

***I F A S F I L E、***  
***I F A S P R O R D B***  
***利用の手引***

# はしがき

本書は、Windows上で動作する統合ファイルアクセスシステムプログラム群であるIFAS FILE Ver2.0、および、IFASPRO RDB Ver2.0について説明したものです。

なお、本書の内容はCOBOL Standard Editionに同梱されるIFAS FILEと単独製品のIFASPRO RDBの共通のものとなっています。IFASPRO RDBをご利用でない場合は、IFASPRO RDBに関する記述を読み飛ばして、ご利用ください。

カスタムコントロール(ActiveX)を使用してIFASファイルやIFASRDB表をアクセスする場合は、**別途製品の「RDB/FILEアクセスコントロール」**をご利用ください。本書の「カスタムコントロール」は「RDB/FILEアクセスコントロール」と読み替えてください。

本書の構成は以下のとおりです。

章	タイトル	内 容
0	製品概要	IFAS FILE、IFASPRO RDBの特徴と構成等の製品概要
1	ファイル構造とデータ形式	IFAS FILE、IFASPRO RDBが扱う各ファイル/RDB表の構造と、取り扱うことの可能なデータ形式についての説明
2	導入と環境設定	IFAS FILEシステムの導入と環境設定方法の説明
3	Cプログラムからの利用	IFAS一般ファイルをCプログラムから利用する場合の各機能の説明と、ファイルアクセス関数の使い方の説明
4	COBOLプログラムからの利用	IFAS一般ファイル、IFAS RDB表をNEC COBOLから利用する場合の手順とIFAS独自の各機能の説明
5	IFASカスタムコントロール(参考) ※サポート中止のため参考	IFAS一般ファイル、IFAS RDB表をVisualBasicから利用するための手順と、アクセスインターフェースの説明
6	運用と保守	IFAS FILE、IFASPRO RDBのクライアント・サーバ機能、ファイル復旧機能を利用するための各種方法の説明
7	ユーティリティ	IFAS FILE、IFASPRO RDBが提供する各種ユーティリティプログラムの使い方の説明
8	関連製品	IFAS FILE、IFASPRO RDBと関わりを持つ関連プログラム製品を利用する上での注意事項等の説明
付録	ステータスコード一覧 IFASコマンド一覧 RFDファイルについて バージョン間機能差異	IFAS FILE、IFASPRO RDBのエラーステータスコード一覧 IFAS FILE、IFASPRO RDBのコマンド機能一覧 データ定義ファイルであるRFDファイルについての説明 バージョン間の対応機能の差異の説明

2019年 10月 第4版

備考

(1)Microsoft、Windows、Windows Server および Visual Basic は米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。

(2)その他、記載されている会社名、製品名は一般に各社の登録商標または商標です。

# 目 次

---

はしがき	II
------	----

---

製品概要	1
------	---

---

1. ファイル構造とデータ形式	5
-----------------	---

---

1-1 IFAS一般ファイル	5
----------------	---

1-1-1 索引順編成ファイル	5
-----------------	---

1-1-2 相対編成ファイル	10
----------------	----

1-1-3 IFAS順編成ファイル	10
-------------------	----

1-2 IFASRDB表	11
--------------	----

1-2-1 基本表	11
-----------	----

1-2-2 仮想表	12
-----------	----

1-3 データ形式	14
-----------	----

2. 導入と環境設定	17
------------	----

---

2-1 イニシエータ	17
------------	----

2-2 IFASのSG設定	20
---------------	----

2-3 環境設定ファイル	22
--------------	----

2-3-1 記述方法	22
------------	----

2-3-2 環境設定ファイルの動的切り替え	26
-----------------------	----

2-3-3 注意事項	28
------------	----

2-4 RDBの環境設定	29
--------------	----

2-4-1 導入手順	29
------------	----

2-4-2 環境設定	29
------------	----

2-5 アクセス方式	30
------------	----

3. Cプログラムからの利用	31
----------------	----

---

3-1 機能概要	31
----------	----

3-1-1 ファイルアクセス機能	31
------------------	----

3-1-2 排他／同時アクセス制御	35
3-1-3 トランザクション処理	36
3-1-4 ファイルリカバリ処理	37
3-2 IFAS FILE関数	39
3-2-1 索引順編成ファイルアクセス関数	39
3-2-2 相対編成ファイルアクセス関数	71
3-2-3 順編成ファイルアクセス関数	95
3-2-4 共通アクセス関数	111
3-2-5 トランザクション関数	116
3-3 注意事項	122
<b>4. COBOLプログラムからの利用</b>	<b>123</b>
4-1 機能概要	123
4-1-1 諸元	123
4-1-2 データアクセス機能	123
4-1-3 OPEN命令のモードと共用制御	125
4-2 ファイル情報と動作環境の設定	127
4-2-1 ファイル情報の設定	127
4-2-2 動作環境の設定	130
4-2-3 更新ログファイル設定	133
4-3 一般ファイルの利用	134
4-3-1 ファイル編成と入出力命令	134
4-3-2 ステータス	135
4-3-3 MIXファイル互換機能	140
4-4 RDB表の利用	143
4-4-1 ファイル編成と入出力命令	143
4-4-2 ステータス	144
4-4-3 各種RDB表使用時の注意事項	150
<b>5. IFASカスタムコントロール(参考)</b>	<b>152</b>
5-1 機能概要	152
5-1-1 諸元	152

5-1-2 データアクセス機能 .....	153
5-1-3 オープンモードと共用制御 .....	155
5-2 動作環境の設定 .....	158
5-2-1 動作環境の設定 .....	158
5-2-2 更新ログファイル設定 .....	162
5-3 利用方法 .....	163
5-3-1 各種資源 .....	163
5-3-2 利用方法 .....	163
5-3-3 ファイル編成と入出力命令 .....	171
5-3-4 IFAS ファイル使用時の注意事項 .....	171
5-4 プロパティ解説 .....	173
5-5 メソッド解説 .....	191
<b>6. 運用と保守 .....</b>	<b>240</b>
6-1 クライアント・サーバ機能 .....	240
6-1-1 サーバ名の指定方法 .....	240
6-1-2 Cプログラムからの利用 .....	243
6-1-3 COBOLプログラムでの指定 .....	244
6-1-4 IFASカスタムコントロールでの指定(参考) .....	245
6-1-5 注意事項 .....	246
6-2 IFAS ファイルのバックアップ .....	247
6-3 更新ログファイルの運用 .....	249
6-3-1 更新ログの収集 .....	249
6-3-2 シングルログによる運用 .....	250
6-3-3 マルチログによる運用 .....	250
6-3-4 更新ログファイルのリセット .....	251
6-3-5 更新ログファイルの設計 .....	252
6-4 障害時のファイル復旧 .....	252
6-4-1 ディスク障害からの復旧 .....	252
6-4-2 システム障害からの復旧 .....	253
6-4-3 RDB管理ファイルの復旧 .....	255

6-5 RDB環境の移行 .....	256
6-6 未クローズファイル .....	257
<b>7. ユーティリティ .....</b>	<b>259</b>
7-1 一般ファイルの管理 .....	259
7-1-1 ファイル管理ユーティリティ .....	259
7-1-2 ファイルコピーユーティリティ .....	268
7-1-3 インデックスメンテナンスユーティリティ .....	280
7-1-4 RFD編集ツール .....	291
7-1-5 移行ユーティリティ .....	294
7-2 RDB表の管理 .....	307
7-2-1 表定義ユーティリティ .....	307
7-3 IFAS ファイルの復旧 .....	341
7-3-1 ロールフォワードユーティリティ .....	341
7-3-2 ディファードリカバリユーティリティ .....	346
7-3-3 ファイルロックリセットユーティリティ .....	349
7-4 運用管理 .....	352
7-4-1 運用支援ツール .....	352
7-4-2 バックアップ支援ユーティリティ .....	362
7-4-3 サーバ管理ユーティリティ .....	366
7-4-4 IFAS管理情報メンテナンスツール .....	367
7-5 定義ファイルのパラメータ置換機能 .....	371
7-6 その他 .....	372
7-6-1 IFASプロパティ表示 .....	372
<b>8. 関連製品 .....</b>	<b>375</b>
8-1 ファイルメンテナンスキット .....	375
8-2 SMART/CS .....	375
8-3 RDB/FILEアクセスコントロール .....	375
<b>付録A ステータスコード一覧 .....</b>	<b>376</b>

A-1 ステータスの見方 .....	376
A-2 イニシエータのステータス .....	377
A-3 C言語関数のステータス .....	378
A-4 COBOL利用時のステータス .....	388
A-4-1 入出力状態 .....	388
A-4-2 詳細ステータス .....	391
A-5 IFASカスタムコントロールのステータス(参考) .....	424
<b>付録B IFASコマンド一覧 .....</b>	<b>425</b>
<b>付録C RFDファイルについて .....</b>	<b>435</b>
<b>付録D バージョン間機能差異 .....</b>	<b>436</b>
<b>索引 .....</b>	<b>437</b>





# 製品概要

## はじめに

本説明書は、IFAS FILE、IFASPRO RDB両製品について説明しています。

IFAS FILEは独自のファイル構造を持つ、索引順編成ファイル、相対編成ファイル、順編成ファイルへのアクセス関数を提供し、C、NEC COBOL(以下COBOL)、Visual-Basic(別途製品の「RDB/FILEアクセスコントロール」が必要)等で作成するアプリケーションプログラムから利用することができます。また、ファイル保守用、システム運用のための各種ユーティリティプログラムを標準装備しています。

IFAS FILEがアクセスする各種編成ファイルを本書ではIFAS一般ファイルと総称します。

IFASPRO RDBはIFAS FILEのすべての機能に加えて、IFAS ファイルを基盤とした独自のRDB表をCOBOL、Visual-Basic(別途製品の「RDB/FILEアクセスコントロール」が必要)等からアクセスする機能を提供する製品です。RDB表の定義、データのインポート/エクスポートを行うことが可能なユーティリティも標準装備しています。IFASPRO RDBがアクセスする各種RDB表を本書ではIFAS RDB表と総称します。

また、本書ではIFAS FILE、IFASPRO RDBを総称して“IFAS”または“IFAS製品”と記載しております。

## 製品の特徴

IFAS FILE、IFASPRO RDBには以下のような特徴があります。

- IFAS一般ファイルにアクセスするアプリケーションプログラム(以下AP)をC、COBOL、Visual-Basic(別途製品の「RDB/FILEアクセスコントロール」が必要)等で作成できます。本書では、C言語用アクセス関数について記載しています。COBOLから利用する場合は、COBOLの説明書、および本書の「4. COBOLプログラムからの利用」を参照してください。Visual-Basicで利用する場合は、RDB/FILEアクセスコントロールの説明書を参照してください。
- IFASPRO RDBではIFAS RDB表にアクセスするAPをCOBOL、Visual-Basic(別途製品の「RDB/FILEアクセスコントロール」が必要)等で作成できます。
- IFAS一般ファイル、IFAS RDB表にアクセスするAPIは、C、COBOLのみ64ビットAPを作成できます。
- クライアントOS、サーバOSに製品を導入することで、クライアントからサーバ上のIFAS一般ファイル、IFAS RDB表にアクセスすることが可能です。
- ファイル排他機能、レコード排他機能、ファイルリカバリ機能が完備され、信頼性の高いアプリケーションシステムの構築が可能です。
- マルチスレッドアプリケーションからも利用可能で、スレッド間での排他/共有オープンも可能です。また、同一ファイルのマルチオープンも可能です。
- リモートデスクトップ(Terminal Service機能、以下TSと呼ぶ)に対応しています。TSクライアントからTS動作中のサーバマシン内のIFAS使用APを利用できます。

## プログラム構成

IFAS FILE、IFASPRO RDBは次の3つの形態のプログラムから構成されます。

### ダイナミックリンクライブラリ(DLL)

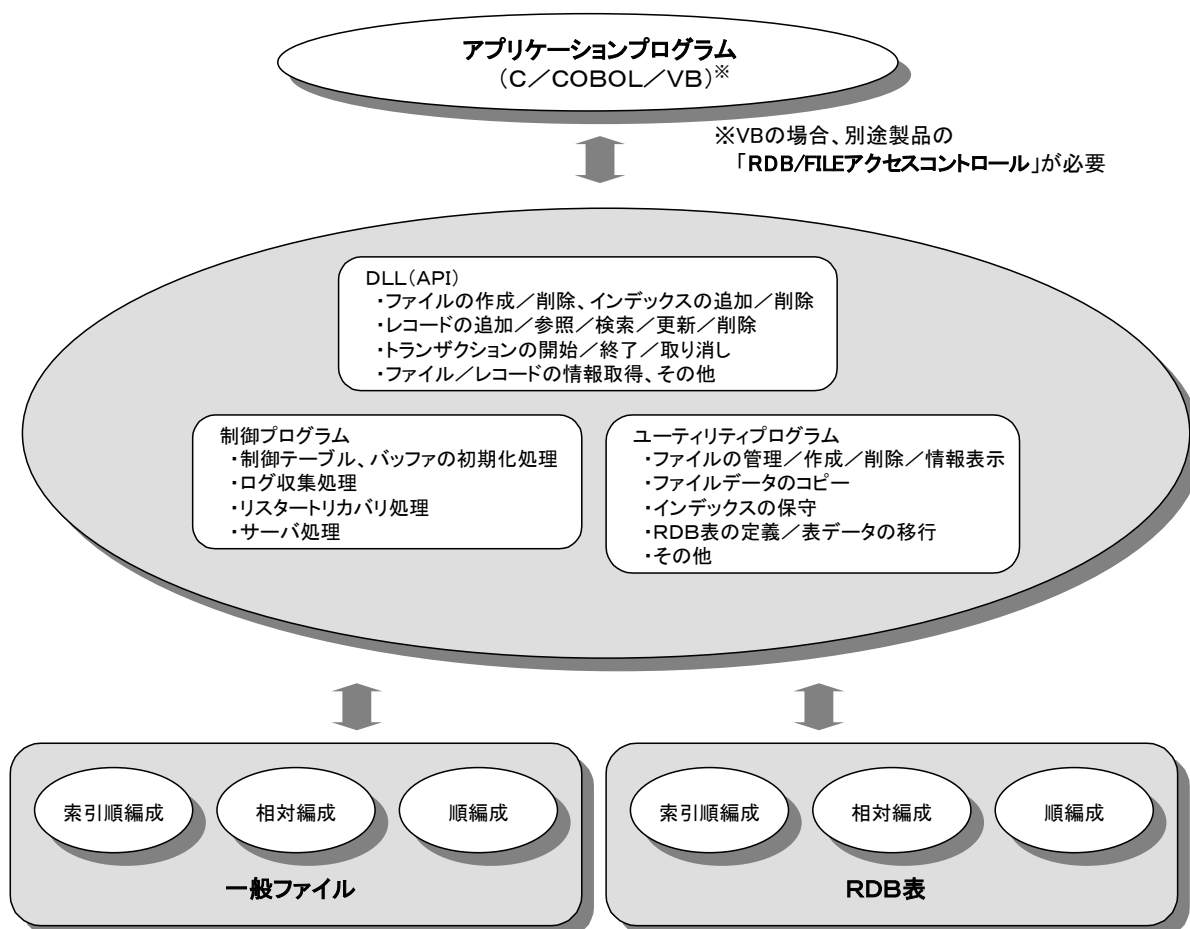
AP実行時に動的にリンクされるプログラムで、APからのファイルアクセス要求に基づきファイル処理を実行します。IFAS FILEにはC言語用(一般ファイルアクセスのみ)、COBOL言語用の2種類のライブラリがあります。

### 自動起動される制御プログラム

システム立ち上げ時やAPからのファイル処理要求の延長で自動的に起動されるプログラムで、IFAS FILEがファイル処理等のために必要とする共有メモリや管理ファイル等を初期化するIFAS FILEイニシエータ、システム立ち上げ時にファイル復旧を行うリスタートリカバリ、クライアントマシンからのファイルアクセス要求を処理するサーバ処理(以下IFASサーバ)、ファイルの更新ログをログファイルへ収集するログライタ等の実行形式プログラム(プロセスあるいはスレッド)があります。

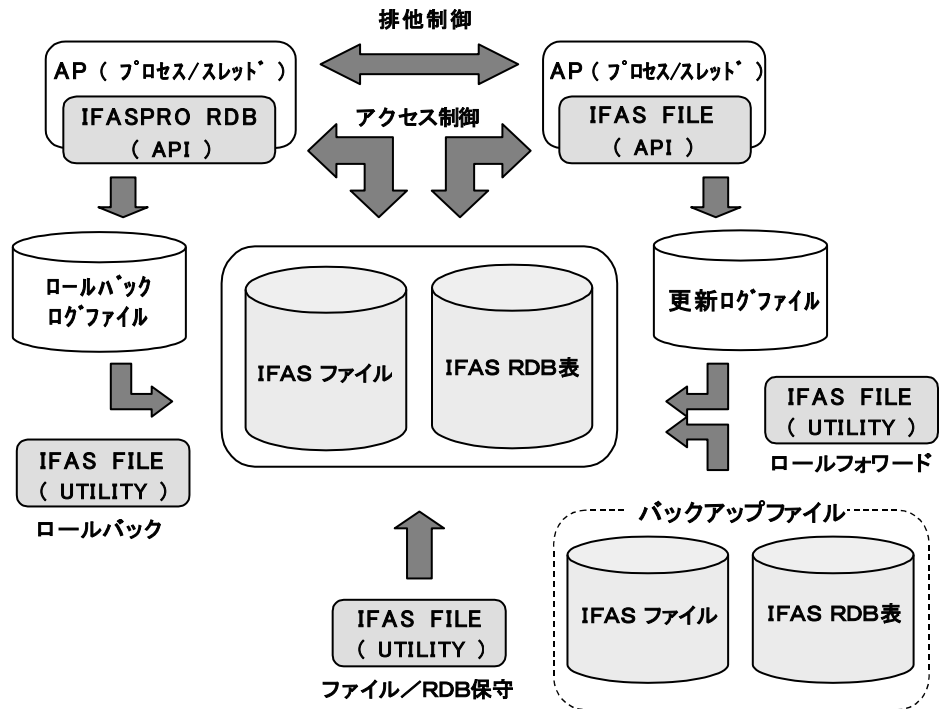
### ユーティリティプログラム

IFAS FILEの動作環境の変更機能やファイル/RDB表の移行・保守・復旧等の機能をユーティリティプログラムとして提供します。

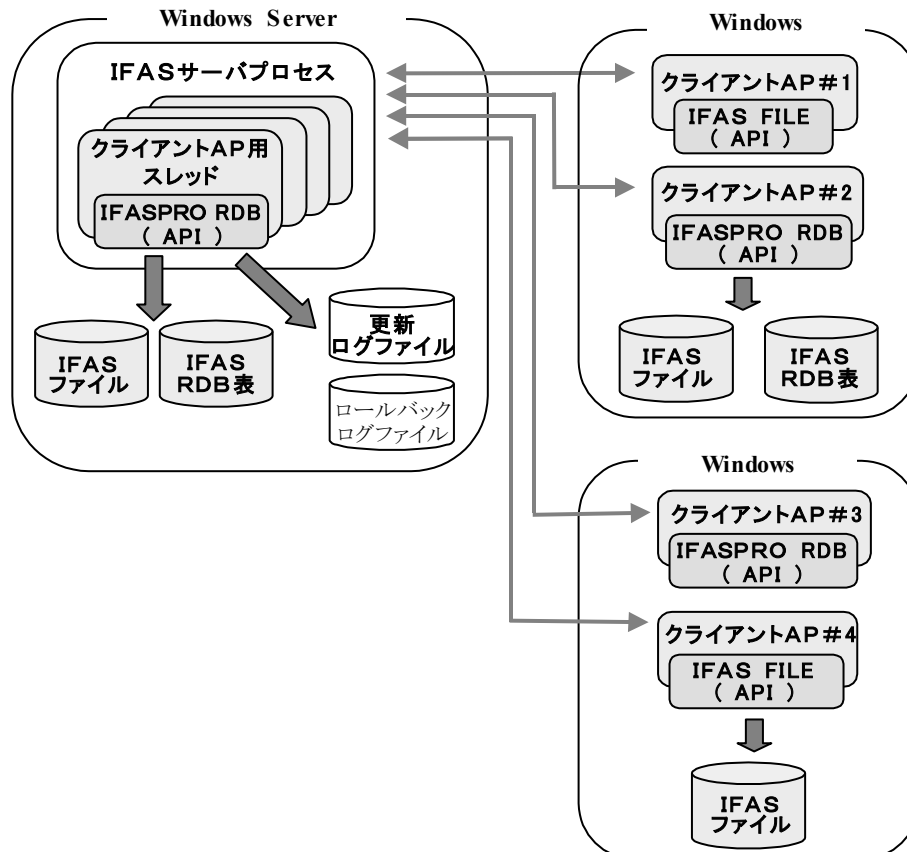


## システム形態

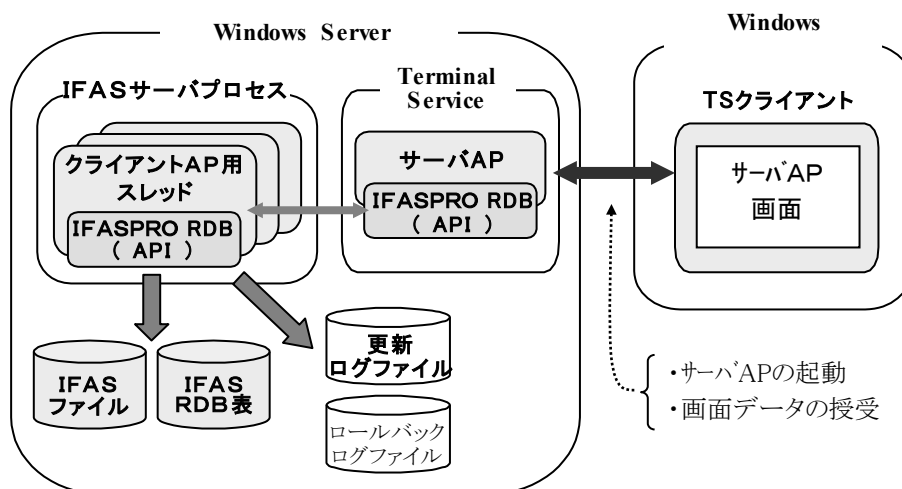
【スタンドアロン形態で利用の場合】



【サーバ・クライアント形態で利用の場合】



【Terminal Service形態で利用の場合】



# 1. ファイル構造とデータ形式

IFAS FILE、IFASPRO RDBがアクセスする各種ファイルおよびRDB表の構造と、扱うことが可能なデータ形式について説明します。

なお、IFAS FILE、IFASPRO RDBがアクセス可能なファイルは以下のとおりです。

- ・索引順編成ファイル
- ・相対編成ファイル
- ・順編成ファイル

上記ファイルを総称して「IFAS一般ファイル」と呼びます。

また、IFASPRO RDBがアクセス可能なRDB表は以下のとおりです。

- ・基本表
- ・仮想表(選択型、結合型、合併型)

上記RDB表を総称して「IFASRDB表」と呼びます。

本書では、「IFAS一般ファイル」、「IFAS RDB表」を総称して「IFAS ファイル」と呼びます。

## 1-1 IFAS一般ファイル

IFASがサポートする一般索引順編成ファイル、一般相対編成ファイル、一般順編成ファイル、それぞれのファイル構造と特徴について説明します。

### 1-1-1 索引順編成ファイル

索引順編成ファイルは、一対のインデックスファイルとデータファイルとで構成され、各データレコードの一意のフィールドをキーとするツリー構造を形成したインデックスによって効率良くレコードの検索処理を行うファイル編成です。

#### 諸元および特徴

IFAS索引順編成ファイルは以下のような特徴を持っています。

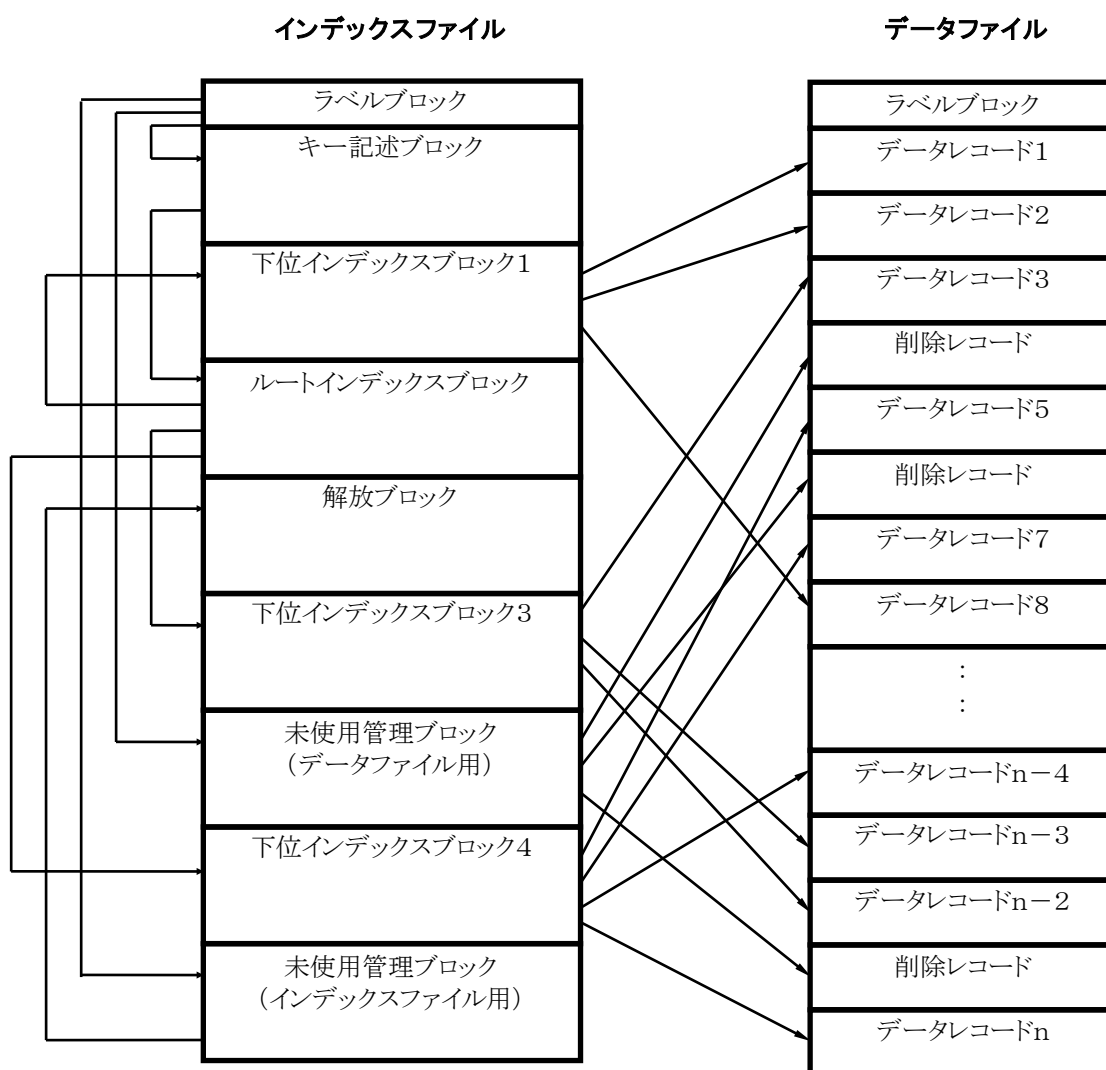
- ・最大約64Kバイトのレコードサイズまで指定可能
- ・最大256バイトのキーサイズまで指定可能
- ・最大255個までのインデックスを作成することが可能
- ・1個のインデックスにつき最大16個のキーパートが設定可能
- ・ファイル初期化時にインデックスブロックにあらかじめキー追加用の空きを設定可能
- ・検索機能を重視したアクセス関数の提供
- ・全自動レコードロック制御によりトランザクションの並列処理が可能
- ・更新ログによるファイルリカバリ機能を提供
- ・データファイルのみを相対編成ファイルや順編成ファイルとして利用可能

## ファイル構造

索引順編成ファイルは1組のインデックスファイルとデータファイルで構成され、この組み合わせはファイルを初期化した時の組み合わせ以外では使用することができません。両ファイルとも先頭にラベルブロックがあり、それぞれのファイル情報が格納されています。

データファイルのラベルブロック以降には固定長のデータレコードが格納され、すべてのデータレコードにはファイルの先頭からの相対位置に相当するレコード番号が割り当てられます。インデックスファイルには、データレコードのキー値とレコード番号を一組とするキーエントリが形成され、すべてのデータレコードのキーエントリがTree構造で格納されます。

【図1-1】に索引順編成ファイルのファイル構造概要例を示します。インデックス数は1、インデックスの階層レベル数は2、という場合の例です。



(矢印は参照関係を示します)

図 1-1 索引順編成ファイル構造概要

## インデックスファイル

インデックスファイルは、ラベルブロック、キー記述ブロック、未使用管理ブロック、インデックスブロックから構成されます。ラベルブロックは128バイトの固定長で作成され、他のブロックは任意の固定サイズで作成されます。利用者がファイルの初期化時にそのサイズを指定しない場合には、1024バイト(最小値)で作成されます。

各ブロックはそれぞれ以下のような目的で使⽤します。

### ラベルブロック

ファイルの属性情報を管理し、キー記述ブロックや未使用管理ブロック等を参照するためのポインタ情報を保持しています。

### キー記述ブロック

インデックスの属性情報を管理するブロックで、インデックス作成指定でオープンした場合やインデックスの追加操作を行った場合に本ブロックを作成します。

インデックスが1個作成される毎に対応したキー記述エントリを作成し、ブロックに格納します。最初に作成されたインデックスはプライマリインデックスと呼び、2個目以降のインデックスはセカンダリインデックスと呼びます。インデックス操作によって、インデックスの追加や削除を行うことができますが、プライマリインデックスを削除することはできません。

インデックス数分のキー記述エントリが1ブロックに格納できない場合は、次のキー記述ブロックを作成しチェーンされます。

### インデックスブロック

キーエントリをキーの昇順に格納するブロックで、キー数(データレコード数)が増えて複数のブロックになるとツリー型階層構造を形成します。

### 未使用管理ブロック

データファイル用とインデックスファイル用とがあり、削除されたレコード番号や解放された各種ブロックのブロック番号を再利用するための管理を行います。レコードの削除が行われた時、および、それに伴ってインデックスブロックが解放された時に本ブロックを作成します。

## インデックスツリー構造

インデックスのツリーの概略構造図を【図1-2】に示します。

この例では、インデックスブロックに4バイトのキーが4個まで格納できるという前提で、インデックスレベルが3レベルのツリー構造になっています。

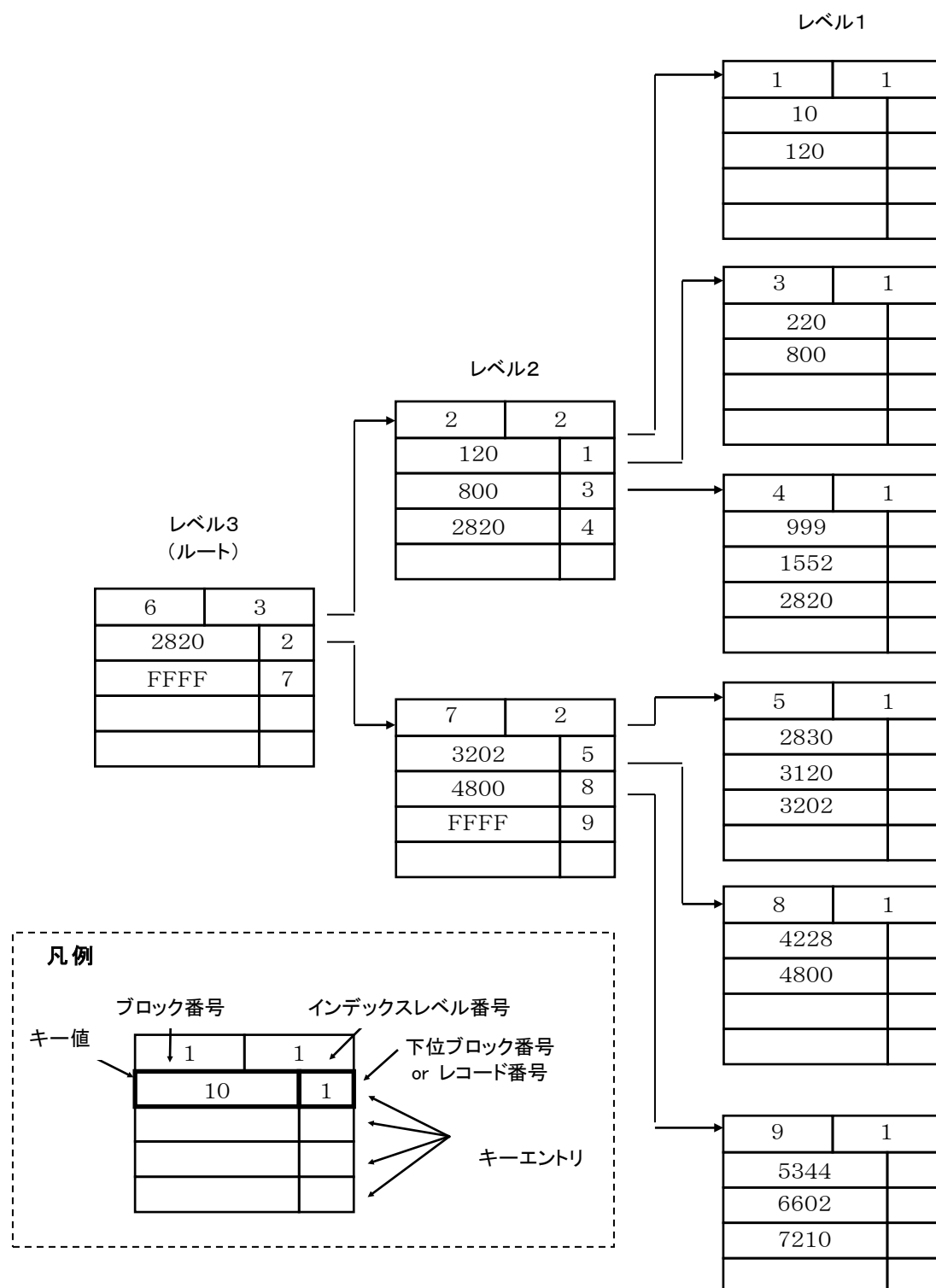


図 1-2 インデックスツリーの概略構造



【図1-2】において、最下位レベル(レベル1)のインデックスブロックに格納されるキーは、直接、データファイル内のレコード番号を指しています。レベル2以上のインデックスブロックに格納するキーは、下位インデックスブロックに格納可能な最大キーを示す値を持ち、それぞれ下位ブロック番号を指しています。

ツリー構造の最高レベル(例の場合レベル3)に1個だけ存在するブロックはルートインデックスブロックと呼び、検索処理で最初に参照するインデックスブロックであり、キー記述ブロックに格納されるキー記述エントリからポイントされています。

レベル2以上のインデックスブロックにある”FFFF”というキーは実在するレコードのキー値ではありませんが、管理上最も大きいキー値を格納するレベル1のブロックを示すために内部的に設定するキー値です。

【図1-2】のインデックスにおいて、仮に”1552”というキー値で検索を行うとした場合、まずルートインデックスブロック(ブロック番号6)を参照し、ブロック内のキーの中で検索キー値”1552”以上で一番近似するキーエントリを見つけだします。この場合は、”2820”になるので”2820”のキーエントリが指す下位レベルのブロック(ブロック番号2)を参照します。同様の方法でブロック内を検索すると、該当するキーはまた”2820”になり、最下位ブロック(ブロック番号4)を参照できます。

このあとは、この最下位ブロック内で検索キーと一致するキーエントリを見つけだせば、目的のレコード番号を知ることができます。インデックスツリーはレコード数が増加すると拡張されていくため、階層レベルが高くなっていきます。逆にレコード削除の操作が繰り返されると、解放されるインデックスブロックができるために階層レベルが下がることもあります。

