進数) で指定します。
本指定は省略可能で、デフォルト値は 100 シリンダです。
指定可能なシリンダ数は
バージョン 01-XX-YY/ZZ (YY = 数字, ZZ < 20) (Large ファイル未サポート版)
でフォーマットした場合 : 4369 シリンダ、空きシリンダの小さい方です。
バージョン 01-XX-YY/ZZ (YY ≥ 48, ZZ ≥ 20) (Large ファイルサポート版) で、
フォーマットした場合 : 空きシリンダ分

(注1) -d/-v/-n/-f/-r/-b/-cの後に付く値は、各々一つだけです。

(注2) devname と VOLSER-name パラメータは、両方もしくはどちらか一方を指定します。
アケータ対象ボリュームの指定ミスを防ぐ上で、両方指定することを推奨します。

(2) リターン値

実行終了後、以下のリターン値を返します。

0：正常終了

1：異常終了

(3) メッセージ

アケータ終了時、使用可能な残シリンダ数と完了メッセージを表示します。

処理中に、異常を検出した場合は、エラーメッセージを表示します。(詳細は、2.5節を参照して下さい)

(4) 注意事項

- 以下のメッセージが表示された時は、アケータは実行されず、ボリューム上にはアケータ実行前の状態が残ります。
「Allocate check error」
- 上記のメッセージが表示されない時は、アケータ実行中(ボリュームへの書込み中)にエラーが発生していますので、アケータを再度実行して下さい。

2.4.2 Windowsの場合

アロケータのアイコンを選択してアロケータを起動します。

アロケータ起動時、フォーマットで生成した中間ボリュームがシステムに存在する場合は図2.4-1の操作画面を表示し、中間ボリュームが存在しない場合は図2.4-2のフォーマットされたボリュームなしの画面を表示します。

フォーマットされたボリュームなしの画面が表示された場合、フォーマットが実行されているか否かチェックして下さい。アロケータで指定可能なシリンダ数/トラック数の合計は最大下記通りです。

バージョン01-XX-YY/ZZ (YY=数字, ZZ<20) (Largeファイル未サポート版) でフォーマットした場合：
4369シリンダまたは空き容量の小さい方です。

バージョン01-XX-YY/ZZ (YY≥48, ZZ≥20) (Largeファイルサポート版) でフォーマットした場合：
空き容量分(ただしバージョン01-XX-YY/ZZ (YY≥48, ZZ≥20)のみアクセス可能)

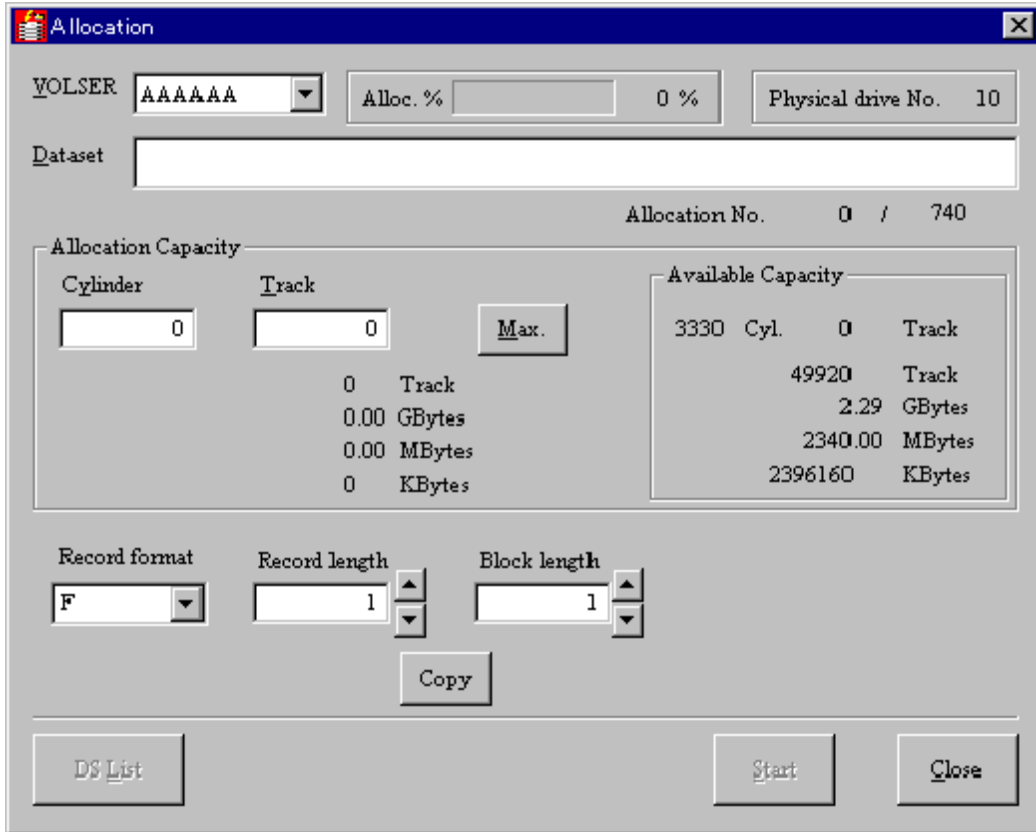


図2.4-1 アロケータの操作画面



図2.4-2 処理可能な中間ボリュームがない場合のダイアログ

操作画面は以下のようにして使用します。

(1) VOLSER

フォーマットでフォーマットした中間ボリュームの中から、アロケータで処理するボリュームを選択します。処理対象の中間ボリュームのVSNを、リストをプルダウンさせてリストの中から選択します。

(2) **Dataset**

生成する中間ファイルの名称を指定します。

- ・1～44桁の英字(A-Z、@、#、¥(\))、数字(0-9)、およびピリオド(.)で指定します。
(英小文字は英大文字に変換して表示します)

(3) **Allocation Capacity**

作成する中間ファイルのサイズを指定します。

(A) **Cylinder**

作成する中間ファイルのサイズをシリンダ数で指定します。

下記(B)のトラック数も合わせて指定すると、指定シリンダ数 + トラック数がファイルサイズとなります。

(B) **Track**

作成する中間ファイルのサイズをトラック数で指定します。

上記(A)のシリンダ数も合わせて指定すると、指定シリンダ数 + トラック数がファイルサイズとなります。

(C) **Max.**

作成する中間ファイルのサイズをシリンダ数/トラック数で指定する代わりに、最大値を指定するときにクリックします。

Available Capacity に表示されている全空きエリアがファイルエリアとして割当てられます。

(4) **Record format**

レコード形式を指定します。

以下の文字列を、リストをプルダウンさせて選択します。

- F : 固定長非ブロック化
- FB : 固定長ブロック化
- V : 可変長非ブロック化
- VB : 可変長ブロック化

(5) **Record length**

レコード長を10進数(バイト数)で指定します。

指定可能な値は以下の範囲です。

- ・レコード形式=Fのとき : レコード長 = ブロック長
- ・レコード形式=FBのとき : レコード長 = (ブロック長) ÷ n n:整数
- ・レコード形式=V/VBのとき : $5 \leq \text{レコード長} \leq (\text{ブロック長} - 4)$

(6) **Block length**

ブロック長を10進数(バイト数)で指定します。

指定可能な値は以下の範囲です。

- ・レコード形式=F/FBのとき : $1 \leq \text{ブロック長} \leq 32760$
- ・レコード形式=V/VBのとき : $9 \leq \text{ブロック長} \leq 32760$

Copy をクリックするとレコード長と同じ値がブロック長として設定されます。

(7) **Start**

必要情報の設定後、アケータを起動するときにクリックします。

アケータの処理が正常終了した場合は、”Allocation complete!”のメッセージを表示します。

アケータの処理が異常終了した場合は、”Allocating dataset failed.”のエラーメッセージを表示します。

本ダイアログで報告するエラーメッセージについては2.5節を参照して下さい。

(注1)

- ダイアログに次のメッセージが表示された時は、アケータは実行されず、ボリューム上にはアケータ実行前の状態が残ります。

「Allocate check error」

- 上記のメッセージが表示されない時は、アケータ実行中(ボリュームへの書込み中)にエラーが発生していますので、アケータを再度実行して下さい。

(注2) **DS List** は機能しません。クリックもできません。

2.5 フォーマッタ/アロケータのエラーメッセージ

フォーマッタ/アロケータが出力するエラーメッセージには以下のものがあります。

2.5.1 UNIX系OSの場合

	エラーメッセージ	内容
フ ォ ー マ ッタ	Format Failed...	フォーマット異常終了
	Format check error	フォーマットチェックでエラー検出 指定したプライマリシリンダ数を確認して下さい
	VOLSER=name error	ボリューム名称に不正な文字が指定されています
	Primary cylinders error primary_cylinders=[2- 5818]	プライマリシリンダの指定範囲エラー または不正な文字が指定されています
	The number of cylinder is too large	指定したシリンダ容量がありません
	Available volume is not found for formatting	フォーマット可能なボリューム (OPEN-X) がありません
ア ロ ケ ー タ	Allocating dataset failed...	アロケート異常終了
	Allocate check error	アロケートチェックでエラー検出 指定したシリンダ数を確認して下さい
	Record_format error	レコード形式指定不正
	Record_length error Record_length=[指定範囲]	レコード長指定不正
	Block_length error Block_length=[指定範囲]	ブロック長指定不正
	Cylinders error	シリンダ数に数値以外の文字が指定されています。又は指定可能範囲外の数値が指定されています。
	Cylinders Large error	シリンダ数が6桁以上指定されています。又は指定可能範囲外の数値、文字が指定されています。
	Dataset name error (too long)	データセット名が、44文字を越えて指定されています
	Dataset name error (Invalid letter)	データセット名に不正な文字が指定されています。又はデータセット名が指定されていません
	This device is not formatted	指定したデバイスがフォーマットされていません
	Dataset is full	作成可能なデータセットの数を越えています
	This dataset is already exists	同一のデータセット名が既に存在しています
	The number of cylinders is too large	指定したシリンダ容量がありません
	Available volume is not found for allocating	アロケート可能なボリューム (OPEN-X) がありません
The VSN of allocating volume is disagreement	指定された (又はボリューム定義ファイル内の) VSNと、ボリューム上のVSNが不一致です	
共 通	Volume definition file is not found	ボリューム定義ファイルがありません
	Volume definition file has no data	ボリューム定義ファイルに定義がされていません
	Volume definition file is not valid data	ボリューム定義ファイル内のパラメータ数が不正です
	The devname and/or VOLSER-name is not found in volume definition file	指定したデバイス名/ボリューム名称、又はその組合せに一致するものが、ボリューム定義ファイルにありません
	The partition name is invalid in volume definition file	ボリューム定義ファイル内のパーティション名が不正です
	Partition name length error in volume definition file	ボリューム定義ファイル内のパーティション名が長すぎます (1024文字以内)
	VSN length error in volume definition file	ボリューム定義ファイル内のVSNの指定が長すぎます (6文字以内)
	Emulation type length error in volume definition file	ボリューム定義ファイル内のエミュレーションタイプの指定が長すぎます (10文字以内)
	Volume definition file record length error	ボリューム定義ファイルの1レコードのサイズが長すぎます (2081文字以内)
	Volume definition file read error	ボリューム定義ファイルのリードに失敗しました (再起動して下さい)
	Volume definition file close error	ボリューム定義ファイルのクローズに失敗しました (再起動して下さい)

2.5.2 Windowsの場合

	エラーメッセージ	内容
フォーマッタ	Format Failed...	フォーマット異常終了
	Format check error	フォーマットチェックでエラー検出 指定したプライマリシリンダ [®] 数を確認して下さい
	Available volume is not found for formatting	フォーマット可能なボリューム(OPEN-X)がありません
アロケータ	Allocating dataset failed..	アロケート異常終了
	Allocating dataset failed.. Allocate check error	アロケートチェックでエラー検出 指定したシリンダ [®] 数を確認して下さい
	There is no volume for allocating!	フォーマットされているボリュームがありません
	This VOLSER is already exists	同一のVOLSER名が既に存在しています
	This dataset is already exists	同一のデータセット名が既に存在しています
	Available volume is not found for allocating	アロケート可能なボリューム(OPEN-X)がありません
	Format information is error	指定したボリュームの情報が不正です ボリュームをフォーマットして下さい
	The VSN of allocating volume is disagreement	指定された(又はボリューム定義ファイル内の)VSNと、ボリューム上のVSNが不一致です
共通	Volume definition file is not found	ボリューム定義ファイルがありません
	Volume definition file has no data	ボリューム定義ファイルに定義がされていません
	The devname and/or VOLSER-name is not found in volume definition file	指定したデバイス名/ボリューム名称、またはその組み合わせに一致するものが、ボリューム定義ファイルにありません
	Partition name length error in volume definition file	ボリューム定義ファイル内のパーティション名が長すぎます。(1024文字以内)
	VSN length error in volume definition file	ボリューム定義ファイル内のVSNの指定が長すぎます(6文字以内)
	Emulation type length error in volume definition file	ボリューム定義ファイル内のエミュレーションタイプ [®] の指定が長すぎます(10文字以内)
	Volume definition file record length error	ボリューム定義ファイルの1レコードのサイズが長すぎます(2081文字以内)
	Volume definition file read error	ボリューム定義ファイルのリード [®] に失敗しました(再起動して下さい)

3 FAL のインストール

3.1 FAL/FCU メモリ使用量

- ・FAL/FCU 起動時のメモリ使用量は約 20MB となります。
- ・FCU (GUI 版) を使用すればさらに約 5MB となります。

3.2 FAL のインストール

3.2.1 32bit用FAL/FCUのインストール/アンインストール

32bit版及び64bit版対応のFAL/FCUをインストール/アンインストールする場合、本章に示す手順に従って行って下さい。
(FAL/FCU64bit版は、01-XX-59版以降の対応となります。)

Code Converter^(注)と連携される場合は次の点に注意して下さい。

- ・FAL及びCode Converterをインストールする場合：
→FALをインストール後、Code Converterをインストールして下さい。
- ・FAL及びCode Converterを共にアンインストールする場合：
→Code Converterのアンインストール後にFALをアンインストールして下さい。
- ・FAL及びCode Converterがインストールされている状態でFALをバージョンアップする場合：
→3.3章を参照して下さい。

(1) UNIX系OSの場合

[インストール方法]

以下の手順によりインストールを行いますが、旧バージョンがインストールされている場合には、最初に旧バージョンをアンインストールする必要があります。後述のアンインストールの手順により旧バージョンのアンインストールを実施した後、新バージョンのインストールを行って下さい。

- ① rootでログインする。
- ② CD-ROMをドライブに入れてマウントする。(CD-ROMはiso9660にて作成しています。)
- ③ 以下のディレクトリの存在を確認し、なければ当該ディレクトリを作成する。
/usr, /usr/bin, /usr/include, /usr/lib, /usr/lib/X11, /usr/lib/X11/app-defaults
(例) ディレクトリX11の作成 : # mkdir /usr/lib/X11
ディレクトリ app-defaultsの作成 : # mkdir /usr/lib/X11/app-defaults
- ④ ルートディレクトリに移動する。
- ⑤ FAL/FCUをCD-ROMからインストールする。
CD-ROMのマウントの仕方によりファイル名/ディレクトリ名の大文字/小文字の見え方が異なり、またファイル名に‘;’等が付加される場合(この場合ファイル名をシングルクォーテーション(‘’)で囲む必要あり)があります。必ず、lsコマンドにて確認してください。
次にFAL/FCUインストール指定例を示します。

FAL/FCUインストール指定例

OS ビット	対象OS		インストール指定例	備考
32	AIX		<code>cpio -iBmuv <(MountPoint)/PROGRAM/HMDE/AIX/AIX.CPI</code>	—
	Solaris		<code>cpio -iBmuv <(MountPoint)/PROGRAM/HMDE/SOLARIS/SOLARIS.CPI</code>	—
	Linux (CPUタイプ)	(x86、AMD64/ Intel64共通)	<code>cpio -iBmuv <(MountPoint)/PROGRAM/HMDE/LINUX/LINUX.CPI</code>	—
64	AIX	AIX5L AIX6.1 AIX7.1 AIX7.2	<code>cpio -iBmuv <(MountPoint)/PROGRAM/FAL64/AIX/AIX5/AIX.CPI</code>	—
	HP-UX (CPUタイプ)	(Itanium)	<code>cpio -iBmuv <(MountPoint)/PROGRAM/FAL64/HP_UX/IA/HP_UX.CPI</code>	—
	Solaris		<code>cpio -iBmuv <(MountPoint)/PROGRAM/FAL64/SOLARIS/SOLARIS.CPI</code>	—
	Linux	AS3.0 AS4.0 AS5.1-AS5.4 AS6.1-AS6.4 SuSE	<code>cpio -iBmuv <(MountPoint)/PROGRAM/FAL64/LINUX/AS3/LINUX.CPI</code>	—

- ・Solarisの場合、FCUを使用するユーザ対応に、リソースファイルへのパスの設定が必要です。
下記の例を参考に設定して下さい。(パスの設定は一度行えば次回からは不要です)
 - ・Cシェルの場合、以下の行をホームディレクトリのファイル「.cshrc」の最終行として追加する。
(「.cshrc」がない場合は、作成して記述する)

```
setenv XFILESEARCHPATH /usr/lib/X11/app-defaults/%N:$XFILESEARCHPATH
```
 - ・Cシェル以外の場合、以下の2行をホームディレクトリのファイル「.dtprofile」の最終行として追加する。
(「.dtprofile」がない場合は、作成して記述する)

```
XFILESEARCHPATH=/usr/lib/X11/app-defaults/%N:$XFILESEARCHPATH
export XFILESEARCHPATH
```
- 注：共通デスクトップ環境がないマシンでは、「.profile」に上記行を追加してください。
ファイルがない場合は作成してください。
- なお、上記の設定後に一度ログアウトし、ログイン直して下さい。

- ⑥ アンマウントして、CD-ROMをドライブから取り出す。(solarisでは、ejectコマンドで、CD-ROMを取り出す場合もあります。)
- ⑦ 一度ログアウトし、再度ログインする。

(補足)

32/64bit版のFAL/FCUがインストールされるファイル一覧

	ディレクトリ名	ファイル名		説明	備考
		32bit版	64ビット版		
1	/usr/bin	fcu(*3)	—	FCU(GUI版)	
2		fcunw	fcunw64	FCU(GUIなし版)	
3		listvol	listvol64	データセット一覧表示	
4		mfformat	mfformat64	フォーマット	
5		allocds	allocds64	アロケータ	
6		ppkeyset	ppkeyset64	ライセンスキーワード設定コマンド	
7		autopkeyset	autopkeyset64	ライセンスキーファイル 入力コマンド	Ver. 01-XX-56/ZZ 以降サポート
8	/usr/lib	libuoc.* (*1)(*2)	libuoc64.* (*1)(*2)	HMDE/KANJI用 タミライブラリ	拡張子はOSに より異なる (#10はVer. 01- XX-59/20以降サ ポート(AIX))
9		libfal.*(*2)	libfal64.*(*2)	FAL	
10		libfalmt.*	—	FALマルチスレッド用	
11		libfal.ver	libfal.ver	FALのVerファイル	
12	/usr/lib /X11/app-defaults	FcuMf	—	GUIファイル	
13	/usr/include	dataset.h	dataset.h	ヘッダファイル	

*1 : libuoc.*及びlibuoc64.*ファイルは、HMDE/KANJIをインストールするとHMDE/KANJIのライブラリに置換わります。HMDE/KANJIをインストールする場合は、libuoc.*及びlibuoc64.*ファイルを別名で保存してからインストールして下さい。

*2 : Solaris及びHP-UXの場合は/usr/libディレクトリ配下のOS固有のディレクトリにファイルがインストールされます。以下を参照して下さい。

・64bit版Solarisの場合 : /usr/lib/sparcv9

・64bit版HP-UX(Itanium)の場合 : /usr/lib/hpux64

*3 : Linuxの場合は当該ファイルはインストールされません。fcunwコマンドのみのサポートとなります。

[アンインストール方法]

FAL/FCUの旧バージョンがインストールされている場合、以下の手順で旧バージョンをアンインストールして下さい。

① インストーラを使用してFDからインストールした場合 (Ver. 01-01-31より前のバージョンが対象です)

インストール時のユーザ IDでログインし、当該FDに格納されているアンインストールを起動してアンインストールする。
(アンインストール手順は当該バージョンに対応するユーザズガイドに記述してあります)
なお、旧バージョンのFDがない場合、下記②と同様の手順でアンインストールして下さい。

② DATからtarコマンドによりインストールした場合 (Ver. 01-01-31より前のバージョンが対象です)

インストール時のユーザ IDでログインし、fcu/fal.o/dataset.hのディレクトリに移動して、以下のコマンドによりアンインストールする。

```
# rm fcu fal.o dataset.h $HOME/FcuMf
```

なお、旧バージョンのディレクトリが不明の場合は、以下の手順により検索します。

・rootでログインする。

・以下のコマンドを実行する。

```
# find / -name "fcu" -print
```

```
# find / -name "fal.o" -print
```

```
# find / -name "dataset.h" -print
```

```
# find / -name "FcuMf" -print
```

- ③ cpioコマンドによりインストールした場合 (Ver. 01-01-31以降のバージョンが対象です)
 通常、バージョンアップ時はアンインストール手順は不要です。旧バージョンに新バージョンをオーバーライトします。
 ただし、バージョンダウン時はアンインストールが必要となります。
 アンインストールだけを実施したい場合は以下の手順で行います。

- ・rootでログインする。
- ・以下のコマンドを実行する。

32bit版FAL/FCUの場合	64bit版FAL/FCUの場合
・Linuxの場合 —	—
・それ以外の場合 # rm /usr/bin/fcu	
# rm /usr/bin/fcunw	# rm /usr/bin/fcunw64
# rm /usr/include/dataset.h	# rm /usr/include/dataset.h
# rm /usr/lib/libfal.*	# rm /usr/lib/libfal.ver ・AIX, Linuxの場合 # rm /usr/lib/libfal64.* ・HP-UX(Itanium)の場合 # rm /usr/lib/hpux64/libfal64.so ・Solarisの場合 # rm /usr/lib/sparcv9/libfal64.so.1
・AIX(Ver. 01-XX-59/20以降)の場合 # rm /usr/lib/libfalmt.a	—
# rm /usr/lib/libuoc.*	・AIX, Linuxの場合 # rm /usr/lib/libuoc64.* ・HP-UX(Itanium)の場合 # rm /usr/lib/hpux64/libuoc64.so ・Solarisの場合 # rm /usr/lib/sparcv9/libuoc64.so.1
# rm /usr/lib/X11/app-defaults/FcuMf	—

HI-UX/WE2以外の、Ver. 01-01-40以降のバージョンでは、以下の操作が必要です。

32bit版FAL/FCUの場合	64bit版FAL/FCUの場合
# rm /usr/bin/mfformat	# rm /usr/bin/mfformat64
# rm /usr/bin/allocds	# rm /usr/bin/allocds64

Ver. 01-01-41以降のバージョンでは、さらに以下の操作が必要です。

32bit版FAL/FCUの場合	64bit版FAL/FCUの場合
# rm /usr/bin/listvol	# rm /usr/bin/listvol64

SANRISE SERIES対応版及びUSP V, VSP, VSP G1000, VSP 5000対応版ではさらに下記操作が必要です。
 (他プロダクトでも使用している場合は削除しないで下さい。)

32bit版FAL/FCUの場合	64bit版FAL/FCUの場合
# rm /usr/bin/ppkeyset	# rm /usr/bin/ppkeyset64
# rm /usr/bin/autopkeyset	# rm /usr/bin/autopkeyset64

(2) Windowsの場合

[インストール方法]

(注) 旧バージョンのFCUが既にインストールされている場合は、先にFCUのアンインストールを行って下さい。(後述のアンインストール方法を参照)

[SANRISE SERIES 及び USP V シリーズ VSP シリーズ VSP G1000 シリーズ VSP 5000 シリーズ]

- ① 提供CD-ROMをCD-ROMドライブにセットして下さい。
- ② 提供CD-ROMの中の「[CD-ROMドライブ]¥PROGRAM¥HMDE¥WIN_NT¥DISK1¥setup.exe」をダブルクリックするとインストールが始まります。
画面の案内に従って完了するまで操作して下さい。

(補足)

インストールされるファイル一覧

	ディレクトリ名	ファイル名		説明	備考
		32bit版	64bit版		
1	C:¥FCU (デフォルト)	fcu.exe	fcu64.exe	FCU(GUI版)	
2		fcunw.exe	fcunw64.exe	FCU(GUIなし版)	Ver. 01-XX-52/ ZZ 以降
3		format.exe	format64.exe	フォーマット	
4		listvol.exe	listvol64.exe	データセット一覧表示	Ver. 01-XX-66/ 21 以降 (XX≥05)
5		alloc.exe	alloc64.exe	アロケータ	
6		ppkeyset.exe	ppkeyset64.exe	ライセンスキーコード設定コマンド	
7		autopkeyset.exe	autopkeyset64.exe	ライセンスキーファイル入力コマンド	Ver. 01-XX-56/ ZZ 以降
8		fal.dll	fal64.dll	FALのDLLファイル	
9		falmt.dll	-		
10		fal.lib	fal64.lib	FALのLIBファイル	
11		falmt.lib	-	FALのマルチスレッド用LIBファイル	
12		uoc.dll (*1)	uoc64.dll	HMDE/KANJI用 タミライブラリ	
13		dataset.h	dataset.h	ヘッダファイル	
14	%SystemRoot% (環境変数)	libfal.ver	-	FALのVerファイル	
		-	libfal.ver		

注：*1 uoc.dllファイルは、HMDE/KANJIをインストールするとHMDE/KANJIのライブラリに置換わります。
HMDE/KANJIをインストールする場合は、uoc.dllを別名で保存してからインストールして下さい。

[アンインストール方法]

- ① 「スタート」ボタンを押して「設定」の「コントロールパネル」を選択して下さい。
- ② 「アプリケーションの追加と削除」アイコンをクリックし、「インストールと削除」タブのソフトウェア一覧の中から「FCU」を選択して下さい。
- ③ 「追加と削除」をクリックするとアンインストールが始まります。画面の案内に従って完了するまで操作して下さい。

- 注 ① Ver. 01-01-33/00 以前の場合は、インストーラを使用していないので、インストール時に作成したフォルダごと手動で削除して下さい。
バージョンダウンを行う場合、C:¥WINNTディレクトリ内にlibfal.verファイルが残っていると誤動作する可能性がある為、必ず削除してください。
- ② HMDE/KANJIをインストールしている場合、次の点に注意して下さい。
FALをアンインストールする前に、必ずインストールしてあるフォルダ毎バックアップを取って下さい。
FALをアンインストールした時に、HMDE/KANJIのファイルも一緒に削除されてしまった場合HMDE/KANJIを再度インストールして下さい。もしHMDE/KANJIのファイルを更新して使用していた場合はインストール後に、バックアップからそのファイルに戻して下さい。

3.2.2 64bit用FAL/FCUのインストール/アンインストール

(1) UNIX の場合

[インストール方法]

64bit用のFAL単体(ライブラリ単体)のインストールは01-XX-58版以前の対応となります。01-XX-59版以降で64bit対応をインストールする場合は、FAL/FCU(FCU込み)のインストールとなります。以下を参照して下さい。

- ① rootでクイックインストールする。
- ② CD-ROMをドライブに入れてマウントする。(CD-ROMはiso9660にて作成しています。)
- ③ 以下のディレクトリの存在を確認し、なければ当該ディレクトリを作成する。

/usr, /usr/include, /usr/lib	: 全プラットフォーム共通
/usr/lib/hpux64	: HP-UX(Itanium)の場合
/usr/lib/sparcv9	: Solarisの場合
- ④ ルートディレクトリに移動する。
- ⑤ FAL(64bit)をCD-ROMからコピーする。
CD-ROMのマウントの仕方によりファイル名/ディレクトリ名の大文字/小文字の見え方が異なり、またファイル名に‘;’等が付加される場合(この場合ファイル名をシングルクォーテーション(‘’)で囲む必要あり)があります。必ず、lsコマンドにて確認して下さい。以下に指定例を示します。

注：AIX版64bitFALは、V4.3.3とV5.xで互換性がありません。

インストールするにはOSのバージョンにより、対象ファイル(ディレクトリ)を使い分けてください。

- HP-UX(Itanium)の場合
cpio -iBmuv <(MountPoint)/PROGRAM/FAL64/HP_UX/IA/HP_UX.CPI
- Solarisの場合
cpio -iBmuv <(MountPoint)/PROGRAM/FAL64/SOLARIS/SOLARIS.CPI
- AIX(V5.x, V6.x, V7.x)の場合
cpio -iBmuv <(MountPoint)/PROGRAM/FAL64/AIX/AIX5/AIX.CPI
- Linux(AS3.0, AS4.0, AS5.1-AS5.4, AS6.1-AS6.4, SuSE)の場合
cpio -iBmuv <(MountPoint)/PROGRAM/FAL64/LINUX/AS3/LINUX.CPI

- ⑥ アンマウントして、CD-ROMをドライブから取り出す。(solarisでは、ejectコマンドで、CD-ROMを取り出す場合もあります。)
- ⑦ 一度クイックアンインストールし、再度クイックインストールする。

[アンインストール方法]

64bit用のFAL単体(ライブラリ単体)のアンインストールは01-XX-58版以前の対応となります。以下を参照して下さい。
(01-XX-59版以降で64bit対応をアンインストールする場合は、FAL/FCU(FCU込み)のアンインストールとなります。)

① rootでログインする。

② 以下のコマンドを実行する。

- # rm /usr/bin/ppkeyset64
- # rm /usr/include/dataset.h
- # rm /usr/lib/libfal.ver
- HP-UX (Itanium) の場合
 - # rm /usr/lib/hpux64/libfal64.so
- Solaris の場合
 - # rm /usr/lib/sparcv9/libfal64.so.1
- AIX の場合
 - # rm /usr/lib/libfal64.a
- Linux の場合
 - # rm /usr/lib/libfal64.so.1

(2) Windows の場合

[インストール方法]

(注) 旧バージョンの **FCU** が既にインストールされている場合は、先に **FCU** のアンインストールを行って下さい。(後述のアンインストール方法を参照)

[USP V シリーズ VSP シリーズ VSP G1000 シリーズ VSP 5000 シリーズ]

① 提供CD-ROMをCD-ROMドライブにセットして下さい。

② Itanium版の場合

提供CD-ROMの中の「[CD-ROMドライブ]¥PROGRAM¥FAL64¥WIN¥IA¥WIN.MSI」をダブルクリックするとインストールが始まります。

画面の案内に従って完了するまで操作して下さい。

③ x64版の場合

提供CD-ROMの中の「[CD-ROMドライブ]¥PROGRAM¥FAL64¥WIN¥X64¥WIN.MSI」をダブルクリックするとインストールが始まります。

画面の案内に従って完了するまで操作して下さい。

[アンインストール方法]

① 「スタート」ボタンを押して「設定」の「コントロールパネル」を選択して下さい。

② Itanium版の場合

「プログラムの追加と削除」アイコンをクリックし、「プログラムの変更と削除」タブの現在インストールされているプログラムの中から「**FALFCU for Windows IA64**」を選択して下さい。

③ x64の場合

「プログラムの追加と削除」アイコンをクリックし、「プログラムの変更と削除」タブの現在インストールされているプログラムの中から「**FALFCU for Windows x64**」を選択して下さい。

④ 「削除」をクリックするとアンインストールが始まります。画面の案内に従って完了するまで操作して下さい。

3.2.3 ライセンスコードについて

SANRISE SERIES 用システム及び USP V シリーズ VSP シリーズ VSP G1000 シリーズ VSP 5000 シリーズ用システムでは、FAL/FCU/LISTVOL/LISTVOL64/FCUNW/FCUNW64 を動作させるにはライセンスコードを設定する必要があります。ライセンス設定コマンドを用いて、ライセンスコードを設定して下さい。(SANRISE SERIES 時及び USP V シリーズ VSP シリーズ VSP G1000 シリーズ VSP 5000 シリーズ時も、フォーマット/アロケータはライセンス不要)

(注)

- 01-XX-55以前のバージョンでのライセンス設定は、ホスト名設定後に行って下さい。ライセンス設定後ホスト名を変更するとエラーとなります。
HI-UXの時、myhostname.rcファイルにホスト名を設定していないと、エラーとなる場合があります。
- AIXの時、プログラム動作後、ライセンス設定を行うとエラーとなることがあります。slibcleanコマンド発行後、再度ライセンス設定の操作を行って下さい。
- ライセンスコードの設定は、新規インストール及びバージョンアップ/ダウン時にも必要となります。
- USP Vシリーズ (01-05-YYのバージョン)でのライセンスコードは以下のようになります。
 - File ExchangeとCode Converterの場合
『Hitachi Cross-OS File Exchange software (P-012R-JC1S1K)』
 - File Exchange Optionの場合 (HULFTオプション用)
『Hitachi Cross-OS File Exchange software Option set (P-F012R-JC11SK)』
 - Code ConverterOptionの場合 (Code Converter-EUC, CODECHK, UNICODE用)
『Hitachi Cross-OS File Exchange software Code Converter Option set (P-F012R-JC12SK)』
- VSPシリーズ (01-06-YYのバージョン)、VSP G1000シリーズ (01-07-YYのバージョン)及びVSP 5000シリーズ (01-08-YYのバージョン)でのライセンスコードは以下のようになります。
 - File ExchangeとCode Converterの場合
『Hitachi Cross-OS File Exchange software (P-012R-JDS1K)』
 - File Exchange Optionの場合 (HULFTオプション用)
『Hitachi Cross-OS File Exchange software Option set (P-F012R-JE11SK)』
 - Code ConverterOptionの場合 (Code Converter-EUC, CODECHK, UNICODE用)
『Hitachi Cross-OS File Exchange software Code Converter Option set (P-F012R-JE12SK)』
- FAL HULFTオプションは32bit版のみであるため、ライセンスコード設定には必ずFAL 32bit版のライセンス設定コマンドを使用してください。FAL 64bit版対応のコマンド (ppkeyset64, autoppkeyset64)でのライセンス設定はできません。

(1) ライセンスコード エラー

「FCU/FCUNW/FCUNW64/LISTVOL/LISTVOL64の場合」

- ライセンスコードを設定せずに、起動すると、下記エラーメッセージが表示されます。
Couldn't permit execution that does not have a software license.
- ソフトウェアライセンス期限を超過しているか、このホストの時刻を不正に戻すと、下記メッセージが表示されます。
Couldn't permit execution that had an expired trying time limit of a software license.
- インストールされたホスト上で動作していないか、ホスト名を変更すると下記メッセージが表示されます。ホスト名を変更した場合、再度FAL/FCUをインストールして、ライセンスコードを設定して下さい。(01-XX-55 以前)
Couldn't permit execution that this host and a host installed are not identical and/or hostname is changed.
- FCU/FCUNW/FCUNW64 立ち上がり後、途中でソフトウェアライセンスの期限が超過すると、次の転送/データセット一覧表示のタイミングで、FAL エラーが返ります。
「FAL の場合(UP等)」FAL エラーが返ります。

3.2.4 ライセンスコードの設定方法

(1) ライセンスコード設定方法 (ライセンス直接入力)

32bit版及び64bit版対応のFAL/FCUのライセンスを設定することにより動作可能とします。
ライセンスコード設定コマンドにライセンスコードを直接入力することにより、ライセンスを設定します。
(FAL/FCU64bit版は、01-XX-59版以降の対応となります。対応OSについては1.2.2.8節を参照して下さい。)

「UNIXの場合」 (32/64bit版)

コマンドラインより、下記に示す通りにppkeysetコマンドを起動して下さい。

①HP-UXの場合

- 64bit版 (Itanium)

```
#ppkeyset64 <ライセンスコード> /usr/bin/fcunw64 /usr/lib/hpux64/libfal64.so
```

②Solarisの場合

- 32bit版

```
#ppkeyset <ライセンスコード> /usr/bin/fcunw /usr/bin/fcu /usr/lib/libfal.so.1
```

- 64bit版

```
#ppkeyset64 <ライセンスコード> /usr/bin/fcunw64 /usr/lib/sparcv9/libfal64.so.1
```

③Linuxの場合

- 32bit版 (x86/AMD64/Intel64共通)

```
#ppkeyset <ライセンスコード> /usr/bin/fcunw /usr/lib/libfal.so.1
```

- 64bit版 (AMD64/Intel64共通)

```
#ppkeyset64 <ライセンスコード> /usr/bin/fcunw64 /usr/lib/libfal64.so.1
```

④AIXの場合

- 32bit版 (01-XX-58版以前)

```
#ppkeyset <ライセンスコード> /usr/bin/fcunw /usr/bin/fcu /usr/lib/libfal.a
```

- 32bit版 (01-XX-59版以降)

```
#ppkeyset <ライセンスコード> /usr/bin/fcunw /usr/bin/fcu /usr/lib/libfal.a  
/usr/lib/libfalmt.a
```

- 64bit版

```
#ppkeyset64 <ライセンスコード> /usr/bin/fcunw64 /usr/lib/libfal64.a
```

「UNIXの場合」 (64bit版FAL)

コマンドラインより、下記に示す通りにppkeyset64コマンドを起動して下さい。

64bit版のFAL単体(ライブラリ単体)のライセンスセットは01-XX-58版以前の対応となります。

01-XX-59版以降64bit対応をライセンスセットする場合は、FAL/FCU(FCU込み)のライセンスセットとなります。

- ppkeysetコマンドでは、ライセンスコードを設定できません。エラーとなります。

① HP-UXの場合

```
#ppkeyset64 <ライセンスコード> /usr/lib/hpux64/libfal64.so
```

② Solarisの場合

```
#ppkeyset64 <ライセンスコード> /usr/lib/sparcv9/libfal64.so.1
```

③ AIXの場合

```
#ppkeyset64 <ライセンスコード> /usr/lib/libfal64.a
```

④Linuxの場合

```
#ppkeyset64 <ライセンスコード> /usr/lib/libfal64.so.1
```

[Windows系の場合] (32/64bit版)

①FAL/FCUインストール先ディレクトリ内の、ppkeyset.exe(32bit版)またはppkeyset64.exe(64bit版)をダブルクリックします。

②MS-DOSプロンプト画面が表示され、以下のメッセージが表示されます。

Please enter <License key> [Installed Directory or File]... or <quit> for exit.

③<ライセンス> <FAL/FCUインストール先ディレクトリ> と入力してEnterキーを押下して下さい。但しディレクトリ名にスペースを含む場合、直接ファイル名を指定してください。(<ライセンス> fal.dll falmt.dll) (32bit版)
(<ライセンス> fal64.dll) (64bit版)

④再度Please enter <License key> [Installed Directory or File]... or <quit> for exit.が表示されますので、”quit”を入力してEnterキーを押下して下さい。

ppkeyset 正常終了時メッセージ

項番	エラーメッセージ	内容
1	set in available for all days.	無制限のライセンスを設定した場合表示します。
2	set in available for xxx days.	期間限定のライセンスを設定した場合表示します。 (注) xxxは日数

ppkeyset エラーメッセージ

項番	エラーメッセージ	内容
1	Invalid <License key>:Decode error.	ライセンスコードが間違えている時に表示します。
2	skipped for inequality to PP_CODE[xxx].	当該P.P.以外のライセンスで実行した場合に表示します。 本メッセージの場合、ライセンスは入っていない状態です。 xxx:PPID
3	available for all days on hostname on Ser# xxxxx.	既にライセンスコードが設定されている時に表示します。 xxxxx:シリアルナンバー
4	available for d:hh Hr on hostname on Ser# xxxxx.	ライセンスの残り有効期間を表示します。 d:日数 hh:時間 xxxxx:シリアルナンバー
5	Couldn't find PP_KEY on [xxx].	ライセンスの機能が組み込まれていないファイルを指定した場合に表示します。 xxx:ディレクトリ or ファイル名

(2) ライセンスキーコード設定方法 (ライセンスファイル入力)

ライセンスファイル入力コマンドに、ライセンスファイルを指定することにより、ライセンスを設定します。
この場合、ライセンスファイルを予めサーバ上に入れておく必要があります。

(注) 本コマンドが実行されると、実行結果がメッセージで表示されます。(下記表示例参照)
コマンド実行時に、メッセージが何も表示されなかった場合は、ライセンスは設定されていません。
この場合は、次の要因が考えられます。

- /usr/bin/fcunw /usr/bin/fcu /usr/lib/libfal.xx が存在しない場合
- /usr/bin/fcunw64 /usr/lib/libfal64.xx が存在しない場合
- ライセンスファイル内に、当該ライセンスが入っていない場合
- 指定したライセンスファイルのファイル形式が異なっている場合

「UNIXの場合」

コマンドラインより、下記に示す通りに autoppkeyset コマンドを起動して下さい。

```
# autoppkeyset <ライセンスファイル> (32bit版)
```

```
# autoppkeyset64 <ライセンスファイル> (64bit版)
```

(注) ライセンスファイルが autoppkeyset と異なるディレクトリに存在する場合は、絶対パスを指定して下さい。

(注) autoppkeyset コマンドでは libfalmt.* に対するライセンス設定は行いません。ppkeyset コマンドを使用して下さい。

(例) 正常終了時の画面表示例

```
# autoppkeyset /FCU/12345.plk (12345.plk : ライセンスファイル)
PP=xxx (注1)PP=xxx : xxxはPPIDが表示されます。
ppkeyset 12345678901234567890 /usr/bin/fcu
[/usr/bin/fcu] set in available for all days.
-----
PP=xxx
ppkeyset 12345678901234567890 /usr/bin/fcunw
[/usr/bin/fcunw] set in available for all days.
-----
PP=xxx
ppkeyset 12345678901234567890 /usr/lib/libfal.xx (注2) libfal.xx : xxはプラットフォームにより異なります。
[/usr/lib/libfal.so.1] set in available for all days.
-----
#
```

「Windows系の場合」

コマンドプロンプト (DOS窓) を開き、下記に示す通りに autoppkeyset.exe コマンドを起動して下さい。

```
autoppkeyset <ライセンスファイル> (32bit版)
```

```
autoppkeyset64 <ライセンスファイル> (64bit版)
```

(注) ライセンスファイルが autoppkeyset.exe と異なるディレクトリに存在する場合は、絶対パスを指定して下さい。

(注) autoppkeyset.exe コマンドでは falmt.dll に対するライセンス設定は行いません。ppkeyset.exe コマンドを使用して下さい。

(例) 正常終了時の画面表示例

```
C:\FCU> autoppkeyset 12345.plk (12345.plk : ライセンスファイル)
PP=xxx (注1)PP=xxx : xxxはPPIDが表示されます。
ppkeyset ABCDEFGHIJKLMNOPQRST fal.dll
[fal.dll] set in available for all days.
-----
C:\FCU>
```

autopksetコマンド実行時にエラーとなった場合に、次のメッセージを表示します。

項番	エラーメッセージ	内 容
1	autopkset [PPID FILE]	autopksetコマンドのみで実行しました。 この場合、コマンドの指定方法を表示します。
2	Too many arguments.	autopksetコマンドに指定した引数が多い。 コマンドに指定できる引数はライセンスファイルのみです。
3	[PPID FILE] does not exist.	ライセンスファイルが存在しません。 サーバにライセンスファイルを入れて、コマンドを実行して下さい。
4	[PPID FILE] was not opened.	ライセンスファイルがオープン出来ません。 再度、コマンドを実行して下さい。
5	Defined line in [PPID FILE] is too long.	ライセンスファイル内の1行の長さが256バイト以上です。 ライセンスファイルを参照し、1行の長さを確認して下さい。
6	ppkset failed	ppksetコマンドが失敗しました。 再度、コマンドを実行して下さい。
7	ppkset64 failed	ppkset64コマンドが失敗しました。 再度、コマンドを実行して下さい。

項番 3, 4, 5 の[PPID FILE]は指定したライセンスファイル名を表示します。
その他のエラーメッセージはppksetのエラーメッセージを参照してください。

3.2.5 バージョン確認方法

(1) UNIX 系 (GUI 版)

8. 6. 1 UNIX 系 OS の場合 (2) (V) を参照して下さい。

(2) UNIX 系 (コマンドライン版)

ボリューム定義ファイルが存在しているのを確認し、fcunw[return]

(3) Windows 系 (GUI 版)

8. 6. 2. 2. 5 Help メニューを参照して下さい。

(4) Windows 系 (コマンドライン版)

8. 5. 2 Windows の場合 (2)GUI 画面を表示しない場合 を参照して下さい。

3.3 HMDE/KANJI 連携時のバージョンアップ/ダウンの注意事項

(1) FAL のみバージョンアップする時は、HMDE/KANJI 用ライブラリ (UNIX : /usr/lib/libuoc.*、Windows : インストール先ディレクトリ¥uoc.dll) を別名にコピーしておき、FAL をインストール後、元に戻して下さい。
(FAL をインストールすると、ダミーのライブラリをコピーしてしまい、この状態では HMDE/KANJI は動作しません。)

この時、HMDE/KANJI のアンインストールと再インストールは不要です。ただし、本操作を忘れた場合は、再度 HMDE/KANJI をインストールして下さい。

(2) Windows 版の場合

KANJI のバージョンが 01-05、01-06、01-06/A の場合、FAL 又は KANJI をアンインストールすると、両者のファイルが削除される場合があります。その場合は、FAL から再度インストールして下さい。

KANJI のオプションファイル等、更新して使用している場合もありますので、万が一に備え

アンインストールを行う前に、次をバックアップしておいて下さい。

① インストール先ディレクトリ毎、別名で保存しておく。

② C:¥WINNT ディレクトリ内の、libfal.ver 及び libuoc.ver ファイルを①で保存したディレクトリへコピーしておく。

更新しているファイルがあった場合、FAL 及び KANJI のインストール後、バックアップしたディレクトリからファイルをコピーして下さい。

(注) HMDE/KANJI (バージョン 01-07 以降) では発生しません。

4 FAL の機能

FALは以下の機能をC言語の関数の形式で提供します。
32bit用のFALと64bit用のFALがあります。

- (A) ボリューム名とデータセット名で指定されたデータセットをオープンする。
 - ・datasetOpen
- (B) 指定したデータセットからレコードを読み出す。
 - ・datasetGet
- (C) 指定したデータセットにレコードを書き込む。
 - ・datasetPut/datasetPut2
- (D) 指定したデータセットをクローズする。
 - ・datasetClose
- (E) エラー情報を得る。
 - ・datasetGetLastError
- (F) データセットの属性情報を得る。
 - ・datasetGetFileInformation

 - ・datasetFindFirstFile
 - ・datasetFindNextFile
 - ・datasetFindClose

 - ・datasetGetDsorgString
 - ・datasetGetRecfmString

<プログラミングの際の注意事項>

- (1) FALはスロットセーフではありません。一つのプロセス内の複数のスロットからFALを使用した場合の動作は保証されません。シグナルハンドラから使用することもできません。
- (2) 以下の語はFALの予約語です。FALを使ってプログラムを組む際に、関数名・変数名・記号定数名として使用することはできません。
 - ・ “dataset” で始まる語
 - ・ “fast_” で始まる語
 - ・ “GetVolSers”
- (3) 32bitAPからは32bit用FALを、64bitAPからは64bit用FALを使用して下さい。
- (4) 同一のデータセットに対してオープンとクローズが対になるようにして下さい。
- (5) 連続で複数のデータセットのオープンは出来ません。
- (6) “datasetOpen” でオープン中のデータセットに対し、“datasetGetFileInformation”、“datasetFindFirstFile” を使用しないで下さい。
- (7) “datasetFindFirstFile” や “datasetFindNextFile” でデータセット属性取得処理中に “datasetOpen”、“datasetGetFileInformation” または “datasetFindFirstFile” を使用しないで下さい。

4.1 データセットのオープン

[記述形式] datasetHandle = datasetOpen(pathname, mode)

	型	内 容
引数 : pathname	char *	バース名 (VolumeName:DatasetName, VSN識別子)
mode	char *	オープンモード
戻値 : datasetHandle	DATASET_HANDLE	ハンドル
-1		異常終了

pathnameで指定されたデータセット(ファイル)を、modeで指定されたオープンモードでオープンします。

pathname : ボリューム名とデータセット名を"/"で区切って指定します。さらに必要であれば、VSN識別子を", "で区切って指定します。(VSN識別子は省略可能です。)ここで、VSN識別子を指定すると、ボリューム定義ファイル上も指定が必要となります。

ボリューム名 : メインフレーム側で定義したボリュームシリアルナンバー (VSN:6桁)

ファイル名 : メインフレーム側で定義したデータセット名 (44桁以下)

VSN識別子: 同一ボリューム名がボリューム定義ファイル上に存在し、ボリューム定義ファイル内にもVSN識別子がある場合指定して下さい。(35桁以下)

mode : "r" : 読出し用にオープンする。
"w" : 書込み用にオープンする。

正常終了の場合、pathnameで指定されたデータセットに関するハンドルを返します。

datasetHandle : オープンしたデータセットを識別するための情報です。

以後のdatasetGet関数、datasetPut関数、datasetClose関数、datasetPut2関数の引数として使用します。

ハンドルの内容を書き換えてはいけません。正常な動作を保証することが出来なくなります。

異常終了の場合、-1を返します。詳細なエラー情報は datasetGetLastError関数(後述)で求めることができます。

(注) ・ボリューム名とオープンシステム側で定義したボリューム(パーティション)との対応付けは「ボリューム定義ファイル」(ファイル名=datasetmount.dat 後述)を用いて行います。

・17セクタ内で同時にオープンできるデータセットは1データセットだけです。

複数のデータセットを処理する場合は、オープンしているデータセットをdatasetClose関数によりクローズしてから、別のデータセットをオープンして下さい。

また、datasetFindFirstFile関数、datasetFindNextFile関数でデータセット属性を取得中の場合も、データセットをオープンすることはできません。

datasetFindClose関数によりデータセット属性の取得処理を終了してから、データセットをオープンして下さい。

なお、オープンシステム側のファイルのオープンについては、本制限に抵触しません。

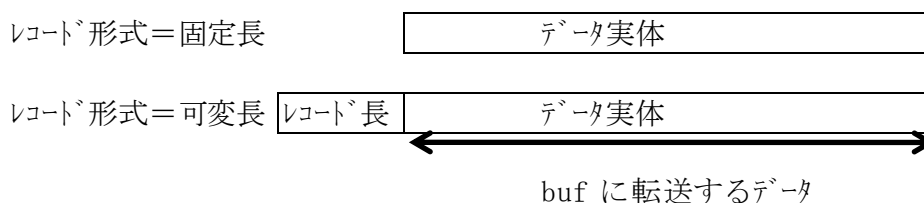
・本ライブラリで処理できるデータセットには制限があります。1.2.2節の記述を参照して下さい。

4.2 データの読出し

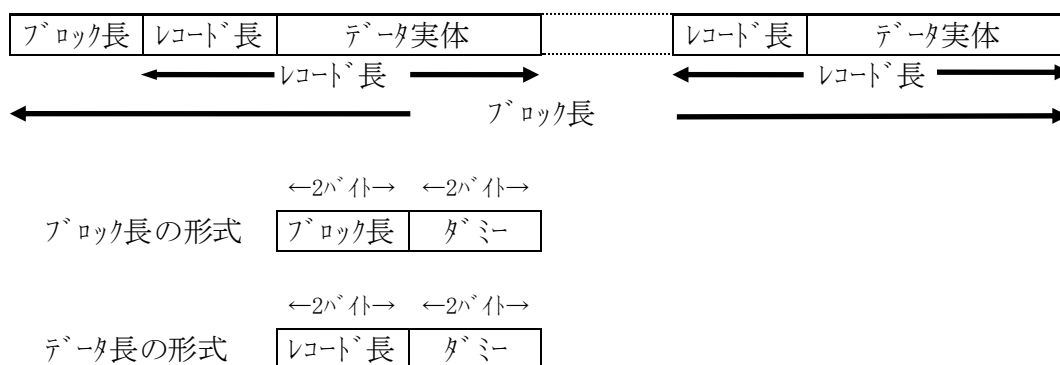
[記述形式] `reclen = datasetGet(datasetHandle, buf, buflen)`

	型	内 容
引数 : <code>datasetHandle</code>	<code>DATASET_HANDLE</code>	データセットハンドル
<code>buf</code>	<code>char *</code>	読出しバッファ
<code>buflen</code>	<code>long</code>	バッファサイズ
戻値 : <code>reclen</code>	<code>long</code>	バッファに転送したデータ実体の長さ
<code>-1</code>		異常終了

`datasetHandle` で指定されたデータセットの1レコードを `buf` に読み出します。
転送するデータは下図のようにデータ実体だけです。



(注) レコード形式が可変長の場合、データセットのブロック及びレコードは以下の形式になっていなければなりません。



`datasetHandle` : `datasetOpen`関数の戻値であるハンドルを指定します。

`buf` : 読み出したデータを格納するエリアを指定します。

`buflen` : データ格納エリアの大きさを指定します。
(実レコード長) > (`buflen`) の場合、`buf` へのデータ転送は実行せず、エラーとします。

正常終了の場合、読み出したレコードの長さを返します。

戻り値は $0 \leq \text{戻り値} \leq 32760$ となり、(実レコード長) = 0 の時でも正常終了します。

異常終了の場合及びEOF(End of File)検出の場合-1を返します。

詳細なエラー情報は `datasetGetLastError`関数(後述)で求めることができます。

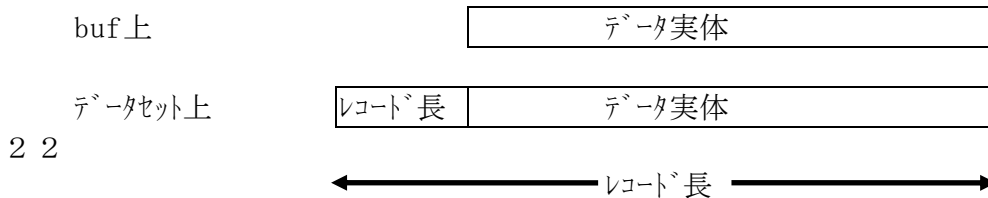
(例 : 上記のEOF検出の場合、「`DATASET_ERROR_END_OF_FILE`」が返ります)

4.3 データの書き込み

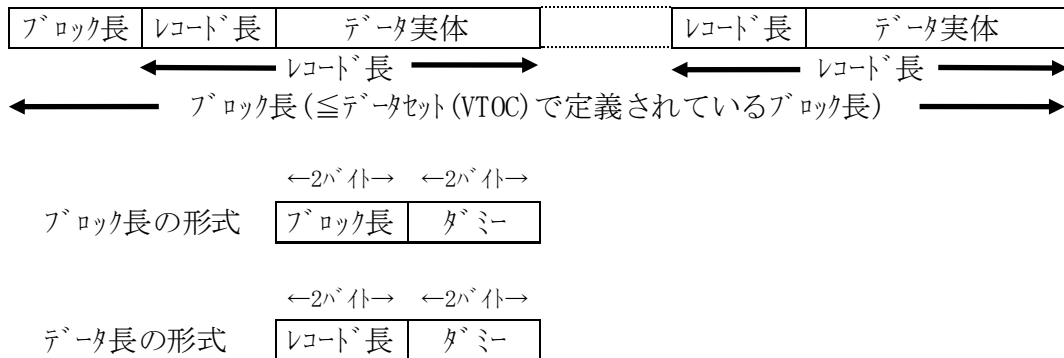
[記述形式] `reclen = datasetPut(datasetHandle, buf, buflen)`
`reclen = datasetPut2(datasetHandle, buf, buflen)`

	型	内 容
引数 : datasetHandle	DATASET_HANDLE	データセットハンドル
buf	char *	書き込みデータ
buflen	long	データサイズ
戻値 : reclen	long	書き込んだデータ実体の長さ
-1		異常終了

- datasetHandle で指定されたデータセットに1レコードを buf から書き込みます。
- レコード形式が可変長の場合、buf上のデータ実体にレコード長を付加してデータセットに書き込みます。



- データセットに書き込む際には、データセット(VTOC)で定義されているブロック長の範囲で複数のレコードをブロックングします。
- レコード形式が可変長の場合の例を以下に示します。



datasetHandle : datasetOpen関数の戻値であるハンドルを指定します。

buf : 書き込むデータを格納しているエリアを指定します。

buflen : データの大きさを指定します。以下の場合エラーとなります。

レコード形式=固定長 : (buflen) ≠ (データセットのレコード長)

レコード形式=可変長 : (buflen+4) > (データセットのレコード長)

(buflen) = 0 (データ実体なし)*1

*1 : datasetPut関数時のみエラーとなります。datasetPut2関数時は正常となります。

0byteレコード時でも正常に処理を可能とする場合はdatasetPut2関数を使用して下さい。

正常終了の場合、書き込んだデータ実体の長さを返します。

異常終了の場合 -1 を返します。

詳細なエラー情報は datasetGetLastError関数(後述)で求めることができます。

4.4 データセットのクローズ

[記述定式] datasetError = datasetClose(datasetHandle)

型	内 容
引数 : datasetHandle DATASET_HANDLE 戻値 : 0 -1	データセットハンドル 正常終了 異常終了

datasetHandleで指定されたデータセットをクローズします。

datasetHandle : datasetOpen関数の戻値であるハンドルを指定します。

正常終了の場合、0 を返します。

異常終了の場合、-1を返します。

詳細なエラー情報は datasetGetLastError関数(後述)で求めることができます。

4.5 エラー情報の取得

[記述形式] datasetLastError = datasetGetLastError()

型	内 容
引数：なし 戻値：datasetLastError long	エラーコード ^① (注)

最後に発生したエラーの詳細情報(エラーコード^①)を返します。

(注) FAL固有のエラーについては、エラーコードファイル(dataset.h)で定義しています。

(エラーコード^①の詳細については4.7節を参照して下さい)

システムエラーについては標準エラーファイル(errno.h)で定義されています。

(Windows の場合は、Microsoft VisualC++ に付属の errno.h または、winerror.h となります。)

4.6 データセット属性情報の取得

(1) 特定データセットの属性情報の取得

[記述形式] datasetError = datasetGetFileInformation(pathname, &ffd)

型	内 容
引数 : pathname char *	パス名 (VolumeName:DatasetName)
ffd DATASET_FIND_DATA	属性情報
戻値 : 0	正常終了
-1	異常終了

pathnameで指定されたデータセットの属性情報を ffd に返します。

pathname : ボリューム名とデータセット名を":"で区切って指定します。
 ボリューム名 : メインフレーム側で定義したボリュームシリアルナンバ (VSN:6桁以下)
 データセット名 : メインフレーム側で定義したデータセット名 (44桁以下)

ffd : データセットの属性情報を格納するエリアを指定します。

データセット属性情報は以下のようになっています。

```
typedef struct DATASET_FIND_DATA {
    unsigned short blockSize;           /* ブロック長          */
    unsigned short recordSize;         /* レコード長          */
    unsigned char dsorg[2];             /* データセット編成種別 */
    unsigned char recfm;                /* レコード形式        */
    char name[44];                      /* データセット名      */
    unsigned short lastBlockTt;         /* 最終ブロックアドレス(相対トラック番号) (*1) */
    unsigned char lastBlockR;          /* 最終ブロックアドレス(相対レコード番号) */
    unsigned char mftype;               /* メインフレームOS (MVS・VOS3・MSP/VSE/VOS1/XSP) */
} DATASET_FIND_DATA;
```

正常終了の場合、0 を返します。

異常終了の場合、-1 を返します。

詳細なエラー情報は datasetGetLastError関数で求めることができます。

(注) ・本関数には、datasetOpen関数と同様の制約事項があります。
 datasetOpen関数の注記を参照して下さい。

・メインフレームファイルを開いている場合、本関数は使用できません。
 メインフレームファイルを開く前か、クローズしてから使用して下さい。

(*1) 最終ブロックアドレス(相対トラック番号)が0xFFFFを超える場合、下位 2byteのみ出力します。
 (ex 0x10000 なら 0x0000を出力します。)

(2) 複数のデータセットの属性情報の取得

datasetFindFirstFile関数、datasetFindNextFile関数、datasetFindClose関数を組み合わせて使用します。

[記述形式] datasetHandle = datasetFindFirstFile(pathname, &ffd)

	型	内 容
引数 : pathname	char *	パス名 (Volumename)
ffd	DATASET_FIND_DATA	属性情報
戻値 : datasetHandle	DATASET_HANDLE	正常終了
-1		異常終了

pathnameで指定されたボリュームに含まれる最初のデータセットの属性情報をffdに返します。引き続き2番目以降のデータセットの属性情報を得たいときは、datasetFindNextFile関数を使用して下さい。

pathname : メインフレーム側で定義したボリュームシリアルナンバー (VSN:6桁以下)。必要であれば、VSN識別子を”,”で区切って指定します (省略可)。VSN識別子を指定する場合ボリューム定義ファイル上でも指定して下さい。

データセット属性情報 (DATASET_FIND_DATA) の内容はdatasetGetFileInformation関数で返すものと同じです。datasetGetFileInformation関数の説明を参照して下さい。

正常終了の場合、属性情報を参照したデータセットに関するハンドルを返します。このハンドルは、以後のdatasetFindNextFile関数、datasetFindClose関数の引数として指定しますので、ハンドルの内容を書き換えてはいけません。正常な動作を保証できなくなります。

異常終了の場合、-1 を返します。

詳細なエラー情報は datasetGetLastError関数で求めることができます。

(例 : VTOC中にデータセットが無かった場合、「DATASET_ERROR_NO_DATASET」が返ります)

(注) ・本関数には、datasetOpen関数と同様の制約事項があります。

datasetOpen関数の注記を参照して下さい。

- ・メインフレームファイルをオープンしている場合、本関数は使用できません。メインフレームファイルをオープンする前か、クローズしてから使用して下さい。また、本関数をコールしてからdatasetFindClose関数をコールするまでの間はメインフレームファイルはオープンできません。

[記述形式] datasetError = datasetFindNextFile(datasetHandle, &ffd)

型	内 容
引数 : datasetHandle DATASET_HANDLE	データセットハンドル
ffd DATASET_FIND_DATA	属性情報
戻値 : 0	正常終了
-1	異常終了及びデータセットなし

2番目以降のデータセットの属性情報を取得したい場合、本関数を使用します。
次のデータセットの属性情報の取得を不要と判断するか、あるいは本関数で「データセットなし」が返されるまで繰り返します。

datasetHandleで指定されたボリュームの次のデータセットの属性情報をffdに返します。

datasetHandle : 先行するdatasetFindFirstFile関数で返されたハンドルを指定します。

ffd : データセット属性情報を格納するエリアを指定します。

データセット属性情報(DATASET_FIND_DATA)の内容はdatasetGetFileInformation関数で返すものと同じです。datasetGetFileInformation関数の説明を参照して下さい。

正常終了の場合、0を返します。

異常終了の場合、-1を返します。

詳細なエラー情報は datasetGetLastError関数で求めることができます。

(例 : VTOC中に次のデータセットが無かった場合「DATASET_ERROR_END_OF_VTOC」が返ります)

(注) ・本関数には、datasetOpen関数と同様の制約事項があります。

datasetOpen関数の注記を参照して下さい。

- ・メインフレームファイルをオープンしている場合、本関数は使用できません。
メインフレームファイルをオープンする前か、クローズしてから使用して下さい。
また、本関数をコールしてからdatasetFindClose関数をコールするまでの間はメインフレームファイルはオープンできません。
- ・本関数を使用する前にdatasetFindFirstFile関数が先行している必要があります。

[記述形式] datasetError = datasetFindClose(datasetHandle)

型	内 容
引数 : datasetHandle DATASET_HANDLE 戻値 : 0 -1	データセットハンドル 正常終了 異常終了

datasetFindFirstFile関数、datasetFindNextFile関数で行ってきたデータセット属性情報の取得処理の終了を宣言します。

datasetHandle : 先行するdatasetFindFirstFile関数で返されたハンドルを指定します。

正常終了の場合、0 を返します。

異常終了の場合、-1 を返します。

詳細なエラー情報は datasetGetLastError関数で求めることができます。

(注) ・本関数を使用する前にdatasetFindFirstFile関数が先行している必要があります

・本関数の実行により、データセットをオープンできる状態になります。

(3) データセット編成種別・レコード形式の文字列への変換

[記述形式] datasetError = datasetGetDsorgString(dsorg, text)

型	内 容
引数 : dsorg u_char [] text char [3]	データセット編成種別を示す値(2バイト) データセット編成種別を示す文字列(3バイト) "PS" : 順編成 "VS" : VSAM編成 "DA" : 直接編成 "PO" : 区分編成 "**" : 上記以外
戻値 : 0 -1	正常終了 異常終了

dsorgで指定されたデータセット編成種別を示す値を文字列textに変換します。
dsorgにはdatasetGetFileInformation/datasetFindFirstFile/datasetFindNextFileで得られた構造体DATASET_FIND_DATAのメンバ dsorg[]の値を設定して本関数を使用します。

[記述形式] datasetError = datasetGetRecfmString(recfm, text)

型	内 容
引数 : recfm u_char text char [5]	レコード形式を示す値(1バイト) レコード形式を示す文字列(5バイト) text[0] : "F" : 固定長レコード "V" : 可変長レコード "U" : 不定長レコード text[1] : "B" : ブロック化レコード text[2] : "sp" : スパントレコード 以降 "st" : スタガードレコード
戻値 : 0 -1	正常終了 異常終了

recfmで指定されたレコード形式を示す値を文字列textに変換します。
recfmにはdatasetGetFileInformation/datasetFindFirstFile/datasetFindNextFileで得られた構造体DATASET_FIND_DATAのメンバ recfmの値を設定して本関数を使用します。

4.7 エラー情報

datasetGetLastError関数で返すエラー情報のうち、FAL固有のエラー情報(ファイル: dataset.hで定義)には以下のものがあります。

(注1) 表4.7-1のエラーコードの中で、エラーコード欄に「**」が付いているエラーコードは、エラーメッセージ欄で説明している原因の他に、当該ファイル/ボリュームへの競合(メインフレームオープンシステム間/オープンシステムオープンシステム間の同時アクセス(リソース競合))が原因となって発生することがあります。エラーメッセージ欄で記述している事項の調査とともに、競合が発生して(リソースが掛かって)いないかを調査して下さい。

なお、このとき以下のシステムエラーが発生していることもあります。

システムエラーコード = 4/5/6/9/11/14/16/22/36/39/45/46/67

また、磁気ディスク装置の異常がオープンシステムのOSからエラーとして報告されないために、「**」を付与したエラーコードがユーザプログラムに戻る場合もあります。この場合は、エラーメッセージ欄で記述している事項の調査とともに、磁気ディスク装置に何らかの異常が発生していないかを調査して下さい。

(注2) エラー情報のうち、正のエラーコードはシステムエラーです。

システムエラーについては、UNIX系OSの場合、標準エラーファイル(errno.h)で定義されています。

Windowsの場合は、Microsoft Visual C++に付属のerrno.hまたは、winerror.hとなります。

(注3) 下記エラーコードは、マルチスレッド版では発生しません。

-2, -6, -20, -23, -32

表4.7-1 FALのエラーコード・エラーコード定義値 及び意味

エラーコード	エラーコード定義値 及び意味
-2	DATASET_ERROR_ABORTED_BY_SIGNAL (abortシグナルを検出しました) ⇒ 再度FAL/FCUを起動してください。 (FAL/FCU起動中にctrl+c,ctrl+z等の操作を実施しないでください。)
-6	DATASET_ERROR_VOLUME_NOT_MOUNTED (ボリューム定義ファイルのボリュームが見つかりません) ⇒ ボリューム定義ファイルに記述しているパーティション名を確認してください。(8.2節参照)
-7	DATASET_ERROR_INVALID_VOLUME (VSNが見つかりません) ⇒ パラメタのVSNを確認してください。(8.4節参照)
**	また、ボリューム定義ファイル内のVSNを確認してください。(8.2節参照)
-8	DATASET_ERROR_DATASET_NOT_FOUND (パラメタのデータセット名が正しくありません) ⇒ データセット名を確認してください。(8.4節参照)
-9	DATASET_ERROR_NOT_SUPPORTED (不正なデータセットを検出しました) ⇒ メインフレームホストからデータセットのレコード長やブロック長を確認してください。(1.2.2節参照)
-10	DATASET_ERROR_DEVICE_TYPE_NOT_SUPPORTED (パラメタのデバイスエミュレーションタイプが正しくありません) ⇒ ボリューム定義ファイル内のデバイスエミュレーションタイプを確認してください。(1.2.2節、8.2節参照)
-11	DATASET_ERROR_DSORG_NOT_SUPPORTED (ボリュームのデータセット編成種別が正しくありません) ⇒ メインフレームホストからデータセット編成種別を確認してください。(1.2.2節参照)
-12	DATASET_ERROR_RECFCM_NOT_SUPPORTED (データセットのレコード形式が正しくありません) ⇒ メインフレームホストからデータセットのレコード形式を確認してください。(1.2.2節参照)
-13	DATASET_ERROR_INVALID_DATA (不正なデータセットを検出しました) ⇒ メインフレームホストからデータセットを確認してください。(1.2.2節参照)
**	また、ボリューム定義ファイル内のデバイスエミュレーションタイプを確認してください。(8.2節参照)
-14	DATASET_ERROR_VOLUME_DEFINITION_INVALID (ボリューム定義ファイルが正しくありません) ⇒ ボリューム定義ファイルの内容を確認してください。(8.2節参照)
**	