## 重複キー

索引順編成ファイルではデータレコードイメージの一部をキーとしているため、別のレコードでもキー部分が同じイメージになる可能性があります。IFAS FILEはプライマリインデックスを除いて、重複するキーの管理を可能とします。レコードの書き出し処理中に、利用者が重複キーを許可すると設定したセカンダリインデックスで既にファイルに存在するレコードと同じキー値を検出しても、IFAS FILEは内部的に識別子を付加してキーを格納します。よって利用者は同一キーレコードの識別を意識することなく、重複キーのレコードを取り扱うことができます。

利用者が重複キー許可指定のあるセカンダリインデックスによって検索を行った時、そのキーと同じキー値を持つレコードが他に実在する場合はその旨のステータスを返却します。

## データファイル

データファイルは128バイトのラベルブロック以外の管理ブロックは保持しません。利用者が定義した固定長レコードの前に管理情報を付加した形態で、相対的にファイルに格納します。この相対的な格納順がレコード番号となります。(【図1-1】参照)レコードを削除した場合はそのレコードの管理情報に削除情報をセットして、インデックスファイル内の未使用管理ブロックに削除レコード番号を登録します。

つまり、削除されたレコードのレコード番号は未使用管理ブロックの情報によって、後で別のレコードを追加する時に再利用されます。

## 他編成ファイル互換機能

IFASでは、索引順編成ファイルのデータファイルを順編成ファイルや相対編成ファイルとしてオープンして処理することが可能です。使用方法の詳細については「1-1-2 相対編成ファイル」、「1-1-3 順編成ファイル」の説明を参照してください。

ただし、順／相対編成ファイルとして更新処理を行ったファイルを再びインデックスファイルと組み合わせると索引順編成ファイルとして使用する場合は、事前にデータとインデックスの整合処理(インデックスファイルの再作成)をインデックスメンテナンスユーティリティで行わなければなりません。

## 1-1-2 相対編成ファイル

相対編成ファイルは、ファイル内のレコードを相対レコード番号によって識別し、ダイレクトにアクセスすることを可能にしたファイル編成です。利用者の指示によってファイル内の最大レコード番号のレコード位置までファイル領域を確保することも可能で、最大レコード番号未満のレコード番号のレコードは論理的に存在しなくても(利用者が書き出していなくても)そのレコードのための領域はファイル内相対位置として確定します。よってファイル構造がシンプルで障害に対する復旧性が高く、何よりもディスク入出力負荷の少ない高速なアクセスを実現します。

### 諸元および特徴

IFAS相対編成ファイルは以下のような特徴を持っています。

- ・最大約64Kバイトのレコードサイズまで指定可能
- ・全自動レコードロック制御によりトランザクションの並列処理が可能
- ・更新ログによるファイルリカバリ機能の提供
- ・データ格納済みの相対編成ファイルにインデックスファイルを追加作成し、索引順編成ファイルにすることが可能
- ・順編成ファイルとしてオープンすることが可能

### ファイル構造

相対編成ファイルはレコード番号によるランダムアクセスを主目的としたファイル編成です。ファイル内の最大レコード番号までのすべてのレコード番号のレコードを格納するための領域はファイル内相対位置として確保され、それぞれのレコードの存在の有無はレコード管理情報によって表されます。

ファイルの先頭にラベルブロックがあり、ファイル情報が格納されています。ラベルブロック以降には固定長のデータレコードが格納され、すべてのデータレコードにファイルの先頭からの相対位置に相当するレコード番号が割り当てられます。

### 最大レコード番号

利用者はファイル作成時または出力モードオープン時にファイルの最大レコード番号を指定することができます。この最大レコード番号指定によって利用者はファイル作成時の初期サイズを設定したり、動的なファイル拡張を抑制したりすることが可能となります。

### ファイルサイズの動的拡張

相対編成ファイルはレコード番号(ファイル内相対レコード位置)を指定してレコードを書き込むことができるため、誤って不正な値(とてつもなく大きな数値)をレコード番号として指定すると、その値を信じてファイルサイズを拡張してしまいます。

そういった誤りを防ぐため、利用者が指定した最大レコード番号、または実際の物理ファイルサイズを超えたレコード番号出力をエラーとして検出するかどうかを、利用者が指定することができます。

## 1-1-3 IFAS順編成ファイル

順編成ファイルは、レコードが書き出した順番に格納され、その順番にアクセスすることを目的としたファイル編成です。

## 諸元および特徴

IFAS順編成ファイルは以下のような特徴を持っています。

- ・最大約64Kバイトのレコードサイズまで指定可能
- ・全自動レコードロック制御によりトランザクションの並列処理が可能
- ・更新ログによるファイルリカバリ機能の提供
- ・データ格納済みの順編成ファイルにインデックスファイルを追加作成し、索引順編成ファイルにすることが可能
- ・相対編成ファイルとしてオープンすることが可能

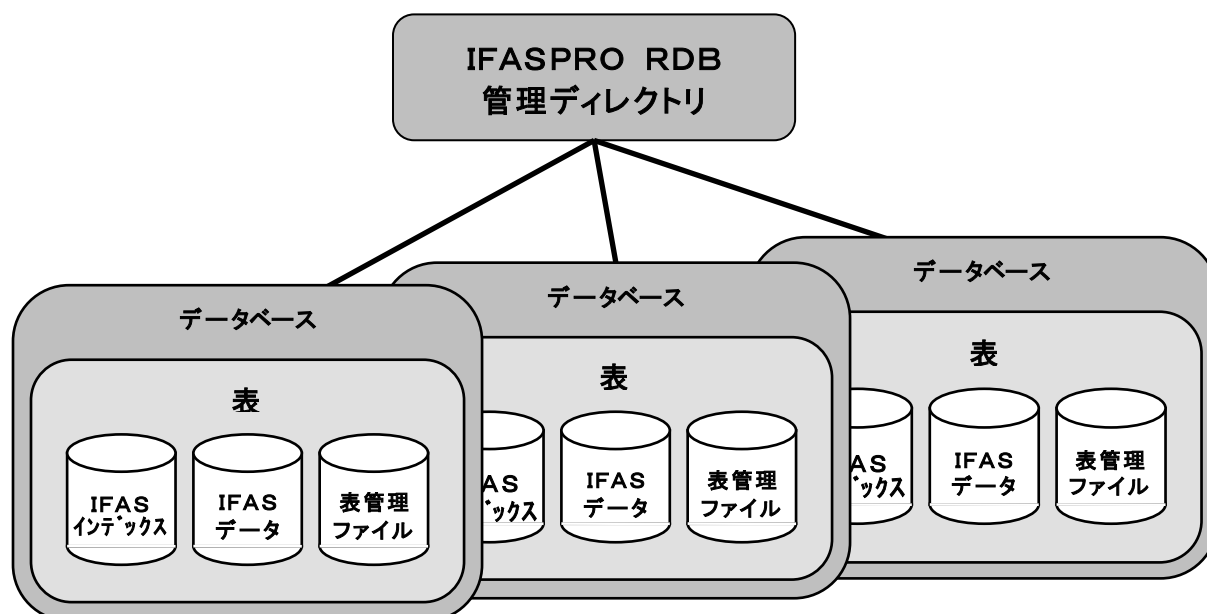
## ファイル構造

順編成ファイルはファイルの先頭にラベルブロックがあり、ファイル情報が格納されています。ラベルブロック以降には固定長のデータレコードが格納され、ラベル情報より最終レコードのレコード番号が管理されています。

基本的に索引順編成ファイルのデータファイル、相対編成ファイルと物理構造は同様の形式になっています。

## 1-2 IFASRDB表

IFASPRO RDBのデータベースの概念図を以下に示します。



### 1-2-1 基本表

実際にデータが格納されている表のことをいい、IFASがサポートしている2つの索引順編成ファイルと表定義情報(列情報等)で構成されます。つまり、1つの基本表は4つの物理ファイルに対応しています。

ファイルを表とみなした時、レコードは行となりフィールド(項目)は列となります。つまり、表定義情報(列情報等)とは、フィールドのデータ項目名、データ型および桁数、さらにインデックスキー定義情報のことを言います。

基本表はデータの参照・更新などを行うことができます。

## 1-2-2 仮想表

1つまたは複数の表内のデータを参照するための表です。基本表のように仮想表自身にデータが格納されているわけではなく、元になる基本表に登録されているデータが仮想表のデータとして扱えます。仮想表の更新は、すなわち基本表の更新となり、その更新は仮想表にも反映されます。

仮想表には、1つあるいは複数の基本表から任意のフィールドを並び換えて作る選択型仮想表と結合型仮想表、複数の基本表のデータを縦に並べた合併型仮想表があります。

IFASPRO RDBで提供している各種仮想表の形態について以下に記述します。

### 選択型仮想表

1つの表から任意のフィールドを選択し並び換えたものです。

選択型仮想表は、基本表と同様、データの参照・更新などを行うことができます。

基本表				選択型仮想表		
商品番号	品名	単価	在庫	商品番号	品名	在庫
0001	テレビ	50.000	100	0001	テレビ	100
0002	パソコン	200.000	150	0002	パソコン	150
0003	ビデオ	150.000	220	0003	ビデオ	220

### 結合型仮想表

複数の表を結合させ任意のフィールドを選択し並び換えたものです。この時、最も優先順位の高い表をプライマリ表といい、その他の表をセカンダリ表（優先順位の高いものから順に、第1セカンダリ表、第2セカンダリ表…）と言います。セカンダリ表は最大62個まで指定できます。

表を結合する場合、互いの表に共通のフィールド（データ型および桁数が同じもの）が存在する必要があります。この時、優先順位の高い表の当該フィールドを対応キー、優先順位の低い表の当該フィールドをアクセスキーと言います。アクセスキーは対応キーのデータを元に索引アクセスするため、インデックスとして定義されていなければなりません。

結合方法にはJOIN（結合）とCHAIN（連鎖）の2種類があり、それぞれデータの見え方が異なります。JOINの一種として、対応キーとアクセスキーをまったく指定しない結合型仮想表を作成することもできます。この場合、プライマリ表とセカンダリ表におけるすべての組み合わせのレコードが、その仮想表のレコードとなります。

選択した表に対して表名とは別の名前を付けることで、同じ表を複数の違う表として扱うことができます。この時に指定する名前を相関名と言います。

結合型仮想表は、データの参照のみ行うことができます。

プライマリ表				セカンダリ表	
商品番号	品名	単価	在庫	商品番号	販売数量
0001	テレビ	50.000	100	0001	150
0002	パソコン	200.000	150	0002	130
0003	ビデオ	150.000	220	0004	55

対応キー				アクセスキー	

結合型仮想表(JOIN)			結合型仮想表(CHAIN)		
商品番号	品名	販売数量	商品番号	品名	販売数量
0001	テレビ	150	0001	テレビ	150
0002	パソコン	130	0002	パソコン	130
			0003	ビデオ	0

## 合併型仮想表

複数の基本表のデータを縦に並べたものです。この時、元となるすべての基本表はフィールドの定義情報(データ型および桁数)が同じである必要があります。各基本表は定義順に第1プライマリ表、第2プライマリ表…といい、最大63個まで指定できます。この仮想表のフィールド情報は元の基本表と同じものになります。

合併型仮想表から任意のフィールドを選択し、選択型仮想表を作成することもできます。

合併型仮想表は、データの参照・更新を行うことができます。ただし、データの追加(WRITE)および相対アクセスはできません。

第1プライマリ表			
商品番号	品名	単価	在庫
0001	テレビ	50.000	100
0002	パソコン	200.000	150
0003	ビデオ	150.000	220

第2プライマリ表			
商品番号	品名	単価	在庫
0004	エアコン	190.000	70
0005	冷蔵庫	100.000	120
0006	洗濯機	80.000	80

合併型仮想表			
商品番号	品名	単価	在庫
0001	テレビ	50.000	100
0002	パソコン	200.000	150
0003	ビデオ	150.000	220
0004	エアコン	190.000	70
0005	冷蔵庫	100.000	120
0006	洗濯機	80.000	80

## 1-3 データ形式

### IFASデータ型

IFAS ファイル内のデータとして使用できるデータ型を以下に示します。

IFASで利用できるデータ形式				IFASデータ型
COBOL上での指定形式				
内部表現形式	USAGE句	SIGN句	PICTURE句	
英数字	USAGE IS DISPLAY	指定不可		IFCHAR
日本語	USAGE IS DISPLAY	指定不可		IFNCHAR
外部10進数	USAGE IS COMP	指定なし		IFUDNS
		SIGN IS TRAILING		IFUDT
		SIGN IS LEADING		IFUDL
		SIGN IS TRAILING SEPARATE		IFUDTS
		SIGN IS LEADING SEPARATE		IFUDLS
内部10進数	USAGE IS COMP-3	指定不可	符号なし	IFPDNS
		指定不可	符号あり	IFPDT
単精度固定2進数	USAGE IS COMP-1	指定不可		IFSBIN2
倍精度固定2進数	USAGE IS COMP-2	指定不可		IFSBIN4
2進数	USAGE IS BINARY	指定不可	符号なし (桁数1～4)	IFUBE2
			符号なし (桁数5～9)	IFUBE4
			符号なし (桁数10～)	IFUBE8
			符号あり (桁数1～4)	IFSBE2
			符号あり (桁数5～9)	IFSBE4
			符号あり (桁数10～)	IFSBE8
	USAGE IS COMP-5	指定不可	符号なし (桁数1～4)	IFUCP2
			符号なし (桁数5～9)	IFUCP4
			符号なし (桁数10～)	IFUCP8
			符号あり (桁数1～4)	IFSCP2
			符号あり (桁数5～9)	IFSCP4
			符号あり (桁数10～)	IFSCP8

注 各データ形式の詳細に関しては、COBOLのマニュアル参照

## RFD編集ツール／表定義ユーティリティ内での表現

RFD編集ツール／表定義ユーティリティのデータ型と、IFASデータ型は以下のように対応しています。

**RFD編集ツール／表定義ユーティリティ内での表現**

RFD編集ツール／表定義ユーティリティ のデータ型	桁数	IFASデータ型
文字		IFCHAR
文字(2BYTE固定)		IFNCHAR
TRAIL符号なし		IFUDNS
TRAIL符号あり		IFUDT
LEAD		IFUDL
TRAIL SEPARATE		IFUDTS
LEAD SEPARATE		IFUDLS
COMP-3符号なし		IFPDNS
COMP-3符号あり		IFPDT
COMP-1		IFSBIN2
COMP-2		IFSBIN4
BINARY符号なし	1～4	IFUBE2
	5～9	IFUBE4
	10～	IFUBE8
BINARY符号あり	1～4	IFSBE2
	5～9	IFSBE4
	10～	IFSBE8
COMP-5符号なし	1～4	IFUCP2
	5～9	IFUCP4
	10～	IFUCP8
COMP-5符号あり	1～4	IFSCP2
	5～9	IFSCP4
	10～	IFSCP8

なお、表定義ユーティリティで、COMP-5(符号あり／なし)の列を作成した場合、列の桁数は以下のようになります。

- ・ 指定桁数が1～4 : 作成される列の桁数は4桁
- ・ 指定桁数が5～9 : 作成される列の桁数は9桁
- ・ 指定桁数が10～ : 作成される列の桁数は18桁

## IFASカスタムコントロール(参考)

IFAS FILE Ver2.0、IFASPRO RDB Ver2.0より、「IFASカスタムコントロール」機能の提供を中止しました。本項の内容は従来バージョンをご利用のお客様が旧製品の仕様を確認いただくために残しています。

カスタムコントロールのGetColumnDataメソッドなどを使用する項目の属性と、IFASデータ型は以下のように対応しています。

カスタムコントロールの項目属性

カスタムコントロールの属性値	内容	IFASデータ型
ifasColumnChar	文字	IFCHAR
ifasColumnJpnChar	日本語	IFNCHAR
ifasColumnDecimal	外部10進数	IFUDNS
		IFUDT
		IFUDL
		IFUDTS
		IFUDLS
ifasColumnNumeric	内部10進数	IFPDNS
		IFPDT
ifasColumnInteger	単精度整数	IFSBIN2
ifasColumnLong	倍精度整数	IFSBIN4
ifasColumnBinary	2進数	IFUBE2
		IFUBE4
		IFUBE8
		IFSBE2
		IFSBE4
		IFSBE8
		IFUCP2
		IFUCP4
		IFUCP8
		IFSCP2
		IFSCP4
		IFSCP8



## 2. 導入と環境設定

IFAS FILE、IFASPRO RDBをシステムに導入して運用する上で、システム動作環境を利用者が設定することができます。利用者の使用形態に合わせて最適な環境設定をすることで、より効率的な運用が可能になります。

またIFAS FILE、IFASPRO RDBは、ファイル障害やシステム障害によって更新途中の不正な状態になってしまったファイルを復旧する手段を実装していますが、そのためには事前の環境設定が必要になります。本章では、IFAS FILE、IFASPRO RDBの導入時初期設定や運用のための環境設定方法について説明しています。

### 2-1 イニシエータ

#### イニシエータとは

IFAS FILEイニシエータは、IFAS FILE、IFASPRO RDBを利用するためのシステム側の環境を初期化します。IFAS FILEイニシエータはIFAS FILE、IFASPRO RDBのセットアップによってシステム立ち上げ時に自動起動されるように設定されます。

したがって、システムを立ち上げると自動的にIFAS FILEイニシエータが起動されて、利用者は特に意識しなくともIFAS利用環境のシステム設定が行われます。

#### アカウントの設定

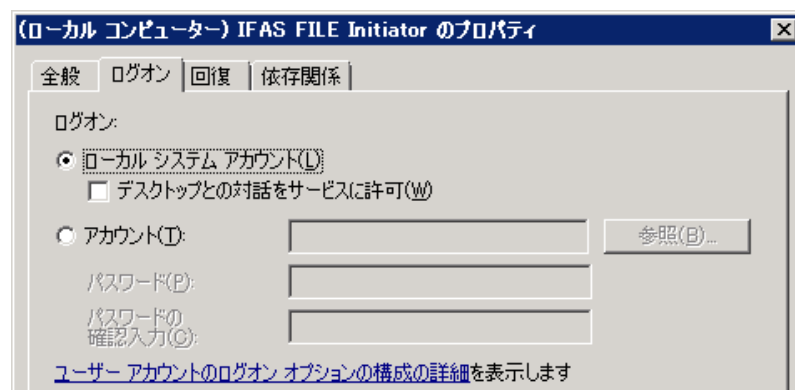
IFAS FILEは、トランザクション処理中に不意の電源ダウンなどがあった場合に、IFAS ファイルを自動復旧するリスタートリカバリ機能(詳しくは「6-4-2 システム障害からの復旧」を参照)、ディスク障害に備えた更新ログの収集機能(詳しくは「6-3 更新ログファイルの運用」を参照)を提供しています。

リスタートリカバリおよび更新ログの収集は、IFAS FILEイニシエータから起動されたプロセスにより行われるため、サービスのアカウントで処理されます。このアカウントの既定値は”ローカルシステム”ですが、以下のような場合には、このアカウントを変更する必要があります。変更先は、”Administrators”グループに属する管理者アカウントである必要があります。

- ・IFASシステムを特定の管理者で管理・運用したい。  
たとえば、リスタートリカバリ機能が有効となるファイルを限定したい、等。
- ・”ローカルシステム”にアクセス権限のないIFASファイルや更新ログを使用したい。

《アカウントの指定例》

1. <コントロールパネル>→<サービス>を開く
2. “サービス(V)”から“IFAS FILE Initiator”の項目を選択
3. [スタートアップ]を選択し、「アカウント」に”Administrators”グループに属する管理者アカウントを設定し、そのパスワードを設定する。



## システムの再初期化

通常、IFAS FILE、IFASPRO RDBを利用している利用者がIFAS FILEイニシエータを意識する必要はありません。しかし、以下の場合には、IFAS FILEイニシエータの再起動によってシステム側の初期化を行う必要があります。

### (1) ロックテーブル初期サイズの変更を行った場合

ロックテーブルサイズが変更された場合には、IFAS FILEイニシエータの再起動が必要です。システム全体で同時に設定可能なロック数はロックテーブルの大きさに依存します。ロックテーブルの初期サイズ変更については「2-2 IFASのSG設定」を参照してください。

### (2) 最大共有バッファサイズの変更を行った場合

最大共有バッファサイズが変更された場合には、IFAS FILEイニシエータの再起動が必要です。システム全体で同時に使用可能なファイル入出力用のバッファ(共有バッファ)領域の最大値を変更した場合はIFAS FILEの初期化が必要です。最大共有バッファサイズの変更については「2-2 IFASのSG設定」を参照してください。

### (3) ロールバックログ出力ディレクトリの変更を行った場合

ロールバックログ出力ディレクトリが変更された場合には、IFAS FILEイニシエータの再起動が必要です。  
ただし、通常は本設定は変更しないでください。

### (4) ファイル識別子管理ファイル設定の変更を行った場合

ファイル識別子管理ファイルの設定が変更された場合には、IFAS FILEイニシエータの再起動が必要です。  
ただし、通常は本設定は変更しないでください。

### (5) IFAS ファイルをクローズせずにプログラムを強制終了させた場合

IFASの利用者プログラムがOPEN関数でオープンしたファイルをCLOSE関数等でクローズしないでプログラムを終了させた場合やファイルがオープンされたままプログラムが異常終了した場合には、IFASの終了処理で未クローズ状態のファイルをクローズします。  
しかし、プログラムが強制終了させられた場合は、IFAS FILEイニシエータの初期化が必要な場合があります。IFAS FILEイニシエータを初期化せずに運用を継続した場合は、IFASの終了処理が不完全なため、ファイルやレコードがビジー状態のままとなります。このような場合、IFAS管理情報メンテナンスツール(詳細は「7-4-4 IFAS管理情報メンテナンスツール」を参照)を使用することにより、ビジー状態を解除することがあります。  
ビジー状態のままやIFAS管理情報メンテナンスツールでビジー状態が解除できなかった場合、他のIFASの利用者プログラムで当該ファイルをアクセスしようとすると、以下のようなエラーステータスが返却されることがあります。

IFFLBUSY : 205 (0x00cd)	ファイルビジー
IFRCBUSY : 302 (0x012e)	レコードロックビジー

また、不正にWait状態になって制御が戻らない場合もあり得ます。

IFAS FILEイニシエータの初期化はサーバ・クライアント管理者が以下の手順でシステム側の初期化を行ってください。

#### システムの初期化手順

IFAS FILEイニシエータを一度停止し、再度起動し直して、システム側の初期化を行います。  
メイングループの<コントロールパネル>を開きます。  
コントロールパネル内の<サービス>を開きます。  
“サービス(V)”から“IFAS FILE Initiator”の項目を選択します。  
“停止(T)”を選択して、IFAS FILEイニシエータを終了させます。  
“開始(S)”を選択して、IFAS FILEイニシエータを再起動します。

## IFAS FILEイニシエータのステータスコード

IFAS FILEイニシエータはIFASシステムの初期化ルーチンで、システムの立ち上げ時に起動します。イニシエータ起動の失敗には以下のような原因が考えられ、以下のようなステータスコードをダイアログにて通知します。また日本語によるメッセージをイベントログ(カテゴリ:アプリケーション)に出力します。

<b>10000 (0x2710)</b>	共有メモリの作成に失敗した
<b>10001 (0x2711)</b>	共有メモリのマッピングに失敗した
<b>10002 (0x2712)</b>	レジストリキーのオープンに失敗した
<b>10003 (0x2713)</b>	レジストリキー値の入手に失敗した
<b>10004 (0x2714)</b>	レジストリキーのクローズに失敗した
<b>10005 (0x2715)</b>	レジストリキー値のサイズが大きすぎる
<b>10006 (0x2716)</b>	サービス制御マネージャの接続に失敗した
<b>10007 (0x2717)</b>	サービス制御関数の登録に失敗した
<b>10008 (0x2718)</b>	サービス状態情報の更新に失敗した
<b>10009 (0x2719)</b>	ディレクトリサービスの初期化に失敗した
<b>10010 (0x271a)</b>	ロック制御サービスの初期化に失敗した
<b>10011 (0x271b)</b>	IFAS FILE、IFASPRO RDBが正しくインストールされていない
<b>10012 (0x271c)</b>	IFASサーバの起動に失敗した

## 2-2 IFASのSG設定

IFASではロックテーブルの初期サイズ、バッファサイズ等のシステム毎の資源や、COBOLアプリケーションのオプションのようなアプリケーション毎のIFASの設定をユーザが容易に行えるように運用支援ツール(IFASSYSG.EXE)を用意しています。運用支援ツールでは以下の項目を設定できます。

- ・ロック制御テーブルの初期サイズ
- ・最大共有バッファサイズ
- ・ロールバックログ出力ディレクトリ(サーバ版のみ)
- ・ファイル識別子管理ファイル名および格納ディレクトリ
- ・COBOL/カスタムコントロール用オプション
- ・更新ログ設定
- ・その他

本ツールはインストールディレクトリ上に存在します。

### ロック制御テーブル設定

IFASでは複数のユーザプログラムから同じIFAS ファイルを同時に矛盾なく処理可能とするために独自のロック制御を行っています。

このロック制御では、ロックに必要とされる資源は特にユーザの上限値設定がないかぎり、システムの仮想メモリサイズまで自動拡張されます。システムの仮想メモリは他のAPでも使用されるため、本ロック制御で使用するこのできるメモリサイズはシステムの設定やAPの起動状況により異なります。他のAPの起動状況に左右されないようにイニシエータ立ち上げ時にあらかじめ確保しておくのがロックテーブル初期メモリです。

ただし、ロックテーブルは共有メモリ上に作成され、利用者プログラムの仮想空間にマッピングされるので、必要以上に大きくとることは好ましくありません。

一度拡張されたロックテーブル用の共有メモリはイニシエータの再立ち上げ(OSの再立ち上げを含む)を行わないかぎり解放されません。

### 最大共有バッファサイズの変更

IFASでは、複数のユーザプログラムから同じIFAS ファイルを同時に使用する場合、入出力用のバッファをシステム一意の共有バッファから確保します。バッファを複数のユーザプログラム間で共有することにより、メモリ使用量を削減するとともに、バッファヒット率の向上が期待できます。共有バッファは1MBを初期値としてユーザの設定する最大共有バッファサイズまで動的に拡張します。

共有バッファはシステム一意に共有メモリ上に1MB確保され、その上限値はIFASのセットアップによって既定値(15Mバイト)の最大共有バッファサイズに設定されます。通常、この値を修正する必要はありませんが、極端に大きいバッファを数多く取るようなユーザプログラムが同時に多数実行されるような場合、共有バッファサイズが不足することがあります。この場合、ユーザプログラムにはIFSHBFERのエラーが返却されます。このエラーが返却された場合は、ユーザプログラムが確保するバッファ数を小さく変更する、システム一意で確保する最大共有バッファサイズを大きくする、等の処置を施す必要があります。

### ロールバックログ出力ディレクトリ

ロールバックログとは、トランザクション処理中のIFAS ファイルの変更を記録したファイルです。

ロールバックログファイルは通常、IFASインストールディレクトリ¥rbk下に自動的に作成されます。通常は、本設定は変更しないでください。

この項目は、クライアントOSでは設定できません(タブも表示されません)。また、サーバOSでも、IFAS PRO RDB Server Runtimeでは利用できません。

## ファイル識別子管理ファイル設定

IFASのファイル識別子管理ファイル(DRSファイル)は、IFAS ファイルのファイル名管理やファイル排他制御等のためにIFASが使用するファイルです。

通常は、本設定は変更しないでください。

## COBOL設定

COBOLアプリケーション/カスタムコントロールでIFAS ファイルのアクセスを行う場合の動作モードやオプションの設定を行ってください。ここで設定した動作は、smart/csやファイルメンテナンスキットなどのCOBOL関連製品やRDB/FILEアクセスコントロールからIFAS ファイルをアクセスする場合にも影響します。

詳細は「4. COBOLプログラムからの利用」を参照してください。

## 更新ログ設定

COBOLアプリケーション/カスタムコントロールでアクセスするIFAS ファイルの更新ログを収集する場合、収集先(更新ログファイル)の設定を行ってください。

表定義ユーティリティでRDB管理ファイルやRDB表を更新した場合も、ここで指定したファイルに更新ログが出力されます。

更新ログの運用方法については、「6-3 更新ログファイルの運用」を参照してください。

## その他

他に、以下のような動作モードを設定できます。

システム再起動時に未クローズファイルのロックを自動的に解除します。

クライアントからネットワークアサインしたファイルを自動的にIFASのクライアント・サーバ接続を行うようにします。

## 2-3 環境設定ファイル

環境設定ファイル“IFASCLT.INI”を使用することで、以下の設定を行うことができます。

- C言語で作成したアプリケーションに対してソースに記述されているファイル名／表名と実ファイル名の関連付けを行います。なお、この機能はカスタムコントロールからも使用できます。
- アプリケーションで使用する最大スレッド数、ファイル数、接続可能なサーバ数、サーバ接続時のタイムアウト値などのIFAS FILE、IFASPRO RDB内部で使用する諸元の設定ができます。

環境設定ファイル“IFASCLT.INI”をクライアント側のWindowsディレクトリ(¥Windows等)に作成してください。IFAS FILE、IFASPRO RDBインストールディレクトリに環境設定ファイルの例がありますので参考にしてください。

### 2-3-1 記述方法

環境設定ファイルは、以下のセクションから構成されています。

#### Table List セクション

サーバ名、IFAS一般ファイル名の関連付け、ファイルオプションの指定を行います。

##### 【書式】

[Table List]

ファイル名A=[<サーバ名A>]ファイル名B[,オプション,[オプション]……]

オプションの一般形式

[,[TRAN|TRANSACTION|NOTRAN|NOTRANSACTIION]][,[LOG|NOLOG]][,NBUF:nnn][,ID  
XBLKSIZ:nn][,[ALTDUPKEY|NOALTDUPKEY]][,[READ02|NOREAD02]]

##### 【説明】

ファイル名Aからファイル名Bに名前が関連付け(変換)されます。オプションは、Cアプリケーションでは無効になりますので、ifas\_logopen関数の呼び出しとオープン関数のパラメータで設定してください。(COBOLアプリケーション、カスタムコントロールを用いたアプリケーションでは、オプションは有効になります)なお、オプションの内容は、COBOLファイルマッパーと同様です。詳細については、「4-2-1 ファイル情報の設定」を参照してください。

##### 【指定例】

[Table List]

FILE01=D:¥LOCAL¥FILE01

FILE01.IDX=D:¥LOCAL¥FILE01.IDX

※具体的な例は「6-1-2 Cプログラムからの利用」を参照してください。

#### Table List for ログオンユーザ名 セクション

サーバ名、IFAS一般ファイル名の関連付け(変換)、ファイルオプションをログオンユーザ毎に指定します。なお、このセクションは、[Table List]セクションより優先されます。

##### 【書式】

[Table List for ログオンユーザ名]

ファイル名A=[<サーバ名A>]ファイル名B[,オプション,[オプション]……]

※ 各項目の内容は[Table List]と同様です。

##### 【指定例】

[Table List]と同様なので省略します。

【補足事項】 ([Table List]、[Table List for ログオンユーザ名]共通の補足事項です。)

#### ファイル名関連付けの指定規則

COBOLアプリケーションでは、下記に示す順にファイル名の関連付けが行われます。

順位 先



順位 後

COBOLファイルマッパーと環境設定ファイルに以下に示すような指定を行った場合、COBOLソース上のファイル名“FILE01”は、まず、COBOLファイルマッパーで“<SVR01>D:¥LOCAL¥FILE01”に変換され、次に環境設定ファイル[Table List]セクションで“<SVR01>D:¥LOCAL¥FILEXX”に変換され、最後に環境設定ファイル[Host List]セクションで“<EXP58-01>D:¥LOCAL¥FILEXX”に変換されます。なお、このような指定はファイル名関連付けの順位を理解するための例ですので、極力避けてください。

COBOLソース上のファイル名の記述

```
FD F0001
  LABEL RECORD IS STANDARD
  VALUE OF IDENTIFICATION IS "FILE01".
```

COBOLファイルマッパーでの指定

```
内部ファイルID    FILE01
デバイス          指定なし
ファイル種別      IFAS FILE
実ファイル名      <SVR01>D:¥LOCAL¥FILE01
```

環境設定ファイル[Table List]セクション

[Table List]

```
<SVR01>D:¥LOCAL¥FILE01=<SVR01>D:¥LOCAL¥FILEXX
```

環境設定ファイル[Host List]セクション

[Host List]

```
SVR01=EXP58-01
```

## DBTable List セクション

サーバ名、IFAS RDB表名の関連付け(変換)、表オプションの指定を行います。

#### 【書式】

[DBTable List]

表A=[<サーバ名A>][データベース名.]表名B[,オプション,[オプション]……]

オプションの一般形式

```
[,[TRAN|TRANSACTION|NOTRAN|NOTRANSACTIION]][,[LOG|NOLOG]][,NBUF:nnn][,ID
XBLKSIZ:nn][,[ALTDUPKEY|NOALTDUPKEY]][,[READ02|NOREAD02]]
```

#### 【説明】

表名Aから表名Bに名前が関連付け(変換)されます。なお、オプションの内容は、COBOLファイルマッパーと同様です。詳細については、「4-2-1 ファイル情報の設定」を参照してください。

#### 【指定例】

[DBTable List]

```
DBTBL01=DB01.DBTBL01
```

## DBTable List for ログオンユーザ名 セクション

サーバ名、IFAS RDB表名の関連付け(変換)、表オプションを**ログオンユーザ毎**に指定します。なお、このセクションは、[DBTable List]セクションより優先されます。

### 【書式】

[DBTable List for ログオンユーザ名]

表A=[<サーバ名A>][データベース名]表名B[,オプション,[オプション]……]

※各項目の内容は[DBTable List]と同様です。

### 【説明】

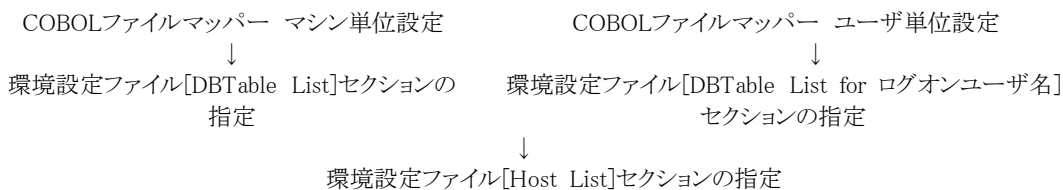
[DBTable List]と同様なので省略します。

【補足事項】([DBTable List]、[DBTable List for ログオンユーザ名]共通の補足事項です。)

#### 表名関連付けの指定規則

COBOLアプリケーションでは、下記に示す順にファイル名の関連付けが行われます。

順位 先



順位 後

COBOLファイルマッパーと環境設定ファイルに以下に示すような指定を行った場合、COBOLソース上の表名“TBL01”は、まず、COBOLファイルマッパーで“<SVR01>IFASDB.TBL01”に変換され、次に環境設定ファイル[DBTable List]セクションで“<SVR01>IFASDB.TBLXX”に変換され、最後に環境設定ファイル[Host List]セクションで“<EXP58-01>IFASDB.TBLXX”に変換されます。なお、この指定はファイル名関連付けの順位を理解するための例ですので、極力避けてください。

COBOLソース上の表名の記述

```
FD F0001
  LABEL RECORD IS STANDARD
  VALUE OF IDENTIFICATION IS "TBL01".
```

COBOLファイルマッパーでの指定

```
内部ファイルID    TBL01
デバイス          RDB
ファイル種別      IFASPRO RDB
実ファイル名      <SVR01>IFASDB.TBL01
```

環境設定ファイル[DBTable List]セクション

```
[DBTable List]
<SVR01>IFASDB.TBL01=<SVR01>IFASDB.TBLXX
```

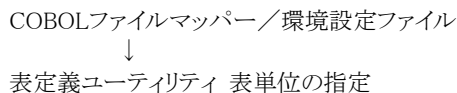
環境設定ファイル[Host List]セクション

```
[Host List]
SVR01=EXP58-01
```

表オプションの優先順位

COBOLアプリケーションでは、IFAS RDB表のオプションの優先順位は下記のとおりです。

順位 高



順位 低



## Host List セクション

[Table List]/[Table List for ログオンユーザ名]/[DBTable List]/[DBTable List for ログオンユーザ名]セクション、COBOLファイルマッパーやソース上に記述されているサーバ名の関連付け(変換)の指定を行います。

### 【書式】

[Host List]  
サーバ名A=サーバ名B

### 【説明】

サーバ名Aから、サーバ名Bに名前が関連付けられます。

### 【指定例】

[Host List]  
HOST01=EXP58-01  
サーバ名<HOST01>は、EXP58-01に関連付けられるようになります  
[Table List]  
FILE02=<HOST01>C:¥IFASFILES¥FILE02  
ファイル名FILE02は、<EXP58-01>C:¥IFASFILES¥FILE02に関連付けられます

## Control セクション

アプリケーションで使用する最大スレッド数、ファイル数、接続可能なサーバ数、サーバ接続時のタイムアウト値などのIFAS FILE、IFASPRO RDB内部で使用する諸元の設定ができます。

### 【書式】

[Control]  
ConnectTimeout=40  
ServerHostMax=16  
FileMax=128  
ThreadMax=128  
※各設定値は既定値を指定しています。

### 【説明】

ConnectTimeout=n  
クライアントアプリケーションがサーバ接続する場合のタイムアウト値(秒)です。タイムアウト時間経過してもサーバへの接続が完了しない場合、IFAS FILE、IFASPRO RDBはエラーを返却します。  
(既定値:40)

ThreadMax=n  
クライアントアプリケーションで、IFAS FILEを使用するスレッドの最大数です。  
マルチスレッドプログラミングを行う場合、この値の指定が必要になる場合があります。  
(指定可能な最大値:1024、既定値:128)

ServerHostMax=n  
アプリケーションからの最大同時接続サーバ数を指定します。IFASではスレッド毎にコネクションを行うので、アプリケーションスレッド数×接続サーバ数を指定してください。ローカルファイルをアクセスする場合、ファイルリカバリ機能を強化するため、擬似的なサーバ(ローカルサーバ)接続を行っているので、アプリケーションスレッド数×接続サーバ+アプリケーションスレッド数を指定してください。  
(指定可能な最大値:1024、既定値:16)

FileMax=n  
IFAS FILE、IFASPRD RDB内部で用いている、ファイル管理領域の初期サイズです。  
同時にアクセスするファイル数を指定してください。  
(指定可能な最大値:4096、既定値:256)

### 【指定例】

[Control]  
ConnectTimeout=5  
ThreadMax=200

ServerHostMax=200  
FileMax=400

最大200スレッド生成する  
最大400個のファイルをオープンする

## 2-3-2 環境設定ファイルの動的切り替え

環境変数「IFASTABLES」を定義することで、[Table List]および[Table List for ログオンユーザ名]セクションを定義するファイルを切り替えることができます。また、環境変数「IFASTABLES」を定義することで、[DBTable List]および[DBTable List for ログオンユーザ名]セクションを定義するファイルを切り替えることができます。この機能とバッチファイルを組み合わせることで、ファイルや表のオプションを動的に切り替えることが可能になります。

### IFAS一般ファイルの場合

#### 【説明】

環境変数「IFASTABLES」を定義することで、[Table List]および[Table List for ログオンユーザ名]セクションを定義するファイルを切り替えることができます。環境変数「IFASTABLES」に[Table List]または[Table List for ログオンユーザ名]セクションを記述した設定ファイル名を指定してください。

IFASCLT.INIファイルの[Table List]および[Table List for ログオンユーザ名]セクションを無視し、環境変数が示すファイルからファイルの定義情報を参照するようになります。なお、[Control]/[DBTable List]/[DBTable List for ログオンユーザ名]セクションには影響しません。

#### 【指定例】

アプリケーションAP.EXEのCOBOLソース上のファイル名の記述

```
FD F0001
  LABEL RECORD IS STANDARD
  VALUE OF IDENTIFICATION IS "FILE01".
```

COBOLファイルマッパーでの指定

```
内部ファイルID    FILE01
デバイス          指定なし
ファイル種別      IFAS FILE
実ファイル名      D:¥LOCAL¥FILE01
```

環境設定ファイルA.INIファイルの[Table List]セクションの指定

```
[Table List]
D:¥LOCAL¥FILE01=D:¥LOCAL¥FILE01,TRANSACTION
```

環境設定ファイルB.INIファイルの[Table List]セクションの指定

```
[Table List]
D:¥LOCAL¥FILE01=D:¥LOCAL¥FILE01
```

アプリケーション起動用バッチファイルA.BATの指定

```
SET IFASTABLES=D:¥DIR¥A.INI
AP.EXE
```

環境設定ファイルA.INIのファイル名はフルパスで指定してください。

バッチファイルA.BATからアプリケーションAP.EXEを起動した場合、トランザクション処理が有効になります。

アプリケーション起動用バッチファイルB.BATの指定

```
SET IFASTABLES=D:¥DIR¥B.INI
AP.EXE
```

環境設定ファイルB.INIのファイル名はフルパスで指定してください。

バッチファイルB.BATからアプリケーションAP.EXEを起動した場合、トランザクション処理が無効になります。

## IFASRDB表の場合

### 【説明】

環境変数「IFASDBTABLES」を定義することで、[DBTable List]および[DBTable List for ログオンユーザ名]セクションを定義するファイルを切り替えることができます。

環境変数「IFASDBTABLES」に[DBTable List]または[DBTable List for ログオンユーザ名]セクションを記述した設定ファイル名を指定してください。

IFASCLT.INIファイルの[DBTable List]および[DBTable List for ログオンユーザ名]セクションを無視し、環境変数が示すファイルからRDB表の定義情報を参照するようになります。なお、[Control]/[Table List]/[Table List for ログオンユーザ名]セクションには影響しません。

### 【指定例】

アプリケーションAP.EXEのCOBOLソース上のファイル名の記述

```
FD F0001
  LABEL RECORD IS STANDARD
  VALUE OF IDENTIFICATION IS "DBTBL01".
```

COBOLファイルマッパーでの指定

内部ファイルID	DBTBL01
デバイス	RDB
ファイル種別	IFASPRO RDB
実ファイル名	DB01.DBTBL01

環境設定ファイルA.INIファイルの[DBTable List]セクションの指定

```
[DBTable List]
DB01.DBTBL01=DB01.DBTBL01,TRANSACTION
```

環境設定ファイルB.INIファイルの[DBTable List]セクションの指定

```
[DBTable List]
DB01.DBTBL01=DB01.DBTBL01
```

アプリケーション起動用バッチファイルA.BATの指定

```
SET IFASDBTABLES=D:¥DIR¥A.INI
AP.EXE
```

環境設定ファイルA.INIのファイル名はフルパスで指定してください。

バッチファイルA.BATからアプリケーションAP.EXEを起動した場合、トランザクション処理が有効になります。

アプリケーション起動用バッチファイルB.BATの指定

```
SET IFASDBTABLES=D:¥DIR¥B.INI
AP.EXE
```

環境設定ファイルB.INIのファイル名はフルパスで指定してください。

バッチファイルB.BATからアプリケーションAP.EXEを起動した場合、トランザクション処理が無効になります。

## 2-3-3 注意事項

IFASCLT.INIは必ずWindowsディレクトリに格納してください。IFASインストールディレクトリ直下では有効になりません。環境設定ファイルは、すべてのIFASを用いるアプリケーション(IFASのユーティリティも含まれます)で有効になります。

COBOLアプリケーションからIFAS一般ファイルやIFAS RDB表を利用する場合には、**COBOLファイルマッパー**を利用してファイルや表のオプション設定等を行ってください。COBOLファイルマッパーにおけるファイルオプションの設定方法については「**4-2 ファイル情報と動作環境の設定**」を参照してください。

環境設定ファイルによるファイル名(表名)の変更やオプションの設定は、以下の場合にのみ使用することを推奨します。

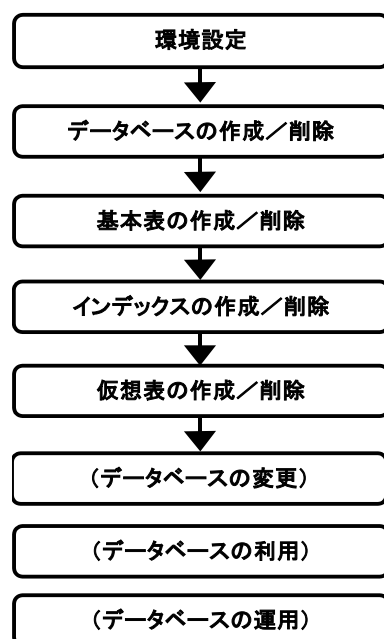
- ・Cアプリケーションを実行する場合
- ・カスタムコントロールアプリケーションを実行する場合
- ・環境設定ファイルの動的切り替え機能を使用する場合

同一ファイルに対して、COBOLファイルマッパー/環境設定ファイルの両方にオプションを設定することは誤動作の原因になりますので避けてください。

## 2-4 RDBの環境設定

### 2-4-1 導入手順

データベースのシステム導入は、次の手順で行います。



導入時には必ず環境設定を行い、RDB管理ディレクトリを初期化してデータベース構築を行ってください。

### 2-4-2 環境設定

環境設定を行うことにより、RDB環境の定義情報を管理する管理ファイルを作成します。これを、RDB管理ファイルと言います。

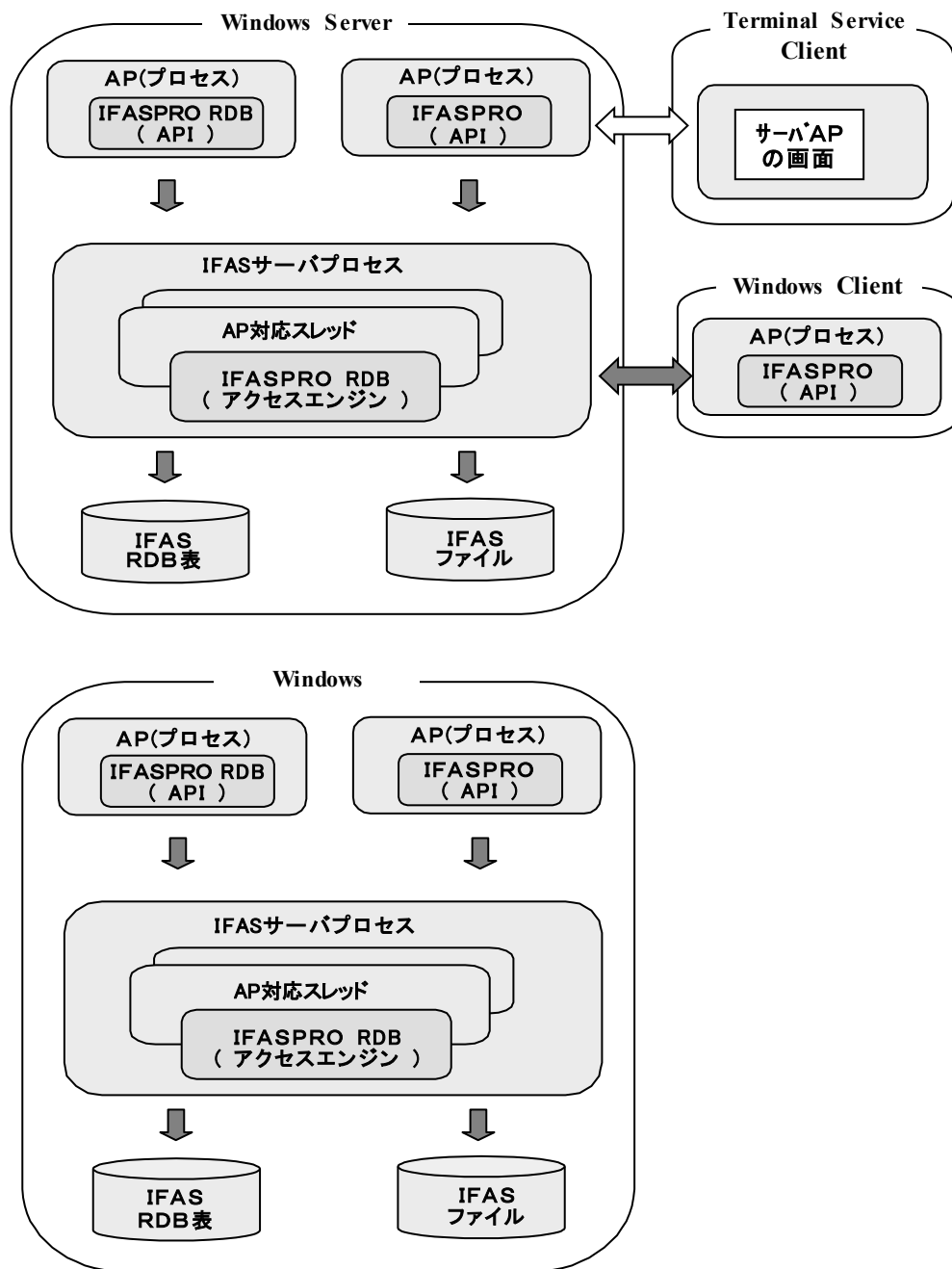
また、各種動作設定も指定することができます。

環境設定の変更は、表定義ユーティリティにて行うことができます。

実行方法は「7-2-1 表定義ユーティリティ」を参照してください。

## 2-5 アクセス方式

IFAS FILE、IFASPRO RDBでは、複数のユーザプロセスがIFASサーバ(IFAS FILE イニシエータ)のアクセスエンジンを経由し、IFASファイルへのアクセスを行います。IFASサーバ内では、各ユーザプロセスに対応したスレッドが起動され、共有メモリを介してユーザプロセスと通信します。IFASサーバ(IFAS FILE イニシエータ)を経由してIFASファイルへのアクセスを行うのは、IFASファイルの保全性を高めるためであり、IFASサービスを経由することにより、ユーザプロセスを強制終了させた場合も、IFASサービス内で確実にクローズ処理を行うことが可能になります。



## 3. Cプログラムからの利用

CプログラムからはIFAS一般ファイル(索引順編成ファイル、相対編成ファイル、順編成ファイル)に対して、各種ファイル操作、データアクセス処理、トランザクション処理を行うことが可能です。

本章ではCプログラムから利用できる、IFAS一般ファイルに対するアクセス機能、およびIFAS FILEの特徴である排他／同時アクセス制御、トランザクション処理機能、ファイルリカバリ機能、またCプログラムから利用できるIFAS一般ファイルアクセス関数について説明します。

なお、CプログラムからIFAS RDB表をアクセスすることはできません。

### 3-1 機能概要

#### 3-1-1 ファイルアクセス機能

IFAS FILEでは各編成ファイルに対して以下のような機能を提供します。ここで紹介する機能は主にC言語向けに提供する関数に対応しています。関数の詳細については「3-2 IFAS FILE関数」を参照してください。

#### ファイルの作成

利用者が指定したファイル名(索引順編成ファイルの場合はインデックスファイル名、データファイル名)のファイルを指定したファイル属性に従ってシステム上に新たに作成します。

#### ファイルのオープン

システム上のIFAS ファイルに対し、レコード入出力操作を行うための準備を行います。処理目的に従ってアクセスモードとロックモード、トランザクションモード等を組み合わせてパラメータで指定します。

##### アクセスモード

###### ・出力モード

ファイルを初期化して新規にデータを作成するためのアクセスモードで、レコードの書き出し操作のみを許可します。

索引順編成ファイルの場合、作成するインデックスの属性を指定することが可能です。

相対編成ファイルの場合、あらかじめ書き込む予定のレコード数分のファイルサイズを確保することが可能です。

###### ・入力モード

ファイル内のデータを参照するためのアクセスモードで、レコードの読み込み操作、レコードの検索操作、レコードの位置付け操作のみを許可します。

###### ・更新モード

ファイルの内容を変更するためのアクセスモードで、すべてのレコード操作を許可します。

##### ロックモード

###### ・ファイル排他ロック

ファイルをクローズするまでの間、他の利用者との共有を一切禁止します。アクセスモードを出力モードでオープンする場合や、索引順編成ファイルにインデックスの追加・削除を行う場合は必ず指定しなければなりません。

###### ・ファイル共有ロック

ファイルをクローズするまでの間、同様のモードでオープンした他の利用者に対してのみ共有を許可します。ただし、アクセスモードが入力モードの場合しか指定できません。

###### ・レコードロック

ファイルをクローズするまでの間、同様のモードでオープンした他の利用者に対してのみ共用を許可します。トランザクション処理等における、レコード単位での排他共有制御をIFASが自動的に行います。すなわち、レコードの参照系の処理時は共有ロックを設定し、更新系の処理時は排他ロックを設定します。ただし、アクセスモードが出力モードの場合は指定できません。

### トランザクションオプション

本オプションはトランザクション処理を行う場合や、更新ログ収集操作を行っている場合にファイル単位で任意に設定可能なオプションパラメータです。

#### ・ロールバックオプション

トランザクション処理を取り消したい時やプログラムが異常終了した時にファイルの状態を直前のトランザクション開始時点まで戻す機能(ロールバック機能)の利用を宣言します。このオプションを指定しない場合はロールバック機能の要求は無視されます。

#### ・更新ログ未収集オプション

更新ログファイルがオープンされている場合でも、本オプションが指定されたファイルの更新ログは収集しません。

### 索引順編成ファイル属性の既定値変更

出力モードでオープンする場合に、以下のファイル属性の既定値を変更してファイルを初期化することができます。

#### ・インデックスブロックサイズ

本パラメータはインデックスファイルへのアクセス性能に影響を与えますので、性能のチューニングのために活用できます。

1つのインデックスブロックに格納できるキーエントリ数(ブロック化係数)が全体のレコード数と比較して少なすぎるとインデックスの階層が深くなり、検索時のディスクI/Oの回数が多くなります。逆に、ブロック化係数が大きすぎると階層が浅くなり検索効率が落ちます。これらは、本パラメータ値(既定値は1024バイト)の変更によって調整できます。また、インデックスの階層は20を超えることができないので、ファイルに格納される最終的なレコード数を考慮して、インデックスブロックサイズを決定する必要があります。

なお、インデックスブロックサイズはファイル毎に一意に設定すべき値であるので、複数のインデックスから成るファイルの場合は、そのインデックスの中で最も大きいキーサイズを基準に設定してください。

#### ・インデックスブロックの占有率

出力モードでキー昇順にデータ出力を行うと、基本的にキーエントリはインデックスブロックに格納できる数だけ格納されます。つまり、各ブロックに隙間なくキーエントリが詰め込まれるので、その後、更新モードでこのファイルをオープンして中間値となるキーエントリを追加すると、そのキー値に対応するインデックスブロックは、必ず分割処理が行われます。ブロック分割処理自体は問題ありませんが、多発するようであればファイルのスペース効率やアクセス性能が低下します。

したがって、ファイルの初期化後にレコードを追加する予定のあるファイルは、出力モードでファイルをオープンする時(初期化時)に、各インデックスブロックに追加用の領域を確保しておくことによって、レコード追加によるブロック分割処理を発生しにくくすることができます。

占有率は100%~50%の範囲で指定できます。

### 相対編成ファイルの最大レコード番号の指定

最大レコード番号はファイルサイズの確保や拡張、入力レコードの論理終端検出のために指定します。オープンモードによって以下のように指定の意味は異なります。

#### ・出力モードの場合

最大レコード番号を指定することによって、レコードを出力していく際に最大レコード番号を超えるレコードの出力をエラーにするか否かを指定することも可能です。また、最大レコード番号までのレコード数を格納するだけのファイルサイズを初期サイズとしてオープン時に確保することもできます。

最大レコード番号およびファイル領域確保指示を指定しない場合はファイルサイズ=ラベルブロックサイズとしてオープンし、レコードを書き込む毎に動的に自動拡張します。

#### ・更新モードの場合

指定最大レコード番号がファイル上の最大レコード番号より大きい場合は、その値をファイルサイズ動的拡張の上限値とします。(拡張エラーチェック指定の場合)



指定最大レコード番号がファイル上の最大レコード番号より小さい場合は、指定値を一時的なファイル終端レコードとして扱うことが可能です。この場合指定値より大きいレコード番号の实在レコードのアクセスを抑制することができます。

最大レコード番号の指定がない場合は実際の最大レコード番号と同じ値を指定したとみなします。

#### ・入力モードの場合

指定最大レコード番号がファイル上の最大レコード番号より大きい場合は無効です。

指定最大レコード番号がファイル上の最大レコード番号より小さい場合は、指定値を一時的なファイル終端レコードとして扱うことが可能です。この場合指定値より大きいレコード番号の实在レコードのアクセスを抑制することができます。

最大レコード番号の指定がない場合は実際の最大レコード番号と同じ値を指定したとみなします。

### 相対編成ファイルのファイル拡張指示

相対編成ファイルはレコード番号を指定してダイレクトにレコード出力することが可能なため、利用者が誤って大きな値を指定した場合、そこまでファイルサイズを拡張してしまいます。そのため指定された最大レコード番号、または従来の最大レコード番号を超えたレコードの出力をエラーにするか否かを選択できます。

### 相対編成ファイルのファイル領域確保指示

相対編成ファイルを出力モードでオープンする場合、指定する最大レコード番号を格納するだけのファイル領域をオープン時にあらかじめ確保しておくことができます。また、従来から存在するファイルを出力モードでオープンする場合、そのファイルの従来のファイル領域をそのまま利用することも可能です。

ファイル領域をあらかじめ確保しておくこととレコード出力時の動的拡張が発生しないため処理性能の向上を期待することができます。

### アクセス属性の既定値変更

ファイル処理における、アクセス効率／メモリ効率等を調整するために以下のアクセス属性の既定値を変更することができます。

#### ・入出力バッファ数

アクセスモードに関係なく、I/O時のバッファ数を変更することができます。

索引順編成ファイルの場合、データファイルとインデックスファイルそれぞれのI/Oバッファ数の既定値は10です。検索処理を主に行う場合は、インデックスファイルのバッファ数を大きく指定することにより、ルートインデックス等、参照が多発する上位インデックスブロックのバッファがメモリに常駐する可能性が高くなるのでディスクI/Oを削減できます。

データファイルについても同一のレコードを何度も参照するような処理を行う場合はデータファイルのバッファ数を大きく指定することにより、バッファ内でのヒット率が高くなり、ディスクI/Oの削減が期待できます。

相対編成ファイルのI/Oバッファ数の既定値は20です。同一のレコードを何度も参照するような処理を行う場合はデータファイルのバッファ数を大きく指定することにより、バッファ内でのヒット率が高くなり、ディスクI/Oの削減が期待できます。

順編成ファイルのI/Oバッファ数の既定値は20です。順編成ファイルの場合はバッファ数分のレコード数分を1回のディスクI/Oの単位とします。よってバッファ数が大きいほどディスクI/Oの回数が削減されます。

ただし、バッファ数を大きく指定することはメモリ領域を大きく使用することになりますので、プログラムサイズや同時にオープンして使用するIFAS ファイルの数を考慮しなければなりません。

## ファイルのクローズ

ファイルに対する処理を終了します。同時にファイル操作に確保したメモリ資源を解放し、ファイルとレコードに対してのロックをすべて解除します。

## レコードの書き出し

ファイルに対してレコードを書き出します。索引順編成ファイルの場合は同時にレコードに対するキーをインデックスに追加します。

## インデックスの選択

索引順編成ファイルのキー検索処理を行うためのインデックスの選択、または変更を行います。

## レコードの読み込み

利用者の指定した条件に従ってファイルからレコードを読み込み、そのレコードをカレントレコードとします。

## レコードの書き換え

ファイル内のレコードを利用者の指定したレコードで書き換えます。

## レコードの削除

ファイル内のレコードを削除します。索引順編成ファイルの場合はそのレコードに対応したキーエントリーも削除します。

順編成ファイルに対して本機能は使用できません。

## レコードの位置付け

インデックスキー、またはレコード番号によって、レコード検索を行い、そのレコードをカレントレコードとします。

順編成ファイルに対して本機能は使用できません。

## カレントレコードの変更

カレントレコードを前後方向に変更します。

## バッファの書き出し

更新されたバッファをすべてファイルに書き出します。

## インデックスの追加

索引順編成ファイルに対し新たにサブインデックスを作成します。

## インデックスの削除

索引順編成ファイルからサブインデックスを削除します。

## ファイルの削除

利用者が指定したファイル名(索引順編成ファイルの場合はインデックスファイル名、データファイル名)のファイルをシステム上から削除します。

## ファイル名の変更

利用者が指定したファイル名(索引順編成ファイルの場合はインデックスファイル名、データファイル名)のファイルを指定した新ファイル名に変更します。

## ファイル情報の取得

IFAS ファイルの属性情報(インデックス数、データレコード長、インデックスブロック長、データレコード数等)を取得します。

## インデックス情報の取得

索引順編成ファイルに対して、指定したインデックス番号のインデックス情報をキー記述構造体の形式で取得します。

## レコード番号の取得

直前に処理したレコードのレコード番号を取得します。

## インデックスの再編成

作成済みの索引順編成ファイルのインデックスを再編成します。

## インデックスファイルの再作成

作成済みの索引順編成ファイルのインデックスファイルがデータファイルと不整合となってオープン不可能になった場合、インデックスファイルを作り直します。

## インデックスファイルの作成

作成済みの相対編成ファイルや順編成ファイルにインデックスファイルを追加作成し、索引順編成ファイルとしても扱えるようにします。本関数ではプライマリインデックスのみ作成することが可能です。

## 各ファイル編成間互換機能

IFASは、ファイルの利用目的によって、ある編成のファイルを他編成ファイルとして使うことが可能なように以下のような互換機能を提供します。

### 索引順編成ファイルへの変換

順編成ファイルや相対編成ファイルとして作成したファイルに、後からインデックスファイルを追加作成し、索引順編成ファイルに作り変えることが可能です。

### 順／相対編成ファイルとしての利用

索引順編成ファイルのデータファイルのみを順編成ファイルや相対編成ファイルとして使用することが可能です。ただし、順編成ファイルや相対編成ファイルとしてレコードの追加／削除／更新を行った場合、そのデータファイルは索引順編成ファイルとしてインデックスファイルとの整合がとれなくなるので、再び索引順編成ファイルとして利用する場合はインデックスファイルを再作成しなければなりません。

### 順編成ファイルと相対編成ファイル

相対編成ファイルとして作成したファイルを順編成ファイルとして、順編成ファイルとして作成したファイルを相対編成ファイルとして使用することが可能です。

相対編成ファイルの順次読み込みを行う場合は、順編成ファイルとしてオープンして処理するとより高速な処理が可能となります。

順編成ファイルを相対編成ファイルとしてオープンすることにより、ダイレクトレコードアクセス、レコード削除処理が可能になります。

## 3-1-2 排他／同時アクセス制御

IFASのロック制御はファイルロックとレコードロックの2種類のロックモードに大別され、それぞれ共有ロックと排他ロックの2種類のロック種別によって制御します。

ロックモードはファイルオープン時にOPEN関数のパラメータで指定し、ファイルをクローズするまで変更することはできません。また、ファイルロックモードの場合はロック種別もファイルオープン時に設定され、ファイルクローズするまで変更はできません。

レコードロックモードの場合は、利用者のレコード操作内容に応じてIFASが自動的にレコードロック種別を決定しロック制御を行います。基本的には参照するレコードは共有ロック、追加／更新／削除するレコードは排他ロックされます。また、索引順編成ファイルの場合は、レコード処理に伴って参照／更新されたインデックスブロックもロック制御の対象となります。

## ファイルロック

ファイルロックモードのロック種別は、ファイルオープン時のアクセスモードとの組み合わせで、以下のように指定可能な場合と不可能な場合があります。

### ファイル排他ロック

出力、入力、更新、すべてのアクセスモードで指定可能です。逆にアクセスモードが出力モードの場合はファイルを初期化するので、必ずファイル排他ロックモードを指定しなければなりません。

ファイル排他ロックモードでオープンすると、他のIFAS利用者はこのファイルをオープンすることはできなくなります。反対に、そのファイルを使用している他の利用者が先に存在する場合は、ファイルビジーでオープンエラーとなります。

つまり、ファイルを1利用者だけで占有して処理する場合に有効な手段です。

### ファイル共有ロック

オープン時のアクセスモードが入力モードの場合のみ指定可能で、ファイルの参照しか行えないので、複数のIFAS利用者が同時にファイルをオープンすることができます。そのファイルを、ファイル排他ロックやレコードロックでオープンしている他の利用者が先に存在する場合は、ファイルビジーでオープンエラーとなります。

## レコードロック

ファイルロックモードはファイルを占有する場合、またはファイルの参照のみを行う場合に指定するロックモードですが、複数利用者が1つのファイルを同時に更新処理を行う場合、つまり、トランザクション処理等を行う場合はレコードロックモードを指定しなければなりません。

レコードロックモードを指定した場合、IFASは利用者のレコード操作に応じて自動的にレコードロック制御を行います。ただし、オープン要求時に他の利用者がファイルロックモードで同ファイルを処理中だった場合は、ファイルビジーでオープンエラーとなります。

IFASが自動的に行ったレコードロックは利用者が以下の処理を実行することによって解除することができます。

- ファイルのクローズ (CLOSE関数)
- トランザクションの終了 (COMMIT関数)
- トランザクションの取り消し (ROLLBACK関数)

### レコード共有ロック

レコードの読み込み操作を行う場合、そのレコードには共有ロックが要求されます。

共有ロックが成功しレコードの読み込みが完了すると、他の利用者は同じレコードを参照することはできますが、削除や書き換えはできなくなります。

逆に、他の利用者が先に更新した(排他ロック中)レコードを読み込もうとすると、共有ロックは先の利用者のロックが解除されるまで待たされます。

### レコード排他ロック

レコードの書き出し・書き換え・削除の処理を行う場合、そのレコードには排他ロックが要求されます。排他ロックが成功すると、他の利用者は同レコードに対するすべての操作ができなくなります。

逆に、他の利用者が先に参照した(共有ロック中)レコードに対し、書き換え・削除をしようとすると、排他ロックは先の利用者のロックが解除されるまで待たされます。

### デッドロック

レコードロック制御において、2者以上の利用者同士で互いにレコードロックを待たせ合う状態をデッドロックと呼びます。この場合、後からロックを要求した側の利用者にはデッドロックエラーが返却されます。

デッドロックエラーを通知された利用者は、直ちにトランザクション処理の取り消しを行わなければなりません。

## 3-1-3 トランザクション処理

トランザクション処理とは、1つまたは複数のファイルに対する複数のレコード操作を一連の完結した処理としてまとめた処理単位のことを言います。

利用者はこのトランザクション処理の開始／終了をIFASに通知することによって、1つのトランザクション処理終了毎に操作内容をファイルに反映させロックを解除できます。また、トランザクション処理途中のエラー発生時に、それまでの操作をすべて取り消してファイルをトランザクション処理開始時点の状態に戻すこともできます。

## トランザクション対象ファイル

トランザクション処理で使用するファイルは、トランザクション取り消し処理が発生した場合にファイルをトランザクション開始時点の状態に戻す(ロールバック)必要があるため、ファイルオープン時にロールバックを行うファイルであることを指定しなければなりません。

なお、ファイル状態を戻す必要のないファイルに対しては、ロールバック指定をしてオープンする必要はありません。

## トランザクション処理の開始

利用者は、1つのトランザクション処理における、最初のIFASに対するレコード操作の前に、トランザクション処理の開始をIFASに通知しなければなりません。

IFASはこの通知処理によって、ファイルをロールバックする必要がある場合の基点を定めることができます。

## トランザクション処理の終了

利用者は、1つのトランザクション処理における、最後のレコード操作が終了した後で、トランザクション処理の終了をIFASに通知(コミット)しなければなりません。

IFASはこの通知処理によって、すべてのファイルの更新をファイルに反映させて、トランザクション処理中のレコードロックをすべて解除します。

## トランザクション処理の取り消し

利用者は、トランザクション処理中に何らかのエラーを検出した時等に、そのトランザクション処理を取り消すことができます。トランザクション処理の取り消しを行った場合には、すべてのファイルをトランザクション開始時の状態にロールバックしトランザクション処理中のレコードロックをすべて解除します。

# 3-1-4 ファイルリカバリ処理

## 更新ログ収集機能

IFASは、利用者が実行したファイルに対する操作内容を、更新ログとして記録する機能を提供します。更新ログは利用者が指定する更新ログファイルに記録収集し、ロールフォワードユーティリティを実行する時に参照されます。

更新ログファイルへのログの記録収集はLOGOPEN関数によって開始され、ログ収集したいファイルをオープンする前には必ず実行しなければなりません。

## ロールバック機能

更新モードでオープンしたファイルにおいて、トランザクション処理の途中でレコード操作が失敗したり、利用者レベルでエラーを検出した場合に、トランザクション処理として途中まで処理したレコード操作をすべて取り消して、ファイルをそのトランザクション処理を開始する前の状態(トランザクションスタート時点)に戻すことができます。

IFASはロールバックログとして出力された、処理中トランザクション開始時点以降のレコード更新前のイメージや削除前のレコードイメージによって、ファイルに対して利用者が行った処理と逆のレコード操作を行いファイルを復旧します。

ロールバック用のログはIFASが内部的に作成するロールバックログファイルに出力します。

また、プログラムが異常終了した場合に、利用者が例外処理を設定してIFASの例外処理用の関数を実行すれば、IFASは自動的にロールバック処理を行って直前のトランザクション終了時までのファイル状態を保証します。

ただし、ロールバックを行うファイルは、オープン時にロールバック対象のファイルであることを指定しておかなければなりません。

なお、以下に記述する処理はロールバックによって取り消すことはできません。

- ・ファイルの作成／削除／名前変更
- ・ファイルのオープン／クローズ
- ・インデックスファイルの作成／再作成
- ・索引順編成ファイルのインデックス再編成

## リスタートリカバリ機能

IFAS ファイルを更新するトランザクションプログラムを実行中、システムの電源ダウン等が発生した場合、システム再起動時に更新処理途中のIFAS ファイルをすべて自動的にロールバックします。

## ディファードリカバリ機能

リスタートリカバリが何らかの原因で失敗した場合、ユーザオペレーションによってリスタートリカバリと同等の処理を行うことができます。本機能はユーティリティとして提供され、ロールバック未完了状態になっているシステム下のIFAS ファイルを一括してロールバックを行います。

## ファイルロックリセット機能

ロールバックオプション指定でオープンして更新処理中だったファイルは、電源ダウン後のリスタート時に、**リスタートリカバリ機能**や**ディファードリカバリ機能**で復旧することが可能ですが、ロールバックオプションを指定せずにオープンして更新中だったファイルはリカバリ対象とはならないため、リスタート後は“正常にクローズされなかったファイル”という未クローズロック状態になり、入力／更新モードでオープンすることが不可能になります。

未クローズ状態のファイルは出力モードオープンして最初から作り直すことは可能ですが、どうしてもこのファイルのデータを参照したい場合のために未クローズロックを解除する機能をユーティリティとして提供します。

本機能は**リスタートリカバリ機能**や**ディファードリカバリ機能**に失敗したファイルに対して適用することも可能です。

## ロールフォワード機能

ディスク障害等によってファイルのデータが破壊された場合、プログラム実行前のIFAS ファイルのバックアップを基に更新ログファイルに記録されたレコード操作を順番に再現し、ファイルを障害発生直前の状態に復旧することが可能です。

利用者はトランザクションプログラム実行前に、更新するIFAS ファイルのバックアップを行っておき、ファイルの復旧が必要となった場合、バックアップファイルをディスク上に復元し、更新ログファイルを指定してロールフォワード機能を実行します。

ロールフォワード機能はユーティリティプログラムとして提供します。

## 3-2 IFAS FILE関数

IFAS FILEは一般ファイルをアクセスするために、以下のようなC言語関数を提供します。なお、IFAS FILE関数を使用する場合は、インクルードファイルとして「**ifas.h**」を指定してください。