-15	DATASET_ERROR_DATASET_NOT_OPENED (データセットをオープンしていません) ⇒ datasetGet/datasetPutを実行する前に、datasetOpenが必要です。ユーザプログラムを確認してください。
-16	DATASET_ERROR_DATASET_NOT_CLOSED (データセットをクローズしていません) ⇒ データセットをクローズせずにオープンしています。アプリケーションプログラムを確認してください。(4節参照)
-17	DATASET_ERROR_BUFLLEN_SHORT (datasetGet関数の引数バッファサイズが正しくありません) ⇒ datasetGet関数の引数バッファサイズを確認してください。(4.2節参照)
-18 **	DATASET_ERROR_VOLUME_LABEL_INVALID (VSNが正しくありません) ⇒メインフレームホストからボリュームのインシャイスが行われているか、または、ボリューム定義ファイルの内容を確認してください。(8.2節参照)
-19 **	DATASET_ERROR_VTOC_INVALID (VTOCが正しくありません) ⇒メインフレームホストからボリュームのインシャイスが行われているか、データセットが作成されているか確認してください。
-20 **	DATASET_ERROR_VOLUME_NOT_DEFINED (指定されたボリュームが見つかりません) ⇒ボリューム定義ファイルの内容を確認してください。(8.2節参照)
-21	DATASET_ERROR_INVALID_ARGUMENT (関数に指定した引数が正しくありません) ⇒アプリケーションプログラムでFAL提供関数に指定した引数を確認してください。(4節参照)
-22	DATASET_ERROR_NO_DATASET (データセットが見つかりません) ⇒メインフレームホストからボリュームのインシャイスが行われているか、データセットが作成されているか確認してください。
-23 **	DATASET_ERROR_NON_STANDARD_RO_EXIST (非標準R0のトラックを検出しました) ⇒ボリュームの全トラックを標準R0にしてください。(1.2.2節参照)
-24	DATASET_ERROR_INVALID_MODE (datasetOpen関数の引数オープンモードが正しくありません) ⇒datasetOpen関数の引数modeは、「r」又は「w」である必要があります。引数modeの指定を確認してください(4.1節参照)
-25 **	DATASET_ERROR_VOLUME_DEFINED_READ_ONLY (指定されたボリュームはライトできません) ⇒ボリューム定義ファイル内のエミュレーションタイプを確認してください。(1.2.2節参照)
-26	FAL_INTERNAL_ERROR (FAL/FCUの内部エラーが発生しました) ⇒トラブル発生時の情報収集をしてください。(10節参照)
-27 **	DATASET_ERROR_END_OF_VOLUME (データセットのサイズまたはパーティションサイズが正しくありません) ⇒メインフレームホストからデータセットのサイズと、オープンシステムからパーティションサイズを確認してください。 また、ボリューム定義ファイルのエミュレーションタイプを確認してください。(8.2節参照) Windowsサーバの場合は、ボリュームがオフライン以外になっている事を確認してください。
-28	DATASET_ERROR_OVERFLOW (データセットが容量不足を検出しました) ⇒データセットのサイズ、形式とデータセットに書き込むファイルサイズを確認してください。(7.5節、7.6節参照)
-29	DATASET_ERROR_CANNOT_MALLOC (malloc関数が異常終了しました) ⇒オープンシステムのメモリ容量、使用状況を確認してください。また、システムエラーコードが表示されている場合、OSのシステムエラーコードが示すエラー内容を確認してください。
-30	DATASET_ERROR_FREE_INVALID_AREA (FALで異常を検出しました) ⇒トラブル発生時の情報収集をしてください。(10節参照) また、システムエラーコードが表示されている場合、OSのシステムエラーコードが示すエラー内容を確認してください。
-31	DATASET_ERROR_CANNOT_FREE (free関数が異常終了しました) ⇒トラブル発生時の情報収集をしてください。(10節参照) また、システムエラーコードが表示されている場合、OSのシステムエラーコードが示すエラー内容を確認してください。
-33	DATASET_ERROR_PARAMETER_MISMATCH (パラメータで指定されたレコード形式/レコード長/ブロック長がVTOC中の定義と合っていません) ⇒メインフレームホストから、データセットのレコード形式・レコード長・ブロック長を確認し、FALに指定したパラメータを確認してください。(7.5節、7.6節、8.4節参照)
-35	DATASET_ERROR_NO_LICENSE (ソフトウェアライセンスコードが設定されていません) ⇒ソフトウェアライセンス(ライセンスコード)を設定してください。(3.2.4節参照)

-36	DATASET_ERROR_TIME_OUT_LICENSE (ライセンスの期限が切れています) ⇒ ライセンスの期限を確認してください。
-37	DATASET_ERROR_HOSTNAME_CHANGE (ホスト名が変更されています) ⇒ ライセンスを設定したホスト名で実行しているか確認してください。(3.2.3節参照)
-39 (*3)	DATASET_ERROR_MULTI_VOLUME_DEFINITION_RECORD_OVER (マルチボリューム定義ファイルのレコード数が正しくありません) ⇒ マルチボリューム定義ファイルを確認してください。(8.3節参照)
-40 (*3)	DATASET_ERROR_MULTI_VOLUME_NO_DATASET (データセットが存在しません) ⇒ ・マルチボリューム定義ファイルの内容を確認してください。(8.3節参照) ・マルチボリューム定義ファイルの記述が正しい場合、メインフレームホストからデータが書かれているか確認してください。 ・環境変数 FAL_MULTI_CHECK の設定は本エラーに影響しません。
-41 (*3)	DATASET_ERROR_MULTI_VOLUME_NO_TRANSFER (指定されたデータセットに対するデータ転送は行えません) ⇒ ・マルチボリューム定義ファイルの内容を確認してください。(8.3節参照) ・マルチボリューム定義ファイルの記述が正しい場合、メインフレームホストからデータが書かれているか確認してください。 ・上記以外で使用する必要が有る場合は、環境変数を設定してください。(4.8節参照)
-42 (*3)	DATASET_ERROR_MULTI_VOLUME_DEFINITION_INVALID_RECORD_LENGTH (マルチボリューム定義ファイル内の1行あたりの文字数が1399バイトを超えています) ⇒ ・マルチボリューム定義ファイルの内容を確認してください。(8.3節参照)
-43 (*3)	DATASET_ERROR_MULTI_VOLUME_DEFINITION_PARAMETER_ERROR (マルチボリューム定義ファイルのパラメータ数が正しくありません) ⇒ ・マルチボリューム定義ファイルの内容を確認してください。(8.3節参照)
-44 (*3)	DATASET_ERROR_MULTI_VOLUME_DEFINITION_NO_DATASET (マルチボリューム定義ファイル内にデータセット名がありません) ⇒ マルチボリューム定義ファイルの内容を確認してください。(8.3節参照)
-45 (*3)	DATASET_ERROR_MULTI_VOLUME_DEFINITION_VSN_LENGTH_ERROR (マルチボリューム定義ファイル内のボリューム名(文字数)が正しくありません) ⇒ マルチボリューム定義ファイルの内容を確認してください。(8.3節参照)
-46 (*3)	DATASET_ERROR_MULTI_VOLUME_DEFINITION_DSN_LENGTH_ERROR (マルチボリューム定義ファイル内のデータセット名(文字数)が正しくありません) ⇒ マルチボリューム定義ファイルの内容を確認してください。(8.3節参照)
-47 (*3)	DATASET_ERROR_MULTI_VOLUME_DEFINITION_VOLID_LENGTH_ERROR (マルチボリューム定義ファイル内のVSN 識別子(文字数)が正しくありません) ⇒ マルチボリューム定義ファイルの内容を確認してください。(8.3節参照)
-48 (*3)	DATASET_ERROR_MULTI_VOLUME_DEFINITION_NO_NEXT_VOLUME (マルチボリューム定義ファイルが正しくありません) ⇒ マルチボリューム定義ファイルに、全てのボリューム名を記載してください。(8.3節参照)
-50 **	DATASET_ERROR_END_OF_FILE (データセットの終端を検出した) ⇒ メインフレームホストからデータセットにデータが書かれているか確認してください
-51 **	DATASET_ERROR_END_OF_VTOC (VTOCの終端を検出した) ⇒ メインフレームホストからボリュームのインチャライズが行われているか確認してください

*3 環境変数FAL_MULTI_DEF_FILEを定義している場合、FCU実行ディレクトリ以外のファイルを定義していないか確認して下さい。

4.8 マルチボリュームデータセットによるデータ転送

バージョン 01-XX-50/ZZ 以降のシステムでは、マルチボリュームにまたがったデータセット(マルチボリュームデータセット)からの MTO が可能 (最大 30 ボリューム(*1)) です。また、バージョン 01-XX-60/ZZ (XX ≥ 03) 以降からはマルチボリュームデータセットへの OTM 転送が可能です。(ZZ ≥ 20 でないと 2Gbyte を超えるファイルサイズを扱うことができません。)

*1 01-XX-66 以降からは最大 61 ボリュームをサポート

マルチボリュームデータセットでのデータ転送を実行する前に以下を確認して下さい。

・ MTO 転送の場合

- ① 転送対象のデータセットを構成する全てのボリュームが OPEN 側にマウント済みでボリューム定義ファイルに登録されていること。
- ② 転送対象のデータセットを割り当てた全てのボリュームがマルチボリューム定義ファイルに登録されていること。
- ③ 転送したいデータが転送対象のデータセットに書き込み済みで、VTOC の DSCB1 にあるデータセットインジケータの最終ボリュームフラグが確定していること。

・ OTM 転送の場合

- ① 転送対象のデータセットを構成する全てのボリュームが OPEN 側にマウント済みでボリューム定義ファイルに登録されていること。
- ② 転送対象のデータセットを割り当てた全てのボリュームがマルチボリューム定義ファイルに登録されていること。
- ③ 転送対象のデータセットが新規に割り当てられたものである場合は、VTOC の DSCB1 にあるデータセットインジケータの最終ボリュームフラグを確定させるため、転送実行前に M/F 側から一度データ書き込みを実行していること。

(注) OTM 転送の場合、FAL からのデータセット自動拡張は出来ません。(マルチボリュームデータセットの先頭ボリュームのみデータセットインジケータの最終ボリュームフラグが確定している状態で 2 番目以降のボリュームが必要となるサイズのデータを転送することは出来ません)

マルチボリュームデータセットを扱うには、あらかじめ FAL/FCU 実行時のカントディレクトリ内にマルチボリューム定義ファイルを作成(multidef.dat)する必要があります。本ファイルのフォーマットは § 8.3 を参照願います。

マルチボリュームを支援するにあたり、VTOC 上 DSCB1 のデータセット通し番号ボリューム順序番号データセットインジケータの最終ボリュームフラグを参照チェックするようになりました。これはシングルボリュームデータセットとして転送しようとしたボリュームがマルチボリュームデータセットの一部だった場合に不当な転送を防ぐことが目的です。その為今まで、マルチボリュームを意識することなく可能だった OTM/MTO が本チェックの追加の為できなくなる場合があります。

(具体的には下表参照)

マルチボリューム定義ファイルに未登録のデータセットへの MTO/OTM 要求時のチェック仕様

チェック 条件	DSCB1 ボリューム順序番号	1	1	1 以外	1 以外
	DSCB1 データセットインジケータ内 最終ボリュームフラグ	ON	OFF	ON	OFF
チェック 結果	シングルボリュームに対する OTM	OK(OK)	NG(OK)	NG(OK)	NG(OK)
	シングルボリュームに対する MTO	OK(OK)	NG(OK)	NG(OK)	NG(OK)

() 内は環境変数 FAL_MULTI_CHECK を OFF とした場合

バージョン 01-XX-50/ZZ 以前と同等に扱いたい場合は環境変数 FAL_MULTI_CHECK を OFF にしてください。本環境変数省略時または、ON の場合上記チェックを行います。

マルチボリューム定義ファイルに登録されているデータセットについては本環境変数の設定に関係なく、マルチボリューム定義ファイルの内容が正しいことを確認するため、以下のチェック処理が実行されます。

マルチボリュームデータセットからの MTO 転送時の VTOC 上 DSCB1 のデータセット通し番号/ボリューム順序番号/データセットインジケータの最終ボリュームフラグの VTOC 整合性チェック仕様については下記を参照して下さい。

マルチボリューム定義ファイルに登録されている MTO 要求時のチェック仕様

項番	ボリューム	VTOC 内 DSCB1			マルチボリューム定義ファイル内次ボリュームの有無	処理内容
		データセット通し番号	ボリューム順序番号	データセットインジケータ		
1	先頭ボリューム	先頭ボリュームと一致	0001	最終ボリュームフラグ=ON	あり	EOF がある場合は EOF までデータ転送を行なう。EOF がない場合は、エクステント最終までのデータ転送を行なう。
					なし	EOF がある場合は EOF までデータ転送を行なう。EOF がない場合は、エクステント最終までのデータ転送を行なう。
				最終ボリュームフラグ=OFF	あり	エクステント最終までのデータ転送を行ない次ボリュームのデータ転送へ移行する。
					なし	DATASET_ERROR_MULTI_VOLUME_DEFINITION_NO_NEXT_VOLUME エラー (FAL error:-48)
			0001 以外	—	—	DATASET_ERROR_VTOC_INVALID エラー (FAL error:-19)
			先頭ボリュームと不一致	—	—	—
2	次ボリューム	先頭ボリュームと一致	0001 以降で、マルチボリューム定義ファイルの並びと同じ	最終ボリュームフラグ=ON	あり	EOF がある場合は EOF までデータ転送を行なう。EOF がない場合は、エクステント最終までのデータ転送を行なう。
					なし	EOF がある場合は EOF までデータ転送を行なう。EOF がない場合は、エクステント最終までのデータ転送を行なう。
				最終ボリュームフラグ=OFF	あり	エクステント最終までのデータ転送を行ない次ボリュームのデータ転送に移行する。
					なし	DATASET_ERROR_MULTI_VOLUME_DEFINITION_NO_NEXT_VOLUME エラー (FAL error:-48)
			0002 以降だが不一致	—	—	DATASET_ERROR_VTOC_INVALID エラー (FAL error:-19)
			先頭ボリュームと不一致	—	—	—

マルチボリュームデータセットへの OTM 転送時の VTOC 上 DSCB1 のデータセット通し番号/ボリューム順序番号/
データセットインジケータの最終ボリュームフラグの VTOC 整合性チェック仕様については下記を参照して下さい。

マルチボリューム定義ファイルに登録されている OTM 要求時のチェック仕様

項番	ボリューム	VTOC 内 DSCB1			マルチボリューム 定義ファイル 内次ボリューム の有無	未転送 データ	処理内容		
		データセット 通し番号	ボリューム 順序番号	データセット インジケータ					
1	先頭ボ リューム	先頭ボリ ュームと一 致	0001	最終ボリ ュームフラ グ=ON	あり	なし	EOF を設定しデータ転送終了		
						あり	データセットインジケータ OFF とし EOF を設定し 次のボリュームへデータ転送		
						なし	EOF を設定しデータ転送終了		
						あり	DATASET_ERROR_OVERFLOW エラー (FAL error:-28)		
				最終ボリ ュームフラ グ=OFF	あり	なし	EOF を設定しデータセットインジケータ ON としデ ータ転送終了		
					あり	EOF を設定し次のボリュームへデータ転送			
					なし	EOF を設定しデータセットインジケータ ON としデ ータ転送終了			
					あり	DATASET_ERROR_OVERFLOW エラー (FAL error:-28)			
			0001 以 外	—	—	—	DATASET_ERROR_VTOC_INVALID エラー (FAL error:-19)		
			先頭ボリ ュームと不 一致	—	—	—	DATASET_ERROR_VTOC_INVALID エラー (FAL error:-19)		
2	次ボリ ューム	先頭ボリ ュームと一 致	0001	—	—	—	DATASET_ERROR_VTOC_INVALID エラー (FAL error:-19)		
					0002 以 降 マルチボリ ューム 定義ファイル の定義 順と 一致	最終ボリ ュームフラ グ=ON	あり	なし	EOF を設定しデータ転送終了
								あり	データセットインジケータ OFF とし EOF を設定し 次のボリュームへ
								なし	EOF を設定しデータ転送終了
				あり			DATASET_ERROR_OVERFLOW エラー (FAL error:-28)		
				最終ボリ ュームフラ グ=OFF	あり	なし	EOF を設定しデータセットインジケータ ON としデ ータ転送終了		
					あり	EOF を設定し次のボリュームへデータ転送			
					なし	EOF を設定しデータセットインジケータ ON としデ ータ転送終了			
			あり		DATASET_ERROR_OVERFLOW エラー (FAL error:-28)				
			0002 以 降 マルチボリ ューム 定義ファイル の定義 順と 不一致	—	—	—	DATASET_ERROR_VTOC_INVALID エラー (FAL error:-19)		
	先頭ボリ ュームと不 一致	—	—	—	DATASET_ERROR_VTOC_INVALID エラー (FAL error:-19)				

4.9 AIX 共用オープン機能

(1) AIXの中間ボリューム共用について

01-XX-59版以前では、AIXから中間ボリュームをオープンすると、AIXはボリュームに対してリザーブを取得する為、他のサーバから同一の中間ボリュームを共用することはできません。01-XX-60版以降(XX \geq 03)からは、環境変数を定義することにより、同一の中間ボリュームを他のサーバと共用することが可能となります。下表に対象バージョンを示します。

対象形名	バージョン	中間ボリュームの共用の可/不可
FAL for AIX (32/64bit)	01-XX-59以前	中間ボリュームを他のサーバと共用不可
	01-XX-60以降 (XX \geq 03)	環境変数を定義することにより中間ボリュームを他のサーバと共用可

(2) 環境変数の定義

環境変数FAL_NO_RESERVEを設定することで、FALから中間ボリュームをオープンする際に排他オープン(リザーブを取得する)と共用オープン(リザーブを取得しない)を切り替えることができます。下記にFAL_NO_RESERVEとオープンモードの関係を示します。

項番	FAL_NO_RESERVE	オープンモード
1	環境変数未定義	排他オープン(01-XX-59以前と同じ動作)
2	ON	共用オープン
3	OFF	排他オープン
4	その他の値	排他オープン

(3) 使用上の注意事項

AIX共用オープン機能を使用した場合は、AIXの仕様によりデータ量が2Gbyteを超える転送を行うことは出来ません。この制限は32bit版OSが対象です。

4.10 FALリザーブトライ機能

新規にリザーブトライ機能をご使用になられる場合は、FCUリザーブトライ機能をご使用されることを推奨致します。既にFALリザーブトライ機能をご使用になられている場合は、FCUリザーブトライ機能に変更することなく、FALリザーブトライ機能のご使用は可能です。

尚、FALリザーブトライ機能においては、本プログラムをインストールするサーバ環境(OS, HBAドライバ等)の影響により正しく動作を行えないことが御座いますので、FALリザーブトライ機能をご使用の場合はこの点にご注意願います。

※ FALリザーブトライ機能及びFCUリザーブトライ機能を同時に有効と設定が可能ですが、両リザーブトライ機能を有効と設定した場合には、FCUリザーブトライ機能が動作致します。

(1) FALリザーブトライ機能について

従来、FAL/FCUによるデータ転送時に対象の中間ボリュームがリザーブされているとシステムエラーが発生して異常終了してしまっていました。しかし、FALリザーブトライ機能は環境変数を定義することにより、指定秒数間隔で指定回数分トライを行い、その間にリザーブが解除されればデータ転送を再開することが可能となります。

(2) サポートバージョン

FALリザーブトライ機能をサポートするOSとOSバージョン及びFAL/FCUサポートバージョンを以下に示します。

※ 下記対象OSバージョン以外で本機能をご使用になる場合は、弊社担当営業もしくは弊社サポートサービスに契約している場合はサポートサービスにお問い合わせ下さい

FALリザーブトライ機能サポートバージョン

対象形名	対象 OS バージョン		FAL/FCU バージョン
FAL for AIX(32/64bit)	4.3.3 ML02(32/64bit)、5.1 ML02(32/64bit)、5.2 ML01(32/64bit)		01-XX-62 以降 (XX=03, 04)
FAL for Solaris(32/64bit)	8(32/64bit)、9(32/64bit)、10(32/64bit)、11(32/64bit)		01-XX-63 以降 (XX=03, 04)
FAL for Linux(32/64bit)	Redhat	AS3.0 Update3 (x86)	
	SuSE	SLES9 (x86) (AMD64/EM64T)	01-XX-64/20 以降 (XX=03, 04)
FAL for HP-UX(64bit)	11.11[2002/06] (64bit)、 11.23[2004/09 版]/[2005/04 版] (64bit)		01-XX-64/25 以降 (XX=03, 04)
FAL for Windows(32bit)	NT4.0 SP6(x86)、2000 SP4(x86)、 2003 SP1 (x86) (Itanium) (AMD64/Intel64) 2008 (x86) (Itanium) (AMD64/Intel64) 2008 R2 (x86) (Itanium) (AMD64/Intel64) 2012 (AMD64/Intel64) 2012 R2 (AMD64/Intel64) 2016 (AMD64/Intel64) 2019 (AMD64/Intel64)		01-XX-64/23 以降 (XX=03, 04)

※ HP-UX 11.0(64bit)使用時は本機能を適用しないで下さい

※ FAL/FCUバージョンがXX=03または04で既にFALリザーブトライ機能をご使用になられている場合は、XX≥05でも継続して使用可能です。

(3) 機能一覧

以下に FAL リザーブトライ機能の一覧を示します。

FAL リザーブトライ機能一覧

項番	対象	機能	内容
1	FAL	リトライ有効化設定機能	環境変数によってリトライの有効化/無効化を設定する
2		リトライ回数設定機能	環境変数によって最大リトライ回数を設定する
3		リトライ待ち時間設定機能	環境変数によってリトライ時の待ち時間を設定する
4		リトライ対象エラーコード設定機能	環境変数によってリトライの対象となるシステムエラーコードを設定する
5		リトライ機能	中間ボリュームで対象となるシステムエラーが発生した時、リトライを行う。 (open, close, read, write, seek, flush)
6		リトライ詳細ログ出力機能	どの操作で、どのシステムエラーにより、何回リトライを行ったか FAL エラーログに出力する。

(4) 環境変数の定義

(4-1) リトライ有効化設定機能(FAL)

環境変数 FAL_RETRY を設定することで、FAL から中間ボリューム (MF/OPEN) に I/O を出した時に特定のシステムエラーが発生した場合、リトライを行うかどうか指定する事ができます。以下に FAL_RETRY と FAL リザーブトライ機能の有効化/無効化の関係を示します。

FAL_RETRY と FAL リザーブトライ機能の有効化/無効化の関係

項番	FAL_RETRY	リトライ回数
1	環境変数未定義	FAL リザーブトライ機能を無効化し、現行動作と同様にリトライを行わず、即時にエラーとする
2	ON	FAL リザーブトライ機能を有効化する
3	OFF	FAL リザーブトライ機能を無効化し、現行動作と同様にリトライを行わず、即時にエラーとする
4	その他の値	FAL リザーブトライ機能を無効化し、現行動作と同様にリトライを行わず、即時にエラーとする

注) FAL_RETRY=ON と FCU_RETRY=ON を両方定義した場合は、FCU_RETRY が有効となり FAL_RETRY は無効となります。

(4-2) リトライ回数設定機能(FAL)

環境変数 FAL_RETRY_COUNT を設定することで、FAL から中間ボリューム (MF/OPEN) に I/O を出した時に特定のシステムエラーが発生した場合、リトライを行う最大回数を指定する事ができます。以下に FAL_RETRY_COUNT とリトライ回数の関係を示します。

FAL_RETRY_COUNT とリトライ回数の関係

項番	FAL_RETRY_COUNT	リトライ回数
1	環境変数未定義	10 回
3	1~600	指定回数分までリトライを行う
4	その他の値	10 回

(4-3) リトライ待ち時間設定機能(FAL)

環境変数 FAL_RETRY_WAIT_TIME を設定することで、FAL から中間ボリューム(MF/OPEN)に I/O を出した時に特定のシステムエラーが発生した場合、リトライを行うまでの待ち時間を秒単位で指定することができます。

以下に FAL_RETRY_WAIT_TIME とリトライ待ち時間の関係を示します。

FAL_RETRY_WAIT_TIME とリトライ待ち時間の関係

項番	FAL_RETRY_WAIT_TIME	リトライ待ち時間
1	環境変数未定義	1 秒間待つ
2	1~60	指定秒数間待つ
3	その他の値	1 秒間待つ

(4-4) リトライ対象エラーコード設定機能(FAL)

環境変数 FAL_RETRY_TARGET を設定することで、リトライを行う対象となるシステムエラーコードを AIX では 16 (EBUSY)、その他 OS では 5 (EIO) に加えて最大 5 つまで「,」で区切って指定することができます。尚、6 つ目以降は無効とします。また、数値以外の指定があった場合にはそれ以降の設定を全て無効とします。以下に FAL_RETRY_TARGET とリトライ対象となるシステムエラーコードの関係を示します。

※デフォルト以外のシステムエラーを指定する場合は、弊社担当営業もしくは弊社サポートサービスに契約している場合はサポートサービスにお問い合わせ下さい。

FAL_RETRY_TARGET とリトライ対象となるシステムエラーコードの関係

項番	FAL_RETRY_TARGET	リトライ対象となるシステムエラーコード	備考
1	環境変数未定義	AIX	16 (EBUSY)
		Solaris	5 (EIO)
		Linux	
		HP-UX	22 (EINVAL) 13 (EACCES)
		Windows	170 (ERROR_BUSY)
2	システムエラー, システムエラー, システムエラー...	「,」で区切られた最大 5 つ (HP-UX は最大 4 つ) のシステムエラーコード	各 OS における初期値として 16 (EBUSY), 5 (EIO), 22 (EINVAL), 13 (EACCES) 及び 170 (ERROR_BUSY) は既に含まれている
3	その他の値	環境変数未定義時と同様	—

(4-5) 注意事項

ボリューム競合時に OS 側で出力されるエラーコードは、OS や HBA ドライバ等のホスト環境やボリューム競合タイミング等の運用により変わります。構築において十分なテストを実施の上、デフォルト値以外でリトライ対象とするエラーコードを選定してください。

(5) リトライ詳細の出力機能(FAL)

システムエラー 16 (EBUSY) が発生したためにリトライ処理を行ったことを FAL エラーに出力します。出力する内容は、日付、ユーザー名、システムエラーコード、I/O 種類、リトライ回数、ソースの行番号、バージョン、プロセス ID、対象データセット名です。以下に FAL エラーに出力するリトライ詳細の書式を示します。

FAL エラーに出力するリトライ詳細の書式

Mon Nov 8 16:21:23 2004	: root	: err=16 open	Retry(1)	at 12345	: 01-03-58/21	PID=1234	VSN:DSN	
日付	ユーザー名	システム エラー コード	I/O 種類※1 回数	リトライ 回数	行番号	バージョン	プロセス ID	対象 データセット

※1 I/O 種類に出力されるものは、” open” , “close” , “read” , “write” , “seek” , “flush” のいずれかである

(6) 異常処理

以下に本機能で発生するエラーを示します。

エラー		エラーメッセージ	内容	備考
コード	名称			
16	EBUSY	Resource busy	Reserve が取得されているデバイスを開こうとした。	AIX のみ
5	EIO	I/O error	デバイスが使用不可	—
22	EINVAL	Invalid argument	引数不正	HP のみ
13	EACCES	Permission denied	アクセス権限がない	
170	ERROR_BUSY	The requested resource is in use.	要求されたリソースは使用中です。	Windows のみ

5 FAL の使用方法

5.1 環境設定

(1) 共用ボリュームの定義

FALでアクセスするデータセットを格納するファイルは「共用ボリューム」として定義しておかなければなりません。また、FALでアクセスする中間ファイルは、フォーマットでフォーマットした中間ボリュームに、アロケータで作成しておかなければなりません。フォーマットについては2.3節を、アロケータについては2.4節を参照して下さい。共用ボリュームのタイプ(デバイスエミュレーションタイプ)については1.2.2節を参照して下さい。

(2) ボリューム定義ファイルの作成

FALが、メインフレーム側で定義した論理ボリュームまたはフォーマットでフォーマットした中間ボリュームとホップシステム側で定義したパーティションを対応付けるために使用する「ボリューム定義ファイル」を、FALを使用するアプリケーションプログラムの実行時にカレントディレクトリとなるディレクトリに作成しておかなければなりません。ボリューム定義ファイルの詳細については8.2節で説明します。

(3) コンパイル方法

FALをユーザが作成したプログラムに組込む場合の例を以下に示します。UNIX系OSの場合、コンパイラはANSI準拠のコンパイラができるのものを使用して下さい(C++には対応していません)。また、ヘッダファイルは、FALを使用するプログラム中でインクルードして下さい。

(i) 32bit用FAL

① AIXの場合

```
# cc -qlanglvl=ansi -D_NO_MT -o 出力ファイル名 ソースファイル名 /usr/lib/libfal.a  
      ・libfal.a : FALのオブジェクトモジュールファイルの名称です。
```

② Solarisの場合

```
# cc -D_NO_MT -o 出力ファイル名 ソースファイル名 /usr/lib/libfal.so.1  
      ・libfal.so.1 : FALのオブジェクトモジュールファイルの名称です。
```

③ Linuxの場合

```
# gcc -D_NO_MT -o 出力ファイル名 ソースファイル名 /usr/lib/libfal.so.1  
  ・libfal.so.1 : FALのオブジェクトモジュールファイルの名称です。
```

④ Windowsの場合

(ア) Developer Studioを起動します。

(イ) プロジェクトを新規作成します。

(ウ) プロジェクトのフォルダに以下の三つのFALファイルをコピーします。

dataset.h

fal.dll

fal.lib

(エ) Developer Studioの「プロジェクト」メニューの「設定」を選択します。

(オ) プロジェクト設定ダイアログ内の「リンク」タブを選択します。

(カ) 「オブジェクトライブラリモジュール」欄に fal.lib を追加します。

(キ) プロジェクト設定ダイアログ内の「C/C++」タブを選択します。

(ク) 「プリプロセッサの定義」欄に _NO_MT を追加します。

(ケ) ビルド/実行します。

(ii) 64bit用FAL

① AIXの場合

```
# cc -qlanglvl=ansi -q64 -D_NO_MT -o 出力ファイル名 ソースファイル名 /usr/lib/libfal64.a  
・libfal64.a : FALの64bit用オブジェクトモジュールファイルの名称です。
```

② HP-UX (Itanium) の場合

```
# cc +DD64 -D_NO_MT -o 出力ファイル名 ソースファイル名 /usr/lib/hpux64/libfal64.so  
・libfal64.so : FALの64bit用オブジェクトモジュールファイルの名称です。
```

③ Solarisの場合

```
# cc -xarch=v9 -D_NO_MT -o 出力ファイル名 ソースファイル名 /usr/lib/sparcv9/libfal64.so.1  
・libfal64.so.1 : FALの64bit用オブジェクトモジュールファイルの名称です。
```

④ Linuxの場合

```
# gcc -D_NO_MT -o 出力ファイル名 ソースファイル名 /usr/lib/libfal64.so.1  
・libfal64.so.1 : FALのオブジェクトモジュールファイルの名称です。
```

⑤ Windowsの場合

(ア) Visual Studioを起動します。

(イ) プロジェクトを新規作成します。

(ウ) プロジェクトのフォルダに以下の三つのFALファイルをコピーします。

dataset.h

fal64.dll

fal64.lib

(エ) Visual Studioの「プロジェクト」メニューの「設定」を選択します。

(オ) プロジェクト設定ダイアログ内の「リンク」タブを選択します。

(カ) 「オブジェクト/ライブラリモジュール」欄に fal64.lib を追加します。

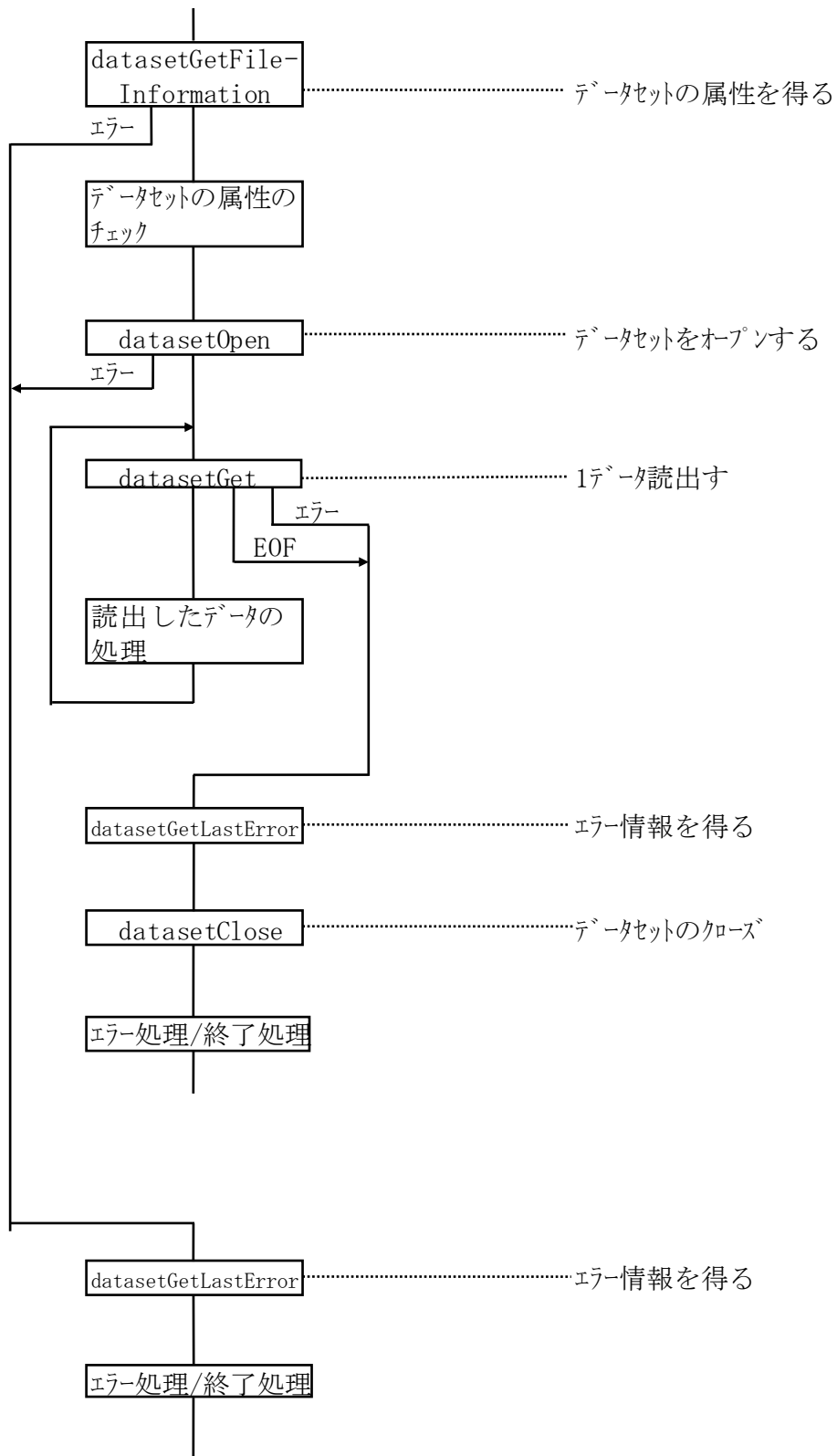
(キ) プロジェクト設定ダイアログ内の「C/C++」タブを選択します。

(ク) 「プリプロセッサの定義」欄に _NO_MT を追加します。

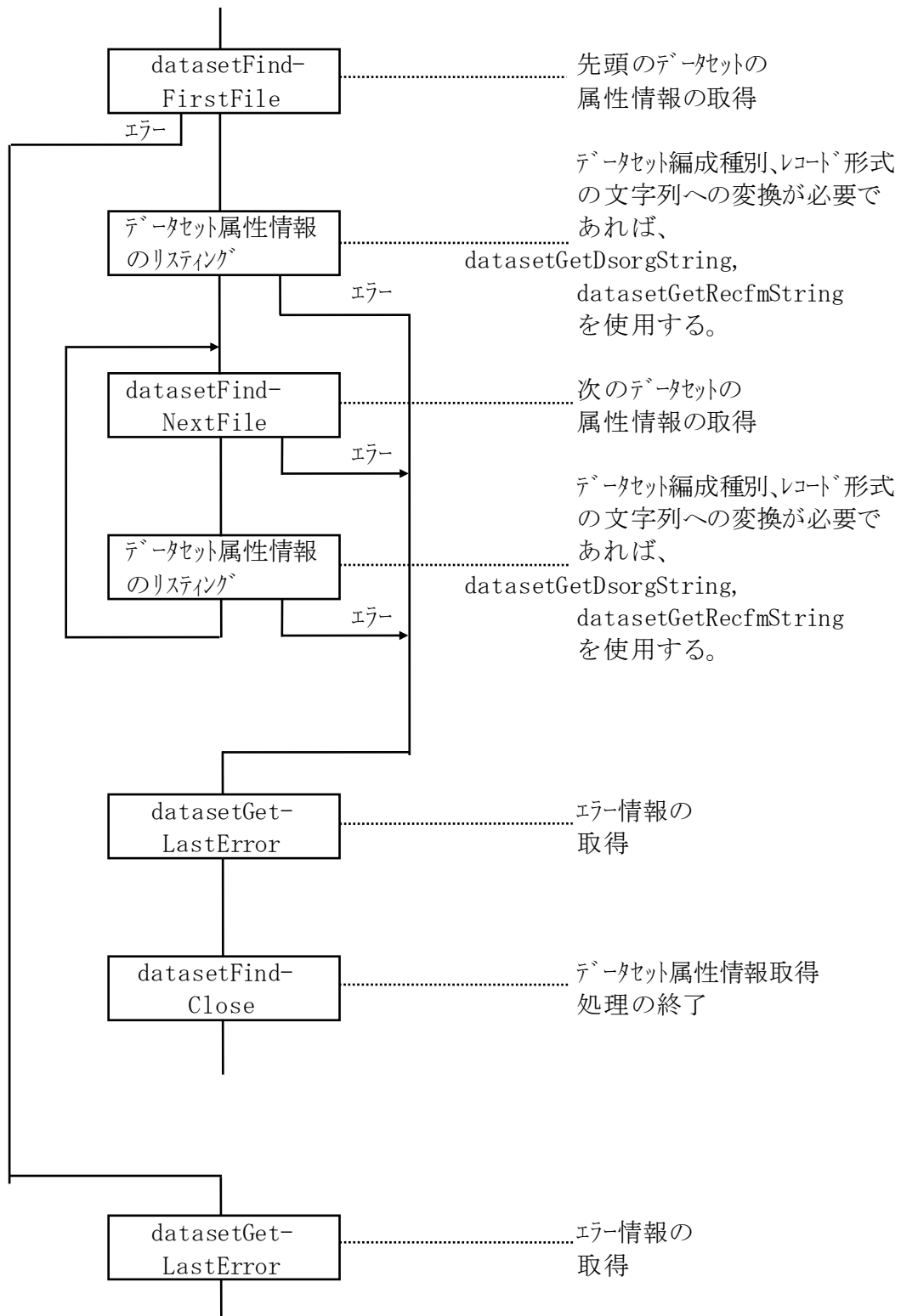
(ケ) ビルド/実行します。

5.2 関数の使用例

(1) 使用例 : データの読出し



(2) 使用例：データセット属性の取得



6 FAL マルチスロットの機能と使用方法

6.1 仕様

FAL は以下の機能を C 言語の関数形式で提供します。(スロットセーフです。)

- (A) 情報格納エリアの確保を行います。
 - ・dataset_AllocGlobal
- (B) ボリューム名とデータセット名で指定されたデータセットをオープンする。
 - ・dataset_Open
- (C) 指定したデータセットからレコードを読み出す。
 - ・dataset_Get/dataset_Get2
- (d) 指定したデータセットにレコードを書き込む。
 - ・dataset_Put/dataset_Put2
- (E) 指定したデータセットをクローズする。
 - ・dataset_Close
- (F) 情報格納エリアの開放を行います。
 - ・dataset_FreeGlobale
- (G) ファイルポインタを先頭に戻します。
 - ・dataset_Rewind
- (H) データセットの属性情報を得る。
 - ・dataset_GetFileInformation
 - ・dataset_FindFirstFile
 - ・dataset_FindNextFile
 - ・dataset_FindClose

<プログラミングの際の注意事項>

- (1) シグナルハンドラから使用することはできません。
- (2) 以下の語はFALの予約語です。FALを使ってプログラムを組む際に、関数名・変数名・記号定数名として使用することはできません。
 - ・ “dataset” で始まる語
 - ・ “fast_” で始まる語
 - ・ “GetVolSers”
- (3) マルチスロット対応 API と非マルチスロット対応 API は混在して使用しないでください。
- (4) 32bit 版の AIX4.3/5L/6.1 と WindowsNT/Windows2000Server/Server2003/Server2008 に対応しています。
- (5) マルチスロット対応 API 使用時、ボリューム定義ファイルは不要です。
- (6) マルチスロット対応 API では複数のデータセットを同時にオープン可能となります。
 - dataset_AllocGlobalデータセット'A'に対する情報格納エリアの確保
 - dataset_AllocGlobalデータセット'B'に対する情報格納エリアの確保
 - dataset_Open.....データセット'A'に対するオープン
 - dataset_Open.....データセット'B'に対するオープン

6.1.1.情報格納エリアの確保

[記述形式] memError= dataset_AllocGlobal(dgpp,derrno,malloc,free)

	型	内容
引数 : dgpp	void **	Global メモリエリア
derrno	long*	エラー情報返却エリア
malloc	void**	malloc()
free	void*	free()
戻値 : memError	int	
0		異常終了

本関数は情報格納エリアの取得を行い、Global メモリエリアに返します。

dgpp Global メモリエリア格納エリア
derrno FAL のエラーコードを格納するアドレスを指定します。
malloc malloc 関数のアドレスを指定します。malloc と指定してください。
free free 関数のアドレスを指定します。free と指定してください。

関数が正常に終了した場合は、1 を返します。
異常終了時は、derrno の内容を参照し詳細なエラーコードを取得して下さい。
エラーの詳細は 4.7 エラー情報を参照してください。

使用例

```
void *memptr; /* global memory area */
long err;     /* global err 情報 */
int retcode;
:
retcode = dataset_AllocGlobal(&memptr, &err, malloc, free);
:
retcode = dataset_FreeGlobal(&memptr, &err);
```

備考

- 本関数を発行したら、処理の最後で、dataset_FreeGlobal()を必ず発行してください。
- dataset_Open()、dataset_FindFirstFile()の前に必ず、本関数を発行してください。

6.1.2.データセットのオープン

[記述形式] datasetError=dataset_Open(global,g_error,devname,dsname,voltype,mode)

型	内容
引数 : global void*	グローバルメモリエリア
g_error long*	エラー情報返却エリア
devname char*	raw デバイス名
dsname char*	データセット名
voltypr char*	ボリュームエミュレーションタイプ
mode char*	オープンモード
戻値 : datasetError long	
-1	異常終了

dsname で指定されたデータセット(ファイル)を、mode で指定されたオープンモードでオープンします。

global Global メモリエリア(dataset_AllocGlobal で取得した Global メモリエリアを指定します。)

g_error FAL のエラーコードを格納するアドレスを指定します。

devname raw デバイス名(special file)

dsname データセット名

voltype Device エミュレーションタイプ (3390-3A,3390-3B)

mode “r” : 読出し用にオープンします。
 “w” : 書込み用にオープンします。

戻り値

関数が異常終了の場合は、-1 を返します。その他の時は正常終了です。関数が異常終了した場合は、g_error を参照して詳細なエラー情報を取得してください。エラーの詳細は 4.7 エラー情報を参照願います。

使用例

```
void *memptr;                    /* global memory area */
long err,datasetError;         /* global err 情報 */
int retcode;
retcode = dataset_AllocGlobal(&memptr, &err, malloc, free);
~
datasetError = dataset_Open(memptr, &err, "¥¥.¥¥PHYSICALDRIVE1", "DSN001", "3390-3A", "r");
~
datasetError=dataset_Close(memptr, &err);
retcode = dataset_FreeGlobal(&memptr, &err);
```

備考

- 本関数を発行したら処理終了時に必ず dataset_Close() を発行してください。
- dataset_Get()/dataset_Get2()/dataset_Put()/dataset_Put2()/dataset_Rewind()/dataset_GetFileInformation()関数の発行前に必ず本関数を発行してください。

6.1.3.データの読出し

[記述形式] `reclen = dataset_Get(global, g_error, buf, buflen)`

`reclen = dataset_Get2(global, g_error, buf, buflen)`

	型	内容
引数 : <code>global</code>	<code>void*</code>	グローバルメモリア
<code>g_error</code>	<code>long*</code>	エラー情報返却エリア
<code>buf</code>	<code>char*</code>	読み出しバッファ
<code>buflen</code>	<code>long</code>	バッファに転送したデータ実体の長さ
戻値 : <code>reclen</code>	<code>long</code>	バッファに転送した実体の長さ
<code>-1</code>		異常終了

先にオープンしたデータセットの1レコードを `buf` に読み出します。転送するデータは、データ実体だけです。詳細は 4.2 データの読出しを参照願います。

`global` Global メモリア(`dataset_AllocGlobal` で取得した Global メモリアを指定します。)

`g_error` FAL のエラーコードを格納するアドレスを指定します。

`buf` 読込んだデータを格納するバッファを指定します。

`buflen` バッファの大きさを指定します。

戻り値

`dataset_Get()`の時

関数が正常に終了した場合は、`reclen` (レコード長) を返します。(1 ≤ レコード長 ≤ 32760)

異常の場合は -1 を返します。

EOF 検出時 0 を返します。

`Dataset_Get2()`の時

関数が正常に終了した場合は、`reclen` (レコード長) を返します。(0 ≤ レコード長 ≤ 32760)

異常の場合は -1 を返します。

EOF 検出時は `DATASET_ERROR_END_OF_FILE` を返します。

関数の戻値 = -1 の場合、`g_error` の内容を参照し、詳細なエラーコードを取得して下さい。

エラーの詳細は 4.7 エラー情報を参照願います。

6.1.4.データの書込み

[記述形式] `recren= dataset_Put(global, g_error, buf, buflen)`

`recren= dataset_Put2(global, g_error, buf, buflen)`

	型	内容
引数 : <code>global</code>	<code>void*</code>	グローバルメモリア
<code>g_error</code>	<code>long*</code>	エラー情報返却エリア
<code>buf</code>	<code>char*</code>	書き込みバッファ
<code>buflen</code>	<code>long</code>	バッファに転送したデータ実体の長さ
戻値 : <code>reclen</code>	<code>long</code>	書き込んだデータ実体の長さ
<code>-1</code>		異常終了

オープンしたデータセットに 1レコード書き込みます。

レコード形式が可変長の場合、`buf` 上のデータ実体にレコード長を付加してデータセットに書き込みます。

詳細は 4.3 データの書込みを参照願います。

`global` Global メモリア(`dataset_AllocGlobal` で取得した Global メモリアを指定します。)

`g_error` FAL のエラーコードを格納するアドレスを指定します。

`buf` 書き込みバッファ

`buflen` バッファサイズを指定します。

戻り値

関数が正常に終了した場合は、`reclen` (レコード長) を返します。

異常の場合は `-1` を返します。関数の戻値 `-1` の場合、`g_error` の内容を参照し、詳細なエラーコードを取得して下さい。エラーの詳細は 4.7 エラー情報を参照願います。

備考

- `buflen=0` の時、`dataset_Put` 関数はエラーとなりますが、`dataset_Put2` 関数は正常終了します(実態なしデータも扱えます)。
- 満杯エラー時、`dataset_Put2` の戻り値 `-1` ですが `dataset_Put` の戻り値 `0` となります。`g_error` はいづれも `-50` です。

6.1.5.dataset_Close

[記述形式] datasetError=dataset_Close(global,g_error)

型	内容
引数 : global void*	グローバルメモリア
g_error long*	エラー情報返却エリア
戻値 : datasetError long	
-1	異常終了

データセットをクローズします。

global Global メモリア(dataset_AllocGlobal で取得した Global メモリアを指定します。)

g_error FAL のエラーコードを格納するアドレスを指定します。

戻り値

関数が正常に終了した場合は、0 を返します。

異常の場合は -1 を返します。関数の戻値 = -1 の場合、**g_error** の内容を参照し、詳細なエラーコードを取得して下さい。エラーの詳細は 4.7 エラー情報の項を参照願います。

6.1.6.情報格納エリアの開放

[表示形式] memError= dataset_FreeGlobal(dgpp, derrno)

	型	内容
引数 : dgpp	void**	グローバルメモリエリアのエリア
derrno	long*	エラー情報返却エリア
戻値 : memError	int	
0		異常終了

本関数は情報格納エリアを開放します。

dgpp Global メモリエリアを指定する。(dataset_AllocGlobal で取得した Global メモリエリアのエリアを指定します。)

derrno FAL のエラーコードを格納するアドレスを指定します。

戻り値

関数が正常に終了した場合は、1 を返します。

異常の場合は0を返します。異常終了時は、g_error を参照し、詳細なエラーコードを取得して下さい。エラーの詳細は 4.7 エラー情報を参照願います。

備考

- ・特になし

6.1.7.処理対象コードポインタの初期化

[表示形式] datasetError=dataset_Rewind(global,g_error)

型	内容
引数 : global void**	グローバルメモリエリアのエリア
g_error long*	エラー情報返却エリア
戻値 : datasetError long	
-1	異常終了

dataset_Put/dataset_Put2/dataset_Get/dataset_Get2 の前に本関数を発行すると、先頭コードポインタが戻り、次回 dataset_Put/dataset_Put2/dataset_Get/dataset_Get2 は、先頭コードから処理を行います。

global Global メモリエリア(dataset_AllocGlobal で取得した Global メモリエリアを指定します。)

g_error FAL のエラーコードを格納するアドレスを指定します。

戻り値

関数が正常に終了した場合は、0 を返します。

異常の場合は -1 を返します。異常終了時は、**g_error** を参照し、詳細なエラーコードを取得して下さい。エラーの詳細は 4.7 エラー情報を参照願います。

6.1.8.データセット属性情報の取得

(1) 特定データセットの属性情報の取得

[記述形式] datasetError= dataset_GetFileInformation(global, g_error, &ffd)

型	内容
引数 : global void*	グローバルメモリエリアのエリア
g_error long*	エラー情報返却エリア
ffd DATASET_FIND_DATA	データセット属性情報格納エリア
戻値 : datasetError long	
-1	異常終了

既にオープン済みのデータセットの属性を ffd に返します。

global Global メモリエリアを指定します。(dataset_AllocGlobal で取得した Global メモリエリアを指定します。)

g_error FAL のエラーコードを格納するアドレスを指定します。

ffd 最初のデータセット属性情報格納エリア

データセット属性情報は以下のようにになっています。

```
typedef struct DATASET_FIND_DATA {
    unsigned short blockSize;                    /* ブロック長            */
    unsigned short recordSize;                 /* レコード長            */
    unsigned char dsorg[2];                     /* データセット編成種別 */
    unsigned char recfm;                        /* レコード形式         */
    char name[44];                              /* データセット名        */
    unsigned short lastBlockTt; /* 最終ブロックアドレス(相対トラック番号) */
    unsigned char lastBlockR; /* 最終ブロックアドレス(相対レコード番号) */
    unsigned char mftype; /* メインフレームOS (MVS・VOS3・MSP/VSE/VOS1/XSP) */
} DATASET_FIND_DATA;
```

戻り値

関数が正常に終了した場合は、0 を返します。

異常の場合は -1 を返します。異常終了時は、g_error を参照し、詳細なエラーコードを取得して下さい。エラーの詳細は 4.7 エラー情報を参照願います。

備考

- 本関数を呼ぶ前に必ず、dataset_Open()を発行してください。

(2) 複数データセットの属性情報の取得

dataset_FindFirstFile 関数、dataset_FindNextFile 関数、dataset_FindClose 関数を組み合わせて使用します。

[記述形式]datasetHandle=dataset_FindFirstFile(global, g_error, pathname, voltype, &ffd)

型	内容
引数 : global void*	グローバルメモリアreaのエリア
g_error long*	エラー情報返却エリア
pathname char*	raw デバイス名
voltype char*	Device エミュレーションタイプ
ffd DATASET_FIND_DATA	データセット属性情報格納エリア
戻値:datasetHandle DATSET_HANDLE	
-1	異常終了

raw デバイス名で指定されたボリュームの先頭データセットの属性情報を ffd に返します。

global Global メモリアreaを指定します。(dataset_AllocGlobal で取得した Global メモリアreaを指定します。)

g_error FAL のエラーコードを格納するアドレスを指定します。

pathname パーティション名/物理ドライブ名のアドレスを指定します。

voltype Device エミュレーションタイプ (3390-3A、3390-3B)

ffd 最初のデータセット属性情報格納エリア

データセット属性情報はdataset_GetFileInformationを参照して下さい。

戻り値

関数が正常に終了した場合は、DATASET_HANDLE を返します。このハンドルは以降の dataset_FindNextFile/dataset_FindClose 関数の引数として使用します。

異常の場合は-1 を返します。異常終了した際は g_error を参照し、詳細なエラーコードを取得して下さい。エラーの詳細は4.7 エラー情報を参照願います。

VTOC 内にデータがなかった場合は、g_error=DATASET_ERROR_NO_DATASET が返ります。

[記述形式] datasetError= dataset_FindNextFile(global,g_error,datasetHandle, &ffd)

型	内容
引数 : global void*	グローバルメモリアのエリア
g_error long*	エラー情報返却エリア
datasetHandle DATASET_HANDLE	データセットハンドル
ffd DATASET_FIND_DATA	データセット属性情報格納エリア
戻値 : datasetError long	
-1	異常終了

2 番目以降のデータセットの取得は本関数を用いて行います。

次のデータセットの属性情報取得を不要と判断するか、あるいは本関数でデータセットなしが返されるまで、繰り返します。

global Global メモリアを指定します。
(dataset_AllocGlobal で取得した Global メモリアを指定します。)

g_error FAL のエラーコードを格納するアドレスを指定します。

datasetHandle データセットハンドルを指定します。

ffd 次のデータセット属性情報格納エリア
データセット属性情報はdataset_GetFileInformationを参照して下さい。

戻り値

関数が正常に終了した場合は、0 を返します。

異常の場合は-1 を返します。異常終了した際は **g_error** を参照し、詳細なエラーコードを取得して下さい。エラーの詳細は 4.7 エラー情報を参照願います。

VTOC の中に次のデータセットがなかった場合は

戻り値-1 でエラー情報として DATASET_ERROR_END_OF_VTOC を返します。

備考

- 本関数を発行する前に必ず dataset_FindFirstFile を発行して下さい。データセットの取得が終了したら、最後に dataset_FindClose を発行して下さい。

[表示形式]datasetError= dataset_FindClose(global,g_error,datasetHandle)

型	内容
引数 : global void*	グローバルメモリアreaのエリア
g_error long*	エラー情報返却エリア
datasetHandle DATASET_HANDLE	データセットハンドル
戻値 : datasetError long	
-1	異常終了

dataset_FindClose 関数は、dataset_FindFirstFile、dataset_FindNextFile で行ってきたデータセット属性情報の取得処理の終了を宣言します。

global Global メモリアreaを指定します。(dataset_AllocGlobal で取得した Global メモリアreaを指定します。)

g_error FAL のエラーコードを格納するアドレスを指定します。

datasetHandle データセットハンドルを指定します。

戻り値

関数が正常に終了した場合は、0 を返します。

異常の場合は-1 を返します。異常終了した際は g_error を参照し、詳細なエラーコードを取得して下さい。エラーの詳細は 4.7 エラー情報を参照願います。

備考

特になし。

6.2 コンパイル方法

FAL をユーザが作成したプログラムに組込む場合の例を以下に示します。
UNIX 系 OS の場合、C コンパイラは ANSI 準拠のコンパイラができるものを使用して下さい。(C++には対応していません。)

また、ヘッダファイルは、FAL を使用するプログラム中でインクルードして下さい。

AIX の場合

```
# cc -qlanglvl=ansi -o 出力ファイル名 ソースファイル名 /usr/lib/libfalmt.a  
  ・libfalmt.a : FALのオブジェクトモジュールファイルの名称です。
```

WindowsNT/Windows2000/Windows2003の場合

- (ア) Developer Studioを起動します。
- (イ) プロジェクトを新規作成します。
- (ウ) プロジェクトのフォルダに以下の三つのFALファイルをコピーします。

```
dataset.h  
falmt.dll  
falmt.lib
```

- (エ) Developer Studioの「プロジェクト」メニューの「設定」を選択します。
- (オ) プロジェクト設定ダイアログ内の「リンク」タブを選択します。
- (カ) 「オブジェクト/ライブラリモジュール」欄に falmt.lib を追加します。
- (キ) ビルド/実行します。

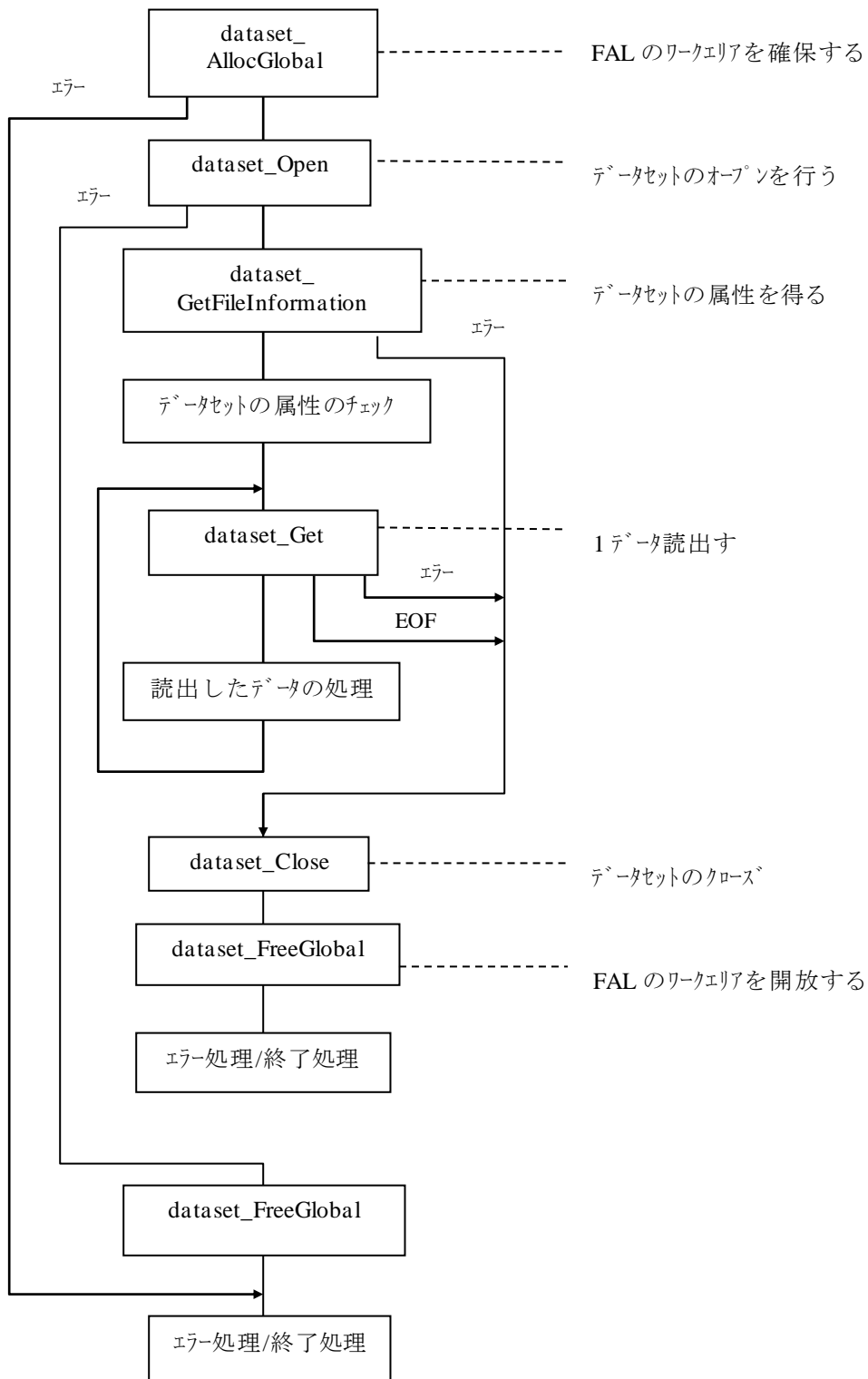
6.3 エラー情報

基本的には、マルチスレッド未対応版と同じです。詳細は4.7エラー情報を参照願います。
下記エラーコードは、マルチスレッド版では発生しません。

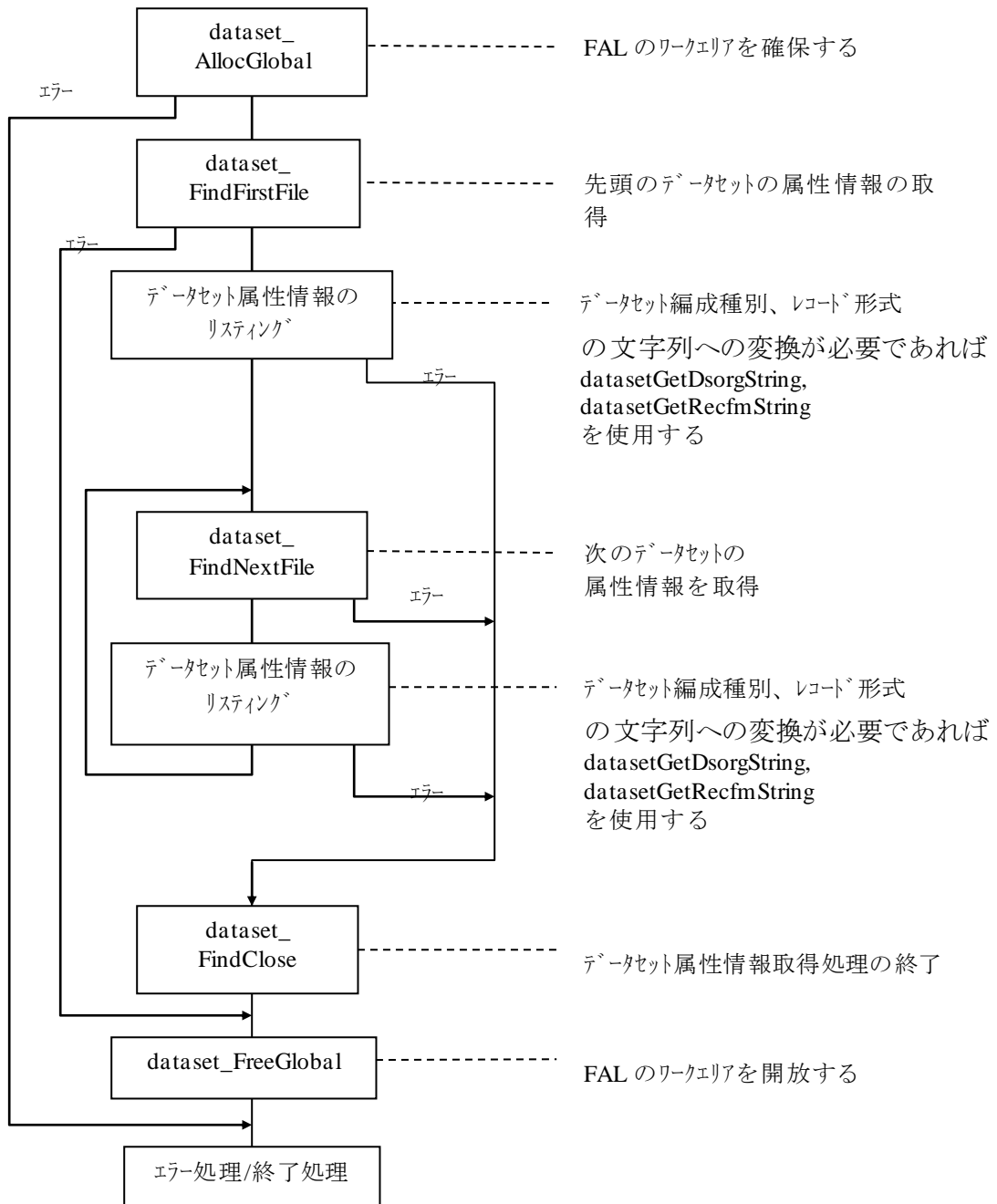
-2, -6, -20, -23, -32

6.4 関数の使用例

(1) 使用例：データの読出し



(2) 使用例：データセット属性の取得



7 FCU の機能

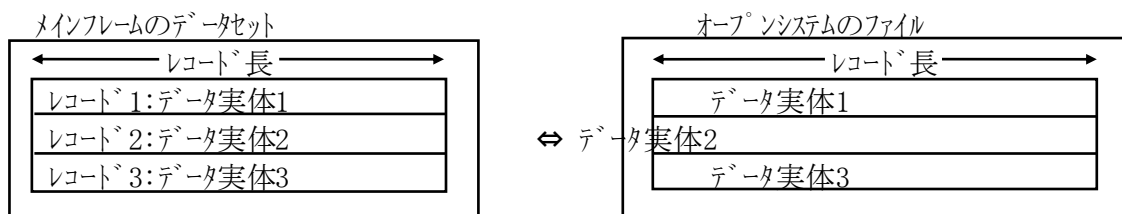
7.1 ファイル変換機能

指定されたメインフレームのデータセットとオープンシステムのファイルとの間でデータ実体だけの転送を行います。
 ・処理の対象とするデータは、ファイルの先頭からファイルの終端までの全レコードのデータ実体です。

(注) ファイルの終端 = ・メインフレームのデータセット : EOFレコードまたは最終エクステントの終端
 ・オープンシステムのファイル : EOF

(1) データセットのレコード形式が固定長の場合のデータ編集形式

メインフレームのデータセットのレコード形式が固定長の場合、データ実体をそのままメインフレームのデータセットとオープンシステムのファイルとの間で転送します。

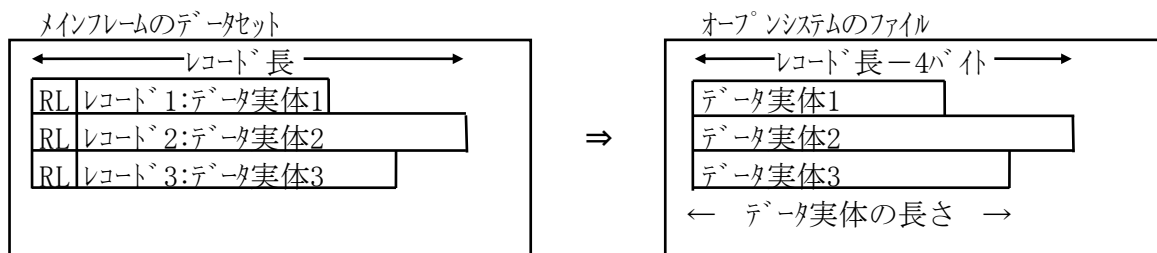


(注) レコード長 : データセットの属性として定義されているレコード長で、レコード実体の長さに一致しています。

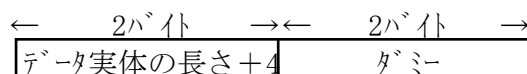
図7.1-1 レコード形式=固定長の場合のデータ編集形式

(2) メインフレームのデータセットのレコード形式が可変長の場合のデータ編集形式

メインフレームのデータセットのレコード形式が可変長の場合、データ実体だけ転送します。



(注) ・レコード長 : データセットの属性として定義されているレコード長で、レコードの最大長を表しています。
 ・RL : 各レコードの「データ実体の長さ+4バイト」の値を以下の形式で示しています。
 メインフレームからオープンシステムへの転送では、オープンシステムのファイルには転送されません。
 ・オープンシステムからメインフレームへの転送で、後述のデリミタが指定されていない場合は、最大レコード長を全レコードのレコード長として変換処理を行います。このため、デリミタなしで、可変長レコードの処理を行うとデータが破壊された状態になります。



・ブロックの長さを示している情報も転送されません。

図7.1-2 レコード形式=可変長の場合のデータ編集形式

7.2 パイプ出力機能

パイプ出力機能が指定されている場合、FCUからのデータ出力先をパイプ (Named pipe) とし、パイプを経由してUNIX系OS上のアプリケーションプログラム/ユーティリティプログラムへデータを転送します。

本機能を使用することにより、ファイル出力の場合よりも高速にメインフレームのデータセットをオープンシステム上に転送することができます。

(注)・本機能は、UNIX系OSのみサポートしています。Windows版はサポートしていません。

(Windows版では、FAL SQL Loader連携時のみパイプ指定可能。

ただし、FAL SQL LoaderはUSP VとVSP, VSP G1000, VSP 5000では未サポートです。)