- ・索引順編成ファイルアクセス関数
- ・相対編成ファイルアクセス関数
- ・順編成ファイルアクセス関数
- ・IFAS ファイル共通関数
- ・トランザクション関数

### 3-2-1 索引順編成ファイルアクセス関数

IFAS索引順編成ファイルをアクセスするための関数を紹介します。

<code>ifas_create_is</code>	索引順編成ファイルを作成する
<code>ifas_open_is</code>	索引順編成ファイルをオープンする
<code>ifas_close_is</code>	索引順編成ファイルをクローズする
<code>ifas_put_is</code>	索引順編成ファイルにレコードを書き出す
<code>ifas_get_is</code>	索引順編成ファイルからレコードを読み込む
<code>ifas_putx_is</code>	索引順編成ファイルのレコードを書き替える
<code>ifas_delete_is</code>	索引順編成ファイルのレコードを削除する
<code>ifas_position_is</code>	索引順編成ファイルの検索準備／検索を行う
<code>ifas_addindex_is</code>	索引順編成ファイルにサブインデックスを追加する
<code>ifas_delindex_is</code>	索引順編成ファイルのサブインデックスを削除する
<code>ifas_skip_is</code>	索引順編成ファイルのカレントレコードを移動する
<code>ifas_erase_is</code>	索引順編成ファイルを削除する
<code>ifas_rename_is</code>	索引順編成ファイルの名前を変更する
<code>ifas_getinfo_is</code>	索引順編成ファイルの情報を取得する
<code>ifas_remakeindex_is</code>	索引順編成ファイルのインデックスを再編成する
<code>ifas_rebuild_idxfile_is</code>	インデックスファイルを再作成する

## CREATE\_IS関数

IFAS索引順編成ファイル(インデックスファイル、データファイル)を作成します。

### 書式

```
DWORD    ifas_create_is(
          CHAR          *filename_i,
          CHAR          *filename_d,
          struct ifkeydesc *keydesc,
          DWORD          reclen
        )
```

### パラメータ

filename\_i      インデックスファイル名 文字列へのポインタ  
 filename\_d      データファイル名 文字列へのポインタ  
 keydesc          プライマリキーにおけるキー記述構造体のポインタ

<キー記述構造体: struct ifkeydesc>

データ型	構造体メンバ名	内 容	既定値
DWORD	ifk_number	キーパート数 (0=インデックスを作成しない)	1
DWORD	ifk_dupflag	重複キー許可 (0=重複不許可: 0以外はエラー)	0
struct ifkeypart	k_prt[n]	キーパート構造体 (キーパートは最大16個まで指定可能)	

<キーパート構造体: struct ifkeypart>

データ型	構造体メンバ名	内 容	既定値
DWORD	ifk_key_offset	レコード内キーオフセット	0
DWORD	ifk_key_size	キーサイズ	0
DWORD	ifk_type	キータイプ:	IFCHAR
		IFCHAR      // 文字	
		IFBIN4      // DWORD (4バイト)	
		IFBIN2      // WORD (2バイト)	
		IFSBIN4     // LONG (4バイト)	
		IFSBIN2     // INT (2バイト)	
		IFFLOAT     // FLOAT (4バイト)	
		IFDOUBLE    // DOUBLE (8バイト)	

reclen            データレコードのバイトサイズ

### 戻り値

ステータスコード

### 規則

- (1) 既に存在しているファイルのファイル名を指定するとエラーとなります。
- (2) 本関数で作成したファイルはデータ0件のファイルとして入力／更新モードでオープンしてアクセスすることができます。



- (3) keydescにNULLを指定した場合はプライマリインデックスのないファイルを作成します。
- (4) reclenに0を指定した場合は規定値256バイトを設定します。

## 主なステータスコード

---

IFNORMAL = 0 (0x0000)

意味 : 処理は正常に終了した

IFALDYFN = 101 (0x0065)

意味 : 指定したファイル名のファイルは既に存在する

IFFLBUSY = 205 (0x00cd)

意味 : このファイルは現在他の利用者が使用中である

IFRSZERR = 206 (0x00ce)

意味 : 指定したレコードサイズが不正である

IFKDSCER = 802 (0x0322)

意味 : キー記述構造体が不正である

IFPARMER = 804 (0x0324)

意味 : パラメータの指定に誤りがある

IFFNAMER = 805 (0x0325)

意味 : ファイル名の指定が不正である

IFDRDBER = 1101 (0x044d)

意味 : ファイル識別子管理ファイルのアクセスエラーが発生した

## OPEN\_IS関数

IFAS索引順編成ファイルをオープンします。

### 書式

```
DWORD ifas_open_is(
    CHAR *filename_i,
    CHAR *filename_d,
    struct file_option *option_block,
    struct ifkeydesc *keydesc,
    DWORD reclen,
    DWORD mode,
    HANDLE *isfd
)
```

### パラメータ

**filename\_i** インデックスファイル名文字列へのポインタ  
**filename\_d** データファイル名文字列へのポインタ  
**option\_block** IFAS FILEオプションブロック構造体のポインタ  
 IFAS FILEが処理を行う上での以下の既定値を変更したい場合に指定します。  
 "0"を指定した項目は既定値が適用されます。

〈IFAS FILEオプションブロック構造体: **struct file\_option**〉

データ型	構造体メンバ名	内 容	既定値
DWORD	f_ind_blksize	インデックスブロックサイズ (出力モードでオープンする場合のみ有効)	1024
DWORD	f_ind_nbuf	インデックスバッファ数	10
DWORD	f_ind_pctf	インデックスブロック占有率 (出力モードでオープンする場合のみ有効)	100
DWORD	f_dat_nbuf	データバッファ数	10

**keydesc** プライマリキーにおけるキー記述構造体のポインタ  
 出力モードの場合のみ有効です。

〈キー記述構造体: **struct ifkeydesc**〉

データ型	構造体メンバ名	内 容	既定値
DWORD	ifk_number	キーパート数 (0=インデックスを作成しない)	1
DWORD	ifk_dupflag	重複キー許可 (0=重複不許可: 0以外はエラー)	0
struct ifkeypart	k_prt[n]	キーパート構造体 (キーパートは最大16個まで指定可能)	

〈キーパート構造体: **struct ifkeypart**〉

データ型	構造体メンバ名	内 容	既定値
DWORD	ifk_key_offset	レコード内キーオフセット	0
DWORD	ifk_key_size	キーサイズ	0
DWORD	ifk_type	キータイプ: IFCHAR // 文字 IFBIN4 // DWORD(4バイト)	IFCHAR

IFBIN2        //WORD (2バイト)  
 IFSBIN4       //LONG (4バイト)  
 IFSBIN2       //INT (2バイト)  
 IFFLOAT       //FLOAT (4バイト)  
 IFDOUBLE      //DOUBLE (8バイト)

## reclen mode

データレコードのバイトサイズ

オープンモード

アクセスモード、ロックモード、トランザクションオプションを以下の例のように組み合わせで指定します。

例 :        IFUPDATE + IFLOCKR + IFROLLBACK

          IFINPUT | IFLOCKS

•アクセスモード(必須)

IFINPUT        入力モードでオープンします。  
 IFOUTPUT       出力モードでオープンします。  
 IFUPDATE       更新モードでオープンします。

•ロックモード(必須)

IFLOCKE        ファイル単位で排他ロックします。  
 IFLOCKS        ファイル単位で共有ロックします。  
 IFLOCKR        レコード単位でロックします。  
 IFFREE         ロック制御を行いません。

•トランザクションオプション

IFROLLBACK     ロールバック時ファイルの更新を取り消します。  
 IFNOLOG        更新ログを収集しません。

## isfd

ファイル記述子

OPENが成功した場合、IFAS FILEはファイル記述子を返却します。この記述子は、このファイルをオープンしたスレッド内でのみ有効となります。

## 戻り値

ステータスコード

## 規則

- (1) 出力モードを指定した場合のみ、指定したファイル名のファイルが存在しなければ自動的にファイルを新規作成します。入力モード、更新モードの場合は存在しないファイルのファイル名を指定するとエラーになります。
- (2) 出力モードを指定した場合はファイルの内容をすべて初期化します。
- (3) 入力モード、更新モードを指定した場合、インデックスファイルとデータファイルはファイルを初期化した時の組み合わせでなければなりません。
- (4) アクセスモードとロックモードでオープン可能な組み合わせは以下のとおりです。

(○:可、×:不可)

	IFINPUT	IFOUTPUT	IFUPDATE
IFLOCKE	○	○	○
IFLOCKS	○	×	×
IFLOCKR	○	×	○
IFFREE	○	×	×

- (5) 他プロセス(スレッド)間で同時にオープン可能なロックモードの組み合わせは以下のとおりです。

(○:可、×:不可、△:IFLOCKR側が入力モードの場合のみ可)

	IFLOCKE	IFLOCKS	IFLOCKR	IFFREE
IFLOCKE	×	×	×	○

IFLOCKS	×	○	△	○
IFLOCKR	×	△	○	○
IFFREE	○	○	○	○

- (6) ロックモードでIFFREEを指定した場合は他のプロセスのロック状態に関わらずそのファイルのデータを入力できるので、入力データに矛盾が生じる場合もあり得ます。
- (7) IFROLLBACKを指定してオープンしたファイルに対してアクセス関数を実行する場合は、その前にTXSTART関数が実行されていなければなりません。
- (8) キー記述構造体はアクセスモードがIFOUTPUTの場合のみ有効です。キーパート数が0だった場合、または構造体ポインタがNULLの場合は、プライマリインデックスを作成しません。以降このファイルにADDINDEX関数でプライマリインデックスを作成することはできません。
- (9) 出力モードでレコードサイズに0を指定した場合は既定値の256バイトをレコードサイズとします。
- (10) キー記述構造体において重複キー許可指定をすることはできません。
- (11) IFAS FILEオプションブロックのインデックスブロックサイズとインデックスブロック占有率は出力モードでオープンした場合のみ有効となります。
- (12) IFNOLOGを指定した場合はLOGOPEN関数によって更新ログファイルがオープンされていてもログ収集を行いません。
- (13) トランザクションオプションに何も指定していない場合は、更新ログファイルがオープンされていれば更新ログを収集しますが、ロールバックは行いません。

## 主なステータスコード

IFNORMAL = 0 (0x0000)

意味 : 処理は正常に終了した

IFNMLNLG = 3 (0x0003)

意味 : 処理は正常終了したが更新ログ出力は行わない

IFNMLNLK = 4 (0x0004)

意味 : 処理は正常に終了したがロック制御は行わない

IFNFNDFN = 201 (0x00c9)

意味 : 指定したファイル名のファイルが存在しない

IFNOIFFL = 203 (0x00cb)

意味 : 指定したファイル名のファイルはIFAS ファイルではない

IFNOINIT = 204 (0x00cc)

意味 : 指定したファイル名のファイルは初期化されていない

IFFLBUSY = 205 (0x00cd)

意味 : このファイルは現在他の利用者が使用中である

IFRSZERR = 206 (0x00ce)

意味 : 指定したレコードサイズが不正である

IFOPBLER = 207 (0x00cf)

意味 : ファイルオプションブロックが不正である

IFUMCHFL = 209 (0x00d1)

意味 : インデックスとデータのファイルの組み合わせが不正である

IFOPCTER = 210 (0x00d2)

意味 : 同時にオープン可能な最大数を超過してオープン要求を行った

IFNOWRIT = 213 (0x00d5)

意味 : 書き込み不可属性のファイルである

IFFORGER = 214 (0x00d6)

意味 : データファイルとインデックスファイルを逆に指定している

IFUSERND = 215 (0x00d7)

意味 : 順／相対編成ファイルとして更新されたファイルである

IFOPMDER = 801 (0x0321)

意味 : アクセスモードとロックモードの組み合わせが不正である

IFKDSCER = 802 (0x0322)

意味 : キー記述構造体が不正である

IFPARMER = 804 (0x0324)

意味 : パラメータの指定に誤りがある

IFFNAMER = 805 (0x0325)

意味 : ファイル名の指定が不正である

IFABTLCK = 1003 (0x03eb)

意味 : ファイルはリカバリ処理に失敗して壊れている可能性がある

IFMMEXER = 1004 (0x03ec)

意味 : メモリの確保に失敗した

IFNOCLSF = 1009 (0x03f1)

意味 : シャットダウン時に終了処理が行われなかったファイルである

IFVERERR = 1012 (0x03f4)

意味 : 旧バージョンのIFAS FILEとの競合が発生した

IFDRDBER = 1101 (0x044d)

意味 : ファイル識別子管理ファイルのアクセスエラーが発生した

IFINTPRC = 1201 (0x0461)

意味 : IFAS FILE初期化プロセスが動作していない

IFSGIFER = 1204 (0x0464)

意味 : IFAS FILEのSG情報が不正である

IFPRTCER = 1205 (0x0465)

意味 : IFAS FILEを使用できない環境である

## CLOSE\_IS関数

IFAS索引順編成ファイルをクローズします。

### 書式

---

```
DWORD    ifas_close_is(  
HANDLE    isfd  
)
```

### パラメータ

---

isfd                    OPEN関数で返却するファイル記述子

### 戻り値

---

ステータスコード

### 規則

---

- (1) オープンしていないファイルに対して本関数を実行するとエラーとなります。
- (2) 本関数によってファイルロック、レコードロックはすべて解除されます。
- (3) IFROLLBACK指定でオープンしたファイルはトランザクション処理中に本関数を実行するとエラーとなります。

### 主なステータスコード

---

IFNORMAL = 0    (0x0000)

意 味    : 処理は正常に終了した

IFBADHND = 211 (0x00d3)

意 味    : 指定したファイルはオープンされていない

IFNENDTX = 703 (0x02bf)

意 味    : トランザクション処理が終了していない

IFDRDBER = 1101 (0x044d)

意 味    : ファイル識別子管理ファイルのアクセスエラーが発生した

## PUT\_IS関数

IFAS索引順編成ファイルにデータレコードを1件書き出します。

### 書式

```
DWORD    ifas_put_is(
HANDLE    isfd
LPVOID    record,
DWORD    curr
)
```

### パラメータ

isfd	OPEN関数で返却するファイル記述子
record	書き出すデータレコード領域のポインタ
curr	以下のいずれかを選択します。 0: カレントレコードを変更しません 1: 書き出したレコードをカレントレコードに変更します

### 戻り値

ステータスコード

### 規則

- (1) オープンしていないファイルに対して本関数を実行するとエラーとなります。
- (2) 入力モードでオープンされたファイルに対して本関数を実行するとエラーとなります。
- (3) 本関数によって書き出したレコードのレコード番号を知るためにはGETRECNO関数を実行します。
- (4) 書き出し要求するレコードにおいて、プライマリインデックス、または、重複キー許可指定のないサブインデックスに同じ値のキーが既に存在していた場合は、ステータス”IFDUPKEY”を返却し、本関数は失敗します。
- (5) 書き出し要求するレコードにおいて、重複キー許可指定のあるサブインデックスに同じ値のキーが既に存在していた場合、ステータス”IFNMDPKY”を返却し本関数は成功します。

### 主なステータスコード

IFNORMAL = 0 (0x0000)  
意味 : 処理は正常に終了した

IFNMDPKY = 2 (0x0002)  
意味 : 重複キー許可指定のサブインデックスに重複キーを検出した

IFBADHND = 211 (0x00d3)  
意味 : 指定したファイルはオープンされていない

IFDEDLOK = 309 (0x0135)  
意味 : デッドロックが発生した

IFLKLT0V = 314 (0x013a)  
意味 : 1スレッドあたりの最大ロック制限値を超えた

IFTXRBFL = 401 (0x0191)  
意味 : TXSTART関数が実行されていない

IFDUPKEY = 402 (0x0192)

意味 : 重複キー許可指定のないインデックスで重複キーを検出した  
IFOPMDER = 801 (0x0321)

意味 : オープンアクセスモードと機能の組み合わせが不正である  
IFPARMER = 804 (0x0324)

意味 : パラメータの指定に誤りがある  
IFERRFIL = 1005 (0x03ed)

意味 : エラーが解決していないファイルにアクセス要求を行った  
IFFLEXER = 1007 (0x03ef)

意味 : ファイルサイズの拡張ができない  
IFIXLVER = 1008 (0x03f0)

意味 : インデックスツリーレベルがオーバーフローした



## GET\_IS関数

IFAS索引順編成ファイルからデータレコードを1件読み込みます。

### 書式

```
DWORD    ifas_get_is(
HANDLE    isfd,
LPVOID    record,
DWORD    access,
DWORD    pmode,
DWORD    recnum
)
```

### パラメータ

<b>isfd</b>	OPEN関数で返却するファイル記述子
<b>record</b>	読み込むデータレコード領域のポインタ キー検索によって読み込む場合は本関数を実行する前に、利用者がキーの位置にキーイメージをセットしておきます。
<b>access</b>	アクセス種別 IFKEY キー値に基づいたアクセスを行います。 IFRECNO レコード番号に基づいたアクセスを行います。 IFCURRECNO カレントレコードに基づいたアクセスを行います。
<b>pmode</b>	位置付けモード accessでIFKEYを選択した場合は、以下のパラメータの中から選択指定します。 IFFIRST インデックスで最小キーのレコードを読み込みます。 IFLAST インデックスで最大キーのレコードを読み込みます。 IFEQUAL 指定キーと一致するキー値のレコードを読み込みます。 IFGREAT 指定キーより大きいキー値のレコードを読み込みます。 IFGTEQ 指定キー以上のキー値のレコードを読み込みます。 accessでIFRECNOを選択した場合は、以下のパラメータの中から選択指定します。 IFFIRST 有効最小レコード番号のレコードを読み込みます。 IFLAST 有効最大レコード番号のレコードを読み込みます。 IFEQUAL 指定レコード番号と一致するレコードを読み込みます。 IFGREAT 指定レコード番号より大きいレコードを読み込みます。 IFGTEQ 指定レコード番号以上のレコードを読み込みます。 IFLITTLE 指定レコード番号より小さいレコードを読み込みます。 IFLTEQ 指定レコード番号以下のレコードを読み込みます。 accessでIFCURRECNOを選択した場合は、以下のパラメータの中から選択指定します。 IFEQUAL カレントレコードを読み込みます。 IFNEXT キー順でカレントの次のレコードを読み込みます。 OPEN後最初のGET関数の場合にかぎり、キー順で先頭のレコードを読み込みます。 POSITION関数でレコードの位置付けを行った後の最初のGET関数の場合にはカレントレコードを読み込みます。 IFPREV キー順でカレントの前のレコードを読み込みます。

オープン後最初のGET関数の場合はステータスとしてIFLGCEOFを返却します。POSITION関数でレコードの位置付けを行った後の最初のGET関数の場合にはカレントレコードを読み込みます。

**recnum**           レコード番号  
レコード番号指定で読み込む場合、レコード番号を指定します。

## 戻り値

---

ステータスコード

## 規則

---

- (1) オープンしていないファイルに対して本関数を実行するとエラーとなります。
- (2) 出力モードでオープンされたファイルに対して本関数を実行するとエラーとなります。
- (3) accessでIFKEYを指定してキーアクセスを行う場合は、あらかじめPOSITION関数によって、検索するインデックスを選択しておかなければなりません。
- (4) 本関数によって読み込んだレコードのレコード番号を知るためにはGETRECNO関数を実行します。
- (5) 重複キー許可指定のあるインデックスでキー検索を行った時、実際に複数の同一キーが存在した場合は、対応するレコード番号が最も小さいレコードを読み込み、まだ次に同一キーのレコードがあることを示すステータス"IFNMDPKY"を返却します。次の同一キーのレコードはIFNEXT指定のGET関数で読み込むことが可能で、その場合も次のレコードが同一キーであればステータス"IFNMDPKY"を返却します。
- (6) 末尾レコード番号または最大キー値のレコードがカレントの場合、IFNEXT指定で本関数を実行するとステータス"IFLGCEOF"を返却します。
- (7) 先頭レコード番号または最小キー値のレコードがカレントの場合、IFPREV指定で本関数を実行するとステータス"IFLGCEOF"を返却します。
- (8) レコード番号またはキーで検索したレコードがファイル内に存在しない場合、ステータス"IFNFNDRC"を返却します。

## 主なステータスコード

---

IFNORMAL = 0   (0x0000)

意味 : 処理は正常に終了した

IFNMDPKY = 2   (0x0002)

意味 : 重複キー許可指定のサブインデックスに重複キーを検出した

IFBADHND = 211 (0x00d3)

意味 : 指定したファイルはオープンされていない

IFNCURIX = 303 (0x012f)

意味 : インデックスが選択されていない

IFNCURRC = 304 (0x0130)

意味 : カレントレコードが設定されていない

IFLGCEOF = 305 (0x0131)

意味 : ファイルの論理的終端を検出した

IFNFNDRC = 306 (0x0132)

意味 : 指定したキーまたはレコード番号のレコードが存在しない

IFDEDLOK = 309 (0x0135)

意味 : デッドロックが発生した

IFLKLTOV = 314 (0x013a)

意味 : 1スレッドあたりの最大ロック制限値を超えた

IFOPMDER = 801 (0x0321)

意味 : オープンアクセスモードと機能の組み合わせが不正である

IFPARMER = 804 (0x0324)

意味 : パラメータの指定に誤りがある  
IFERRFIL = 1005 (0x03ed)  
意味 : エラーが解決していないファイルにアクセス要求を行った

## PUTX\_IS関数

IFAS索引順編成ファイルのデータレコードを1件書き換えます。

### 書式

```
DWORD ifas_putx_is(
HANDLE isfd,
LPVOID record,
DWORD access,
DWORD recnum
)
```

### パラメータ

<b>isfd</b>	OPEN関数で返却するファイル記述子
<b>record</b>	書き換えるデータレコード領域のポインタ
<b>access</b>	アクセス種別
	IFKEY キー値に基づいたアクセスを行います。
	IFRECNO レコード番号に基づいたアクセスを行います。
	IFCURR カレントレコードをアクセスします。
<b>recnum</b>	レコード番号
	レコード番号指定で書き換える場合レコード番号を指定します。

### 戻り値

ステータスコード

### 規則

- (1) オープンしていないファイルに対して本関数を実行するとエラーとなります。
- (2) 入力モード、出力モードでオープンされたファイルに対して本関数を実行するとエラーとなります。
- (3) accessでIFKEYを指定してキーアクセスを行う場合は、あらかじめPOSITION関数によって、プライマリインデックスを選択しておかなければなりません。
- (4) 本関数によって書き換えたレコードのレコード番号を知るためにはGETRECNO関数を実行します。
- (5) accessでIFRECNOまたはIFCURRを選択した場合、プライマリキーを書き換えるような操作を実行するとエラーとなります。
- (6) 書き換え要求するレコードにおいて、重複キー許可指定のあるサブインデックスに同じ値のキーが既に存在していた場合、ステータス”IFNMDPKY”を返却し本関数の実行は成功します。

### 主なステータスコード

IFNORMAL = 0 (0x0000)  
意味 : 処理は正常に終了した  
IFBADHND = 211 (0x00d3)  
意味 : 指定したファイルはオープンされていない  
IFNCURIX = 303 (0x012f)  
意味 : インデックスが選択されていない  
IFNCURRC = 304 (0x0130)

意味 : カレントレコードが設定されていない  
IFNFNDRC = 306 (0x0132)  
意味 : 指定したキーまたはレコード番号のレコードが存在しない  
IFNRWPMK = 308 (0x0134)  
意味 : プライマリキーを書き換えようとした  
IFDEDLOK = 309 (0x0135)  
意味 : デッドロックが発生した  
IFNPRMIX = 310 (0x0136)  
意味 : 選択されているインデックスはプライマリインデックスではない  
IFLKLT0V = 314 (0x013a)  
意味 : 1スレッドあたりの最大ロック制限値を超えた  
IFTXRBFL = 401 (0x0191)  
意味 : TXSTART関数が実行されていない  
IFDUPKEY = 402 (0x0192)  
意味 : 重複キー許可指定のないインデックスで重複キーを検出した  
IFOPMDER = 801 (0x0321)  
意味 : オープンアクセスモードと機能の組み合わせが不正である  
IFPARMER = 804 (0x0324)  
意味 : パラメータの指定に誤りがある  
IFERRFIL = 1005 (0x03ed)  
意味 : エラーが解決していないファイルにアクセス要求を行った

## DELETE\_IS関数

IFAS索引順編成ファイルのデータレコードを1件削除します。

### 書式

```
DWORD    ifas_delete_is(
HANDLE    isfd,
LPVOID    record,
DWORD     access,
DWORD     recnum
)
```

### パラメータ

<b>isfd</b>	OPEN関数で返却するファイル記述子
<b>record</b>	削除するデータレコード領域のポインタ キー検索によって削除する場合は本関数を実行する前に、キーイメージをセットしておきます。
<b>access</b>	アクセス種別 IFKEY                      キー値に基づいたアクセスを行います。 IFRECNO                    レコード番号に基づいたアクセスを行います。 IFCURR                    カレントレコードをアクセスします。
<b>recnum</b>	レコード番号 レコード番号指定で削除する場合レコード番号を指定します。

### 戻り値

ステータスコード

### 規則

- (1) オープンしていないファイルに対して本関数を実行するとエラーとなります。
- (2) 入力モード、出力モードでオープンされたファイルに対して本関数を実行するとエラーとなります。
- (3) accessでIFKEYを指定してキーアクセスを行う場合は、あらかじめPOSITION関数によって、プライマリインデックスを選択しておかなければなりません。
- (4) 本関数によって削除したレコードのレコード番号を知るためにはGETRECNO関数を実行します。
- (5) accessでIFKEYを選択した場合は、カレントインデックスにプライマリインデックス以外を選択してはなりません。

### 主なステータスコード

IFNORMAL = 0    (0x0000)  
 意味 : 処理は正常に終了した  
 IFBADHND = 211 (0x00d3)  
 意味 : 指定したファイルはオープンされていない  
 IFNCURIX = 303 (0x012f)  
 意味 : インデックスが選択されていない  
 IFNCURRC = 304 (0x0130)

意味 : カレントレコードが設定されていない  
IFNFNDRC = 306 (0x0132)  
意味 : 指定したキーまたはレコード番号のレコードが存在しない  
IFDEDLOK = 309 (0x0135)  
意味 : デッドロックが発生した  
IFNPRMIX = 310 (0x0136)  
意味 : 選択されているインデックスはプライマリインデックスではない  
IFLKLT0V = 314 (0x013a)  
意味 : 1スレッドあたりの最大ロック制限値を超えた  
IFTXRBFL = 401 (0x0191)  
意味 : TXSTART関数が実行されていない  
IFOPMDER = 801 (0x0321)  
意味 : オープンアクセスモードと機能の組み合わせが不正である  
IFPARMER = 804 (0x0324)  
意味 : パラメータの指定に誤りがある  
IFERRFIL = 1005 (0x03ed)  
意味 : エラーが解決していないファイルにアクセス要求を行った

## POSITION\_IS関数

IFAS索引順編成ファイルに対し、キー検索用のインデックスを選択します。  
または、データレコードの位置付けを行い、カレントレコードを変更します。

### 書式

```
DWORD    ifas_position_is(
HANDLE    isfd,
struct ifkeydesc *keydesc,
DWORD    length,
LPVOID    record,
DWORD    access,
DWORD    pmode,
DWORD    recnum
)
```

### パラメータ

**isfd** OPEN関数で返却するファイル記述子  
**keydesc** 選択するインデックスのキー記述構造体のポインタ

〈キー記述構造体: **struct ifkeydesc**〉

データ型	構造体メンバ名	内 容
DWORD	ifk_number	キーパート数
DWORD	ifk_dupflag	重複キー許可 (0 = 重複不許可: 1 = 重複許可)
struct ifkeypart	k_prt[n]	キーパート構造体 (キーパートは最大16個まで指定可能)

〈キーパート構造体: **struct ifkeypart**〉

データ型	構造体メンバ名	内 容	既定値
DWORD	ifk_key_offset	レコード内キーオフセット	0
DWORD	ifk_key_size	キーサイズ	0
DWORD	ifk_type	キータイプ:	IFCHAR
		IFCHAR	// 文字
		IFBIN4	// DWORD (4バイト)
		IFBIN2	// WORD (2バイト)
		IFSBIN4	// LONG (4バイト)
		IFSBIN2	// INT (2バイト)
		IFFLOAT	// FLOAT (4バイト)
		IFDOUBLE	// DOUBLE (8バイト)

**length** 検索するキーフィールド長  
 キーサイズ全体をキーフィールドとする場合は、0またはキーサイズを指定します。キーサイズより大きい値を指定してもキーサイズ全体とみなします。

**record** 検索するキーイメージを指定するデータレコード領域のポインタ  
 キーイメージはレコード内のキーオフセット位置に設定します。

**access** アクセス種別  
 IFKEY キー値に基づいたアクセスを行います。



	IFRECNO	レコード番号に基づいたアクセスを行います。
	IFCURR	カレントレコードをアクセスします。
	IFNONE	レコードの位置付けを行いません。
<b>pmode</b>	位置付けモード	
	accessでIFKEYを選択した場合は、以下のパラメータの中から選択指定します。	
	IFFIRST	インデックスで最小キーのレコードに位置付けます。
	IFLAST	インデックスで最大キーのレコードに位置付けます。
	IFEQUAL	指定キーと一致するキー値のレコードに位置付けます。
	IFGREAT	指定キーより大きいキー値のレコードに位置付けます。
	IFGTEQ	指定キー以上のキー値のレコードに位置付けます。
	accessでIFRECNOを選択した場合は、以下のパラメータの中から選択指定します。	
	IFFIRST	有効最小レコード番号のレコードに位置付けます。
	IFLAST	有効最大レコード番号のレコードに位置付けます。
	IFEQUAL	指定レコード番号と一致するレコードに位置付けます。
	IFGREAT	指定レコード番号より大きいレコードに位置付けます。
	IFGTEQ	指定レコード番号以上のレコードに位置付けます。
	IFLITTLE	指定レコード番号より小さいレコードに位置付けます。
	IFLTEQ	指定レコード番号以下のレコードに位置付けます。
	accessでIFCURRを選択した場合は、以下のパラメータの中から選択指定します。	
	IFNEXT	カレントの次のレコードに位置付けます。
	IFPREV	カレントの前のレコードに位置付けます。
	accessでIFNONEを選択した場合は、本パラメータは無効です。	
<b>recnum</b>	レコード番号	
	レコード番号指定で位置付ける場合、レコード番号を指定します。	

## 戻り値

ステータスコード

## 規則

- (1) オープンしていないファイルに対して本関数を実行するとエラーとなります。
- (2) 出力モードでオープンされたファイルに対して本関数を実行するとエラーとなります。
- (3) 本関数によって位置付けたレコードのレコード番号を知るためにはGETRECNO関数を実行します。
- (4) レコード番号またはキーで検索したレコードがファイル内に存在しない場合はステータス”IFNFNDRC”を返却します。
- (5) インデックス選択とレコードの位置付けを同時に要求した場合に、レコードの位置付けに失敗してもインデックスの選択は成功しています。
- (6) インデックス選択のみを要求した場合には直前のレコード位置付けは無効となります。
- (7) レコードの位置付けを行った場合、本関数実行後最初のGET関数でのみIFCURR指定をすれば IFNEXT、IFEQUAL、IFPREVのどれを指定しても、位置付けたレコードを読み込むことができます。

## 主なステータスコード

IFNORMAL = 0 (0x0000)

意味 : 処理は正常に終了した

IFBADHND = 211 (0x00d3)

意味 : 指定したファイルはオープンされていない

IFNCURIX = 303 (0x012f)

意味 : インデックスが選択されていない  
IFNCURRC = 304 (0x0130)  
意味 : カレントレコードが設定されていない  
IFLGEOF = 305 (0x0131)  
意味 : ファイルの論理的終端を検出した  
IFNFNDRC = 306 (0x0132)  
意味 : 指定したキーまたはレコード番号のレコードが存在しない  
IFNFNDKD = 307 (0x0133)  
意味 : 指定したキー記述に相当するインデックスが見つからない  
IFDEDLOK = 309 (0x0135)  
意味 : デッドロックが発生した  
IFLKLTOV = 314 (0x013a)  
意味 : 1スレッドあたりの最大ロック制限値を超えた  
IFOPMDER = 801 (0x0321)  
意味 : オープンアクセスモードと機能の組み合わせが不正である  
IFKDSCER = 802 (0x0322)  
意味 : キー記述構造体が不正である  
IFPARMER = 804 (0x0324)  
意味 : パラメータの指定に誤りがある  
IFERRFIL = 1005 (0x03ed)  
意味 : エラーが解決していないファイルにアクセス要求を行った

## ADDINDEX\_IS関数

IFAS索引順編成ファイルにサブインデックスを追加します。

### 書式

```
DWORD    ifas_addindex_is(
HANDLE    isfd,
struct ifkeydesc *keydesc
)
```

### パラメータ

**isfd** OPEN関数で返却するファイル記述子  
**keydesc** 追加するサブインデックスのキー記述構造体のポインタ

〈キー記述構造体: **struct ifkeydesc**〉

データ型	構造体メンバ名	内 容	既定値
DWORD	ifk_number	キーパート数 (0=インデックスを作成しない)	1
DWORD	ifk_dupflag	重複キー許可 (0=重複不許可、1=重複許可)	0
struct ifkeypart	k_prt[n]	キーパート構造体 (キーパートは最大16個まで指定可能)	

〈キーパート構造体: **struct ifkeypart**〉

データ型	構造体メンバ名	内 容	既定値
DWORD	ifk_key_offset	レコード内キーオフセット	0
DWORD	ifk_key_size	キーサイズ	0
DWORD	ifk_type	キータイプ:	IFCHAR
		IFCHAR //文字	
		IFBIN4 //DWORD(4バイト)	
		IFBIN2 //WORD(2バイト)	
		IFSBIN4 //LONG(4バイト)	
		IFSBIN2 //INT(2バイト)	
		IFFLOAT //FLOAT(4バイト)	
		IFDOUBLE //DOUBLE(8バイト)	

### 戻り値

ステータスコード

### 規則

- (1) オープンしていないファイルに対して本関数を実行するとエラーとなります。
- (2) 本関数を実行する場合はファイル排他ロックモードでオープンしていなければなりません。

### 主なステータスコード

IFNORMAL = 0 (0x0000)  
 意 味 : 処理は正常に終了した  
 IFBADHND = 211 (0x00d3)

意味 : 指定したファイルはオープンされていない  
IFALDYKD = 311 (0x0137)  
意味 : 指定したキー記述と同一のインデックスが既に存在する  
IFAICTER = 313 (0x0139)  
意味 : 最大インデックス数を越えたインデックス追加を行った  
IFOPMDER = 801 (0x0321)  
意味 : オープンアクセスモードと機能の組み合わせが不正である  
IFKDSCER = 802 (0x0322)  
意味 : キー記述構造体が不正である  
IFPARMER = 804 (0x0324)  
意味 : パラメータの指定に誤りがある  
IFERRFIL = 1005 (0x03ed)  
意味 : エラーが解決していないファイルにアクセス要求を行った

## DELINDEX\_IS関数

IFAS索引順編成ファイルからサブインデックスを削除します。

### 書式

```
DWORD    ifas_delindex_is(
HANDLE    isfd,
struct ifkeydesc *keydesc
)
```

### パラメータ

**isfd**            OPEN関数で返却するファイル記述子  
**keydesc**        削除するインデックスのキー記述構造体のポインタ

〈キー記述構造体: **struct ifkeydesc**〉

データ型	構造体メンバ名	内 容
DWORD	ifk_number	キーパート数
DWORD	ifk_dupflag	重複キー許可 (0=重複不許可: 1=重複許可)
struct ifkeypart	k_prt[n]	キーパート構造体 (キーパートは最大16個まで指定可能)

〈キーパート構造体: **struct ifkeypart**〉

データ型	構造体メンバ名	内 容
DWORD	ifk_key_offset	レコード内キーオフセット
DWORD	ifk_key_size	キーサイズ
DWORD	ifk_type	キータイプ: IFCHAR        // 文字 IFBIN4        // DWORD (4バイト) IFBIN2        // WORD (2バイト) IFSBIN4       // LONG (4バイト) IFSBIN2       // INT (2バイト) IFFLOAT       // FLOAT (4バイト) IFDOUBLE      // DOUBLE (8バイト)

### 戻り値

ステータスコード

### 規則

- (1) オープンしていないファイルに対して本関数を実行するとエラーとなります。
- (2) 本関数を実行する場合はファイル排他ロックモードでオープンしていなければなりません。

### 主なステータスコード

IFNORMAL = 0    (0x0000)

意 味 : 処理は正常に終了した

IFBADHND = 211 (0x00d3)

意 味 : 指定したファイルはオープンされていない

IFNFNDKD = 307 (0x0133)

意味 : 指定したキー記述に相当するインデックスが見つからない

IFDLPMKD = 312 (0x0138)

意味 : プライマリインデックスを削除しようとした

IFOPMDER = 801 (0x0321)

意味 : オープンアクセスモードと機能の組み合わせが不正である

IFKDSCER = 802 (0x0322)

意味 : キー記述構造体が不正である

IFPARMER = 804 (0x0324)

意味 : パラメータの指定に誤りがある

IFERRFIL = 1005 (0x03ed)

意味 : エラーが解決していないファイルにアクセス要求を行った

## SKIP\_IS関数

IFAS索引順編成ファイルに対し、指定したレコード数分、前後方向にカレントレコードの位置を移動します。

### 書式

```
DWORD ifas_skip_is(
HANDLE isfd,
DWORD access,
DWORD pmode,
DWORD recnt
)
```

### パラメータ

isfd	OPEN関数で返却するファイル記述子
access	アクセス種別
	IFKEY インデックスキー順の位置付けを行います。
	IFRECNO レコード番号順の位置付けを行います。
pmode	位置付けモード
	IFBFILE アクセス種別に従った先頭のレコードに位置付けます。
	IFFWDCNT 指定したレコード数分先のレコードに位置付けます。
	IFRWDCNT 指定したレコード数分後ろのレコードに位置付けます。
recnt	レコード数
	pmodeでIFFWDCNT、IFRWDCNTを指定した場合、カレントレコードからスキップするレコード数を指定します。

### 戻り値

ステータスコード

### 規則

- (1) オープンしていないファイルに対して本関数を実行するとエラーとなります。
- (2) 出力モードでオープンされたファイルに対して本関数を実行するとエラーとなります。
- (3) 本関数によって位置付けたレコードのレコード番号を知るためにはGETRECNO関数を実行します。
- (4) 本処理によってファイルの終端を検出した場合は、ステータス”IFLGCEOF”を返却します。
- (5) 本処理によってカレントレコードを変更した後、GET関数でIFCURR指定のIFNEXTを実行すると位置付けたカレントの次のレコードを、GET関数でIFCURR指定のIFPREVを実行するとカレントの前のレコードを読み込みます。

### 主なステータスコード

IFNORMAL = 0 (0x0000)

意味 : 処理は正常に終了した

IFBADHND = 211 (0x00d3)

意味 : 指定したファイルはオープンされていない

IFNCURRC = 304 (0x0130)

意味 : カレントレコードが設定されていない

IFLGEOF = 305 (0x0131)

意味 : ファイルの論理的終端を検出した

IFDEDL0K = 309 (0x0135)

意味 : デッドロックが発生した

IFLKLT0V = 314 (0x013a)

意味 : 1スレッドあたりの最大ロック制限値を超えた

IFOPMDER = 801 (0x0321)

意味 : オープンアクセスモードと機能の組み合わせが不正である

IFPARMER = 804 (0x0324)

意味 : パラメータの指定に誤りがある

IFERRFIL = 1005 (0x03ed)

意味 : エラーが解決していないファイルにアクセス要求を行った



## ERASE\_IS関数

IFAS索引順編成ファイルを削除します。

### 書式

---

```
DWORD    ifas_erase_is(  
          CHAR          *filename_i,  
          CHAR          *filename_d  
          )
```

### パラメータ

---

filename_i	インデックスファイル名文字列へのポインタ
filename_d	データファイル名文字列へのポインタ

### 戻り値

---

ステータスコード

### 規則

---

- (1) 存在しないファイルのファイル名を指定するとエラーとなります。
- (2) アクセス処理中のファイルをファイル名に指定するとエラーとなります。

### 主なステータスコード

---

IFNORMAL = 0 (0x0000)

意味 : 処理は正常に終了した

IFNFNDFN = 201 (0x00c9)

意味 : 指定したファイル名のファイルが存在しない

IFFLBUSY = 205 (0x00cd)

意味 : このファイルは現在他の利用者が使用中である

IFUMCHFL = 209 (0x00d1)

意味 : インデックスとデータのファイルの組み合わせが不正である

IFFNAMER = 805 (0x0325)

意味 : ファイル名の指定が不正である

## RENAME\_IS関数

IFAS索引順編成ファイルのファイル名を変更します。

### 書式

---

```
DWORD    ifas_rename_is(
            CHAR                *old_filename_i,
            CHAR                *old_filename_d,
            CHAR                *new_filename_i,
            CHAR                *new_filename_d
        )
```

### パラメータ

---

old_filename_i	旧インデックスファイル名文字列へのポインタ
old_filename_d	旧データファイル名文字列へのポインタ
new_filename_i	新インデックスファイル名文字列へのポインタ
new_filename_d	新データファイル名文字列へのポインタ

### 戻り値

---

ステータスコード

### 規則

---

- (1) 存在しないファイルのファイル名を旧ファイル名に指定するとエラーとなります。
- (2) 既に存在するファイルのファイル名を新ファイル名に指定するとエラーとなります。
- (3) アクセス処理中のファイルを旧ファイル名に指定するとエラーとなります。

### 主なステータスコード

---

IFNORMAL = 0 (0x0000)

意味 : 処理は正常に終了した

IFALDYFN = 101 (0x0065)

意味 : 指定したファイル名のファイルは既に存在する

IFNFNDFN = 201 (0x00c9)

意味 : 指定したファイル名のファイルが存在しない

IFFLBUSY = 205 (0x00cd)

意味 : このファイルは現在他の利用者が使用中である

IFUMCHFL = 209 (0x00d1)

意味 : インデックスとデータのファイルの組み合わせが不正である

IFFNAMER = 805 (0x0325)

意味 : ファイル名の指定が不正である

## GETINFO\_IS関数

指定したIFAS索引順編成ファイルおよびインデックスの情報を取得します。

### 書式

```
DWORD    ifas_getinfo_is(
HANDLE    isfd,
LPVOID    buffer,
DWORD    number
)
```

### パラメータ

**isfd** OPEN関数で返却するファイル記述子  
**buffer** 返却する情報格納領域のポインタ  
**number** 0を指定した場合はファイル情報構造体を返却し、0以外の場合はキー記述構造体を返却します。

〈ファイル情報構造体: **struct dectinfo**〉

データ型	構造体メンバ名	内 容
DWORD	de_nkeys	インデックス数
DWORD	de_recsz	データレコード長
DWORD	de_idxsz	インデックスブロック長
DWORD	de_nrecords	データレコード数

**number**に0以外を指定した場合はインデックス番号とみなし、そのインデックス番号のキー記述構造体を返却します。

〈キー記述構造体: **struct ifkeydesc**〉

データ型	構造体メンバ名	内 容
DWORD	ifk_number	キーパート数
DWORD	ifk_dupflag	重複キー許可 (0=重複不許可: 1=重複許可)
struct ifkeypart	k_prt[n]	キーパート構造体(*) *キーパートは最大16個まで指定できます

〈キーパート構造体: **struct ifkeypart**〉

データ型	構造体メンバ名	内 容
DWORD	ifk_key_offset	レコード内キーオフセット
DWORD	ifk_key_size	キーサイズ
DWORD	ifk_type	キータイプ: IFCHAR //文字 IFBIN4 //DWORD(4バイト) IFBIN2 //WORD(2バイト) IFSBIN4 //LONG(4バイト) IFSBIN2 //INT(2バイト) IFFLOAT //FLOAT(4バイト) IFDOUBLE //DOUBLE(8バイト)

**number** 取得情報種別  
0: ファイル全体の情報を取得します

n: 情報を取得したいインデックスの番号を指定します

## 戻り値

---

ステータスコード

## 規則

---

- (1) オープンしていないファイルに対して本関数を実行するとエラーとなります。

## 主なステータスコード

---

IFNORMAL = 0 (0x0000)

意味 : 処理は正常に終了した

IFBADHND = 211 (0x00d3)

意味 : 指定したファイルはオープンされていない

IFNFNDKD = 307 (0x0133)

意味 : 指定したインデックス番号のインデックスが見つからない

IFERRFIL = 1005 (0x03ed)

意味 : エラーが解決していないファイルにアクセス要求を行った

## REMAKEINDEX\_IS関数

IFAS索引順編成ファイルに対し、データと不整合状態にあるインデックスの再編成を行います。

### 書式

```
DWORD    ifas_remakeindex_is(
HANDLE    isfd,
DWORD    number,
DWORD    func
)
```

### パラメータ

isfd	OPEN関数で返却するファイル記述子
number	再編成するインデックスの番号を指定します。 0: すべてのインデックスを再編成対象にします。 n: 再編成したいインデックスの番号を指定します。
func	機能コード
	IFCHK                   インデックスの正当性をチェックして不正なインデックスであれば再編成します。
	IFNCHK                無条件にインデックスの再編成を行います。
	IFGABG                データファイルの削除レコードエリアを無くし、全インデックスを再編成します。本パラメータを指定した場合は、number指定は無効になります。

### 戻り値

ステータスコード

### 規則

- (1) オープンしていないファイルに対して本関数を実行するとエラーとなります。
- (2) 本関数を実行する場合は、ファイル排他ロックモード+更新モードでオープンしていなければなりません。
- (3) 重複なしで設定されたインデックスの再編成中に重複キーレコードを検出した場合、そのキーの作成はスキップして最後まで再編成処理を行います。ただし、このインデックスの不整合状態は解除されません。この場合は、不正なレコードをDELETE関数で削除してから本関数を再実行するか、このインデックスをDELINDEX関数で削除してから「重複あり」の指定に変えてインデックスを再追加(ADDINDEX関数)してください。

### 主なステータスコード

IFNORMAL = 0   (0x0000)

意味 : 処理は正常に終了した

IFWRDPKY = 50   (0x0032)

意味 : 重複キーエラーを検出したが処理は最後まで行った

## REBUILD\_IDXFILE\_IS関数

データファイルとインデックスファイルが不整合となってオープンできなくなった索引順編成ファイルに対し、データファイルを基にインデックスファイルの再構成を行います。

### 書式

---

```
DWORD    ifas_rebuild_idxfile_is(
        CHAR    *filename_i,
        CHAR    *filename_d
    )
```

### パラメータ

---

filename_i	インデックスファイル名文字列へのポインタ
filename_d	データファイル名文字列へのポインタ

### 戻り値

---

ステータスコード

### 規則

---

- (1) 存在しないファイル名を指定するとエラーとなります。
- (2) プライマリインデックスまたは重複なしで設定されたインデックスの再構築中に、実際に重複キーを検出した場合、本機能はエラーとなり実行を中止します。build\_idxfile\_rl関数で新規にインデックスファイルを作成するか、データファイルを相対編成ファイルとしてオープンして重複キーエラーの要因となるデータを修正してから本関数を再実行してください。

### 主なステータスコード

---

IFNORMAL = 0 (0x0000)

意味 : 処理は正常に終了した

IFNFNDFN = 201 (0x00c9)

意味 : 指定したファイル名のファイルが存在しない

IFFLBUSY = 205 (0x00cd)

意味 : このファイルは現在他の利用者が使用中である

IFDUPKEY = 402 (0x0192)

意味 : プライマリインデックスで重複キーを検出した

IFFNAMER = 805 (0x0325)

意味 : ファイル名の指定が不正である

IFDRDBER = 1101 (0x044d)

意味 : ファイル識別子管理ファイルのアクセスエラーが発生した

## 3-2-2 相対編成ファイルアクセス関数

IFAS相対編成ファイルをアクセスするための関数を紹介します。

<code>ifas_create_rl</code>	相対編成ファイルを作成する
<code>ifas_open_rl</code>	相対編成ファイルをオープンする
<code>ifas_close_rl</code>	相対編成ファイルをクローズする
<code>ifas_put_rl</code>	相対編成ファイルにレコードを書き出す
<code>ifas_get_rl</code>	相対編成ファイルからレコードを読み込む
<code>ifas_putx_rl</code>	相対編成ファイルのレコードを書き替える
<code>ifas_delete_rl</code>	相対編成ファイルのレコードを削除する
<code>ifas_position_rl</code>	相対編成ファイルの検索を行う
<code>ifas_skip_rl</code>	相対編成ファイルのカレントレコードを移動する
<code>ifas_erase_rl</code>	相対編成ファイルを削除する
<code>ifas_rename_rl</code>	相対編成ファイルの名前を変更する
<code>ifas_getinfo_rl</code>	相対編成ファイルの情報を取得する
<code>ifas_build_idxfile_rl</code>	相対編成ファイルにインデックスファイルを作成する

## CREATE\_RL関数

IFAS相対編成ファイルを作成します。

### 書式

---

```
DWORD ifas_create_rl(
    CHAR *filename_r,
    DWORD maxrecno,
    DWORD reclen
)
```

### パラメータ

---

filename_r	ファイル名文字列へのポインタ
maxrecno	ファイルの最大レコード番号
reclen	データレコードのバイトサイズ

### 戻り値

---

ステータスコード

### 規則

---

- (1) 本関数で作成したファイルはデータ0件のファイルとして入力／更新モードでオープンしてアクセスすることができます。
- (2) reclenに0を指定した場合は規定値256バイトを設定します。
- (3) 最大レコード番号を指定した場合は、そのレコード番号まで格納するファイル領域を確保します。

### 主なステータスコード

---

IFNORMAL = 0 (0x0000)

意味 : 処理は正常に終了した

IFALDYFN = 101 (0x0065)

意味 : 指定したファイル名のファイルは既に存在する

IFFLBUSY = 205 (0x00cd)

意味 : このファイルは現在他の利用者が使用中である

IFRSZERR = 206 (0x00ce)

意味 : 指定したレコードサイズが不正である

IFPARMER = 804 (0x0324)

意味 : パラメータの指定に誤りがある

IFFNAMER = 805 (0x0325)

意味 : ファイル名の指定が不正である

IFFLEXER = 1007 (0x03ef)

意味 : ファイルサイズを確保できない

IFDRDBER = 1101 (0x044d)

意味 : ファイル識別子管理ファイルのアクセスエラーが発生した



## OPEN\_RL関数

IFAS相対編成ファイルをオープンします。

### 書式

```
DWORD ifas_open_rl(
    CHAR *filename_r,
    DWORD maxrecno,
    struct file_option_rl
    DWORD reclen,
    DWORD mode,
    DWORD extcheck,
    HANDLE *isfd
)
```

### パラメータ

**filename\_r** ファイル名文字列へのポインタ

**maxrecno** ファイルの最大レコード番号  
出力処理に対するファイル拡張の上限値、入力処理に対するファイル終端検出の値として設定します。

**option\_block** IFAS FILEオプションブロック構造体のポインタ  
IFAS FILEが処理を行う上での以下の既定値を変更したい場合に指定します。  
"0"を指定した項目は既定値が適用されます。

〈IFAS FILEオプションブロック構造体: **struct file\_option\_rl**〉

データ型	構造体メンバ名	内 容	既定値
DWORD	f_rel_nbuf	バッファ数	20
DWORD	f_rel_area	システム用(0をセット)	

**reclen** データレコードのバイトサイズ

**mode** オープンモード  
アクセスモード、ロックモード、トランザクションオプションを以下の例のように組み合わせて指定します。  
例 : IFUPDATE + IFLOCKR + IFROLLBACK  
IFINPUT | IFLOCKS

- アクセスモード(必須)
  - IFINPUT 入力モードでオープンします。
  - IFOUTPUT 出力モードでオープンします。
  - IFUPDATE 更新モードでオープンします。
- ロックモード(必須)
  - IFLOCKE ファイル単位で排他ロックします。
  - IFLOCKS ファイル単位で共有ロックします。
  - IFLOCKR レコード単位でロックします。
  - IFFREE ロック制御を行いません。
- トランザクションオプション
  - IFROLLBACK ロールバック時ファイルの更新を取り消します。
  - IFNOLOG 更新ログを収集しません。

**extcheck** ファイル拡張指示設定  
レコード出力時のファイル拡張に対するエラー指示と、オープン時のファイル領域

確保指示(出力モードの場合のみ)を以下の例のように組み合わせて指定します。

例 : IFEXTEND + IFALOCAT  
IFEXTERR | IFREFORM

・拡張エラー指示

IFEXTEND 最大レコード番号を超えたレコード出力を認めます。  
IFEXTERR 最大レコード番号を超えたレコード出力をエラーにします。

・ファイル領域確保指示(出力モードの場合のみ)

IFNOALOC オープン時にレコード格納用のファイル領域を確保しません。  
IFALOCAT maxrecnoパラメータで指定した最大レコード番号までのレコード格納用のファイル領域をオープン時に確保します。  
IFREFORM 既にファイルが存在する場合そのファイル領域を有効として初期化します。  
ファイルが存在しない場合はIFNOALOCと同様の処理をします。

isfd ファイル記述子

OPENが成功した場合、IFAS FILEはファイル記述子を返却します。この記述子は、このファイルをオープンしたスレッド内でのみ有効となります。

## 戻り値

ステータスコード

## 規則

- (1) 出力モードを指定した場合のみ、指定したファイル名のファイルが存在しなければ自動的にファイルを新規作成します。入力モード、更新モードの場合は存在しないファイルのファイル名を指定するとエラーになります。
- (2) 出力モードを指定した場合はファイルの内容をすべて初期化します。
- (3) アクセスモードとロックモードでオープン可能な組み合わせは以下のとおりです。

(○:可、×:不可)

	IFINPUT	IFOUTPUT	IFUPDATE
IFLOCKE	○	○	○
IFLOCKS	○	×	×
IFLOCKR	○	×	○
IFFREE	○	×	×

- (4) 他プロセス(スレッド)間で同時にオープン可能なロックモードの組み合わせは以下のとおりです。

(○:可、×:不可、△:IFLOCKR側が入力モードの場合のみ可)

	IFLOCKE	IFLOCKS	IFLOCKR	IFFREE
IFLOCKE	×	×	×	○
IFLOCKS	×	○	△	○
IFLOCKR	×	△	○	○
IFFREE	○	○	○	○

- (5) ロックモードでIFFREEを指定した場合は他のプロセスのロック状態に関わらずそのファイルのデータを入力できるので、入力データに矛盾が生じる場合もあり得ます。
- (6) IFROLLBACKを指定してオープンしたファイルに対してアクセス関数を実行する場合は、その前にTXSTART関数が実行されていなければなりません。
- (7) 出力モードでレコードサイズに0を指定した場合は既定値の256バイトをレコードサイズとします。

- (8) IFNOLOGを指定した場合はLOGOPEN関数によって更新ログファイルがオープンされていてもログ収集を行いません。
- (9) トランザクションオプションに何も指定していない場合は、更新ログファイルがオープンされていれば更新ログを収集しますが、ロールバックは行いません。
- (10) 出力モードの場合、extcheckパラメータの組み合わせによる処理は以下のようになります。

	IFNOALOC	IFALOCAT	IFREFORM
IFEXTEND	*1	*2	*3
IFEXTERR	*4	*5	*6

- \*1: オープン時はレコード格納用のファイル領域は確保せず、レコード出力によってディスク容量があるかぎり動的に拡張します。
- \*2: オープン時にmaxrecnoパラメータで指定したレコードまで格納するファイル領域を確保し、レコード出力によってディスク容量があるかぎり動的に拡張します。maxrecnoパラメータが指定されていない場合は \*1と同様の処理をします。
- \*3: 既にファイルが存在する場合そのファイルサイズを有効としてオープンし、レコード出力によってディスク容量があるかぎり動的に拡張します。新規ファイルの場合は \*1と同様の処理をします。
- \*4: オープン時はレコード格納用のファイル領域は確保せずレコード出力によってmaxrecnoパラメータで指定したレコード番号を格納できるサイズまで動的に拡張します。maxrecnoパラメータが指定されていない場合はオープンパラメータエラーになります。
- \*5: オープン時にmaxrecnoパラメータで指定したレコードまで格納するファイル領域を確保し、以降ファイルの動的拡張は行いません。maxrecnoパラメータが指定されていない場合はオープンパラメータエラーになります。
- \*6: 既にファイルが存在する場合そのファイルサイズを有効としてオープンし、maxrecnoで指定された値が従来のファイル領域より大きい場合はレコード出力によって動的に拡張します。maxrecnoパラメータが指定されていない場合や従来のファイルサイズ以下の場合は動的拡張を行いません。新規ファイルの場合は \*4と同様の処理をします。
- (11) 更新モードの場合、extcheckパラメータの指定は以下のようになります。
- IFEXTEND : maxrecnoパラメータ指定の有無に関わらず、レコード出力によってディスク容量があるかぎり動的に拡張します。
- IFEXTERR : maxrecnoパラメータで指定された最大レコード番号を超えた動的拡張、および参照はエラーとなります。  
maxrecnoパラメータの指定がない場合は従来のコード番号を超えた動的拡張はエラーになります。
- (12) 他編成ファイルを相対編成ファイルとしてオープンする場合は、そのファイルを本来の編成ファイルとして使用する利用者と同時にオープンすることはできません。

## 主なステータスコード

IFNORMAL = 0 (0x0000)

意味 : 処理は正常に終了した

IFNMLNLG = 3 (0x0003)

意味 : 処理は正常終了したが更新ログ出力は行わない

IFNMLNLK = 4 (0x0004)

意味 : 処理は正常に終了したがロック制御は行わない

IFNFNDFN = 201 (0x00c9)

意味 : 指定したファイル名のファイルが存在しない

IFNOIFFL = 203 (0x00cb)

意味 : 指定したファイル名のファイルはIFAS ファイルではない

IFNOINIT = 204 (0x00cc)

意味 : 指定したファイル名のファイルは初期化されていない

IFFLBUSY = 205 (0x00cd)

意味 : このファイルは現在他の利用者が使用中である

IFRSZERR = 206 (0x00ce)

意味 : 指定したレコードサイズが不正である

IFOPCTER = 210 (0x00d2)

意味 : 同時にオープン可能な最大数を超過してオープン要求を行った

IFNOWRIT = 213 (0x00d5)

意味 : 書き込み不可属性のファイルである

IFOPMDER = 801 (0x0321)

意味 : アクセスモードとロックモードの組み合わせが不正である

IFPARMER = 804 (0x0324)

意味 : パラメータの指定に誤りがある

IFFNAMER = 805 (0x0325)

意味 : ファイル名の指定が不正である

IFABTLCK = 1003 (0x03eb)

意味 : ファイルはリカバリ処理に失敗して壊れている可能性がある

IFMMEXER = 1004 (0x03ec)

意味 : メモリの確保に失敗した

IFFLEXER = 1007 (0x03ef)

意味 : ファイルサイズを確保できない

IFNOCLSF = 1009 (0x03f1)

意味 : シャットダウン時に終了処理が行われなかったファイルである

IFDRDBER = 1101 (0x044d)

意味 : ファイル識別子管理ファイルのアクセスエラーが発生した

IFINTPRC = 1201 (0x04b1)

意味 : IFAS FILE初期化プロセスが動作していない

IFSGIFER = 1204 (0x01204)

意味 : IFAS FILEのSG情報が不正である

IFPRTCER = 1205 (0x01205)

意味 : IFAS FILEを使用できない環境である

## CLOSE\_RL関数

IFAS相対編成ファイルをクローズします。

### 書式

---

```
DWORD    ifas_close_rl(  
HANDLE    isfd  
)
```

### パラメータ

---

isfd            OPEN関数で返却するファイル記述子

### 戻り値

---

ステータスコード

### 規則

---

- (1) オープンしていないファイルに対して本関数を実行するとエラーとなります。
- (2) 本関数によってファイルロック、レコードロックはすべて解除されます。
- (3) IFROLLBACK指定でオープンしたファイルはトランザクション処理中に本関数を実行するとエラーとなります。

### 主なステータスコード

---

IFNORMAL = 0    (0x0000)

意味 : 処理は正常に終了した

IFBADHND = 211 (0x00d3)

意味 : 指定したファイルはオープンされていない

IFNENDTX = 703 (0x02bf)

意味 : トランザクション処理が終了していない

IFDRDBER = 1101 (0x044d)

意味 : ファイル識別子管理ファイルのアクセスエラーが発生した

## PUT\_RL関数

IFAS相対編成ファイルにデータレコードを1件書き出します。

### 書式

```
DWORD ifas_put_rl(
HANDLE isfd,
LPVOID record,
DWORD access,
DWORD recnum,
DWORD curr
)
```

### パラメータ

isfd	OPEN関数で返却するファイル記述子
record	書き出すデータレコード領域のポインタ
access	アクセス種別
	IFADDREC 最大レコード番号のレコードとして書き出します。
	IFRECNO 指定したレコード番号のレコードとして書き出します。
recnum	レコード番号
	IFRECNO指定の場合、書き出すレコード番号を指定します。
curr	以下のいずれかを選択します。
	0: カレントレコードを変更しません
	1: 書き出したレコードをカレントレコードに変更します

### 戻り値

ステータスコード

### 規則

- (1) オープンしていないファイルに対して本関数を実行するとエラーとなります。
- (2) 入力モードでオープンされたファイルに対して本関数を実行するとエラーとなります。
- (3) 本関数によって書き出したレコードのレコード番号を知るためにはGETRECNO関数を実行します。
- (4) オープンで拡張エラー指示を指定した場合、最大レコード番号を超えた書き出し要求に対して”IFWTMXRN”を返却します。

### 主なステータスコード

IFNORMAL = 0 (0x0000)  
意味 : 処理は正常に終了した  
IFBADHND = 211 (0x00d3)  
意味 : 指定したファイルはオープンされていない  
IFDEDLOK = 309 (0x0135)  
意味 : デッドロックが発生した  
IFLKLT OV = 314 (0x013a)  
意味 : 1スレッドあたりの最大ロック制限値を超えた  
IFTXRBFL = 401 (0x0191)

意味 : TXSTART関数が実行されていない  
IFALDYRN = 405 (0x0195)  
意味 : 指定したレコード番号のレコードは既に存在する  
IFWTMXRN = 406 (0x0196)  
意味 : 指定最大レコード番号を超える書き出し要求を行った  
IFOPMDER = 801 (0x0321)  
意味 : オープンアクセスモードと機能の組み合わせが不正である  
IFPARMER = 804 (0x0324)  
意味 : パラメータの指定に誤りがある  
IFFLEXER = 1007 (0x03ef)  
意味 : ファイルサイズの拡張ができない  
IFERRFIL = 1005 (0x03ed)  
意味 : エラーが解決していないファイルにアクセス要求を行った

## GET\_RL関数

IFAS相対編成ファイルからデータレコードを1件読み込み、カレントレコードを変更します。

### 書式

```
DWORD    ifas_get_rl(
HANDLE    isfd,
LPVOID    record,
DWORD     access,
DWORD     pmode,
DWORD     recnum
)
```

### パラメータ

<b>isfd</b>	OPEN関数で返却するファイル記述子
<b>record</b>	読み込むデータレコード領域のポインタ
<b>access</b>	アクセス種別 IFRECNO                      レコード番号に基づいたアクセスを行います。 IFCURRE                      カレントレコードに基づいたアクセスを行います。
<b>pmode</b>	位置付けモード accessでIFRECNOを選択した場合は、以下のパラメータの中から選択指定します。 IFFIRST                      有効最小レコード番号のレコードを読み込みます。 IFLAST                      有効最大レコード番号のレコードを読み込みます。 IFEQUAL                      指定レコード番号と一致するレコードを読み込みます。 IFGREAT                      指定レコード番号より大きいレコードを読み込みます。 IFGTEQ                      指定レコード番号以上のレコードを読み込みます。 IFLITTLE                      指定レコード番号より小さいレコードを読み込みます。 IFLTEQ                      指定レコード番号以下のレコードを読み込みます。 accessでIFCURREを選択した場合は、以下のパラメータの中から選択指定します。 IFEQUAL                      カレントレコードを読み込みます。 IFNEXT                      カレントの次のレコードを読み込みます。 オープン後最初のGET関数の場合にかぎり、キー順で先頭のレコードを読み込みます。 POSITION関数でレコードの位置付けを行った後の最初のGET関数の場合にはカレントレコードを読み込みます。 IFPREV                      カレントの前のレコードを読み込みます。 オープン後最初のGET関数の場合はステータスとしてIFLGCEOFを返却します。POSITION関数でレコードの位置付けを行った後の最初のGET関数の場合にはカレントレコードを読み込みます。
<b>recnum</b>	レコード番号 レコード番号指定で読み込む場合、レコード番号を指定します。

### 戻り値

ステータスコード

### 規則



- (1) オープンしていないファイルに対して本関数を実行するとエラーとなります。
- (2) 出力モードでオープンされたファイルに対して本関数を実行するとエラーとなります。
- (3) 本関数によって読み込んだレコードのレコード番号を知るためにはGETRECNO関数を実行します。
- (4) 末尾レコード番号のレコードがカレントの場合、IFNEXT指定で本関数を実行するとステータス”IFLGCEOF”を返却します。
- (5) 先頭レコード番号のレコードがカレントの場合、IFPREV指定で本関数を実行するとステータス”IFLGCEOF”を返却します。
- (6) レコード番号で検索したレコードがファイル内に存在しない場合、ステータス ”IFNFNDRC”を返却します。
- (7) オープンで指定した最大レコード番号を超えた読み込み要求がされた場合はステータス”IFRDMXRN”が返却されます。

## 主なステータスコード

---

IFNORMAL = 0 (0x0000)

意味 : 処理は正常に終了した

IFBADHND = 211 (0x00d3)

意味 : 指定したファイルはオープンされていない

IFNCURRC = 304 (0x0130)

意味 : カレントレコードが設定されていない

IFLGCEOF = 305 (0x0131)

意味 : ファイルの論理的終端を検出した

IFNFNDRC = 306 (0x0132)

意味 : 指定したレコード番号のレコードが存在しない

IFDEDLCK = 309 (0x0135)

意味 : デッドロックが発生した

IFLKLTOV = 314 (0x013a)

意味 : 1スレッドあたりの最大ロック制限値を超えた

IFRDMXRN = 316 (0x013c)

意味 : 指定最大レコード番号を超える読み込み要求を行った

IFOPMDER = 801 (0x0321)

意味 : オープンアクセスモードと機能の組み合わせが不正である

IFPARMER = 804 (0x0324)

意味 : パラメータの指定に誤りがある

IFERRFIL = 1005 (0x03ed)

意味 : エラーが解決していないファイルにアクセス要求を行った

## PUTX\_RL関数

IFAS相対編成ファイルのデータレコードを1件書き換えます。

### 書式

---

```
DWORD ifas_putx_rl(
    HANDLE isfd,
    LPVOID record,
    DWORD access,
    DWORD recnum
)
```

### パラメータ

---

isfd	OPEN関数で返却するファイル記述子
record	書き換えるデータレコード領域のポインタ
access	アクセス種別
	IFRECNO                      レコード番号に基づいたアクセスを行います。
	IFCURR                      カレントレコードをアクセスします。
recnum	レコード番号
	レコード番号指定で書き換える場合レコード番号を指定します。

### 戻り値

---

ステータスコード

### 規則

---

- (1) オープンしていないファイルに対して本関数を実行するとエラーとなります。
- (2) 入力モード、出力モードでオープンされたファイルに対して本関数を実行するとエラーとなります。
- (3) 本関数によって書き換えたレコードのレコード番号を知るためにはGETRECNO関数を実行します。
- (4) オープンで指定した最大レコード番号を超えた書き換え要求がされた場合はステータス”IFWTMXRN”が返却されます。

### 主なステータスコード

---

IFNORMAL = 0 (0x0000)  
 意味 : 処理は正常に終了した

IFBADHND = 211 (0x00d3)  
 意味 : 指定したファイルはオープンされていない

IFNCURRC = 304 (0x0130)  
 意味 : カレントレコードが設定されていない

IFNFNDRC = 306 (0x0132)  
 意味 : 指定したレコード番号のレコードが存在しない

IFDEDLOK = 309 (0x0135)  
 意味 : デッドロックが発生した

IFLKLT0V = 314 (0x013a)  
 意味 : 1スレッドあたりの最大ロック制限値を超えた

IFTXRBFL = 401 (0x0191)

意味 : TXSTART関数が実行されていない

IFWTMXRN = 406 (0x0196)

意味 : 指定最大レコード番号を超える書き換え要求を行った

IFOPMDER = 801 (0x0321)

意味 : オープンアクセスモードと機能の組み合わせが不正である

IFPARMER = 804 (0x0324)

意味 : パラメータの指定に誤りがある

IFERRFIL = 1005 (0x03ed)

意味 : エラーが解決していないファイルにアクセス要求を行った

## DELETE\_RL関数

IFAS相対編成ファイルのデータレコードを1件削除します。

### 書式

```
DWORD    ifas_delete_rl(
HANDLE    isfd,
DWORD    access,
DWORD    recnum
)
```

### パラメータ

isfd	OPEN関数で返却するファイル記述子
access	アクセス種別
	IFRECNO                      レコード番号に基づいたアクセスを行います。
	IFCURR                      カレントレコードをアクセスします。
recnum	レコード番号
	IFRECNO指定の場合、削除するレコード番号を指定します。

### 戻り値

ステータスコード

### 規則

- (1) オープンしていないファイルに対して本関数を実行するとエラーとなります。
- (2) 入力モード、出力モードでオープンされたファイルに対して本関数を実行するとエラーとなります。
- (3) 本関数によって削除したレコードのレコード番号を知るためにはGETRECNO関数を実行します。
- (4) オープンで指定した最大レコード番号を超えた削除要求がされた場合はステータス ” IFWTMXRN” が返却されます。

### 主なステータスコード

IFNORMAL = 0 (0x0000)  
 意味 : 処理は正常に終了した

IFBADHND = 211 (0x00d3)  
 意味 : 指定したファイルはオープンされていない

IFNCURRC = 304 (0x0130)  
 意味 : カレントレコードが設定されていない

IFNFNDRC = 306 (0x0132)  
 意味 : 指定したレコード番号のレコードが存在しない

IFDEDLOK = 309 (0x0135)  
 意味 : デッドロックが発生した

IFNPRMIX = 310 (0x0136)  
 意味 : 選択されているインデックスはプライマリインデックスではない

IFLKLTOV = 314 (0x013a)  
 意味 : 1スレッドあたりの最大ロック制限値を超えた

IFTXRBFL = 401 (0x0191)

意味 : TXSTART関数が実行されていない  
IFWTMXRN = 406 (0x0196)  
意味 : 指定最大レコード番号を超える削除要求を行った  
IFOPMDER = 801 (0x0321)  
意味 : オープンアクセスモードと機能の組み合わせが不正である  
IFPARMER = 804 (0x0324)  
意味 : パラメータの指定に誤りがある  
IFERRFIL = 1005 (0x03ed)  
意味 : エラーが解決していないファイルにアクセス要求を行った

## POSITION\_RL関数

IFAS相対編成ファイルに対し、データレコードの位置付けを行い、カレントレコードを変更します。

### 書式

```
DWORD    ifas_position_rl(
HANDLE    isfd,
DWORD    access,
DWORD    pmode,
DWORD    recnum
)
```

### パラメータ

<b>isfd</b>	OPEN関数で返却するファイル記述子
<b>access</b>	アクセス種別 IFRECNO                      レコード番号に基づいたアクセスを行います。 IFCURRE                      カレントレコードから位置付けを行います。
<b>pmode</b>	位置付けモード accessでIFRECNOを選択した場合は、以下のパラメータの中から選択指定します。 IFFIRST                      有効最小レコード番号のレコードに位置付けます。 IFLAST                      有効最大レコード番号のレコードに位置付けます。 IFEQUAL                      指定レコード番号と一致するレコードに位置付けます。 IFGREAT                      指定レコード番号より大きいレコードに位置付けます。 IFGTEQ                      指定レコード番号以上のレコードに位置付けます。 IFLITTLE                      指定レコード番号より小さいレコードに位置付けます。 IFLTEQ                      指定レコード番号以下のレコードに位置付けます。 accessでIFCURREを選択した場合は、以下のパラメータの中から選択指定します。 IFNEXT                      カレントの次のレコードに位置付けます。 IFPREV                      カレントの前のレコードに位置付けます。
<b>recnum</b>	レコード番号 レコード番号指定で位置付ける場合、レコード番号を指定します。