・サポートするデータ転送方向は、メインフレームからオープンシステムへの転送のみです。

・本機能を使用する場合は、パイプからデータを受取るアプリケーションプログラム/ユーティリティプログラムを用意する必要があります。

タイムアウト値の設定

パイプによるデータ転送において、FCUはパイプへの出力可能状態(アプリケーションプログラム/ユーティリティプログラム側がデータ読み込み可能状態)になるまで待ちます。一定時間待っても書き込み可能状態にならない場合は、タイムアウトエラーとなります。

タイムアウト値は、環境変数に「WAIT_TIME_VALUE」として定義することで設定できます。

定義できる範囲は、0~1440 (分) です。ここで”0”は無制限です。

環境変数が定義されていない場合(デフォルト)のタイムアウト値は、10分です。

環境変数は以下の例を参考に設定して下さい。

- ・シェルの場合：ホームディレクトリの「.cshrc」に以下の1行を追加する。(ファイルが無い場合は作成する)

```
setenv WAIT_TIME_VALUE 300
```

- ・シェル以外の場合：ホームディレクトリの「.dtprofile」に以下の2行を追加する。(ファイルが無い場合は作成する)

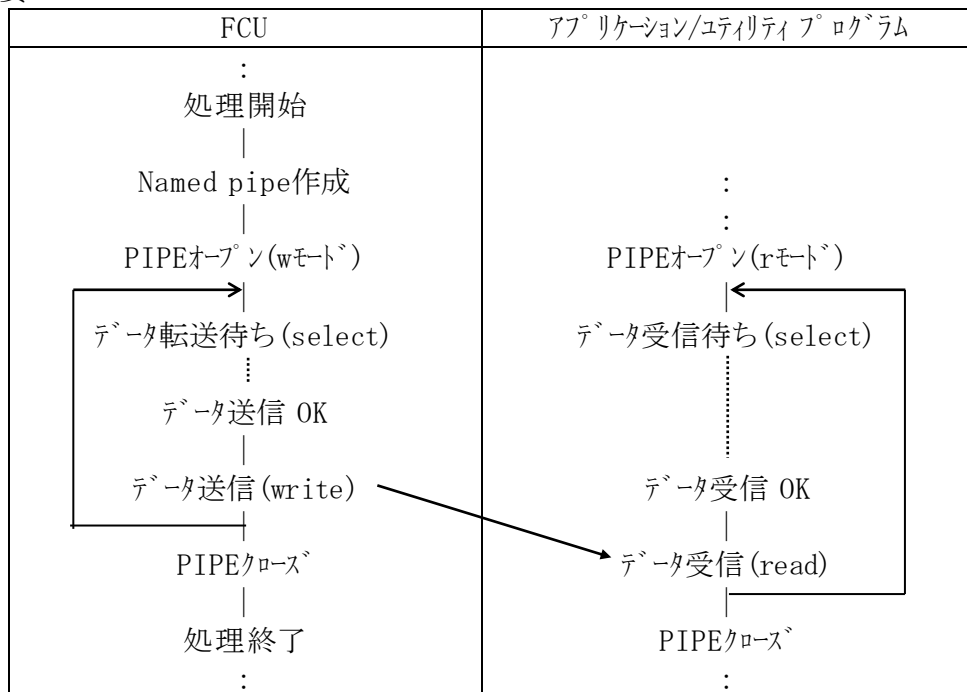
```
WAIT_TIME_VALUE=300
```

```
export WAIT_TIME_VALUE
```

注：共通デスクトップ環境がないマシンでは、「.profile」に上記行を追加してください。ファイルがない場合は作成してください。

なお、上記の設定後に一度ログアウトし、ログインし直して下さい。

処理シーケンス概要



→ : データの書き込み

→ : データ書き込み処理の繰返し

7.3 コード変換機能

コード変換が指定されている場合、データ実体のコード変換を行います。

コード変換の方法として、FCUが所持するコード変換テーブルを使用する方法とユーザがコード変換テーブルを指定する方法とがあります。

(1) FCUが所持するコード変換テーブルを使用する場合

サポートするコード変換は、EBCDIKとJIS8、EBCDICとASCIIの変換です。

コード変換は表7.3-1、表7.3-2に従って行います。

尚、MF側での文字表示内容は、MF側の端末等の仕様により下表と異なる場合がありますので、MF側マニュアルを参照してください。

表7.3-1 EBCDIK-JIS8 コード変換表

H L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL (00)	DLE (10)	(80)	(90)	SP (20)	& (26)	- (2D)	j (6A)	s (73)	ソ (BF)	w (77)	y (79)	{ (7B)	} (7D)	\$ (24)	0 (30)
1	SOH (01)	DC1 (11)	(81)	(91)	。 (A1)	⌘ (AA)	/ (2F)	k (6B)	ア (B1)	カ (C0)	- (7E)	z (7A)	A (41)	J (4A)	(9F)	1 (31)
2	STX (02)	DC2 (12)	(82)	SYN (16)	Γ (A2)	オ (AB)	b (62)	l (6C)	イ (B2)	キ (C1)	^ (CD)	(E0)	B (42)	K (4B)	S (53)	2 (32)
3	ETX (03)	DC3 (13)	(83)	(93)	∟ (A3)	ヤ (AC)	c (63)	m (6D)	ウ (B3)	ツ (C2)	ホ (CE)	(E1)	C (43)	L (4C)	T (54)	3 (33)
4	(9C)	(9D)	(84)	(94)	、 (A4)	ユ (AD)	d (64)	n (6E)	エ (B4)	テ (C3)	マ (CF)	(E2)	D (44)	M (4D)	U (55)	4 (34)
5	HT (09)	LF (0A)	(85)	(95)	・ (A5)	ヨ (AE)	e (65)	o (6F)	オ (B5)	ト (C4)	ミ (D0)	(E3)	E (45)	N (4E)	V (56)	5 (35)
6	(86)	BS (08)	ETB (17)	(96)	ヲ (A6)	ツ (AF)	f (66)	p (70)	カ (B6)	ナ (C5)	ム (D1)	(E4)	F (46)	O (4F)	W (57)	6 (36)
7	DEL (7F)	(87)	ESC (1B)	EOT (04)	ヲ (A7)	(A0)	g (67)	q (71)	キ (B7)	ニ (C6)	メ (D2)	(E5)	G (47)	P (50)	X (58)	7 (37)
8	(97)	CAN (18)	(88)	(98)	イ (A8)	- (B0)	h (68)	r (72)	ク (B8)	ヌ (C7)	モ (D3)	(E6)	H (48)	Q (51)	Y (59)	8 (38)
9	(8D)	EM (19)	(89)	(99)	ウ (A9)	a (61)	i (69)	´ (60)	ケ (B9)	ネ (C8)	ヤ (D4)	(E7)	I (49)	R (52)	Z (5A)	9 (39)
A	(8E)	(92)	(8A)	(9A)	[(5B)] (5D)	 (7C)	: (3A)	コ (BA)	ノ (C9)	ユ (D5)	レ (DA)	(E8)	(EE)	(F4)	(FA)
B	VT (0B)	(8F)	(8B)	(9B)	・ (2E)	¥ (5C)	, (2C)	# (23)	t (74)	u (75)	x (78)	□ (DB)	(E9)	(EF)	(F5)	(FB)
C	FF (0C)	FS (1C)	(8C)	DC4 (14)	< (3C)	* (2A)	% (25)	@ (40)	サ (BB)	v (76)	ヨ (D6)	ワ (DC)	(EA)	(F0)	(F6)	(FC)
D	CR (0D)	GS (1D)	ENQ (05)	NAK (15)	((28)) (29)	_ (5F)	´ (27)	シ (BC)	ハ (CA)	ラ (D7)	ン (DD)	(EB)	(F1)	(F7)	(FD)
E	SO (0E)	RS (1E)	ACK (06)	(9E)	+ (2B)	; (3B)	> (3E)	= (3D)	ス (BD)	ヒ (CB)	リ (D8)	´ (DE)	(EC)	(F2)	(F8)	(FE)
F	SI (0F)	US (1F)	BEL (07)	SUB (1A)	! (21)	^ (5E)	? (3F)	“ (22)	セ (BE)	フ (CC)	ル (D9)	´ (DF)	(ED)	(F3)	(F9)	E0 (FF)

() 内がJIS8コード

表7.3-2 EBCDIC-ASCII コード変換表

H L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL (00)	DLE (10)	DS (80)	(90)	SP (20)	& (26)	- (2D)	(BA)	(C3)	(CA)	(D1)	(D8)	{ (7B)	} (7D)	¥ (5C)	0 (30)
1	SOH (01)	DC1 (11)	SOS (81)	(91)	(A0)	(A9)	/ (2F)	(BB)	a (61)	j (6A)	(E5)	(D9)	A (41)	J (4A)	(9F)	1 (31)
2	STX (02)	DC2 (12)	FS (82)	SYN (16)	(A1)	(AA)	(B2)	(BC)	b (62)	k (6B)	s (73)	(DA)	B (42)	K (4B)	S (53)	2 (32)
3	ETX (03)	DC3 (13)	(83)	(93)	(A2)	(AB)	(B3)	(BD)	c (63)	l (6C)	t (74)	(DB)	C (43)	L (4C)	T (54)	3 (33)
4	PF (9C)	TM (9D)	BYP (84)	PN (94)	(A3)	(AC)	(B4)	(BE)	d (64)	m (6D)	u (75)	(DC)	D (44)	M (4D)	U (55)	4 (34)
5	HT (09)	(85)	LF (0A)	RS (95)	(A4)	(AD)	(B5)	(BF)	e (65)	n (6E)	v (76)	(DD)	E (45)	N (4E)	V (56)	5 (35)
6	LC (86)	BS (08)	ETB (17)	UC (96)	(A5)	(AE)	(B6)	(C0)	f (66)	o (6F)	w (77)	(DE)	F (46)	O (4F)	W (57)	6 (36)
7	DEL (7F)	IL (87)	ESC (1B)	EOT (04)	(A6)	(AF)	(B7)	(C1)	g (67)	p (70)	x (78)	(DF)	G (47)	P (50)	X (58)	7 (37)
8	GE (97)	CAN (18)	(88)	(98)	(A7)	(B0)	(B8)	(C2)	h (68)	q (71)	y (79)	(E0)	H (48)	Q (51)	Y (59)	8 (38)
9	RLF (8D)	EM (19)	(89)	(99)	(A8)	(B1)	(B9)	(60)	i (69)	r (72)	z (7A)	(E1)	I (49)	R (52)	Z (5A)	9 (39)
A	SMM (8E)	CC (92)	SW (8A)	(9A)	(D5)	(21)	(CB)	(3A)	! (C4)	: (5E)	(D2)	(E2)	(E8)	(EE)	(F4)	(FA)
B	VT (0B)	CUI (8F)	CUI (8B)	CU3 (9B)	. (2E)	\$ (24)	, (2C)	# (23)	(C5)	(CC)	(D3)	(E3)	(E9)	(EF)	(F5)	(FB)
C	FF (0C)	IFS (1C)	(8C)	DC4 (14)	< (3C)	* (2A)	% (25)	@ (40)	(C6)	(CD)	(D4)	(E4)	(EA)	(F0)	(F6)	(FC)
D	CR (0D)	IGS (1D)	ENQ (05)	NAK (15)	((28)) (29)	- (5F)	' (27)	(C7)	(CE)	[(5B)] (5D)	(EB)	(F1)	(F7)	(FD)
E	SO (0E)	IRS (1E)	ACK (06)	(9E)	+ (2B)	; (3B)	> (3E)	= (3D)	(C8)	(CF)	(D6)	(E6)	(EC)	(F2)	(F8)	(FE)
F	SI (0F)	IUS (1F)	BEL (07)	SUB (1A)	 (7C)	~ (7E)	? (3F)	“ (22)	(C9)	(D0)	(D7)	(E7)	(ED)	(F3)	(F9)	(FF)

() 内がASCIIコード

(2) ユーザがコード変換テーブルを指定する場合

ユーザが変換テーブルを作成し、テーブルを格納したファイルの名称をFCUのパラメータとして指定します。
(コード変換テーブルファイルの指定方法については7.4節を参照して下さい)

- ・コード変換テーブルは256バイトの大きさで、バイトデータです。
- ・1バイト単位のコード変換です(漢字コードのような2バイトコードの変換はできません)。
- ・コード変換テーブルは、変換前データの1バイトのコードを変換テーブルのオフセットとした位置に、変換後コードを埋め込むようにして作成します。

7.4 パディング機能

メインフレームからオープンシステムへの転送で、メインフレームデータセットのレコード形式が可変長の場合に、オープンシステムのファイルとしてはデータ長を一定(メインフレームデータセットで定義されている最大データ長)にそろえ、データ実体が不足する部分をパディングした形式で出力する機能です。

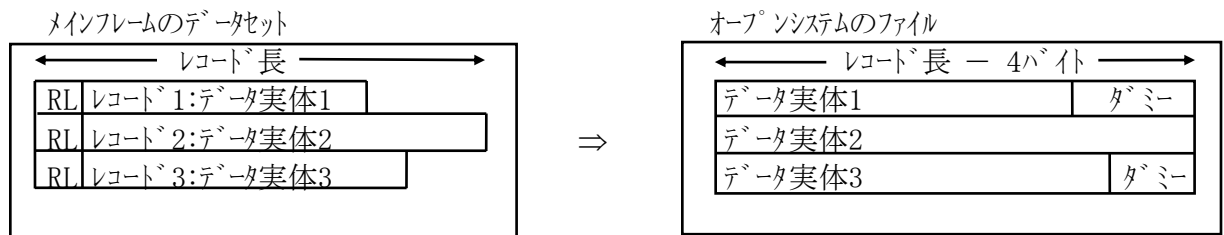
同様にオープンシステムからメインフレームへの転送で、メインフレームデータセットのレコード形式が固定長の場合に、データ長を一定(メインフレームデータセットのレコード長)にそろえ、データ実体が不足する部分をパディングした形式で出力することもできます。

- (注)・メインフレームからオープンシステムへの転送で、パディングされたオープンシステムのファイルをメインフレームに戻すことはできません。
- ・オープンシステムからメインフレームへの転送で、パディングされたメインフレームのデータセットをオープンシステムに戻すことはできません。
 - ・オープンシステムからメインフレームへの転送では、オープンシステムのファイル内の各データの末尾には、デリミタ(7.5節参照)が付加されている必要があります。

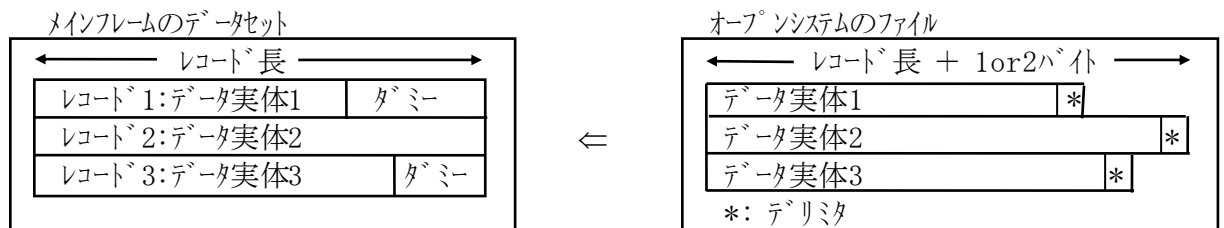
タミリアには以下のデータをパディングします。

- (1) コード変換が指定されていない場合(バイナリ形式のファイルとみなします)、0x00 をパディングします。
- (2) コード変換が指定されている場合(テキスト形式のファイルとみなします)、「空白」をパディングします。

① メインフレームからオープンシステムへの転送 (レコード形式=可変長の場合)

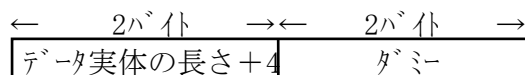


② オープンシステムからメインフレームへの転送 (レコード形式=固定長の場合)



(注)・レコード長: データセットの属性として定義されているレコード長です。レコード形式が可変長のデータセットでは、レコードの最大長を表しています。固定長のデータセットでは、全てのレコードが同じ長さです。

- ・RL: 各レコードの「データ実体の長さ+4バイト」の値を以下の形式で示しています。この部分はオープンシステムのファイルには転送されません。



- ・ブロックの長さを示している情報も転送されません。

図7.4-1 パディング時のデータ編集形式例

7.5 デリミタ付加機能

オープンシステムのファイル上で各データを区切るためのデリミタを付加する機能です。

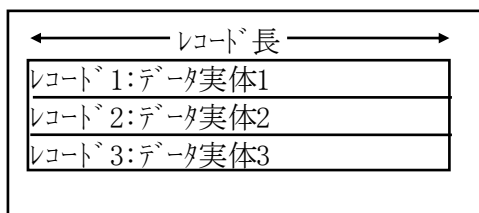
デリミタとしては、

- ・UNIX系OS の場合：“CR”(キャリッジリターン)または“LF”(ラインフィード)を使用します。
- ・Windowsの場合：“CR”(キャリッジリターン)＋“LF”(ラインフィード)を使用します。

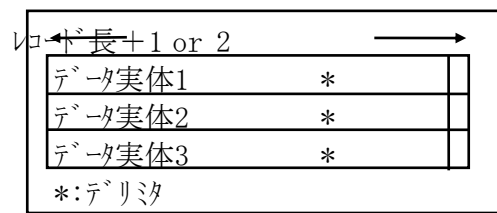
(1) データセットのレコード形式が固定長の場合のデータ編集形式

- ・メインフレームからオープンシステムへの転送では、データ実体の直後にデリミタを付加してオープンシステムのファイルに格納します。
- ・オープンシステムからメインフレームへの転送では、データセットのレコード長に従ってデータを切り出し、メインフレームのデータセットに格納します。(パディングなしの場合。パディングありの場合は、7.4節参照)
- ・デリミタは転送しません。
- ・レコード長で切り出したデータの直後(レコード長の直後)にデリミタがない場合は、エラーとなります。

メインフレームのデータセット



オープンシステムのファイル

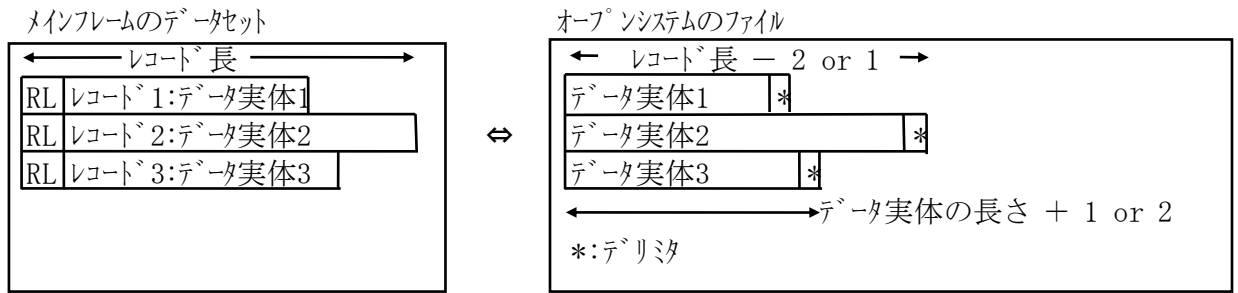


- (注) ・レコード長：データセットの属性として定義されているレコード長で、レコード実体の長さ一致しています。
- ・オープンシステムのファイルのデータ長は以下のようになります。
 - ・UNIX系OS の場合：レコード長＋デリミタ長(1 or 2バイト)
 - ・Windowsの場合：レコード長＋デリミタ長(2バイト)

図7.5-1 レコード形式＝固定長でデリミタ付加の場合のデータ編集形式

(2) メインフレームのデータセットのレコード形式が可変長でパディングなしの場合のデータ編集形式

- ・メインフレームからオープンシステムへの転送では、データ実体の直後にデリミタを付加してオープンシステムのファイルに格納します。
- ・オープンシステムからメインフレームへの転送では、デリミタの直前までのデータ実体だけを取り出してメインフレームのデータセットに格納します。
 - ・メインフレームのデータセットにデリミタは転送しません。
 - ・デリミタまでのデータ実体の長さが、メインフレームの(レコード長-4)よりも大きい場合、エラーとなります。
 - ・メインフレームのデータセットにデータを格納する際にRLを付加します。



- (注) ・レコード長：データセットの属性として定義されているレコード長で、レコードの最大長を表しています。
 ・オープンシステムのファイルのデータ長は以下のようになります。
 ・UNIX系OS の場合：データ実体の長さ+デリミタ長(1 or 2バイト)
 ・Windowsの場合：データ実体の長さ+デリミタ長(2バイト)
 ・RL：各レコードの「データ実体の長さ+4バイト」の値を以下の形式で示しています。
 RLはオープンシステムのファイルには転送しません。
 ブロックの長さを示している情報も転送されません。

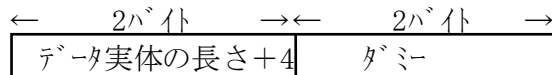
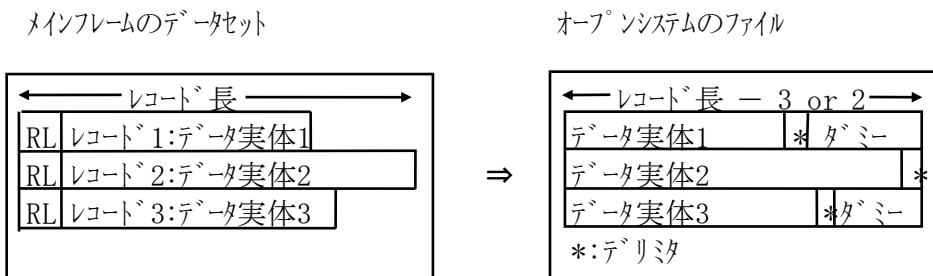


図 7. 5-2 レコード形式=可変長でパディングなし・デリミタ有りの場合のデータ編集形式

(3) メインフレームのデータセットのレコード形式が可変長でパディングありの場合のデータ編集形式

メインフレームからオープンシステムへの転送の場合に、データ実体の直後にデリミタを付加します。

- (注) パディングされたオープンシステムのファイルをメインフレームにもどした場合、ダミーエリアを削除しないため、元のデータセットと同じ形にはなりませんので注意が必要です。



- (注) ・レコード長：データセットの属性として定義されているレコード長で、レコードの最大長を表しています。
 ・オープンシステムのファイルのデータ長は以下のようになります。
 ・UNIX系OS の場合：レコード長-3
 ・Windowsの場合：レコード長-2
 ・RL：各レコードの「データ実体の長さ+4バイト」の値を以下の形式で示しています。
 RLはオープンシステムのファイルには転送されません。
 ブロックの長さを示している情報も転送されません。

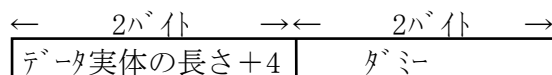
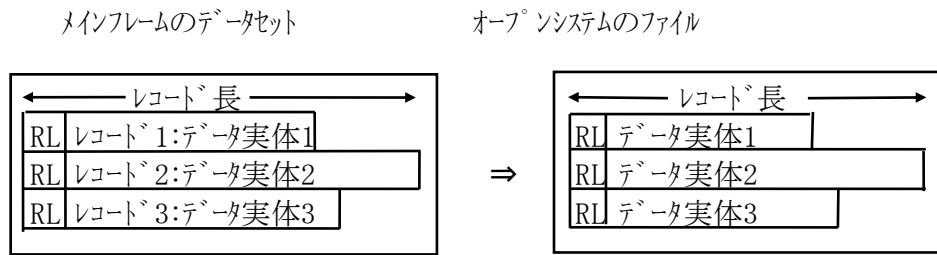


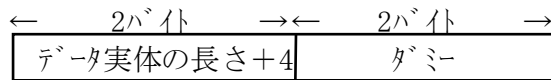
図 7. 5-3 レコード形式=可変長でパディング有り・デリミタ有りの場合のデータ編集形式

7.6 レコード長付加機能

メインフレームからオープンシステムにデータ転送を行う場合でレコード形式が可変長のとき、レコード長を付加した形でオープンシステムのファイルを作成します。



(注) ・レコード長：データセットの属性として定義されているレコード長で、レコードの最大長を表しています。
・RL：各レコードの「データ実体の長さ+4バイト」の値を以下の形式で示しています。



・ブロックの長さを示している情報は転送されません。

図7.6-1 レコード長付加機能

7.7 メインフレームオープンシステム双方向データ転送を行う場合の注意事項

- (1) レコード形式が可変長のメインフレームのデータセットをオープンシステムに転送し、さらに当該ファイルをメインフレーム側に戻して利用する場合、「デリミタ付加機能」を使用して下さい。「デリミタ付加機能」を使用しないと、元のデータセットと互換性のあるデータセットに戻すことができません。
- (2) デリミタと同じコードを持つデータをデリミタ以外の意味に使用しているデータセット/ファイルに対して「デリミタ付加機能」は使用しないで下さい。メインフレームからオープンシステムへのデータ転送は問題ありませんが、オープンシステムからメインフレームへのデータ転送では、本来デリミタでないものがデリミタとみなされ、意図しないデータセットができたり、FCUの処理でエラーになったりします。
- (3) メインフレームからオープンシステムへの転送時にパディングを指定したファイルを、メインフレームに戻すと元のデータセットとは互換性がないデータセットになります。また、オープンシステムからメインフレームへの転送時にパディングを指定したファイルをオープンシステムに戻すと元のファイルとは互換性のないファイルになります。これは、データ転送において、パディングしたデータを削除することができないためです。
- (4) レコード長付加機能を使用する場合、コード変換/デリミタ付加/パディング機能は使用できません。コード変換/デリミタ付加/パディング機能を指示するパラメータは全て「No」を指定して下さい。また、オープンシステムからメインフレームへのデータ転送時には本機能は使用できません。このため、レコード長を付加したファイルをメインフレームに戻すと、元のデータセットとは互換性のないデータセットが作成されます。

オープンシステムーオープンシステム間のデータ転送の場合

- (5) 中間ファイルーオープンシステム間のデータ転送時に、コード変換機能は使用できません。コード変換を指示するパラメータは、「No」を指定して下さい。

7.8 ログ強化機能

(1) ログ強化機能について

ログ強化機能によってOTM転送開始前、転送中、転送終了後のファイルサイズ及び更新日時を採取します。更に、転送前入力ファイルのコピー取得、転送前とエラー発生時のデータ属性情報をログファイルに出力することができます。また、ログ採取に失敗した場合は、OTM転送終了時にFCUをアボート終了することでcoreファイルを作成します。

(2) サポートバージョン

ログ強化機能をサポートする OS は AIX/Tru64/Solaris/Linux です。対象 OS バージョンは FAL/FCU でサポートしているバージョンとなります。対象の FAL/FCU バージョンは 01-XX-63 以降 (XX \geq 03) となります。以下のログ強化機能サポートバージョンを参照して下さい。

ログ強化機能サポートバージョン

対象形名	対象 OS バージョン	FAL/FCU バージョン
FAL for AIX(32/64bit)	対象 OS バージョンは FAL/FCU でサポートしているバージョンです	01-XX-63 以降 (XX \geq 03)
FAL for Solaris(32/64bit)		
FAL for Linux(32/64bit)		
FAL for HP-UX(64bit)		

(3)機能一覧

以下にログ強化機能の一覧を示します。

ログ強化機能一覧

項番	対象	機能	内容	備考
1	FCU	OPEN側入力ファイル監視モード設定機能	環境変数によってOPEN側入力ファイルの監視モードの通常/詳細を設定する	初期値は「通常」 詳細は(4-1)参照
2	FCU	OPEN側入力ファイルコピー作成モード設定機能	環境変数によってOTM転送開始前に作成するOPEN側入力ファイルのコピーの作成モードのON/OFFを設定する	初期値は「OFF」 詳細は(4-2)参照
3	FCU	トレースログファイル名設定機能	環境変数によってトレースログのファイル名を設定する	初期値は(4-3)参照
4	FCU	OPEN側入力ファイルコピー作成機能	OTM転送開始前に入力ファイルのコピーを作成する	詳細は(4-4)参照
5	FCU	OPEN側入力ファイル監視機能	OTM転送中にOPEN側入力ファイルが改ざんされていないか監視する	詳細は(4-5)参照
6	FCU	転送レコード長ログ出力機能	OTM転送で1レコード毎に処理されたレコード長をトレースログファイルへ出力する	詳細は(4-6)参照
7	FCU	FCUエラー(-363)発生時ログ出力機能	FCUエラー(-363)が発生時に転送開始日時/終了日時、OPEN側入力ファイルの情報、入出力バッファの内容をトレースログファイルに出力し、OPEN側入力ファイルのコピーを作成する	項番2、4の機能とは別にコピーを作成 詳細は(4-7)参照
8	FCU	coreダンプ出力機能	FCUエラー(-363)が発生時で且つ、トレースログファイルが何らかの理由により作成することが出来なかった場合、FCUをアボート終了させる事で情報収集用のcoreダンプファイルを作成する	ディスクの空き領域が足りなかった場合等 詳細は(4-8)参照

(4) 環境変数の定義

(4-1) OPEN 側入力ファイル監視モード設定機能(FCU)

環境変数 FCU_E363_TRACE_MODE を設定することで、OPEN 側入力ファイルの監視モードを設定することができます。以下に FCU_E363_TRACE_MODE と監視モードの関係を示します。

FCU_E363_TRACE_MODE と監視モードの関係

項番	FCU_E363_TRACE_MODE	監視モード
1	環境変数未定義	OTM 転送の開始前と終了後に OPEN 側入力ファイルの情報を取得し、相違点がないか調べる。相違点が合った場合には転送終了後に FCU エラー(-363)とする
2	ON	OTM 転送の開始前と終了後及び、1レコード処理毎に OPEN 側入力ファイルの情報を取得し、相違点がないか調べる。相違点が合った場合にはその時点で FCU エラー(-363)とする
3	その他の値	未定義時と同じ

(4-2) OPEN 側入力ファイルコピー作成モード設定機能(FCU)

環境変数 FCU_E363_TRACE_COPY_FILE を設定することで、OTM 転送開始前に OPEN 側入力ファイルのコピーを作成するかどうかを設定することができます。以下に FCU_E363_TRACE_COPY_FILE とコピー作成モードの関係を示します。

FCU_E363_TRACE_COPY_FILE とコピー作成モードの関係

項番	FCU_E363_TRACE_COPY_FILE	コピー作成モード
1	環境変数未定義	OTM 転送開始前に OPEN 側入力ファイルのコピーを作成しない
2	ON	OTM 転送開始前に OPEN 側入力ファイルのコピーを作成する
3	その他の値	未定義時と同じ

(4-3) トレースログファイル名設定機能(FCU)

環境変数 FCU_E363_TRACE_LOG を設定することで、トレースログのファイル名を設定することができる。以下に FCU_E363_TRACE_LOG とトレースログ出力ディレクトリ関係を示します。

FCU_E363_TRACE_LOG とトレースログ出力ディレクトリ関係

項番	FCU_E363_TRACE_LOG	トレースログファイル名
1	環境変数未定義	FCU 起動時のカレントディレクトリ + "FCU_E363." + ログ採取開始日時(yymmddHHMMSS) + FCU 起動時のプロセス ID + ".log"
2	任意のファイル名	任意のファイル名 + "." + ログ採取開始日時(yymmddHHMMSS) + FCU 起動時のプロセス ID + ".log" 尚、任意のファイル名にディレクトリ名が含まれる場合、“/”から始まっていればフルパス指定、それ以外はカレントディレクトリからの指定とみなされる
3	その他の値	未定義時と同じ

(4-4) OPEN 側入力ファイルコピー作成機能(FCU)

環境変数 FCU_E363_TRACE_COPY_FILE に ON が設定されていた場合、OTM 転送の開始前で且つ OPEN 側入力ファイルの情報取得後に、OPEN 側入力ファイルのコピーを作成します。

コピーされたファイル名は「OPEN 側入力ファイル名 + "." + ログ採取開始日時(yymmddHHMMSS) + FCU 起動時のプロセス ID + ".cpy"」となります。

尚、このコピーされたファイルは FCU エラー(-363)が発生せずに OTM 転送が正常終了した場合には削除されません。

(4-5) OPEN 側入力ファイル改ざん監視機能

OTM 転送中に OPEN 側入力ファイルが改ざんされていないか監視します。

環境変数 FCU_E363_TRACE_MODE が未定義または ON 以外の値が設定されていた場合には、OTM 転送開始前と転送終了時に OPEN 側入力ファイルの情報を取得し、i-node 番号、ファイル更新日時、ファイルサイズに相違点がないか調べます。相違点が合った場合には OTM 転送終了後に FCU エラー(-363)とします。

また、環境変数 FCU_E363_TRACE_MODE に ON が設定されていた場合には、OTM 転送開始前と転送終了後及び、1レコード処理毎に OPEN 側入力ファイルの情報を取得し、i-node 番号、ファイル更新日時、ファイルサイズに相違点がないか調べます。相違点が合った場合にはその時点で FCU エラー(-363)とします。

(4-6) 転送レコード長ログ出力機能

OTM 転送中に OPEN 側入力ファイルから読み込まれた 1レコードのレコード長をトレースログファイルに出力します。尚、このトレースログファイルは FCU エラー(-363)が発生せずに OTM 転送が正常終了した場合には削除されます。

(4-7) FCU エラー(-363)発生時ログ出力機能

OTM 転送において FCU エラー(-363)が発生した場合には、OTM 転送の開始時刻及びエラー発生時刻、OPEN 側入力ファイルの情報、OTM 転送時に使用していた入出力バッファの内容をトレースログファイルに出力します。

また、OPEN 側入力ファイルのコピーを作成する。コピーされたファイル名は「OPEN 側入力ファイル名 + "." + ログ採取開始日時(yymmddHHMMSS) + FCU 起動時のプロセス ID + ".end.cpy"」となります。

(4-8) core ダンプ出力機能

OTM 転送において FCU エラー(-363)が発生した場合で且つ、ディスクの空き領域が足りなかった等の理由によりトレースログファイルを作成することが出来なかった場合には、OTM 転送の開始時刻及びエラー発生時刻、OTM 転送開始前の OPEN 側入力ファイルの情報及び、エラー発生時の OPEN 側入力ファイルの情報を画面に出力後、FCU をアボート終了させる事で情報収集用の core ダンプファイルを作成します。

尚、複数の転送パラメータがパラメータファイルから指定されていても、アボート終了をした時点でそれ以降の転送は行われずに異常終了します。

また、core ダンプファイルを作成するためには、OTM 転送開始前に「ulimit -c unlimited」とし、core ダンプファイルのサイズを無制限にする必要があります。

(5) ディスク容量の確認

ログ採取に必要なディスク容量を以下に示します。本プログラムを実行する前に必ず以下の条件を確認して下さい。以下条件のディスク容量を満たしていない場合、ログが正常に採取できない場合があります。

- OPENサーバ側の入力ファイルを格納するディレクトリ容量
→OPENサーバ側の入力ファイルのコピーを作成する為、入力ファイルのサイズと同等以上のディレクトリを準備して下さい。
- FCU起動時のカレントディレクトリの容量
→coreファイルの生成に約15MBを準備して下さい。
- トレースログを出力するディレクトリ容量
→下記の計算式によりトレースログのファイルサイズを計算しディレクトリ容量を準備して下さい。
 $100\text{KB}(*1) + (\text{OPEN側入力ファイルのレコード件数}) \times 2\text{Byte}$
* : データセットのレコード長(RL)及び転送パラメータで可変しますが最大で100KB分必要となります。
例) OPEN側入力ファイルの総レコード件数が100,000レコードの場合、
 $100\text{KB} + (100,000\text{レコード}) \times 2\text{Byte}$ で、最大で約296KBとなります。

(6) エラー発生時に必要となるファイル

実際にエラーが発生した場合は以下のファイルを採取して下さい。

- トレースログファイル
→FCU起動時のカレントディレクトリか、環境変数”FCU_E363_TRACE_LOG”が示すディレクトリ内に存在します。
- OPEN側入力ファイルのコピー
→OPEN側入力ファイルと同じディレクトリに存在します。
- coreファイル
→FCU起動時のカレントディレクトリに存在します。
- エラー発生時の画面のハードコピー
- FALエラー、FALエラーのバックアップ(1世代前のFALエラー)
→/tmpか、環境変数ERR_LOG_FILEが示すディレクトリ内に存在します。
- FALダンプ及びFALダンプのバックアップ(1世代前のFALダンプ)
→/tmpか、環境変数ERR_DUMP_FILEが示すディレクトリ内に存在します。
- FCUデータ転送時のパラメータ
- 対象ボリュームのLISTVOLの結果

(7) 特記事項

本機能はMTO転送時には動作しません。

7.9 競合時に発生する各OSのシステムエラー

表 7.12-1 M/F との競合に対する各OSのシステムエラー

	AIX(32/64bit)	Solaris(32/64bit)	Linux(32/64bit)
M/F	16 (EBUSY)	5 (EIO)	5 (EIO)

表 7.12-2 AIXとの競合による各OSのシステムエラー

	AIX(32/64bit)	Solaris(32/64bit)	Linux(32/64bit)
AIX	16 (EBUSY)	5 (EIO)	5 (EIO)

Windowsでは、システムエラーは発生しないがREAD/WIRITEが正常に出来ないためFALエラー(-27)が発生する。

7.10 FCUリザーブトライ機能

新規にリザーブトライ機能をご使用になられる場合は、FCUリザーブトライ機能をご使用されることを推奨致します。FCUリザーブトライ機能は、従来版でサポートしておりましたFALリザーブトライ機能と同等機能・性能となっております。また、FALリザーブトライ機能は、本プログラムをインストールするサーバ環境(OS, HBAドライバ等)の影響で正しく動作を行えないことが御座いました。FCUリザーブトライ機能ではサーバ環境に影響されることなく動作するよう改善を行っております。

※ FCUリザーブトライ機能及びFALリザーブトライ機能を同時に有効と設定が可能ですが、両リザーブトライ機能を有効と設定した場合には、FCUリザーブトライ機能が動作致します。

(1) FCUリザーブトライ機能について

従来、FAL/FCUによるデータ転送時に対象の中間ボリュームがリザーブされているとシステムエラーが発生して異常終了してしまいました。しかし、FCUリザーブトライ機能は環境変数を定義することにより、指定秒数間隔で指定回数分リトライを行い、その間にリザーブが解除されればデータ転送を最初から開始し再開することが可能となります。

注1) FALリザーブトライ機能とFCUリザーブトライ機能の両方を有効化した場合は、FCUリザーブトライ機能が有効となりFALリザーブトライ機能は無効となります。

(2) サポートバージョン

FCUリザーブトライ機能をサポートするOSとOSバージョン及びFAL/FCUサポートバージョンを以下に示します。

※ 下記対象OSバージョン以外で本機能をご使用になる場合は、弊社担当営業もしくは弊社サポートサービスに契約している場合はサポートサービスにお問い合わせ下さい

FCU リザーブ リトライ機能サポートバージョン

対象形名	対象 OS バージョン	FAL/FCU バージョン
FAL for AIX(32/64bit)	4.3.3 ML02(32/64bit)※4、5.1 ML02(32/64bit)、 5.2 ML01(32/64bit)、5.3 ML01(32/64bit)、 6.1 TL01(32/64bit)※2、7.1(32/64bit)、7.2(32/64bit)	01-XX-65 以降 (XX ≥03)
FAL for Solaris(32/64bit)	8(32/64bit)、9(32/64bit)、10(32/64bit)、11(32/64bit)	
FAL for Linux(32/64bit)	Redhat AS3.0 Update3(32bit) (x86)(AMD64/Intel64) AS3.0 Update4(64bit) AS4.0 Update5(32bit) (x86)(AMD64/Intel64)※1 AS4.0 Update5(64bit)※1 AS5.1-AS5.4(32bit) (x86)(AMD64/Intel64)※1 AS5.1-AS5.4(64bit)※1 AS6.1-AS6.4(32bit) (x86)(AMD64/Intel64)※1	
	SuSE SLES9 (x86)(Itanium)(AMD64/Intel64) SLES10 (x86)(Itanium)(AMD64/Intel64)※1 SLES11 (x86)(Itanium)(AMD64/Intel64)※1	
FAL for HP-UX(64bit)	11.11[2002/06](32/64bit)、 11.23[2004/09版]/[2005/04版](64bit) 11.31[2007/02版](64bit)	
FAL for Windows(32/64bit)	NT4.0 SP6(x86)※4、2000Server SP4(x86)※4、 Server2003 SP1 (x86)(Itanium)(AMD64/Intel64)、 Server2008 (x86)(Itanium)(AMD64/Intel64)※3 Server2008 R2 (Itanium)(AMD64/Intel64)※3 Server2012 (AMD64/Intel64)※5 Server2012 R2 (AMD64/Intel64)※5 Server2016 (AMD64/Intel64)※6 Server2019 (AMD64/Intel64)※6	32bit 版 は 01-XX-65 以降 (XX≥03) 64bit 版 は 01-XX-66 以降

※1 Linux の AS4.0 Update5 以降及び AS5.1-AS5.4、AS6.1-AS6.4、SLES10、SLES11 に対する FAL/FCU サポートバージョンは 01-XX-66/26 以降 (XX≥05) です。

※2 AIX6.1 以降に対する FAL/FCU サポートバージョンは 01-XX-65/25 以降 (XX≥03) 及び 01-XX-66/25 以降 (XX≥05) です。

※3 Windows Server2008/Server2008 R2 に対する FAL/FCU サポートバージョンは 01-XX-66/24 以降 (XX≥05) です。

※4 01-06-67/21 22 以降は非サポート。

※5 Windows Server2012/Server2012 R2 に対する FAL/FCU サポートバージョンは 01-XX-66/26 以降 (XX≥05)、
01-XX-67/24 以降 (XX≥06)、01-XX-68/01 以降 (XX≥07) です。

※6 Windows Server2016/Server2019 に対する FAL/FCU サポートバージョンは 01-XX-67/26 以降 (XX≥06)、
01-XX-68/01 以降 (XX≥07)、01-XX-69/00 以降 (XX=08) です。

(3) 機能一覧

以下に FCU リザーブトライ機能の一覧を示します。

FCU リザーブトライ機能一覧

項番	対象	機能	内容
1	FCU	リトライ有効化設定機能	環境変数によってリトライの有効化/無効化を設定する
2		リトライ回数設定機能	環境変数によって最大リトライ回数を設定する
3		リトライ待ち時間設定機能	環境変数によってリトライ時の待ち時間を設定する
4		リトライ対象エラーコード設定機能	環境変数によってリトライの対象となるシステムエラーコードを設定する
5		リトライ機能	中間ボリュームで対象となるシステムエラーが発生した時、リトライを行う。
6		画面出力/ログ出力機能	システムエラーにより、何回リトライを行ったか FAL エラーログ (UNIX)、prm.log (Windows) に出力する。

(4) 環境変数の定義

(4-1) リトライ有効化設定機能 (FCU)

環境変数 FCU_RETRY を設定することで、FAL から中間ボリューム(MF/OPEN)に I/O を出した時に特定のシステムエラーが発生した場合、リトライを行うかどうか指定する事ができます。以下に FCU_RETRY と FCU リザーブトライ機能の有効化/無効化の関係を示します。

FCU_RETRY と FCU リザーブトライ機能の有効化/無効化の関係

項番	FCU_RETRY	リトライ回数
1	環境変数未定義	FCU リザーブトライ機能を無効化し、現行動作と同様にリトライを行わず、即時にエラーとする
2	ON	FCU リザーブトライ機能を有効化する
3	OFF	FCU リザーブトライ機能を無効化し、現行動作と同様にリトライを行わず、即時にエラーとする
4	その他の値	FCU リザーブトライ機能を無効化し、現行動作と同様にリトライを行わず、即時にエラーとする

注) FAL_RETRY=ON と FCU_RETRY=ON を両方定義した場合は、FCU_RETRY が有効となり FAL_RETRY は無効となります。

(4-2) リトライ回数設定機能(FCU)

環境変数 FCU_RETRY_COUNT を設定することで、FAL から中間ボリューム(MF/OPEN)に I/O を出した時に特定のシステムエラーが発生した場合、リトライを行う最大回数を指定する事ができます。以下に FCU_RETRY_COUNT とリトライ回数の関係を示します。

FCU_RETRY_COUNT とリトライ回数の関係

項番	FCU_RETRY_COUNT	リトライ回数
1	環境変数未定義	10 回
3	1～600	指定回数分までリトライを行う
4	その他の値	10 回

(4-3) リトライ待ち時間設定機能(FCU)

環境変数 FCU_RETRY_WAIT_TIME を設定することで、FAL から中間ボリューム(MF/OPEN)に I/O を出した時に特定のシステムエラーが発生した場合、リトライを行うまでの待ち時間を秒単位で指定する事ができます。以下に FCU_RETRY_WAIT_TIME とリトライ待ち時間の関係を示します。

FCU_RETRY_WAIT_TIME とリトライ待ち時間の関係

項番	FCU_RETRY_WAIT_TIME	リトライ待ち時間
1	環境変数未定義	1 秒間待つ
2	1～60	指定秒数間待つ
3	その他の値	1 秒間待つ

(4-4) リトライ対象エラーコード設定機能(FCU)

環境変数 FCU_RETRY_TARGET を設定することで、リトライを行う対象となるシステムエラーコードを AIX では 16(EBUSY)、その他 OS では 5(EIO)に加えて最大 5 つまで「,」で区切って指定する事ができます。尚、6 つ目以降は無効とします。また、数値以外の指定があった場合にはそれ以降の設定を全て無効とします。以下に FCU_RETRY_TARGET とリトライ対象となるシステムエラーコードの関係を示します。

※デフォルト以外のシステムエラーを指定する場合は、弊社担当営業もしくは弊社サポートサービスに契約している場合はサポートサービスにお問い合わせ下さい。

FCU_RETRY_TARGET とリトライ対象となるシステムエラーコードの関係

項番	FCU_RETRY_TARGET	リトライ対象となるシステムエラーコード		備考
1	環境変数未定義	AIX	16 (EBUSY)	
		Solaris	5 (EIO)	
		Linux		
		HP-UX	22 (EINVAL)	
			13 (EACCES)	
Windows	170 (ERROR_BUSY)			
2	システムエラー, システムエラー, システムエラー...	「,」で区切られた最大 5 つ (HP-UX は最大 4 つ) のシステムエラーコード		各 OS における初期値として 16 (EBUSY), 5 (EIO), 22 (EINVAL), 13 (EACCES) 及び 170 (ERROR_BUSY) は既に含まれている
3	その他の値	環境変数未定義時と同様		—

(5) 画面出力/ログ出力機能(FCU)

(5-1) リトライ情報画面出力機能(FCU-NW 版のみ)

システムエラーが発生したためにリトライ処理を行ったことを画面に出力する。出力する内容は、FCU エラー値、FAL エラー値、システムエラー値、リトライ回数で、リトライ処理を行う毎に出力する。

図 7.10-5.1. に画面出力の書式を示す。

(A) リトライ最大回数以内で正常終了した場合

```
# fcunw -nc -P mto VSN:DSN a.dat EA No LF
mto VSN:DSN a.dat EA No LF Emp=No RDW=No VSE=No PIPE=No
Now checking ...
Start
Processing ( 40%)
FCU Retry(-160, 0, 5)... count(1)
FCU Retry(-160, 0, 5)... count(2)
FCU Retry(-160, 0, 5)... count(3)
Processing ( 40%)
Processing ( 80%)
Complete
#
```

(B) リトライ最大回数を超えてエラーとなった場合

```
# fcunw -nc -P mto VSN:DSN a.dat EA No LF
mto VSN:DSN a.dat EA No LF Emp=No RDW=No VSE=No PIPE=No
Now checking ...
Start
Processing ( 40%)
FCU Retry(-160, 0, 5)... count(1)
FCU Retry(-160, 0, 5)... count(2)
FCU Retry(-160, 0, 5)... count(3)
Input file : Read error (-160)
( FAL Error      :    0 )
( System Error   :    5 )
#
```

図 7.10-5.1. 画面出力書式

(5-2) リトライ詳細ログ出力機能 (FCU)

システムエラーが発生したためにリトライ処理を行ったことを FAL エラーログファイルに出力する。出力する内容はリトライ回数で、処理ログの末尾に付加する。図 7. 10-5. 2. にリトライ詳細ログの書式を示す。

(Windows の場合システムエラーは、FAL エラーログに、FCU リトライ情報は prm. log の実行時間行右端に表示)

(A) リトライ最大回数以内で正常終了した場合			
Mon Jun 12 15:03:22 2006 : root : FAL Error (5) at 3595 : 01-04-64/26 PID=7110	システムエラー (5) でリトライ処理を 3 回行ったため、FAL エラーが 3 回発生 FCU リトライ発生時は、 ←通常の処理ログの末尾にリトライ回数を付加する	VSN:DSN L	
Mon Jun 12 15:03:23 2006 : root : FAL Error (5) at 3595 : 01-04-64/26 PID=7110		VSN:DSN L	
Mon Jun 12 15:03:24 2006 : root : FAL Error (5) at 3595 : 01-04-64/26 PID=7110		VSN:DSN L	
06/12 15:03:22 15:03:24 VSN:DSN a.dat 125000 125000 0 0 mto L FCU_retry_count(3)			
(B) リトライ最大回数を超えてエラーとなった場合			
Mon Jun 12 15:03:22 2006 : root : FAL Error (5) at 3595 : 01-04-64/26 PID=7110	システムエラー (5) でリトライ処理を 3 回行ったため、FAL エラーが 3 回発生 FCU リトライ発生時は、 ←通常の処理ログの末尾にリトライ回数を付加する	VSN:DSN L	
Mon Jun 12 15:03:23 2006 : root : FAL Error (5) at 3595 : 01-04-64/26 PID=7110		VSN:DSN L	
Mon Jun 12 15:03:24 2006 : root : FAL Error (5) at 3595 : 01-04-64/26 PID=7110		VSN:DSN L	
06/12 15:03:22 15:03:24 VSN:DSN a.dat 5271 5271 0 0 mto L FCU_retry_count(3)			
Mon Jun 12 15:03:25 2006 : root : FCU Error (-160, 0, 5) PRM=mto, EA, No, LF, Emp=No, RDW=No, VSE=No, PIPE=No L			

図 7. 10-5. 2. FAL エラーに出力する FCU リトライ詳細ログの書式

(6) 異常処理

以下に本機能で発生するエラーを示します。

エラー		エラーメッセージ	内容	備考
コード	名称			
16	EBUSY	Resource busy	Reserve が取得されているデバイスを開しようとした。	AIX のみ
5	EIO	I/O error	デバイスが使用不可	—
22	EINVAL	Invalid argument	引数不正	HP のみ
13	EACCES	Permission denied	アクセス権限がない	
170	ERROR_BUSY	The requested resource is in use.	要求されたリソースは使用中です。	Windows のみ

7.11 OTM 終了時インターバル機能 (Linux のみ)

OTM 終了時インターバル機能は、Linux 版 FAL/FCU の OTM 転送機能において OTM 転送正常終了時に指定された時間 (秒) でインターバルをとった後に OTM 転送を終了する機能です。

OTM 転送を行った直後にメインフレーム側から当該ターゲットに対して I/O を出す場合、本機能によって OTM 転送終了後にインターバルをとる必要があります。(RHEL7 以降ではインターバルは必須ではありません。)

(1) OTM 終了時インターバル機能について

OTM 終了時インターバル機能は、1 回の OTM 転送が終了する度に環境変数 FCU_OTM_INTERVAL に設定した秒数間分のインターバルをとってから OTM 転送を終了します。もし、OTM 転送時に何らかのエラーが発生した場合には、インターバルをとらずに異常終了処理を行って即時終了します。

パラメータファイルによる複数転送指定時においてもそれぞれの OTM 転送終了毎にインターバルをとります。

なお、MTO 転送の終了時には、インターバルはとりません。

(2) サポートプラットフォーム

以下にサポートするオープンプラットフォームを示します。

サポートするオープンプラットフォーム

プラットフォーム		OS	File Exchange FAL/FCU	備考
LINUX	Redhat	7.2 (32bit)	32bit 版	<ul style="list-style-type: none"> 32bit 版の CPU アーキテクチャは x86 のみ (7.2, AS2.1) 32bit 版の CPU アーキテクチャは x86 (32bit), AMD64/Intel64 (64bit) (AS3.0 以降) 64bit 版の CPU アーキテクチャは AMD64/Intel64 (64bit) のみ (Itanium はサポートしていません)
		AS2.1 (32bit)		
		AS3.0		
	SuSE	AS4.0	32bit 版	
		AS5.1-AS5.4		
		AS6.1-AS6.4		
SuSE	AS2.1 (64bit)	64bit 版 (IA64 AS2.1 用)		
	AS3.0 (64bit)	64bit 版 (IA64 AS3.0 用)		
	AS4.0 (64bit)			
SuSE	AS5.1-AS5.4 (64bit)			
	SLES9	64bit 版 (IA64 AS3.0 用)		
	SLES10			
SLES11				
SuSE	SLES9 (64bit)	64bit 版 (IA64 AS3.0 用)		
	SLES10 (64bit)			
	SLES11 (64bit)			

※ AS4.0 Update5 以降と AS5.1-AS5.4, AS6.1-AS6.4, SLES10, SLES11 については 01-XX-66/26 以降 (XX ≥ 05) でのサポートとなります。

(3) 機能一覧

以下に OTM 終了時インターバル機能一覧を示します。

OTM 終了時インターバル機能一覧

項番	対象	機能	内容
1	FCU	インターバル時間設定機能	環境変数 FCU_OTM_INTERVAL によってインターバル時間を設定する。
2		OTM 終了時インターバル機能	1 回の OTM 転送が正常終了後、指定した時間 (秒) でインターバルをとってから OTM 転送を完了する。
3		インターバル中画面出力機能	インターバルをとっていることを画面に出力する。

(4) 環境変数の定義

(4-1) インターバル時間設定機能 (FCU)

環境変数 FCU_OTM_INTERVAL を設定することで、OTM 転送が正常終了する度にとるインターバルの時間を指定します。以下に FCU_OTM_INTERVAL とインターバル時間の関係を示します。

FCU_OTM_INTERVAL とインターバル時間の関係

項番	FCU_OTM_INTERVAL	インターバル時間	備考
1	環境変数未定義	OTM 転送が正常終了後、インターバルをとらずに終了する。	
2	60～600 の値	OTM 転送が正常終了後、指定秒数間のインターバルをとってから終了する。	・ 値の単位は秒。
3	その他の値	OTM 転送が正常終了後、インターバルをとらずに終了する。	

(5) 画面出力 (FCU)

(5-1) インターバル中画面出力機能 (FCU)

インターバル中であることを画面に出力します。図 7.11-5.1. に画面出力の書式を示します。

```
# fcunw -nc -P otm a.dat VSN:DSN EA No LF
otm a.dat VSN:DSN EA No LF Emp=No RDW=No VSE=No PIPE=No
Now checking ...
Start
Processing ( 40%)
Processing ( 80%)
OTM Interval ...
Complete
#
```

図 7.11-5.1. インターバル中画面出力機能

8 FCU の使用方法

8.1 環境設定 (共用ボリュームの定義)

FCUからアクセスするメインフレームのデータセットが格納されているボリュームは「共用ボリューム」として定義しておかなければなりません。共用ボリュームのタイプ (デバイスエミュレーションタイプ) については1.2.2節を参照して下さい。

(注1) ・オープンシステム側でラベル付けを行う際には、メインフレーム側のボリュームに合わせてデータシリンダを定義して下さい。一致していないと正しいデータ処理ができなくなります。

・オープンシステム側のOSがSolarisの場合、デバイスエミュレーションに合わせて以下のシリンダをデータシリンダとして定義して下さい。

3390-3A : シリンダ 番号0から3345	6588-3A : シリンダ 番号0から3450
3390-3B : シリンダ 番号0から3339	6588-3B : シリンダ 番号0から3436
3390-3C : シリンダ 番号0から3345	6588-3C : シリンダ 番号0から3450
3380-KA : シリンダ 番号0から2661	6586-KA : シリンダ 番号0から2661
3380-KB : シリンダ 番号0から2655	6586-KB : シリンダ 番号0から2655
3380-KC : シリンダ 番号0から2661	6586-KC : シリンダ 番号0から2661
3390-9A : シリンダ 番号0から10035	3390-LA : シリンダ 番号0から32763
3390-9B : シリンダ 番号0から10017	3390-LB : シリンダ 番号0から32760
3390-9C : シリンダ 番号0から10035	3390-LC : シリンダ 番号0から32763
3380-3A : シリンダ 番号0から3342	6588-9A : シリンダ 番号0から10035
3380-3B : シリンダ 番号0から3339	6588-9B : シリンダ 番号0から10017
3380-3C : シリンダ 番号0から3342	6588-9C : シリンダ 番号0から10035
6588-LA : シリンダ 番号0から32763	
6588-LB : シリンダ 番号0から32760	
6588-LC : シリンダ 番号0から32763	

OPEN-3/8/9/K/Eボリュームは、フォーマットするシリンダ数+4を指定して下さい。

フォーマット可能なシリンダ数は § 2.3 を参照願います。

OPEN-L/M/Vボリューム及びLUSEボリュームに関しては、ジオメトリの設定値が異なりますので詳細はSolarisのSCSI/Fibre Installation Manualを参照して下さい。

(注2) ・オープンシステム側のOSがSolarisの場合、ボリュームのフォーマット処理のラベルの書込みで以下のワーニングメッセージが出力されることがありますが、問題ありませんので無視して下さい。

Warning: error writing VTOC.

Warning: no backup labels

Write label failed

(注3) ・Solaris/NCR UNIXはボリュームのラベルが必須であり、Windows/Digital UNIXは指定によりラベルを書き込みます。このため、SolarisとNCR UNIX間、Solaris間とNCR UNIX間でも同じボリュームを共用することはできません。またWindows/Digital UNIXでラベルを書き込むとSolaris/NCR UNIXとWindows/Digital UNIXの間、Windows間とDigital UNIX間で同じボリュームを共用することができなくなります。

(注 : Digital UNIXの記載部分にはTru64 UNIXを含みます。)

(注4) ・共用ボリュームに対するアクセス権は、デフォルトの状態ではrootに対してのみ与えられています。ユーザがFAL/FCUを使用するためには、あらかじめ共用ボリュームに対してアクセス権を設定しておく必要があります。

(注5) ・一度ラベル付けを行ったボリュームのデバイスエミュレーションタイプを変更した場合には、再度ラベル付けを行って下さい。

・以下のデバイスエミュレーションタイプの変更を実施した場合、Solarisでは、リブート後に古いデバイスエミュレーションタイプが表示されますが、無視して新タイプに対応したラベル付けを再実行して下さい。

- ・3390-3A→3390-3C / 3390-3B→3390-3A / 3390-3B→3390-3C / 3390-3C→3390-3A
- ・3380-KA→3380-KC / 3380-KB→3380-KA / 3380-KB→3380-KC / 3380-KC→3380-KA
- ・3380-3A→3390-3C / 3380-3B→3390-3A / 3380-3B→3380-3C / 3380-3C→3380-3A
- ・6588-3A→6588-3C / 6588-3B→6588-3A / 6588-3B→6588-3C / 6588-3C→6588-3A
- ・6586-KA→6586-KC / 6586-KB→6586-KA / 6586-KB→6586-KC / 6586-KC→6586-KA
- ・3390-9A→3390-9C / 3390-9B→3390-9A / 3390-9B→3390-9C / 3390-9C→3390-9A
- ・3390-LA→3390-LC / 3390-LB→3390-LA / 3390-LB→3390-LC / 3390-LC→3390-LA
- ・6588-9A→6588-9C / 6588-9B→6588-9A / 6588-9B→6588-9C / 6588-9C→6588-9A
- ・6588-LA→6588-LC / 6588-LB→6588-LA / 6588-LB→6588-LC / 6588-LC→6588-LA

・以下のデバイスエミュレーションタイプの変更を実施した場合、Solarisでは、リブート後に「corrupt label -bad geometry」のエラーメッセージが出力されますが、無視して新タイプに対応したラベル付けを再実行して下さい。

- ・3390-3A→3390-3B / 3390-3C→3390-3B
- ・3380-KA→3380-KB / 3380-KC→3380-KB
- ・3380-3A→3380-3B / 3380-3C→3380-3B
- ・6588-3A→6588-3B / 6588-3C→6588-3B
- ・6586-KA→6586-KB / 6586-KC→6586-KB
- ・3390-9A→3390-9B / 3390-9C→3390-9B
- ・3390-LA→3390-LB / 3390-LC→3390-LB
- ・6588-9A→6588-9B / 6588-9C→6588-9B
- ・6588-LA→6588-LB / 6588-LC→6588-LB

(注6) ・メインフレームがMVS/VSEで、以下に示すデバイスエミュレーションタイプの変更を実施した場合、中間ボリュームとして使用できない場合があります。

- ・3390-3R→3390-3A / 3390-3R→3390-3B / 3390-3R→3390-3C

これに対処する為には、下記のDSF(Device Support Facilities)コマンドを実行して下さい。

- ・ボリュームのデータを残したい時 : "REFORMAT"コマンドの "REFVTOC"パラメータ指定
- ・ボリュームのデータを残す必要の無い時 : "INIT"(イニシャリス)コマンド

8.2 ボリューム定義ファイルの作成

ボリューム定義ファイルのファイル名は「datasetmount.dat」です。

FCU実行時のカントディレクトリに作成して下さい。

UNIX系OSの場合、1個のボリューム定義ファイル中に999件までの定義が可能です。

(999件を超えるパラメータを定義した場合、1000件目の処理でエラーとなります。)

Windows系の場合メモ容量に依存します。

ボリューム定義ファイルの1行の文字数は、2079文字以内で記載して下さい。

定義したボリュームに新たにデータセットを割り当てた場合は、VTOCのDSCB1にあるデータセットインジケータの最終ボリュームフラグを確定させるため、必ずM/F側から一度データの書込みを行ってから転送に使用して下さい。

8.2.1 メインフレーム-オープンシステム間のデータ交換の場合

メインフレーム上でのボリュームとオープンシステム上でのボリューム(パーティション)とを対応付けるための情報を以下の形式で格納したファイルを作成します。

01-XX-48以降のバージョンでは、ボリューム定義ファイル上同一のVSNが存在しても、VSN識別子を指定することにより、両方のボリュームにアクセス可能となりました。01-XX-47以前のバージョンではボリューム定義ファイル上、上に定義されているボリュームに対して、処理を行っていました。

VSN識別子は英数字、英記号を使用できます。

01-XX-48以降における、ボリューム定義ファイルと、パラメータ定義ファイル/画面指定との関係

項番	ボリューム定義ファイル 画面又は パラメータ定義ファイル	同一 VSN あり		
		VSN 識別子なし	VSN 識別子あり	VSN 識別子あり・なし混在
1	VSN 識別子指定なし	最初に VSN が一致した行が対象となる	最初に VSN が一致した行が対象となる	最初に VSN が一致した行が対象となる
2	VSN 識別子指定あり	該当行なし	VSN と VSN 識別子が一致した行が対象となる	VSN と VSN 識別子が一致した行が対象となる

(注) 本機能はOT0時も可能ですが、同一RAIDに複数のメインフレームが接続された構成において同一VSNが存在してもメインフレーム-オープンシステム間のデータ交換を可能とする為に設けたものです。

(1) AIXの場合

①	②	③	④	⑤	⑥
/dev/rhdiskn1	aaaaaa	3390-3A	MFN	MVS01	改行
/dev/rhdiskn2	aaaaaa	3390-3B	MFN	MSP01	改行
/dev/rhdiskn3	cccccc	3390-3C			改行
/dev/rhdiskn4	dddddd	3380-KA	MFA		改行
/dev/rhdiskn5	eeeeee	3380-KB			改行
/dev/rhdiskn6	ffffff	3380-KC			改行
end					
⑦					

(注) 上記の記述例はメインフレームのOSがMVS/VSEの場合です。

③で指定するデバイスエミュレーションタイプは「1.2.2 サポート対象」のデバイスエミュレーションタイプを参照して下さい。

① オープンシステム上で定義したパーティション名を指定します。

FCU (FAL) はメインフレームのデータセットに対してraw I/Oでアクセスするため、rawデバイスとして定義する必要があります。ボリューム定義ファイルには複数のボリューム情報を指定することができます。ただし、複数の行に同じVSNが指定されている場合は、最初の行が選択されます。

(注1) JP1/HiCommand Dynamic Link Managerを使用する場合は、①の書式が異なります。JP1/HiCommand Dynamic Link Managerのソフトウェア添付資料を参照して定義してください。

② メインフレーム側で設定したボリュームシリアルナンバを、6桁以下の英文字及び英数字で指定します。

③ 共有ボリュームのデバイスエミュレーションタイプを指定します。
・デバイスエミュレーションタイプの種類については1.2.2節を参照して下さい。

④ MFAの指定があると、XSPとして動作します。なければXSP以外として動作します。
VSN識別子を使用する場合、XSP以外でも本パラメータは必須となります(MFNと指定して下さい)。

⑤ VSN識別子を35文字以内で指定します。同一VSNボリュームに複数アクセスする場合に指定します。省略可能です。

⑥ 「Return」キーを押下します。

⑦ end : ボリューム制御情報の終了を示します。
「end」と入力して下さい。

(注2) 各パラメータの間は、1個以上の「空白」で区切ります。

上記の例の中で、「_」が付いている箇所は実行環境に合わせた文字/文字列を入力します。「_」が付いていない箇所は例の通りに入力します。

(注3) 先頭の文字が「#」である行はコメント行とみなします。

(2) HP-UXの場合

①	②	③	④	⑤	⑥
/dev/rdisk/cu11tu12du13	aaaaaa	3390-3A	MFN	MVS01	改行
/dev/rdisk/cy11tv12dv13	aaaaaa	3390-3B	MFN	MSP01	改行
/dev/rdisk/cw11tw12dw13	cccccc	3390-3C			改行
/dev/rdisk/cx11tx12dx13	dddddd	3380-KA	MFA		改行
/dev/rdisk/cy11ty12dy13	eeeeee	3380-KB			改行
/dev/rdisk/cz11tz12dz13	ffffff	3380-KC			改行
end					
⑦					

(注) 上記の記述例はメインフレームのOSがMVS/VSEの場合です。

③で指定するデバイスエミュレーションタイプは「1.2.2 サポート対象」のデバイスエミュレーションタイプを参照して下さい。

① オープンシステム上で定義したパーティション名を指定します。

FCU (FAL) はメインフレームのデータセットに対してraw I/Oでアクセスするため、rawデバイスとして定義する必要があります。ボリューム定義ファイルには複数のボリューム情報を定義することができます。ただし、複数の行に同じVSNが指定されている場合は、最初の行が選択されます。

(注1) JP1/HiCommand Dynamic Link Managerを使用する場合は、①の書式が異なります。
JP1/HiCommand Dynamic Link Managerのソフトウェア添付資料を参照して定義してください。

(注2) Legacy DSF以外のデバイス・ファイル表記形式には対応していません。

② メインフレーム側で設定したボリュームシリアル番号を、6桁以下の英文字及び英数字で指定します。

③ 共有ボリュームのデバイスエミュレーションタイプを指定します。

・デバイスエミュレーションタイプの種類については1.2.2節を参照して下さい。

④ MFAの指定があると、XSPとして動作します。なければXSP以外として動作します。

VSN識別子を使用する場合、XSP以外でも本パラメータは必須となります(MFNと指定して下さい)。

⑤ VSN識別子を35文字以内で指定します。同一VSNボリュームに複数アクセスする場合に指定します。
省略可能です。

⑥ 「Return」キーを押下します。

⑦ end : ボリューム制御情報の終了を示します。

「end」と入力して下さい。

(注2) 各パラメータの間は、1個以上の「空白」で区切ります。

上記の例の中で、「_」が付いている箇所は実行環境に合わせた文字/文字列を入力します。「_」が付いていない箇所は例の通りに入力します。

(注3) 先頭の文字が「#」である行はコメント行とみなします。

(3) Solarisの場合

①	②	③	④	⑤	⑥
/dev/rdisk/cu11tu12du13su14	aaaaaa	3390-3A	MFN	MVS01	改行
/dev/rdisk/cw11tw12dw13sw14	aaaaaa	3390-3B	MFN	MSP01	改行
/dev/rdisk/cw11tw12dw13sw14	ccccc	3390-3C			改行
/dev/rdisk/cx11tx12dx13sx14	dddddd	3380-KA	MFA		改行
/dev/rdisk/cy11ty12dy13sy14	eeeeee	3380-KB			改行
/dev/rdisk/cz11tz12dz13sz14	ffffff	3380-KC			改行
end					
⑦					

(注) 上記の記述例はメインフレームのOSがMVS/VSEの場合です。

③で指定するデバイスエミュレーションタイプは「1.2.2 サポート対象」のデバイスエミュレーションタイプを参照して下さい。

① オープンシステム上で定義したパーティション名を指定します。

FCU (FAL) はメインフレームのデータセットに対してraw I/Oでアクセスするため、rawデバイスとして定義する必要があります。ボリューム定義ファイルには複数のボリューム情報を定義することができます。ただし、複数の行に同じVSNが指定されている場合は、最初の行が選択されます。