### 戻り値

ステータスコード

### 規則

- (1) オープンしていないファイルに対して本関数を実行するとエラーとなります。
- (2) 出力モードでオープンされたファイルに対して本関数を実行するとエラーとなります。
- (3) 本関数によって位置付けたレコードのレコード番号を知るためにはGETRECNO関数を実行します。
- (4) レコード番号で検索したレコードがファイル内に存在しない場合はステータス     "IFNENDRC"を返却します。
- (5) レコードの位置付けを行った場合、本関数実行後最初のGET関数でのみIFCURRE指定をすれば IFNEXT、IFEQUAL、IFPREVのどれを指定しても、位置付けたレコードを読み込むことができます。
- (6) オープンで指定した最大レコード番号を超えた位置付け要求がされた場合はステータス"IFRDMXRN"が返却されます。

## 主なステータスコード

---

IFNORMAL = 0 (0x0000)

意味 : 処理は正常に終了した

IFBADHND = 211 (0x00d3)

意味 : 指定したファイルはオープンされていない

IFNCURRC = 304 (0x0130)

意味 : カレントレコードが設定されていない

IFLGEOF = 305 (0x0131)

意味 : ファイルの論理的終端を検出した

IFNFNDRC = 306 (0x0132)

意味 : 指定したレコード番号のレコードが存在しない

IFDEDLOK = 309 (0x0135)

意味 : デッドロックが発生した

IFLKLTOV = 314 (0x013a)

意味 : 1スレッドあたりの最大ロック制限値を超えた

IFRDMXRN = 316 (0x013c)

意味 : 指定最大レコード番号を超える位置付け要求を行った

IFOPMDER = 801 (0x0321)

意味 : オープンアクセスモードと機能の組み合わせが不正である

IFPARMER = 804 (0x0324)

意味 : パラメータの指定に誤りがある

IFERRFIL = 1005 (0x03ed)

意味 : エラーが解決していないファイルにアクセス要求を行った

## SKIP\_RL関数

IFAS相対編成ファイルに対し、指定したレコード数分、前後方向にカレントレコードの位置を移動します。

### 書式

```
DWORD ifas_skip_rl(
HANDLE                                     isfd,
DWORD                                     pmode,
DWORD                                     recnt
)
```

### パラメータ

isfd	OPEN関数で返却するファイル記述子
pmode	位置付けモード
	IFBFILE                      先頭のレコードに位置付けます。
	IFFWDCNT                    指定したレコード数分先のレコードに位置付けます。
	IFRWDCNT                   指定したレコード数分後ろのレコードに位置付けます。
recnt	レコード数
	pmodeでIFFWDCNT、IFRWDCNTを指定した場合、カレントレコードからスキップするレコード数を指定します。

### 戻り値

ステータスコード

### 規則

- (1) オープンしていないファイルに対して本関数を実行するとエラーとなります。
- (2) 出力モードでオープンされたファイルに対して本関数を実行するとエラーとなります。
- (3) 本関数によって位置付けたレコードのレコード番号を知るためにはGETRECNO関数を実行します。
- (4) 本処理によってファイルの終端を検出した場合は、ステータス”IFLGEOF”を返却します。
- (5) 本処理によってカレントレコードを変更した後、GET関数でIFCURR指定のIFNEXTを実行すると位置付けたカレントの次のレコードを、GET関数でIFCURR指定のIFPREVを実行するとカレントの前のレコードを読み込みます。

### 主なステータスコード

```
IFNORMAL = 0    (0x0000)
意味  : 処理は正常に終了した
IFBADHND = 211 (0x00d3)
意味  : 指定したファイルはオープンされていない
IFNCURRC = 304 (0x0130)
意味  : カレントレコードが設定されていない
IFLGEOF = 305 (0x0131)
意味  : ファイルの論理的終端を検出した
IFDEDLOK = 309 (0x0135)
意味  : デッドロックが発生した
```



IFLKLT0V = 314 (0x013a)

意味 : 1スレッドあたりの最大ロック制限値を超えた

IFOPMDER = 801 (0x0321)

意味 : オープンアクセスモードと機能の組み合わせが不正である

IFPARMER = 804 (0x0324)

意味 : パラメータの指定に誤りがある

IFERRFIL = 1005 (0x03ed)

意味 : エラーが解決していないファイルにアクセス要求を行った

## ERASE\_RL関数

IFAS相対編成ファイルを削除します。

### 書式

---

```
DWORD    ifas_erase_rl(
          CHAR    *filename_r
          )
```

### パラメータ

---

filename\_r      ファイル名文字列へのポインタ

### 戻り値

---

ステータスコード

### 規則

---

- (1) 存在しないファイルのファイル名を指定するとエラーとなります。
- (2) アクセス処理中のファイルをファイル名に指定するとエラーとなります。
- (3) BUILD\_IDXFILE関数でインデックスファイルを追加作成したファイルに対し本関数を実行すると“IFFORGER”を返却します。索引順編成ファイルとしてERASE関数を実行してください。

### 主なステータスコード

---

IFNORMAL = 0    (0x0000)

意 味    : 処理は正常に終了した

IFNFNDFN = 201 (0x00c9)

意 味    : 指定したファイル名のファイルが存在しない

IFFLBUSY = 205 (0x00cd)

意 味    : このファイルは現在他の利用者が使用中である

IFFORGER = 214 (0x00d6)

意 味    : ファイル編成が誤っている

IFFNAMER = 805 (0x0325)

意 味    : ファイル名の指定が不正である

## RENAME\_RL関数

IFAS相対編成ファイルのファイル名を変更します。

### 書式

---

```
DWORD    ifas_rename_rl(
           CHAR                *old_filename_r,
           CHAR                *new_filename_r
           )
```

### パラメータ

---

old_filename_r	旧ファイル名文字列へのポインタ
new_filename_r	新ファイル名文字列へのポインタ

### 戻り値

---

ステータスコード

### 規則

---

- (1) 存在しないファイルのファイル名を旧ファイル名に指定するとエラーとなります。
- (2) 既に存在するファイルのファイル名を新ファイル名に指定するとエラーとなります。
- (3) アクセス処理中のファイルを旧ファイル名に指定するとエラーとなります。
- (4) BUILD\_IDXFILE関数でインデックスファイルを追加作成したファイルに対し本関数を実行すると“IFFORGER”を返却します。索引順編成ファイルとしてRENAME関数を実行してください。

### 主なステータスコード

---

IFNORMAL = 0 (0x0000)

意味 : 処理は正常に終了した

IFALDYFN = 101 (0x0065)

意味 : 指定したファイル名のファイルは既に存在する

IFNFNDFN = 201 (0x00c9)

意味 : 指定したファイル名のファイルが存在しない

IFFLBUSY = 205 (0x00cd)

意味 : このファイルは現在他の利用者が使用中である

IFFORGER = 214 (0x00d6)

意味 : ファイル編成が誤っている

IFFNAMER = 805 (0x0325)

意味 : ファイル名の指定が不正である

## GETINFO\_RL関数

指定したIFAS相対編成ファイルの情報を取得します。

### 書式

---

```
DWORD    ifas_getinfo_rl(
HANDLE    isfd,
LPVOID    buffer
)
```

### パラメータ

---

**isfd**            OPEN関数で返却するファイル記述子  
**buffer**          返却する情報格納領域のポインタ

〈ファイル情報構造体: **struct relfinfo**〉

データ型	構造体メンバ名	内 容
DWORD	rl_recsize	データレコード長
DWORD	rl_nrecords	レコード数
DWORD	rl_maxrecno	最大レコード番号

### 戻り値

---

ステータスコード

### 規則

---

- (1) オープンしていないファイルに対して本関数を実行するとエラーとなります。
- (2) 索引順編成ファイルのデータファイルをIFAS相対編成ファイルとしてオープンして本関数を実行した場合、レコード数は返却できません。値は0がセットされます。

### 主なステータスコード

---

IFNORMAL = 0 (0x0000)

意 味 : 処理は正常に終了した

IFBADHND = 211 (0x00d3)

意 味 : 指定したファイルはオープンされていない

IFFORGER = 214 (0x00d6)

意 味 : ファイル編成が誤っている

IFERRFIL = 1005 (0x03ed)

意 味 : エラーが解決していないファイルにアクセス要求を行った

## BUILD\_IDXFILE\_RL関数

作成済みのIFAS相対編成ファイルに対してインデックスファイルを追加作成し、IFAS索引順編成ファイルとして扱えるようにします。

### 書式

```
DWORD    ifas_build_idxfile_rl(
        CHAR    *filename_r,
        CHAR    *filename_i,
        struct ifkeydesc    *keydesc
    )
```

### パラメータ

**filename\_r** 作成済み相対編成ファイル名文字列へのポインタ  
**filename\_i** 新規作成インデックスファイル名文字列へのポインタ  
**keydesc** プライマリキーにおけるキー記述構造体のポインタ

〈キー記述構造体: **struct ifkeydesc**〉

データ型	構造体メンバ名	内 容	既定値
DWORD	ifk_number	キーパート数 (0=インデックスを作成しない)	1
DWORD	ifk_dupflag	重複キー許可 (0=重複不許可: 0以外はエラー)	0
struct ifkeypart	k_prt[n]	キーパート構造体 (キーパートは最大16個まで指定可能)	

〈キーパート構造体: **struct ifkeypart**〉

データ型	構造体メンバ名	内 容	既定値
DWORD	ifk_key_offset	レコード内キーオフセット	0
DWORD	ifk_key_size	キーサイズ	0
DWORD	ifk_type	キータイプ:	IFCHAR
		IFCHAR // 文字	
		IFBIN4 // DWORD (4バイト)	
		IFBIN2 // WORD (2バイト)	
		IFSBIN4 // LONG (4バイト)	
		IFSBIN2 // INT (2バイト)	
		IFFLOAT // FLOAT (4バイト)	
		IFDOUBLE // DOUBLE (8バイト)	

### 戻り値

ステータスコード

### 規則

- (1) 存在しない相対編成ファイル名を指定するとエラーとなります。
- (2) 既にインデックスファイルが作られている順／相対編成ファイル、または作成済みの索引順編成ファイルのデータファイルに対し、新規に別のインデックスファイルを作成すると従来のインデックスファイルは使用できなくなります。
- (3) 本関数によって設定可能なインデックスはプライマリインデックスのみです。サブインデックスを作

成する場合は、索引順編成ファイルとしてオープンしADDINDEX関数を用います。

- (4) インデックスファイル作成後でもデータファイルは順／相対編成ファイルとして扱うことができます。また、最初から索引順編成ファイルとして作成したファイルのデータファイルを順／相対編成ファイルとして扱うこともできます。ただし、相対編成ファイルとしてレコードの追加／削除／更新処理、順編成ファイルとして追加／更新処理を行った後、再び索引順編成ファイルとして使用する場合は、REBUILD\_IDXFILE関数によってインデックスファイルを再作成する必要があります。
- (5) 本関数が処理途中でエラーを検出して失敗した場合は、追加作成しようとしたインデックスファイルは消去します。

### 主なステータスコード

---

IFNORMAL = 0 (0x0000)

意味 : 処理は正常に終了した

IFNFNDFN = 201 (0x00c9)

意味 : 指定したファイル名のファイルが存在しない

IFFLBUSY = 205 (0x00cd)

意味 : このファイルは現在他の利用者が使用中である

IFDUPKEY = 402 (0x0192)

意味 : プライマリインデックスで重複キーを検出した

IFKDSCER = 802 (0x0322)

意味 : キー記述構造体が不正である

IFFNAMER = 805 (0x0325)

意味 : ファイル名の指定が不正である

IFDRDBER = 1101 (0x044d)

意味 : ファイル識別子管理ファイルのアクセスエラーが発生した

### 3-2-3 順編成ファイルアクセス関数

IFAS順編成ファイルをアクセスするための関数を紹介します。

<code>ifas_create_sq</code>	順編成ファイルを作成する
<code>ifas_open_sq</code>	順編成ファイルをオープンする
<code>ifas_close_sq</code>	順編成ファイルをクローズする
<code>ifas_put_sq</code>	順編成ファイルにレコードを書き出す
<code>ifas_get_sq</code>	順編成ファイルからレコードを読み込む
<code>ifas_putx_sq</code>	順編成ファイルのレコードを書き替える
<code>ifas_skip_sq</code>	順編成ファイルのカレントレコードを移動する
<code>ifas_erase_sq</code>	順編成ファイルを削除する
<code>ifas_rename_sq</code>	順編成ファイルの名前を変更する
<code>ifas_getinfo_sq</code>	順編成ファイルの情報を取得する
<code>ifas_build_idxfile_sq</code>	順編成ファイルにインデックスファイルを作成する

## CREATE\_SQL関数

IFAS順編成ファイルを作成します。

### 書式

---

```
DWORD    ifas_create_sq(
          CHAR    *filename_s,
          DWORD    reclen
          )
```

### パラメータ

---

filename_s	ファイル名文字列へのポインタ
reclen	データレコードのバイトサイズ

### 戻り値

---

ステータスコード

### 規則

---

- (1) 本関数で作成したファイルはデータ0件のファイルとして入力／更新モードでオープンしてアクセスすることができます。
- (2) reclenに0を指定した場合は規定値256バイトを設定します。

### 主なステータスコード

---

IFNORMAL = 0 (0x0000)

意味 : 処理は正常に終了した

IFALDYFN = 101 (0x0065)

意味 : 指定したファイル名のファイルは既に存在する

IFFLBUSY = 205 (0x00cd)

意味 : このファイルは現在他の利用者が使用中である

IFRSZERR = 206 (0x00ce)

意味 : 指定したレコードサイズが不正である

IFPARMER = 804 (0x0324)

意味 : パラメータの指定に誤りがある

IFFNAMER = 805 (0x0325)

意味 : ファイル名の指定が不正である

IFFLEXER = 1007 (0x03ef)

意味 : ファイルサイズを確保できない

IFDRDBER = 1101 (0x044d)

意味 : ファイル識別子管理ファイルのアクセスエラーが発生した



## OPEN\_SQ関数

IFAS順編成ファイルをオープンします。

### 書式

```
DWORD ifas_open_sq(
    CHAR *filename_s,
    struct file_option_sq *option_block,
    DWORD reclen,
    DWORD mode,
    HANDLE *isfd
)
```

### パラメータ

**filename\_s** ファイル名文字列へのポインタ  
**option\_block** IFAS FILEオプションブロック構造体のポインタ  
 IFAS FILEが処理を行う上での以下の既定値を変更したい場合に指定します。  
 "0"を指定した項目は既定値が適用されます。

〈IFAS FILEオプションブロック構造体: **struct file\_option\_sq**〉

データ型	構造体メンバ名	内 容	既定値
DWORD	f_seq_nbuf	バッファ数	20
DWORD	f_seq_area	システム用 (0をセット)	

**reclen** データレコードのバイトサイズ  
**mode** オープンモード  
 アクセスモード、ロックモード、トランザクションオプションを以下の例のように組み合わせて指定します。  
 例 : IFUPDATE + IFLOCKR + IFROLLBACK  
 IFINPUT | IFLOCKS  
 ・アクセスモード(必須)  
 IFINPUT 入力モードでオープンします。  
 IFOUTPUT 出力モードでオープンします。  
 IFUPDATE 更新モードでオープンします。  
 ・ロックモード(必須)  
 IFLOCKE ファイル単位で排他ロックします。  
 IFLOCKS ファイル単位で共有ロックします。  
 IFLOCKR レコード単位でロックします。  
 IFFREE ロック制御を行いません。  
 ・トランザクションオプション  
 IFROLLBACK ロールバック時ファイルの更新を取り消します。  
 IFNOLOG 更新ログを収集しません。  
**isfd** ファイル記述子  
 OPENが成功した場合、IFAS FILEはファイル記述子を返却します。この記述子は、このファイルをオープンしたスレッド内でのみ有効となります。

### 戻り値

ステータスコード

## 規則

- (1) 出力モードを指定した場合のみ、指定したファイル名のファイルが存在しなければ自動的にファイルを新規作成します。入力モード、更新モードの場合は存在しないファイルのファイル名を指定するとエラーになります。
- (2) 出力モードを指定した場合はファイルの内容をすべて初期化します。
- (3) アクセスモードとロックモードでオープン可能な組み合わせは以下のとおりです。

(○:可、×:不可)

	IFINPUT	IFOUTPUT	IFUPDATE
IFLOCKE	○	○	○
IFLOCKS	○	×	×
IFLOCKR	○	×	○
IFFREE	○	×	×

- (4) 他プロセス(スレッド)間で同時にオープン可能なロックモードの組み合わせは以下のとおりです。

(○:可、×:不可、△:IFLOCKR側が入力モードの場合のみ)

可)

	IFLOCKE	IFLOCKS	IFLOCKR	IFFREE
IFLOCKE	×	×	×	○
IFLOCKS	×	○	△	○
IFLOCKR	×	△	○	○
IFFREE	○	○	○	○

- (5) ロックモードでIFFREEを指定した場合は他のプロセスのロック状態に関わらずそのファイルのデータを入力できるので、入力データに矛盾が生じる場合もあり得ます。
- (6) IFROLLBACKを指定してオープンしたファイルに対してアクセス関数を実行する場合は、その前にTXSTART関数が実行されていなければなりません。
- (7) 出力モードでレコードサイズに0を指定した場合は既定値の256バイトをレコードサイズとします。
- (8) IFNOLOGを指定した場合はLOGOPEN関数によって更新ログファイルがオープンされていても更新ログ収集を行いません。
- (9) トランザクションオプションに何も指定していない場合は、更新ログファイルがオープンされていれば更新ログを収集しますが、ロールバックは行いません。
- (10) 他編成ファイルを順編成ファイルとしてオープンする場合は、そのファイルを本来の編成ファイルとして使用する利用者と同時にオープンすることはできません。

## 主なステータスコード

IFNORMAL = 0 (0x0000)

意味 : 処理は正常に終了した

IFNMLNLG = 3 (0x0003)

意味 : 処理は正常終了したが更新ログ出力は行わない

IFNMLNLK = 4 (0x0004)

意味 : 処理は正常に終了したがロック制御は行わない

IFNFNDFN = 201 (0x00c9)

意味 : 指定したファイル名のファイルが存在しない

IFNOIFFL = 203 (0x00cb)

意味 : 指定したファイル名のファイルはIFAS ファイルではない

IFNOINIT = 204 (0x00cc)

意味 : 指定したファイル名のファイルは初期化されていない

IFFLBSY = 205 (0x00cd)

意味 : このファイルは現在他の利用者が使用中である  
IFRSZERR = 206 (0x00ce)  
意味 : 指定したレコードサイズが不正である  
IFOPCTER = 210 (0x0210)  
意味 : 同時にオープン可能な最大数を超えてオープン要求を行った  
IFNOWRIT = 213 (0x0213)  
意味 : 書き込み不可属性のファイルである  
IFOPMDER = 801 (0x0321)  
意味 : アクセスモードとロックモードの組み合わせが不正である  
IFPARMER = 804 (0x0324)  
意味 : パラメータの指定に誤りがある  
IFFNAMER = 805 (0x0325)  
意味 : ファイル名の指定が不正である  
IFABTLCK = 1003 (0x03eb)  
意味 : ファイルはリカバリ処理に失敗して壊れている可能性がある  
IFMMEXER = 1004 (0x03ec)  
意味 : メモリの確保に失敗した  
IFFLEXER = 1007 (0x03ef)  
意味 : ファイルサイズを確保できない  
IFNOCLSF = 1009 (0x03f1)  
意味 : シャットダウン時に終了処理が行われなかったファイルである  
IFDRDBER = 1101 (0x044d)  
意味 : ファイル識別子管理ファイルのアクセスエラーが発生した  
IFINTPRC = 1201 (0x04b1)  
意味 : IFAS FILE初期化プロセスが動作していない  
IFSGIFER = 1204 (0x01204)  
意味 : IFAS FILEのSG情報が不正である  
IFPRTCER = 1205 (0x01205)  
意味 : IFAS FILEを使用できない環境である

## CLOSE\_SQ関数

IFAS順編成ファイルをクローズします。

### 書式

---

```
DWORD ifas_close_sq(
HANDLE isfd
)
```

### パラメータ

---

isfd                      OPEN関数で返却するファイル記述子

### 戻り値

---

ステータスコード

### 規則

---

- (1) オープンしていないファイルに対して本関数を実行するとエラーとなります。
- (2) 本関数によってファイルロック、レコードロックはすべて解除されます。
- (3) IFROLLBACK指定でオープンしたファイルはトランザクション処理中に本関数を実行するとエラーとなります。

### 主なステータスコード

---

IFNORMAL = 0    (0x0000)

意 味    : 処理は正常に終了した

IFBADHND = 211 (0x00d3)

意 味    : 指定したファイルはオープンされていない

IFNENDTX = 703 (0x02bf)

意 味    : トランザクション処理が終了していない

IFDRDBER = 1101 (0x044d)

意 味    : ファイル識別子管理ファイルのアクセスエラーが発生した

## PUT\_SQ関数

IFAS順編成ファイルにデータレコードを1件書き出します。

### 書式

```
DWORD    ifas_put_sq(
          HANDLE          isfd,
          LPVOID          record,
          DWORD           curr
          )
```

### パラメータ

isfd	OPEN関数で返却するファイル記述子
record	書き出すデータレコード領域のポインタ
curr	以下のいずれかを選択します。 0: カレントレコードを変更しません 1: 書き出したレコードをカレントレコードに変更します

### 戻り値

ステータスコード

### 規則

- (1) オープンしていないファイルに対して本関数を実行するとエラーとなります。
- (2) 入力モードでオープンしたファイルに対して本関数を実行するとエラーとなります。
- (3) 本関数によって書き出したレコードのレコード番号を知るためにはGETRECNO関数を実行します。

### 主なステータスコード

IFNORMAL = 0 (0x0000)	意味 : 処理は正常に終了した
IFBADHND = 211 (0x00d3)	意味 : 指定したファイルはオープンされていない
IFDEDLOK = 309 (0x0135)	意味 : デッドロックが発生した
IFLKLT OV = 314 (0x013a)	意味 : 1スレッドあたりの最大ロック制限値を超えた
IFTXRBFL = 401 (0x0191)	意味 : TXSTART関数が実行されていない
IFOPMDER = 801 (0x0321)	意味 : オープンアクセスモードと機能の組み合わせが不正である
IFPARMER = 804 (0x0324)	意味 : パラメータの指定に誤りがある
IFFLEXER = 1007 (0x03ef)	意味 : ファイルサイズの拡張ができない
IFERRFIL = 1005 (0x03ed)	意味 : エラーが解決していないファイルにアクセス要求を行った

## GET\_SQ関数

IFAS順編成ファイルから順次にデータレコードを1件読み込み、カレントレコードを変更します。

### 書式

```
DWORD    ifas_get_sq(
HANDLE    isfd,
LPVOID    record
)
```

### パラメータ

isfd	OPEN関数で返却するファイル記述子
record	読み込むデータレコード領域のポインタ

### 戻り値

ステータスコード

### 規則

- (1) オープンしていないファイルに対して本関数を実行するとエラーとなります。
- (2) 出力モードでオープンされたファイルに対して本関数を実行するとエラーとなります。
- (3) 本関数によって読み込んだレコードのレコード番号を知るためにはGETRECNO関数を実行します。
- (4) 末尾レコード番号のレコードがカレントの場合、本関数を実行するとステータス "IFLGCEOF" を返却します。

### 主なステータスコード

```
IFNORMAL = 0    (0x0000)
意味  : 処理は正常に終了した
IFBADHND = 211  (0x00d3)
意味  : 指定したファイルはオープンされていない
IFLGCEOF = 305  (0x0131)
意味  : ファイルの論理的終端を検出した
IFDEDL0K = 309  (0x0135)
意味  : デッドロックが発生した
IFLKLT0V = 314  (0x013a)
意味  : 1スレッドあたりの最大ロック制限値を超えた
IFOPMDER = 801  (0x0321)
意味  : オープンアクセスモードと機能の組み合わせが不正である
IFPARMER = 804  (0x0324)
意味  : パラメータの指定に誤りがある
IFERRFIL = 1005 (0x03ed)
意味  : エラーが解決していないファイルにアクセス要求を行った
```

## PUTX\_SQ関数

IFAS順編成ファイルのカレントデータレコードを書き換えます。

### 書式

---

```
DWORD    ifas_putx_sq(
HANDLE    isfd,
LPVOID    record
)
```

### パラメータ

---

isfd	OPEN関数で返却するファイル記述子
record	書き換えるデータレコード領域のポインタ

### 戻り値

---

ステータスコード

### 規則

---

- (1) オープンしていないファイルに対して本関数を実行するとエラーとなります。
- (2) 入力モード、出力モードでオープンされたファイルに対して本関数を実行するとエラーとなります。
- (3) 本関数によって書き換えたレコードのレコード番号を知るためにはGETRECNO関数を実行します。
- (4) 直前に実行した関数が正常終了したGET関数でなかった場合は、ステータス "IFNCURRC" が返却されます。

### 主なステータスコード

---

```
IFNORMAL = 0    (0x0000)
意味      : 処理は正常に終了した
IFBADHND = 211  (0x00d3)
意味      : 指定したファイルはオープンされていない
IFNCURRC = 304  (0x0130)
意味      : カレントレコードが設定されていない
IFDEDLOK = 309  (0x0135)
意味      : デッドロックが発生した
IFLKLT0V = 314  (0x013a)
意味      : 1スレッドあたりの最大ロック制限値を超えた
IFTXRBFL = 401  (0x0191)
意味      : TXSTART関数が実行されていない
IFOPMDER = 801  (0x0321)
意味      : オープンアクセスモードと機能の組み合わせが不正である
IFPARMER = 804  (0x0324)
意味      : パラメータの指定に誤りがある
IFERRFIL = 1005 (0x03ed)
意味      : エラーが解決していないファイルにアクセス要求を行った
```

## SKIP\_SQ関数

IFAS順編成ファイルに対し、指定したレコード数分、前後方向にカレントレコードの位置を移動します。

### 書式

```
DWORD    ifas_skip_sq(
HANDLE    isfd,
DWORD    pmode,
DWORD    recnt
)
```

### パラメータ

isfd	OPEN関数で返却するファイル記述子
pmode	位置付けモード
	IFBFILE 先頭のレコードに位置付けます。
	IFEFILE 最後のレコードに位置付けます。
	IFFWDCNT 指定したレコード数分先のレコードに位置付けます。
	IFRWDCNT 指定したレコード数分後ろのレコードに位置付けます。
recnt	レコード数
	pmodeでIFFWDCNT、IFRWDCNTを指定した場合、カレントレコードからスキップするレコード数を指定します。

### 戻り値

ステータスコード

### 規則

- (1) オープンしていないファイルに対して本関数を実行するとエラーとなります。
- (2) 出力モードでオープンされたファイルに対して本関数を実行するとエラーとなります。
- (3) 本関数によって位置付けたレコードのレコード番号を知るためにはGETRECNO関数を実行します。
- (4) 本処理によってファイルの終端を検出した場合は、ステータス”IFLGEOF”を返却します。
- (5) 本処理によってカレントレコードを変更した後、GET関数でIFCURR指定のIFNEXTを実行すると位置付けたカレントの次のレコードを読み込みます。

### 主なステータスコード

```
IFNORMAL = 0    (0x0000)
意味  : 処理は正常に終了した
IFBADHND = 211 (0x00d3)
意味  : 指定したファイルはオープンされていない
IFNCURRC = 304 (0x0130)
意味  : カレントレコードが設定されていない
IFLGEOF = 305 (0x0131)
意味  : ファイルの論理的終端を検出した
IFDEDLOK = 309 (0x0135)
意味  : デッドロックが発生した
IFLKLT0V = 314 (0x013a)
```



意味 : 1スレッドあたりの最大ロック制限値を超えた  
IFOPMDER = 801 (0x0321)  
意味 : オープンアクセスモードと機能の組み合わせが不正である  
IFPARMER = 804 (0x0324)  
意味 : パラメータの指定に誤りがある  
IFERRFIL = 1005 (0x03ed)  
意味 : エラーが解決していないファイルにアクセス要求を行った

## ERASE\_SQ関数

IFAS順編成ファイルを削除します。

### 書式

---

```
DWORD    ifas_erase_sq(
          CHAR    *filename_s
          )
```

### パラメータ

---

filename\_r      ファイル名文字列へのポインタ

### 戻り値

---

ステータスコード

### 規則

---

- (1) 存在しないファイルのファイル名を指定するとエラーとなります。
- (2) アクセス処理中のファイルをファイル名に指定するとエラーとなります。
- (3) BUILD\_IDXFILE関数でインデックスファイルを追加作成したファイルに対し本関数を実行すると“IFFORGER”を返却します。索引順編成ファイルとしてRENAME関数を実行してください。

### 主なステータスコード

---

IFNORMAL = 0    (0x0000)

意 味    : 処理は正常に終了した

IFNFNDFN = 201 (0x00c9)

意 味    : 指定したファイル名のファイルが存在しない

IFFLBUSY = 205 (0x00cd)

意 味    : このファイルは現在他の利用者が使用中である

IFFORGER = 214 (0x00d6)

意 味    : ファイル編成が誤っている

IFFNAMER = 805 (0x0325)

意 味    : ファイル名の指定が不正である

## RENAME\_SQ関数

IFAS順編成ファイルのファイル名を変更します。

### 書式

---

```
DWORD    ifas_rename_sq(
          CHAR    *old_filename_s,
          CHAR    *new_filename_s
          )
```

### パラメータ

---

old_filename_s	旧ファイル名文字列へのポインタ
new_filename_s	新ファイル名文字列へのポインタ

### 戻り値

---

ステータスコード

### 規則

---

- (1) 存在しないファイルのファイル名を旧ファイル名に指定するとエラーとなります。
- (2) 既に存在するファイルのファイル名を新ファイル名に指定するとエラーとなります。
- (3) アクセス処理中のファイルを旧ファイル名に指定するとエラーとなります。
- (4) BUILD\_IDXFILE関数でインデックスファイルを追加作成したファイルに対し本関数を実行すると“IFFORGER”を返却します。索引順編成ファイルとしてRENAME関数を実行してください。

### 主なステータスコード

---

IFNORMAL = 0	(0x0000)	意味 : 処理は正常に終了した
IFALDYFN = 101	(0x0065)	意味 : 指定したファイル名のファイルは既に存在する
IFNFNDFN = 201	(0x00c9)	意味 : 指定したファイル名のファイルが存在しない
IFFLBUSY = 205	(0x00cd)	意味 : このファイルは現在他の利用者が使用中である
IFFORGER = 214	(0x00d6)	意味 : ファイル編成が誤っている
IFFNAMER = 805	(0x0325)	意味 : ファイル名の指定が不正である

## GETINFO\_SQ関数

指定したIFAS順編成ファイルの情報を取得します。

### 書式

```
DWORD    ifas_getinfo_sq(
HANDLE    isfd,
LPVOID    buffer
)
```

### パラメータ

**isfd**            OPEN関数で返却するファイル記述子  
**buffer**          返却する情報格納領域のポインタ

〈ファイル情報構造体: **struct seqfinfo**〉

データ型	構造体メンバ名	内 容
DWORD	sq_recsz	データレコード長
DWORD	sq_nrecords	レコード数

### 戻り値

ステータスコード

### 規則

- (1) オープンしていないファイルに対して本関数を実行するとエラーとなります。
- (2) 索引順編成ファイルのデータファイルや、相対編成ファイルとして作成したファイルを順編成ファイルとしてオープンして本関数を実行した場合は、最大レコード番号をレコード数として返却します。

### 主なステータスコード

IFNORMAL = 0 (0x0000)

意 味 : 処理は正常に終了した

IFBADHND = 211 (0x00d3)

意 味 : 指定したファイルはオープンされていない

IFFORGER = 214 (0x00d6)

意 味 : ファイル編成が誤っている

IFERRFIL = 1005 (0x03ed)

意 味 : エラーが解決していないファイルにアクセス要求を行った

## BUILD\_IDXFILE\_SQ関数

作成済みのIFAS順編成ファイルに対してインデックスファイルを追加作成し、IFAS 索引順編成ファイルとして扱えるようにします。

### 書式

```
DWORD    ifas_build_idxfile_sq(
        CHAR                *filename_s,
        CHAR                *filename_i,
        struct ifkeydesc    *keydesc
    )
```

### パラメータ

filename_s	作成済み順編成ファイル名文字列へのポインタ
filename_i	新規作成インデックスファイル名文字列へのポインタ
keydesc	プライマリキーにおけるキー記述構造体のポインタ

〈キー記述構造体: struct ifkeydesc〉

データ型	構造体メンバ名	内 容	既定値
DWORD	ifk_number	キーパート数 (0=インデックスを作成しない)	1
DWORD	ifk_dupflag	重複キー許可 (0=重複不許可: 0以外はエラー)	0
struct ifkeypart	k_prt[n]	キーパート構造体 (キーパートは最大16個まで指定可能)	

〈キーパート構造体: struct ifkeypart〉

データ型	構造体メンバ名	内 容	既定値
DWORD	ifk_key_offset	レコード内キーオフセット	0
DWORD	ifk_key_size	キーサイズ	0
DWORD	ifk_type	キータイプ:	IFCHAR
		IFCHAR    // 文字	
		IFBIN4    // DWORD (4バイト)	
		IFBIN2    // WORD (2バイト)	
		IFSBIN4   // LONG (4バイト)	
		IFSBIN2   // INT (2バイト)	
		IFFLOAT   // FLOAT (4バイト)	
		IFDOUBLE   // DOUBLE (8バイト)	

### 戻り値

ステータスコード

### 規則

- (1) 存在しない順編成ファイル名を指定するとエラーとなります。
- (2) 既にインデックスファイルが作られている順／相対編成ファイル、または作成済みの索引順編成ファイルのデータファイルに対し、新規に別のインデックスファイルを作成すると従来のインデックスファイルは使用できなくなります。
- (3) 本関数によって設定可能なインデックスはプライマリインデックスのみです。サブインデックスを作

成する場合は、索引順編成ファイルとしてオープンしADDINDEX関数を用います。

- (4) インデックスファイル作成後でもデータファイルは順／相対編成ファイルとして扱うことができます。また、最初から索引順編成ファイルとして作成したファイルのデータファイルを順／相対編成ファイルとして扱うこともできます。ただし、相対編成ファイルとしてレコードの追加／削除／更新処理、順編成ファイルとして追加／更新処理を行った後、再び索引順編成ファイルとして使用する場合は、REBUILD\_IDXFILE関数によってインデックスファイルを再作成する必要があります。
- (5) 本関数が処理途中でエラーを検出して失敗した場合は、追加作成しようとしたインデックスファイルは消去します。

### 主なステータスコード

---

IFNORMAL = 0 (0x0000)

意味 : 処理は正常に終了した

IFNFNDFN = 201 (0x00c9)

意味 : 指定したファイル名のファイルが存在しない

IFFLBUSY = 205 (0x00cd)

意味 : このファイルは現在他の利用者が使用中である

IFDUPKEY = 402 (0x0192)

意味 : プライマリインデックスで重複キーを検出した

IFKDSCER = 802 (0x0322)

意味 : キー記述構造体が不正である

IFFNAMER = 805 (0x0325)

意味 : ファイル名の指定が不正である

IFDRDBER = 1101 (0x044d)

意味 : ファイル識別子管理ファイルのアクセスエラーが発生した

## 3-2-4 共通アクセス関数

IFAS FILEの全ファイル編成共通のサービス関数を紹介します。

<code>ifas_flush</code>	バッファの内容をファイルに書き出す
<code>ifas_getrecno</code>	直前に処理したレコードのレコード番号を取得する
<code>ifas_stopend</code>	IFAS FILEの処理を終了する
<code>ifas_wrapup</code>	異常終了時の後処理を行う

## FLUSH関数

指定したIFAS ファイルの更新されたバッファをすべてディスクに書き出します。

### 書式

---

```
DWORD    ifas_flush(
HANDLE    isfd
)
```

### パラメータ

---

isfd                      OPEN関数で返却するファイル記述子

### 戻り値

---

ステータスコード

### 規則

---

- (1) オープンしていないファイルに対して本関数を実行するとエラーとなります。
- (2) 入力モードでオープンされたファイルに対して本関数を実行するとエラーとなります。

### 主なステータスコード

---

IFNORMAL = 0    (0x0000)

意味 : 処理は正常に終了した

IFBADHND = 211 (0x00d3)

意味 : 指定したファイルはオープンされていない

IFOPMDER = 801 (0x0321)

意味 : オープンアクセスモードと機能の組み合わせが不正である

IFERRFIL = 1005 (0x03ed)

意味 : エラーが解決していないファイルにアクセス要求を行った



## GETRECNO関数

直前に行ったレコード処理のレコード番号を取得します。

### 書式

---

```
DWORD    ifas_getrecno(  
HANDLE    isfd,  
LPDWORD   recnum  
)
```

### パラメータ

---

isfd	OPEN関数で返却するファイル記述子
recnum	レコード番号を返却する領域のポインタ

### 戻り値

---

ステータスコード

### 規則

---

- (1) オープンしていないファイルに対して本関数を実行するとエラーとなります。
- (2) 本関数で返却するレコード番号はカレントレコードとは異なります。たとえば、DELETE関数を実行した後では、削除したレコード番号を返却します。
- (3) 本関数実行前にレコード操作を実行していない場合はrecnumには0を返却します。

### 主なステータスコード

---

IFNORMAL = 0 (0x0000)

意味 : 処理は正常に終了した

IFBADHND = 211 (0x00d3)

意味 : 指定したファイルはオープンされていない

IFERRFIL = 1005 (0x03ed)

意味 : エラーが解決していないファイルにアクセス要求を行った

## STOPEND関数

スレッド内におけるIFAS ファイルに対する処理をすべて終了します。

### 書式

---

```
DWORD    ifas_stopend(  
    )
```

### パラメータ

---

なし

### 戻り値

---

ステータスコード

### 規則

---

- (1) 本関数を実行すると、同じスレッド内でオープン中のすべてのファイルをクローズし、IFAS FILE 内部処理用のメモリをすべて解放します。

## WRAPUP関数

プログラム異常終了時のIFAS FILEの後処理を行います。

### 書式

---

```
DWORD    ifas_wrapup(  
    )
```

### パラメータ

---

なし

### 戻り値

---

ステータスコード

### 規則

---

- (1) 本関数は、利用者がプログラム異常終了時の例外処理として呼び出す関数で、IFAS FILEは本関数を実行したスレッドに対して以下のような後処理を行います。
  - ・トランザクション処理中の場合はロールバック処理を行います。
  - ・オープン中のすべてのファイルをクローズします。
  - ・更新ログファイルのクローズを行います。
  - ・IFAS FILE内部処理用のメモリをすべて解放します。

## 3-2-5 トランザクション関数

IFAS FILEのトランザクション処理機能を実現するための関数を紹介します。

<code>ifas_logopen</code>	更新ログ収集を開始する
<code>ifas_logclose</code>	更新ログ収集を終了する
<code>ifas_txstart</code>	トランザクション処理を開始する
<code>ifas_commit</code>	トランザクション処理を終了する
<code>ifas_rollback</code>	トランザクション処理を取り消す

## LOGOPEN関数

IFAS FILEの更新ログファイルをオープンし、更新ログ収集を開始します。

### 書式

---

```
DWORD    ifas_logopen(  
          CHAR    *logfile  
          )
```

### パラメータ

---

logfile      更新ログファイル名文字列へのポインタ

### 戻り値

---

ステータスコード

### 規則

---

- (1) 存在しない更新ログファイルのファイル名を指定した場合は新規に作成します。
- (2) 既に存在する更新ログファイルのファイル名を指定した場合は追加モードでログ収集します。
- (3) 1つの利用者プログラムが2つ以上の更新ログファイルをオープンしようするとエラーとなります。

### 主なステータスコード

---

IFNORMAL = 0    (0x0000)  
意 味    : 処理は正常に終了した  
IFALDYLG = 701 (0x02bd)  
意 味    : 更新ログファイルは既にオープンされている  
IFLGIOER = 1701 (0x06a5)  
意 味    : 更新ログファイルのI/Oエラーが発生した  
IFLGMMER = 1702 (0x06a6)  
意 味    : ログマネージャでメモリの確保に失敗した  
IFLGFNER = 1703 (0x06a7)  
意 味    : 更新ログファイル名が不正である

## LOGCLOSE関数

IFAS FILEの更新ログファイルをクローズし、更新ログ収集を終了します。

### 書式

---

```
DWORD    ifas_logclose(  
    )
```

### パラメータ

---

なし

### 戻り値

---

ステータスコード

### 規則

---

- (1) トランザクション処理を終了していない状態で本関数を実行するとエラーとなります。

### 主なステータスコード

---

IFNORMAL = 0 (0x0000)

意 味 : 処理は正常に終了した

IFNOPNLG = 702 (0x0702)

意 味 : 更新ログファイルがオープンされていない

IFCLTGCL = 705 (0x0705)

意 味 : 更新ログ収集対象のファイルがオープン中である

IFLGIOER = 1701 (0x06a5)

意 味 : 更新ログファイルのI/Oエラーが発生した

## TXSTART関数

トランザクション処理の開始位置を設定します。

### 書式

---

```
DWORD    ifas_txstart(  
    )
```

### パラメータ

---

なし

### 戻り値

---

ステータスコード

### 規則

---

- (1) 本関数はトランザクション処理毎の起点を示す関数であり、ROLLBACK関数でトランザクションをキャンセルする場合の戻り点を示します。
- (2) トランザクション処理を終了していない状態で本関数を実行するとエラーとなります。
- (3) IFROLLBACK指定でオープンしたファイルの更新を行う場合は、あらかじめ本関数を実行しておかなければなりません。

### 主なステータスコード

---

IFNORMAL = 0 (0x0000)

意味 : 処理は正常に終了した

IFNENDTX = 703 (0x02bf)

意味 : トランザクション処理が終了していない

## COMMIT関数

トランザクション処理を終了します。

### 書式

---

```
DWORD    ifas_commit(  
    )
```

### パラメータ

---

なし

### 戻り値

---

ステータスコード

### 規則

---

- (1) 本関数を実行すると、すべてのファイルのレコードロックは解除され、更新のあったすべてのファイルのバッファは実際のファイルに書き出されます。

### 主なステータスコード

---

IFNORMAL = 0 (0x0000)

意 味 : 処理は正常に終了した

IFLRCERR = 1202 (0x04b2)

意 味 : ロック／アンロック処理で回復不可能なエラーが発生した



## ROLLBACK関数

トランザクション処理を取り消し、ファイルの状態をトランザクションスタート時点の状態に戻します。

### 書式

---

```
DWORD    ifas_rollback(  
    )
```

### パラメータ

---

なし

### 戻り値

---

ステータスコード

### 規則

---

- (1) 本関数を実行すると、すべてのファイルのレコードロックは解除され、更新のあったすべてのファイルは直前のTXSTART時点の状態に戻されます。
- (2) 本関数を実行するとすべてのファイルのカレントレコードはキャンセルされるため、レコードの再位置付けを行う必要があります。
- (3) IFROLLBACK指定のファイルがオープンされていない状態で本関数を実行するとエラーとなります。

### 主なステータスコード

---

IFNORMAL = 0 (0x0000)

意味 : 処理は正常に終了した

IFTXRBFL = 401 (0x0191)

意味 : TXSTART関数が実行されていない

IFLRCERR = 1202 (0x04b2)

意味 : ロック／アンロック処理で回復不可能なエラーが発生した

IFRBCLER = 1801 (0x0709)

意味 : ロールバックログ収集でエラーが発生した

IFRBMGER = 1802 (0x01802)

意味 : ロールバックマネージャでエラーが発生した

## **3-3 注意事項**

### **32/64ビットAP作成時のリンクモジュールについて**

- (1) 32ビットAPを作成する場合は、IFAS FILEインストールディレクトリ¥BIN配下の”IFASAPI.DLL”をリンクして作成してください。
- (2) 64ビットAPを作成する場合は、IFAS FILEインストールディレクトリ¥BIN配下の”IFASAPI64.DLL”をリンクして作成してください。

### **C言語のコンパイルオプションについて**

- (1) IFAS FILEは内部でマルチスレッドプログラムとして動作することがあります。C言語のプログラムを作成する場合、コンパイルオプション「/MD」を設定し、マルチスレッド用ランタイムルーチンが使用されるようにしてください。
- (2) 32ビットAPを作成する場合は、IFAS FILEのC言語関数の呼び出し規約は「\_cdecl」を設定してください。

### **マルチスレッドプログラミングについて**

マルチスレッドプログラミングを行う場合、IFAS FILEでは下記の考慮が必要になります。

- (1) IFAS ファイルアクセスを行うスレッドは、ifas\_stopend関数を呼び出してからスレッドを終了させてください。ifas\_stopend関数を呼び出すことで、IFAS FILEがスレッド毎に確保した資源の解放が行われます。
- (2) 生成するスレッド数によっては、環境設定ファイル[Control]セクションのThreadMax、ServerHostMaxの指定が必要になります。環境設定ファイルについては、「2-3 環境設定ファイル」を参照してください。

### **一般的な注意事項**

ここでは、マルチスレッドプログラムを行う上での一般的な注意事項を記述しています。

- (1) スレッドを生成/終了する場合、ランタイムルーチンの\_beginthread/\_beginthreadex/\_endthread/\_endthreadex関数を使用してください。CreateThread/ExitThreadを使用した場合、C/C++ランタイムルーチンで確保された作業領域が解放されないため、メモリリークが発生することがあります。
- (2) プロセス/スレッドを終了させる場合、TerminateProcess/TerminateThread関数の使用は避けてください。内部でクリティカルセクション等の同期資源をつかんだままスレッドが終了してしまい、プロセスがストールすることがあります。

## 4. COBOLプログラムからの利用

COBOLからIFAS一般ファイルやIFAS RDB表を索引順編成ファイル、相対編成ファイル、順編成ファイルとしてアクセスする機能を提供します。

COBOLファイルマッパーの指示によって、COBOLからIFAS一般ファイル、IFAS RDB表をアクセスできるようになっています。COBOLファイルマッパーの使用方法については、使用されるIFAS FILE、IFASPRO RDBのバージョンと対応したCOBOL言語説明書およびCOBOLプログラミング手引書を参照してください。本章ではCOBOLプログラムから利用できる、IFAS ファイルに対するアクセス機能、排他／同時アクセス制御、およびトランザクション処理機能について説明します。

### 4-1 機能概要

#### 4-1-1 諸元

IFAS一般ファイルおよびIFAS RDB表はそれぞれ下表のような諸元となっています。

	IFAS一般ファイル	IFAS RDB表
最大レコードサイズ	64K バイト	32K-1 バイト
最大キーサイズ	256 バイト	255バイト
最大複数索引数	255 個	
集合／合成キーの最大キーパート数 (1つの索引に指定できる項目数)	16 個	
	(ただし、合計キーサイズが 256バイトまで)	(ただし、合計キーサイズが 255バイトまで)
最大列数(項目数)	—	2000 列

##### 一般索引順編成ファイルのファイル名について

一般索引順編成ファイルは、指定されたファイル名がデータファイル名となり、データファイル名に拡張子“.idx”を付加したものがインデックスファイル名となります。データファイル名に拡張子が付加されている場合は、その拡張子を“.idx”に変更し、インデックスファイル名とします。これは、IFAS FILE システムが指定されたファイル名を自動的に変換しています。したがって、一般索引順編成ファイルのファイル名に拡張子“.idx”は指定できません。

#### 4-1-2 データアクセス機能

##### (1) 一般ファイルのアクセス

COBOLから以下の3種類のファイル編成が使用可能です。各ファイルをアクセスする場合の動作はCOBOLのファイルアクセス命令の仕様に従っています。詳細はCOBOL言語説明書を参照してください。

- ・ 一般索引順編成ファイル
- ・ 一般相対編成ファイル
- ・ 一般順編成ファイル

## (2) RDB表のアクセス

COBOLからIFAS RDB表をアクセスする場合、以下の3種類のファイル編成として使用することが可能です。各ファイルをアクセスする場合の動作はCOBOLのファイルアクセス命令の仕様に従っています。詳細はCOBOL言語説明書を参照してください。

- ・ RDB索引順編成ファイル
- ・ RDB相対編成ファイル
- ・ RDB順編成ファイル

## (3) トランザクション機能

IFAS ファイルの場合、IFAS RDB表だけでなく、一般ファイルでもトランザクション機能を利用することができます。この機能を使用するにはCOBOLファイルマッパーでオプションを指定します。IFAS RDB表の場合は表定義ユーティリティで宣言することもできます。IFAS RDB表の指定は、表定義ユーティリティで行うことを推奨します。

トランザクション処理の指定を行ったIFAS ファイルには、COMMIT/ROLLBACK命令が有効となります。更新したレコードはレコード排他制御が働き、COMMITまたはROLLBACK命令まで保持されます。また、不意の電源ダウンなどが発生した場合、システム再起動時に更新処理途中であったトランザクションを自動的にロールバックするリカバリ機能を使用することができます。詳しくは「6-4-2 システム障害からの復旧」を参照してください。

COBOL言語説明書には、データベース機能としてCOMMIT/ROLLBACK命令が説明されていますので、IFAS一般ファイルに対してトランザクション処理を行いたい場合は、データベース機能に記載されている各命令の説明を確認してください。

## (4) 更新ログ収集機能

COBOLファイルマッパーでオプションを指定することにより、更新ログを収集することができます。IFAS RDB表の場合は表定義ユーティリティで宣言することもできます。IFAS RDB表の指定は、表定義ユーティリティで行うことを推奨します。

更新ログの収集を行っていれば、ディスク障害などでIFAS ファイルが使用できなくなってしまった場合、バックアップファイルと更新ログファイルから最新の状態に復旧すること(ロールフォワード)が可能です。

## 4-1-3 OPEN命令のモードと共用制御

IFAS ファイルをCOBOLからアクセスする場合のOS互換モードによる共用制御を以下の表に示します。

### 表における記号の説明

E	EXCLUSIVE	RD	READ命令
P	PROTECTED	ST	START命令
S	SHARED	—	その他の命令 (READ WITH UNLOCKを含む)
◎	ファイルは共用されレコード排他が働く <sup>注1</sup>		
○	ファイルは共用されるがレコード排他は働かない <sup>注1</sup>		
●	ファイルは共用されレコード排他が働くが、レコード排他が働く機会はない <sup>注1</sup>		
空欄	ファイルは共用されない		

注1. 本共用制御の説明はトランザクション処理を行わない場合に適用されます。トランザクション処理を行っている場合、更新系の命令 (WRITE, REWRITE, DELETE) が行われたレコードは COMMIT/ROLLBACK命令またはCLOSE命令を行うまでレコード排他が働きます。

### OPEN命令のモードと共用制御 (A-VX II 互換モード)

先行 後行		入力 モード					入出力 モード								出力 モード			拡張 モード		
		E	P	S			E	P			S			E	P	S	E	P	S	
				—	RD	ST		—	RD	ST	—	RD	ST							
入力 モード	E																			
	P		○	○	○	○														
	S		○	○	○	○		○	◎	◎	○	◎	◎					●	●	
入出力 モード	E																			
	P			○	注2	注4														
	S			○							○	◎	◎							●
出力 モード	E																			
	P																			
	S																			
拡張 モード	E																			
	P			○	注3	注5														
	S			○							○	●	●							●

注2. 既定値では○ (レコード排他は働かない) 動作ですが、運用支援ツールのCOBOL設定において「INPUTモードのREAD命令でレコードロックする」とした場合、◎ (レコード排他が働く) の動作になります。後者の方がA-VX IIとの互換性が高くなります。

注3. 既定値では○ (レコード排他は働かない) 動作ですが、運用支援ツールのCOBOL設定において「INPUTモードのREAD命令でレコードロックする」とした場合、● (レコード排他が働くが、レコード排他が働く機会はない) の動作になります。後者の方がA-VX IIとの互換性が高くなります。

注4. 既定値では○ (レコード排他は働かない) 動作ですが、運用支援ツールのCOBOL設定において「INPUTモードのSTART命令でレコードロックする」とした場合、◎ (レコード排他が働く) の動作になります。後者の方がA-VX IIとの互換性が高くなります。

注5. 既定値では○ (レコード排他は働かない) 動作ですが、運用支援ツールのCOBOL設定において「INPUTモードのSTART命令でレコードロックする」とした場合、● (レコード排他が働くが、レコード排他が働く機会はない) の動作になります。後者の方がA-VX IIとの互換性が高くなります。

## OPEN命令のモードと共用制御（A-V X 5 互換モード）

先行 後行		入力 モード					入出力 モード								出力 モード			拡張 モード		
		E	P	S			E	P			S			E	P	S	E	P	S	
				—	RD	ST		—	RD	ST	—	RD	ST							
入力 モード	E																			
	P		○	○	○	○														
	S		○	○	○	○		○	◎	○	○	◎	○					●	●	
入出力 モード	E																			
	P			○	注 6	○														
	S			○		○					○	◎	○						●	
出力 モード	E																			
	P																			
	S																			
拡張 モード	E																			
	P			○	注 7	○														
	S			○		○					○	●	○						●	

注6. 既定値では○（レコード排他は働かない）動作ですが、運用支援ツールのCOBOL設定において「INPUTモードのREAD命令でレコードロックする」とした場合、◎（レコード排他が働く）の動作になります。

注7. 既定値では○（レコード排他は働かない）動作ですが、運用支援ツールのCOBOL設定において「INPUTモードのREAD命令でレコードロックする」とした場合、●（レコード排他が働くが、レコード排他が働く機会はない）の動作になります。

## 4-2 ファイル情報と動作環境の設定

本説では、COBOLプログラムからIFAS ファイルを使用する場合のファイル情報、および、各種動作環境の設定について説明します。

### 4-2-1 ファイル情報の設定

COBOLプログラムからIFAS ファイルを使用する場合、COBOLファイルマッパーという外部ツールで、COBOLソース上に指定した内部ファイルIDと、実ファイル名／表名を関連付けます。COBOLファイルマッパーの使用方法については、使用されるIFAS FILEおよびIFASPRO RDBのバージョンと対応したCOBOLのマニュアルを参照してください。

#### (1) IFAS一般ファイルの指定形式

内部ファイルID    COBOLソース上の内部ファイルID  
 デバイス            指定なし  
 ファイル種別        IFAS FILE  
 実ファイル名        [<サーバ名>]ファイル名[, オプション[, オプション]…]  
                          <サーバ名>は、クライアント・サーバ機能を使用する時に指定します(「6-1 クラ  
                          イアント・サーバ機能」参照)。<サーバ名>を省略した場合は、ローカルマシン  
                          上のファイルが対象となります。  
                          ファイル名はファイルパスを指定してください。

#### (2) IFAS RDB表の指定形式

内部ファイルID    COBOLソース上の内部ファイルID  
 デバイス            RDB  
 ファイル種別        IFASPRO RDB  
 実ファイル名        [<サーバ名>][データベース名.]表名[, オプション[, オプション]…]<サーバ  
                          名>は、クライアント・サーバ機能を使用する時に指定します(「6-1 クライアント・  
                          サーバ機能」参照)。<サーバ名>を省略した場合は、ローカルマシン上の表が  
                          対象となります。  
                          データベース名を省略した場合は、データベースの既定値が有効になります。

#### (3) オプションの種類

オプション種別	オプション値	オプションの説明
トランザクション処理	TRANSACTION または TRAN	トランザクション処理する
	NOTTRANSACTION または NOTTRAN	トランザクション処理しない
	なし	デフォルト値 (NOTTRANSACTION)
更新ログ	LOG	更新ログを保存する
	NOLOG	更新ログを保存しない
	なし	デフォルト値 (NOLOG)
バッファ数	NBUF:nnn	バッファ数を指定する nnn は、10～500 までの数字 1 バッファには 1 レコード分セットされます

	なし	デフォルト値(NBUF:010)
インデックスブロック サイズ	IDXBLKSIZ:nn	インデックスブロックサイズ (K バイト) を指定する nn は、1～63 までの数字
	なし	デフォルト値(NBUF:1=1024 バイト)
一般ファイルの REWRITE /DELETE	RWTDEL CUR	一般索引順編成ファイル (乱/動アクセス) の REWRITE/DELETE は、「直前の命令が READ 命令で、主キー値の変更がない」という条件を満たしていれば、読み込んだレコードが REWRITE/DELETE の対象レコードとなる。なお、条件を満たしていない場合は、RWTDEL KEY 指定時と同じである。
	RWTDEL KEY	一般索引順編成ファイル (乱/動アクセス) の REWRITE/DELETE 命令は、キー値に基づいてレコードの更新を行う。二重キーが存在する場合は、最初のレコードが対象になる。
	なし	デフォルト値 (運用支援ツールの COBOL 設定に従う)
二重キー検出 (更新系)	ALTDUPKEY	重複許可インデックスに対する WRITE/REWRITE 命令で二重キーを検出する
	NOALTDUPKEY	重複許可インデックスに対する WRITE/REWRITE 命令で二重キーを検出しない
	なし	デフォルト値 (運用支援ツールの COBOL 設定に従う)
二重キー検出 (参照系)	READ02	重複許可インデックスに対する READ 命令で二重キーを検出する
	NOREAD02	重複許可インデックスに対する READ 命令で二重キーを検出しない
	なし	デフォルト値 (運用支援ツールの COBOL 設定に従う)
OUTPUT モードオープン	OUTPUT CRE	OUTPUT モードオープン時にファイルの属性も含めて初期化する。
	OUTPUT EX	OUTPUT モードオープン時にファイル属性は既存のままにしてレコードを 0 件化する。
	なし	デフォルト値 (OUTPUT CRE)

#### (4) オプション指定時の注意

- IFAS RDB表のトランザクション処理オプションおよび更新ログオプションの指定は、表定義ユーティリティで行うことを推奨します。これはDBリンクキット利用のアプリケーションからはオプション指定ができないためです。
- 環境設定ファイルでもファイル名(表名)の変更やオプションの設定は可能ですが、環境設定ファイルとCOBOLファイルマッパーの併用は、誤動作の原因ともなりますので行わないでください。



## (5) ファイルマッパー指定例

### 例1 ローカルマシン上のIFAS一般ファイルをアクセス

COBOLファイルマッパーにて、COBOLアプリケーションがアクセスするIFAS一般ファイル名を指定します。

COBOLソース上のファイル名の記述

```
FD F0001
  LABEL RECORD IS STANDARD
  VALUE OF IDENTIFICATION IS "FILE01".
```

COBOLファイルマッパーでの指定

内部ファイルID FILE01

デバイス 指定なし

ファイル種別 IFAS FILE

「IFAS FILE」を指定することでIFAS一般ファイルとしてアクセス可能になります。

実ファイル名 D:¥LOCAL¥FILE01,TRANSACTION,LOG

ソース上「FILE01」と記述されている内部ファイルIDは、

「D:¥LOCAL¥FILE01」に変換されます。

また、トランザクション処理を行い、更新ログを保存します。

### 例2 クライアントマシンからサーバマシン上のIFAS RDB表をアクセス

COBOLファイルマッパーにて、COBOLアプリケーションが接続するサーバ(IFASPRO RDBのサーバ)とIFAS RDB表名を指定します。

COBOLソース上の表名の記述

```
FD F0002
  LABEL RECORD IS STANDARD
  VALUE OF IDENTIFICATION IS "DBTBL02".
```

COBOLファイルマッパーでの指定

内部ファイルID DBTBL02

デバイス RDB

ファイル種別 IFASPRO RDB

「IFASPRO RDB」を指定することでIFAS RDB表としてアクセス可能になります。

実ファイル名 <HOST01>DB02.DBTBL02

サーバ名「<HOST01>」を指定することでサーバ(HOST01)上のIFAS RDB表がアクセス可能になります。

ソース上「DBTBL02」と記述されている内部ファイルIDは、

「<HOST01>DB02.DBTBL02」に変換されます。

## (6) 表オプションの優先順位

IFAS RDB表の場合、表オプションとCOBOLファイルマッパーのオプションとの間に、以下の優先順位があります。

順位 高

COBOL ファイルマッパーのオプション

↓

表定義ユーティリティ 表単位の指定

順位 低

## (7) 環境設定ファイル

COBOLプログラムや、カスタムコントロール、および、その他のCOBOL関連製品では、環境設定ファイルを用いて、アプリケーション毎に各種オプションを変更することができます。環境設定ファイルについては、「2-3 環境設定ファイル」を参照してください。

## 4-2-2 動作環境の設定

運用支援ツールの[COBOL設定]タブで、COBOLプログラムの動作環境を設定することができます(運用支援ツールについては「7-4-1 運用支援ツール」を参照)。

なお、[COBOL設定]タブの各項目は、COBOL関連製品での共通の設定になっています。[COBOL設定]タブを変更した場合、カスタムコントロールや、他の製品(DBリンクキット等)の動作にも影響があるのでご注意ください。

以下に、動作環境の各項目について説明します。

### (1) OS互換モード

[OS互換モード]ラジオボタンによって動作モードをA-VXⅡ互換とするかA-VX5互換とするかを選択できます。

下表にOS互換モードによる差分動作を示します。

OS互換モードによる差分動作

差分動作	A-VXⅡ	A-VX5
選択型仮想表を OUTPUT/EXTEND モードでオープン時	“00”(正常)	“37”エラー
結合型仮想表を OUTPUT/EXTEND モードでオープン時	実行時エラー <sup>注3</sup>	“37”エラー
結合型仮想表を I/O モードでオープン時	実行時エラー <sup>注3</sup>	“00”(正常) <sup>注5</sup>
ファイルビジー時	ファイルビジー解除を待つ <sup>注4</sup>	“91”エラー
OPEN 命令直後の READ(PRIOR)命令時	“10”(AT END)	先頭レコードを読む
SELECT 命令直後の READ(PRIOR)命令時 <sup>注1</sup>	最終レコードを読む	先頭レコードを読む
SELECT T 命令直後の START (PRIOR)命令時 <sup>注1</sup>	最終から 2 番目のレコードに位置付ける	“23”エラー
START 命令時	レコードロックする	レコードロックしない
REWRITE(KEY) 命令で同一キーのレコードが 2 件以上存在時 <sup>注2</sup>	同一キーの最初のレコードを REWRITE する	“22”エラー
DELETE(KEY) 命令で同一キーのレコードが 2 件以上存在時 <sup>注2</sup>	同一キーの最初のレコードを DELETE する	“22”エラー
READ(KEY) 命令で指定キーのインデックスが存在しない時	実行時エラー <sup>注3</sup>	“92”エラー
START(KEY) 命令で指定キーのインデックスが存在しない時	実行時エラー <sup>注3</sup>	“92”エラー
出力モードで OPEN したファイルに SELECT 命令を行ったとき <sup>注1</sup>	“47”エラー	“92”エラー
SELECT 命令を行わずに SCRATCH 命令を行ったとき <sup>注1</sup>	“00”(正常)	“92”エラー
相対編成ファイルに対して、START(1 KEY IS >= 0 / KEY IS > 0) 命令を行った時	先頭レコードに位置付ける	“23”エラー
デッドロック発生時	“9J”エラー	実行時エラー <sup>注3</sup>

ロールバックログ収集エラー時	“9K”エラー	実行時エラー <sup>注3</sup>
ログ収集エラー時	“9T”エラー	実行時エラー <sup>注3</sup>
更新ログファイルオーバーフロー時	“9O”エラー	実行時エラー <sup>注3</sup>
RDB 破壊検出時	“9M”エラー	実行時エラー <sup>注3</sup>
ロックテーブルオーバーフロー時	“9P”エラー	実行時エラー <sup>注3</sup>

- 注1 IFAS一般ファイルに対する、SELECT／SCRATCH命令は内部的に実装している機能であり、COBOLからは利用できません。
- 注2 COBOLからの使用で、IFAS一般ファイルに対するREWRITE(KEY)／DELETE(KEY)命令は、相対編成／索引順編成ファイルを乱／動呼び出した時のREWRITE／DELETE命令になります。
- 注3 COBOLアプリケーションは実行時エラーとなります(関連製品エラー)。カスタムコントロールや、その他のCOBOL関連製品ではエラーコードが異なります。詳細は各製品の説明書を参照してください。
- 注4 COBOLの指定によっては“91”エラーが返却される場合もあります。
- 注5 更新系の命令(WRITE／REWRITE／DELETE)はエラーになります。

## (2) OPEN命令時のレコードサイズ

「OPEN命令でレコードサイズのチェックをしない」チェックボックスによって、OPEN命令時のレコードサイズ(COBOLの場合はファイル記述項の後ろに記載されているレコード記述項により決定)チェックの緩和を指定できます。

チェックボックスをチェックしていない場合、レコードサイズとIFAS ファイルのレコードサイズは一致していなければなりません(一致していない場合はエラーとなります)。

チェックボックスをチェックすると、レコードサイズはIFAS ファイルのレコードサイズ以上の値を指定すればOPEN命令が実行されます(小さい値を指定した場合はエラーとなります)。

## (3) 二重キー検出動作

[READ命令で二重キーを検出しない]および[WRITE／REWRITE命令で二重キーを検出しない]チェックボックスによって、二重キーを検出しないようにできます。

チェックボックスをチェックすると、IFAS一般索引順編成ファイル／IFAS RDB索引順編成の、重複キー許可インデックスに対するREAD／WRITE／REWRITE命令で二重キーが発生しても、入出力状態に二重キー(02)ステータスが返却されないで正常(00)ステータスが返却されるようになります。

なお、重複キー不許可インデックスに対するWRITE／REWRITE命令では、本オプションによらず二重キーエラー(22)が返却されます。

## (4) 二重キー検出範囲

[REWRITE命令で変更するキーのみ二重キーを検出する]チェックボックスによって、REWRITE命令での重複キー許可インデックスへの二重キー検出範囲を指定することができます。

チェックボックスをチェックすると、IFAS一般索引順編成ファイル／IFAS RDB索引順編成の重複キー許可インデックスに対するREWRITE命令で、データの変更があったキーに対してのみ二重キーチェックを行います。この方がA-VX IIとの互換性が高くなります。

チェックボックスをチェックしないと、すべてのキーに対して二重キーチェックを行います(データの変更がないキーも含まれます)。

どちらの場合も、二重キーが検出された場合は、二重キー(02)ステータスが返却されます。

## (5) 共用制御動作

[INPUTモードのREAD命令でレコードロックする]および[INPUTモードのSTART命令でレコードロックする]チェックボックスによって、INPUTモードでレコード排他が働くようにすることができます。

チェックボックスをチェックすると、INPUTモードのREAD命令やSTART命令が実行されたレコードは次の命令までレコード排他が働くようになります。この方がA-VX IIとの互換性が高くなります。

## (6) 索引順編成ファイルのREWRITE/DELETE

[索引順編成ファイルのREWRITE/DELETEをカレントレコード対象とする]チェックボックスによって、IFAS一般索引順編成ファイル(動/乱呼び出し)のREWRITE/DELETE命令で処理対象となるレコードを選択できます。

チェックボックスをチェックしないと、指定されたキー値に基づいてレコードのREWRITE/DELETE命令を実行します。二重キーが存在する場合は最初のレコードが対象になります(下記例では、レコード1が処理対象)。なお、A-VX5互換モードの場合は”22”エラーになります。

チェックボックスをチェックすると、下記の条件を両方満たしている時、先のREAD命令で読み込んだレコードがREWRITE/DELETE命令の対象レコードとなります(下記例では、レコード3が処理対象)。A-VX5互換モードの場合でもREWRITE(KEY)/DELETE(KEY)は、先のREAD命令で読み込んだレコードが対象となります。

- ・ 直前の命令がREAD(保護/共有の場合はWITH UNLOCK指定がないREAD)
- ・ 主キー値の変更がない

なお、条件を満たしていない場合は、チェックしていない場合と同じになります。

例 Bが主キーの部分である(この場合すべてB)

3回READ NEXTを実行している(レコード3を入手している)

B	レコード 1
B	レコード2
B	レコード3
B	レコード4

## (7) ファイル/表削除時のRFDファイル

[ファイル/表削除時、RFDファイルを削除する]チェックボックスによって、COBOL関連製品でIFASファイルを削除すると同時に、RFDファイルを削除することができます。**COBOLアプリケーションでは直接影響しません。**

チェックボックスをチェックすると、IFAS ファイル削除時、RFDファイルを削除します。

なお、ファイル管理ユーティリティでIFAS一般ファイルを削除する場合は、RFDファイルは削除されません。また、表定義ユーティリティでIFAS RDB表を削除する場合は、表定義ユーティリティのオプションに従います。

## (8) COMP-5/BINARY互換モード

[COMP-5/BINARYの仕様をCOBOL85モードにする]チェックボックスによって、IFAS ファイルの列に指定するCOMP-5/BINARYの仕様を決定することができます。

チェックボックスをチェックすると、COMP-5/BINARYの仕様をCOBOL85に合わせます。

チェックボックスをチェックしないと、IFAS FILE、IFASPRO RDBの独自仕様になります。なお、この場合は、COBOLから該当IFAS ファイルを使用することはできません。

## (9) CLOSE命令時のトランザクション動作

[CLOSE命令時のトランザクション動作]ラジオボタンによって、CLOSE命令で暗黙的にCOMMITするかROLLBACKするかを選択できます。

なお、プログラム終了時にオープンされたままのファイルは自動的にクローズされますが、トランザクション処理中のファイルは、本オプションの設定にかかわらず、自動的にロールバックされてからクローズされます。

## (10) 主キーが重複キー許可属性の場合の動作

[RECORD KEYで指定したキーが重複キー許可属性の場合]ラジオボタンによって、OPEN命令のRECORD KEYで指定されたキーに重複キー許可属性があった場合、正常とするか実行時エラーとするかを選択できます。

## (11) 副キーを指定しない場合の動作

[ALTERNATE RECORD KEYを指定しない場合]ラジオボタンによって、OPEN命令でALTERNATE RECORD KEYが指定されなかった場合、IFAS ファイルに定義されているすべてのキーを更新するかRECORD KEYのキーのみ更新するかを選択できます。

## (12) SELECT命令でのソート処理メモリサイズ

[SELECT命令でのソート処理用メモリサイズ]の項目で、IFAS RDB表に対するSELECT命令で確保するメモリサイズをKBで指定します。

最小値はSELECT命令で確保する最小メモリサイズを指定します。最小分のメモリが確保できなかった場合、SELECT命令はエラーとなります。

最大値はSELECT命令で確保する最大メモリサイズを指定します。最大分のメモリが確保できなかった場合は、サイズを小さくてメモリを確保します。

最小値/最大値が0の場合、以下の表のレコード数をもとにメモリを自動取得します。

- ・ 基本表の場合→基本表
- ・ 選択型仮想表→選択型仮想表を構成する基本表
- ・ 結合型仮想表→プライマリ表として構成する基本表
- ・ 合併型仮想表→第一プライマリ表として構成する基本表

なお、必要となるメモリの見積もりは、レコード数(WHERE句／条件式を満たす件数)、および、ORDER BY句(ソートキー)によって異なりますが、およそ以下の計算式で計算できます。

$$(a \times 2 + 4) \times b \times 1.5$$

a = ORDER BY句で指定された項目の合計サイズ

b = WHERE句がある場合は、WHERE句で指定された条件式を満足するレコード数  
WHERE句がない場合は、各表のレコード数

## 4-2-3 更新ログファイル設定

COBOLプログラムからIFAS ファイルを更新した場合の更新ログは、運用支援ツール(「7-4-1 運用支援ツール」を参照)の[更新ログ設定]タブにおける[更新ログファイルパス]に指定されているファイルに出力されます。なお、本項目はCOBOL関連製品での共通設定になっています。