(注1) JP1/HiCommand Dynamic Link Managerを使用する場合は、①の書式が異なります。JP1/HiCommand Dynamic Link Managerのソフトウェア添付資料を参照して定義してください。

② メインフレーム側で設定したボリュームシリアル番号を、6桁以下の英文字及び英数字で指定します。

③ 共有ボリュームのデバイスエミュレーションタイプを指定します。

・デバイスエミュレーションタイプの種類については1.2.2節を参照して下さい。

④ MFAの指定があると、XSPとして動作します。なければXSP以外として動作します。

VSN識別子を使用する場合、XSP以外でも本パラメータは必須となります(MFNと指定して下さい)。

⑤ VSN識別子を35文字以内で指定します。同一VSNボリュームに複数アクセスする場合に指定します。省略可能です。

⑥ 「Return」キーを押下します。

⑦ end : ボリューム制御情報の終了を示します。

「end」と入力して下さい。

(注2) 各パラメータの間は、1個以上の「空白」で区切ります。

上記の例の中で、「_」が付いている箇所は実行環境に合わせた文字/文字列を入力します。「_」が付いていない箇所は例の通りに入力します。

(注3) 先頭の文字が「#」である行はコメント行とみなします。

(4) NCR UNIXの場合

①	②	③	④	⑤	⑥
/dev/rdsk/cu1tu2du3s0	aaaaaa	3390-3A	MFN	MVS01	改行
/dev/rdsk/cv1tv2dv3s0	aaaaaa	3390-3B	MFN	MSP01	改行
/dev/rdsk/cw1tw2dw3s0	cccccc	3390-3C			改行
/dev/rdsk/cx1tx2dx3s0	dddddd	3380-KA	MFN		改行
/dev/rdsk/cy1ty2dy3s0	eeeeee	3380-KB			改行
/dev/rdsk/cz1tz2dz3s0	ffffff	3380-KC			改行
end					
⑦					

(注) 上記の記述例はメインフレームのOSがMVS/VSEの場合です。

③で指定するデバイスエミュレーションタイプは「1.2.2 サポート対象」のデバイスエミュレーションタイプを参照して下さい。

① オープンシステム上で定義したパーティション名を指定します。

FCU (FAL)はメインフレームのデータセットに対してraw I/Oでアクセスするため、rawデバイスとして定義する必要があります。スライス番号は「0」を使用します。ボリューム定義ファイルには複数のボリューム情報を定義することができます。ただし、複数の行に同じVSNが指定されている場合は、最初の行が選択されます。

② メインフレーム側で設定したボリュームシリアル番号を、6桁以下の英文字及び英数字で指定します。

③ 共有ボリュームのデバイスエミュレーションタイプを指定します。
・デバイスエミュレーションタイプの種類については1.2.2節を参照して下さい。

④ VSN識別子を使用する場合、XSP以外でも本パラメータは必須となります(MFNと指定して下さい)。
(MFAの指定があるとXSPとして動作します)

⑤ VSN識別子を35文字以内で指定します。同一VSNボリュームに複数アクセスする場合に指定します。
省略可能です。

⑥ 「Return」キーを押下します。

⑦ end : ボリューム制御情報の終了を示します。

「end」と入力して下さい。

(注1) 各パラメータの間は、1個以上の「空白」で区切ります。
上記の例の中で、「_」が付いている箇所は実行環境に合わせた文字/文字列を入力します。「_」が付いていない箇所は例の通りに入力します。

(注2) 先頭の文字が「#」である行はコメント行とみなします。

(5) HI-UX/WE2の場合

①	②	③	④	⑤	⑥
/dev/array/rhdu <u>1u2u3u4u</u> 480	aaaaaa	<u>3390-3A</u>	MFN	MVS01	改行
/dev/array/rhdv <u>1v2v3v4v</u> 480	aaaaaa	<u>3390-3B</u>	MFN	MSP01	改行
/dev/array/rhdw <u>1w2w3w4w</u> 480	ccccc	<u>3390-3C</u>			改行
/dev/array/rhdx <u>1x2x3x4x</u> 480	dddddd	<u>3380-KA</u>	MFN		改行
/dev/array/rhdy <u>1y2y3y4y</u> 480	eeeeee	<u>3380-KB</u>			改行
/dev/array/rhdz <u>1z2z3z4z</u> 480	ffffff	<u>3380-KC</u>			改行
end					
⑦					

(注) 上記の記述例はメインフレームのOSがMVS/VSEの場合です。

③で指定するデバイスエミュレーションタイプは「1.2.2 サポート対象」のデバイスエミュレーションタイプを参照して下さい。

- ① オープンシステム上で定義したデバイススペシャルファイル名を指定します。
 ただし、拡張LUN(arrayinitを-xオプション指定時)を使用する場合デバイススペシャル名は以下のフォーマットとなります。 /dev/array/rhdxS1S2L1L280
 u1/v1/w1/x1/y1/z1/S1 : BAのストロブ番号(0x0~0x7)
 u2/v2/w2/x2/y2/z2/S2 : SCSIアダプタの論理ストロブ番号(0x0~0xf)
 u3/v3/w3/x3/y3/z3 : SCSI ID(0x0~0x6及び0x8~0xf)
 u4/v4/w4/x4/y4/z4 : LUN(0x0~0x7)
 L1L2: 16進の2桁のLUN(0x00~0x7f)
 80 : パーティション番号(FAL用)

FCU (FAL)はメインフレームのデータセットに対してraw I/Oでアクセスするため、rawデバイスとして定義する必要があります。
 複数のボリューム情報を定義することができます。
 ただし、複数の行に同じVSNが指定されている場合は、最初の行が選択されます。

- ② メインフレーム側で設定したボリュームシリアルナンバを、6桁以下の英文字及び英数字で指定します。
- ③ 共有ボリュームのデバイスエミュレーションタイプを指定します。
 ・デバイスエミュレーションタイプの種類については1.2.2節を参照して下さい。
- ④ VSN識別子を使用する場合、XSP以外でも本パラメータは必須となります(MFNと指定して下さい)。(MFAの指定があるとXSPとして動作します)
- ⑤ VSN識別子を35文字以内で指定します。同一VSNボリュームに複数アクセスする場合に指定します。省略可能です。
- ⑥ 「Return」キーを押下します。
- ⑦ end : ボリューム制御情報の終了を示します。
 「end」と入力して下さい。

(注1) 各パラメータの間は、1個以上の「空白」で区切ります。
 上記の例の中で、「_」が付いている箇所は実行環境に合わせた文字/文字列を入力します。「_」が付いていない箇所は例の通りに入力します。

(注2) 先頭の文字が「#」である行はコメント行とみなします。

(6) Digital UNIX / Tru64 UNIXの場合

①	②	③	④	⑤	⑥
/dev/rrzX1Y1c	aaaaaa	3390-3A	MFN	MVS01	改行
/dev/rrzX2Y2c	aaaaaa	3390-3B	MFN	MSP01	改行
/dev/rrzX3Y3c	ccccc	3390-3C			改行
/dev/rrzX4Y4c	dddddd	3380-KA	MFN		改行
/dev/rrzX5Y5c	eeeeee	3380-KB			改行
/dev/rrzX6Y6c	ffffff	3380-KC			改行
end					
⑦					

(注) 上記の記述例はメインフレームのOSがMVS/VSEの場合です。

③で指定するデバイスエミュレーションタイプは「1.2.2 サポート対象」のデバイスエミュレーションタイプを参照して下さい。

- ① オペンシステム上で定義したデバイス名を指定します。上記表示はTru64 UNIX 4.0F以前のもので、XnYm(n,m=1~6)の値は、file /dev/rrz* コマンドにて取得して設定してください。Tru64 UNIX V5.0は、/dev/rdisk/dskX1c という表示形式となります。X1の値は、file /dev/rdisk/dsk*コマンドにて取得して設定して下さい。FCU (FAL)はメインフレームのデータセットに対してraw I/Oでアクセスするため、rawデバイスとして定義する必要があります。ボリューム定義ファイルには複数のボリューム情報を定義することができます。ただし、複数の行に同じVSNが指定されている場合は、最初の行が選択されます。
- ② メインフレーム側で設定したボリュームシリアル番号を、6桁以下の英文字及び英数字で指定します。
- ③ 共用ボリュームのデバイスエミュレーションタイプを指定します。
・デバイスエミュレーションタイプの種類については1.2.2節を参照して下さい。
- ④ VSN識別子を使用する場合、XSP以外でも本パラメータは必須となります(MFNと指定して下さい)。(MFAの指定があるとXSPとして動作します)
- ⑤ VSN識別子を35文字以内で指定します。同一VSNボリュームに複数アクセスする場合に指定します。省略可能です。
- ⑥ 「Return」キーを押下します。
- ⑦ end : ボリューム制御情報の終了を示します。

「end」と入力して下さい。

(注1) 各パラメータの間は、1個以上の「空白」で区切ります。
上記の例の中で、「_」が付いている箇所は実行環境に合わせた文字/文字列を入力します。「_」が付いていない箇所は例の通りに入力します。

(注2) 先頭の文字が「#」である行はコメント行とみなします。

(7) Linuxの場合

①	②	③	④	⑤	⑥
/dev/raw/raw1	<u>aaaaaa</u>	<u>3390-3A</u>	MFN	MVS01	改行
/dev/raw/raw2	<u>aaaaaa</u>	<u>3390-3B</u>	MFN	MSP01	改行
/dev/raw/raw3	<u>cccccc</u>	<u>3390-3C</u>			改行
/dev/raw/raw4	<u>dddddd</u>	<u>3380-KA</u>	MFA		改行
/dev/raw/raw5	<u>eeeeee</u>	<u>3380-KB</u>			改行
/dev/raw/raw6	<u>ffffff</u>	<u>3380-KC</u>			改行
end					
⑦					

(注) 上記の記述例はメインフレームのOSがMVS/VSEの場合です。

③で指定するデバイスエミュレーションタイプは「1.2.2 サポート対象」のデバイスエミュレーションタイプを参照して下さい。

① オープンシステム上で定義したパーティション名を指定します。

FCU (FAL)はメインフレームのデータセットに対してraw I/Oでアクセスするため、rawデバイスとして定義する必要があります。ボリューム定義ファイルには複数のボリューム情報を定義することができます。ただし、複数の行に同じVSNが指定されている場合は、最初の行が選択されます。

(注1) rawデバイスの設定には、rawコマンドが必要です。rawコマンドによりLinux rawキャラクタデバイスをブロックデバイスにバイトして下さい。rawコマンドについての詳細はオンラインマニュアルを参照して下さい。

② メインフレーム側で設定したボリュームシリアル番号を、6桁以下の英文字及び英数字で指定します。

③ 共有ボリュームのデバイスエミュレーションタイプを指定します。

・デバイスエミュレーションタイプの種類については1.2.2節を参照して下さい。

④ MFAの指定があると、XSPとして動作します。なければXSP以外として動作します。

VSN識別子を使用する場合、XSP以外でも本パラメータは必須となります(MFNと指定して下さい)。

⑤ VSN識別子を35文字以内で指定します。同一VSNボリュームに複数アクセスする場合に指定します。省略可能です。

⑥ 「Return」キーを押下します。

⑦ end : ボリューム制御情報の終了を示します。

「end」と入力して下さい。

(注2) 各パラメータの間は、1個以上の「空白」で区切ります。

上記の例の中で、「_」が付いている箇所は実行環境に合わせた文字/文字列を入力します。「_」が付いていない箇所は例の通りに入力します。

(注3) 先頭の文字が「#」である行はコメント行とみなします。

(8) Windows NT/2000Server/Server2003/Server2008/Server2008 R2/Server2012/Server2012 R2/Server2016/Server2019の場合

①	②	③	④	⑤	⑥
¥¥. ¥PHYSICALDRIVE <u>1</u>	<u>aaaaaa</u>	<u>3390-3A</u>	MFN	MVS01	復帰改行
¥¥. ¥PHYSICALDRIVE <u>2</u>	<u>aaaaaa</u>	<u>3390-3B</u>	MFN	MSP01	復帰改行
¥¥. ¥PHYSICALDRIVE <u>3</u>	<u>cccccc</u>	<u>3390-3C</u>			復帰改行
¥¥. ¥PHYSICALDRIVE <u>4</u>	<u>dddddd</u>	<u>3380-KA</u>	MFA		復帰改行
¥¥. ¥PHYSICALDRIVE <u>5</u>	<u>eeeeee</u>	<u>3380-KB</u>			復帰改行
¥¥. ¥PHYSICALDRIVE <u>6</u>	<u>ffffff</u>	<u>3380-KC</u>			復帰改行
end					
⑦					

(注) 上記の記述例はメインフレームのOSがMVS/VSEの場合です。

③で指定するデバイスエミュレーションタイプは「1.2.2 非ポート対象」のデバイスエミュレーションタイプを参照して下さい。

① 物理ドライブ番号を指定します。

② ボリュームシリアル番号を、6桁以下の英数字で指定します。

③ デバイスエミュレーションタイプを指定します。

・デバイスエミュレーションタイプの種類については1.2.2節を参照して下さい。

④ MFAの指定があると、XSPとして動作します。なければXSP以外として動作します。
VSN識別子を使用する場合、XSP以外でも本パラメータは必須となります(MFNと指定して下さい)。

⑤ VSN識別子を35文字以内で指定します。同一VSNボリュームに複数アクセスする場合に指定します。
省略可能です。

⑥ 「Enter」キーを押下します。

⑦ end : ボリューム定義の終了を指示します。
省略可能です。

(注2) 各パラメータの間は、1個以上の「空白」で区切ります。

上記の例の中で、「_」が付いている箇所は実行環境に合わせた文字/文字列を入力します。「_」が付いていない箇所は例の通りに入力します。

(注3) 先頭の文字が「#」である行はコメント行とみなします。

8.2.2 オープンシステム—オープンシステム間のデータ交換の場合

中間ボリュームのボリュームシリアル番号とオープンシステム上でのボリューム(パーティション)とを対応付けるための情報を以下の形式で格納したファイルを作成します。

ボリューム定義ファイルのファイル名は「datasetmount.dat」です。FCU実行時のカレントディレクトリに作成して下さい。

①	②	③	④	⑤	⑥
デバイス名	ボリュームシリアル番号	OPEN-3 MFN	N01		改行
:	:	OPEN-8 MFN	N02		改行
:	:	OPEN-9			改行
:	:	OPEN-K			改行
end					
⑦					

① オープンシステム上で定義した中間ボリュームのrawデバイス名を指定します。

- ・AIXの場合 : /dev/rhdiskN
- ・HP-UXの場合 : /dev/rdisk/cN1tN2dN3
- ・Solarisの場合 : /dev/rdisk/cN1tN2dN3sN4
- ・Linuxの場合 : /dev/raw/rawN
- ・WindowsNT/2000Server/Server2003/Server2008/Server2008 R2/Server2012/Server2012 R2/Server2016/Server2019の場合 : ¥¥. ¥PHYSICALDRIVE_n (全大文字)

FCU (FAL) は中間ボリュームのファイルに対してraw I/Oでアクセスするため、rawデバイスとして定義する必要があります。

複数のボリューム情報を定義することができます。

ただし、複数の行に同じVSNが指定されている場合は、最初の行が選択されます。(ただし、フォーマットはデバイス名とVSNの両方を指定する為、指定されたデバイス名が一致する行の処理をします。)

(注1) HP-UXの場合はLegacy DSF以外のデバイス・ファイル表記形式には対応していません。

② フォーマットで設定したボリュームシリアル番号を、6桁以下の英字(A-Z、@、#、¥(\))、数字(0-9)で指定します。

③ 中間ボリュームのデバイスエミュレーションタイプを指定します。(全大文字)

「OPEN-3」/「OPEN-8」/「OPEN-9」/「OPEN-K」/「OPEN-E」/「OPEN-L」/「OPEN-M」/「OPEN-V」と指定します。

④ VSN識別子を指定する場合、MFNを指定して下さい。

⑤ VSN識別子を35文字以内で指定します。同一VSNボリュームに複数アクセスする場合に指定します。省略可能です。(フォーマット/アケータでは、VSN識別子を意識した処理は行えません。)

⑥ 「Return」キーを押下します。

⑦ end : ボリューム制御情報の終了を示します。

「end」と入力して下さい。

(注1) 各パラメータの間は、1個以上の「空白」で区切ります。

上記の例の中で、「_」が付いている箇所は実行環境に合わせた文字/文字列を入力します。「_」が付いていない箇所は例の通りに入力します。

(注2) 先頭の文字が「#」である行はコメント行とみなします。

8.3 マルチボリューム定義ファイル

FAL/FCU では、マルチボリューム定義ファイルを作成することにより、マルチボリュームにまたがったデータセットの MTO/OTM が可能となります。動作可能なバージョンについては下記を参照して下さい。

(詳細は §4.8 を参照して下さい)

FAL/FCU のバージョン	マルチボリュームデータセットの転送
01-XX-50 以降	MTO 転送が可能
01-XX-60 以降 (XX ≥ 03)	MTO / OTM 転送が可能

(1) マルチボリューム定義ファイルの格納場所

FCU 実行時のカレントディレクトリに **multidef.dat** という名称で作成します。その名称を変更したい場合は、環境変数 **FAL_MULTI_DEF_FILE** に定義して下さい。

(ディレクトリパス等を含まないファイル名のみ定義して下さい)

(2) フォーマット

VSN:DSN[, VOLID 1] VSN[, VOLID 2] VSN[, VOLID 3] … VSN[, VOLID n]	改行			
①	②	③	④	⑤
end				
⑥				

① ボリュームシリアル番号:データセット名 [, VSN 識別子]

先頭ボリュームのボリュームシリアル番号、データセット名、VSN 識別子(省略可)を定義して下さい。

② ボリュームシリアル番号 [, VSN 識別子]

2 番目のボリュームシリアル番号、VSN 識別子(省略可)を定義する

③ ボリュームシリアル番号 [, VSN 識別子]

3 番目のボリュームシリアル番号、VSN 識別子(省略可)を定義する

④ ボリュームシリアル番号 [, VSN 識別子]

n 番目のボリュームシリアル番号、VSN 識別子(省略可)を定義する (n:MAX 30(*1))

*1 01-XX-66 以降の場合には n の最大が 61 ボリュームとなります。

⑤ 改行コードを定義する (UNIX : LF, NT : CRLF)

⑥ マルチボリューム定義ファイルの終了を示します。

- 各ボリュームシリアル番号の間には空白を定義して下さい。
- マルチボリューム情報は複数行指定可能 (MAX:999 レコード) です。
- 各項目ごとの指定可能文字数は、下記通りです。
 - ボリュームシリアル番号 : 1-6 文字, データセット名 : 1-44 文字, VSN 識別子 : 1-35 文字
 - 1 行の文字数は 1399 バイト (01-XX-66 以降は 2699 バイト) 以下 (改行を含まない) です。
- 先頭ボリュームの VSN 識別子の指定は、ボリューム定義ファイル上と同じにしてください。同じでないとは同一とみなされず、シングルボリュームとして処理します。
- 新規に作成したマルチボリュームデータセットは、必ず M/F 側から一度データの書き込みを行って VTOC の DSCB1 にあるデータセットインジケータの最終ボリュームフラグを確定させてから転送に使用して下さい。

[指定例]

```
VSN001:FAL. DATA. F01 VSN002 VSN003
VSN005:FAL. DATA. M03, MVS01 VSN006 VSN007 VSN008, MVS01
End
```

8.4 パラメタ定義ファイルの作成

パラメタ定義ファイルは **FCU** を起動する際の詳細パラメタを格納しておくファイルです。
 パラメタ定義ファイルはエディタ等であらかじめ作成しておくか、**FCU** を起動して作成します。
 パラメタ定義ファイルのデフォルトのファイル名は以下です。

UNIX系OSの場合 「fcudata.param」
 Windowsの場合 「fcudata.prm」 (-nc 指定時)

なお、常に操作画面からパラメタを入力する場合には、パラメタ定義ファイルは不要です。

- UNIX系OSの場合、1個のパラメタ定義ファイル中に999件までのパラメタセットを定義できます。
 (999件を越えるパラメタセットを定義した場合、1000件目の処理でエラーとなります)
 先頭文字が「#」の行はコメント行として扱います。(注5)
- Windowsの場合、1000件を越えて定義してエラーとなる件数はメモリ容量に依存します。
 先頭文字が「#」の行はコメント行として扱います。(注6)
- データ転送の処理はパラメタセットの配列順に行います。

①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑪
mto	VSN:データセット名, VSN識別子	オープンシステムのファイル名	CC	PAD	DEL	Emp=Yes	RDW=Yes	PIPE=Yes	VSE=RF, RL, BL	改行
mto	VSN:データセット名	オープンシステムのファイル名	CC	PAD	DEL	Emp=Yes	RDW=Yes	PIPE=Yes	VSE=RF, RL, BL	改行
otm	オープンシステムのファイル名	VSN:データセット名, VSN識別子	CC	PAD	DEL	Emp=Yes			VSE=RF, RL, BL	改行
otm	オープンシステムのファイル名	VSN:データセット名	CC	PAD	DEL	Emp=Yes			VSE=RF, RL, BL	改行
end										
⑫										

(注)HMDE/KANJI連携時は⑩と⑪の間に「USER-EDIT フィールド 定義ファイル名, 編集オプションファイル名」を指定して下さい。詳細はHMDE/KANJIの「ユーザズガイド」を参照願います。

- パラメタ一行全体が改行を含めず最大3199文字以内となります。

①データ転送の方向を指定します。

mto : メインフレームからオープンシステムへのデータ転送であることを示します。
 「mto」と入力します。

otm : オープンシステムからメインフレームへのデータ転送であることを示します。
 「otm」と入力します。

②入力となるメインフレームのデータセット名またはオープンシステムのファイル名を指定します。

メインフレームのデータセット名の場合

- VSN(ボリュームシリアルナンバ)とデータセット名を「コロン」で区切って指定します。さらに必要であれば、VSN識別子を「,」で区切って指定します。
- VSNは6桁以下の英文字及び英数字で指定します。
- マルチボリューム時は先頭ボリュームのVSNを指定して下さい。
- データセット名は44桁以下の英文字及び英数字で指定します。
- データセット名の中に「空白」を含むことはできません。
- VSN識別子は省略可能で、指定する場合は35文字以内の英数字または英記号で指定します。
 パラメタ定義ファイル上指定する場合、ボリューム定義ファイル上にもVSN識別子の指定が必要になります。

オープンシステムのファイル名の場合

- ・以下のいずれかの形式で指定します。
- ・指定名称として最大1024文字以内で定義できます。

[パスありの場合]

 /ディレクトリ名...../ディレクトリ名/ファイル名 (UnIX)
 ドライブ:¥ディレクトリ名.....¥ディレクトリ名¥ファイル名 (Windows)

[パスなしの場合]

 ファイル名

 ファイル名だけが指定された場合は、カレントディレクトリのファイルが対象となります。

- ・オープンシステムファイル名の中に「空白」を含むことはできません。

③出力となるメインフレームのデータセット名またはオープンシステムのファイル名を指定します。

- ・データセット名、ファイル名の指定形式は入力ファイル名の指定形式と同様です。

④CC : データ実体のコード変換種別を指定します。

- ・FCUが持っているコード変換テーブルを使用してコード変換を行う場合

EA : メインフレーム : EBCDICコード ⇔ オープンシステム : ASCIIコード の変換をします。

EcA : メインフレーム : EBCDICコード ⇔ オープンシステム : ASCIIコード の変換をします。

EkJ : メインフレーム : EBCDICコード ⇔ オープンシステム : JIS8コード の変換をします。

(メインフレームのOSがMVS/VSE/MSP/XSPの時には、EkJの指定を行わないでください。)

No : コード変換は行いません。

- ・ユーザがコード変換テーブルを指定してコード変換を行う場合、コード変換テーブルを格納したファイルの名称を指定します。

- ・コード変換テーブルのサイズは256バイトです。

- ・コード変換は1バイトずつ実行します。

- ・EA/EcA/EkJ/Noはファイル名として使用できません。文字列:EA/EcA/EkJ/Noが指定された場合は、FCUが持っているコード変換テーブルによる変換とみなします。

- ・指定されたファイルが存在しない場合はエラーとなります。

(注) オープンシステム—オープンシステム間のデータ交換時はコード変換をサポートしていません。

コード変換種別は、「No」を指定して下さい。

コード変換、ファイル名を指定した場合、エラーとなります。

HMDE/KANJI連携時もFCUのコード変換をサポートしていませんので、「No」を指定して下さい。

No以外を指定すると、UNIX版ではエラーとなり、NT版ではNoとして処理を行います。

⑤PAD : パディングの可否を指定します。

Yes : パディングします。

(注) 「Yes」を指定すると、「RDW=Yes」は指定出来ません。

No : パディングしません。

(注) HMDE/KANJI連携時はFCUでのパディングをサポートしていませんので、「No」を指定して下さい。Yesを指定するとエラーとなります。

⑥DEL：mtoの場合、デリミタを付加するか否かを指定します。otmの場合、デリミタが付加されているか否かを指定します。

・データ転送方向がmtoのとき

・UNIX系OSの場合

CR：CR(キャリッジリターン)を各データの後に付加します。

LF：LF(ラインフィード)を各データの後に付加します。

No：デリミタは付加しません。データは大きな一つのまとまりになって見えます。

・Windowsの場合

CRLF：CR+LF(復帰改行)を各データの後に付加します。

No：デリミタは付加しません。

・データ転送方向がotmのとき

・UNIX系OSの場合

CR：CRまでを1データとして切り出します。

LF：LFまでを1データとして切り出します。

CRLF：CR+LFまでを1データとして切り出します。

本指定はコマンドライン実行時のみ可能です。

No：データセットのレコード長に従ってデータを切り出します。

・Windowsの場合

CRLF：CR+LFまでを1データとして切り出します。

No：データセットのレコード長に従ってデータを切り出します。

(注) データセットのレコード形式が固定長で、No以外を指定した場合は、パディングの有無によって転送時のデータ編集形式(データを切り出す方法)が異なります。

詳しくは7.5節を参照して下さい。HMDE/KANJI連携時は本パラメタでのデリミタをサポートしていませんので、「No」を指定して下さい。No以外を指定するとUNIX版ではエラーとなり、NT版ではNoとして処理を行います。

(注) UNIX系OSでmto転送時にデリミタ指定をCRLFとした場合はエラー(-109)となります。

(注) UNIX系OSでGUI機能使用時にデリミタ指定をCRLFとした場合はエラー(-109)となります。

(注) UNIX系OSでHMDE/KANJI連携時にHMDE/KANJIの定義としてデリミタ指定をCRLFとした場合はHMDE/KANJIでデリミタとしてCRLFをサポートしていませんのでエラーとなります。

⑦データなしファイルも転送するか否かを指定します。データなしファイルとは以下のファイルのことです。

・転送元がメインフレームのデータセットの場合：データレコードもEOFレコードも書かれていないデータセット

・転送元がオープンシステムの場合：ファイルサイズ=0のファイル

本パラメタは省略可能です。省略時は「Emp=No」の解釈となります。

・Emp=Yes：転送元がデータなしファイルでもデータ転送の処理を行います。

転送先がオープンシステムの場合、ファイルサイズ=0のファイルが作成されます。

転送先がメインフレームの場合、EOFレコードのみのデータセットが作成されます。

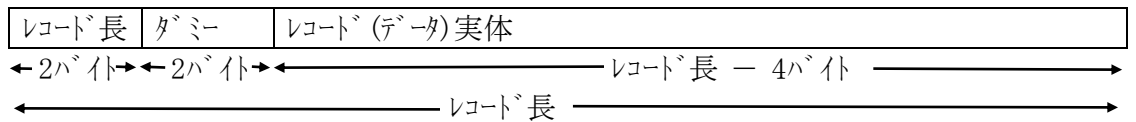
・Emp=No：転送元がデータなしファイルの場合、データ転送は実行せず、エラーとします。

ただし、メインフレームOSがVSE/VOS1の場合、0byteレコードをMTO時エラーとなりますが、0byteファイルを作成してしまいます。

⑧データ転送方向がmtoでメインフレームのデータセットのレコード形式が可変長のとき、レコード長を付加した形でオープンシステムのファイルを作成するか否かを指定します。

なお、本パラメータは省略可能です。省略時は「RDW=No」の解釈となります。

- RDW=Yes : ・次図のように各レコードの先頭にレコード長(2進数)を付加します。
 - ・本指定のときコード変換、パディング、デリミタの指定は「No」を指定しないとエラーになります。
 - ・otmの場合本指定はエラーとなります。
 - ・レコード形式が固定長の場合、本指定でもRDWは付加されません。



- RDW=No : レコード長は付加しません。データ実体だけをファイル出力します。

(注) RDW=Yesを指定してメインフレームからオープンシステムに転送したデータセットをメインフレームに戻すと元のデータセットとは互換性のないデータセットが作成されます。

HMDE/KANJI連携時はレコード長付加をサポートしていませんので、「RDW=Yes」は指定しないで下さい。「RDW=Yes」を指定すると、UNIX版ではエラーとなりますが、NT版ではNoとして処理を行います。

⑨データ転送方向がmtoのとき、ファイル出力先をオープンシステム専用ボリュームではなく、パイプ (Named pipe) を使用するか否かを指定します。なお、本パラメータは省略可能です。省略時は「PIPE=No」の解釈となります。

- PIPE=Yes : ファイル出力をパイプに指定します。otmの場合、本指定はエラーとなります。(Named pipe)をファイル名で指定した場合は元のファイル内容を消去し(Named pipe)として使用します。
- PIPE=No : パイプを指定しません。出力先は通常のファイルとなります。

⑩レコード形式・レコード長・ブロック長が定義されていないデータセットを使用する際に、レコード形式・レコード長・ブロック長を指定します。(VTOC上にレコード形式/レコード長/ブロック長のいずれかが設定されていない場合、レコード形式/レコード長/ブロック長の全てを設定して下さい。)

- ・VSE= : レコード形式・レコード長・ブロック長を指定することを意味します。
レコード形式・レコード長・ブロック長を指定しない場合、「VSE=RF, RL, BL」を省略します。
- ・RF : レコード形式を以下の文字列で指定します。
 - F : 固定長非ブロック化
 - FB : 固定長ブロック化
 - V : 可変長非ブロック化
 - VB : 可変長ブロック化上記以外の文字列を指定するとエラーとなります。
- ・RL : レコード長を10進数(ハレド数)で指定します。
指定可能な値は以下の範囲です。
 - ・レコード形式=Fのとき : レコード長 = ブロック長
 - ・レコード形式=FBのとき : レコード長 = (ブロック長) ÷ n n:整数
 - ・レコード形式=V/VBのとき : $5 \leq \text{レコード長} \leq (\text{ブロック長} - 4)$上記の範囲外の値を指定するとエラーとなります。
- ・BL : ブロック長を10進数(ハレド数)で指定します。
指定可能な値は以下の範囲です。
 - ・レコード形式=F/FBのとき : $1 \leq \text{ブロック長} \leq 32760$
 - ・レコード形式=V/VBのとき : $9 \leq \text{ブロック長} \leq 32760$上記の範囲外の値を指定するとエラーとなります。
- ・RF, RL, BLの間は1個の「,」で区切ります。違反するとエラーとなります。
- ・レコード形式・レコード長・ブロック長が定義されているデータセットに本パラメタが指定された場合、定義されている情報とパラメタとの一致をチェックし、不一致であればエラーとします。
一致している場合はデータ転送等の処理を実行します。

(注) オープンシステムーオープンシステム間のデータ交換では、使用しません。(指定した場合のFAL/FCUの扱いは、メインフレームーオープンシステム間のデータ交換の場合と同様です)

VSE2.4 を FAL/FCUバージョン(01-03-59)によりデータ転送を行なう場合、VSEパラメタの指定値により転送が無効になるケースがあります。また、VSEパラメタを指定しない場合でもVTOC上のデータのRL、BLの値により転送が無効になるケースもあります。以下に転送不可時のVSEパラメタの指定条件及びVTOC上のデータのRL、BLの値の条件を示します。

- ・レコード形式がV(可変長)でVSEパラメタを指定する場合

正常なデータセット属性($5 \leq RL \leq BL - 4$ が成り立つデータセット属性)では転送可能ですが、以下に示すVSEパラメタの指定条件に一致しない場合は転送が無効になります。

- ・ $RL \leq 32756$
- ・ $BL \leq 32760$
- ・ $BL = RL + 4$

また、VTOC上データのRL及びBLの値が $RL > 32756$ 、 $BL > 32760$ の場合は、上記の条件に一致する様にVSEパラメタを指定しなければ、転送が無効になります。更にまた、VTOC上データのRL及びBLの値が $RL \leq 32756$ 、 $BL \leq 32760$ の場合で、 $RL = BL$ の場合は、

- ・ $RL(\text{VSEパラメタにする値}) = RL(\text{VTOC上のデータの値}) + 4 \leq 32756$
 - ・ $BL(\text{VSEパラメタにする値}) = BL(\text{VTOC上のデータの値}) + 8 \leq 32760$
- 上記の指定条件でVSEパラメタを指定しなければ、転送が無効になります。

- ・レコード形式がV(可変長)でVSEパラメタを指定しない場合

正常なデータセット属性($5 \leq RL \leq BL - 4$ が成り立つデータセット属性)では転送可能ですが、VTOC上のデータのRL、BLの値が以下に示す条件に一致しない場合は転送が無効になります。

- ・ $BL = RL + 4 \leq 32760$

また、VTOC上データのRL及びBLの値が $RL > 32756$ 、 $BL > 32760$ の場合は、VSEパラメタの値として $RL = 32756$ 、 $BL = 32760$ が指定され転送が行なわれます。更にまた、VTOC上データのRL及びBLの値が $RL = BL$ の場合は、

- ・ $RL(\text{VSEパラメタとして指定される値}) = RL(\text{VTOC上のデータの値}) + 4 \leq 32756$
- ・ $BL(\text{VSEパラメタとして指定される値}) = BL(\text{VTOC上のデータの値}) + 8 \leq 32760$

上記の条件でVSEパラメタの値としてRL、BLそれぞれが指定され転送が行なわれます。

- ・レコード形式がVB(可変長ブロック化)でVSEパラメタを指定する場合

正常なデータセット属性($RL = BL$ が成り立つデータセット属性)では転送可能ですが、以下に示すVSEパラメタの指定条件に一致しない場合は転送が無効になります。

- ・ $RL(\text{VSEパラメタにする値}) = RL(\text{VTOC上のデータの値}) + 4 \leq 32756$
- ・ $BL(\text{VSEパラメタにする値}) = BL(\text{VTOC上のデータの値}) + 8 \leq 32760$

- ・レコード形式がVB(可変長ブロック化)でVSEパラメタを指定しない場合

上のデータのRL、BLの値が以下に示す条件に一致しない場合は転送が無効になります。VTOC上のデータのRL、BLの値が以下に示す条件に一致しない場合は転送が無効になります。

- ・ $RL = BL > 32752$

また、VTOC上データのRL及びBLの値が $RL = BL \leq 32752$ の場合は、VSEパラメタの値として $RL(\text{VTOC上のデータの値}) + 4$ 、 $BL(\text{VTOC上のデータの値}) + 8$ が指定され転送が行なわれます。

⑪改行を指示します。

- ・UNIX系OSの場合、「Return」キーです。
- ・Windowsの場合、「Enter」キーです。

⑫end : パラメタの終了を示します。「end」と入力します。
省略可。

(注1) 各パラメタの間は、1個以上の「空白」で区切ります。

(注2) パラメタ定義ファイルの先頭から「end」行までの間に空白行を含むことはできません。
空白行があるとエラーとなります。ただし、起動パラメタとして「-nw」が指定されている場合は、エラーとせずに空白行をスキップします。

(注3) Empパラメタ、RDWパラメタ、PIPEパラメタ及びVSEパラメタはどれを先に指定してもかまいません。
ただし、Emp/RDW/PIPE/VSE以外のパラメタより先に指定することはできません。
また、Emp/RDW/PIPE/VSE以外のパラメタは記述例の順番で指定します。

(注4) Windowsのバージョン 01-01-34より古いバージョンを既にお使い頂いている場合、
01-01-34以降の新しいバージョンにアップする際、次の点に特にご注意ください。
(1) デフォルトのパラメタ定義ファイルの名称が「fcudata.param」→「fcudata.prm」と
新しくなっています。今お使いのパラメタ定義ファイルの名称を新しい名称に合
わせて変更 (リネーム) してください。
(2) 操作画面の表示が必須となり、起動パラメタで「-nw」の指定が無くなりました。

(注5) UNIX版ではコメント行を以下のように扱います。

(1) 操作画面を使用しない場合

- ・コメント行はスキップします。
- ・「パラメタ定義ファイルの最大行数=999行」のチェックの対象から除外します。

(2) 操作画面を使用する場合

- ・コメント行をLoadした場合コメント行の内容は表示せず、「Parameter file : Comment line」をステータス欄に表示します。
- ・画面操作によるコメント行の作成はできません。
コメント行の削除及びコメント行を有効なパラメタ行に変更することはできます。
- ・コメント行も「パラメタ定義ファイルの最大行数=999行」のチェックの対象とします。

(注6) Windows版ではコメント行を以下のように扱います。

- ・コメント行は処理の対象とせずにスキップします。
- ・コメント行は行数のカウントに含めません。
- ・FCUの画面操作によるコメント行の作成/削除/有効なパラメタ行への変更はできません。
エディタで処理して下さい。

(注7) 指定するパラメタとして、①～⑥は必須です。パラメタの数が5以下だとエラーとなります。
パラメタとして11個以上指定すると、それ以降のパラメタは無視されます。
UNIX版では7個目以降でEmp/RDW/PIPE/VSE以外のパラメタを指定すると無視されます。
Windows版では、Emp/RDW/PIPE/VSE以外のパラメタを指定するとエラーとなります。
(Windows版では、SQL Loader連携時以外はPIPEの指定は行わないで下さい。
ただし、FAL SQL LoaderはUSP VとVSP, VSP G1000, VSP 5000では未サポートです。)

8.5 起動方法

FAL/FCUのインストール、ボリューム定義ファイルの作成完了後に起動します。
(パラメタ定義ファイルはFCU起動後に作成することもできます)

8.5.1 UNIX系OSの場合

- (1) GUI実行環境がある場合(64bit版のFCUは、GUIを対応していません。)
コマンドラインから以下のようにコマンドを入力して下さい。

`fcu [-nw] [-nc] [param]`

- `fcu` : FCUのパラメタ名称です。
「fcu」と入力して下さい。
- `-nw` : 本ユーティリティの操作画面を使用しないで処理を行うときに指定します。
本指定があり、後述の「param」がパラメタの直接指定でない場合、パラメタ定義ファイル中の全行(最大999行)について連続的に処理します。
「-nw」と入力して下さい。
本パラメタを指定しない場合、図8.6-2に示す操作画面が表示されます。画面上でパラメタの入力・確認・訂正を行い、ファイル変換処理を実行し、結果を確認する方法です。
- `-nc` : パラメタの内容の確認が不要である場合に指定します。
本指定があるとFCUはパラメタの確認応答(「OK」)を待たずにパラメタ定義ファイル中の全行(最大999行)について連続的に処理し、実行終了後はFCUを終了します。
「-nc」と入力して下さい。
本パラメタが指定されていない場合、パラメタの表示後「OK」の応答があるまでファイル変換処理を実行しません。
- `param` : FCUの詳細パラメタを格納しているパラメタ定義ファイル名または詳細パラメタを直接指定します。
(パラメタ定義ファイル及び詳細パラメタについては8.4節を参照して下さい)
① パラメタ定義ファイルのファイル名を指定する場合
絶対パスまたは相対パスで指定します。
例 : ・絶対パス : /ディレクトリ名/ディレクトリ名/ファイル名
・相対パス : ファイル名 . /ファイル名
ディレクトリ名/ファイル名 . /ディレクトリ名/ファイル名
② 詳細パラメタを直接指定する場合(Windows版では本形式の指定はできません)
「-P」と入力した後にパラメタ定義ファイル中に記述する形式と同じ形式で指定します。
例 : # `fcu -nw -nc -P mto VSNAAB:DSNAME outfile No No No`
(注) 「-P」の後に指定できるものは詳細パラメタだけです。
「-nw」、「-nc」を指定する場合は「-P」の前に指定して下さい。
(注) Windows版で「-P」指定を行った場合の動作については9節を参照して下さい。
本パラメタが指定されていない場合、パラメタ定義ファイルのファイル名を「fcudata.param」とみなして詳細パラメタの読出しを行います。
パラメタ定義ファイルが存在せず、「-nw」、「-nc」が指定されている場合はFCUを終了します。

(注) `fcu`を「-nw」指定でバックグラウンドプロセスとして起動する場合は、必ず「-nc」指定を行って下さい。「-nc」指定を行わないでバックグラウンド起動すると、確認応答待ち状態のまま動作が止まる場合があります。

実行終了後、以下のリターン値を返します。(詳細は8.8節を参照して下さい)

- 0 : 正常終了
- 1 : エラー終了

(2) GUI実行環境がない場合

コマンドラインから以下のようにコマンドを入力して下さい。

- ・ 32bit版の場合： `fcunw [-nc] [param]`
- ・ 64bit版の場合： `fcunw64 [-nc] [param]`

- ・ `fcunw/fcunw64` : GUIなしで実行するFCUのプログラム名称です。
- ・ `-nc / param` : GUI実行環境がある場合と同じ扱いです。

(注) 本起動形態の場合、前述の「-nw」が指定されたときと同様の動作をします。
`fcunw`をバックグラウンドプロセスとして起動する場合は、必ず「-nc」を指定して下さい。

8.5.2 Windowsの場合

アドミニストレータ権限のあるユーザでログインします。

(1) GUI画面を表示する場合

以下の何れかの方法で起動します。

- ① 「スタート」-「プログラム」-「FCU」に登録された **FCU** アイコンをクリックする。
- ② **C:\FCU** に移動し、**FCU** アイコンを(ダブル)クリックする。
(**C:\FCU** は、インストール時のデフォルトディレクトリです。インストール時に変更した場合は、任意に読み替えて下さい。)
- ③ **FCU** のショートカットを作成し、そのアイコンを(ダブル)クリックする。
- ④ コマンドプロンプト (DOS窓) を開き、下記起動オプションによるコマンド入力。

[起動オプション]

- ・ 32bit版の場合： `fcu [-nc] [-cl] [param]`
- ・ 64bit版の場合： `fcu64 [-nc] [-cl] [param]`

・ **-nc** : パラメタ定義ファイル中の全行について連続的に処理します。
実行終了後は、**FCU** を終了します。指定したパラメタ定義ファイルが存在しない時(指定なし時は、`fcudata.prm`)は、`fcu`を起動しません。

・ **-cl** : 実行前にログファイルをクリアします。

・ **param** : パラメタ定義ファイル名を指定します。
本パラメタが省略されている場合は、カレントに存在する「**fcudata.prm**」をパラメタ定義ファイルとみなして処理します。「**fcudata.prm**」が存在しない場合、`-nc`指定がないと、空の「**fcudata.prm**」を表示します。
指定したパラメタ定義ファイルが存在しない場合、`-nc`指定がないと空画面を表示します。

[リターン値]

- 0 : 正常終了
- 1 : エラー終了

(注) このリターン値は、「**-nc**」が指定された場合のみです。
パラメタ定義ファイルに指定したパラメタ行の1つの処理に、成功した処理があったとしても、エラーとなった処理が1つでもあればエラー終了が返されます。

(注) **FCU** のアイコンにパラメタ定義ファイルをドラッグ&ドロップすることでも、**FCU** は起動しますが、正常に動作しません。

(2) GUI画面を表示しない場合

以下の方法で起動します。

① コマンドプロンプト (DOS窓) を開き、下記起動オプションによるコマンド入力。

[起動オプション]

- 32bit版の場合: `fcunw [-cl] [param] ([-v])`
- 64bit版の場合: `fcunw64 [-cl] [param] ([-v])`

• **-cl** : 実行前にログファイルをクリアします。

• **param** : パラメタ定義ファイル名を指定します。

本パラメタが省略されている場合は、カレントに存在する「**fcudata.prm**」をパラメタ定義ファイルとみなして処理します。

• **-v** : FCUのバージョン情報を表示します。

本オプションは、他のオプションと一緒に使用できません。

[リターン値]

0 : 正常終了

-1 : エラー終了

8.6 画面操作

8.6.1 UNIX系OSの場合

FCUを起動すると以下のコピー画面と操作画面が表示されます。

コピー画面は数秒後に自動的に消えます。

操作画面を操作することによりパラメタの設定・変更、ファイル変換処理の開始を指示することができます。

FAL for AIX
File Conversion Utility
Ver. XX-XX-XX/XX
All rights reserved, Copyright (C) 1997, 2002, Hitachi, Ltd.

(注) タイトルは、HP-UXの場合「FAL for HP-UX」、Solarisの場合「FAL for Solaris」と表示されます。

図8.6-1 コピー画面

File Conversion Utility	
File	Help
Parameter File	<input type="text"/>
Volume File	<input type="text"/>
Direction	<input checked="" type="radio"/> M to 0 <input type="radio"/> 0 to M
Input File	<input type="text"/> <input type="button" value="OK"/>
Output File	<input type="text"/> <input type="checkbox"/> PIPE <input type="button" value="Cancel"/>
Code Conversion	<input type="radio"/> Ek<-> J <input type="radio"/> Ec/E<->A <input checked="" type="radio"/> No <input type="radio"/> FILE <input type="text"/>
Padding	<input type="radio"/> Yes <input checked="" type="radio"/> No
Delimiter	<input type="radio"/> CR <input type="radio"/> LF <input checked="" type="radio"/> No
Emp	<input type="radio"/> Yes <input checked="" type="radio"/> No
RDW	<input type="radio"/> Yes <input checked="" type="radio"/> No
VSE/VOS1	<input type="text"/>
Status	<input type="text"/>

図8.6-2 操作画面

(注 1) FCU で表示する画面中の文字は英字フォントを使用しているため、パラメタ定義ファイル、キーボード等から入力した時の文字と画面に表示される文字とが異なることがあります。
(例：「¥」が「\」に変わります。か文字は表示できません(表示されないか、他の文字に変わります)。)

(注2) 組合せが不可の機能に対応するボタンは押下できない状態で表示します。

(例：

- "Direction"が「0 to M」で"Padding"「Yes」選択時の "Delimiter"「No」指定。
- "Direction"が「M to 0」の場合で、"Input File"のボリュームのエミュレーションタイプが OPEN-X (X:3,8,9,K,E,L,M,V)の場合は、"Code Conversion"の「Ek<->J」,「Ec/E<->A」,「FILE」(*) ボタンは選択不可(グレー表示)となります。
- "Direction"が「0 to M」の場合で、"Output File"のボリュームのエミュレーションタイプが OPEN-X(X:3,8,9,K,E,L,M,V)の場合は、Code Conversion の「Ek<->J」,「Ec/E<->A」, 「FILE」(*)ボタンは選択不可(グレー表示)となります。
(*)コード変換ファイル入力エリアも入力禁止状態となります)

(1) **File**

FCUに起動パラメタの入力以外の処理を指示するときに使用します。
本ボタンをクリックすると以下のメニューが表示(プルダウン)されます。

-----	*1
Load	Ctrl+L
Save	Ctrl+S
Delete	Ctrl+D
Exit	Alt+F4

*1：本メニューが切離し可能であることを示しています。

本メニューを消去する場合は、本メニュー以外の場所をクリックして下さい。

(i) **Load** (Ctrl+L)

パラメタ定義ファイルから詳細パラメタを読み出し、本画面の各入力パラメタの入力領域に表示します。

- ・パラメタ定義ファイルが作成されていない場合及びパラメタ定義ファイル中に有効な情報がない場合には、本指示は無効です。
- ・本ユーティリティ起動時に、パラメタ定義ファイルが作成されていて、当該ファイルに有効なパラメタがあれば、初期画面にすでに表示されています。
- ・「Load」を指示する毎に次のパラメタセットに切り替わり、最後まで表示するとパラメタ定義ファイル「End line」メッセージを出力します。さらにLoadが指示されると、先頭のパラメタセットに戻ります。

(ii) **Save** (Ctrl+S)

現在、本画面に入力/表示されている詳細パラメタセットを、ロード元のパラメタセットと置換する形でパラメタ定義ファイルに格納します。

- ・パラメタ定義ファイルが作成されていない場合は、新たにファイルを作成してパラメタを格納します。
- ・パラメタ定義ファイルの最後まで表示した状態(End lineメッセージを表示した状態)で、操作画面にパラメタを入力し、「Save」を指示するとパラメタセットの追加となります。

(iii) **Delete** (Ctrl+D)

現在、操作画面にロードされているパラメタセットをパラメタ定義ファイルから削除します。

- ・パラメタセットがロードされていない状態での本コマンドの指示は無効です。

(iv) **Exit** (Alt+F4)

FCUプログラムを終了させます。

(2) **Help**

ヘルプ 情報を表示します。

本ボタンをクリックすると以下のメニューが表示(プルダウン)されます。

----- *1	
Vol_tbl...	Ctrl+T
MF-File...	Ctrl+M
UX-File...	Ctrl+U
Error...	Ctrl+E
On Version...	Ctrl+O

*1 : 本メニューが切離し可能であることを示しています。

本メニューを消去する場合は、本メニュー以外の場所をクリックして下さい。

HMDE/KANJIインストール時はメニューの最後にEdit_prmメニューが表示されHMDE/KANJIのパラメータが表示可能となります。

(i) **Vol_tb** (Ctrl+T)

ボリューム情報(ボリューム定義ファイルの内容)を表示します。

DEV Type表示の左に '*'が表示された場合、データ転送が出来ないことを意味します。

(例 : 3390-3Bに対し、otm方向のデータ転送を指定している場合 '*'を表示します)

・ボリューム定義ファイルの内容については8.2節を参照して下さい。

(ii) **MF-File** (Ctrl+M)

Directionがmtoの時はInputFileに、otmの時はOutputFileに、VSNまたはVSN, VSN識別子を入力した状態で、**MF-File** を選択すると、指定されたメインフレームのデータセット名のリストを表示します。

VSN識別子を指定する場合は、ボリューム定義ファイル上にもVSN識別子の指定が必要になります。

・本機能を使用する場合、

データ転送方向が「M to O」のとき、「Input File」のファイル名入力領域に

データ転送方向が「O to M」のとき、「Output File」のファイル名入力領域に

ボリュームシリアルナンバーが指定されていなければなりません。

・上記手順により指定されたボリューム内のデータセット名、データセット編成種別、レコード形式、ブロック長、レコード長、データセットサイズ、本ユーティリティによる処理の可否を表示します。

・VSN識別子(VSN identification)はボリューム定義ファイル上記載がないとスペースとなります。

Dataset Information : VSN = xxxxxx		Device Emulation Type = 3390-3B				
VSN identification = AAAAABBBBBCCCCDDDDDEEEEFFFFFGGGG						
Dataset Name	DO	RF	BL	RL	DS	Ex (Cyl:Trk)
*SAMFILE01.FIX	SAM	FB	4096	128	150	3:0
-DAMFILE.F	DAM	FB	4096	128	30	0:2
*SAMFILE02.VAL	SAM	VB	4000	80	50	10:0
-PAMFILE	PAM	FB	5000	100	200	5:10
-VIRTUALSTORAGEACCESS	VSAM	VB	32768	4096	3000	20:0
-UNDEFSAMFILE	SAM	U	8000	200	80	1:0
-SAMFILESANDED	SAM	S	8192	8192	300	6:0
?SAMFILE.VSE	SAM	??	0	0	0	10:0

図8.6-3 メインフレームのデータセットのリスト

・"Dataset Name":データセット名

データセット名の前の1文字

*:本ユーティリティで処理できるデータセットであることを示す。

-:本ユーティリティで処理できないデータセットであることを示す。

? :本ユーティリティで必要とする情報が定義されていないため処理できるか否かが不明であるか、または表示内容が正しくない可能性があります。RF=??またはBL=0 または RL=0の時、VSEパラメータでレコード形式・レコード長・ブロック長を正しくすれば処理できるようになります。またデータ量は、DSの値によりません。

- ・“DO” : データセット編成種別
 - SAM : 順編成
 - DAM : 直接編成
 - PAM : 区分編成
 - VSAM : Virtual Storage Access Method編成
 - ??? : 上記以外の編成種別で、本ユーティリティでは識別できません。

- ・“RF” : レコード形式
 - F : 固定長レコード
 - FB : 固定長ブロック化レコード (XSP時はFと表示します。)
 - V : 可変長レコード
 - VB : 可変長ブロック化レコード (XSP時はVと表示します。)
 - U : 不定長レコード
 - S : スパンドレコード
 - ?? : 上記以外のレコード形式で、本ユーティリティでは識別できません。

・“BL” : ブロック長(バイト)

・“RL” : レコード長(バイト)

- ・“DS” : データセットサイズ(トラック数)

マルチボリューム定義ファイルに当該データセットが登録してある場合、全ボリュームのデータセットサイズの合計を表示します。

DSの値が0の場合、対象データセットは0バイトデータセットとなります。

- ・“Ex(Cyl:Trk)” : アーク時のデータセットのサイズをシリンダ数:トラック数で表示します。

マルチボリューム定義ファイルに当該データセットが登録してある場合でも、先頭ボリュームのみのアーク時のサイズを示します。

補足 : GUI環境が無い場合には、LISTVOL/LISTVOL64でメインフレームのデータセット名のリストを表示できます。
(8.7.1.2参照)

(iii) **UX-File** (Ctrl+U)

出力となるUNIXファイルの確認のために、ファイル名のリストを表示します。

- ・本機能を使用する場合、
 - データ転送方向が「M to 0」のとき、“Output File”のファイル名入力領域にデータ転送方向が「0 to M」のとき、“Input File”のファイル名入力領域にディレクトリ名(ディレクトリ名の後に"/"を付加する)を指定します。(ファイル名は無視します)
- ・ファイル名入力領域にディレクトリ名が指定されている場合、当該ディレクトリ内のファイル名を表示します。
- ・ファイル名入力領域にディレクトリ名が指定されていない場合、カレントディレクトリ内のファイル名を表示します。

UNIX FILE LIST : DIR = /aaaaa/bbbbb/ccccc

dddd.d	eeeeee	ffffff.ffffff
hhhhh.hhhh	zzzzz.z	xxxx.x
yyyyyyyyy		

図8.6-4 UNIXファイルリスト

(iV) **Error** (Ctrl+E)

最後に発生したエラーに関する情報を表示します。

(V) **OnVersion** (Ctrl+O)

FCUのバージョン情報を表示します。

(3) **Parameter File**

パラメータ定義ファイルの名称を指定します。

- ・起動時にパラメータ定義ファイルを指定した場合、当該ファイル名を表示します。
指定しなかった場合はデフォルトのパラメータ定義ファイル名 (fcudata.param) を表示します。

(4) **Volume File**

ボリューム定義ファイル名を表示します。

- ・ボリューム定義ファイル名は「datasetmount.dat」です。

(5) **Direction**

データの転送方向を表示します。

M to O : メインフレームのデータセットからオープンシステムのファイルへの転送を意味します。
O to M : オープンシステムのファイルからメインフレームのデータセットへの転送を意味します。

- ・パラメータ定義ファイルが作成されている場合、パラメータ定義ファイル中の「mto」/「otm」の指定に従って上記ボタンのどちらかのボタンを押下状態にして表示します。

(6) **Input File**

データ転送方向が「M to O」の場合、入力となるメインフレームのデータセット名を指定します。

データ転送方向が「O to M」の場合、入力となるオープンシステムのファイル名を指定します。

- ・パラメータ定義ファイルが作成されている場合、パラメータ定義ファイル中の入力側データセット名またはファイル名を表示します。

入力がメインフレームのデータセットの場合

- ・「ボリューム・シリアル・ナンバ」と「データセット名」を“:”で区切った形式で指定します。
- ・「ボリューム・シリアル・ナンバ」は6桁以下、「データセット名」は44桁以下の英文字及び英数字で指定します。

(例) **DK7CE0:SAM.F80**

入力がオープンシステムのファイルの場合

- ・絶対パスまたは相対パスで指定します。

(例) ・絶対パス: /ディレクトリ名/ディレクトリ名/ファイル名
・相対パス: ファイル名
./ファイル名
ディレクトリ名/ディレクトリ名/ファイル名
./ディレクトリ名/ディレクトリ名/ファイル名

- ・パラメータ定義ファイルが作成されている場合、パラメータ定義ファイル中の入力側データセット名またはファイル名を表示します。

(7) Output File

データ転送方向が「M to O」の場合、出力となるホップシステムのファイル名を指定します。
データ転送方向が「M to O」でパイプ指定の場合、パイプ (Named pipe) 名を指定します。
データ転送方向が「O to M」の場合、出力となるメインフレームのデータセット名を指定します。
データセット名、ファイル名の指定方法は「Input File」と同様です。

・パラメタ定義ファイルが作成されている場合、パラメタ定義ファイル中の出力側データセット名、
ファイル名またはパイプ名を表示します。

(注1) データ転送方向が「M to O」でファイルがすでに存在する場合は、当該ファイルの先頭からデータを
オーバーライトします。このとき、オーバーライトを行ってよいかどうかの確認メッセージを「Status」
表示領域に出力します。オーバーライトしてよければ「OK」をクリックし、ファイル変換を中止するの
であれば「Cancel」をクリックして下さい。

ファイルが存在しない場合は、新たにファイルを作成してデータを書き込みます。

(注2) データ転送方向が「O to M」の場合、常にオーバーライトになります。

データセットの新規作成、エリアの自動拡張は行いません。

(8) PIPE

データ転送方向がmtoのとき、パイプ (Named pipe) 出力の可否を指定します。

・パイプ出力を指定する時は、PIPEの前のトグルボタンをチェックして下さい。

・ファイル出力をする時はトグルボタンのチェックをはずして下さい。

・パラメタ定義ファイルが作成されている場合、パラメタ定義ファイルのパイプ指定の可否に合わせて、
チェックの有無を表示します。

(9) Code Conversion

データ実体に対するコード変換の可否を指定します。

・EBCDIC⇔ASCIIのコード変換が必要であれば「Ec/E<->A」を、EBCDIK⇔JIS8のコード変換が必要で
あれば「Ek<->J」を、コード変換テーブルファイルを使用してコード変換を行うのであれば「FILE」を、
コード変換不要であれば「No」をクリックして下さい。

・パラメタ定義ファイルが作成されている場合、パラメタ定義ファイル中のコード変換可否の指定に合わせて、
「Ek<->J」か「Ec/E<->A」か「No」か「FILE」のいずれかを押下状態にして表示します。

・コード変換テーブルファイルを使用してコード変換を行う場合、「FILE」の後ろのテキストボックスに当該ファイル名
を入力して下さい。ファイル名の指定形式はInput File / Output FileのUNIXファイル名の指定形式
と同様です。

ホップシステムーホップシステム間データ転送の場合

・コード変換はサポートしていません。コード変換指定パラメタは「No」を指定して下さい。

・コード変換、ファイル名を指定するとエラーになります。

・パラメタ定義ファイルが作成されている場合、パラメタ定義ファイル中のコード変換の指定がされていると、
エラーとなります。

(注) GUIなし環境の場合

起動パラメタにコード変換が指定された場合はエラーとし次のパラメタの処理を行います。

起動オプションとして、「-nc」指定で起動された場合に、パラメタにコード変換が指定された場合も
エラーとし、次のパラメタの処理を行います。次のパラメタが無い場合は、処理を終了します。

(10) **Padding**

パディングの可否を指定します。

- ・メインフレームからオープンシステムへの転送で、メインフレームデータセットのデータ形式が可変長の場合、UNIXファイルとしてはデータ長を一定にし、データ実体が不足する部分をパディングした形式で出力するか否かを指定します。
- ・オープンシステムからメインフレームへの転送で、メインフレームデータセットのデータ形式が固定長の場合、データ長を一定にし、データ実体が不足する部分をパディングした形式で出力するか否かを指定します。
- ・パディングが必要であれば「Yes」を、不要であれば「No」をクリックして下さい。
- ・パラメタ定義ファイルが作成されている場合、パラメタ定義ファイル中のパディング可否の指定に合わせて、「Yes」か「No」のいずれかを押下状態にして表示します。

(11) **Delimiter**

(i) データ転送方向が「M to 0」の場合

UNIXファイルへのデータ書き込み時に、デリミタを付加するか否かを指定します。

- ・デリミタの付加が必要であれば「CR」または「LF」を、不要であれば「No」をクリックして下さい。
- ・「CR」/「LF」が指定されている場合、各データ実体の後に「CR」/「LF」を付加してUNIXファイルに書き込みます。

(ii) データ転送方向が「0 to M」の場合

UNIXファイルからのデータ読出し時に、デリミタが付加されているか否か(デリミタを取り除くか否か)を指定します。

- ・UNIXファイルにデリミタとして「CR」/「LF」が付加されていれば「CR」/「LF」を、デリミタが付加されていない場合は「No」をクリックして下さい。
- ・「CR」/「LF」が指定されている場合、デリミタを取り除いてメインフレームのデータセットに書き込みます。
- ・「No」が指定された場合は、デリミタが無いものとみなして転送します。
- ・Paddingを指定した場合は、UNIXファイルにデリミタが付加されている必要があります。「No」は指定できません。
- ・転送時のデータ編集形式は、7.5節を参照して下さい。
- ・UNIXファイルにデリミタとして「CRLF」が使用されている場合はエラーとなります。

(iii) パラメタ定義ファイルが作成されている場合、パラメタ定義ファイル中のデリミタ可否の指定に従って「CR」、「LF」または「No」を押下状態にして表示します。

(12) **Emp**

データなしファイルを転送の対象とするか否かを指定します。

- ・データなしファイルを転送の対象とする場合、「Yes」をクリックして下さい。
- ・データなしファイルを転送の対象としない場合、「No」をクリックして下さい。
- ・パラメタ定義ファイルが作成されている場合、パラメタ定義ファイル中のEmpパラメタの指定に従って「Yes」または「No」を押下状態にして表示します。

(1 3) **RDW**

RDW(各レコード対応のレコード長)を付加するか否かを指定します。

- ・RDWを付加するとき「Yes」、付加しないとき「No」を指定して下さい。
- ・パラメタ定義ファイルが作成されている場合、パラメタ定義ファイル中のRDWパラメタの指定に従って「Yes」または「No」を押下状態にして表示します。

(1 4) **VSE/VOS1**

レコード形式・レコード長・ブロック長が定義されていないデータセットを使用する場合に、レコード形式・レコード長・ブロック長を「,」で区切って指定します。レコード形式・レコード長・ブロック長の指定形式、値の制限等については8.4節を参照して下さい。(一部が定義されていない場合でも全て指定が必要となります。

パラメタ定義ファイルが作成されている場合、パラメタ定義ファイル中のVSEパラメタの内容を表示します。

(注) オープンシステム-オープンシステム間のデータ交換時では使用しません。

(1 5) **OK**

ファイル変換処理の開始を指示します。

- ・必要データの入力を終了し、FCUにファイル変換処理の開始を指示するときにクリックして下さい。
- ・オーバーライトの確認に対して「OK」をクリックするとオーバーライトが実行されます。

(注) ファイル変換処理開始後に本ボタン(及び他のボタン)をクリックすると、当該指示は保留され、ファイル変換処理終了後に実行されます。ファイル変換処理実行中にみだりにボタンをクリックすることは避けて下さい。

(1 6) **Cancel**

FCUへの処理依頼をキャンセルします。

- ・本画面へのデータ入力中にのみ有効です。
- 「Cancel」が指示されると操作画面は初期状態に戻ります。
- ・ファイル変換処理開始後に「Cancel」を指示しても、処理は中断できません。

(1 7) **Status**

ファイル変換処理の進捗状況またはエラー情報を表示します。

- ・データセットのサーチ/属性チェック中は、「Now checking」を表示します。
- ・ファイル変換開始時、「Start」を表示します。
- ・ファイル変換処理実行中は、処理の進捗度を「%」で表示します。(ただし、UNIX版ではメインフレームがVSE/VOS1の場合、MTO時は進行度を表示しません。)
- ・ファイル変換処理が終了したときに「Complete」を表示します。
- ・エラーが発生し、ファイル変換処理が中断した場合は、エラー情報を表示します。

(18) ファイル変換処理の再実行の確認

ファイル変換処理を実行後、パラメータを全く変えることなくファイル変換処理を指示すると、同じ処理を再実行してもよいか否かを確認するために以下の画面を表示します。

同じ処理を再実行するのであれば「Execute」、再実行しないのであれば「Cancel」を指示して下さい。

(注) ファイル変換処理の実行を指示する際に、「OK」ボタンを何回も連続してクリックした場合も本画面が表示されます。

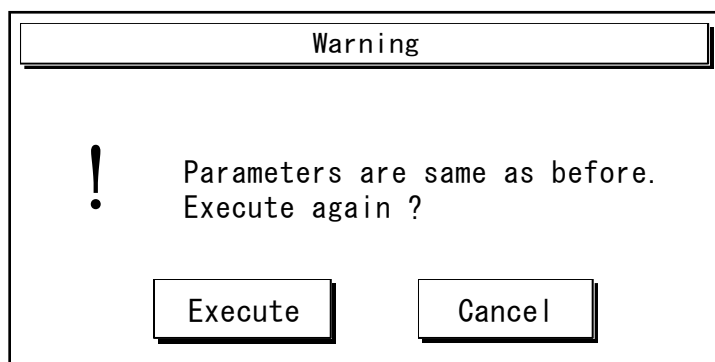


図8.6-5 ワーニング画面

8.6.2 Windows の場合

8.6.2.1 基本操作説明

ここでは、基本的な FCU の使い方をご理解いただくために、標準的な操作手順を例で示します。

以下の例は、図 8.6.2-1 に示すようなパラメタ定義ファイル(Test.prm)を FCU の画面を使い定義作成した後、さらに、これを使って実際にファイル変換を実行するまでの基本手順です。

[この例の前提条件]

- ① メインフレームの VOLSER (AAAAAA)に、データセット(DATASET1, DATASET2)が正しく存在するものとします。
- ② オープンシステムファイル(E:¥test¥data.txt)は、既に作成され、DATASET2 の編成、レコード形式等に合致するデータであるものとします。
- ③ SCSI ケーブル等の物理的、論理的な接続は既に完了しているものとします。
- ④ オープンシステム、RAID 等の機器類の電源は投入され、正常に動作しているものとします。

```
mt0 AAAAAA:DATASET1 E:test¥out1.txt EA No CRLF Emp=No RDW=No
otm E:¥test¥data.txt AAAAAA:DATASET2 EA No CRLF Emp=No RDW=No
mt0 AAAAAA:DATASET2 E:¥test¥out2.txt EA No CRF Emp=No RDW=No
end
```

図 8.6.2-1 基本操作で例として使用するパラメタ定義ファイル(Test.prm)

この例の操作は、次の(1)～(11)の手順に大きく分かれています。

- (1) FCU 起動前に行うこと
- (2) FCU の起動
- (3) 1 件目のパラメタ行の入力
- (4) 2 件目のパラメタ行の入力
- (5) 3 件目のパラメタ行の入力
- (6) 3 件目のパラメタ行の修正
- (7) ディスク上のパラメタ定義ファイルへの保存
- (8) ディスク上の既存のパラメタ定義ファイルを開く
- (9) 1 件目のパラメタ行の実行(単独実行)
- (10) 2～3 件目のパラメタ行の実行(連続実行)
- (11) FCU の終了

(1) FCU 起動前に行うこと

1) アドミニストレータ権限のあるユーザでログインします。

2) WindowsNT の場合「スタート」 - 「設定」 - 「コントロールパネル」から「SCSI アダプタ」を選択します。

Windows2000Server の場合「スタート」 - 「プログラム」 - 「管理ツール」 - 「コンピュータの管理」 - 「デバイスマネージャ」から「ディスクドライバ」を選択します。

WindowsServer2003/Server2008/Server2008 R2/Server2012/Server2012 R2
/Server2016/Server2019 の場合

「スタート」-「コントロールパネル」-「管理ツール」-「コンピュータの管理」-「システムツール」から「デバイスマネージャ」を選択します。

3) ハードディスク(RAID)がどのように認識されているか確認します。

4) WindowsNT の場合「スタート」 - 「プログラム」 - 「管理ツール(共通)」 - 「ディスクアドミニストレータ」を選択します。

Windows2000Server の場合「スタート」 - 「プログラム」 - 「管理ツール」 - 「コンピュータの管理」から「ディスクの管理」を選択します。

WindowsServer2003/Server2008/Server2008 R2/Server2012/Server2012 R2
/Server2016/Server2019 の場合

「スタート」-「コントロールパネル」-「管理ツール」-「コンピュータの管理」-「ストレージ」から「デバイスマネージャ」を選択します。

5) 物理ディスク番号を確認します。

6) ホリューム定義ファイルを作成します

(これは、作成例です。)


```
¥¥. ¥PHYSICALDRIVE2 AAAAAA 3380-KA
¥¥. ¥PHYSICALDRIVE3 BBBBBB 3380-KA
end
```

図 8. 6. 2-2 ホリューム定義ファイル(datasetmount. dat)

(2) FCU の起動

1) 「スタート」 - 「プログラム」 - 「Fcu」 - 「fcu」を選択します。

2) FCU 本体画面が起動するので、

「Volume Info」ボタン  を押し、ボリューム情報ダイアログを表示します。

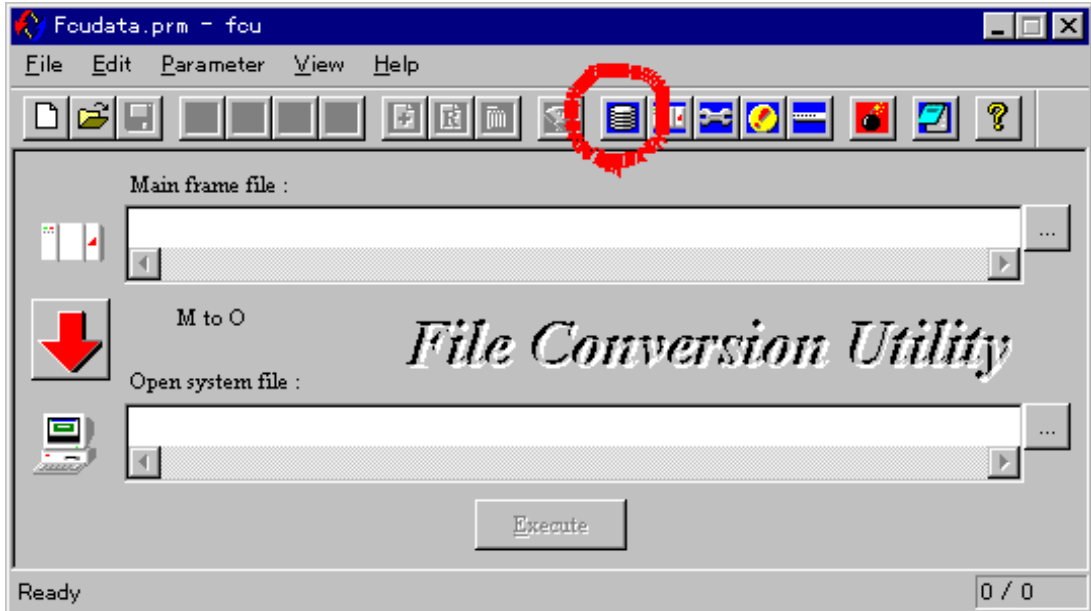


図 8.6.2-3 FCU 本体画面

3) ボリューム情報ダイアログに ○ が表示されていることを確認します。

(×が表示されている場合は、ボリューム定義ファイルを修正します。)

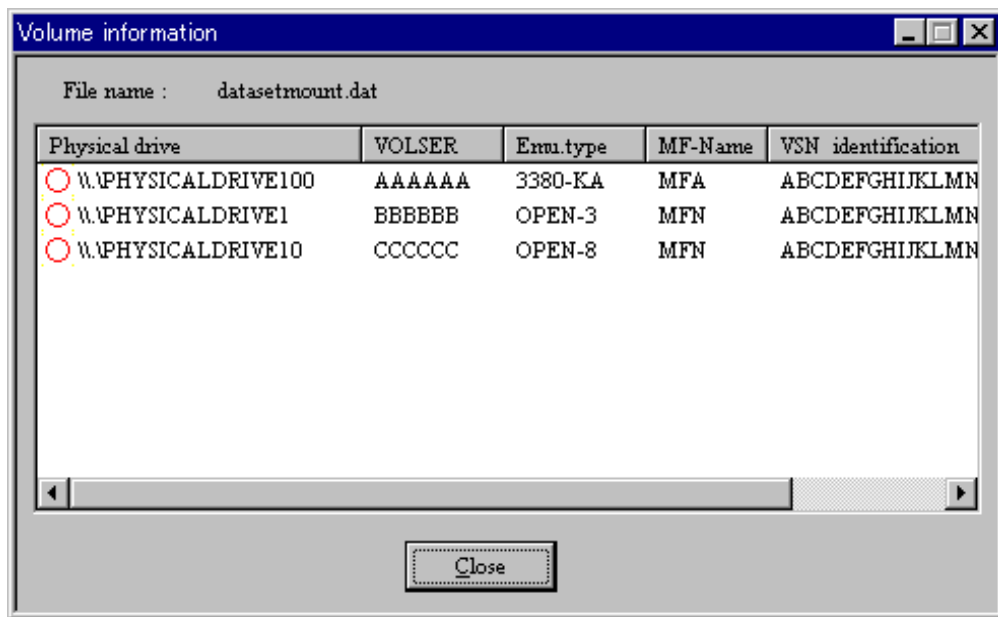


図 8.6.2-4 Volume information ダイアログ

(3) 1件目のパラメタ行の入力

1) データ転送方向は、既に mto 方向が選択されているので、何もしません。

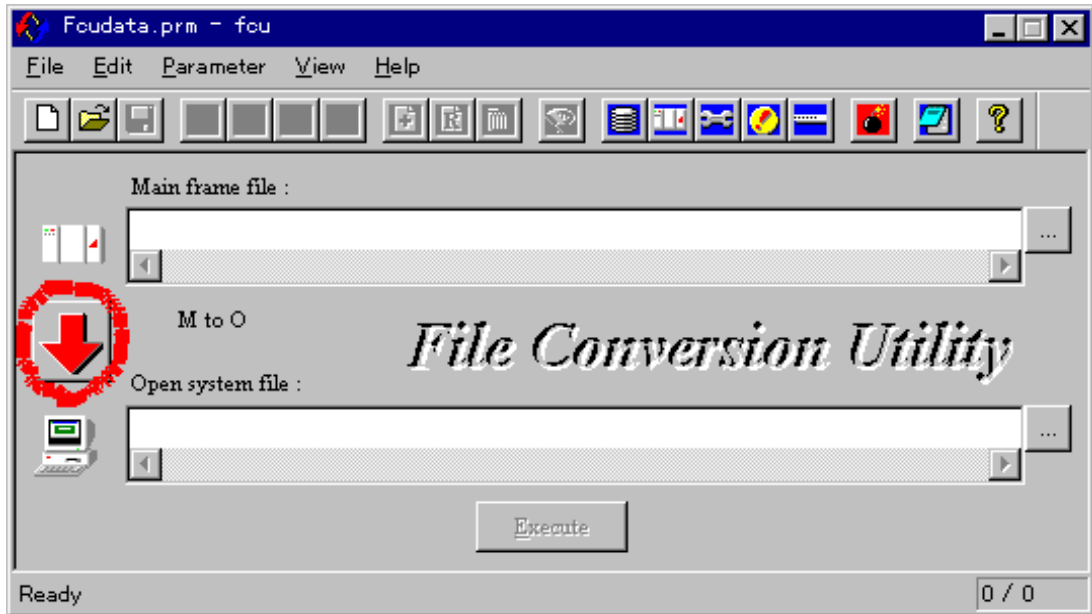



図 8.6.2-5 FCU 本体画面

2) メインフレームファイル選択ボタン  を押します。

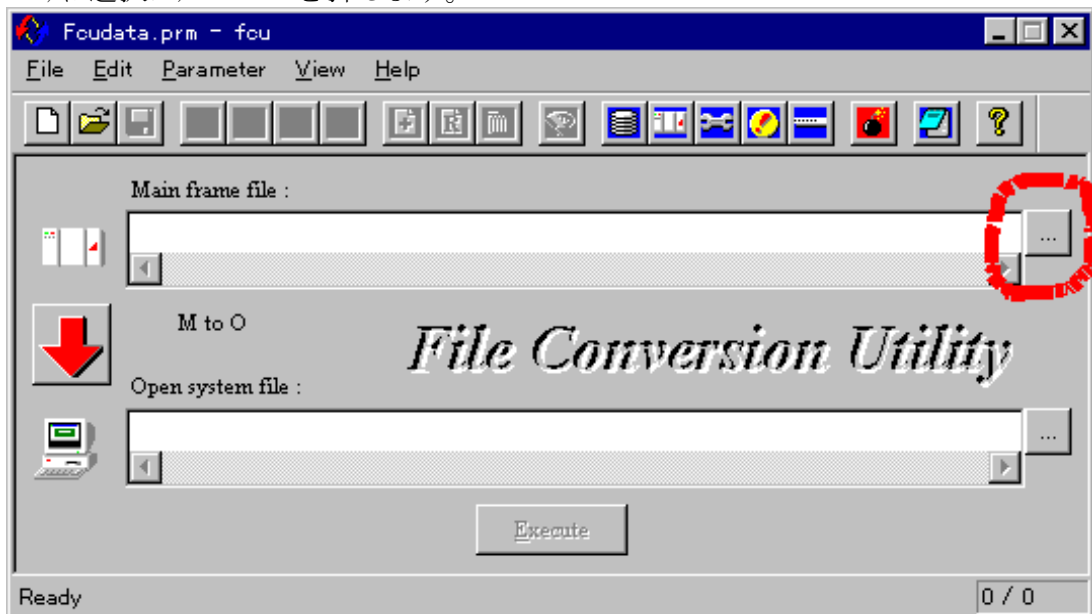


図 8.6.2-6 FCU 本体画面

3) メインフレームファイル選択ダイアログが表示されます。

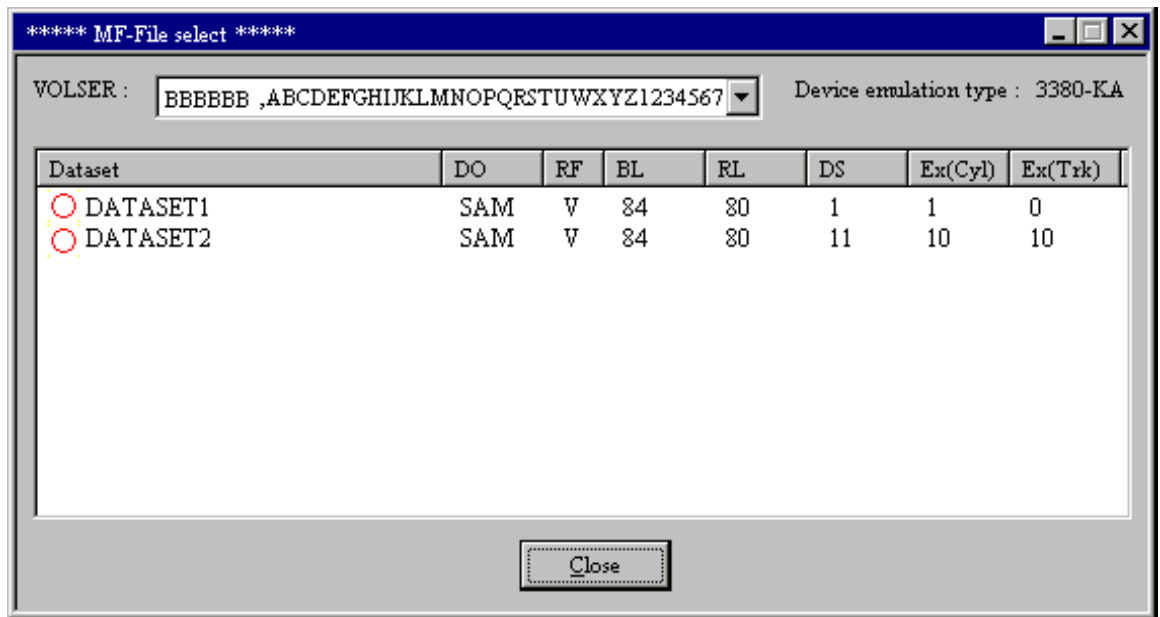


図 8.6.2-7 メインフレームファイル選択ダイアログ

4) メインフレーム選択ダイアログの「VOLSER」コンボボックスより AAAAAA を選択します。

5) DATASET1 をダブルクリックします。

6) Main frame file 欄に入力されました。

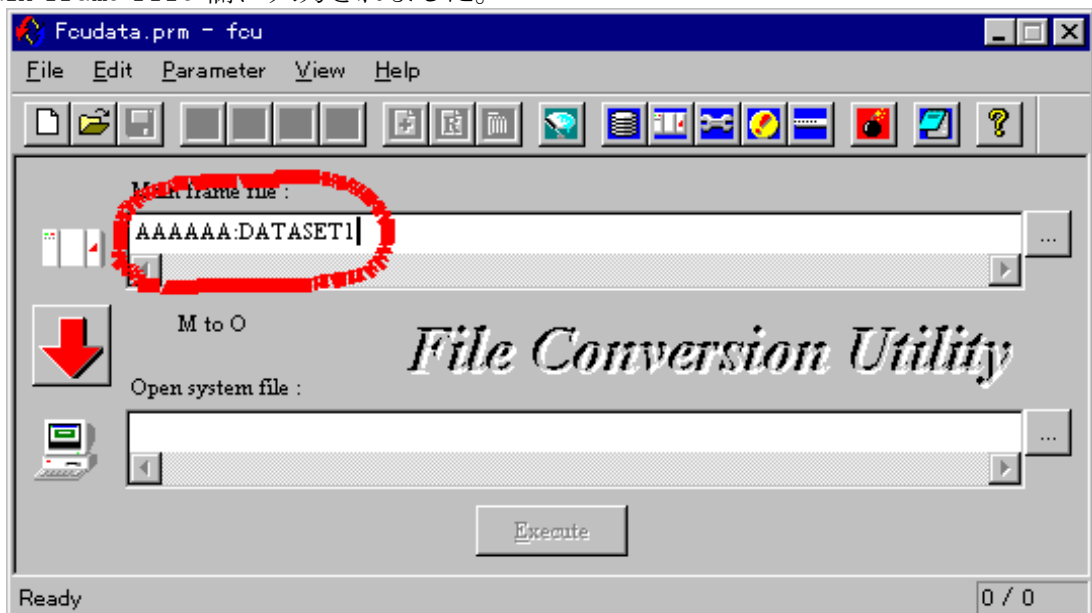



図 8.6.2-8 FCU 本体画面

7) オープンシステムファイル選択ボタン  を押します。

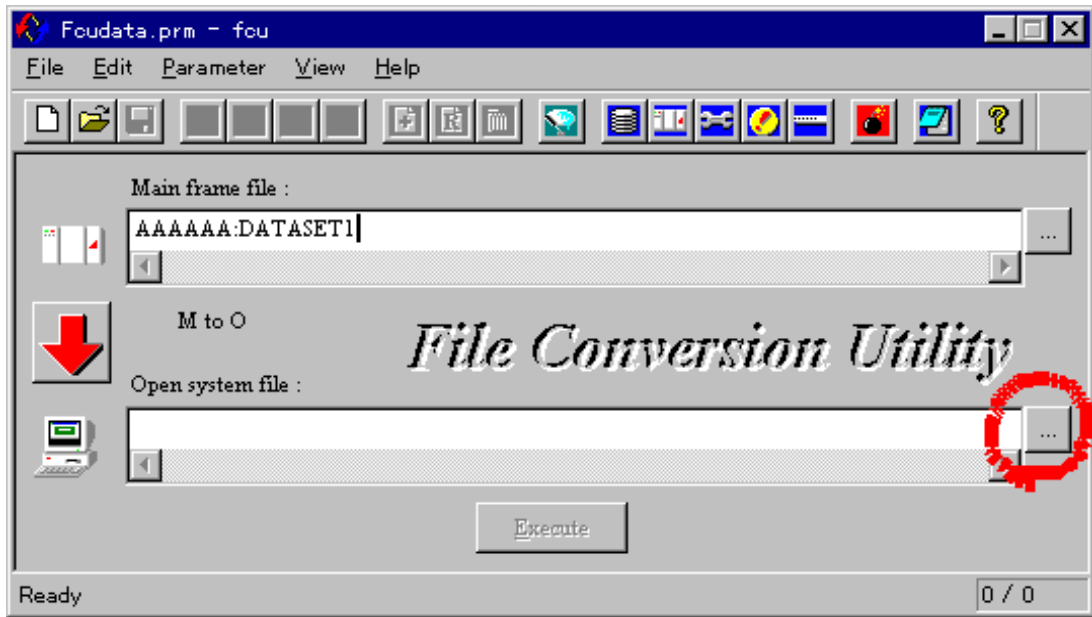


図 8.6.2-9 FCU 本体画面

8) ファイルを開くダイアログが表示されます。

ファイルを開くダイアログのファイル名欄に直接オープンシステムファイル名を入力し、開くボタンを押します。

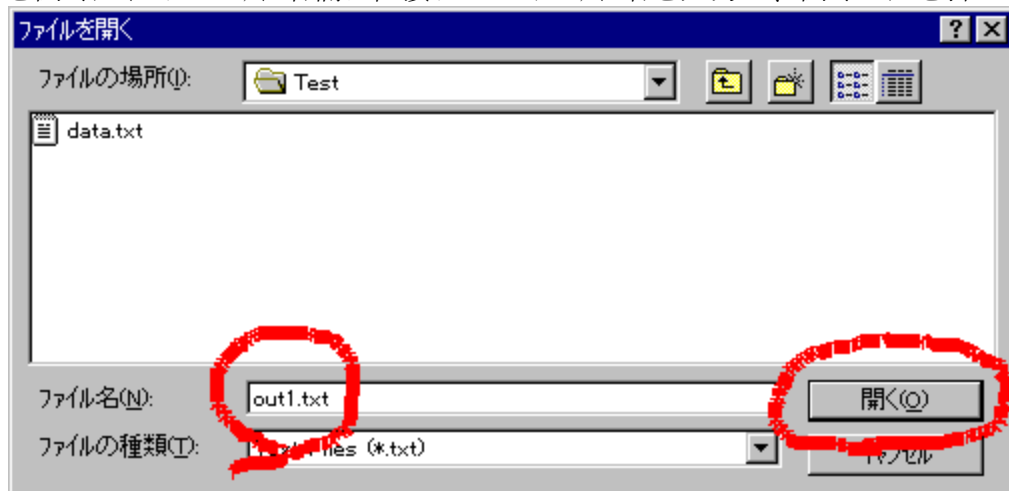


図 8.6.2-10 ファイルを開くダイアログ

9) Open system file 欄に入力されました。

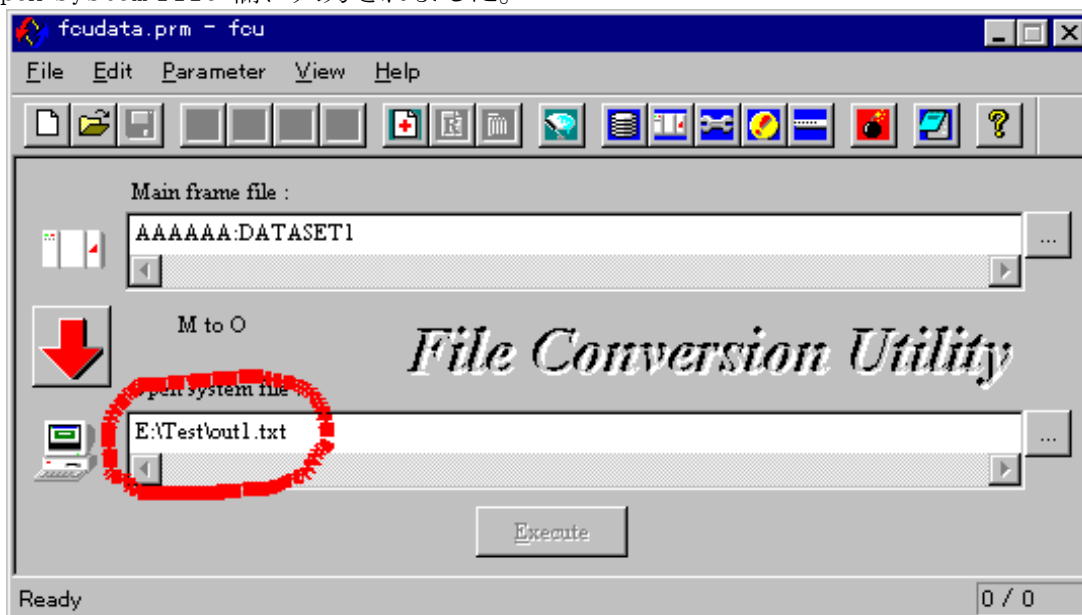


図 8.6.2-11 FCU 本体画面

- 10) 「Option」ボタン  を押します。

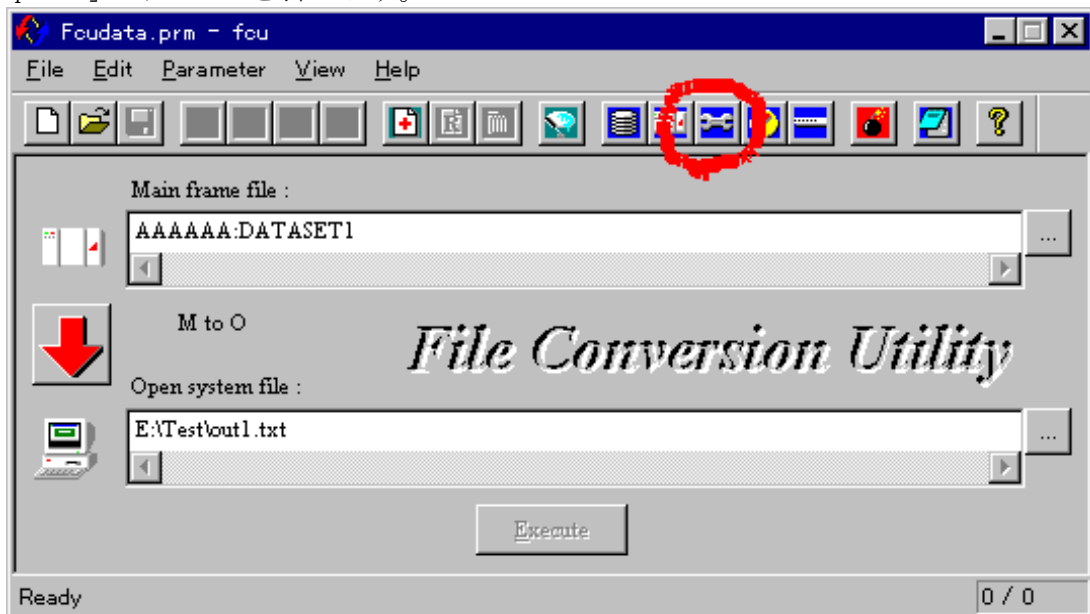


図 8. 6. 2-12 FCU 本体画面

- 11) Option ダイアログが表示されるので、「EBCDIC <--> ASCII」ラジオボタンを選択し、「Delimiter」チェックボックスをチェックします。設定終了後、「Close」ボタンを押下します。

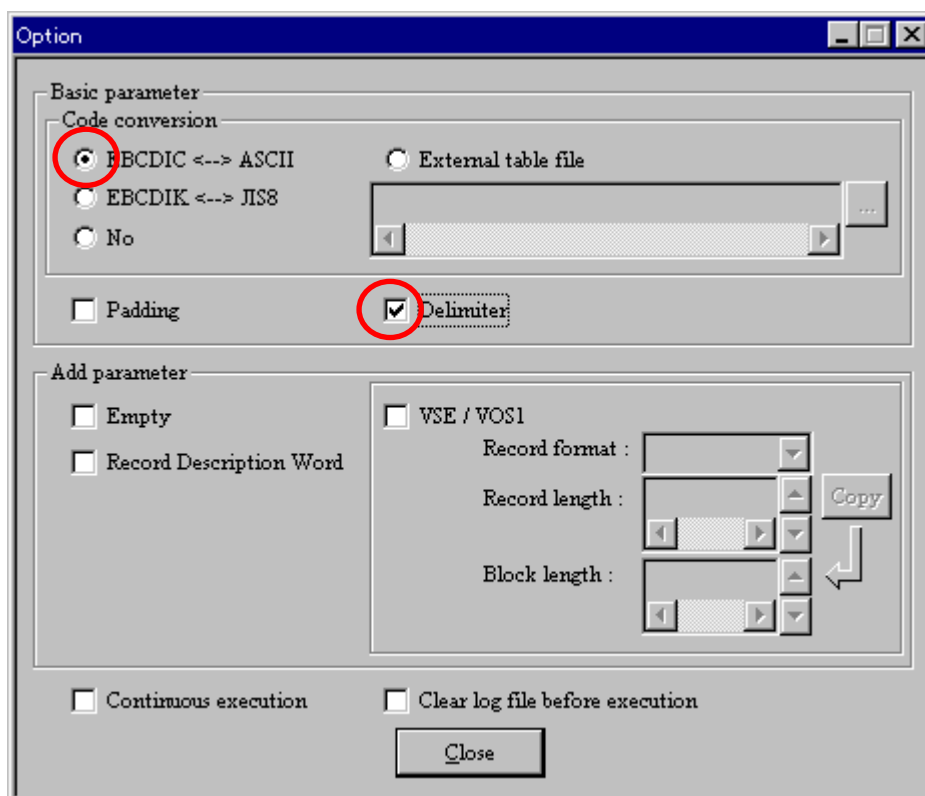



図 8. 6. 2-13 Option ダイアログ

- 12) 「Insert」ボタン  を押して、メソッド上で編集中のパラメタ定義ファイルに登録します。
この操作により、FCU 本体画面及び、Option ダイアログで指定した内容が、メソッド上で編集中のパラメタ定義ファイルに取り込まれます。

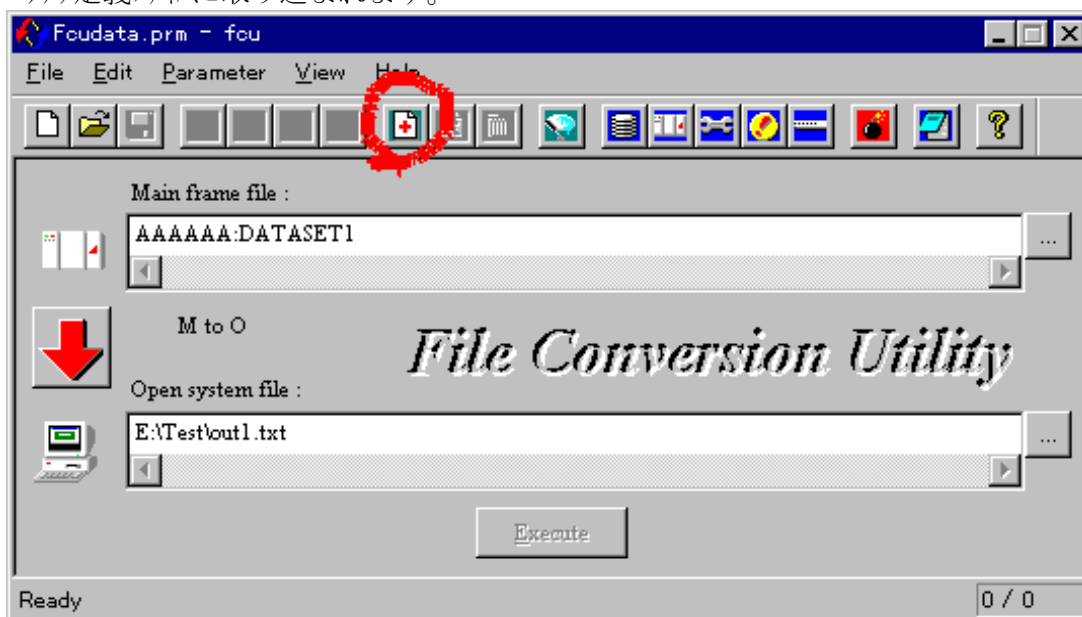


図 8. 6. 2-14 FCU 本体画面

- 13) ステータスバーが 1 / 1 になり、1 件登録されたことが分かります。

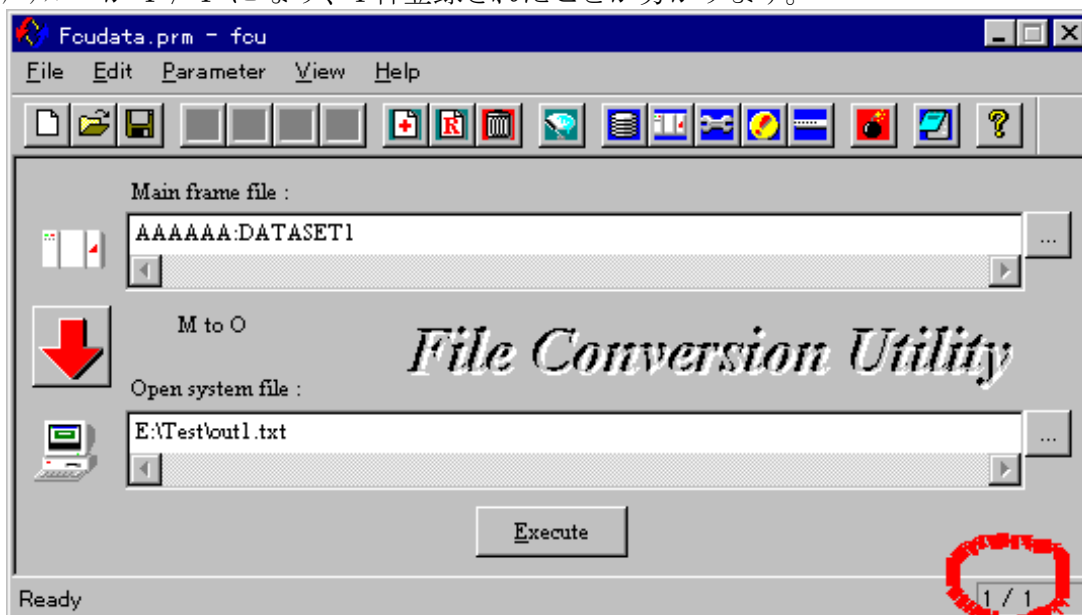


図 8. 6. 2-15 FCU 本体画面

(4) 2件目のパラメタ行の入力

- 1) 2件目は、データ転送方向を otm とする為、データ転送方向ボタンをクリックさせ矢印を↑します。データ転送方向が otm になります。

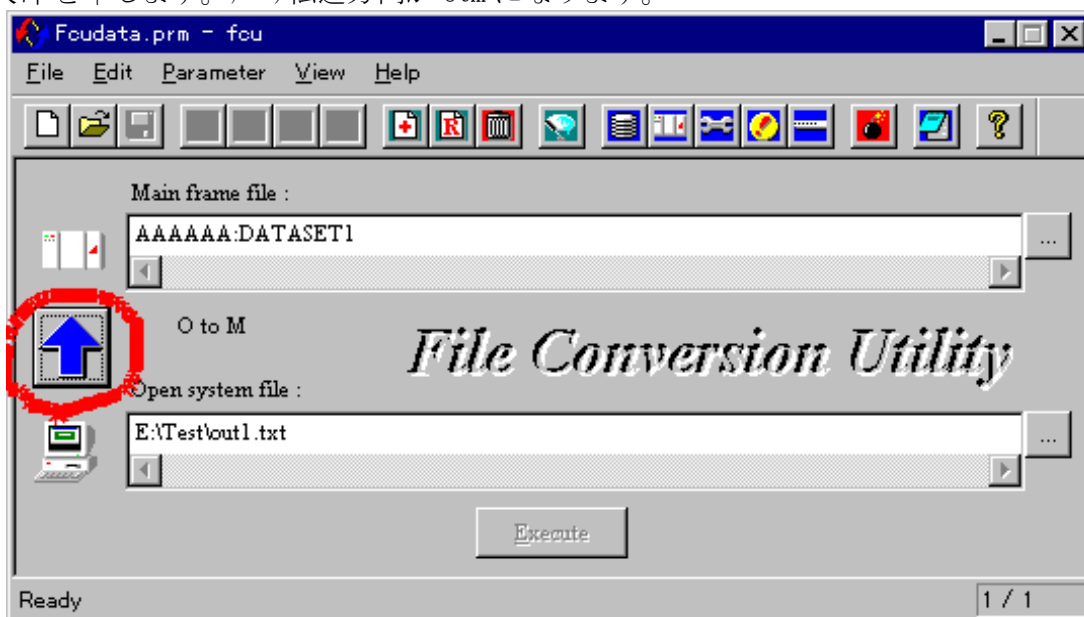



図 8. 6. 2-16 FCU 本体画面

- 2) メインフレームファイル選択ボタン  を押します。

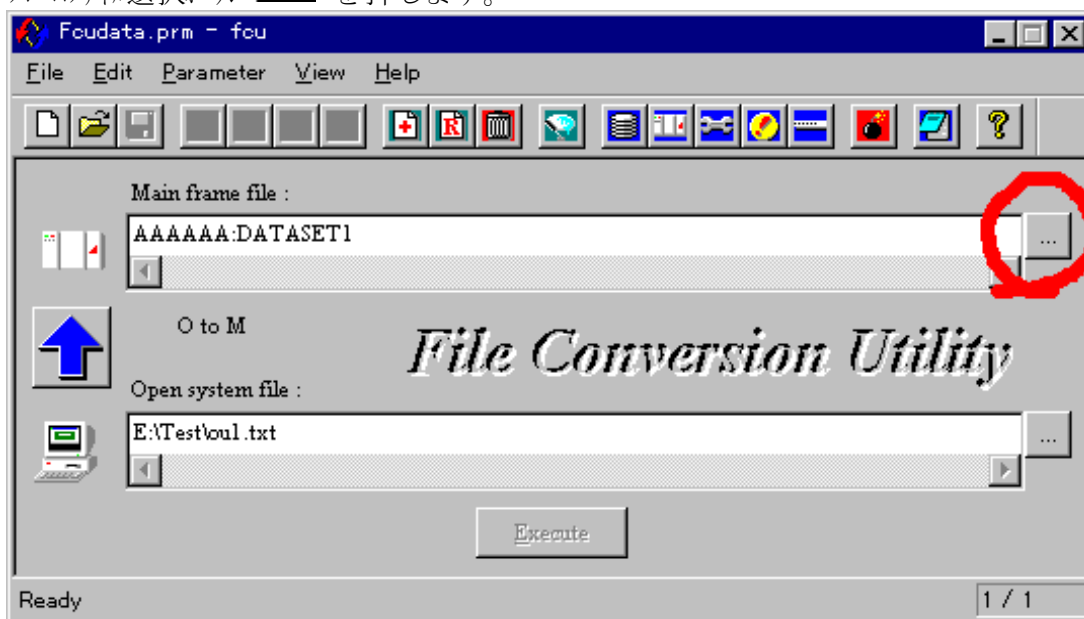


図 8. 6. 2-17 FCU 本体画面

3) メインフレームファイル選択ダイアログが表示されます。

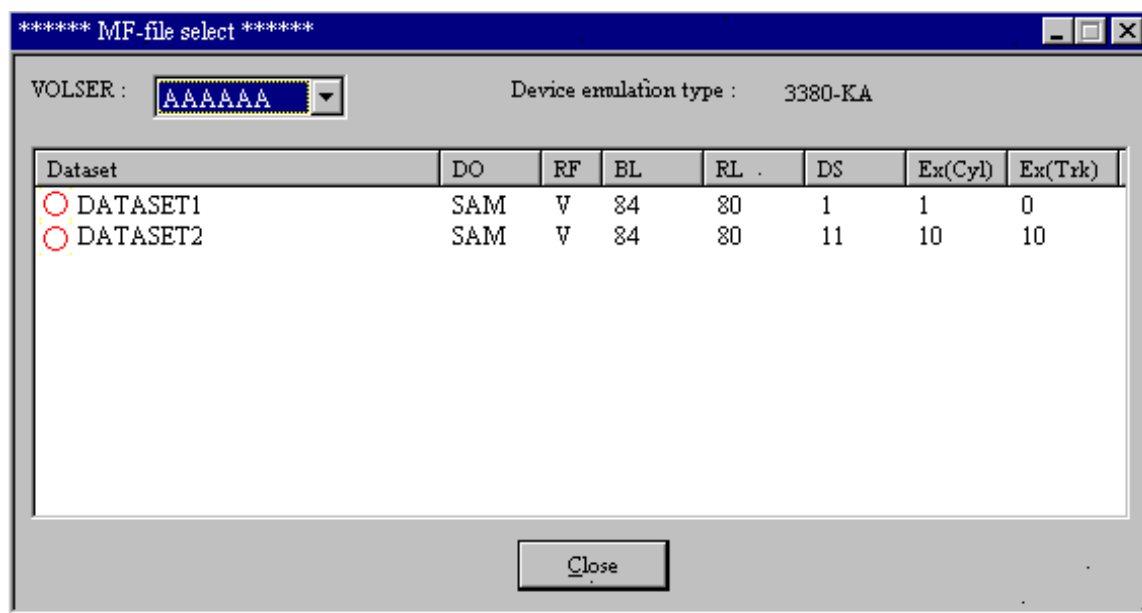



図 8. 6. 2-18 メインフレームファイル選択ダイアログ

4) メインフレーム選択ダイアログの「VOLSER」コンボボックスより AAAAAA を選択します。

5) DATASET2 をダブルクリックします。

6) Main frame file 欄に file 名が入力されます。

次にオープンシステムファイル選択ボタン  を押します。

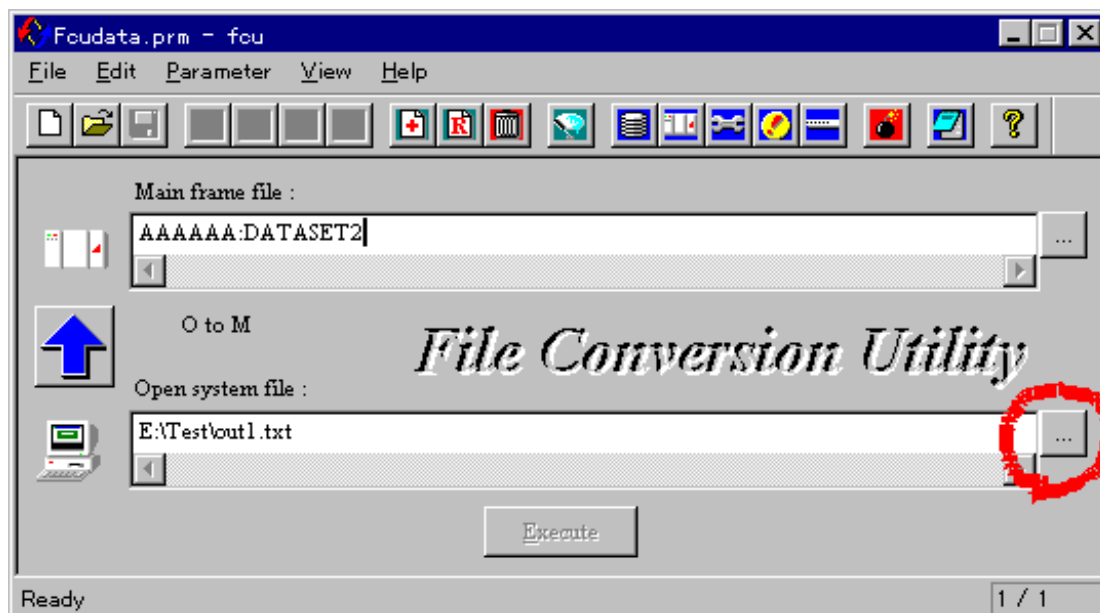


図 8. 6. 2-19 FCU 本体画面

7) ファイルを開くダイアログが表示されます。次に表示されている data.txt をダブルクリックします。

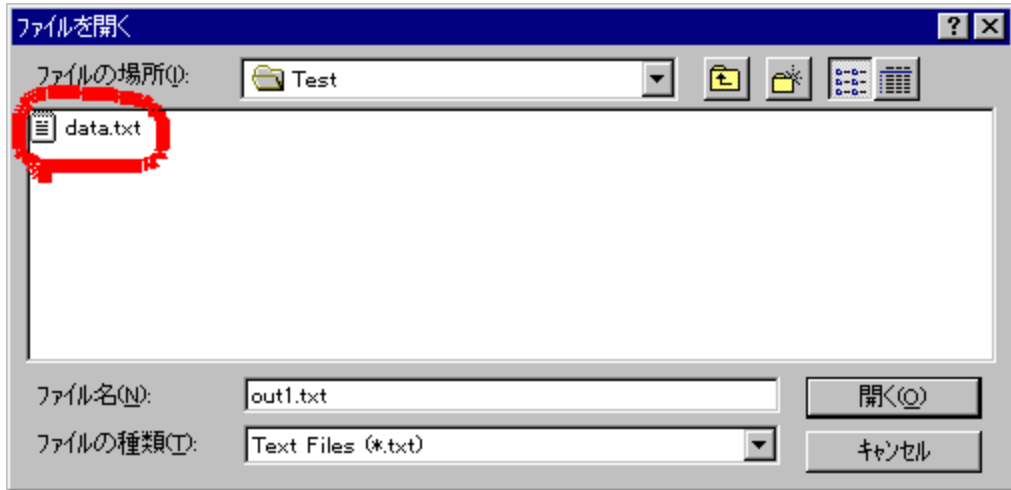


図 8. 6. 2-20 ファイルを開くダイアログ

8) Open system file 欄に入力されました。

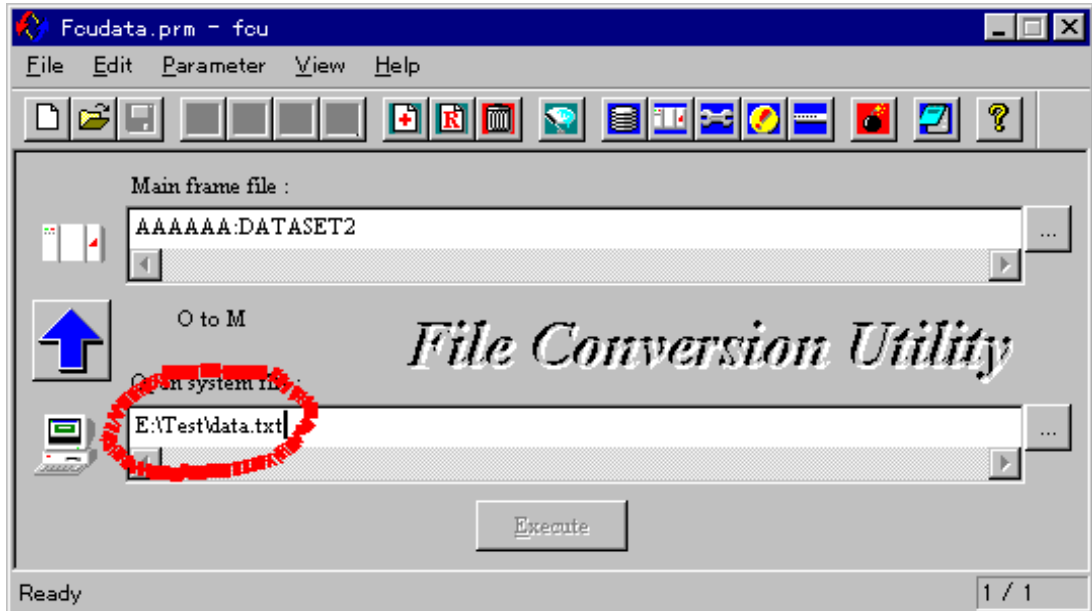


図 8. 6. 2-21 FCU 本体画面

9) Option ダイアログの内容は、1 件目と同じなので、何もしません。

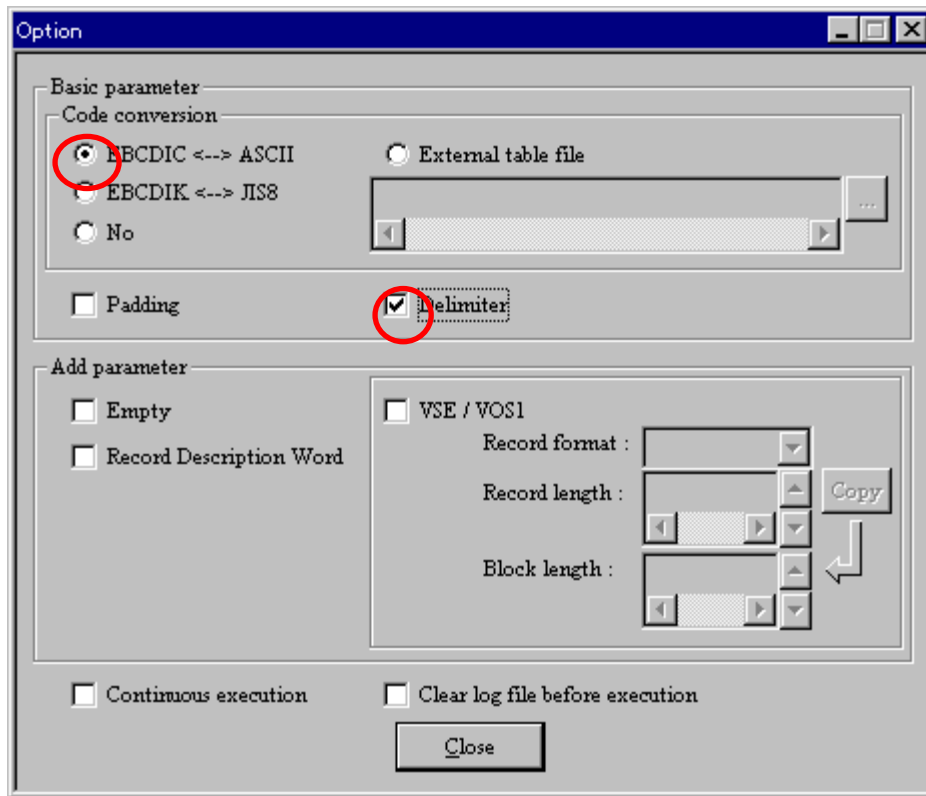



図 8.6.2-22 Option ダイアログ

10) 「Insert」ボタン  を押して、メモリ上で編集中のパラメタ定義ファイルに登録します。

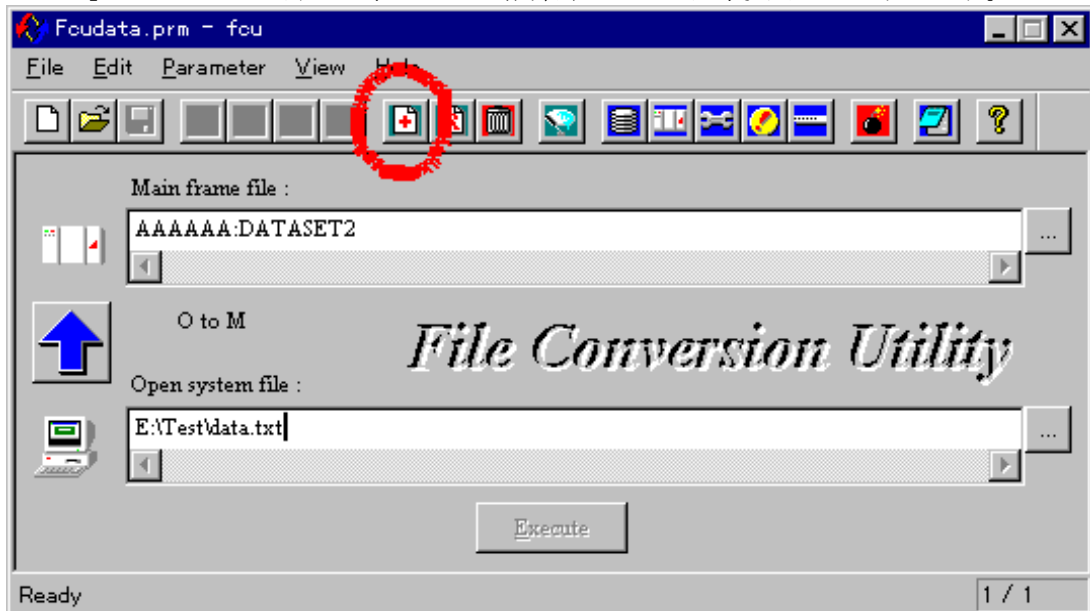


図 8.6.2-23 FCU 本体画面

- 11) ステータスバーが 2 / 2 になり、2 件登録されたことが分かります。

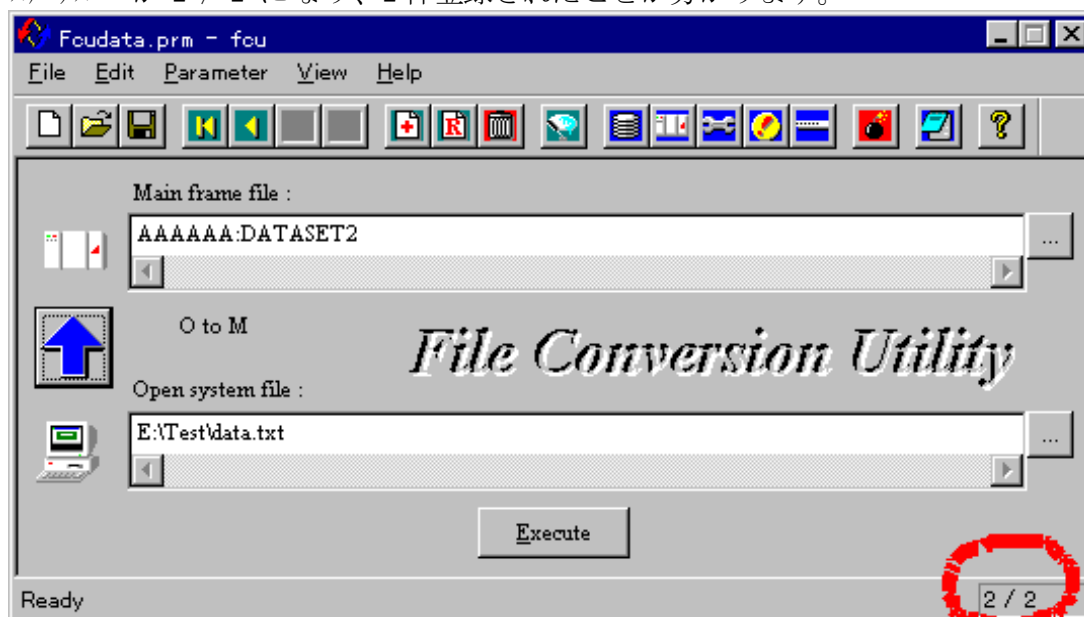


図 8. 6. 2-24 FCU 本体画面

(5) 3 件目のパラメタ行の入力

- 1) 3 件目は、データ転送方向を mto とする為、データ転送方向ボタンをクリックさせ矢印を↓します。データ転送方向が mto になります。

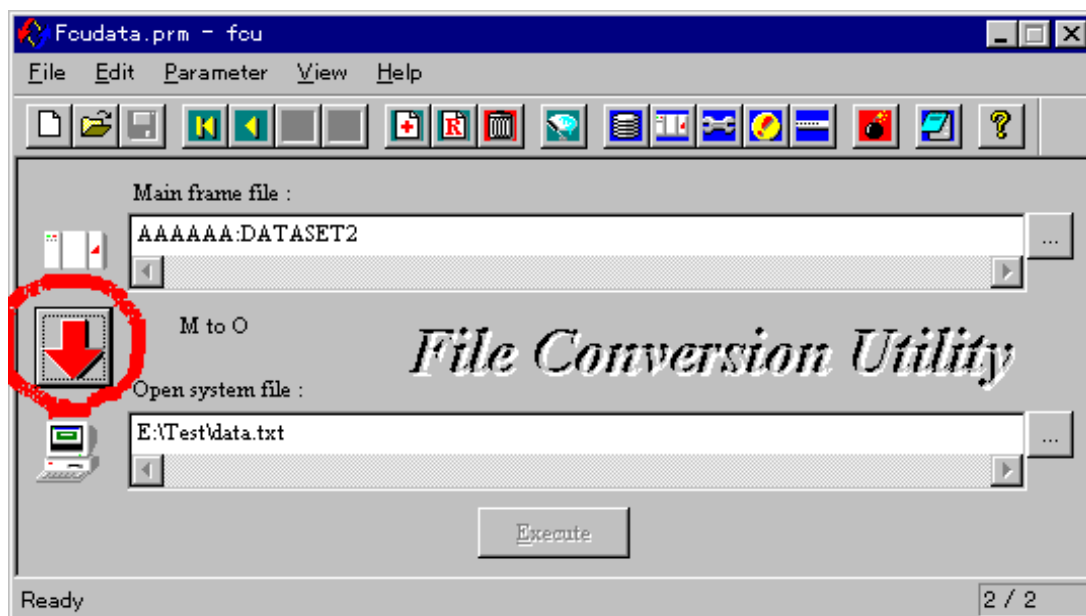


図 8. 6. 2-25 FCU 本体画面

- 2) メインフレームファイル名は、2 件目と同じなので、何もしません。

次にオープンシステムファイル名は、Open system file 欄に直接入力することにします。

(ここで、オープンシステムファイル名が予定と違うのは、後で修正の手順を示すためです。)

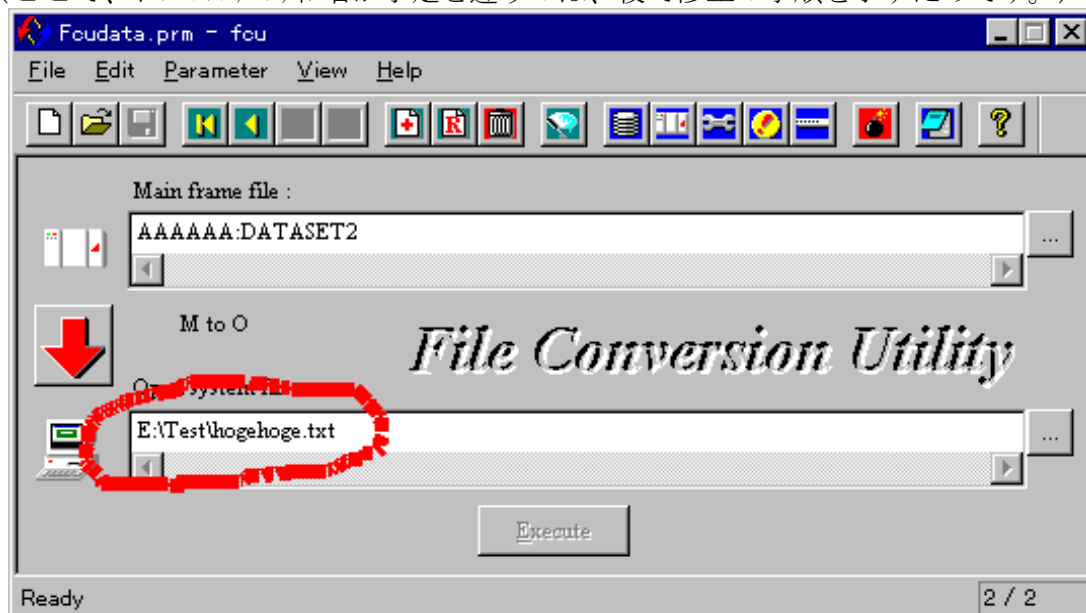


図 8. 6. 2-26 FCU 本体画面

3) オプションダイアログの Padding をチェックします。

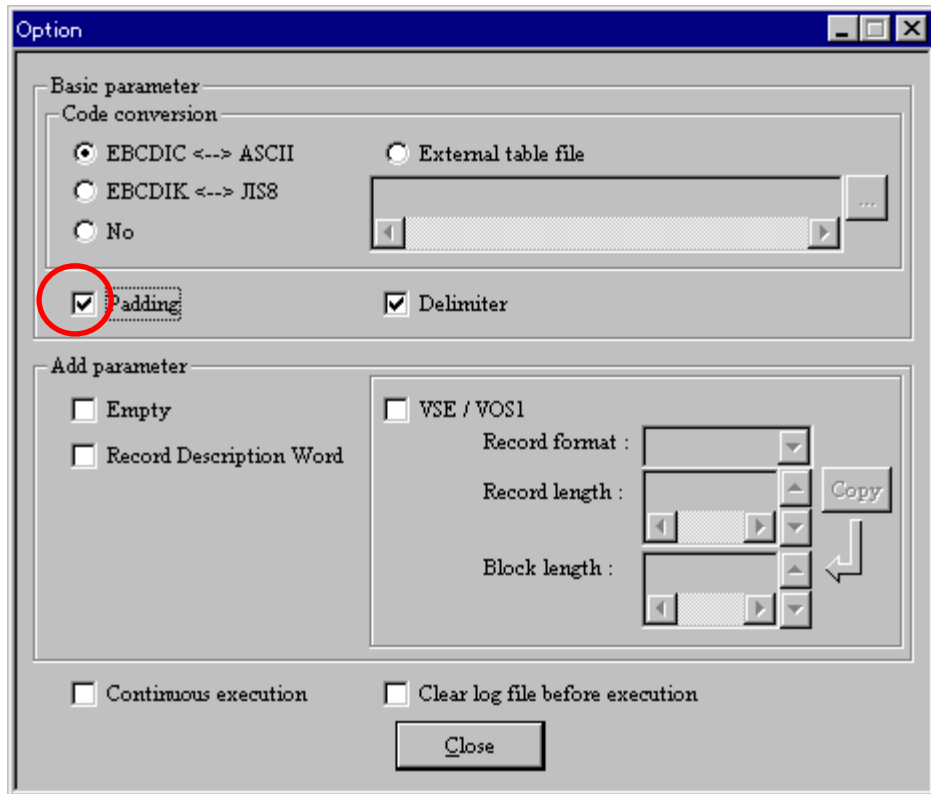



図 8.6.2-27 Option ダイアログ

4) 「Insert」ボタン  を押して、メモリ上で編集集中のパラメタ定義ファイルに登録します。

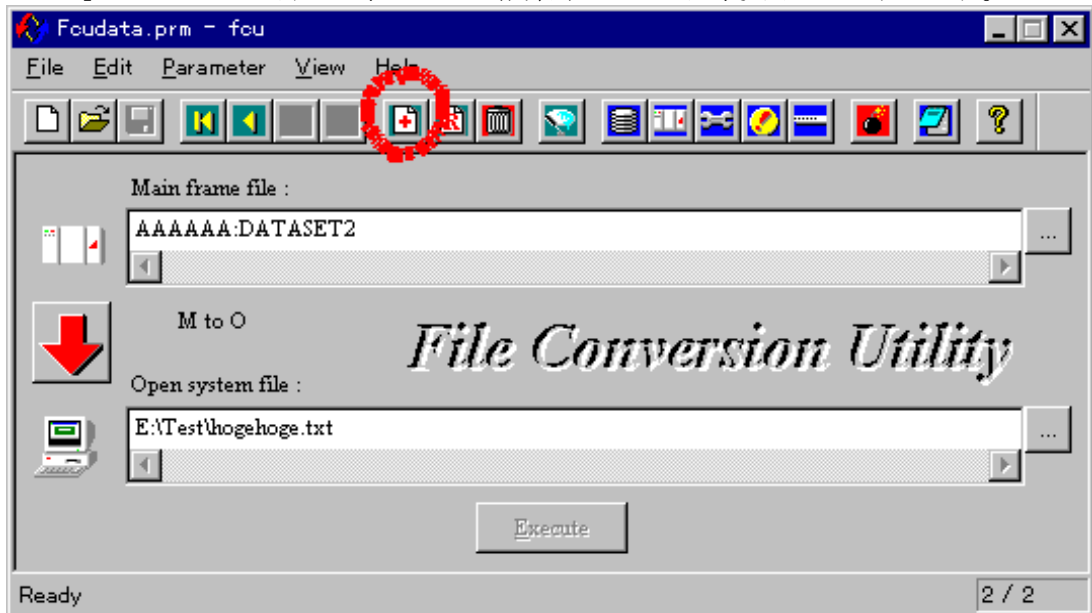


図 8.6.2-28 FCU 本体画面

5) ステータスバーが 3 / 3 になり、3 件登録されたことが分かります。

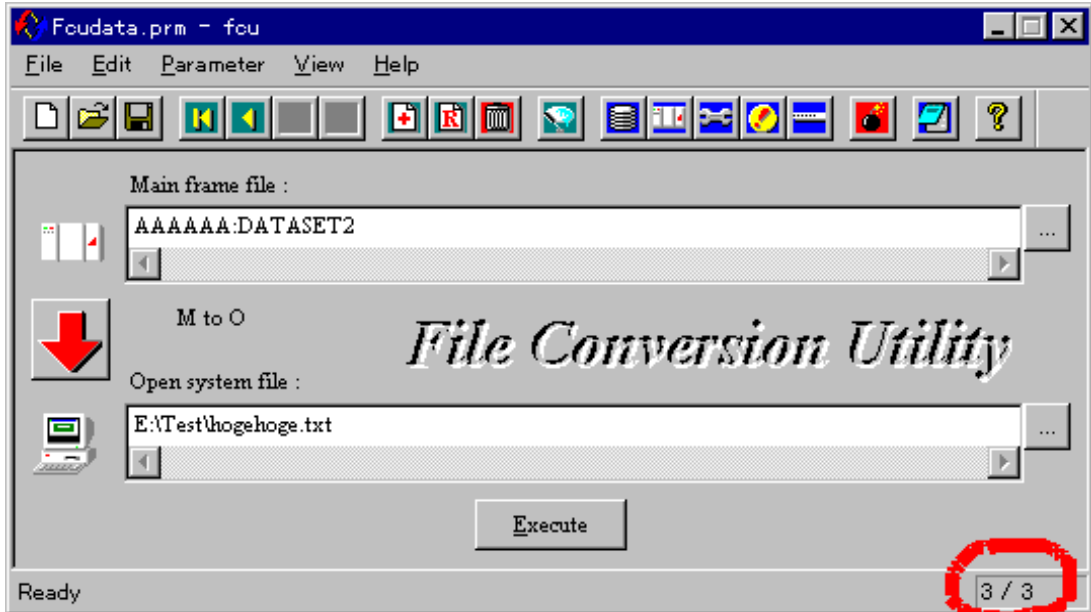


図 8. 6. 2-29 FCU 本体画面

(6) 3 件目のパラメタ行の修正

1) 修正の前に 3 件目のパラメタ行は、どのように記録されているか見てみることにします。


2) Parameter line ボタン  を押します。



図 8. 6. 2-30 FCU 本体画面

3) Parameter line ダイアログが表示されます。

3 件目のパラメタ行は、このように記録されていることが分かります。

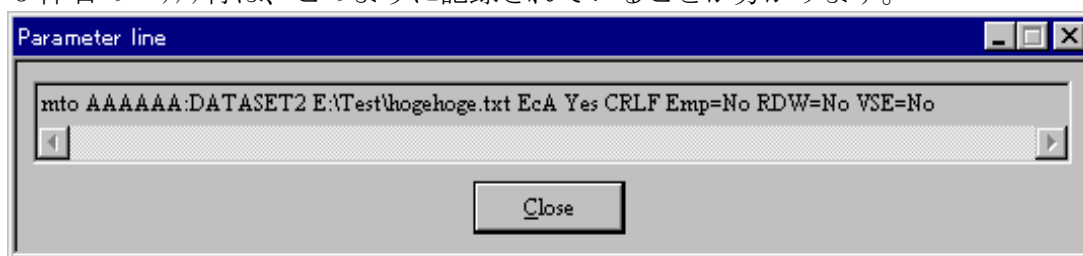


図 8. 6. 2-31 Parameter line ダイアログ

4) オープンシステムファイル名を予定のファイル名に直接修正します。

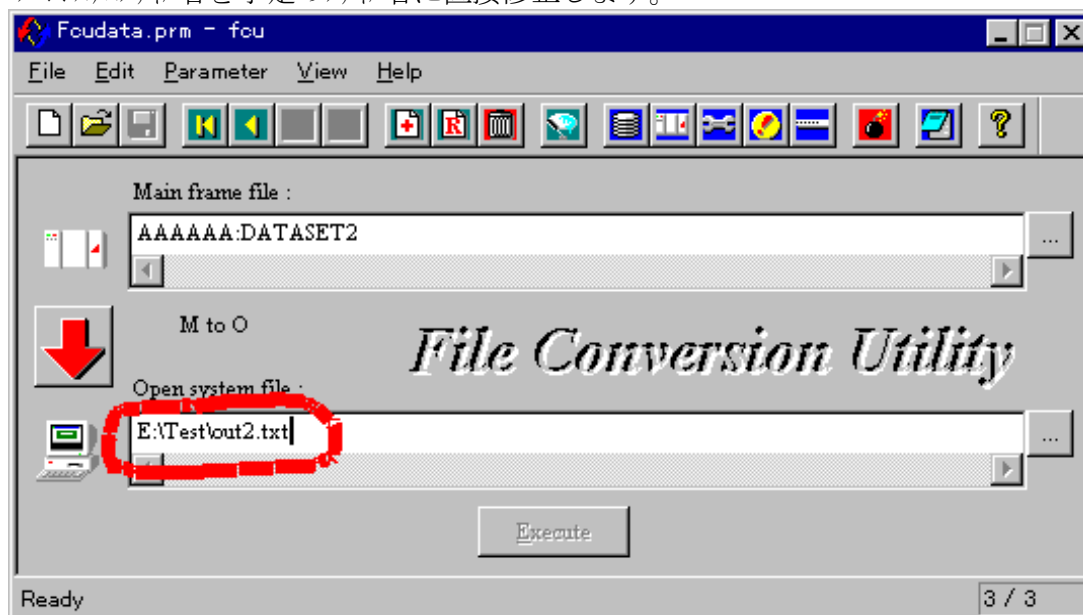

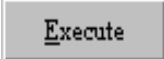


図 8. 6. 2-32 FCU 本体画面

- 5) オープンシステムファイル名を修正したことにより、FCU 本体画面上のボタンの状態に変化が現れています。このように、FCU 本体画面のメインフレームファイル名、又は、オープンシステムファイル名、又は、オプションダイアログの Basic parameter 又は、Add parameter を修正すると、「Save」ボタン  及び「Execute」ボタン  が無効になり選択できなくなります。

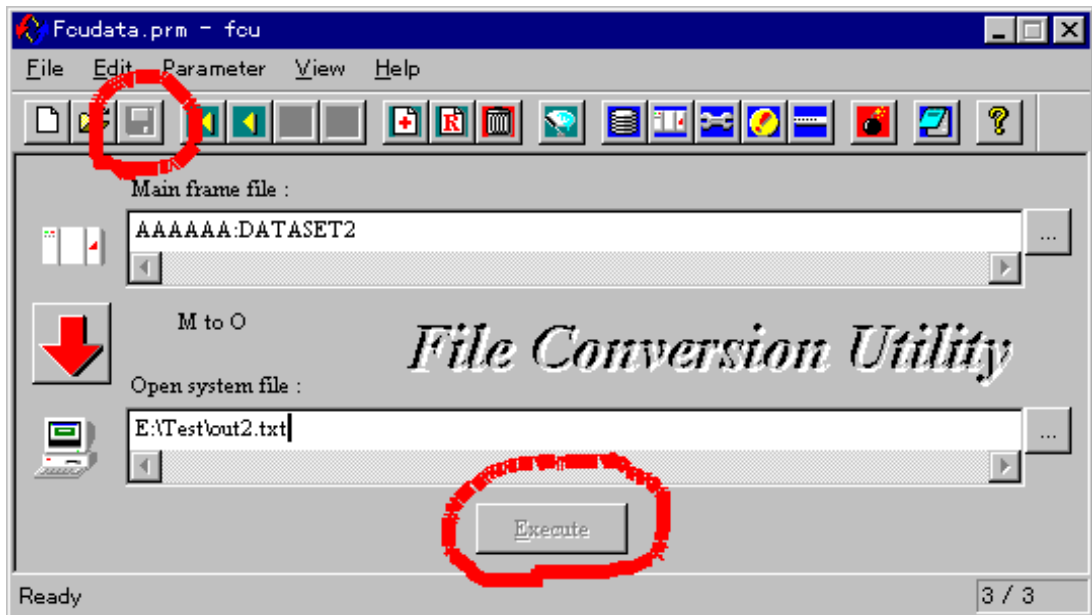


図 8. 6. 2-33 FCU 本体画面

- 6) ここまでの操作で、3 件目のパラメタ行がどのように記録されているかを確認するために、Parameter line ダイアログをしてみます。オープンシステムファイル名の修正前とまだ同じ内容です。

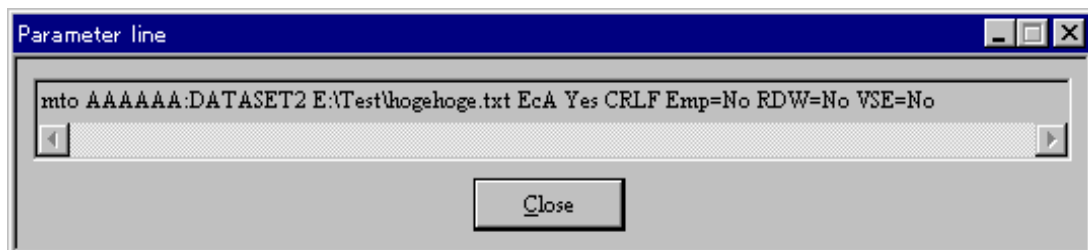





図 8. 6. 2-34 Parameter line ダイアログ

- 7) Parameter line ダイアログに表示される内容を更新するためには、「Insert」ボタン  又は、「Replace」ボタン  を押して(まだ押しません。)、メモリ上で編集集中のパラメタ定義ファイルに登録します。(今回は、3件目の修正なので、「Replace」ボタン  を押す必要があります。(まだ押しません。))しかし、その前にオプションダイアログに加えていた変更を予定通りに修正しておきます。
- 8) Option ダイアログの「Padding」チェックボックスを外します。

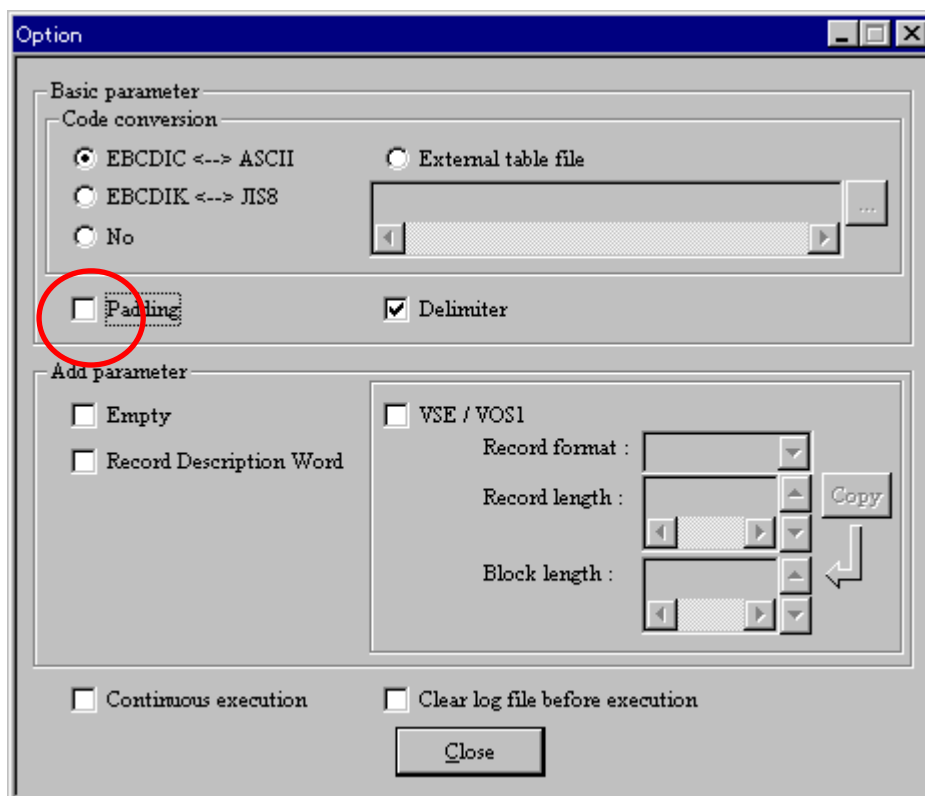



図 8.6.2-35 Option ダイアログ

- 9) 「Replace」ボタン  を押して、メソッド上で編集集中のパラメタ定義ファイルに、現在、画面上で指定されている内容を反映します。

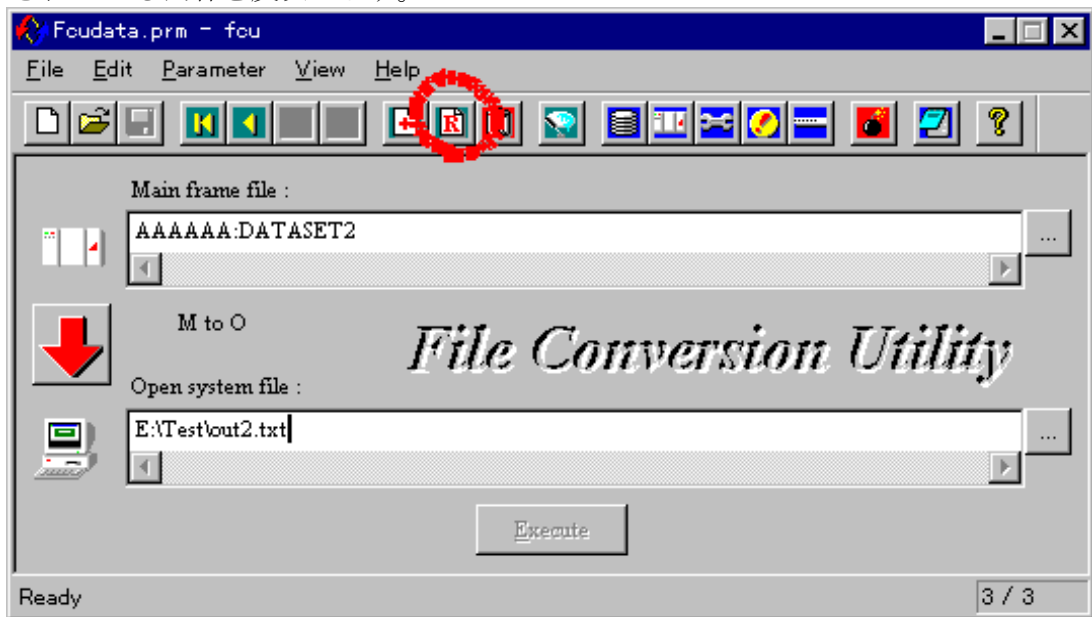


図 8. 6. 2-36 FCU 本体画面

- 10) FCU 本体画面の「Save」ボタン  及び、「Execute」ボタン  が有効になりました。

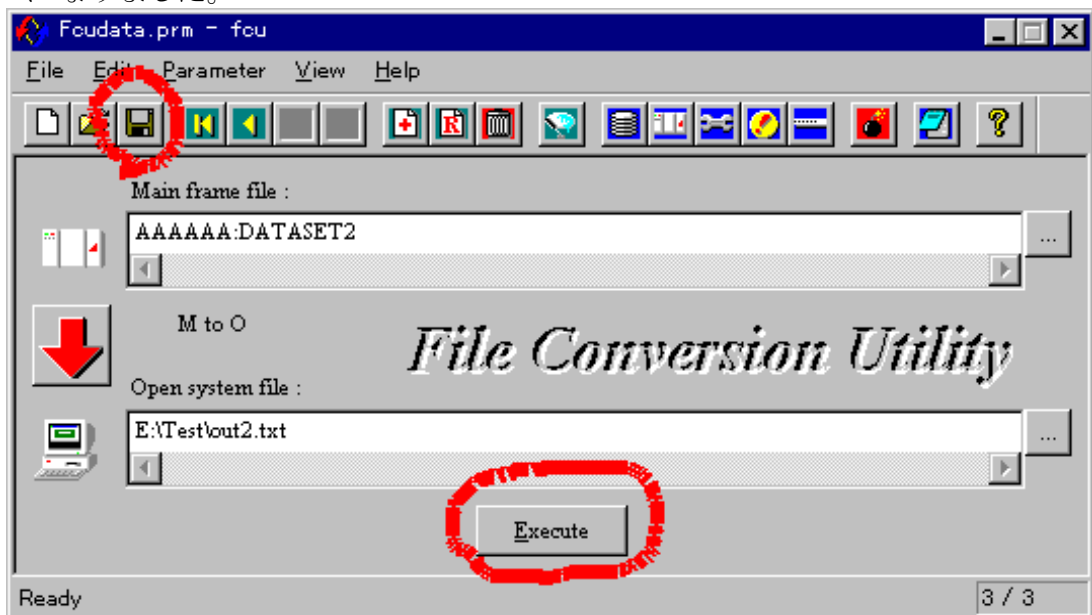
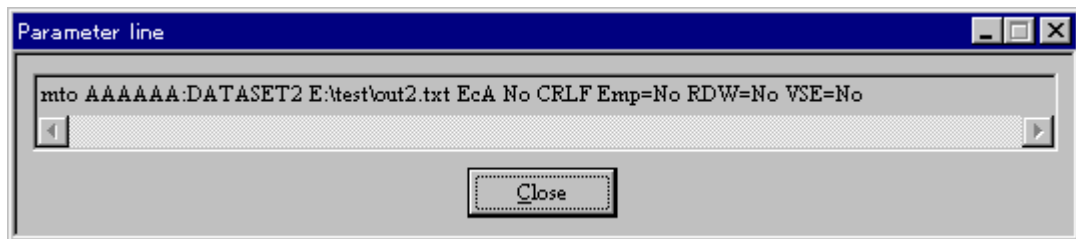


図 8. 6. 2-37 FCU 本体画面


11) Parameter line ダイアログを見てみましょう。



修正した内容に更新されていることが分かります。

図 8.6.2-38 Parameter line ダイアログ

(7) ディスク上のパラメタ定義ファイルへの保存

1) 「Save」ボタン  を押します。ここからの操作により初めて、メモリ上で編集中的パラメタ定義ファイルの内容が、カレントディレクトリの「Foudata.prm」ファイルに保存されます。

(この操作を行わなければ FCU の終了時、パソコンメモリ上で編集した結果が一時的なものとして全て破棄されます。)

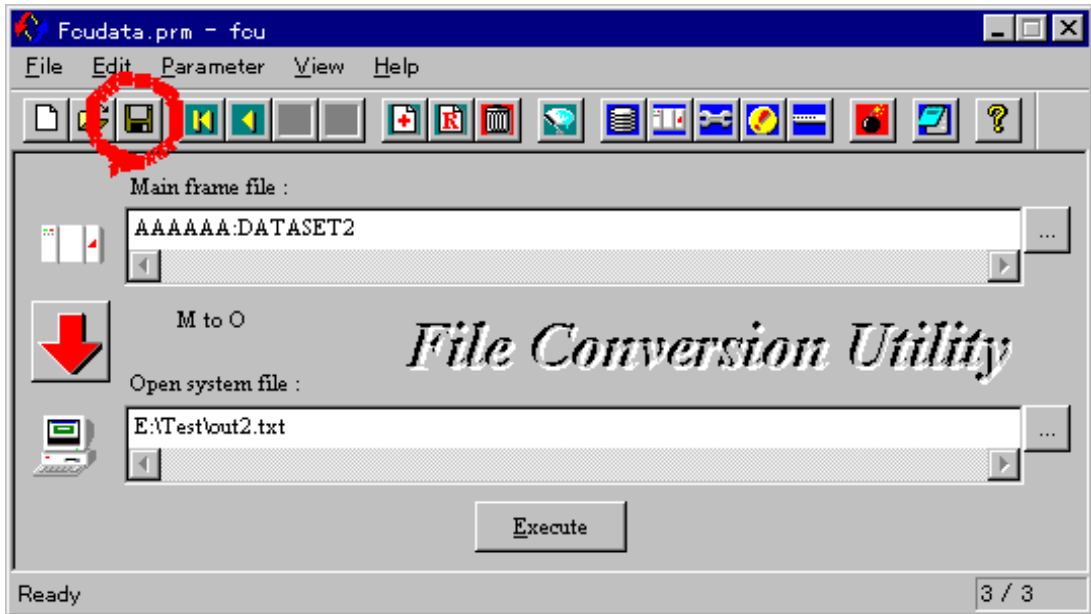


図 8. 6. 2-39 FCU 本体画面

2) 他の名前で、保存する場合、[File]-[Save As]メニューを選択してください。「名前を付けて保存」ダイアログが表示されます。



図 8. 6. 2-40 名前を付けて保存ダイアログ

3) ファイル名に保存するパラメ定義ファイル名 (Test.prm) を入力し、保存ボタンを押します。

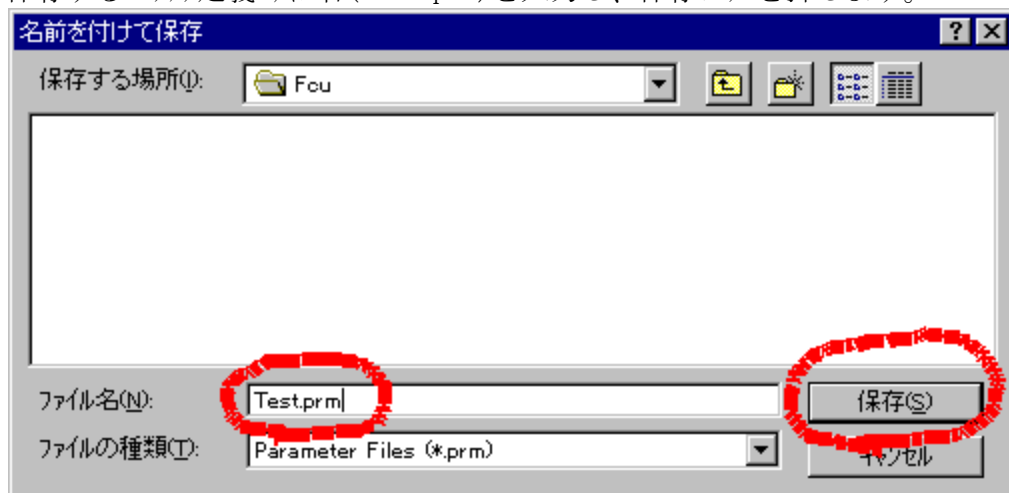


図 8. 6. 2-41 名前を付けて保存ダイアログ

4) ディスク上のパラメ定義ファイルに保存されました。
(タイトルバーの文字が、Test.prm に変わりました。)

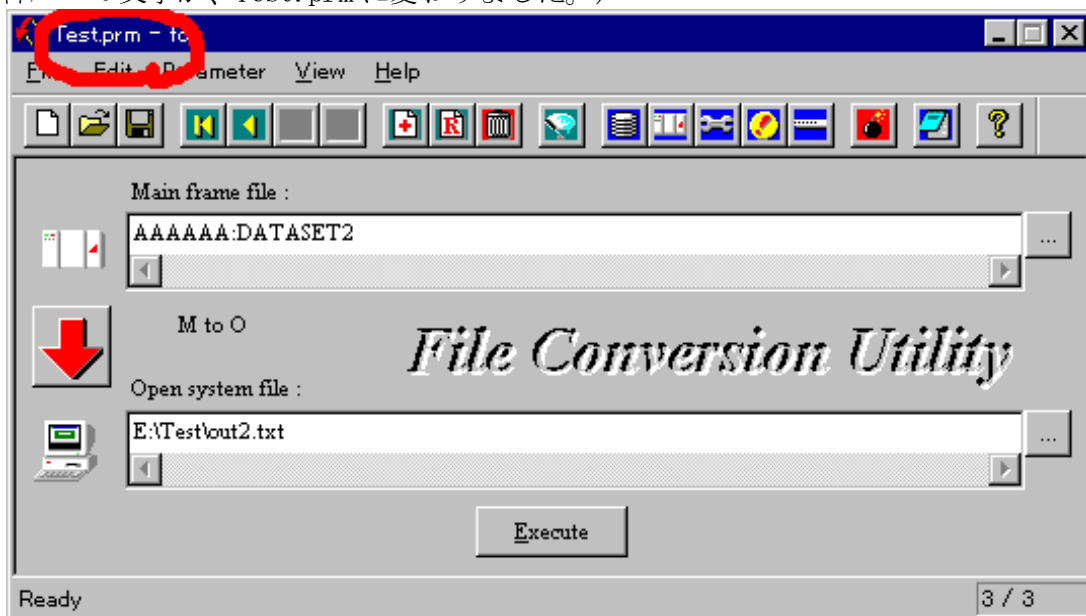



図 8. 6. 2-42 FCU 本体画面

5) ディスク上のパラメタ定義ファイルを確認します。

```
mto AAAAAA:DATASET1 E:¥test¥out1. txt EcA No CRLF Emp=No RDW=No VSE=No  
otm E:¥test¥data. txt AAAAAA:DATASET2 EcA No CRLF Emp=No RDW=No VSE=No  
mto AAAAAA:DATASET2 E:¥test¥out2. txt EcA No CRLF Emp=No RDW=No VSE=No
```

図 8. 6. 2-43 パラメタ定義ファイル

(8) ディスク上の既存のパラメタ定義ファイルを開く

- 1) 既存のパラメタ定義ファイルを開く手順を示すために、ここで一度「New」ボタン  を押します。

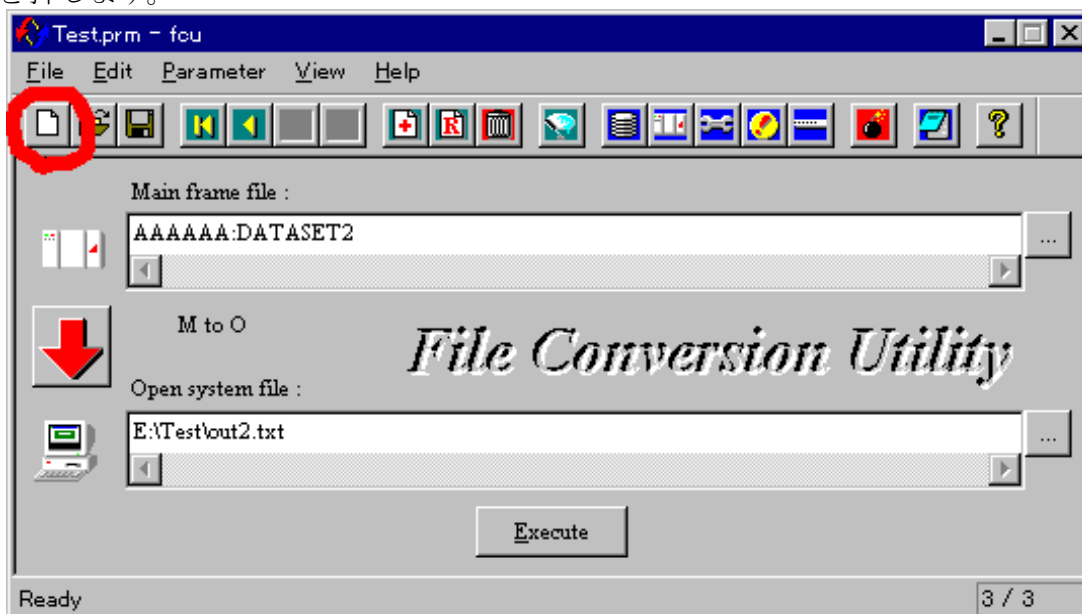


図 8. 6. 2-44 FCU 本体画面

2) 新規作成状態になりました。

(タイトルバーとツールバーの表示からも新規であることが分かります。)

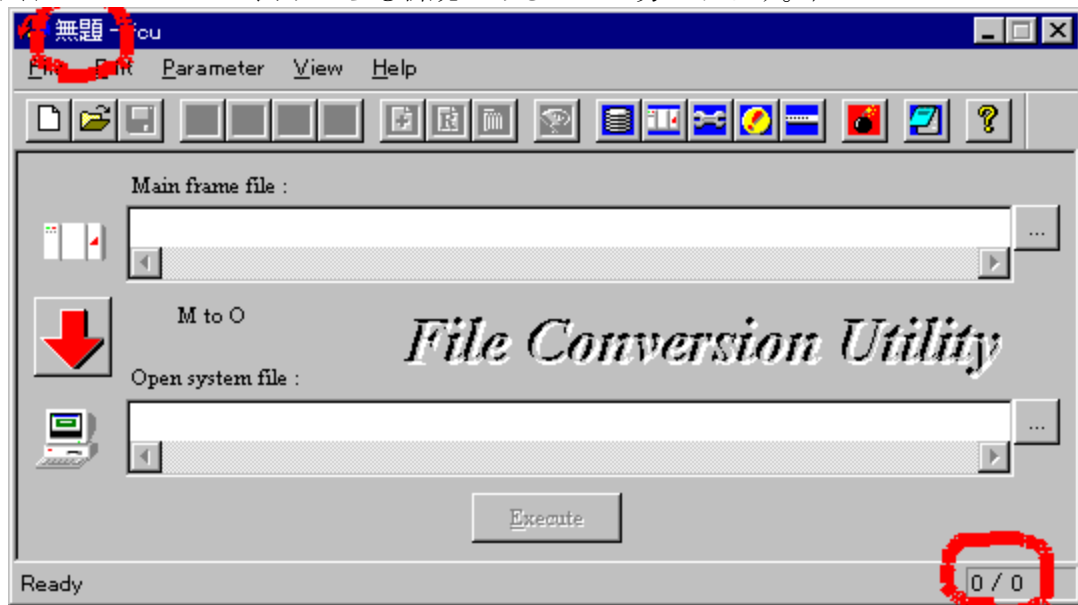


図 8. 6. 2-45 FCU 本体画面

3) 「Open」ボタンを押します。

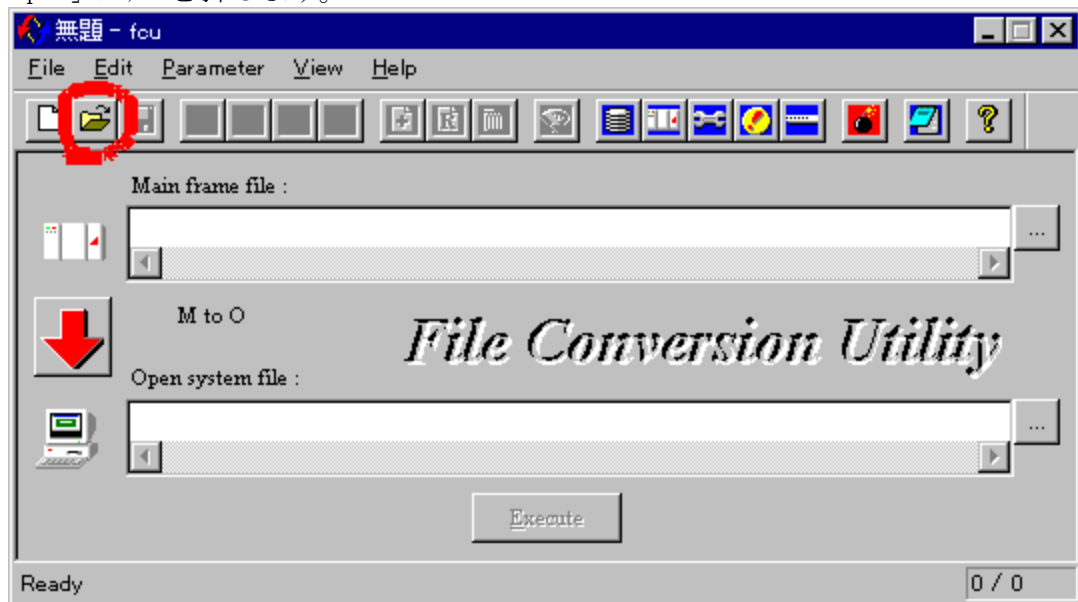


図 8. 6. 2-46 FCU 本体画面

4) 開くダイアログが表示されます。

次に既存のパラメタ定義ファイル(Test.prm)をダブルクリックします。

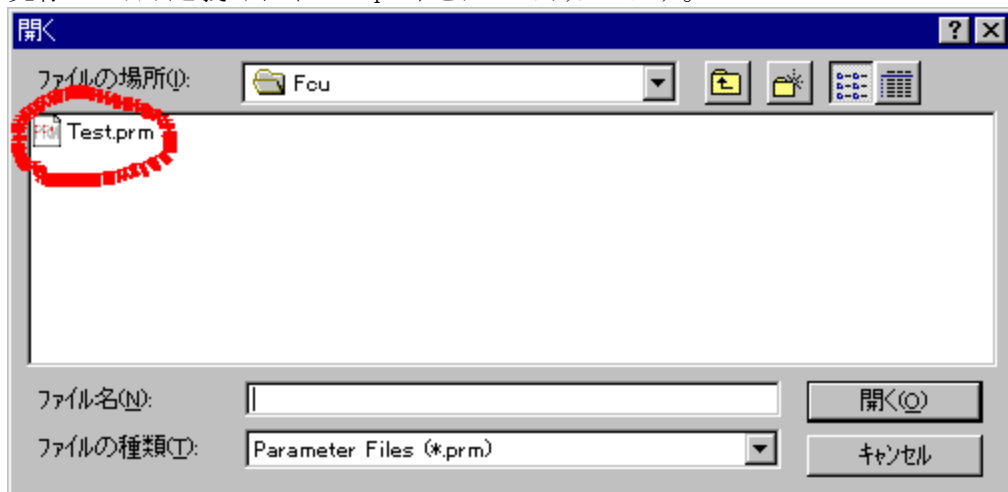


図 8. 6. 2-47 開くダイアログ

5) 既存のパラメタ定義ファイルが読み込まれました。

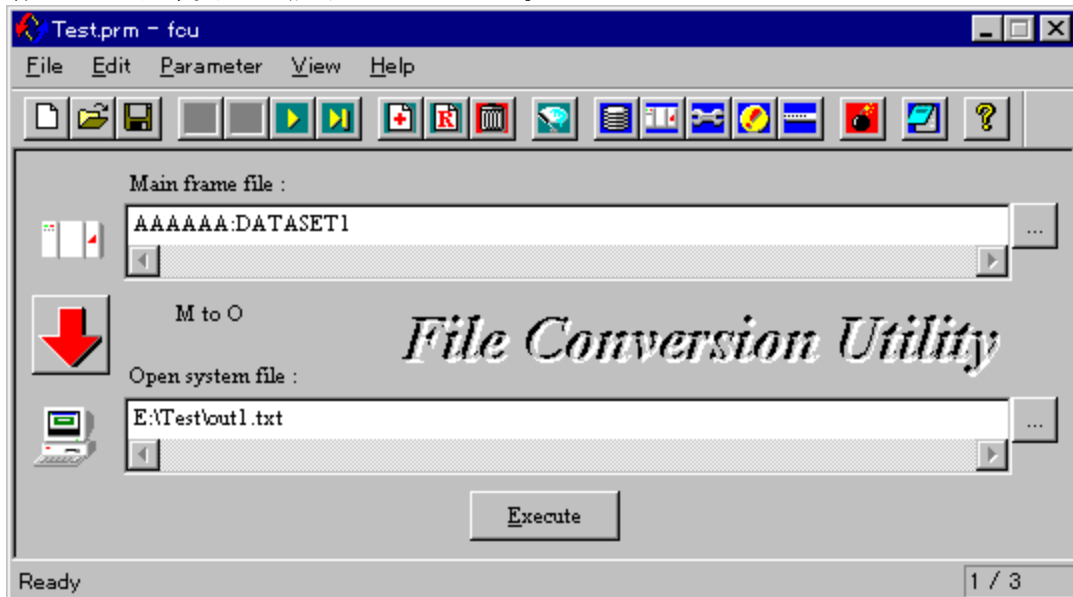


図 8. 6. 2-48 FCU 本体画面

(9) 1 件目のパラメタ行の実行 (単独実行)

(これは、例です。1 件目の実行を単独実行で行わなければならない訳ではありません。)

- 1) ログファイルを見易くするために、Option ダイアログの「Clear log file before execution」チェックボックスをチェックしておきます。

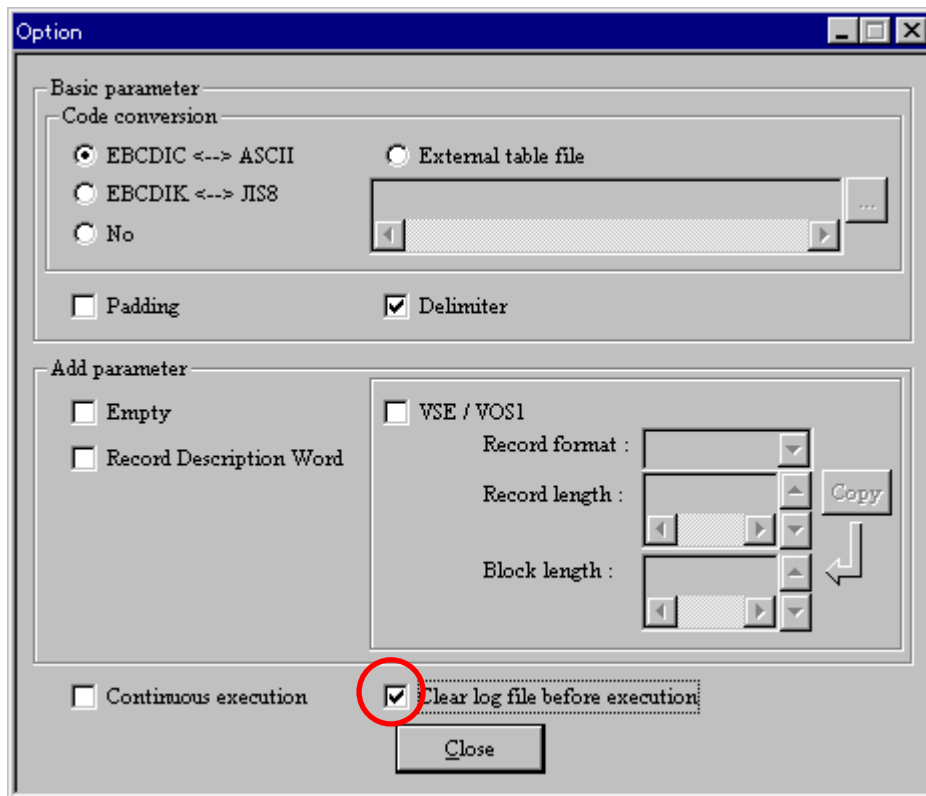


図 8.6.2-49 Option ダイアログ

- 2) 連続実行 (Continuous execution) 時、途中結果 (個々の正常/異常) を目視確認しながら作業したい場合は、Error Information ダイアログを予め開いて、画面上 FCU 本体画面と重ならない見易い位置に配置してください。

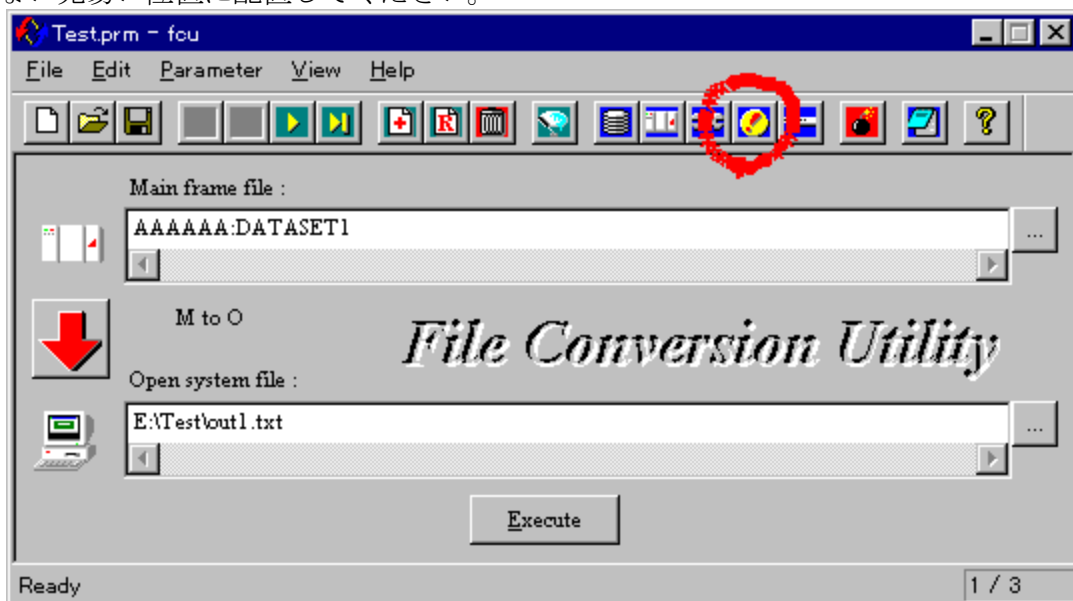


図 8.6.2-50 FCU 本体画面

3) Error information ダイアログを表示しました。

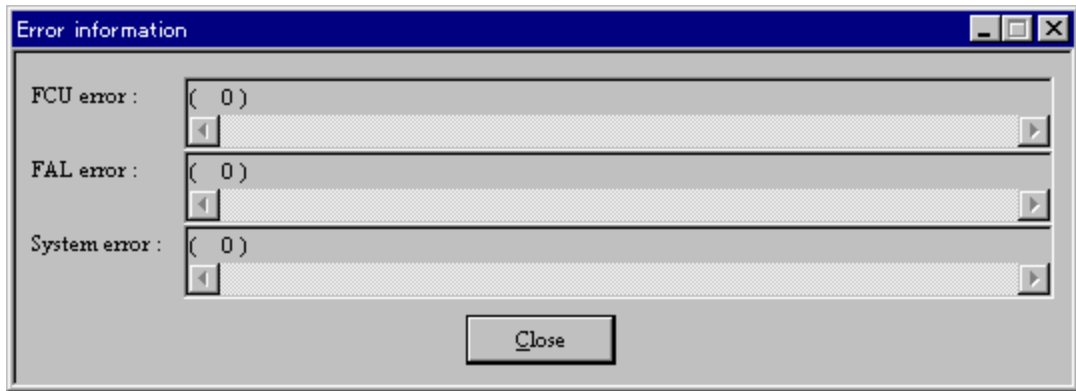
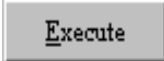


図 8. 6. 2-51 Error Information ダイアログ

4) 「Execute」ボタン  を押します。

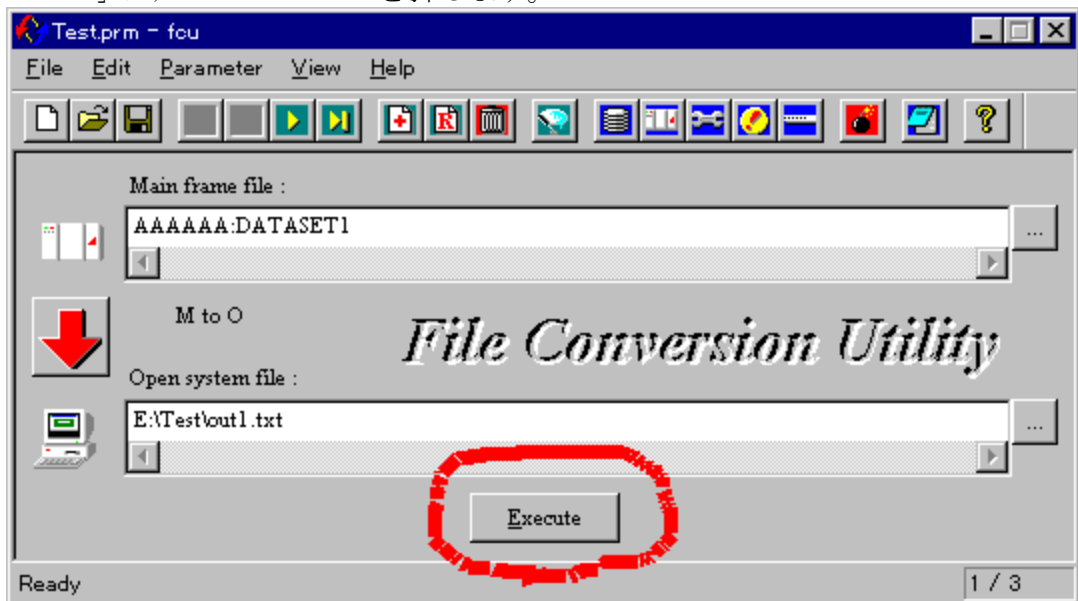


図 8. 6. 2-52 FCU 本体画面

5) 「Execute」ダイアログが表示され、進行状態を表示します。

(ただし、VSEパラメタの指定を行うと、MTO時は進行状態の表示を行いません。)

正常終了すると、下図の状態となります。

単独実行の場合、「Execute」ダイアログを表示したままの状態を終了します。

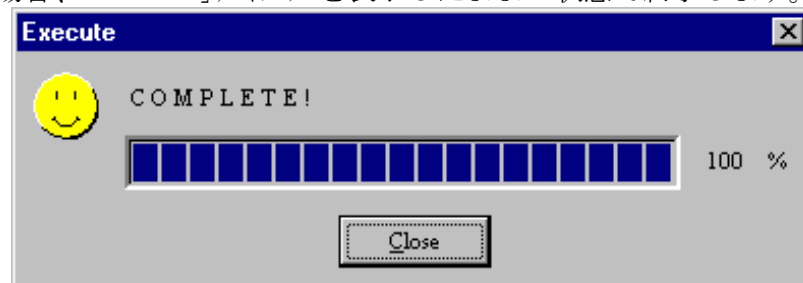


図 8. 6. 2-53 Execute ダイアログ

6) 「Log」ボタン  を押して、ログファイルを見てみます。

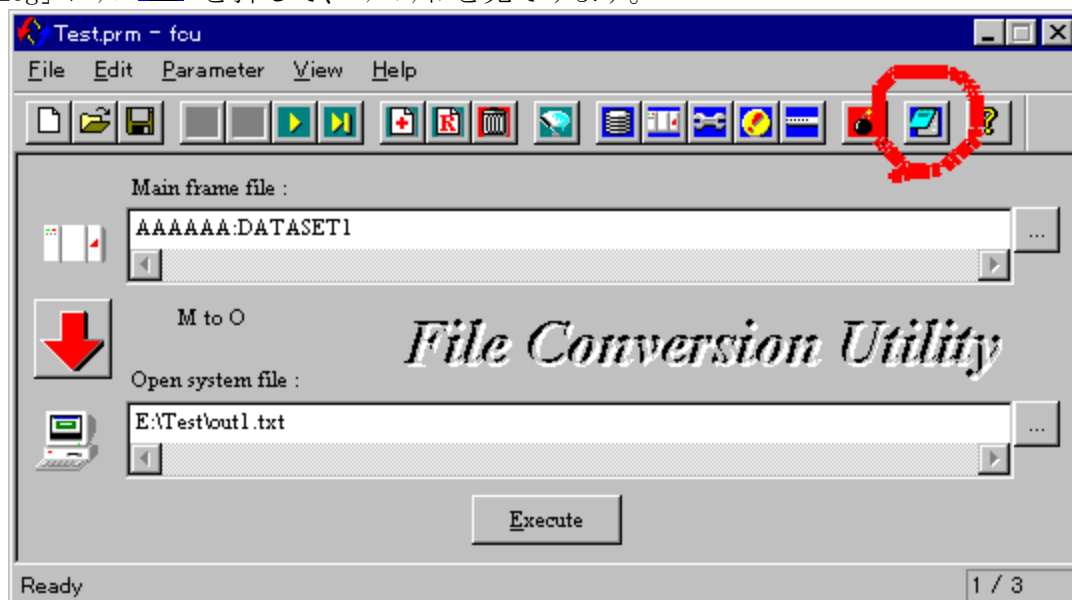
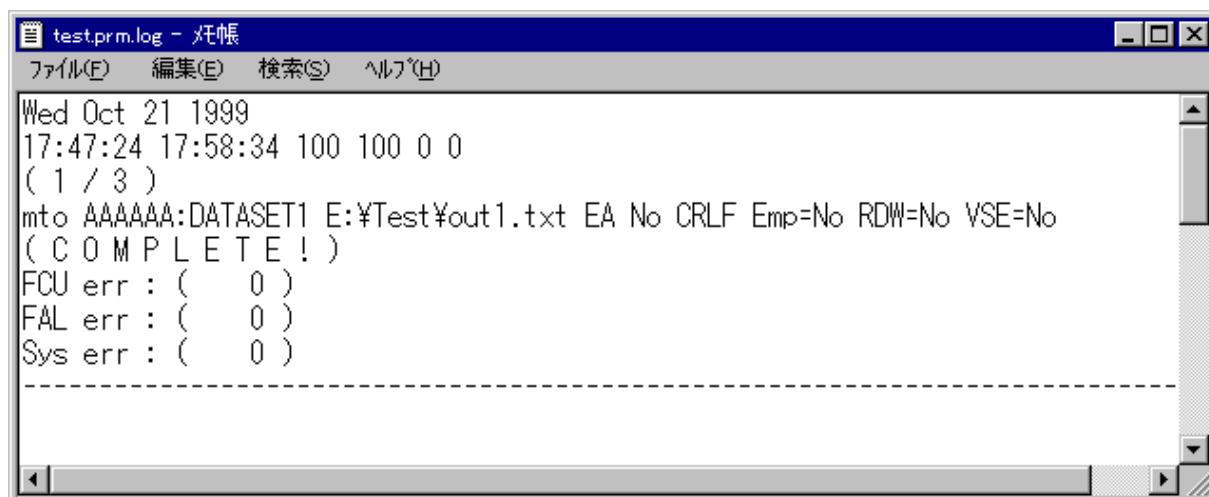


図 8. 6. 2-54 FCU 本体画面



7) ログファイルが表示されました。

図 8. 6. 2-55 ログファイル

(10) 2~3件目のパラメタ行の実行(連続実行)

(これは、例です。2~3件目の実行を連続実行で行わなければならない訳ではありません。)


1) 「Next」ボタン  を押し、2件目のパラメタ行をロードします。



図 8. 6. 2-56 FCU 本体画面

2) 2件目のパラメタ行をロードしました。

(ステータスバーの右下の表示でも、2件目がロードされたことが分かります。)

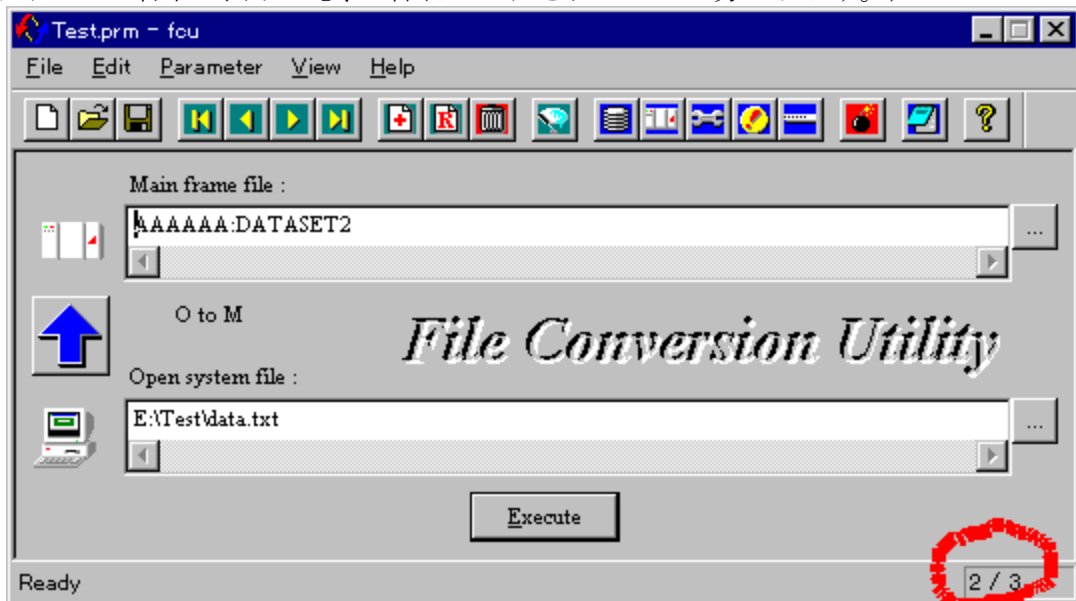


図 8. 6. 2-57 FCU 本体画面

3) 連続実行させるため、Option ダイアログの「Continuous execution」チェックボックスを
チェックします。

次に1件目のログを削除したくないため、Option ダイアログの
「Clear log file before execution」チェックボックスのチェックを外します。

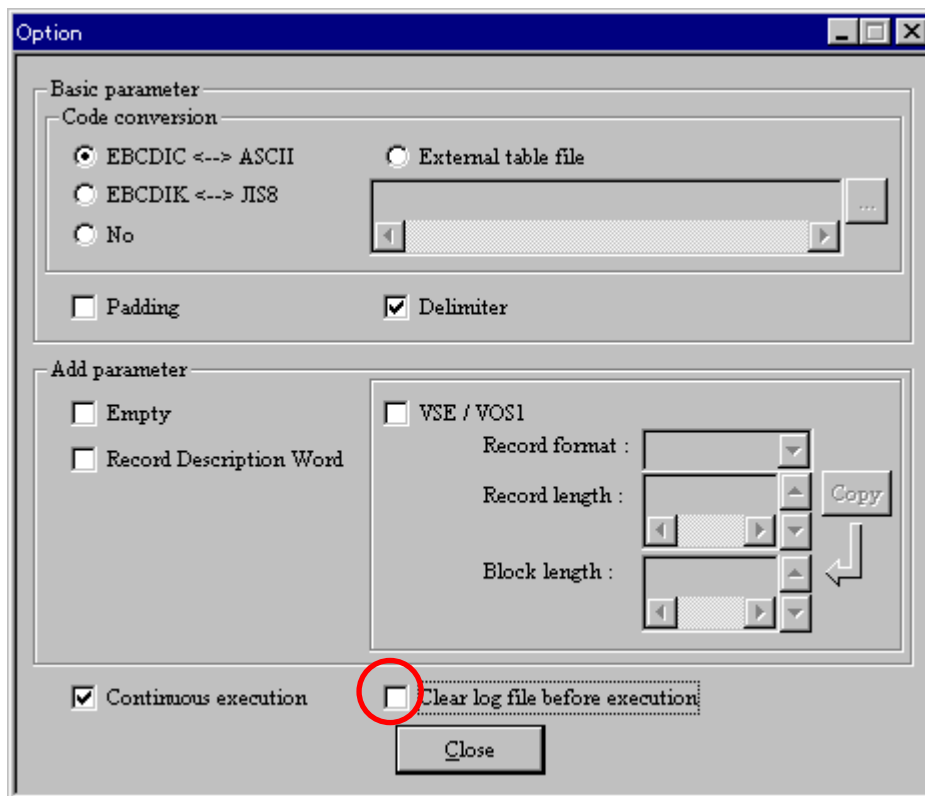


図 8.6.2-58 Option ダイアログ

- 4) 「Execute」ボタン  を押します。

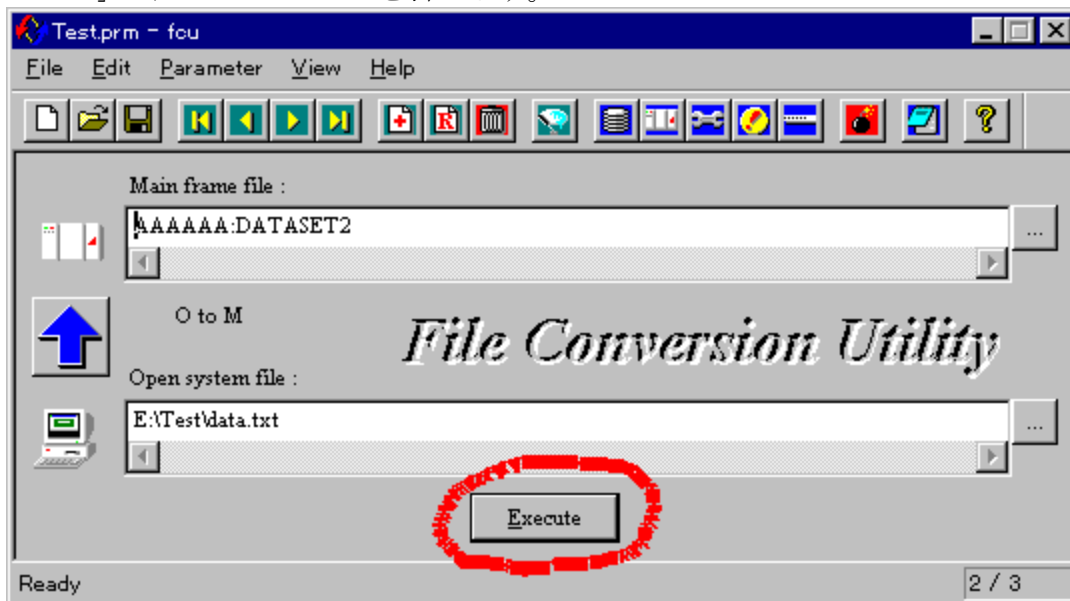


図 8. 6. 2-59 FCU 本体画面

- 5) 2～3 件目の処理毎に、「Execute」ダイアログが表示されます。
(これは、正常終了した場合です。)

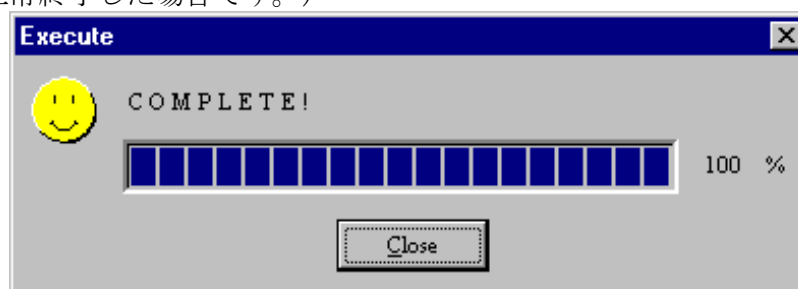


図 8. 6. 2-60 Execute ダイアログ

- 6) 連続実行が終了しました。
連続実行の場合、「Execute」ダイアログを閉じた状態で終了します。

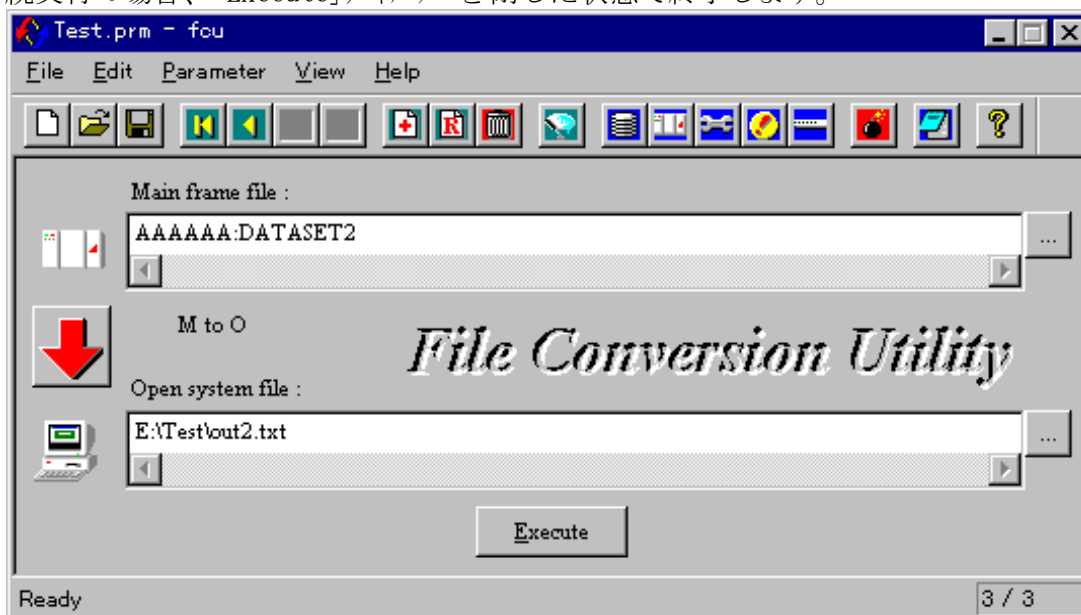


図 8. 6. 2-61 FCU 本体画面

7) 「Log」ボタン  を押して、ログファイルを見てみます。

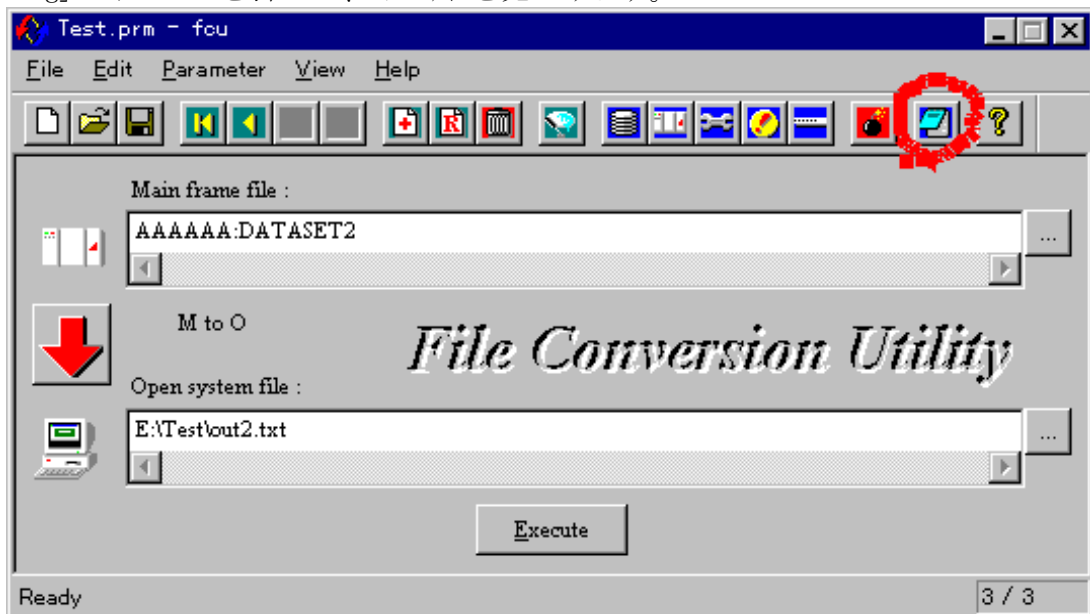


図 8. 6. 2-62 FCU 本体画面

8) ログファイルが表示されました。

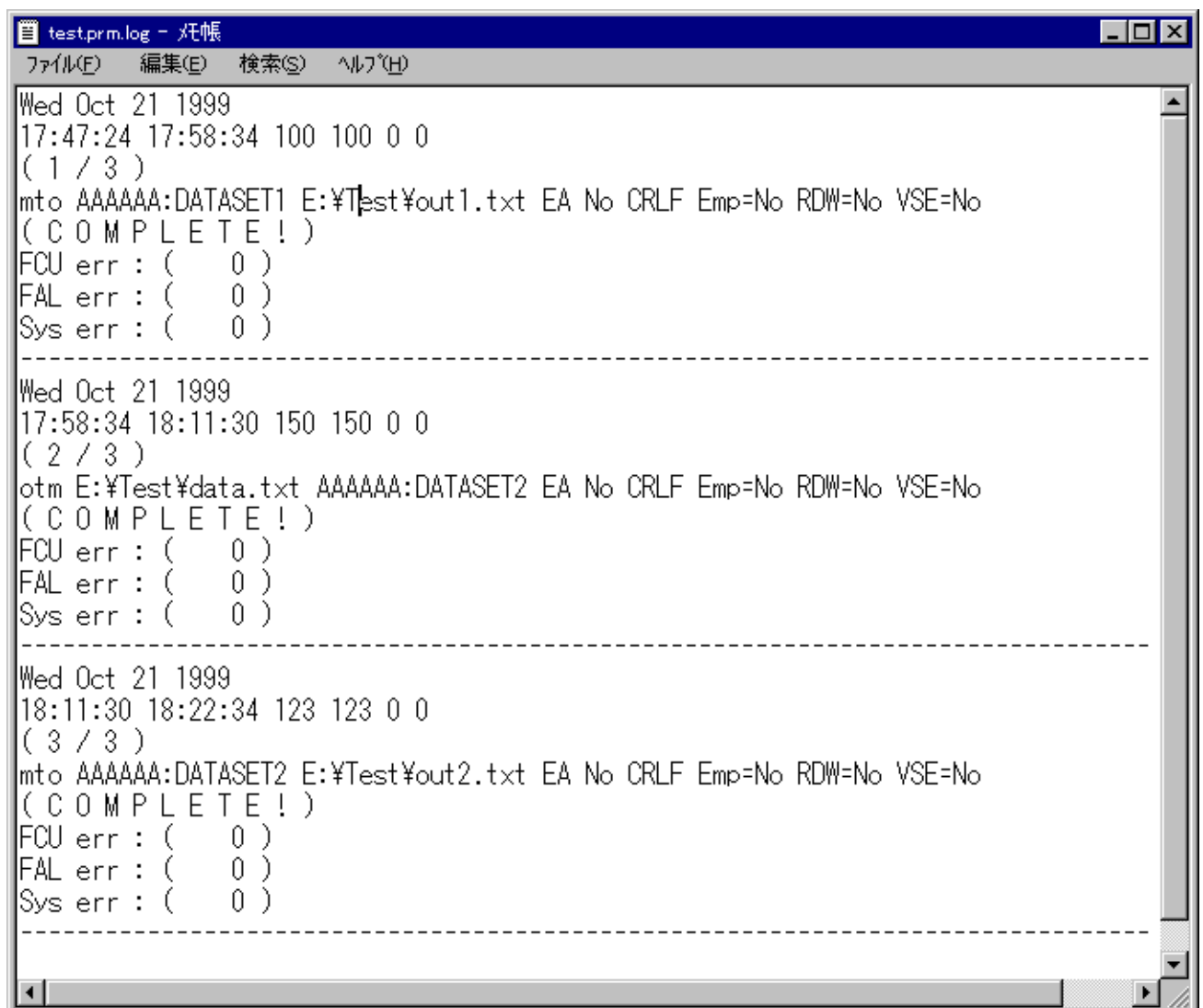


図 8. 6. 2-63 ログファイル

(1 1) FCU の終了

1) クロスボタンをクリックします。

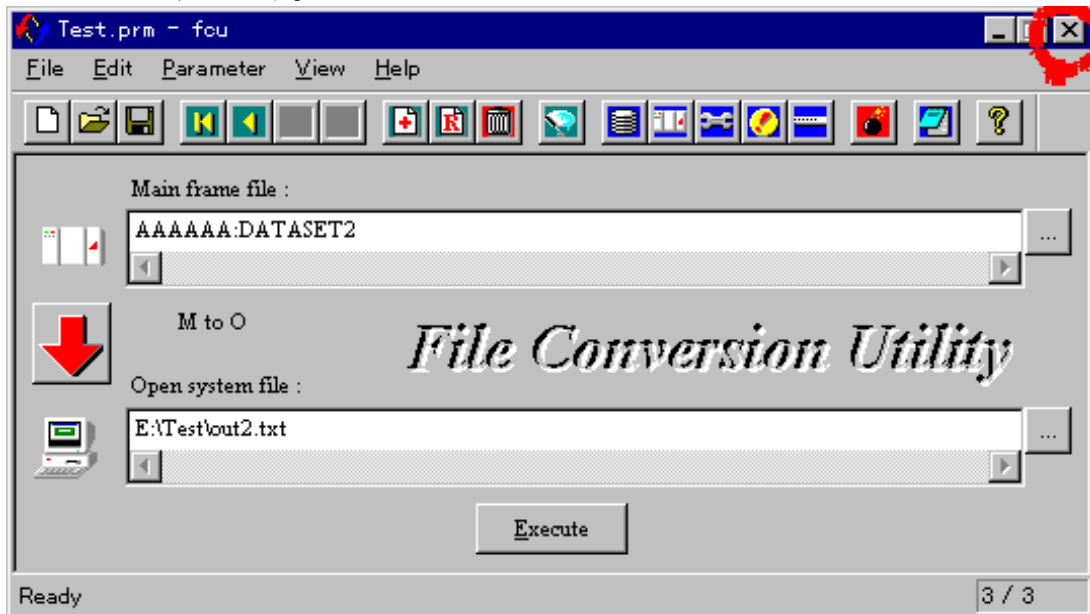


図 8. 6. 2-64 FCU 本体画面

2) FCU を終了します。

以上で、全ての操作が完了しました。

8.6.2.2 画面機能詳細

ここでは、各画面に設けられたメニューやボタンの機能を説明します。

起動すると最初にスプラッシュ画面、次に **FCU** 本体画面が表示されます。
(スプラッシュ画面は、一定時間が経過すると自動的にクローズします。)



図 8. 6. 2-65 スプラッシュ画面

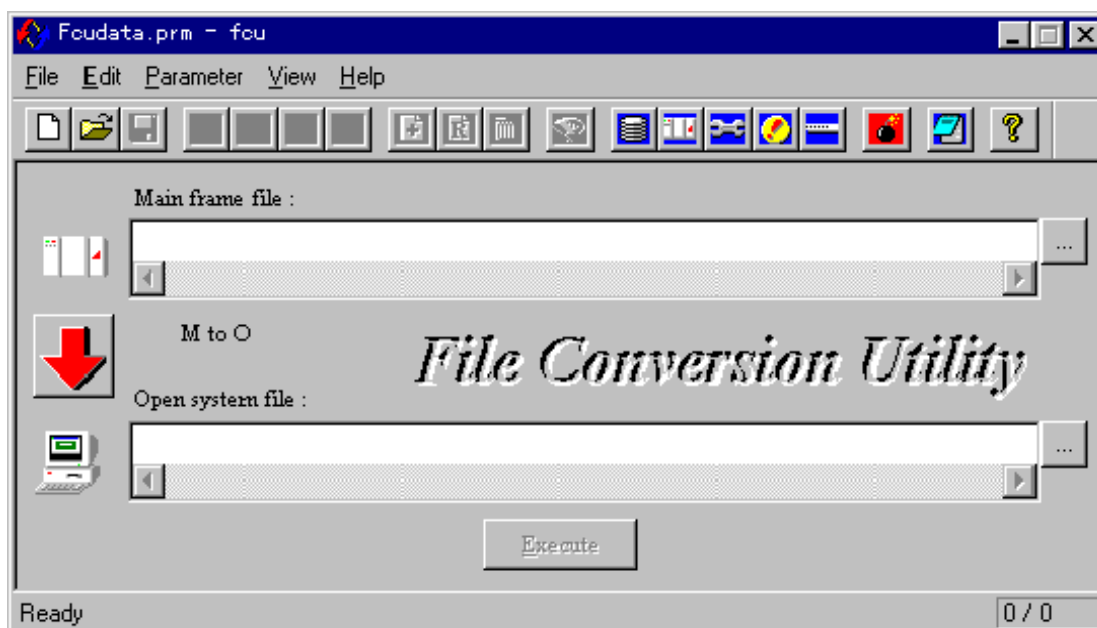


図 8. 6. 2-66 **FCU** 本体画面

(注) **FCU** 画面中の文字は英字フォントを使用しているため、日本語が表示できません。
オープンシステムファイルに指定するフォルダ及び、ファイル名に日本語を使用しないで下さい。
又、「¥」は、「\」と表示されます。

8.6.2.2.1 File メニュー

パラメタ定義ファイルの操作、FCU の終了を指示する場合に選択します。

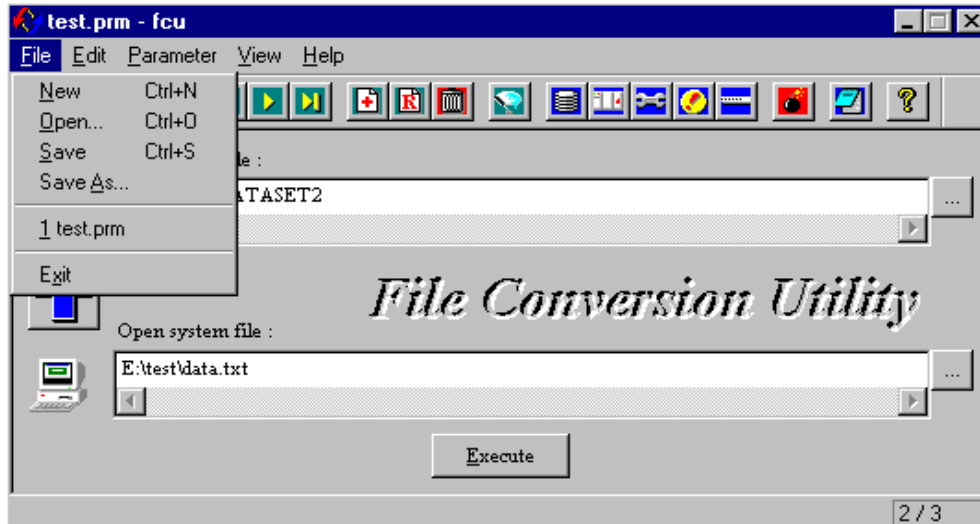


図 8.6.2-67 File メニュー

(1) New (Ctrl + N)

新規にパラメタ定義ファイルを作成する場合に選択します。

(注) 「Save」を行うまでパラメタ定義ファイルは、メモリ上に編集されるだけで、ディスク上のファイルには反映されません。

(2) Open (Ctrl + O)

既存のパラメタ定義ファイルを開く場合に選択します。

(3) Save (Ctrl + S)

メモリ上に編集中のパラメタ定義ファイルをディスクへ保存する場合に選択します。

(注) 現在位置のパラメタ行は、パラメタ行ダイアログに表示されている内容が保存されます。

FCU 本体画面及び、オプションダイアログのパラメタは、単に変更しただけではパラメタ行ダイアログに反映されません。

パラメタ行ダイアログに反映させるためには、「Insert」又は、「Replace」ボタン(又は、メニュー)を選択する必要があります。

パラメタ行ダイアログに反映した場合は、「Save」が有効になります。

Save 時、パラメタ定義ファイルの「end」文以降は削除されます。

(4) Save As...

ファイル参照ダイアログを使って、メモリ上に編集中のパラメタ定義ファイルを別名でディスク保存する場合に選択します。

(5) Exit

FCU を終了する場合に選択します。

8.6.2.2.2 Edit メニュー

将来の拡張用で、現在使用していません。

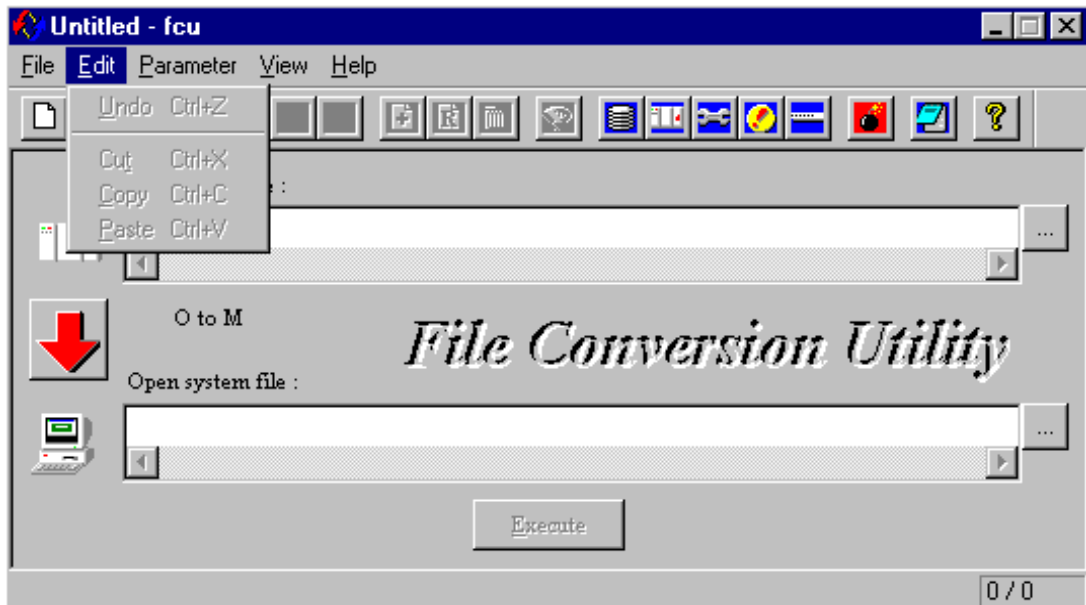


図 8.6.2-68 Edit メニュー

8.6.2.2.3 Parameter メニュー

メモリ上で編集集中のパラメタ定義ファイル进行操作したり、画面のクリアを行う場合に選択します。

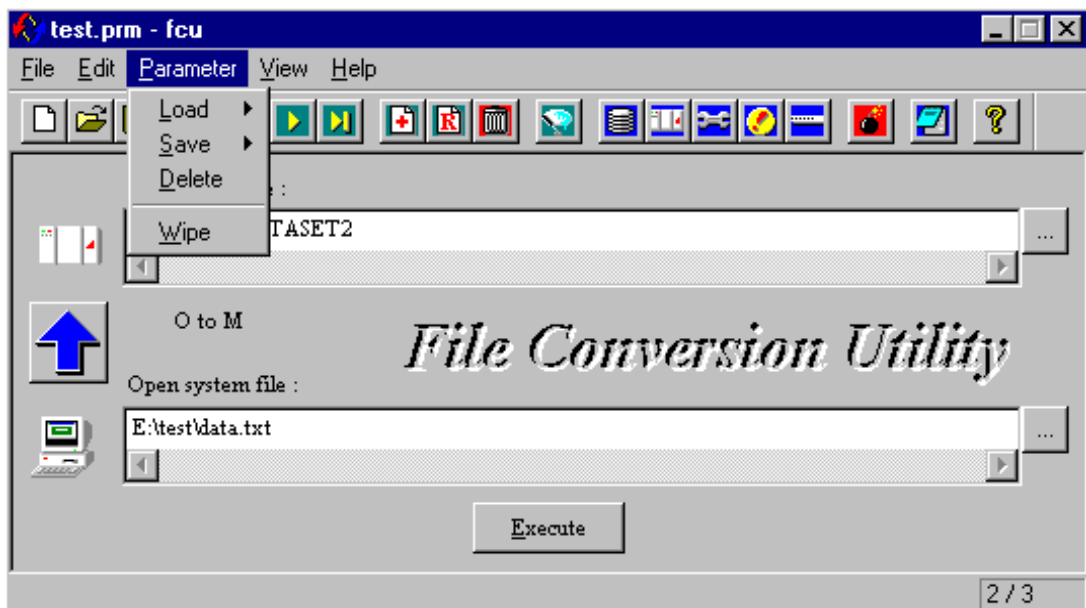


図 8.6.2-69 Parameter メニュー

(1) Load

メモリ上で編集集中のパラメタ定義ファイルに対する次の操作メニューを表示します。

- ① **Previous** : 現在位置の前のパラメタ行をロードします。
- ② **Next** : 現在位置の次のパラメタ行をロードします。
- ③ **Top** : 先頭位置のパラメタ行をロードします。
- ④ **Bottom** : 最終位置のパラメタ行をロードします。

(注) ××位置とは、パラメタ定義ファイルの何行目かと言うことに相当します。
ステータスバーの右下に表示されている数字がそうです。

(2) Save

メモリ上で編集中のパラメタ定義ファイルに対する次の操作メニューを表示します。

- ① **Insert** : 現在位置のパラメタ行の次に挿入します。
- ② **Replace** : 現在位置のパラメタ行を置換します。

(注) **FCU** 本体画面及び、オプションダイアグのパラメタを変更した場合は、本項目を選択する必要があります。
本項目を選択しないと、「Execute」及び、「Save」ボタンが有効になりません。

(3) Delete

メモリ上で編集中のパラメタ定義ファイルに対して、現在位置のパラメタを削除する場合に選択します。

(注) 「Save」を行うまでは、パラメタ定義ファイルから実際に削除される訳ではありません。

(4) Wipe

画面に表示中のパラメタを再入力し易くするため、一斉にクリアしたい場合に選択します。

(注) 現在位置のパラメタが削除される訳ではありません。

8.6.2.2.4 View メニュー

各情報ダイアグを操作する場合に選択します。

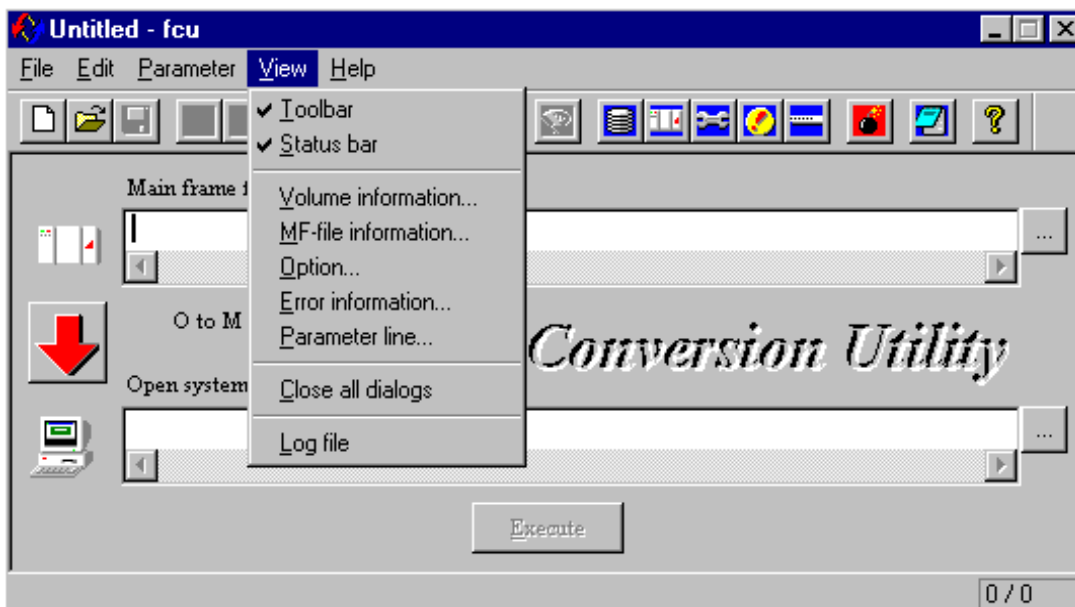


図 8.6.2-70 View メニュー

(1) Toolbar

ツールバーの表示／非表示を切り替えます。

(2) Status Bar

ステータスバーの表示／非表示を切り替えます。

(3) Volume information

ボリューム情報を表示する場合に選択します。

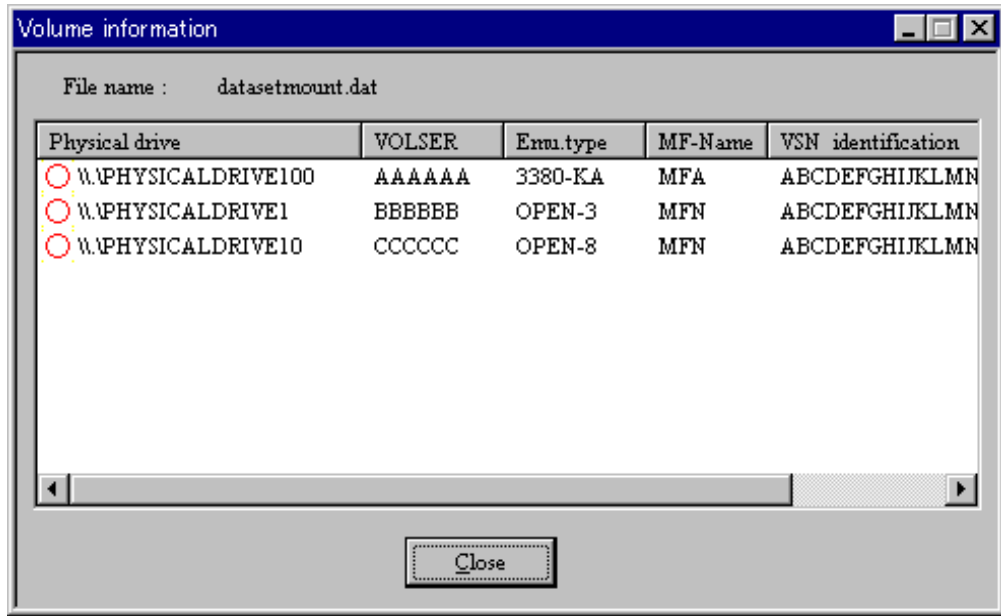


図 8.6.2-71 ボリューム情報ダイアログ

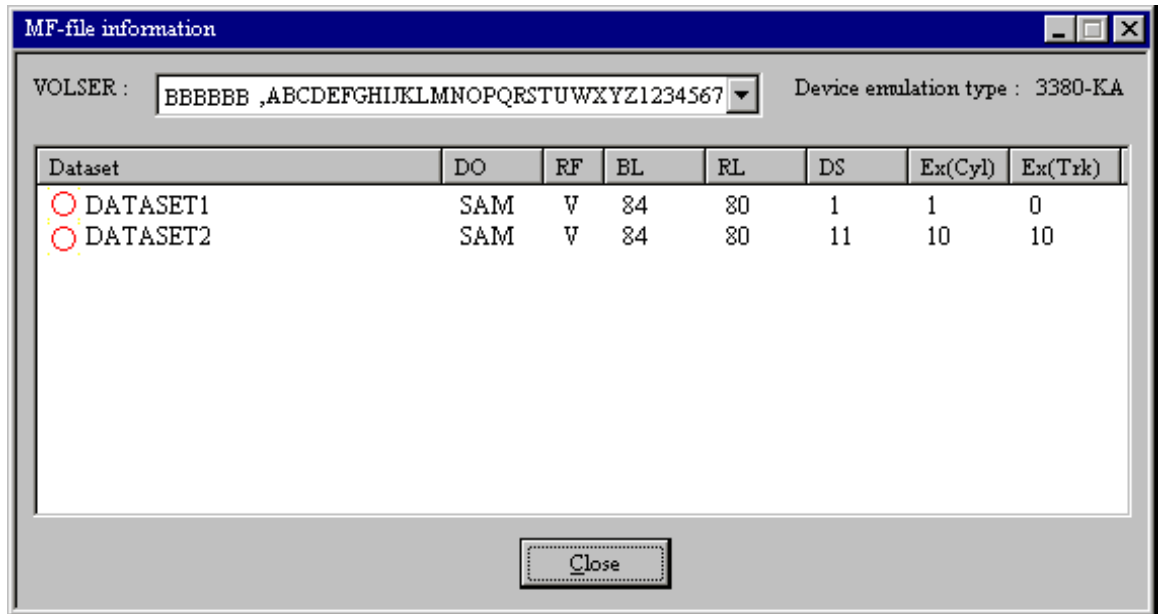
表示される内容は、ボリューム定義ファイルの内容です。

先頭の ○ は、ボリューム定義ファイルの記述が正しいことを示します。

記述が正しくない場合は、× を表示します。

(4) MF-file information

次の①～⑨のメインフレームファイル情報(メインフレームのデータセットの情報)を表示する場合に選択します。



#01-XX-47 以前では VOLSER 欄に VSN 識別子を表示しません。

図 8.6.2-72 メインフレーム情報ダイアログ

① データセット有効/無効フラグ

○ : FCU で処理できるデータセットであることを示します。

× : FCU で処理できないデータセットであることを示します。

? : 本ユーティリティで必要とする情報が定義されていないため処理できるか否かが不明であるか、または表示内容が正しくない可能性があります。RF=??または BL=0 または RL=0の時、VSEパラメでレコード形式・レコード長・ブロック長を正しくすれば処理できるようになります。またデータ量は、DSの値によりません。

② Dataset (データセット名)

③ DO (データセット編成)

SAM : 順編成

DAM : 直接編成

PAM : 区分編成

VSAM : Virtual Storage Access Method 編成

??? : 上記以外の編成

④ RF (レコード形式)

F : 固定長レコード

FB : 固定長ブロック化レコード (XSP 時は F と表示します。)

V : 可変長レコード

VB : 可変長ブロック化レコード (XSP 時は V と表示します。)

U : 不定長レコード

S : スパントレコード

? : 上記以外のレコード形式

⑤ BL (ブロック長)

⑥ RL (レコード長)

⑦ DS (データセットサイズ)

単位 : トラック

マルチボリューム定義ファイルに当該データセットが登録してある場合、全ボリュームのデータセットサイズの合計を表示します。

DS の値が 0 の場合、対象データセットは 0 バイトデータセットとなります。

- ⑧ **Ex(Cyl)** (データセットアロケート時のシリンダ数)
 マルチボリューム定義ファイルに当該データセットが登録してある場合でも、先頭ボリュームのみのアロケート時のサイズを示します。
- ⑨ **Ex(Trk)** (データセットアロケート時のトラック数)
 マルチボリューム定義ファイルに当該データセットが登録してある場合でも、先頭ボリュームのみのアロケート時のサイズを示します。

(5) Option

次の①～⑨のオプションを表示し、その指定内容を確認・変更したい場合に選択します。

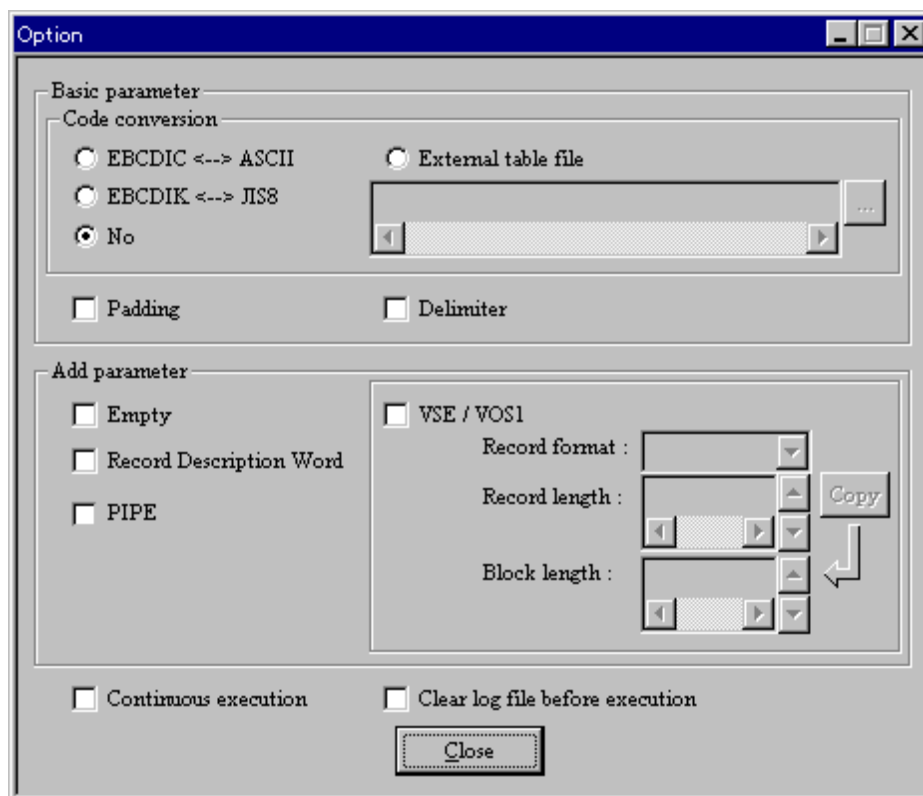


図 8.6.2-73(1) オプションダイアログ (その 1)

① Code conversion

コード変換の要否を指定します。

- ・ **EBCDIC <--> ASCII**: EBCDIC と ASCII 間のコード変換を行います。
- ・ **EBCDIK <--> JIS8** : EBCDIK と JIS8 間のコード変換を行います。
- ・ **No** : コード変換を行いません。
- ・ **External table file** : ユーザーが指定したコード変換テーブルに従ってコード変換を行います。
 下のテキストボックスにコード変換テーブルファイルの名称を入力します。

(注 1) オープンシステムーオープンシステム間のデータ交換時は、コード変換をサポートしていません。
 デバイスエミュレーションタイプが OPEN-X(X:3, 8, 9, K, E, L, M, V) の場合は、Code Conversion の「EBCDIC<->ASCII」, 「EBCDIK<->JIS8」, 「External table file」ボタンは選択不可(グレー表示)となります。パラメータ定義ファイルからの読み込み時、コード変換の指定が No 以外の時エラーとなります。

② **Padding**

パディングの可否を指定します。
詳細については7.4節を参照して下さい。

③ **Delimiter**

デリミタの可否を指定します。
詳細については7.5、8.4節を参照して下さい。

④ **Empty**

ファイルサイズが0のファイルを転送の対象とするか否かを指定します。
詳細については8.4節を参照して下さい。

⑤ **Record Description Word**

各データの先頭にデータ長を示す情報を付加した形でオープンシステム側のファイルを作成するか否かを指定します。
詳細については8.4節を参照して下さい。

(注) 本チェックボックスをチェックした場合、以下のものは無効となります。

EBCDIC <--> ASCII ラジホたん
EBCDIK <--> JIS8 ラジホたん
Padding チェックボックス
Delimiter チェックボックス

⑥ **Continuous execution**

連続実行するか否かを指定します。

(注) 連続実行は、現在位置のパラメタ行から最後のパラメタ行までを連続実行します。

連続実行中(-ncで起動した場合は除く。)に、本チェックを外せば、現在実行中の処理が終了した時点で、連続実行を止めることができます。

⑦ **Clear log file before execution**

実行前にログファイルをクリアするか否かを指定します。

⑧ **PIPE**

FAL SQL Loader 連携時に指定します。MTO時の出力ファイルがパイプファイルになります。

(注) FAL SQL Loader 連携時以外は、指定しないでください。

ただし、FAL SQL Loader は USP V と VSP, VSP G1000, VSP 5000 では未サポートです。

⑨ VSE/VOS1

レコード形式、レコード長、ブロック長が定義されていないデータセットを扱う場合にレコード形式、レコード長、ブロック長を指定します。(メインフレームが VSE/VOS1 の場合指定が必要です。)

(注) オープンシステム-オープンシステム間のデータ交換では使用しません。

- **Record format** : レコード形式を以下の文字列で指定します。
 - F** : 固定長非ブロック化
 - FB** : 固定長ブロック化
 - V** : 可変長非ブロック化
 - VB** : 可変長ブロック化
- **Record length** : レコード長(バイト数)を以下の範囲で指定します(十進数)。
 - レコード形式=**F** : レコード長=ブロック長
 - レコード形式=**FB** : レコード長=ブロック長÷n n : 整数
 - レコード形式=**V** : $5 \leq \text{レコード長} \leq (\text{ブロック長}-4)$
 - レコード形式=**VB** : $5 \leq \text{レコード長} \leq (\text{ブロック長}-4)$
- **Block length** : ブロック長(バイト数)を以下の範囲で指定します(十進数)。
 - レコード形式=**F** : $1 \leq \text{ブロック長} \leq 32760$
 - レコード形式=**FB** : $1 \leq \text{ブロック長} \leq 32760$
 - レコード形式=**V** : $9 \leq \text{ブロック長} \leq 32760$
 - レコード形式=**VB** : $9 \leq \text{ブロック長} \leq 32760$

各情報値は手入力及びメニューからの選択が可能です。

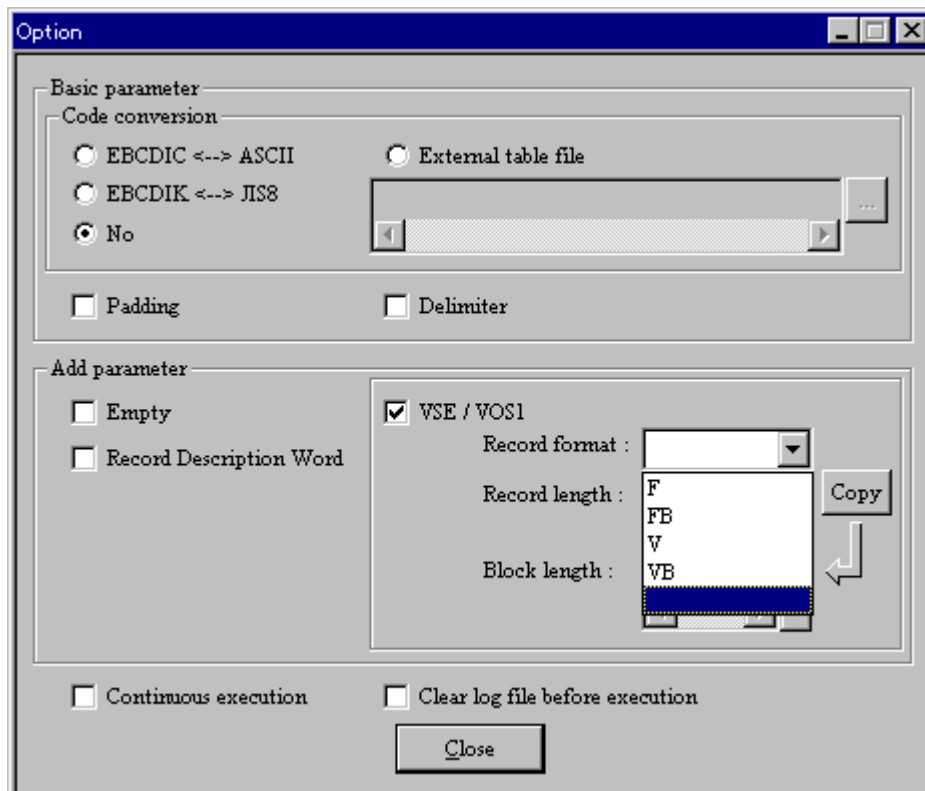


図 8.6.2-73(2) オプションダイアログ (その2)

(6) **Error information**

エラー情報を表示する場合に選択します。

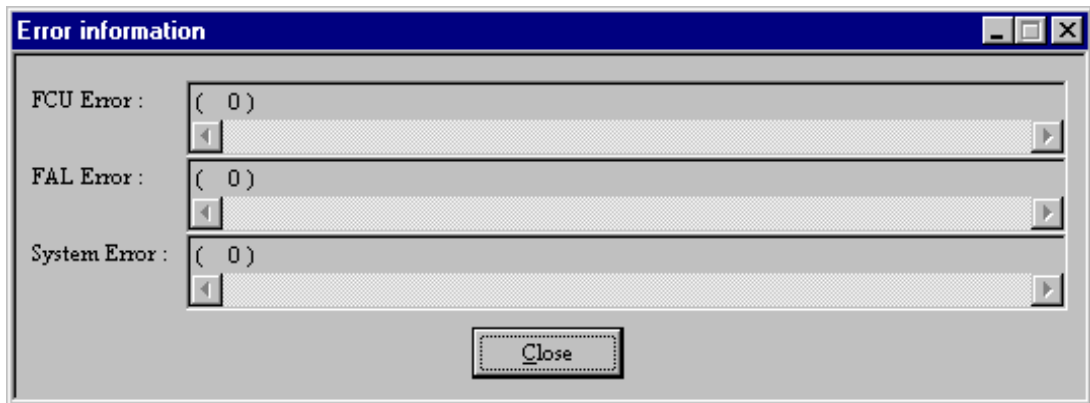


図 8. 6. 2-74 エラー情報ダイアログ

FCU Error : **FCU** のエラー番号及び、エラーメッセージを表示します。

FAL Error : **FAL** のエラー番号及び、エラーメッセージを表示します。

System Error : システムのエラー番号及び、エラーメッセージを表示します。

(7) **Parameter line**

パラメタ行を表示する場合に選択します。

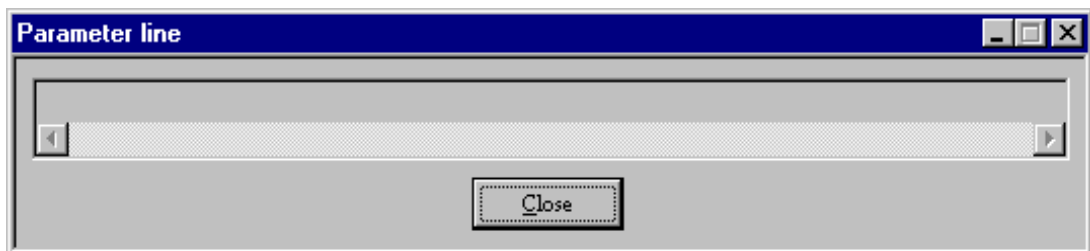


図 8. 6. 2-75 パラメタ行ダイアログ

(注) パラメタ行は、パラメタ定義ファイルの 1 行に相当する情報で、現在位置のパラメタ行が表示されます。

(8) **Close all dialog**

全ダイアログをクローズする場合に選択します。

(9) Log file

ログファイルの内容を表示する場合に選択します。

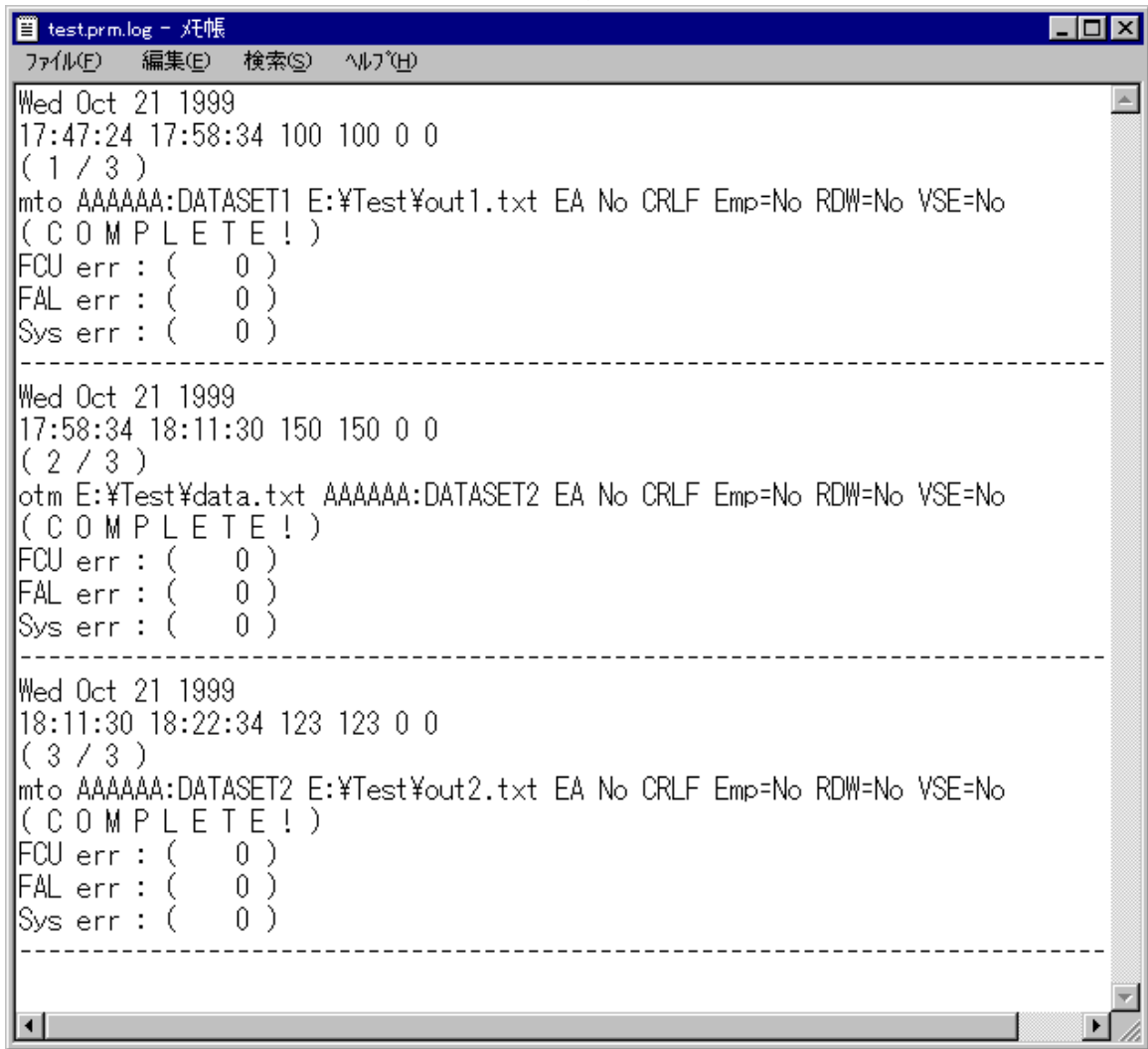


図 8.6.2-76 ログファイル

ログは、パラメタ定義ファイル名に「.log」を付加したファイルに記録されます。

例) パラメタ定義ファイル : **Test.prm**
 ログファイル : **Test.prm.log**

(注) ログファイルをクリアしないまましていると、実行中ディスクが一杯になる場合があります。

オプションダイアログの「**Clear log file before execution**」をチェックしておくと、実行前にログファイルのクリアを行います。

但し、ログファイルはパラメタ定義ファイル毎に分散して作成される為、「**Clear log file before execution**」をチェックしておくだけでは上記問題を回避できないかも知れません。

その場合は、任意にログファイル(又は、他の不要なファイル)を削除して、ディスクの空領域を確保して下さい。

8.6.2.2.5 Help メニュー

バージョン情報を表示する場合に選択します。

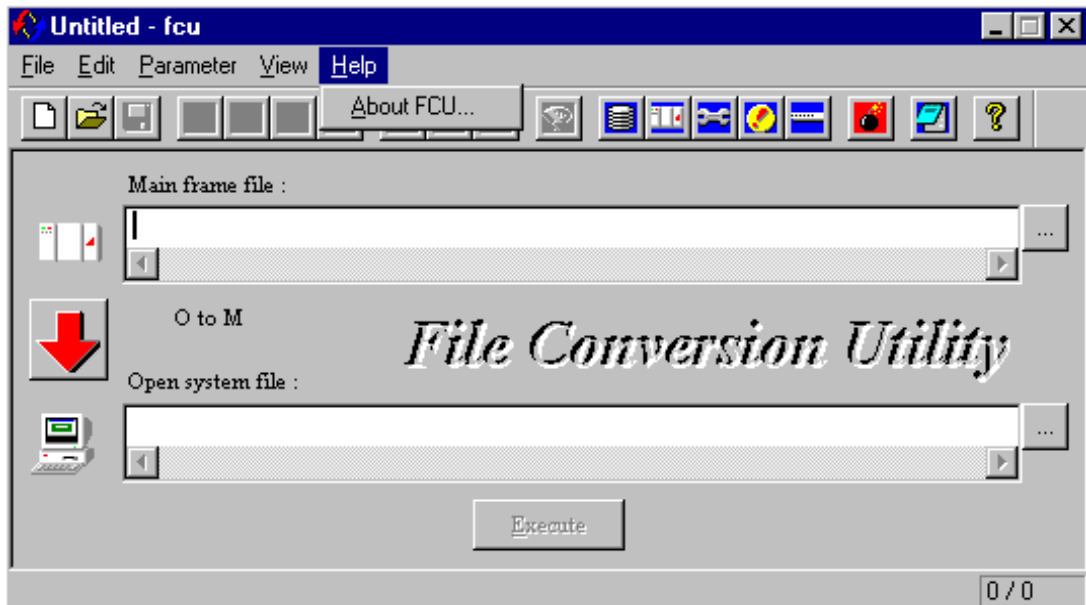


図 8.6.2-77 Help メニュー




図 8.6.2-78 バージョン情報

8.6.2.2.6 ツールバー

- (1) **New** ボタン 
メニューの「File」 - 「New」と同じ処理を行います。
- (2) **Open** ボタン 
メニューの「File」 - 「Open」と同じ処理を行います。
- (3) **Save** ボタン 
メニューの「File」 - 「Save」と同じ処理を行います。
- (4) **Top** ボタン 
メニューの「Parameter」 - 「Load」 - 「Top」と同じ処理を行います。
- (5) **Previous** ボタン 
メニューの「Parameter」 - 「Load」 - 「Previous」と同じ処理を行います。
- (6) **Next** ボタン 
メニューの「Parameter」 - 「Load」 - 「Next」と同じ処理を行います。
- (7) **Bottom** ボタン 
メニューの「Parameter」 - 「Load」 - 「Bottom」と同じ処理を行います。
- (8) **Insert** ボタン 
メニューの「Parameter」 - 「Save」 - 「Insert」と同じ処理を行います。
- (9) **Replace** ボタン 
メニューの「Parameter」 - 「Save」 - 「Replace」と同じ処理を行います。
- (10) **Delete** ボタン 
メニューの「Parameter」 - 「Delete」と同じ処理を行います。
- (11) **Wipe screen** ボタン 
メニューの「Parameter」 - 「Wipe」と同じ処理を行います。
- (12) **Volume info** ボタン 
メニューの「View」 - 「Volume information」と同じ処理を行います。
- (13) **MF-file info** ボタン 
メニューの「View」 - 「MF-file information」と同じ処理を行います。
- (14) **Option** ボタン 
メニューの「View」 - 「Option」と同じ処理を行います。


(15) **Error info** ボタン 
メニューの「View」 - 「Error information」と同じ処理を行います。



(16) **Parameter line** ボタン 
メニューの「View」 - 「Parameter line」と同じ処理を行います。

(17) **Close all dialogs** ボタン 
メニューの「View」 - 「Close all dialogs」と同じ処理を行います。

(18) **Log** ボタン 
メニューの「View」 - 「Log file」と同じ処理を行います。

(19) **About FCU** ボタン 
メニューの「Help」 - 「About FCU」と同じ処理を行います。

8.6.2.2.7 **Direction** ボタン 
データ転送方向を指定します。クリック毎に反転します。

 : メインフレーム → オープンシステム への転送を指定します。
 : オープンシステム → メインフレーム への転送を指定します。

8.6.2.2.8 Main frame file 欄



メインフレームファイル名を指定します。
メインフレームファイル名は、「ボリュームシリアル番号」と「データセット名」を“:”で区切った形式で指定します。

例) **AAAAAA:DATASET**

必要であれば、VSN 識別子を“,”で区切った形式で指定します。

例) **AAAAAA:DATASET,MF-MVS01**

「ボリュームシリアル番号」は 6 桁以下、「データセット名」は 44 桁以下の英数字で指定します。
01-XX-48 以降は小文字も入力可能です。

8.6.2.2.9 Open system file 欄



オープンシステムファイル名を指定します。

(注) オープンシステムファイル名及び、そのパスにスペースが含まれる入力を行い、パラメタ定義ファイルに保存した場合、読み出し時にそのパラメタ行はエラーとなります。(パラメタ定義ファイルは、各パラメタのセパレータとしてスペースを用いているためです。)

8.6.2.2.10 メインフレームファイル選択ボタン



本ボタンをクリックすると、メインフレームファイル選択ダイアログが表示されます。

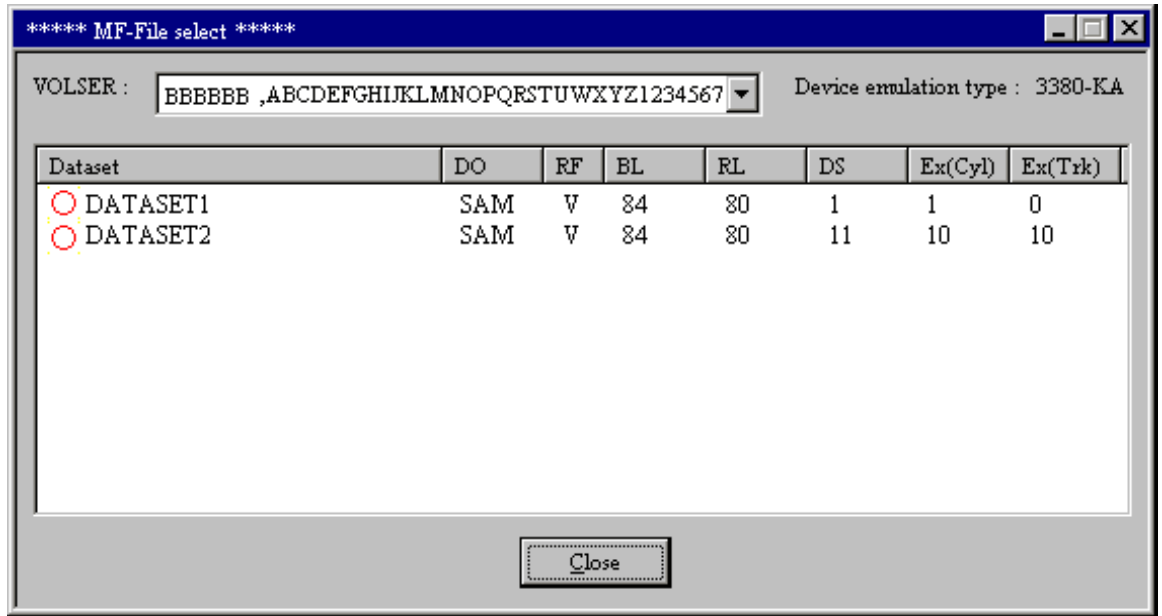


図 8.6.2-79 メインフレームファイル選択ダイアログ

データセットを選択するには、選択したいデータセットをダブルクリックします。

(注) 本ダイアログは、メインフレーム情報ダイアログと同じですが、有効なデータセット（先頭に○がついているデータセット）のみ表示されます。
又、タイトルバーに表示される文字が異なります。

8.6.2.2.11 オープンシステムファイル選択ボタン



本ボタンをクリックすると、ファイル選択ダイアログが表示されます。

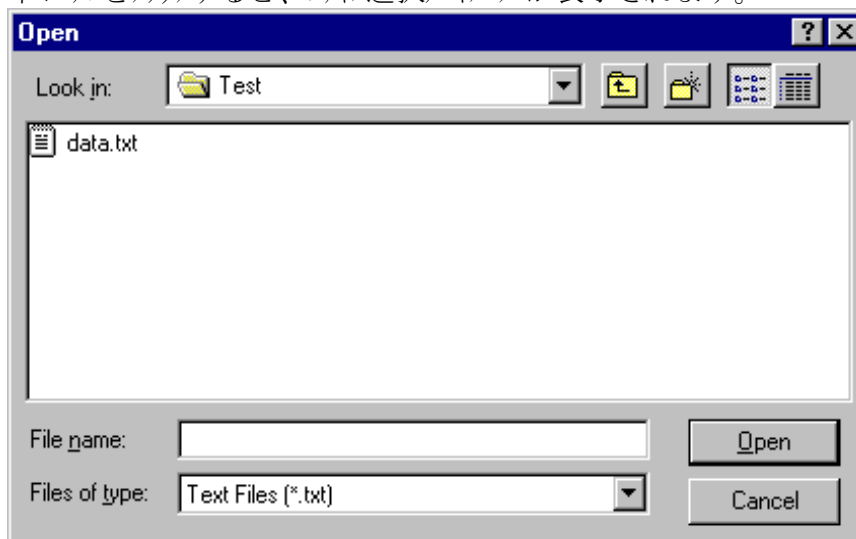


図 8.6.2-80 オープンシステムファイル選択ダイアログ

Execute

8.6.2.2.12 Execute ボタン

データ転送の開始を指示します。

データ転送先のファイルに既にデータが作成されていても、FCU はファイルの先頭からオーバーライトします。このため、FCU 実行すると元のデータは書き換えられてしまいますので、データ転送先ファイルの指定には十分注意して下さい。

(注) FCU 本体画面及び、オプションダイアログのパラメタを変更した場合は、「Insert」又は、「Replace」を選択しないと本ボタンが有効になりません。
又、パラメタにエラーがある場合も、本ボタンは有効になりません。

8.6.2.2.13 実行ダイアログ

「Execute」ボタンが押された場合に進行状態を表示します。

(ただし、VSE パラメタの指定を行うと、MTO 時は進行状態の表示を行いません。) 実行中は、「Close」ボタンでなく、「Cancel」ボタンを表示します。

(1) 正常終了した場合

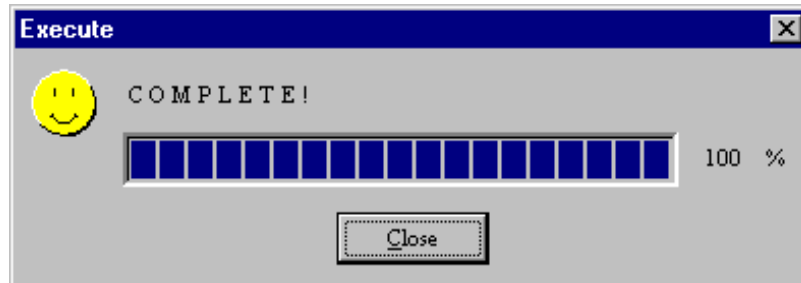


図 8.6.2-81 正常終了時

(2) エラー終了した場合

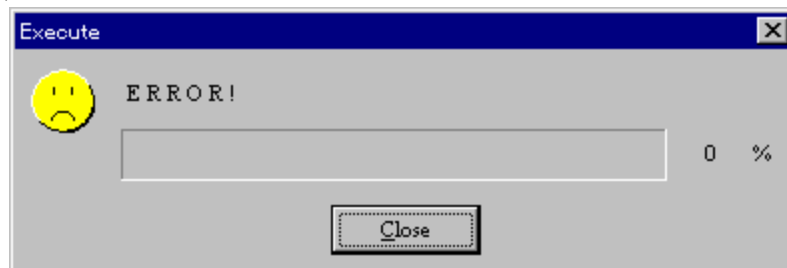


図 8.6.2-82 エラー終了時

(3) キャンセルした場合

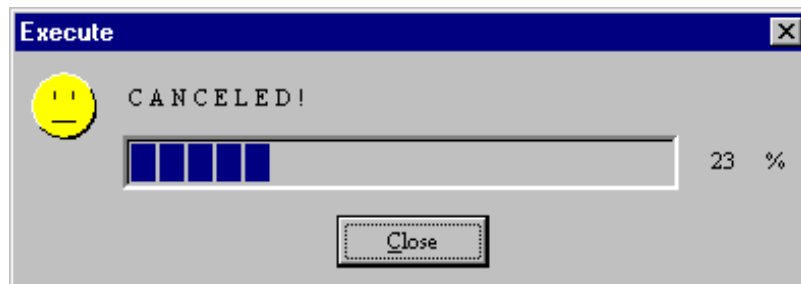


図 8.6.2-83 キャンセル時

8.6.2.2.14 ステータスバー

現在のパラメタ位置 / 全体のパラメタ件数 を表示します。

0 / 0

8.7 操作画面以外の操作及び情報の表示

8.7.1 UNIX系OSの場合

8.7.1.1 FCU

操作画面を使用しない(fcuを"-nw"指定で起動またはfcunwを起動) 場合の操作及びFCUが表示する情報は次のようになります。

(1) 詳細パラメタの表示及びパラメタの確認

FCUは、起動コマンドを投入したウィンドウに、パラメタ定義ファイルから得た詳細パラメタを表示し、確認応答を要求します。パラメタを確認し、ファイル変換処理を実行してもよければ"ok"を、実行してはならないのであれば"cancel"を応答して下さい。FCUは、"ok"が応答された場合はファイル変換処理を開始し、"cancel"が応答された場合は次の詳細パラメタを読み出し、確認応答を要求します。詳細パラメタの終了を検出した場合は処理を終了します。なお、起動パラメタで"-nc"が指定された場合、確認応答は要求しません。パラメタ表示後、直ちにファイル変換処理を開始します。

```
File Conversion Utility Ver.バージョン名  
  
mto VSN:データセット名 UNIXファイル名 CC PAD DEL  
ok/cancel ?
```

(注) パラメタの詳細については8.4節を参照して下さい。

図8.7-1 パラメタの確認

(2) チェック中の表示

ファイルの有無等をチェック中のとき、「Now checking」を表示します。

(3) オーバライトの確認

出力となるオープンシステムファイルがすでに存在している場合、オーバライトの可否を確認します。オーバライトしてよければ"ok"を、オーバライトしないのであれば"cancel"を応答して下さい。"cancel"が応答された場合のFCUの動作は、上記(1)と同様です。なお、起動パラメタで"-nc"が指定されている場合、オーバライトの可否の確認応答を要求することなく、ファイル変換処理(オーバライト)を開始します。

```
Overwrite ? ok/cancel ?
```

図8.7-2 オーバライトの確認

(4) 処理開始の表示

ファイル変換処理を開始するとき、「Start」を表示します。

(5) ファイル変換処理結果の表示

ファイル変換処理が正常終了した場合は、「Complete」のメッセージを表示します。

異常終了の場合は、エラーメッセージを出力します。

また、正常終了の場合0、異常終了の場合1を終了ステータスとして返します。(この終了ステータスは、Cシェルの場合「\$status」、Bシェル/Kシェルの場合「\$?」に入ります)

(6) ファイル変換処理の繰り返し

パラメタ定義ファイル中の全パラメタの処理が終了するまで、上記(1)から(3)の処理を繰り返します。終了ステータスは全処理が正常に終了した場合にのみ「0」を返します。

8.7.1.2 LISTVOL

指定されたボリュームのデータセット名の一覧を標準出力に表示します。

コマンドを実行する前に以下を確認して下さい。

- ①コマンド実行対象のボリュームがボリューム定義ファイルに登録されていること。
- ②マルチボリュームデータセットを割り当てた場合は転送対象のデータセットを割り当てた全てのボリュームがマルチボリューム定義ファイルに登録されていること。
- ③コマンド実行対象のボリュームにデータセットが新規に割り当てられた場合は、VTOCのDSCB1にあるデータセットインジケータの最終ボリュームフラグを確定させるため、コマンド実行前にM/F側から一度データ書込みが実行済みであること。

(1) 実行方法

以下のコマンドとパラメタによりLISTVOL(32bit版)/LISTVOL64(64bit版)を実行させます。

listvol VSN, VSN識別子

listvol64 VSN, VSN識別子

VSN : VSN(ボリュームシリアルナンバー)は6桁以下の英文字及び英数字で指定します。

1～6文字以外のVSNを指定するとエラーとなります。またボリューム定義ファイルに存在しないVSNを指定するとエラーとなります。

VSN識別子 : VSN識別子を35文字以内の英数字及び英記号で指定可能です。(省略可)

(2) 表示内容

上記手順により指定されたボリューム内のデータセット名、データセット編成種別、レコード形式、レコード長、ブロック長、最終ブロックのアドレス、データセットエクステントサイズ(シリンダ数:トラック数)、FAL/FCUによる処理の可否を表示します。

# listvol xxxxxx, YYYY	DO	RF	RL	BL	TT	R	EX(Cyl:Trk)
*SAMFILE01. FIX	SAM	F	4096	4096	1	10	100:0
-DAMFILE. F	DAM	FB	128	4096	0	10	100:0
?SAMFILE. VSE	SAM	??	0	0	0	0	0:0

図8.7-3 LISTVOLの表示内容

・"Dataset Name":データセット名

データセット名の前の1文字

*:FCUで処理できるデータセットであることを示します。

-:FCUで処理できないデータセットであることを示します。

? :本ユーティリティで必要とする情報が定義されていないため処理できるか否かが不明であるか、または表示内容が正しくない可能性があります。RF=??またはBL=0またはRL=0の時、VSEパラメタでレコード形式・レコード長・ブロック長を正しくすれば処理できるようになります。またデータ量は、TT Rの値によりません。

・"DO":データセット編成種別

SAM : 順編成

DAM : 直接編成

PAM : 区分編成

VSAM : Virtual Storage Access Method編成

??? : 上記以外の編成種別で、本ユーティリティでは識別できません。

・“RF” :レコード形式

- F :固定長レコード
- FB :固定長ブロック化レコード
- V :可変長レコード
- VB ;可変長ブロック化レコード
- U :不定長レコード
- S :スポットレコード
- ?? :上記以外のレコード形式で、本エリティでは識別できません。

・“RL” :レコード長(バイト)

・“BL” :ブロック長(バイト)

・“TT+R” :最終ブロックのアドレス

マルチボリューム定義ファイルに当該データセットが登録してある場合、全ボリュームのデータセットサイズの合計を表示します。

TT+Rの値が共に0の場合、対象データセットは0バイトデータセットとなります。

・“EX(Cyl:Trk)” :データエクステンション(シリンダ数:トラック数)

マルチボリューム定義ファイルに当該データセットが登録してある場合でも、先頭ボリュームのみのアロケート時のサイズを示します。

(3) リターン値

実行終了後、以下のリターン値を返します。

0 : 正常終了

1 : エラー終了

(4) エラーメッセージ

処理中に異常を検出した場合は、エラーコード、エラーメッセージを標準エラー出力へ表示します。

また、エラーログファイルにもエラー情報を出力します。

(詳細は、8.8節を参照して下さい)

(5) 注意事項

ボリューム定義ファイルを作成しておく必要があります。

(詳細は、8.2節を参照して下さい)

レコード形式がFBXやVBXの時LISTVOLのRFはFB、VBと表示されます。

8.7.2 Windowsの場合

8.7.2.1 FCU

操作画面を使用しない(fcunwを起動) 場合の操作及びFCUが表示する情報は次のようになります。

(1) 詳細パラメタの表示及びパラメタの確認

FCUは、起動コマンドを投入したウィンドウに、パラメタ定義ファイルから得た詳細パラメタを表示し、直ちにファイル変換処理を開始します。

詳細パラメタの終了を検出した場合は処理を終了します。

```
fcunw fcudata.prm
*** fcudata.prm *** start

mto VSN:データセット名 Windowsファイル名 CC PAD DEL
```

(注) パラメタの詳細については8.4節を参照して下さい。

図8.7-4 FCUパラメタの表示

(2) チェック中の表示

ファイルの有無等をチェック中のとき、「Now checking」を表示します。

(3) 処理開始の表示

ファイル変換処理を開始するとき、「Start」を表示します。

(4) ファイル変換処理結果の表示

ファイル変換処理が正常終了した場合は、「COMPLETE」のメッセージを表示します。

異常終了の場合は、エラーメッセージを出力します。

また、正常終了の場合0、異常終了の場合1を終了ステータスとして返します。

(5) ファイル変換処理の繰り返し

パラメタ定義ファイル中の全パラメタの処理が終了するまで、上記(1)から(4)の処理を繰り返します。

終了ステータスは全処理が正常に終了した場合にのみ「0」を返します。

(6) 注意事項

パラメタ定義ファイルを作成しておく必要があります。

パラメタ定義ファイルが存在しないか、もしくはパラメタが不正の場合、次のメッセージが表示されます。

A parameter definition file doesn't exist, or it is illegal.

8.7.2.2 LISTVOL

指定されたボリュームのデータセット名の一覧を標準出力に表示します。(01-XX-66以降適用)
コマンドを実行する前に以下を確認して下さい。

- ①コマンド実行対象のボリュームがボリューム定義ファイルに登録されていること。
- ②マルチボリュームデータセットを割り当てた場合は転送対象のデータセットを割り当てた全てのボリュームがマルチボリューム定義ファイルに登録されていること。
- ③コマンド実行対象のボリュームにデータセットが新規に割り当てられた場合は、VTOCのDSCB1にあるデータセットインジケータの最終ボリュームフラグを確定させるため、コマンド実行前にM/F側から一度データ書込みが実行済みであること。

(1) 実行方法

以下のコマンドとパラメタによりLISTVOL(32bit版)/LISTVOL64(64bit版)を実行させます。

listvol VSN, VSN識別子

listvol64 VSN, VSN識別子

VSN : VSN(ボリュームシリアルナンバー)は6桁以下の英文字及び英数字で指定します。

1～6文字以外のVSNを指定するとエラーとなります。またボリューム定義ファイルに存在しないVSNを指定するとエラーとなります。

VSN識別子 : VSN識別子を35文字以内の英数字及び英記号で指定可能です。(省略可)

(2) 表示内容

上記手順により指定されたボリューム内のデータセット名、データセット編成種別、レコード形式、レコード長、ブロック長、最終ブロックのアドレス、データセットエクステントサイズ(シリンダ数:トラック数)、FAL/FCUによる処理の可否を表示します。

# listvol xxxxxx, YYYY	DO	RF	RL	BL	TT	R	EX(Cyl:Trk)
*SAMFILE01. FIX	SAM	F	4096	4096	1	10	100:0
-DAMFILE. F	DAM	FB	128	4096	0	10	100:0
?SAMFILE. VSE	SAM	??	0	0	0	0	0:0

図8.7-5 LISTVOLの表示内容

・"Dataset Name":データセット名

データセット名の前の1文字

*:FCUで処理できるデータセットであることを示します。

-:FCUで処理できないデータセットであることを示します。

? :本ユーティリティで必要とする情報が定義されていないため処理できるか否かが不明であるか、または表示内容が正しくない可能性があります。RF=??またはBL=0またはRL=0の時、VSEパラメタでレコード形式・レコード長・ブロック長を正しくすれば処理できるようになります。またデータ量は、TT Rの値によりません。

・"DO":データセット編成種別

SAM : 順編成

DAM : 直接編成

PAM : 区分編成

VSAM : Virtual Storage Access Method編成

??? : 上記以外の編成種別で、本ユーティリティでは識別できません。

・"RF" :レコード形式

- F :固定長レコード
- FB :固定長ブロック化レコード
- V :可変長レコード
- VB ;可変長ブロック化レコード
- U :不定長レコード
- S :スポットレコード
- ?? :上記以外のレコード形式で、本エリティでは識別できません。

・"RL" :レコード長(バイト)

・"BL" :ブロック長(バイト)

・"TT+R" :最終ブロックのアドレス

マルチボリューム定義ファイルに当該データセットが登録してある場合、全ボリュームのデータセットサイズの合計を表示します。

TT+Rの値が共に0の場合、対象データセットは0バイトデータセットとなります。

・"EX(Cyl:Trk)" :データエクステンション(シリンダ数:トラック数)

マルチボリューム定義ファイルに当該データセットが登録してある場合でも、先頭ボリュームのみのアロケート時のサイズを示します。

(3) リターン値

実行終了後、以下のリターン値を返します。

0 : 正常終了

1 : エラー終了

(4) エラーメッセージ

処理中に異常を検出した場合は、エラーコード、エラーメッセージを標準エラー出力へ表示します。

また、エラーログファイルにもエラー情報を出力します。

(詳細は、8.8節を参照して下さい)

(5) 注意事項

ボリューム定義ファイルを作成しておく必要があります。

(詳細は、8.2節を参照して下さい)

8.8 エラー情報

FCUが出力するエラーコードとエラーメッセージには以下のものがあります。

(注1) 表8.8-1,2のエラーコードの中で、エラーコード欄に「**」が付いているエラーコードは、エラーメッセージ欄で説明している原因の他に、当該ファイル/ボリュームへの競合(メインフレーム-オープンシステム間/オープンシステム-オープンシステム間の同時アクセス(リソース競合))が原因となって発生することがあります。エラーメッセージ欄で記述している事項の調査とともに、競合が発生して(リソースが掛かって)いないかを調査して下さい。

なお、このとき以下のシステムエラーが発生していることもあります。

システムエラーコード = 4/5/6/9/11/14/16/22/36/39/45/46/67

また、磁気ディスク装置の異常がオープンシステムのOSからエラーとして報告されないために、「**」を付与したエラーコードがFCUの実行結果として報告される場合もあります。この場合は、エラーメッセージ欄で記述している事項の調査とともに、磁気ディスク装置に何らかの異常が発生していないかを調査して下さい。

(注2) エラー情報のうち、正のエラーコードはシステムエラーです。

システムエラーについては、UNIX系OSの場合、標準エラーファイル (errno.h) で定義されています。

Windowsの場合は、Microsoft Visual C++に付属の errno.h となります。

(注3) 表8.8.1-1,2のエラーコードの中でエラーコード欄に「(K)」が付いているエラーコードは、HMDE/KANJIと連携のみ発生します。詳細については、HMDE/KANJIユーザーズガイドを参照願います。

8.8.1 UNIX系OSの場合

表 8.8-1 FCU (LISTVOL/LISTVOL64) のエラーコード・エラーメッセージ

エラーコード	内 容
-100	No Parameter file (パラメタ定義ファイルが見つかりません) ⇒ パラメタ定義ファイル名を指定している場合：指定したパラメタ定義ファイルが作成してあるかを確認してください。 パラメタ定義ファイル名を指定していない場合：カントディレクトリにデフォルトのパラメタ定義ファイル(fcudata.param)が作成してあるかを確認してください。
-101 **	Parameter file : Open error (パラメタ定義ファイルのオープンに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-102 **	Parameter file : Read error (パラメタ定義ファイルのリードに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-103 **	Parameter file : No Valid error (パラメタ定義ファイルが正しくありません) ⇒ パラメタ定義ファイルが正しく記述されているか確認してください(8.4節参照)
-107	Parameter file : CODE_CONV error (パラメタのコード変換種別指定が正しくありません。) ⇒ オープンシステム-オープンシステム間のデータ交換時のコード変換種別は「No」である必要があります。コード変換種別は「No」を指定してください。(8.4節参照)
-108	Parameter : PADDING error (パラメタのパディングが正しくありません) ⇒ パディングは「Yes」または「No」である必要があります。パディングは「Yes」または「No」を指定してください。(8.4節参照)
-109	Parameter : DELIMITER error (パラメタのデリミタが正しくありません) ⇒ デリミタは「CR」/「LF」/「No」のいずれかである必要があります。デリミタは「CR」/「LF」/「No」のいずれかを指定してください(8.4節参照)

表8. 8-1 FCU (LISTVOL/LISTVOL64)のエラーコード・エラーメッセージ

エラーコード	内 容
-110 **	Parameter file : Open error (パラメタ定義ファイルのオープンに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-111 **	Parameter file : Write error (パラメタ定義ファイルのライトに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-112 **	Parameter file : Close error (パラメタ定義ファイルのクローズに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-114	Parameter : No input file name (パラメタの入力ファイル名が指定されていません) ⇒ 入力ファイル名を指定して下さい。
-115	Parameter : VSN error (VSN が見つかりません) ⇒ パラメタの VSN が正しいか確認してください。(8.4 節参照) またボリューム定義ファイル内に VSN が定義してあるか確認してください。
-116	Parameter : Input file name error (パラメタの入力ファイル名が正しくありません) ⇒ 入力ファイル名が正しいか確認してください。
-117	Parameter : Dataset name error (パラメタのデータセット名が正しくありません) ⇒ データセット名が正しいか確認してください。(8.4 節参照)
-118	Parameter : Output file name error (パラメタの出力ファイル名が正しくありません) ⇒ 出力ファイル名が正しいか確認してください。
-119 **	Input file : Open error (入力ファイルのオープンに失敗しました) ⇒ FAL エラーコード、システムエラーコードを確認してください。(付録 ログフォーマット例参照) FAL エラーコードが表示されている場合 : FAL エラーコードの内容を確認してください。(表 4. 7-1 参照) またシステムエラーコードが表示されている場合 : OS のシステムエラーコードが示すエラー内容を確認してください。
-120	Overwrite ? (OK/Cancel) (出力ファイルにオーバーライトします) ⇒ 出力ファイルが既に存在しています。 オーバーライトしてよい場合は「OK」を、別のファイルを指定し直す場合は「Cancel」を応答します。
-121	Output file : No output file name (パラメタの出力ファイル名が指定されていません) ⇒ 出力ファイル名を指定して下さい。
-122 **	Output file : Open error (出力ファイルのオープンに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-124	Volume definition : MFtype error (ボリューム定義ファイル内の MF タイプが正しくありません。) ⇒ ボリューム定義ファイル内の MF タイプは「MFA」か「MFN」のいずれかである必要があります。MF タイプは「MFA」か「MFN」を指定してください(8.2 節参照)
-125 **	Volume definition : VSN error (VSN が正しくありません) ⇒ ボリューム定義ファイル内およびパラメタの VSN を確認してください。(8.2 節、8.4 節参照)
-126	Volume definition : Partition name error (ボリューム定義ファイルのパーティション名が正しくありません) ⇒ ボリューム定義ファイル内のパーティション名が正しいか確認してください。

表8.8-1 FCU (LISTVOL/LISTVOL64)のエラーコード・エラーメッセージ

エラーコード	内 容
-127 **	Volume definition : Emulation error (ボリューム定義ファイルのエミュレーションタイプが正しくありません) ⇒ ボリューム定義ファイル内のデバイスエミュレーションタイプを確認してください。(1.2.2節参照)
-128 **	Volume definition file : Open error (ボリューム定義ファイルのオープンに失敗しました) ⇒ OSでエラーを検出しています。OSのシステムエラーコードが示すエラー内容を確認してください。 (例) ボリューム定義ファイルが作成されていない場合、エラーコード=2 (No such file or directory)が発生します。
-129 **	Volume definition file : Read error (ボリューム定義ファイルのリードに失敗しました) ⇒ OSでエラーを検出しています。OSのシステムエラーコードが示すエラー内容を確認してください。
-130 **	Volume definition file : No data (ボリューム定義ファイルにデータがありません) ⇒ ボリューム定義ファイル内に有効なデータがありません。ボリューム定義ファイルを確認してください。
-131 **	Volume definition file : Close error (ボリューム定義ファイルのクローズに失敗しました) ⇒ OSでエラーを検出しています。OSのシステムエラーコードが示すエラー内容を確認してください。
-132	Volume definition file : No valid error (ボリューム定義ファイル内のパラメータが正しくありません) ⇒ ボリューム定義ファイルを確認してください。(8.2節参照)
-135	Parameter error : No dataset name (HelpのMF-Fileの指示で入力ファイル名(VSN)が指定されていません) ⇒ Input FileにVSNを入力して、Help→MF-Fileを指示してください。(8.6.1節参照)
-136	Parameter error : VSN error (HelpメニューのMF-Fileの指示でInput FileのVSNが正しくありません) ⇒ Input FileのVSNを確認してください。(8.6.1節参照)
-137	Dataset error : No dataset (HelpメニューのMF-Fileが指示されましたがデータセットがありません) ⇒ データセットが存在するVSNを指定して、Help→MF-Fileを指示してください。(8.6.1節参照)
-138 **	Dataset error : Search error (データセットが見つかりません) ⇒ パラメータで指定したVSNを確認してください。 FALエラーコード、システムエラーコードを確認してください。(付録ログフォーマット例参照) FALエラーコードが表示されている場合:FALエラーコードの内容を確認してください。(表4.7-1参照) またシステムエラーコードが表示されている場合:OSのシステムエラーコードが示すエラー内容を確認してください。
-139	Dataset error : Close error (データセットのクローズに失敗しました) ⇒ FALエラーコード、システムエラーコードを確認してください。(付録ログフォーマット例参照) FALエラーコードが表示されている場合:FALエラーコードの内容を確認してください。(表4.7-1参照) またシステムエラーコードが表示されている場合:OSのシステムエラーコードが示すエラー内容を確認してください。
-140	Input file error : Invalid organization type (データセットの編成種別が正しくありません) ⇒ データセットの編成種別を確認してください。(1.2.2節参照)

表 8. 8-1 FCU (LISTVOL/LISTVOL64) のエラーコード・エラーメッセージ

エラーコード	内 容
-141	Input file error : Invalid record format (データセットのレコード形式が正しくありません) ⇒ データセットのレコード形式を確認してください。(1. 2. 2 節参照)
-142	Input file error : Invalid block length (データセットのブロック長が正しくありません) ⇒ データセットのブロック長を確認してください。(1. 2. 2 節参照)
-143	Input file error : Invalid record length (データセットのレコード長が正しくありません) ⇒ データセットのレコード長を確認してください。(1. 2. 2 節参照)
-144 **	Input file error : No data (データセットにデータがありません) ⇒ データセットにデータがあるか確認してください。 listvol で表示された TT+R の値が 0 か、Help の MF-File を指示してデータセットの属性情報を表示した際の DS の値が 0 の場合はデータセットにデータがありません。
-150 **	Input file : Open error (mto 転送実行時、入力ファイルのオープンに失敗しました) ⇒ FAL エラーコード、システムエラーコードを確認してください。(付録 ログフォーマット例参照) FAL エラーコードが表示されている場合:FAL エラーコードの内容を確認してください。(表 4. 7-1 参照) またシステムエラーコードが表示されている場合:OS のシステムエラーコードが示すエラー内容を確認してください。
-151 **	Output file : Open error (mto 転送実行時、出力ファイルのオープンに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-152	Output file : Get file data error (mto 転送実行時の出力ファイルのデータ取得エラー) ⇒ トラブル発生時の情報収集をしてください。(10 節参照)
-153	Processing data : Length check error (mto 時の処理データ長不一致) ⇒ トラブル発生時の情報収集をしてください。(10 節参照)
-155	Buffer : Memory allocation error (バッファメモリのアロケーションに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-160 **	Input file : Read error (mto 転送実行時、入力ファイルのリードに失敗しました) ⇒ FAL エラーコード、システムエラーコードを確認してください。(付録 ログフォーマット例参照) FAL エラーコードが表示されている場合:FAL エラーコードの内容を確認してください。(表 4. 7-1 参照) またシステムエラーコードが表示されている場合:OS のシステムエラーコードが示すエラー内容を確認してください。
-161 **	Output file : Write error (mto 時、出力ファイルのライトに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-162	Output file : Code convert error (mto 時、出力ファイルのコード変換に失敗しました) ⇒ FAL のエラーコードを参照し、表 4. 7-1 によりエラーの内容を確認してください。
-163	Get processing data error (プロセッサデータの取得に失敗しました) ⇒ FAL のエラーコードを参照し、表 4. 7-1 によりエラーの内容を確認してください。
-165	Dataset error : Invalid data (データセット中にレコード長が不当なデータがあります) ⇒ データセットを再作成してください。

表8. 8-1 FCU (LISTVOL/LISTVOL64)のエラーコード・エラーメッセージ

エラーコード	内 容
-170 **	Input file : Close error (入力ファイルのクローズに失敗しました) ⇒ FAL エラーコード、システムエラーコードを確認してください。(付録 ログフォーマット例参照) FAL エラーコードが表示されている場合:FAL エラーコードの内容を確認してください。(表 4. 7-1 参照) またシステムエラーコードが表示されている場合:OS のシステムエラーコードが示すエラー内容を確認してください。
-171 **	Output file : Close error (出力ファイルのクローズに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-180	UNIX file : Invalid directory name (Help メニューの UX-File の指示でディレクトリが見つかりません) ⇒ ディレクトリ名の指定を確認してください。(8. 6. 1 節参照)
-181	UNIX file : Not a directory (Help メニューの UX-File の指示でディレクトリでないものがディレクトリとして指定されました) ⇒ ディレクトリ名の指定が正しいか確認してください。
-182 **	UNIX file : Open Directory error (Help メニューの UX-File の指示でディレクトリのオープンに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-183	UNIX file : Close directory error (Help メニューの UX-File の指示でディレクトリのクローズに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-190	Input file name : No data (パラメタの Save が指示されましたが、入力ファイル名が指定されていません) ⇒ 入力ファイル名を入力してから Save を指示して下さい。
-191	Output file name : No data (パラメタの Save が指示されましたが、出力ファイル名が指定されていません) ⇒ 出力ファイル名を入力してから Save を指示して下さい。
-192	Parameter file name : No data (パラメタ定義ファイル名が指定されていません) ⇒ パラメタ定義ファイル名を指定して下さい。
-200	Parameter file : End line (パラメタファイルの最終行に到達しました) ⇒ 正常に終了しました。FCU を終了してください。
-201	Parameter : Direction error (パラメタのデータ転送方向が正しくありません) ⇒ データ転送方向は「mto」または「otm」である必要があります。データ転送方向が「mto」または「otm」となっているか確認してください。(8. 4 節参照)
-202	Parameter file : Too many data (パラメタの個数が多すぎます) ⇒ パラメタ数を規定値以下としてください。(8. 4 節参照)
-203	Parameter : Empty select error (パラメタ「Emp=」の指定が正しくありません) ⇒ パラメタ「Emp=」は「Emp=Yes」または「Emp=No」である必要があります。パラメタ「Emp=」が「Emp=Yes」または「Emp=No」となっているか確認してください。(8. 4 節参照)
-204	Parameter : RDW select error (パラメタ「RDW=」の指定が正しくありません) ⇒ パラメタ「RDW=」は「RDW=Yes」または「RDW=No」である必要があります。パラメタ「RDW=」が「RDW=Yes」または「RDW=No」となっているか確認してください。(8. 4 節参照)
-205	RDW error : CODE_CONV unsupported (「RDW=Yes」でコード変換指定が「No」でない) ⇒ パラメタ「RDW=Yes」を指定した場合コード変換は「No」を指定して下さい。(8. 4 節参照)
-206	RDW error : PADDING unsupported (「RDW=Yes」でパディング指定が「No」でない) ⇒ パラメタ「RDW=Yes」を指定した場合パディングは「No」を指定して下さい。(8. 4 節参照)

表8. 8-1 FCU (LISTVOL/LISTVOL64)のエラーコード・エラーメッセージ

エラーコード	内 容
-207	RDW error : DELIMITER unsupported (「RDW=Yes」でデリミタ指定が「No」でない) ⇒ パラメタ「RDW=Yes」を指定した場合デリミタは「No」を指定して下さい。(8.4節参照)
-210	Parameter file : Comment line (パラメタファイルのコメント行を検出しました) ⇒ 異常ではありません。Loadを指示して次の処理を行ってください。(8.6.1節参照)
-220	Parameter : VSE select error (VSEパラメタが正しくありません) ⇒ VSEパラメタの個数とパラメタ間のカマの有無を確認してください。(8.4節参照)
-221	Parameter : VSE record format error (VSEパラメタのレコード形式が正しくありません) ⇒ VSEパラメタのレコード形式を確認してください。(8.4節参照)
-222	Parameter : VSE record length error (VSEパラメタのレコード長が正しくありません) ⇒ VSEパラメタのレコード長が規定の値になっているか確認してください。(8.4節参照)
-223	Parameter : VSE block length error (VSEパラメタのブロック長が正しくありません) ⇒ VSEパラメタのブロック長が規定の値になっているか確認してください。(8.4節参照)
-230	No codeconv table file (コード変換テーブルファイルが見つかりません) ⇒ 指定したコード変換テーブルファイルが格納されていることを確認してください。
-231	Codeconv table : Open error (コード変換テーブルファイルのオープンに失敗しました) ⇒ OSでエラーを検出しています。OSのシステムエラーコードが示すエラー内容を確認してください。
-233	Codeconv table : Close error (コード変換テーブルファイルのクローズに失敗しました) ⇒ OSでエラーを検出しています。OSのシステムエラーコードが示すエラー内容を確認してください。
-234	Codeconv table : Get file data error (コード変換テーブルファイルのサイズ取得に失敗しました) ⇒ OSでエラーを検出しています。OSのシステムエラーコードが示すエラー内容を確認してください。
-235	Codeconv table : File size error (コード変換テーブルファイルのサイズが正しくありません) ⇒ コード変換テーブルファイルのサイズを確認してください。(7.3節参照)
-236	Codeconv table function : Invalid argument (コード変換元データがありません) ⇒ トラブル発生時の情報収集をしてください。(10節参照)
-238	Codeconv table name : No data (コード変換テーブルファイル名が指定されていません) ⇒ コード変換テーブルファイル名を指定してください。(8.6.1節参照)
-250	Named pipe : Create error (Named PIPEの作成に失敗しました) ⇒ OSでエラーを検出しています。OSのシステムエラーコードが示すエラー内容を確認してください。
-251	Parameter : PIPE error (パラメタ「PIPE=」の指定が正しくありません) ⇒ パラメタ「PIPE=」は「PIPE=Yes」または「PIPE=No」である必要があります。パラメタ「PIPE=」が「PIPE=Yes」または「PIPE=No」になっているか確認してください。 otmでは「PIPE=No」を指定するか、「PIPE=」の指定を削除して下さい。
-252	Named pipe : Time out (パイプ処理がタイムアウトしました) ⇒ パイプを経由してデータを受け取るアプリケーションプログラム/ユーティリティプログラムが起動されているか確認してください。 また、環境変数の「WAIT_TIME_VALUE」の設定値を見直してください。(7.2節参照)
-253	Named pipe : Wait time out value error (パイプ処理のタイムアウトの設定値が正しくありません) ⇒ 環境変数の「WAIT_TIME_VALUE」の値を確認して下さい。(7.2節参照)
-254	Named pipe : Select error (パイプ処理に失敗しました) ⇒ OSでエラーを検出しています。OSのシステムエラーコードが示すエラー内容を確認してください。

表8.8-1 FCU (LISTVOL/LISTVOL64)のエラーコード・エラーメッセージ

エラーコード	内 容
-260 (K)	Parameter : CODE_CONVE select error(USER_EDIT) (USER-EDIT 時のコード変換指定が正しくありません) ⇒ パラメタ「USER-EDIT」を指定した場合コード変換は「No」を指定してください。(8.4 節参照)
-261 (K)	Parameter : Delimiter select error(USER_EDIT) (USER-EDIT 時のデリミタ指定が正しくありません) ⇒ パラメタ「USER-EDIT」を指定した場合デリミタは「No」を指定してください。(8.4 節参照)
-263 (K)	Parameter : RDW select error(USER_EDIT) (パラメタUSER-EDIT 時のRDW 指定が正しくありません) ⇒ パラメタ「USER-EDIT」を指定した場合は「RDW=No」を指定してください。(8.4 節参照)
-266 (K)	Edit log buf : Allocation error (バッファの確保に失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-268 (K)	OtoM Edit log buf : Allocation error (バッファの確保に失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-270 (K)	UOC error : Record Length Error (HMDE/KANJI で変換したレコード長が正しくありません) ⇒ HMDE/KANJI で変換したレコード長がデータセットのレコード長を越えています。HMDE/KANJI のフィールド定義ファイルの内容を確認してください。(HMDE/KANJI ユーザーズガイド参照)
-271 (K)	Parameter : UOC I/O Mode select error (HMDE/KANJI 連携でエラーが発生しました) ⇒ トラブル発生時の情報収集をしてください。(10 節参照)
-272 (K)	UOC error : Invalid delimiter or padding value (HMDE/KANJI のパディング/デリミタ指定が正しくありません) ⇒ HMDE/KANJI のフィールド定義ファイルのパディング/デリミタ指定を確認してください。(HMDE/KANJI ユーザーズガイド参照)
-273 (K)	UOC error : Version file open error (バージョンファイルのオープンに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-274 (K)	UOC error : Version file read error (バージョンファイルのリードに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-275 (K)	UOC error : Version file close error (バージョンファイルのクローズに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-300	Data error : Invalid record length (otm 転送(パディング指定あり)実行時のレコード長が正しくありません) ⇒ 入力ファイルのレコード長がデータセットのレコード長より短いことを確認して下さい。(7.4 節参照)
-301	Dataset error : Invalid record format (otm 転送(パディング指定あり)実行時のレコード形式が正しくありません) ⇒ データセットのレコード形式を確認してください。(7.4 節参照)
-302	Parameter : Delimiter error (otm 転送(パディング指定あり)実行時のデリミタ指定が正しくありません) ⇒ otm 転送(パディング指定あり)時はデリミタ「No」は指定できません。デリミタ指定を確認してください。(7.4 節、8.4 節参照)
-319 **	Dataset : Open error (データセットのオープンに失敗しました) ⇒ FAL エラーコード、システムエラーコードを確認してください。(付録 ログフォーマット例参照) FAL エラーコードが表示されている場合:FAL エラーコードの内容を確認してください。(表 4.7-1 参照) またシステムエラーコードが表示されている場合:OS のシステムエラーコードが示すエラー内容を確認してください。 (例)ボリューム定義ファイル中のパーティション名が実際のパーティション名と違っている場合、エラーコード=6 (No such device)が発生します。

表 8. 8-1 FCU (LISTVOL/LISTVOL64) のエラーコード・エラーメッセージ

エラーコード	内 容
-321	UNIX file : File name error (otm 転送実行時に入力ファイルの種別がパイプとなっています) ⇒ 入力ファイルの種別を確認してください。
-324	OtoM error : RDW unsupport (otm 転送実行時に「RDW=Yes」が指定されています) ⇒ otm 転送実行時は「RDW=No」である必要があります。パラメ「RDW=No」を指定してください。(8.4 節参照)
-340	Dataset error : Invalid organization type (データセットの編成種別が正しくありません) ⇒ データセットの編成種別を確認してください。(1.2.2 節参照)
-341	Dataset error : Invalid record format (データセットのレコード形式が正しくありません) ⇒ データセットのレコード形式を確認してください。(1.2.2 節参照)
-342	Dataset error : Invalid block length (データセットのブロック長が正しくありません) ⇒ データセットのブロック長を確認してください。(1.2.2 節参照)
-343	Dataset error : Invalid record length (データセットのレコード長が正しくありません) ⇒ データセットのレコード長を確認してください。(1.2.2 節参照)
-350 **	Input file : Open error (otm 転送実行時、入力ファイルのオープンに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-351 **	Output file : Open error (otm 転送実行時、出力ファイルのオープンに失敗しました) ⇒ FAL エラーコード、システムエラーコードを確認してください。(付録 ログフォーマット例参照) FAL エラーコードが表示されている場合:FAL エラーコードの内容を確認してください。(表 4.7-1 参照) またシステムエラーコードが表示されている場合:OS のシステムエラーコードが示すエラー内容を確認してください。
-352	Input file : Get file data error (otm 転送実行時、入力ファイルのデータ取得に失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-353	Processing data : Length check error (otm 転送実行時、処理データ長が正しくありません) ⇒ トラブル発生時の情報収集をしてください。(10 節参照)
-355	Buffer : Memory allocation error (メモリの確保に失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-360 **	Input file : Read error (入力ファイルのリードに失敗しました) ⇒ FAL エラーコード、システムエラーコードを確認してください。(付録 ログフォーマット例参照) FAL エラーコードが表示されている場合:FAL エラーコードの内容を確認してください。(表 4.7-1 参照) またシステムエラーコードが表示されている場合:OS のシステムエラーコードが示すエラー内容を確認してください。
-361 **	Output file : Write error (出力ファイルのライトに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-362	Output file : Code convert error (コード変換に失敗しました) ⇒ FAL エラーコード、システムエラーコードを確認してください。(付録 ログフォーマット例参照) FAL エラーコードが表示されている場合:FAL エラーコードの内容を確認してください。(表 4.7-1 参照)
-363	Get processing data error (データ転送の進捗度を表示する処理でエラーが発生しました) ⇒ トラブル発生時の情報収集をしてください。(10 節参照)

表8. 8-1 FCU (LISTVOL/LISTVOL64)のエラーコード・エラーメッセージ

エラーコード	エラーメッセージ
-370 **	Input file : Close error (入力ファイルのクローズに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-371 **	Output file : Close error (出力ファイルのクローズに失敗しました) ⇒ FAL エラーコード、システムエラーコードを確認してください。(付録 ログフォーマット例参照) FAL エラーコードが表示されている場合:FAL エラーコードの内容を確認してください。(表 4. 7-1 参照) またシステムエラーコードが表示されている場合:OS のシステムエラーコードが示すエラー内容を確認してください。
-379 **	UNIX file : No data (オープンシステムファイルにデータがありません) ⇒ パラメタに指定したオープンシステムファイル名が正しいか確認してください。またデータを作成してください。
-380	No UNIX file (指定した入力ファイルがありません) ⇒ 入力ファイル名を確認してください。また入力ファイルが存在するか確認してください。
-381 **	UNIX file : Open error (入力ファイルのオープンに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-382	Output file : Unsupported record format (出力ファイルのレコード形式が正しくありません) ⇒ 指定したデータセットのレコード形式を確認してください。(1. 2. 2 節参照)
-383 **	Input file : Invalid format (入力ファイルのレコード長が正しくありません) ⇒ 入力ファイルのレコード長がデータセットのレコード長よりも短くなっています。入力ファイルのデリミタ位置を確認してください。
-384 **	Input file : Invalid delimiter position (入力ファイルのレコード長が正しくありません) ⇒ 入力ファイルのレコード長がデータセットのレコード長を超えています。入力ファイルのデリミタ位置を確認してください。
-385	Input file : File seeking error (入力ファイルの検索に失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-390	Input File name : Length error (パラメタ定義ファイル内の入力ファイル名の長さが正しくありません) ⇒ パラメタ定義ファイル内の入力ファイル名を確認してください。(8. 4 節参照)
-391	Output File name : Length error (パラメタ定義ファイル内の出力ファイル名の長さが正しくありません) ⇒ パラメタ定義ファイル内の出力ファイル名を確認してください。(8. 4 節参照)
-392	Codeconv table : Length error (パラメタ定義ファイル内のコード変換ファイル名の長さが正しくありません) ⇒ パラメタ定義ファイル内のコード変換ファイル名を確認してください。(8. 4 節参照)
-393	VSE Parameter : Length error (パラメタ定義ファイル内の VSE パラメタの長さが正しくありません) ⇒ パラメタ定義ファイル内の VSE パラメタを確認してください。(8. 4 節参照)
-394	Partition name : Length error (ボリューム定義ファイル内のパーティション名の長さが正しくありません) ⇒ ボリューム定義ファイル内のパーティション名を確認してください。(8. 2 節参照)
-395	VSN : Length error (ボリューム定義ファイル内の VSN の長さが正しくありません) ⇒ ボリューム定義ファイル内の VSN を確認してください。(8. 2 節参照)
-396	Emulation type : Length error (ボリューム定義ファイル内のエミュレーションタイプが正しくありません) ⇒ ボリューム定義ファイル内のエミュレーションタイプを確認してください。(8. 2 節参照)
-397	Volume definition file : Length error (ボリューム定義ファイルの 1 行の文字数が正しくありません) ⇒ ボリューム定義ファイルの 1 行の文字数を確認してください。(8. 2 節参照)

表 8.8-1 FCU (LISTVOL/LISTVOL64) のエラーコード・エラーメッセージ

エラーコード	エラーメッセージ
-398	Parameter file : Length error (パラメタ定義ファイルの 1 行の文字数が正しくありません) ⇒ パラメタ定義ファイルの 1 行の文字数を確認してください。(8.4 節参照)
-399	Volume definition : VSN identification length error (ボリューム定義ファイル内の VSN 識別子の長さが正しくありません) ⇒ ボリューム定義ファイル内の VSN 識別子を確認してください。(8.2 節参照)
-400	Parameter : Invalid input file name (コード変換指定は無効です) ⇒ トラブル発生時の情報収集をしてください。(10 節参照)

補足：UNIX系OSシステムエラーコード一覧

errno.hに記載のものと、異なる意味合いの場合があります。

エラーコード	意味	備考
5 EIO	Solarisの場合、パーティションの設定を実際のリサイズより小さくした場合に発生します。⇒フォーマットコマンドにてたたくパーティションの設定を行ってください。	Otm時 FCU -351 Mto時 FCU -160
	AIXでかつFibreの場合root以外のユーザーで、FAL/FCUを動かすと発生します。⇒AIXのパッチが必要です。OSメーカーに問い合わせして下さい。	FCU -138
	ボリューム定義ファイルのエミュレーションタイプが間違えています。⇒ SANRISE側及びUSP VまたはVSP, VSP G1000, VSP 5000側のディスクタイプとボリューム定義ファイルのエミュレーションタイプを合わせてから再度転送を行ってください。	
6 ENXIO	ボリューム定義ファイルのエミュレーションタイプが間違えています。⇒ SANRISE側及びUSP VまたはVSP, VSP G1000, VSP 5000側のディスクタイプとボリューム定義ファイルのエミュレーションタイプを合わせてから再度転送を行ってください。	
	共有メモリとして確保可能な最大値/最小値の範囲が小さい場合発生します。FAL/FCUでは8~1048576byteの共有メモリを確保します。この値が最大値/最小値の範囲内に収まるように、共有メモリとして確保できる最小/最大値を変更して下さい。	
22 EINVAL	ボリューム定義ファイルのエミュレーションタイプが間違えています。⇒ SANRISE側及びUSP VまたはVSP, VSP G1000, VSP 5000側のディスクタイプとボリューム定義ファイルのエミュレーションタイプを合わせてから再度転送を行ってください。	
	共有メモリとして確保可能な識別子の数を超えた場合に発生します。FAL/FCUでは3つの共有メモリを確保しますので多重起動する場合又は他のアプリケーションを同時に起動する場合は、共有メモリとして確保できる識別子の数を変更して下さい。	
28 ENOSPC	共有メモリとして確保可能な識別子の数を超えた場合に発生します。FAL/FCUでは3つの共有メモリを確保しますので多重起動する場合又は他のアプリケーションを同時に起動する場合は、共有メモリとして確保できる識別子の数を変更して下さい。	

8.8.1.1 UNIX系OSその他のエラーメッセージ一覧

以下のメッセージは画面に出力されます。

表8.8.1-1 FCUのエラーメッセージ

項番	メッセージ	発生条件	対処方法	備考
1	Parameter data number is invalid	転送パラメータを-Pで直接指定する際に、-Pの後に記述するパラメータの個数が異なります	-Pの後に正しく転送パラメータを設定して下さい 例) USER-EDIT 指定が間違えていないか等	
2	Input File name : Length error	転送パラメータを-Pで直接指定する際に、入力ファイル名が長すぎます	入力ファイル名を1024文字以内にして下さい	
3	Output File name : Length error	転送パラメータを-Pで直接指定する際に、出力ファイル名が長すぎます	出力ファイル名を1024文字以内にして下さい	
4	Codeconv table : Length Error	転送パラメータを-Pで直接指定する際に、コード変換ファイル名が長すぎます	コード変換ファイル名を1024文字以内にして下さい	
5	VSE Parameter : Length Error	転送パラメータを-Pで直接指定する際に、VSEパラメータの指定が長すぎます	VSEパラメータ指定を20文字以内にして下さい	
6	Parameter filename is too long	FCUコマンドに渡すパラメータ定義ファイル名が長すぎます	パラメータ定義ファイル名を1024文字以内にして下さい	
7	Table : Memory allocation error	FCU内部で使用するメモリを確保することができません	十分なメモリを確保してから再度実行して下さい	
8	Couldn't write this message to 'FALエラーログファイル名'	FALエラーログファイルを開くことができません	FALエラーログファイルが正しく開ける状態になっているか確認して下さい 環境変数 ERR_LOG_FILE を指定時は正しいパス名になっているか確認して下さい	
9	Couldn't write UOC error log	UOCログファイルを開くことができません	UOCログファイルが正しく開ける状態になっているか確認して下さい	
10	CodeConvTBLfile: calloc error	コード変換ファイルに使用するメモリを確保することができません	十分なメモリを確保してから再度実行して下さい	FCUエラー(-155)参照
11	dataOffset=数値, st.st_size=数値, ret=数値, offset=数値	MTO転送終了後、入力側データ内のデータバイト数と出力側オープンファイルのバイト数が異なります	エラー情報を収集して問い合わせ下さい エラー情報収集方法 *1	FCUエラー(-153)参照
12	**File IO Record number unmatched	MTO/OTM転送終了後、入力側から読み込んだレコード件数と出力側へ書き込んだレコード件数が異なります	エラー情報を収集して問い合わせ下さい エラー情報収集方法 *1	
13	**KANJI IO Record number unmatched	MTO/OTM転送終了後、UOC変換へ渡したレコード件数とUOC変換後のレコードが異なります	エラー情報を収集して問い合わせ下さい エラー情報収集方法 *1	

表8.8.1-1 FCUのエラーメッセージ

項番	メッセージ	発生条件	対処方法	備考
14	fast_close: error または close	MTO 転送時のエラー処理中に出力側プログラム のクローズに失敗	エラー情報を収集して問い合わせ下さい エラー情報収集方法 *1	
15	st.st_size=数値, dataOffset= 数値, error=数値	OTM転送終了後、入力側プログラムのバイト数 と出力側プログラムへ書き込んだバイト数が 異なる	エラー情報を収集して問い合わせ下さい エラー情報収集方法 *1	
16	st.st_size=数値, dataOffset= 数値, error=数値	OTM転送終了後、入力側プログラムのバイト数 と出力側プログラムへ書き込んだバイト数が異 なる	エラー情報を収集して問い合わせ下さい エラー情報収集方法 *1	
17	close error または fclose	OTM 転送時のエラー処理中に入力側プログラムの クローズに失敗	エラー情報を収集して問い合わせ下さい エラー情報収集方法 *1	
18	-363 error found! Now copying...	OTM 転送時に出力側オープンファイルへ外 部から改ざんが発生	エラー情報を収集して問い合わせ下さい エラー情報収集方法 *1	
19	Couldn't write this message to 'ERR_LOG_FILE の値'	環境変数『ERR_LOG_FILE』に不正なパス 名を指定し、FAL エラーへの書き込みが失敗	適正なパス名を指定して下さい。	

*1：第10章トラブル発生時の情報収集を参照してください。

8.8.2 Windows の場合

表8.8-2 FCU (LISTVOL/LISTVOL64)のエラーコード・エラーメッセージ

エラーコード	内 容
-100	Parameter definition file : Open error (パラメタ定義ファイルのオープンに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-101	Parameter : Count error (パラメタ定義ファイルのパラメタ数が正しくありません) ⇒ パラメタ定義ファイルのパラメタ数を確認してください。(8.4 節参照)
-102	Parameter : Direction error (パラメタ定義ファイルのデータ転送方向が正しくありません) ⇒ データ転送方向は「mto」または「otm」である必要があります。データ転送方向は「mto」または「otm」を指定してください。(8.4 節参照)
-103	Parameter : Main frame file name error (メインフレームファイル名が入力されていません) ⇒ メインフレームファイル名を入力してください。
-104	Parameter : Open system file name error (オープンシステムファイル名が入力されていません) ⇒ オープンシステムファイル名を入力してください。
-105	Parameter : Code conversion error (パラメタのコード変換種別指定が正しくありません) ⇒ オープンシステム-オープンシステム間のコード変換種別の指定は「No」である必要があります。コード変換種別は「No」を指定してください。(8.4 節参照)
-106	Parameter : Padding error (パラメタのパディング指定が正しくありません) ⇒ パディング指定は「Yes」または「No」である必要があります。パディング指定は「Yes」または「No」を指定してください。(8.4 節参照)
-107	Parameter : Delimiter error (パラメタのデリミタ指定が正しくありません) ⇒ デリミタの指定は「CRLF」または「No」のいずれかである必要があります。デリミタの指定は「CRLF」または「No」を指定してください。(8.4 節、8.7 節参照)
-108	Parameter : Add parameter error (パラメタが正しくありません) ⇒ デリミタパラメタ以降のパラメタを確認して下さい。(8.4 節参照)
-109	Parameter : Empty duplication error (パラメタ「Emp=」が複数指定されています) ⇒ パラメタ「Emp=」の指定は1つだけにしてください。(8.4 節参照)
-110	Parameter : RDW duplication error (パラメタ「RDW=」が複数指定されています) ⇒ パラメタ「RDW=」の指定は1つだけにしてください。(8.4 節参照)
-111	Parameter : VSE duplication error (パラメタ「VSE=」が複数指定されています) ⇒ パラメタ「VSE=」の指定は1つだけにしてください。(8.4 節参照)
-112	Parameter : VSE record format error (パラメタ「VSE=」のレコード形式が正しくありません) ⇒ パラメタ「VSE=」のレコード形式を確認してください。(8.4 節参照)
-113	Parameter : VSE record length error (パラメタ「VSE=」のレコード長が正しくありません) ⇒ パラメタ「VSE=」のレコード長を確認してください。(8.4 節参照)
-114	Parameter : VSE block length error (パラメタ「VSE=」のブロック長が正しくありません) ⇒ パラメタ「VSE=」のブロック長を確認してください。(8.4 節参照)
-115	Parameter : VSE count error (パラメタ「VSE=」のパラメタ数が正しくありません) ⇒ パラメタ「VSE=」のパラメタの個数とパラメタ間のカンマの有無を確認してください。(8.4 節参照)

表 8. 8-2 FCU (LISTVOL/LISTVOL64) のエラーコード・エラーメッセージ

エラーコード	内 容
-116	Parameter : VSN unmach error (パラメタ VSN が正しくありません) ⇒ パラメタ VSN の文字数を確認してください。(8.2 節、8.4 節参照)
-117	Parameter : PIPE duplication error (パラメタ「PIPE=」が複数指定されています) ⇒ パラメタ「PIPE=」の指定は1つだけにしてください。
-118	Parameter : PIPE error (パラメタ「PIPE=」の指定が正しくありません) ⇒ otm 転送実行時、「PIPE=」は「No」を指定するか、指定を削除する必要があります。「PIPE=No」を指定するか、「PIPE=」指定を削除してください。(8.4 節参照)
-120	Volume definition file : Open error (ボリューム定義ファイルのオープンに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-121	Volume definition file : Length error (ボリューム定義ファイルの1行の文字数が正しくありません) ⇒ ボリューム定義ファイルの1行の文字数を確認してください。(8.2 節参照)
-122	Volume definition : Pysical drive Length error (ボリューム定義ファイルのパーティション名の文字数が正しくありません) ⇒ ボリューム定義ファイルのパーティション名の文字数を確認してください。(8.2 節参照)
-123	Volume definition : VOLSER Length error (ボリューム定義ファイルの VSN の文字数が正しくありません) ⇒ ボリューム定義ファイルの VSN の文字数を確認してください。(8.2 節参照)
-124	Volume definition : Emulation type length error (ボリューム定義ファイルのエミュレーションタイプが正しくありません) ⇒ ボリューム定義ファイルのエミュレーションタイプを確認してください。(8.2 節参照)
-125	Volume definition : MF type length error (ボリューム定義ファイルの MF タイプが正しくありません) ⇒ ボリューム定義ファイルの MF タイプを確認してください。(8.2 節参照)
-126	Volume definition : VSN identification length error (ボリューム定義ファイルの VSN 識別子の文字数が正しくありません) ⇒ ボリューム定義ファイルの VSN 識別子の文字数を確認してください。(8.2 節参照)
-127	Volume definition : No valid error (ボリューム定義ファイルのパラメタが正しくありません) ⇒ ボリューム定義ファイルを確認してください。(8.2 節参照)
-128	Volume definition : Read error (ボリューム定義ファイルのリードに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-129	Volume definition : Close error (ボリューム定義ファイルのクローズに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-130	Dataset : No dataset error (パラメタで指定した VSN にデータセットがありません) ⇒ パラメタで指定した VSN を確認してください。 FALエラーコード、システムエラーコードを確認してください。(付録 ログフォーマット例参照) FALエラーコードが表示されている場合：FALエラーコードの内容を確認してください。(表4.7-1参照) またシステムエラーコードが表示されている場合：OSのシステムエラーコードが示すエラー内容を確認してください。
-131	Dataset : Search error (パラメタで指定した VSN にデータセットが見つかりません) ⇒ パラメタで指定した VSN を確認してください。 FALエラーコード、システムエラーコードを確認してください。(付録 ログフォーマット例参照) FALエラーコードが表示されている場合：FALエラーコードの内容を確認してください。(表4.7-1参照) またシステムエラーコードが表示されている場合：OSのシステムエラーコードが示すエラー内容を確認してください。

表 8. 8-2 FCU (LISTVOL/LISTVOL64) のエラーコード・エラーメッセージ

エラーコード	内 容
-132	<p>Dataset : Information get error (データセットのオープンに失敗しました) ⇒ FAL エラーコード、システムエラーコードを確認してください。(付録 ログフォーマット例参照) FALエラーコードが表示されている場合：FALエラーコードの内容を確認してください。(表4. 7-1参照) またシステムエラーコードが表示されている場合：OSのシステムエラーコードが示すエラー内容を確認してください。</p>
-133	<p>Dataset : Organization error (データセットの編成種別が正しくありません) ⇒ データセットの編成種別を確認してください。(1. 2. 2節参照)</p>
-134	<p>Dataset : Record format error (データセットのレコード形式が正しくありません) ⇒ データセットのレコード形式を確認してください。(1. 2. 2節参照)</p>
-135	<p>Dataset : Block length error (データセットのブロック長が正しくありません) ⇒ データセットのブロック長を確認してください。(1. 2. 2節参照)</p>
-136	<p>Dataset : Record length error (データセットのレコード長が正しくありません) ⇒ データセットのレコード長を確認してください。(1. 2. 2節参照)</p>
-137	<p>Dataset : Dataset size error (データセットのデータがありません) ⇒ 「Emp=No」を指定していますが、データセットにデータがありません。データを確認してください。 (8. 4 章参照)</p>
-138	<p>Dataset : Close error (データセットのクローズに失敗しました) ⇒ FAL エラーコード、システムエラーコードを確認してください。(付録 ログフォーマット例参照) FALエラーコードが表示されている場合：FALエラーコードの内容を確認してください。(表4. 7-1参照) またシステムエラーコードが表示されている場合：OS のシステムエラーコードが示すエラー内容を確認してください。</p>
-150	<p>Main frame file : Open error (メインフレームファイルのオープンに失敗しました) ⇒ FAL エラーコード、システムエラーコードを確認してください。(付録 ログフォーマット例参照) FALエラーコードが表示されている場合：FALエラーコードの内容を確認してください。(表4. 7-1参照) またシステムエラーコードが表示されている場合：OS のシステムエラーコードが示すエラー内容を確認してください。</p>
-151	<p>Main frame file : Read error (メインフレームファイルのリードに失敗しました) ⇒ FAL エラーコード、システムエラーコードを確認してください。(付録 ログフォーマット例参照) FALエラーコードが表示されている場合：FALエラーコードの内容を確認してください。(表4. 7-1参照) またシステムエラーコードが表示されている場合：OSのシステムエラーコードが示すエラー内容を確認してください。</p>
-152	<p>Main frame file : Write error (メインフレームファイルのライトに失敗しました) ⇒ FAL エラーコード、システムエラーコードを確認してください。(付録 ログフォーマット例参照) FALエラーコードが表示されている場合：FALエラーコードの内容を確認してください。(表4. 7-1参照) またシステムエラーコードが表示されている場合：OSのシステムエラーコードが示すエラー内容を確認してください。</p>
-153	<p>Main frame file : Close error (メインフレームファイルのクローズに失敗しました) ⇒ FAL エラーコード、システムエラーコードを確認してください。(付録 ログフォーマット例参照) FALエラーコードが表示されている場合：FALエラーコードの内容を確認してください。(表4. 7-1参照) またシステムエラーコードが表示されている場合：OS のシステムエラーコードが示すエラー内容を確認してください。</p>
-154	<p>Main frame file : Record format error (メインフレームファイルのレコード形式に対しパディング指示が正しくありません) ⇒ レコード形式が可変長かつデータ転送方向「otm」の場合、パディングは「No」を指定してください。 (8. 4 節参照)</p>
-170	<p>Open system file : Open error (オープンシステムファイルのオープンに失敗しました) ⇒ オープンシステムファイル名を確認して下さい。システムエラーコードが表示されている場合、OS のシステムエラーコードが示すエラー内容を確認してください。</p>

表 8. 8-2 FCU (LISTVOL/LISTVOL64) のエラーコード・エラーメッセージ

エラーコード	内 容
-171	Open system file : Read error (オープンシステムファイルのリードに失敗しました) ⇒ OS のシステムエラーコードが示すエラー内容を確認してください。
-172	Open system file : Write error (オープンシステムファイルのライトに失敗しました) ⇒ OS のシステムエラーコードが示すエラー内容を確認してください。
-173	Open system file : Close error (オープンシステムファイルのクローズに失敗しました) ⇒ OS のシステムエラーコードが示すエラー内容を確認してください。
-174	Open system file : No data error (オープンシステムファイルにデータがありません) ⇒ パラメタに指定したオープンシステムファイル名が正しいか確認してください。またデータを作成してください。
-175	Open system file : Delimiter(CR) position error (オープンシステムファイルのデリミタ (CR) 位置がデータセットのレコード長と異なっています) ⇒ オープンシステムファイルのデリミタ位置とデータセットのレコード長が一致しているか確認してください。(7.5 節参照)
-176	Open system file : Delimiter(LF) position error (オープンシステムファイルのデリミタ (LF) 位置がデータセットのレコード長と異なっています) ⇒ オープンシステムファイルのデリミタ位置とデータセットのレコード長が一致しているか確認してください。(7.5 節参照)
-178	Open system file : Record length error (オープンシステムファイルのデータ長が正しくありません) ⇒ オープンシステムファイルのデータ長がデータセットのレコード長と異なっています。 オープンシステムファイルのデータ長を確認して下さい。(4 節、7 節参照)
-190	Code conversion error (コード変換に失敗しました) ⇒ FAL のエラーコードを参照してください。(表 4.7-1 参照)
-200	Process data get error (データ転送の進捗度を表示する処理でエラーが発生しました) ⇒ トラブル発生時の情報収集をしてください。(10 節参照)
-220	External table file : Open error (コード変換テーブルファイルのオープンに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-221	External table file : Size error (コード変換テーブルファイルが正しくありません) ⇒ コード変換テーブルファイルのサイズを確認してください。(7.3 節参照)
-222	External table file : Read error (コード変換テーブルファイルのリードに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-223	External table file : Close error (コード変換テーブルファイルのクローズに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-240	Parameter : Direction, Padding, Delimiter combination error (パラメタのデータ転送方向/パディング/デリミタの指定が正しくありません) ⇒ パラメタのデータ転送方向/パディング/デリミタの指定を確認してください。(8.4 節参照)

表 8. 8-2 FCU (LISTVOL/LISTVOL64) のエラーコード・エラーメッセージ

エラーコード	内 容
-241	Parameter : Direction, RDW combination error (パラメタのデータ転送方向と「RDW=」の指定が正しくありません) ⇒ パラメタのデータ転送方向「otm」を指定する場合、「RDW=No」を指定してください。(8.4節参照)
-242	Parameter : Code conversion, RDW combination error (パラメタの「RDW=」とコード変換種別が正しくありません) ⇒ パラメタ「RDW=Yes」を指定した場合コード変換種別は「No」を指定してください。(8.4節参照)
-243	Parameter :Padding, RDW combination error (パラメタの「RDW=」とパディングの指定が正しくありません) ⇒ パラメタ「RDW=Yes」を指定した場合パディングは「No」を指定してください。(8.4節参照)
-244	Parameter :Delimiter, RDW combination error (パラメタの「RDW=」とデリミタの指定が正しくありません) ⇒ パラメタ「RDW=Yes」を指定した場合デリミタは「No」を指定してください。(8.4節参照)
-245	Parameter : Specified VOLSER isn't defined in Volume Definition file. (指定した VSN はボリューム定義ファイルに見つかりません) ⇒ パラメタの VSN が、ボリューム定義ファイルに定義されているか確認してください。(8.2節参照)
-260 (K)	Parameter : UOC Edit mode duplication error (UOC 編集モードが複数指定されています) ⇒ パラメタの UOC 編集モードの指定は 1 つだけにしてください。
-261 (K)	Parameter : UOC Edit option duplication error (UOC 編集オプションが複数指定されています) ⇒ パラメタの UOC 編集オプションの指定は 1 つだけにしてください。
-262 (K)	Parameter : UOC I/O mode duplication error (UOC 入出力モードが複数指定されています) ⇒ パラメタの UOC 入出力モードの指定は 1 つだけにしてください。
-263 (K)	Parameter : UOC I/O option duplication error (UOC 入出力オプションが複数指定されています) ⇒ パラメタの UOC 入出力オプション指定は 1 つだけ指定してください。
-264 (K)	Parameter : Add parameter or UOC parameter error (デリミタ指定パラメタ以降の指定が正しくありません) ⇒ パラメタのデリミタ指定以降が正しいか確認してください。(8.4節参照)
-272 (K)	UOC error : Invalid delimiter or padding value (パディング/デリミタ指定が正しくありません) ⇒ パラメタの HMDE/KANJI フィールド定義ファイルのパディング/デリミタ指定を確認してください。
-273 (K)	UOC error : Version file open error (バージョンファイルのオープンに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-274 (K)	UOC error : Version file read error (バージョンファイルのリードに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-275 (K)	UOC error : Version file close error (バージョンファイルのクローズに失敗しました) ⇒ OS でエラーを検出しています。OS のシステムエラーコードが示すエラー内容を確認してください。
-281 (K)	Parameter : Code conversion, UOC Edit mode combination error (パラメタ USER-EDIT 時のコード変換指定が正しくありません) ⇒ パラメタ「USER-EDIT」を指定する場合はコード変換に「No」を指定してください。(8.4節参照)
-282 (K)	Parameter : Delimiter, UOC Edit mode combination error (パラメタ USER-EDIT 時のデリミタ指定が正しくありません) ⇒ パラメタ「USER-EDIT」を指定する場合はデリミタ指定に「No」を指定してください。(8.4節参照)

表 8. 8-2 FCU (LISTVOL/LISTVOL64) のエラーコード・エラーメッセージ

エラーコード	内 容
-284 (K)	Parameter : RDW, UOC Edit mode combination error (パラメタ USER-EDIT 時の「RDW=」指定が正しくありません) ⇒ パラメタ「USER-EDIT」を指定する場合は「RDW=No」を指定してください。(8.4 節参照)
-285 (K)	Parameter : Direction, UOC I/O mode combination error (パラメタのデータ転送方向と UOC 入出力モード指定の組み合わせが正しくありません) ⇒ パラメタで指定したデータ転送方向と UOC 入出力モードを組み合わせ確認してください。(HMDE/KANJI ユーザーズガイド参照)
-300	Parameter definition file : Length error (パラメタ定義ファイルの 1 行の文字数が正しくありません) ⇒ パラメタ定義ファイルを確認してください。(8.4 節参照)
-301	Mainframe file name : Length error (メインフレームファイル名の文字数が正しくありません) ⇒ パラメタ定義ファイルのメインフレームファイル名の文字数を確認してください。(8.4 節参照)
-302	Opensystem file name : Length error (オープンファイル名の文字数が正しくありません) ⇒ パラメタ定義ファイル内のオープンファイル名を確認してください。(8.4 節参照)
-303	Code conversion : Length error (コード変換ファイル名の文字数が正しくありません) ⇒ パラメタ定義ファイルのコード変換ファイル名文字数を確認してください。(8.4 節参照)
-304	VSE : Length error (パラメタ定義ファイル内の VSE パラメタの文字数が正しくありません) ⇒ パラメタ定義ファイルの VSE パラメタの文字数を確認してください。(8.4 節参照)
-305	VSE record-format : Length error (VSE パラメタのレコード形式が正しくありません) ⇒ パラメタ定義ファイルの VSE パラメタのレコード形式を確認してください。(8.4 節参照)
-306	VSE record-length : Length error (VSE パラメタのレコード長が正しくありません) ⇒ パラメタ定義ファイルの VSE パラメタのレコード長を確認してください。(8.4 節参照)
-307	VSE block-length : Length error (パラメタ定義ファイル内の VSE パラメタ中、ブロック長の指定が正しくありません) ⇒ パラメタ定義ファイルの VSE パラメタのブロック長を確認してください。(8.4 節参照)

補足：Windowsにおけるシステムエラー一覧

errono.h winerror.h 記載内容と異なる意味合いの場合があります。(下表参照)

エラーコード	意味	備考
1 EPERM	MTO 時オープン側システムが満杯の時に発生する場合があります。 ⇒サーバーの空き容量を確認して下さい。	FCU-172
2 ENOENT	ファイルが存在しない場合に発生します。何のファイルがないかは、FCU 側エラーコードを参照ねがいます。 ⇒ファイル名の指定に誤りがないか、またファイルが存在するか確認ねがいます。	
5 EIO	書き込もうとしたファイルと同じ名前のディレクトリが存在しているか、権限がないディレクトリまたはファイルに書き込もうとする場合に発生します。	
11 EAGAIN	書き込もうとしたファイルが他で使用中の場合に発生することがあります。	
13 EACCES	アクセス権が与えられていません。使用ファイルのアクセス権を確認して下さい。 また、#50 までは、FCU を多重で起動した時にも発生する場合があります。	

8.8.2.1 Windowsにおけるその他のエラーメッセージ一覧

以下のメッセージは、パラメタ定義ファイル名.logに出力されます。

表8.8.2-1 FCUのエラーメッセージ

項番	メッセージ	発生条件	対処方法	備考
1	**File IO Record number unmatched	MTO/OTM 転送終了後、入力側から読み込んだレコード件数と出力側へ書き込んだレコード件数が異なる	エラー情報を収集して問い合わせ下さい エラー情報収集方法 *1	
2	**KANJI IO Record number unmatched	MTO/OTM 転送終了後、UOC 変換へ渡したレコード件数とUOC 変換後のレコードが異なる	エラー情報を収集して問い合わせ下さい エラー情報収集方法 *1	

*1：第10章トラブル発生時の情報収集を参照してください。

9 使用上の注意事項

- (1) FCUは、メインフレーム側でSAMファイルの作成が完了したか否かをチェックする機能は持っていません。
このため、メインフレーム側ジョブの終了を目視により確認する、あるいはメインフレーム側とオープンシステム側との連絡機構を構築する必要があります。
またオープンシステムからのリザーブよりもメインフレームからのリザーブが優先されます。オープンシステムでリザーブを確保中であってもメインフレームでリザーブを確保してI/Oが可能です。そのため当該ファイル/ボリュームへの競合(メインフレーム-オープンシステム間/オープンシステム-メインフレーム間のFAL/FCU/listvolの同時アクセス)が発生しないように運用して下さい。競合が発生した場合のFCUエラー情報については、8.8節を参照して下さい。
Linux版FCUからOTM転送を行った直後にメインフレーム側から当該データセットに対してI/Oを出す場合、OTM転送処理とメインフレーム側I/O処理の間に60秒以上のインターバルをとって下さい。
- (2) Solarisの場合、FCUの起動時に「libXm. so. xxがない」というエラーが発生することがあります。
上記エラーが発生した場合、Xmライブラリへのパスの設定が必要です。以下の例を参考に設定して下さい。
また、下記の処理終了後、一度ログアウトし、再度ログインして下さい。
 - (A) Cシェルの場合：ホームディレクトリの.cshrcに以下の1行を追加する。
(ファイルが無い場合は作成する)

```
setenv LD_LIBRARY_PATH /usr/dt/lib:$LD_LIBRARY_PATH
```
 - (B) Cシェル以外の場合：ホームディレクトリの.dtprofileに以下の2行を追加する。
(ファイルが無い場合は作成する)

```
LD_LIBRARY_PATH=/usr/dt/lib:$LD_LIBRARY_PATH  
export LD_LIBRARY_PATH
```注：共通デスクトップ環境がないマシンでは、「.profile」に上記行を追加してください。  
ファイルがない場合は作成してください。
- (3) FCUで処理するUNIXファイルを外付けのディスクに置く場合、FCUを実行する前にマウント処理を済ませておいて下さい。FCU実行中にマウント処理を行うとデータ交換処理の正常性は保証できません。(FCUで異常を検出する場合としない場合があります)
- (4) HPの時、PB-BUSの装置でFAL/FCUを動作させないでください。データの保証ができません。
- (5) FCUはVTOC以外のメインフレーム側の管理情報は参照しません。
従って、カタログ等によるアクセス制御、機密保護等の機能は作用しません。
- (6) FCUは、索引VTOCをサポートしていませんので索引を利用した効率的な処理はできません。
ただし、VTOCの全領域をシケンシャルにアクセスしてデータセットのサーチをおこないますので、索引VTOCで管理されているデータセットにもアクセスは可能です。
- (7) 「空白」付きのデータセット名称はサポートしていません。「空白」付きのデータセット名称を持つデータセットが処理の対象となった場合、「空白」までの文字列をデータセット名称として処理します。
- (8) オープンシステムがSolarisの場合、定義するパーティションのサイズはメインフレーム側のボリュームのサイズと一致させて下さい。パーティションのサイズがメインフレーム側のボリュームより小さい場合、ボリュームの最後までアクセスすることが出来なくなります。
- (9) ユーザID=root(スーパーユーザ)で共用ボリューム(3390-3B等)のインストールを行った状態のままでは一般ユーザはFCUを使用することができません(スーパーユーザは使用可能)。一般ユーザにFCUを使用させたい場合は共用ボリュームの所有者(root)と当該ユーザとでグループを構成し、共用ボリュームに対するアクセス権を当該ユーザに与えて下さい。

- (10) UNIX版で操作画面を使用する指定でFCUを起動すると以下のワーニングメッセージが出力されることがありますが、FCUの実行には全く問題がありませんので無視して下さい。

Warning: Missing charsets in String to FontSet conversion

Warning: Cannot convert string “-dt-interface system-medium-r-normal-m*-*-*-*-*-*-*” to type FontSet

- (11) FAL/FCU (Windows版を除く) のプロセスに対して以下のシグナルが送信されると、プロセスが異常終了して他のFAL/FCUプロセスが残存し、また、保持していた共有メモリが解放されないままメモリ空間を占有し続けることがあります。この場合、killコマンドを用いてプロセスを、ipcrmコマンドを用いてKEY=0の共有メモリを削除して下さい。killおよびipcrmの使用方法については、各OSのマニュアルを参照して下さい。

SIGUSR1, SIGUSR2, SIGILL, SIGTRAP, SIGIOT, SIGABRT, SIGEMT, SIGFPE, SIGKILL, SIGBUS, SIGSEGV, SIGSYS, SIGALRM, SIGPOLL, SIGIO, SIGSTOP, SIGTSTP, SIGCONT, SIGTTIN, SIGTTOU, SIGVTALRM, SIGPROF, SIGXCPU, SIGXFSZ, SIGWAITING, SIGLWP, SIGFREEZE, SIGTHAW, SIGCANCEL

- (12) WindowsNTのバージョン 01-01-34より古いバージョンを既にお使い頂いている場合、01-01-34以降の新しいバージョンにアップする際、次の点に特にご注意ください。

- ① デフォルトのパラメタ定義ファイルの名称が「fcudata.param」→「fcudata.prm」と新しくなっています。今お使いのパラメタ定義ファイルの名称を新しい名称に合わせて変更(リネーム)してください。操作画面の表示が必須となり、起動パラメタで「-nw」の指定が無くなりました。
- ② エラーコードの体系が新しくなりました。またログファイルも、FCUに関する情報がより詳細になり、独立したファイルに出力するよう構成を新しくしています。ご注意ください。
- ③ 連続実行(Continuous execution)時、途中結果(個々の正常/異常)を目視確認しながら作業したい場合は、Error Informationダイアログを予め開いて、画面上FCU本体画面と重ならない見易い位置に配置してください。(連続実行の最終結果のみ知りたい場合、本操作は不要です。)
- ④ 3.2.2節の(2)に示したアンインストールの操作は、01-01-34以降の新しいバージョンからさらに新しいバージョンにアップする際に有効な方法を示しています。01-01-34より古いバージョンからアップする際には、FCUのインストール先ディレクトリから、Delコマンドにより不要ファイルを削除してください。
- ⑤ [-nc]が指定されていない場合でも、次の簡単な操作によりデフォルトのパラメタ定義ファイル「fcudata.prm」を使って動作することができます。

→ FCUの起動に続き、**Openボタン**  を使ってファイル参照ダイアログから「fcudata.prm」を選択してください。

(デフォルトで最初、FCUのインストール先ディレクトリに存在する全てのパラメタ定義ファイル「xxxxxx.prm」を表示していますので、その中にある「fcudata.prm」がすぐに選べます。)

- (13) オープンシステム—オープンシステム間のデータ交換時は、コード変換をサポートしていません。

コード変換パラメタは、必ず「No」を指定して下さい。

コード変換、ファイル名を指定するとエラーになります。

- (14) 中間ボリュームに対するクラス構成(WindowsのMSCS及びAIXのHACMPを除く)/交替バス構成(HDLMを除く)の動作保証はできません。クラスリングソフトを使用しフェイルオーバーした場合、転送中のデータに関してはデータ保証できません。データ転送を再実行して下さい。

- (15) Windowsより中間ボリュームへの書き込み後、本ボリュームにアクセスする場合は、1~3秒程度のデレイを持たせて下さい。デレイの時間はサーバーの性能により異なります。(NTではNTキャッシュ上に書き込み後、HDにデータが書き込まれるまでに時間がかかることがあります。)

- (16) Windows上で、mtoを行うオープン側ボリュームの拡張を行う場合には必ず、装置のリポートを行って下さい。
(mtoした後、ボリュームの拡張を行い装置のリポートを行わないと、データを他装置(ネットワーク接続装置等)から更新すると、そのデータを読み出した時、変更が反映されない場合があります。)
- (17) FCUでパイプ出力機能を使用してデータを転送し、転送結果に不具合が生じた場合、次の操作でFCUが出力したパイプ上のデータ (UNIX系OS上のアプリケーションプログラム/ユーティリティプログラムが加工していないデータ) を取得し、その内容に問題がないか確認して下さい。
- ① FAL/FCUで、パイプ出力機能を使用したデータ転送を実行する
② UNIX OSの標準コマンド“dd”を使用して、パイプ上のデータをファイルに格納する
- 例：“dd if=pipe-file of=outfile“
pipe-file : FCUのパイプ出力時に指定したパイプ名
outfile : パイプ上のデータを格納するファイル名
- (18) バージョン01-XX-YY/ZZ (YY=数字, ZZ<20) (Largeファイル未サポート版)のFAL/FCUで、2Gbyte以上のファイル进行处理することはできません。(異常終了を返します)
- (19) ボリュームシリアル番号やデータセット名に使用する“¥”と“\”の文字は、共にASCIIコードでは16進数で(5C)です。(日本語環境では“¥”、英語環境では“\”になります)
- (20) FAL/FCUにて使用する共有メモリ (shmget ()による)のサイズは8byte~1048576byteです。
Solarisの場合、/etc/systemファイル内の下記設定をする場合は、YYYY≤8, XXXXX≥1048576となるようにしてください。(デフォルトの範囲内であるため下記設定は特に必要はありません。)
set shm:shminfo_shmmax=XXXXX
set shm:shminfo_shmmin=YYYYY
- (21) UNIX SVR4(NCR)の時、「ulimit」のデフォルト値が65536ブロックとなっており、32Mbyteより大きいファイルにアクセスできません。それ以上のサイズの転送を行う場合には、「ulimit」のパラメータを無効にして下さい。
- (22) 中間ボリュームに対するクラスタ構成(WindowsのMSCS及びAIXのHACMPを除く)/交替パス構成(HDLMを除く)の動作保証はできません。
・拡張VTOCはサポートしません。
・パスポートのついているボリューム/データセットに対しても、パスポートのチェックなしでOTM/MTO可能となります。
・LISTVOL/LISTVOL64/FCUにおいて、レポート形式の表示は、F/Vのみです。
・LISTVOL/LISTVOL64/FCUにおいて、データセット編成種別がSAM以外の時、TT R EX (Cyl:Trk)の表示内容は保証できません。
- (23) 01-01-42/01以前のバージョンでは可変長でデータ実体の無い(レポート長=4)データを含むデータセットは扱えませんでした。以降のバージョンでは実体なしのデータを含むデータセットも扱えるようになりました。
- (24) ボリューム定義ファイル上の記載に誤りがある時、それ以降の行に記載されているボリュームに対し、fcu (otm&mto)/listvol/フォーマット/アロケータ等の操作は行えません。
- (25) 中間ボリュームはメインフレームとオープンシステム間で共用するファイルを確認する為のボリュームです。中間ボリューム上のデータはメインフレーム側より直接更新しないで下さい。特にメインフレームのOSがVOS1の場合、下記制限があります。
・中間ボリュームにOTMしたデータは、FACYで別ボリュームにコピーするか、データ管理を用いて作成したUPにてデータの読み込みを行ってください。中間ボリューム上のデータセットは直接拡張処理を行えません。(FDUMPによるコピー/バックアップ後のデータも拡張処理は行えません)。

- (26) FALにてアクセス中の中間ボリュームに対し、他（メインフレーム/他オープンシステムまたは同一システム上の他スロット/プロセス）からの処理は保証していません。アクセス中の中間ボリュームに対する他からの処理は行わないでください。ただし、AIXは除いてMTO同士であれば同一ボリュームへの競合は可能です（AIXで他サーバと中間ボリュームを共用する場合はバージョン01-XX-60 (XX≥03)以降で環境変数を定義することで可能となります）。UNIX系OSにおいて、バージョン01-XX-48以降は、他データセットであれば、AIXは除いてOTMの競合も可能です。ただし、同一のAIXにおいては、他データセットのOTMの競合は可能です（AIXで他サーバと中間ボリュームを共用する場合はバージョン01-XX-60以降 (XX≥03)で環境変数を定義することで可能となります）。Windows系OSにおいては、バージョン01-XX-50以降は、他データセットであれば、OTMの競合も可能です。
- (27) メインフレームOSが、MVS/MSP/VOS3/VSE/VOS1の場合、同一データセットに対しデータを入力する場合、メインフレームからの書き込み時より、otmでデータを入力したほうが書き込み可能レコード数が少なくなる場合があります。（otmでは、格納可能ブロックを入れたあとの残エリアに格納可能なレコードを短ブロック化して格納していません。）
- (28) HMDE/KANJI連携時の注意事項を以下に示します。
- ① HMDE/KANJI 変換後/変換前のデータサイズは1レコード 64kbyte 以内になるよう設定してください。
 - ② 固定長レコードに OTM する時、旧 KANJI(01-03/XX 以前)か HMDE/KANJI 側にパディング指定がないとき、デリミタ位置が固定でないと処理できません。またその時、デリミタを除いたサイズ (HMDE/KANJI 変換前)がレコード長より短くないと処理できません。
 - ③ HMDE/KANJI は FAL/FCU をインストールした後にインストールして下さい。
FAL/FCU のみバージョンアップする時は、3.2 章を参照して下さい。HMDE/KANJI 用のライブラリ (UNIX: /usr/lib/libuoc.*, NT:インストール先¥uoc.dll) を別名にコピーしておき、FAL/FCU インストール後、元に戻して下さい。（FAL/FCU をインストールすると、タミのライブラリをコピーしてしまい、この状態では HMDE/KANJI は動作しません）本操作を忘れた場合、再度 HMDE/KANJI をインストールして下さい。
 - ④ ライセンスコード入力前に MTO を行なうと 0byte ファイルが作成されます。その場合、ライセンスキーを入力後に再度 MTO を行なって下さい。
- (29) 中間ボリュームに対しオープン側から、FAL 以外のアクセスは行わないでください。（たとえば、dd コマンド）
- (30) メインフレーム側 OS が VSE の時、OTM したデータをメインフレーム側にて別ボリュームにコピーした後でも、DFSORT は行えません。（Ditto によりコピーすれば可能です。）
- (31) FAL は RAW デバイスアクセスを使用しています。オープン側 OS が AIX4.3.3 で Fibre 接続の場合、root 以外のユーザーから中間ボリュームのアクセスを行う時には、AIX のパッチ (APAR:IY11245) が必要です。
- (32) Windows2000 の場合、中間ボリューム(エミュレーションタイプ が 3390-3x/3390-9x/3390-Lx/3390-Mx/3380-3x /3380-Kx/6588-3x/6588-9x/6588-Lx/6586-Kx(x=A,B,C))に対しディスクの署名を行なわないで下さい。
Windows2000 で中間ボリュームに対してディスクの署名を行なった場合は、他 OS と中間ボリュームの共用使用は保証できません。
また、Windows2000 に中間ボリュームが単独で接続され(他 OS との共用無し)、中間ボリュームに対してディスクの署名を行なった場合は、system log に write エラーが発生しますが、FAL/FCU の動作上、問題ありません。
MSCS 構成を組む場合、Windows2000 Service Pack 1 以降が必須です。もし Service Pack 1 をインストールしないと、署名なしのディスクを MSCS 構成配下で扱えません。Service Pack 1 の扱いに関しては別途相談願います。
- (33) Windows で、MSCS 構成を組む場合、中間ボリュームに対し署名書きを行う必要があります。
書かないと中間ボリュームにアクセスできません。

- (34) Solaris で OTO する場合、OPEN-L/M/V 及び LUSE で 32768 シンク以上のボリュームは、他プラットフォームで使用可能なシンク数よりも小さくなります。
Solaris に合わせて、ボリュームをフォーマットするようにして下さい。
(OPEN-L/M/V をフォーマットする際のデフォルト値は、Solaris で使用可能なシンク数となっています。)
- (35) 64bit FAL を使用してデータ転送を行う場合、Fibre は 64bit 対応のものを使用して下さい。
AIX の場合、Fibre は IBM 純正のものを使用して下さい。
- (36) AIX 版の 64bit FAL は、AIX V4.3.3 と AIX V5.x で互換性がありません。インストールして使用する場合は、OS バージョンにあったものをインストールするようにして下さい。
インストールに関しては、3.2.3 章を参照して下さい。
- (37) FCU は 32bit FAL を使用して動作します。
- (38) M/F から新たにデータセットを割当てた時は、MTO 又は OTM を実行する前に VTOC の DSCB1 にあるデータセットインジケータの最終ボリュームフラグを確定させるため、必ず M/F から当該ボリュームにダミーデータを書き込んでから実行して下さい。
- (39) I/H シェア環境で FAL/FCU を使用する場合は、FAL/FCU のバージョンを 01-XX-57 以上で使用して下さい。(XX ≥ 02)
- (40) AIX で FAL/FCU を使用する場合、中間ボリュームに対して PVID を設定しないで下さい。
PVID を設定した場合、動作保証はできません。
- (41) 中間ボリュームに対して check sum コマンドは実行しないで下さい。
- (42) FCU コマンドでデータを書き込む際に、転送先のサイズよりも大きいデータを書き込んだ場合
mto 時：書き込んでいって OS のシステム容量をオーバーした場合に FCU のシステムエラーとなる。
otm 時：書き込んでいって最終トラック書き込み後トラックが無い場合に FAL エラー(-28)となる。
- (43) AIX 共用オープン機能を使用した場合は、AIX の仕様によりデータ量が 2Gbyte を超える転送を行うことは出来ません。この制限は 32bit 版 OS が対象です。
- (44) AIX V5.2 ML01(Maintenance Level 01)以上をご使用になられる場合は、日立ディスクレイブシステム向け HTC_ODM パッケージの適用が必須となります。
- (45) AIX サーバで HDLM 連携を行う場合は、AIX サーバに現在適用されている日立ディスクレイブシステム向け HTC_ODM パッケージが、中間ボリュームに使用するデバイスエミュレーションタイプに対応していることを、ご確認願います。
- (46) UNIX の場合、autopkeyset コマンドでは libfalmt.* に対するライセンス設定は行いません。
ppkeyset コマンドを使用して下さい。
- (47) Windows の場合、autopkeyset.exe コマンドでは falmt.dll に対するライセンス設定は行いません。
ppkeyset.exe コマンドを使用して下さい。
- (48) Windows の場合、fcu コマンド実行時に -P オプションを指定した場合はこれを無視します。また、fcu コマンドはオプション(“-”付の指定値)以外の最初の引数をパラメタ定義ファイルとみなして動作するため、引数と同じ名前のファイル(絶対パスの場合はその指定先ファイル)が存在する場合は FCU を終了せずにこれを読み出して処理を行うこととなります。
例：# fcu -nw -nc -P mto VSNA AAA:DSNAME outfile No No No
この場合は「mto」をパラメタ定義ファイルとみなします。

- (49)CPUタイプがAMD64/Intel64でOSが64ビット版Linuxの場合、32ビット用のFAL/FCUをインストールする必要がありますが、以下の点に注意してください。
- 64ビット版のLinuxで32ビット用 FAL/FCUを動作させるためには32ビットアプリケーションを動作させるためのライブラリ (ld-linux.so.2)が必要となります。
- このライブラリを含むパッケージ(OSのバージョン毎に異なる場合があります)をOS提供元に問い合わせ、インストールの有無を確認の上適用してください。
- (OS提供元にパッケージを確認する際、このライブラリと依存関係にあるライブラリが存在しないか確認し、存在する場合は併せてインストールして下さい)
- (50)WindowsにてCodeConverterを使用する際には、WindowsのOSビットにかかわらず32bit版のFAL/FCUを使用してください。
- (51)VMWareの仮想マシン上でFAL/FCUを使用する場合には、中間VOLをRAWデバイスマッピング(RDM)ディスクとして使用してください。RDMディスクの使用方法はVMWareのマニュアルを参照してください。

10 トラブル発生時の情報収集

トラブルが発生し、原因が究明できない場合は以下の情報を収集して連絡して下さい。

(1) UNIX系OSの場合

- ① エラーコード (FCUエラーコード、FALエラーコード、SYSエラーコード)
⇒ 「-nw」、「-nc」指定でFCUを使用している場合、コマンドラインに表示されています。
FCUの画面を使用している場合、HelpメニューのErrorでも取り出せます。
- ② FCUのパラメタ
⇒ 「-nw」指定でFCUを使用している場合、コマンドラインに表示されています。
FCUの画面を使用している場合、操作画面にも表示されています。
- ③ 使用したボリューム定義ファイルの内容 (datasetmount.dat)
- ④ パラメタ定義ファイルを使用している場合には、パラメタ定義ファイルの内容 (デフォルト名 : fcudata.param)
- ⑤ マルチボリューム定義ファイルを使用している場合には、マルチボリューム定義ファイルの内容 (デフォルト名 : multidef.dat)
- ⑥ コマンドラインの履歴が取れる場合は、その履歴。
- ⑦ FALでエラーを検出している場合には、以下のエラーログファイル、ダンプファイルを作成していますので、その内容を採取して下さい。
ただし、環境変数で「ERR_LOG_FILE」、「ERR_DUMP_FILE」が指定されている場合には、指定されたファイルにエラーログ、ダンプを出力しています。
/tmp/fal_error 及び /tmp/fal_error.bak
/tmp/fal_dump 及び /tmp/fal_dump.bak
/tmp/uocmtoedit_log 及び /tmp/uocmtoedit_log.bak (HMDE/KANJI^(注)連携時のみ)
/tmp/uocmtoout_log 及び /tmp/uocmtoout_log.bak (HMDE/KANJI連携時のみ)
/tmp/uocotmedit_log 及び /tmp/uocotmedit_log.bak (HMDE/KANJI連携時のみ)
- ⑧ syslogの内容 (特にエラー情報)
- ⑨ プラットフォーム名
- ⑩ メインフレームのOS
- ⑪ 「uname -a」コマンド実行時の表示
- ⑫ 中間ボリュームの情報
・ listvolの表示結果
- ⑬ データの特徴
・ テキストファイル または バイナリデータ
- ⑭ HMDE/KANJI連携時はHMDE/KANJIのトラブル発生時の情報
採取する情報はHMDE/KANJIのユーザズガイドを参照してください。
- ⑮ 中間ボリュームに対して HDLM(JP1/HiCommand Dynamic Link Manager)を使用している場合は
HDLM のユーザズガイドを参照して必要な情報を採取して下さい。
- ⑯ /usr/lib/libuoc.ver

(注) 「Code Conveter」を使用しているお客様は、「HMDE/KANJI」を「Code Converter」と読み替えて下さい。

(2) Windows の場合

① FCU ログファイル

FCU ログは、パラメタ定義ファイル名に「.log」を付加したファイルに記録されます。

(例) パラメタ定義ファイル : **Test.prm**

FCU ログファイル : **Test.prm.log**

② パラメタ定義ファイルを使用している場合には、パラメタ定義ファイルの内容

(デフォルト名 : fcudata.prm)

③ 使用したボリューム定義ファイルの内容 (datasetmount.dat)

④ マルチボリューム定義ファイルを使用している場合には、マルチボリューム定義ファイルの内容

(デフォルト名 : multidef.dat)

⑤ FAL ログファイル

C:¥fal_error 及び、**C:¥fal_error.bak**

C:¥fal_dump 及び、**C:¥fal_dump.bak**

C:¥tmp¥uocmtoedit_log 及び、**C:¥tmp¥uocmtoedit_log.bak** (HMDE/KANJI^(注)連携時のみ)

C:¥tmp¥uocmtoout_log 及び、**C:¥tmp¥uocmtoout_log.bak** (HMDE/KANJI連携時のみ)

C:¥tmp¥uocotmedit_log 及び、**C:¥tmp¥uocotmedit_log.bak** (HMDE/KANJI連携時のみ)

C:¥tmp¥uocpipeout_log 及び、**C:¥tmp¥uocpipeout_log.bak** (HMDE/KANJI連携時のみ)

C:¥tmp¥uocpipein_log 及び、**C:¥tmp¥uocpipein_log.bak** (HMDE/KANJI連携時のみ)

(**C:¥tmp**ディレクトリがない場合は、上記HMDE/KANJI連携時のログは出力しません)

⑥ ワatson博士ログファイル(エラー時にWatson博士のメッセージが表示された場合)

DRWTSN32.LOG

Watson博士ログファイルは、デフォルトでは、Windowsをインストールした際に指定したディレクトリに作成されます。

(例) **C:¥WINNT**

Watson博士については、Windows のオンラインヘルプを参照して下さい。

⑦ Windows イベントログ(システムセキュリティアプリケーションを各々ファイルに出力して下さい。)

⑧ Windowsのバージョンまたはサービスパック

⑨ メインフレームのOS

⑩ 中間ボリュームの情報

・ MF-file information画面のハードコピー

⑪ データの特徴

・ テキストファイル または バイナリデータ

⑫ HMDE/KANJI連携時はHMDE/KANJIのトラブル発生時の情報

採取する情報はHMDE/KANJIのユーザズガイドを参照してください。

⑬ 中間ボリュームに対して HDLM(JP1/HiCommand Dynamic Link Manager)を使用している場合は

HDLM のユーザズガイドを参照して必要な情報を採取して下さい。

⑭ C:¥WINDOWS¥libuoc.ver

(注)「Code Conveter」を使用しているお客様は、「HMDE/KANJI」を「Code Converter」と読み替えて下さい。

付録

ログフォーマット例

次は FAL/FCU で出力するログの例です。
尚、このフォーマットは、バージョンにより変更されることがあります。

UNIX 版の場合 (fal_error ファイル)

/tmp/fal_error にログファイルを作成します。
ただし、環境変数で「ERR_LOG_FILE」が指定されている場合には、指定されたログファイルを作成します。

正常時のログフォーマット

08/19 15:32:38 15:32:39 VOL001:TEST1 test1.out 5 5 5 5 mt0 N L
実行時刻 転送元ファイル名 転送先ファイル名 ① ② ③ ④ ⑤

- ① FCU 入力／出力レコード数
- ② HMDE/KANJI 入力／出力レコード数
- ③ ファイル転送方向
- ④ HMDE/KANJI 動作モード → O:OLD, N:NEW, 空白:非 KANJI 転送時(*1)
- ⑤ FAL 動作モード → L:ラージファイル対応, 空白:非 KANJI 転送時

*1:

O(OLD):FAL45 版以降対応 KANJI バージョン(01-03 以降の場合)

N(NEW):FAL45 版以降対応 KANJI バージョン(01-04 以降の場合)

U(UNICODE):FAL49 版対応 Windows 版のみ KANJI バージョン(01-05 以降で UUNICODE 転送時(*2))

*2:

KANJI バージョンは CODECONVERT 時の SI バージョンとは異なります。

01-03-xx の比較バージョンは 03xx

01-04-xx の比較バージョンは 04xx

と対象になる為、'N' となります。

異常時のログフォーマット

Fri Aug 16 16:50:37 2002 : root : FCU Error(-319,-8,0) PRM=otm,No,No,No,Emp=No,... VSE=No,PIPE=No L
↓ ↓ ↓ ↓ ↓ ↓ ↓
実行時刻 実行 ID エラーコード FCU エラー FAL エラー システムエラー 実行時 FCU パラメータ

Windows 版の場合 (パラメタ定義ファイル.log ファイル)

パラメタ定義ファイル名に「.log」を付加したログファイルを作成します。

(例) パラメタ定義ファイル : **Test.prm**
ログファイル : **Test.prm.log**

正常時のログフォーマット

実行時刻

Thu Aug 01 2002 ① ② ③ ④ ⑤
12:52:01 12:52:01 100 100 0 0 0 N 0
(1 / 2)
mnto VOL002:FILE1 C:¥FCU¥file1.txt No No No Emp=No ... VSE=No PIPE=No → 実行時 FCU パラメータ
(COMPLETE !)
FCU err : (0)
FAL err : (0)
Sys err : (0)

} 終了コード

- ① FCU 入力/出力レコード数
- ② HMDE/KANJI 入力/出力レコード数
- ③ PIPE 出力レコード数 (*3)
- ④ HMDE/KANJI 動作モード → O: OLD, N: NEW, U: UNICODE, 空白: 非 KANJI 転送時(*1)
- ⑤ PIPE 接続リトライ回数 (*3)

*1:

O(OLD): FAL45 版以降対応 KANJI バージョン(01-03 以降の場合)

N(NEW): FAL45 版以降対応 KANJI バージョン(01-04 以降の場合)

U(UNICODE): FAL49 版対応 Windows 版のみ KANJI バージョン(01-05 以降で UUNICODE 転送時(*2))

*2

KANJI バージョンは CODECONVERT 時の SI バージョンとは異なります。

01-03-xx の比較バージョンは 03xx

01-04-xx の比較バージョンは 04xx

と対象になる為、'N' となります。UNICODE 転送時は'U'となります。

*3

PIPE は SQL Loader 連携時のみ使用。ただし、FAL SQL Loader は USP V と VSP, VSP G1000, VSP 5000 では未サポートです。

異常時のログフォーマット

実行時刻

Thu Aug 01 2002
12:52:02 12:52:02 0 0 0 0 0 N 0
(2 / 2)
mnto VOL002:FILE2 C:¥FCU¥file2.txt No No No Emp=No RDW=No VSE=No PIPE=No → 実行時 FCU パラメータ
(ERROR !)
FCU err : (-132) Dataset : Information get error
FAL err : (-8) Couldn't find the appointed dataset.
Sys err : (0)

} エラーコード

| | FAL 環境変数 | デフォルト(環境変数未指定時) | 記載章 |
|---|---------------------|--|----------------------|
| 1 | ERR_LOG_FILE | /tmp/fal_error (UNIX) c:%fal_error (WIN) | 7.8(6),10(1)⑦ |
| 2 | ERR_DUMP_FILE | /tmp/fal_dump (UNIX) c:%fal_dump (WIN) | 7.8(6),10(1)⑦ |
| 3 | FAL_MULTI_DEF_FILE | multidef.dat | 8.3(1) |
| 4 | FAL_MULTI_CHECK | ON | 4.7 表 4.7-1(-41),4.8 |
| 5 | FAL_NO_RESERVE | OFF | 4.9(2) |
| 6 | FAL_RETRY | OFF | 4.10(4)(4-1) |
| 7 | FAL_RETRY_COUNT | 10(回) | 4.10(4)(4-2) |
| 8 | FAL_RETRY_WAIT_TIME | 1(秒) | 4.10(4)(4-3) |
| 9 | FAL_RETRY_TARGET | AIX(16)AIX,HP 以外(5) | 4.10(4)(4-4) |

| | FCU (UNIX)環境変数 | デフォルト(環境変数未指定時) | 記載章 |
|----|--------------------------|-----------------------|--------------------------|
| 1 | WAIT_TIME_VALUE | 10(分) PIPE 待ち時間 | 7.2,8.8.1 表 8.8-1 (-253) |
| 2 | FCU_E363_TRACE_MODE | 環境変数未定義 | 7.8(4)(4-1)(4-5) |
| 3 | FCU_E363_TRACE_COPY_FILE | 環境変数未定義 | 7.8(4)(4-2) |
| 4 | FCU_E363_TRACE_LOG | 環境変数未定義 | 7.8(4)(4-3) |
| 5 | ERR_LOG_FILE | /tmp/fal_error (UNIX) | 7.8(6),10(1)⑦ |
| 6 | FCU_RETRY | OFF | 7.10(4)(4-1) |
| 7 | FCU_RETRY_COUNT | 10(回) | 7.10(4)(4-2) |
| 8 | FCU_RETRY_WAIT_TIME | 1(秒) | 7.10(4)(4-3) |
| 9 | FCU_RETRY_TARGET | AIX(16)AIX,HP 以外(5) | 7.10(4)(4-4) |
| 10 | FCU_OTM_INTERVAL | インターバルをとらない | 7.11(4)(4-1) |

| | FCU (WIN)環境変数 | デフォルト(環境変数未指定時) | 記載章 |
|---|-----------------|----------------------------|-----|
| 1 | WAIT_TIME_VALUE | 600000 (ミリ秒)10分 SQL 連携時に有効 | 7.2 |

- ・ C シェルの場合：ホームディレクトリの「.cshrc」に以下の 1 行を追加する。(ファイルが無い場合は作成する)
setenv 環境変数名 値
- ・ C シェル以外の場合：ホームディレクトリの「.dtprofile」に以下の 2 行を追加する。(ファイルが無い場合は作成する)
環境変数名=値
export 環境変数名

注：共通デスクトップ環境がないマシンでは、「.profile」に上記行を追加してください。ファイルがない場合は作成してください。

- ・ Windows の場合：コントロールパネル→システム (システムのプロパティ) →詳細→環境変数を選択しシステムの環境変数 新規に変数名 環境変数名に値を入力 後 OK ボタン押下

用語集

| 見出し | 用語 | 意味 |
|-----|------------------------|---|
| U | UOC 編集モード
USER-EDIT | パラメタに USER-EDIT を指定した状態
HMDE/KANJI 連携を実施する場合、パラメタに指定する文字 |
| V | VSN | ボリュームシリアル番号の略 |
| テ | データセット | メインフレーム側でデータを処理する単位(ファイル相当) |
| ホ | ボリューム | データセットやファイルを読み書きするディスク |
| メ | メインフレームファイル | データセットと同義 |