更新ログファイルの運用に関しては、「6-3 更新ログファイルの運用」を参照してください。

## 4-3 一般ファイルの利用

### 4-3-1 ファイル編成と入出力命令

利用可能なファイル編成と入出力命令を以下に示します。各種入出力命令の詳細エラーステータスは「4-3-2 ステータス」を参照してください。

#### 一般ファイルで実行可能な入出力命令

ファイル編成			順編成				相対編成				索引順編成			
OPEN モード			入力	出力	入出力 両用	拡張	入力	出力	入出力 両用	拡張	入力	出力	入出力 両用	拡張
命令														
ファイル呼出方	順呼出	READ	○	×	○	×	○	×	○	×	○	×	○	×
		WRITE	×	○	×	○	×	○	×	○	×	○	×	○
		REWRITE	×	×	○	×	×	×	○	×	×	×	○	×
		START	×	×	×	×	○	×	○	×	○	×	○	×
		DELETE	×	×	×	×	×	×	○	×	×	×	○	×
		COMMIT	○	○	○	○	○	○	○	○	○	○	○	○
		ROLLBACK	○	○	○	○	○	○	○	○	○	○	○	○
	乱／動的呼出	READ	—	—	—	—	○	×	○	—	○	×	○	—
		WRITE	—	—	—	—	×	○	○	—	×	○	○	—
		REWRITE	—	—	—	—	×	×	○	—	×	×	○	—
		START	—	—	—	—	○	×	○	—	○	×	○	—
		DELETE	—	—	—	—	×	×	○	—	×	×	○	—
		COMMIT	—	—	—	—	○	○	○	—	○	○	○	—
		ROLLBACK	—	—	—	—	○	○	○	—	○	○	○	—

## 4-3-2 ステータス

COBOLでIFAS一般ファイルを使用している場合に入出力状態として返却するステータス、および「IFAS FILEエラー」のメッセージが画面に表示される詳細ステータスコードについて説明します。

### 入出力状態

#### OPEN命令でのステータス

- 35 ファイルが存在しません  
RFDファイルが存在しません
- 37 書き込み不可ファイルに対するアクセス要求です
- 39 IFAS ファイルではありません  
レコードサイズの指定が不正です  
(指定レコードサイズはファイルのレコードサイズと一致していなければなりません)  
インデックス／データファイルの組み合わせが不正です  
指定キー情報がファイルにありません  
キー情報が不一致です  
RFDファイルのフォーマットが不正です。
- 41 2重オープンです
- 91 ファイルビジーです  
IFAS システムがバックアップ中です
- 99 ファイル名が不正です
- 9F フェイルオーバーが発生しました  
フェイルオーバー処理中です
- 9J デッドロックが発生しました
- 9K ロールバックログ収集エラーです  
ロールバックマネージャのエラーです
- 9O 更新ログファイルに空きスペースがないためファイルの更新ができません
- 9P ロックテーブルのオーバーフローが発生しました
- 9T 更新ログファイルのI/Oエラーです

#### CLOSE命令でのステータス

- 42 ファイルがオープンされていません
- 9F フェイルオーバーが発生しました  
フェイルオーバー処理中です
- 9J デッドロックが発生しました
- 9K ロールバックログ収集エラーです  
ロールバックマネージャのエラーです
- 9O 更新ログファイルに空きスペースがないためファイルの更新ができません
- 9P ロックテーブルのオーバーフローが発生しました
- 9T 更新ログファイルのI/Oエラーです

#### READ命令でのステータス

- 02 正常終了(2重キーあり)
- 10 ファイルの終端を検出しました
- 14 指定最大レコードを超える読み込み要求です(順アクセス)
- 23 指定レコードが存在しません  
指定最大レコードを超える読み込み要求です(乱アクセス)
- 46 インデックスが設定されていません  
カレントレコードが存在しません
- 47 ファイルがオープンされていません  
オープンモード不正です
- 92 指定インデックスが存在しません  
キー情報が不正です

- 9F フェイルオーバーが発生しました  
フェイルオーバー処理中です
- 9J デッドロックが発生しました
- 9K ロールバックログ収集エラーです  
ロールバックマネージャのエラーです
- 9P ロックテーブルのオーバーフローが発生しました
- 9T 更新ログファイルのI/Oエラーです

**WRITE命令でのステータス**

- 02 正常終了(2重キーあり)
- 21 主キーが昇順になっていません
- 22 重複キーが存在します  
指定レコード番号のレコードは存在します
- 24 ファイルサイズの拡張に失敗しました  
指定最大レコードを超える書き出し要求です
- 48 ファイルがオープンされていません  
オープンモード不正です
- 9F フェイルオーバーが発生しました  
フェイルオーバー処理中です
- 9J デッドロックが発生しました
- 9K ロールバックログ収集エラーです  
ロールバックマネージャのエラーです
- 9O 更新ログファイルに空きスペースがないためファイルの更新ができません
- 9P ロックテーブルのオーバーフローが発生しました
- 9T 更新ログファイルのI/Oエラーです

**REWRITE命令でのステータス**

- 02 正常終了(2重キーあり)
- 21 主キーを更新しようとしてしました
- 22 重複キーが存在します
- 23 指定レコードが存在しません  
指定レコード番号のレコードが存在しません
- 24 指定最大レコードを超える書き出し要求です
- 43 カレントレコードが存在しません  
カレントレコードが読み込まれていません  
SELECT中のためキーアクセスできません
- 49 ファイルがオープンされていません  
オープンモード不正です
- 9F フェイルオーバーが発生しました  
フェイルオーバー処理中です
- 9J デッドロックが発生しました
- 9K ロールバックログ収集エラーです  
ロールバックマネージャのエラーです
- 9O 更新ログファイルに空きスペースがないためファイルの更新ができません
- 9P ロックテーブルのオーバーフローが発生しました
- 9T 更新ログファイルのI/Oエラーです

**DELETE命令でのステータス**

- 22 重複キーが存在します
- 23 指定レコードが存在しません  
指定レコード番号のレコードが存在しません
- 24 指定最大レコードを超える削除要求です
- 43 カレントレコードが存在しません  
カレントレコードが読み込まれていません  
SELECT中のためキーアクセスできません



- 49 ファイルがオープンされていません  
オープンモード不正です
- 9F フェイルオーバーが発生しました  
フェイルオーバー処理中です
- 9J デッドロックが発生しました
- 9K ロールバックログ収集エラーです  
ロールバックマネージャのエラーです
- 9O 更新ログファイルに空きスペースがないためファイルの更新ができません
- 9P ロックテーブルのオーバーフローが発生しました
- 9T 更新ログファイルのI/Oエラーです

#### START命令でのステータス

- 14 指定最大レコードを超える位置付け要求です(順アクセス)
- 23 指定レコードが存在しません  
指定最大レコードを超える位置付け要求です(乱アクセス)
- 46 インデックスが設定されていません  
カレントレコードが存在しません
- 47 ファイルがオープンされていません  
オープンモード不正です
- 92 指定インデックスが存在しません  
キー情報が不正です
- 9F フェイルオーバーが発生しました  
フェイルオーバー処理中です
- 9J デッドロックが発生しました
- 9K ロールバックログ収集エラーです  
ロールバックマネージャのエラーです
- 9P ロックテーブルのオーバーフローが発生しました
- 9T 更新ログファイルのI/Oエラーです

## 詳細エラーステータス

プログラムがアプリケーションエラーとなり、「IFAS FILEエラー」のメッセージ画面が表示された場合の詳細ステータスコードについて説明します。

ここに示すエラーステータス以外にもIFAS FILE関数のステータスが返却されることもあります。該当するステータスが見つからない場合は「付録A-3 C言語関数のステータス」も参照してください。

#### OPEN命令でのステータス

- 50010(0xc35a) ファイル名が不正です
- 50011(0xc35b) レコード長が不正です
- 50012(0xc35c) 指定レコード長が既存ファイルのレコード長と一致しません
- 50013(0xc35d) オープンモードまたはアクセスモードが不正です
- 50014(0xc35e) 主キー情報に二重キー指定があります
- 50015(0xc35f) ファイルテーブルの登録に失敗しました
- 50016(0xc360) キーテーブル用のメモリ確保に失敗しました
- 50018(0xc362) 最大キー用のメモリ確保に失敗しました
- 50019(0xc363) 最大キー用のメモリ解放に失敗しました
- 50020(0xc364) キー情報が不一致です
- 50022(0xc366) キー変換に失敗しました
- 50023(0xc367) キー変換に失敗しました
- 50024(0xc368) 主キー情報の比較に失敗しました
- 50025(0xc369) キーテーブルの作成に失敗しました
- 50026(0xc36a) 指定キー情報がファイルに見つかりません
- 50027(0xc36b) 主キー値の更新に失敗しました
- 50028(0xc36c) カレントキー情報の更新に失敗しました
- 50029(0xc36d) 指定キー情報のインデックス属性が不正です

50310(0xc486)	RFDファイルのオープンに失敗しました
50311(0xc487)	RFDファイルが存在しません
50312(0xc488)	RFDファイルのフォーマットが不正です
50313(0xc489)	RFDファイルの入力に失敗しました
50314(0xc48a)	RFDファイルの内部処理でエラーが発生しました
50315(0xc48b)	RFDファイルのクローズに失敗しました
50316(0xc48c)	メモリ確保に失敗しました
54009(0xd2f9)	ファイル名変換でエラーです
54011(0xd2fb)	更新ログファイルのオープンでエラーです

**CLOSE命令でのステータス**

50030(0xc36e)	ファイルテーブルの削除に失敗しました
50032(0xc370)	直前キー用のメモリ解放に失敗しました
50033(0xc371)	キー情報用のメモリ解放に失敗しました
54013(0xd2fd)	commitでエラーです
54014(0xd2fe)	rollbackでエラーです

**READ命令でのステータス**

50034(0xc372)	現在のオープンモードでは本要求を実行できません
50035(0xc373)	カレントレコードが存在しません
50036(0xc374)	実際のレコードサイズより小さいレコードサイズが指定されました
50040(0xc378)	ロックモードが不正です
50041(0xc379)	アクセスモードが不正です
50042(0xc37a)	レコード長が不正です
50043(0xc37b)	キー情報が不正です
50044(0xc37c)	アクセスモードが不一致です
50045(0xc37d)	カレントキー情報の更新が不正です
50046(0xc37e)	主キー値の更新に失敗しました
50047(0xc37f)	セレクト中のため、乱アクセスは指定できません
50110(0xc3be)	ロックモードが不正です
50111(0xc3bf)	アクセスモードが不正です
50112(0xc3c0)	レコード長が不正です
50113(0xc3c1)	アクセスモードが不一致です
50114(0xc3c2)	セレクト中のため、乱アクセスは指定できません
50115(0xc3c3)	現在のオープンモードでは本要求を実行できません
50116(0xc3c4)	カレントレコードが存在しません
50117(0xc3c5)	実際のレコードサイズより小さいレコードサイズが指定されました
54020(0xd304)	トランザクション開始処理でエラーです

**WRITE命令でのステータス**

50050(0xc382)	レコード長が不正です
50051(0xc383)	指定レコード内のキー値の順序が不正です
50052(0xc384)	主キー値の更新に失敗しました
50053(0xc385)	現在のオープンモードでは本要求を実行できません
50054(0xc386)	実際のレコードサイズより小さいレコードサイズが指定されました
50120(0xc3c8)	レコード長が不正です
50121(0xc3c9)	現在のオープンモードでは本要求を実行できません
50122(0xc3ca)	実際のレコードサイズより小さいレコードサイズが指定されました
54015(0xd2ff)	トランザクション開始処理でエラーです

**REWRITE命令でのステータス**

50061(0xc38d)	レコード長が不正です
50062(0xc38e)	キー情報が不正です
50063(0xc38f)	アクセスモードが不一致です
50064(0xc390)	カレントレコードが読み込まれていません
50066(0xc392)	主キー値を更新しようとした

50067(0xc393)	セレクト中のため、キーアクセスは指定できません
50068(0xc394)	現在のオープンモードでは本要求を実行できません
50069(0xc395)	カレントレコードが存在しません
50070(0xc396)	実際のレコードサイズより小さいレコードサイズが指定されました
50130(0xc3d2)	レコード長が不正です
50131(0xc3d3)	アクセスモードが不一致です
50132(0xc3d4)	カレントレコードが読み込まれていません
50133(0xc3d5)	アクセスモードが不一致です
50134(0xc3d6)	セレクト中のため、キーアクセスは指定できません
50135(0xc3d7)	現在のオープンモードでは本要求を実行できません
50136(0xc3d8)	カレントレコードが存在しません
50137(0xc3d9)	実際のレコードサイズより小さいレコードサイズが指定されました
54016(0xd300)	トランザクション開始処理でエラーです

**DELETE命令でのステータス**

50071(0xc397)	キー情報が不正です
50072(0xc398)	アクセスモードが不一致です
50073(0xc399)	カレントレコードが読み込まれていません
50074(0xc39a)	セレクト中のため、キーアクセスは指定できません
50075(0xc39b)	現在のオープンモードでは本要求を実行できません
50076(0xc39c)	カレントレコードが存在しません
50140(0xc3dc)	アクセスモードが不一致です
50141(0xc3dd)	カレントレコードが読み込まれていません
50142(0xc3de)	アクセスモードが不一致です
50143(0xc3df)	セレクト中のため、キーアクセスは指定できません
50144(0xc3e0)	現在のオープンモードでは本要求を実行できません
50145(0xc3e1)	カレントレコードが存在しません
54017(0xd301)	トランザクション開始処理でエラーです

**START命令でのステータス**

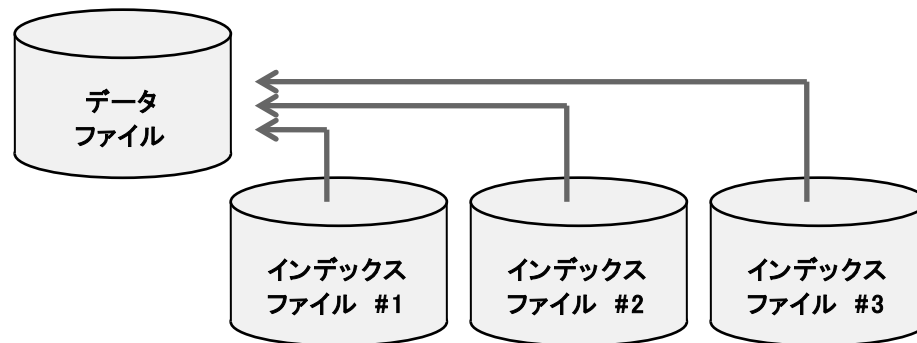
50080(0xc3a0)	アクセスモードが不正です
50081(0xc3a1)	キー情報が不正です
50082(0xc3a2)	カレントキー情報の更新が不正です
50083(0xc3a3)	セレクト中のため、乱アクセスは指定できません
50084(0xc3a4)	現在のオープンモードでは本要求を実行できません
50085(0xc3a5)	カレントレコードが存在しません
50150(0xc3e6)	アクセスモードが不正です
50151(0xc3e7)	セレクト中のため、乱アクセスは指定できません
50152(0xc3e8)	現在のオープンモードでは本要求を実行できません
54021(0xc305)	トランザクション開始処理でエラーです

## 4-3-3 MIXファイル互換機能

オフィスサーバ、オフィスプロセッサのA-VXシリーズのCOBOLで利用される複数索引順編成ファイル(MIXファイル)のアクセス機能と同等の機能をIFAS一般索引順編成ファイルにて提供します。

### MIXファイル概要

A-VXシリーズのMIXファイルは以下の図のように1つのデータファイルに対して複数(最大8個)のインデックスファイルを作成することが可能です。



利用者はプログラムでファイル定義する場合、必ずしも全インデックスファイルの記述をする必要はなく、どれか1つのインデックスファイルだけをRECORD KEYとして定義しても、他のいくつかのインデックスファイルをALTERNATE RECORD KEYとして定義して使用することも可能です。また、インデックスファイルそれぞれに更新属性を定義することが可能であり、データファイルの更新に伴ってプログラムで定義していないインデックスファイルも常に更新を反映させることや、プログラムで定義したインデックスファイルのみ更新し、他のインデックスファイルはそのプログラム処理終了後、別プログラム(ユーティリティ)で一括して更新するなど、利用者がインデックス更新方式を選択することができます。

インデックスファイルの更新属性には以下のようなものがあります。

#### 非同時更新

プログラムでRECORD KEYとしてキー指定していない場合はレコードの追加／削除／更新を行ってもインデックスに更新を反映しません。その結果、データとインデックスの不整合状態になり、その後、このインデックスを参照した場合はエラーとなります。この不整合状態は、利用者がユーティリティで修復することができます。

#### 同時更新

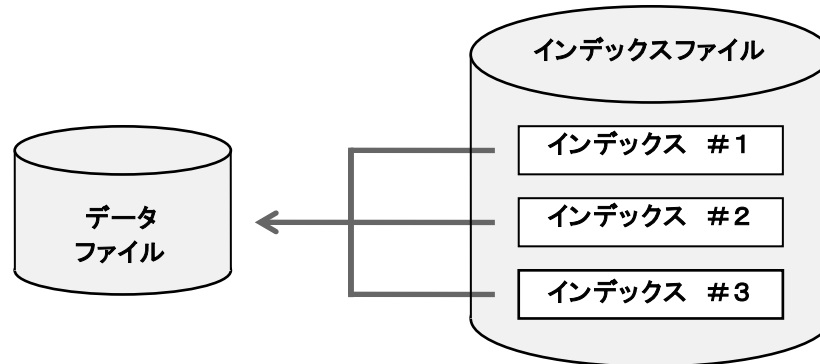
プログラムでキー指定していない場合でも、レコードの追加／削除／更新の結果を即時インデックスに反映します。

#### 実行時生成

通常は非同時更新属性と同様ですが、更新非同期状態になった後で参照されてもエラーとせず、動的にインデックスの再編成処理(キーの生成)を行い、属性を「同時更新」に変更します。

## IFAS FILEでのサポート

IFAS一般索引順編成ファイルは1つのインデックスファイルに複数(最大255個)のインデックス構造を構築することで複数索引形態を実現しています。



A-VXのMIXファイルと同等の機能をIFAS FILEの一般索引順編成ファイルで実現する上で、**プライマリインデックスを設定しないで作成したファイル**をMIX互換機能のサポート対象ファイルとします。

COBOLで複数インデックスを定義し、作成した(OUTPUTモードでオープンした)ファイルは、すべてこのプライマリインデックスのないサブインデックスだけで構成された形式のファイル、つまりMIX互換機能対応のファイルになります。ファイルメンテナンスキットで作成したファイルも同様です。

C言語用IFAS FILE関数でMIX互換機能対応のファイルを作成する場合は、OUTPUTモードのOPEN関数やCREATE関数ではキー記述の指定をしないようにし、インデックスはすべてADDINDEX関数で作成するようにします。また、ファイルコピーユーティリティや移行ユーティリティでMIX互換機能対応のファイルを作成する場合も、プライマリインデックス(インデックス番号1のインデックス)を定義しないようにします。

SORTKITでMIX互換機能対応のファイルを作成する場合は、SORTKIT関数およびSORTKITユーティリティでプライマリキーの設定をしないようにします。

## 仕様について

インデックス毎の同時更新／非同時更新／実行時生成の指定はインデックスメンテナンసుユーティリティの[ATTRIBUTE(更新属性の変更)]で行い、ファイル内に属性として記憶します。この機能はプライマリインデックスが定義されているファイルに対しては実行できません。

ALTERNATE RECORD KEY指定がある場合は、指定されたすべてのインデックスが同時更新属性でなければなりません。この場合、指定されていない同時更新属性のインデックスも同時に更新されます。また、指定された中に1つでも不整合状態のインデックスが存在すると、そのファイルはオープンできません。

RECORD KEY のみの指定の場合は、指定したRECORD KEYのインデックスが不整合状態でなければ、そのファイルはオープンできます。この場合、指定されていない同時更新属性のインデックスも同時に更新されます。

## データとインデックスの非同期(不整合)状態

### A-VX

データファイルとインデックスファイルの更新回数 (OPEN-CLOSEまでの処理を1回の更新処理と考えます) が不一致の場合、そのインデックスファイルを非同期状態と判断します。非同期状態を発生させたプログラムがファイルをクローズした後、この非同期インデックスを定義してオープンするとコンソールにその旨を示す通知エラーメッセージを出力しますが、オープン処理自体は成功します。

### IFAS FILE

レコードの追加／更新／削除が発生した時点で、更新が反映されない属性のインデックスは不整合状態となります。不整合状態のインデックスをRECORD KEYやALTERNATE RECORD KEYで指定してオープンした場合は実行時エラーとなります。不整合状態を発生させたプログラムがファイルをクローズしないうちに、別プログラムがこの不整合状態のインデックスをRECORD KEYやALTERNATE RECORD KEYで指定してオープンした場合も同様です。この場合はインデックスメンテナンスユーティリティの[REMAKE (インデックスの再編成)]でインデックスの再編成を行うことにより、不整合状態を修復できます。

## 4-4 RDB表の利用

### 4-4-1 ファイル編成と入出力命令

利用可能なファイル編成と入出力命令を以下に示します。各種入出力命令の詳細エラーステータスは「4-4-2 ステータス」を参照してください。

RDB表で実行可能な入出力命令

ファイル編成			RDB順編成				RDB相対編成				RDB索引順編成			
OPEN モード			入力	出力	入出力 両用	拡張	入力	出力	入出力 両用	拡張	入力	出力	入出力 両用	拡張
命令														
ファイル呼出方	順呼出	READ	○	×	○	×	○	×	○	×	○	×	○	×
		WRITE	×	○	×	○	×	○	×	○	×	○	×	○
		REWRITE	×	×	○	×	×	×	○	×	×	×	○	×
		START	○	×	○	×	○	×	○	×	○	×	○	×
		DELETE	×	×	○	×	×	×	○	×	×	×	○	×
		SELECT	○	×	○	×	○	×	○	×	○	×	○	×
		SCRATCH	○	×	○	×	○	×	○	×	○	×	○	×
		COMMIT	○	○	○	○	○	○	○	○	○	○	○	○
		ROLLBACK	○	○	○	○	○	○	○	○	○	○	○	○
	乱／動的呼出	READ	—	—	—	—	○	×	○	—	○	×	○	—
		WRITE	—	—	—	—	×	○	○	—	×	○	○	—
		REWRITE	—	—	—	—	×	×	○	—	×	×	○	—
		START	—	—	—	—	○	×	○	—	○	×	○	—
		DELETE	—	—	—	—	×	×	○	—	×	×	○	—
		SELECT	—	—	—	—	○	×	○	—	○	×	○	—
		SCRATCH	—	—	—	—	○	×	○	—	○	×	○	—
		COMMIT	—	—	—	—	○	○	○	—	○	○	○	—
		ROLLBACK	—	—	—	—	○	○	○	—	○	○	○	—

## 4-4-2 ステータス

COBOLでIFAS RDB表を使用している場合に入出力状態として返却するステータス、および「IFAS FILE エラー」のメッセージ画面に表示される詳細ステータスコードについて説明します。

### 入出力状態

#### OPEN命令でのステータス

- 35 ファイルが存在しません  
指定されたDBが存在しません  
指定された表が存在しません  
RFD ファイルが存在しません
- 37 書き込み不可ファイルに対するアクセス 要求です  
仮想表に対しオープンモードにOUTPUT／EXTENDモードが指定されました
- 39 IFAS ファイルではありません  
レコードサイズの指定が不正です  
インデックス／データファイルの組み合わせが不正です  
レコード長が不正です  
指定された主キーがIFAS ファイルに存在しません  
RFD ファイルのフォーマットが不正です
- 41 表の2重オープンです
- 91 ファイルビジーです  
IFAS システムがバックアップ中です  
DBシステムがビジーです
- 99 ファイル名が不正です
- 9F フェイルオーバーが発生しました  
フェイルオーバー処理中です
- 9J デッドロックが発生しました
- 9K ロールバックログ収集エラーです  
ロールバックマネージャのエラーです
- 9M RDB管理ファイルのアクセスエラー(DB破壊)です
- 9O 更新ログファイルに空きスペースがないためファイルの更新ができません
- 9P ロックテーブルのオーバーフローが発生しました
- 9T 更新ログファイルのI／Oエラーです

#### CLOSE命令でのステータス

- 42 ファイルがオープンされていません
- 9F フェイルオーバーが発生しました  
フェイルオーバー処理中です
- 9J デッドロックが発生しました
- 9K ロールバックログ収集エラーです  
ロールバックマネージャのエラーです
- 9O 更新ログファイルに空きスペースがないためファイルの更新ができません
- 9P ロックテーブルのオーバーフローが発生しました
- 9T 更新ログファイルのI／Oエラーです

#### READ命令でのステータス

- 02 正常終了(2重キーあり)
- 10 ファイルの終端を検出しました
- 14 指定最大レコードを超える読み込み 要求です(順アクセス)
- 23 指定レコードが存在しません  
指定最大レコードを超える読み込み 要求です(乱アクセス)
- 46 インデックスが設定されていません



- カレントレコードが存在しません
- 47 ファイルがオープンされていません  
オープンモード不正です
- 92 指定インデックスが存在しません  
キー情報が不正です
- 9F フェイルオーバーが発生しました  
フェイルオーバー処理中です
- 9J デッドロックが発生しました
- 9K ロールバックログ収集エラーです  
ロールバックマネージャのエラーです
- 9P ロックテーブルのオーバーフローが発生しました
- 9T 更新ログファイルのI/Oエラーです

#### WRITE命令でのステータス

- 02 正常終了(2重キーあり)
- 21 主キーが昇順になっていません
- 22 重複キーが存在します  
指定レコード番号のレコードは存在します
- 24 ファイルサイズの拡張に失敗しました  
指定最大レコードを超える書き出し要求です
- 48 ファイルがオープンされていません  
オープンモード不正です
- 9F フェイルオーバーが発生しました  
フェイルオーバー処理中です
- 9J デッドロックが発生しました
- 9K ロールバックログ収集エラーです  
ロールバックマネージャのエラーです
- 9O 更新ログファイルに空きスペースがないためファイルの更新ができません
- 9P ロックテーブルのオーバーフローが発生しました
- 9T 更新ログファイルのI/Oエラーです

#### REWRITE命令でのステータス

- 02 正常終了(2重キーあり)
- 21 主キーを更新しようとした
- 22 重複キーが存在します
- 23 指定レコードが存在しません  
指定レコード番号のレコードが存在しません
- 24 指定最大レコードを超える書き出し要求です
- 43 カレントレコードが存在しません  
カレントレコードが読み込まれていません  
SELECT中のためキーアクセスできません
- 49 ファイルがオープンされていません  
オープンモード不正です
- 9F フェイルオーバーが発生しました  
フェイルオーバー処理中です
- 9J デッドロックが発生しました
- 9K ロールバックログ収集エラーです  
ロールバックマネージャのエラーです
- 9O 更新ログファイルに空きスペースがないためファイルの更新ができません
- 9P ロックテーブルのオーバーフローが発生しました
- 9T 更新ログファイルのI/Oエラーです

#### DELETE命令でのステータス

- 22 重複キーが存在します
- 23 指定レコードが存在しません

- 指定レコード番号のレコードが存在しません
- 24 指定最大レコードを超える削除要求です
- 43 カレントレコードが存在しません  
カレントレコードが読み込まれていません  
SELECT中のためキーアクセスできません
- 49 ファイルがオープンされていません  
オープンモード不正です
- 9F フェイルオーバーが発生しました  
フェイルオーバー処理中です
- 9J デッドロックが発生しました
- 9K ロールバックログ収集エラーです  
ロールバックマネージャのエラーです
- 9O 更新ログファイルに空きスペースがないためファイルの更新ができません
- 9P ロックテーブルのオーバーフローが発生しました
- 9T 更新ログファイルのI/Oエラーです

**START命令でのステータス**

- 14 指定最大レコードを超える位置付け要求です(順アクセス)
- 23 指定レコードが存在しません  
指定最大レコードを超える位置付け要求です(乱アクセス)
- 46 インデックスが設定されていません  
カレントレコードが存在しません
- 47 ファイルがオープンされていません  
オープンモード不正です
- 92 指定インデックスが存在しません  
キー情報が不正です
- 9F フェイルオーバーが発生しました  
フェイルオーバー処理中です
- 9J デッドロックが発生しました
- 9K ロールバックログ収集エラーです  
ロールバックマネージャのエラーです
- 9P ロックテーブルのオーバーフローが発生しました
- 9T 更新ログファイルのI/Oエラーです

**SELECT命令でのステータス**

- 47 ファイルがオープンされていません  
オープンモード不正です
- 92 オープンモードにOUTPUT/EXTENDが指定されました
- 9F フェイルオーバーが発生しました  
フェイルオーバー処理中です
- 9J デッドロックが発生しました
- 9K ロールバックログ収集エラーです  
ロールバックマネージャのエラーです
- 9P ロックテーブルのオーバーフローが発生しました
- 9T 更新ログファイルのI/Oエラーです

**SCRATCH命令でのステータス**

- 47 ファイルがオープンされていません  
オープンモード不正です
- 92 SELECTされていません
- 9F フェイルオーバーが発生しました  
フェイルオーバー処理中です
- 9J デッドロックが発生しました
- 9K ロールバックログ収集エラーです  
ロールバックマネージャのエラーです
- 9P ロックテーブルのオーバーフローが発生しました

9T 更新ログファイルのI/Oエラーです

## 詳細エラーステータス

プログラムがアプリケーションエラーとなり、「IFAS FILEエラー」のメッセージ画面が表示された場合の詳細ステータスコードについて説明します。

ここに示すエラーステータス以外にもIFAS FILE関数のステータスが返却されることもあります。該当するステータスが見つからない場合は「付録A-3 C言語関数のステータス」も参照してください。

### OPEN命令でのステータス

50521(0xc559)	アクセスモードが正しく指定されていません
50522(0xc55a)	オープンモードが正しく指定されていません
50523(0xc55b)	指定レコードサイズとカラム情報の合計サイズが一致しません
50524(0xc55c)	インデックス情報が正しく指定されていません
50525(0xc55d)	カラム情報が正しく指定されていません
50526(0xc55e)	表の2重オープンです
50527(0xc55f)	主キーとして指定されたキーが存在しません
50528(0xc560)	オープン情報の登録に失敗しました
50529(0xc561)	条件式に合うレコードの抽出に失敗しました
50530(0xc562)	メモリ確保に失敗しました
50531(0xc563)	表定義情報入手でアクセスエラーです
50533(0xc565)	指定された表が存在しません
50534(0xc566)	RDB管理ファイルのアクセスエラーです(DB破壊)
50535(0xc567)	指定されたDBが存在しません
50536(0xc568)	データ変換に失敗しました
50537(0xc569)	仮想表に対しオープンモードにOUTPUT/EXTENDが指定されました
50538(0xc56a)	仮想表に対しオープンモードにI/Oが指定されました
50540(0xc56c)	アクセスキーに対応するインデックスが見つかりません
50541(0xc56d)	アクセスキーと対応キーの関係が不正です
50542(0xc56e)	表定義ユーティリティで作成された仮想表ではないためアクセスできません
50543(0xc56f)	相対編成アクセスができない仮想表です
50544(0xc570)	各プライマリ表のレコードサイズが一致しません
50545(0xc571)	第2プライマリ表以降に主キーとして指定されたキーが存在しません
50546(0xc572)	RFDファイルのオープンに失敗しました
50547(0xc573)	RFDファイルが存在しません
50548(0xc574)	RFDファイルのフォーマットが不正です
50549(0xc575)	RFDファイルの入力に失敗しました
50550(0xc576)	RFDファイルの内部処理でエラーが発生しました
50551(0xc577)	RFDファイルのクローズに失敗しました
54007(0xd2f7)	IFASPRO RDBがインストールされていません
54008(0xd2f8)	表名変換でエラーです
54010(0xd2fa)	表定義オプション入手でエラーです
54011(0xd2fb)	更新ログファイルのオープンでエラーです
54012(0xd2fc)	RDB管理ファイル一括オープンでエラーです RDB管理ファイルがありません 表定義ユーティリティで環境設定が行われていません
54096(0xd350)	DBシステムがビジーです IFAS FILE システムがバックアップ中です

### CLOSE命令でのステータス

54013(0xd2fd)	commitでエラーです
54014(0xd2fe)	rollbackでエラーです

**READ命令でのステータス**

50561(0xc581)	アクセスモードが正しく指定されていません
50562(0xc582)	ロックモードが正しく指定されていません
50563(0xc583)	実際のレコードサイズより小さいレコードサイズが指定されました
50564(0xc584)	順次呼び出しのファイルに乱アクセスを指定しました
50565(0xc585)	指定されたキーが存在しません
50566(0xc586)	オープン時のアクセスモードが不正です
50567(0xc587)	セレクト中のため、乱アクセスは指定できません
50569(0xc589)	オープンモードとの組み合わせが不正です
50570(0xc58a)	データ変換に失敗しました
50571(0xc58b)	カレントレコードが見つかりません
50572(0xc58c)	セカンダリ表のレコードが存在しません
50573(0xc58d)	ロック無視 READ の宣言／解除に失敗しました
50574(0xc58e)	メモリ確保に失敗しました
50575(0xc58f)	EOFを検出しました
50576(0xc590)	指定されたレコードが存在しません
50577(0xc591)	第2プライマリ表以降に指定されたキーが存在しません
54020(0xd304)	トランザクション開始処理でエラーです

**WRITE命令でのステータス**

50581(0xc595)	実際のレコードサイズより小さいレコードサイズが指定されました
50582(0xc596)	オープン時のアクセスモードが不正です
50583(0xc597)	表種別が不正です
50584(0xc598)	主キーが昇順ではありません
50585(0xc599)	オープンモードとの組み合わせが不正です
50586(0xc59a)	データ変換に失敗しました
54015(0xd2ff)	トランザクション開始処理でエラーです

**REWRITE命令でのステータス**

50601(0xc5a9)	アクセスモードが正しく指定されていません
50602(0xc5aa)	実際のレコードサイズより小さいレコードサイズが指定されました
50603(0xc5ab)	順次呼び出しのファイルにキーアクセスを指定しました
50604(0xc5ac)	指定されたキーが存在しません
50605(0xc5ad)	オープン時のアクセスモードが不正です
50606(0xc5ae)	表種別が不正です
50607(0xc5af)	セレクト中のため、キーアクセスは指定できません
50608(0xc5b0)	カレントレコードが存在しません
50609(0xc5b1)	主キーを更新しようとして失敗しました
50610(0xc5b2)	オープンモードとの組み合わせが不正です
50611(0xc5b3)	データ変換に失敗しました
50612(0xc5b4)	重複キーを検出したため更新できません
50613(0xc5b5)	指定されたレコードが存在しません
50614(0xc5b6)	メモリ確保に失敗しました
54016(0xd300)	トランザクション開始処理でエラーです

**DELETE命令でのステータス**

50611(0xc5b3)	データ変換に失敗しました
50621(0xc5bd)	アクセスモードが正しく指定されていません
50622(0xc5be)	順次呼び出しのファイルにキーアクセスを指定しました
50623(0xc5bf)	指定されたキーが存在しません
50624(0xc5c0)	オープン時のアクセスモードが不正です
50625(0xc5c1)	表種別が不正です
50626(0xc5c2)	セレクト中のため、キーアクセスは指定できません
50627(0xc5c3)	カレントレコードが存在しません
50628(0xc5c4)	オープンモードとの組み合わせが不正です
50629(0xc5c5)	データ変換に失敗しました

50630(0xc5c6)	重複キーを検出したため削除できません
50631(0xc5c7)	指定されたレコードが存在しません
50632(0xc5c8)	メモリ確保に失敗しました
54017(0xd301)	トランザクション開始処理でエラーです

**START命令でのステータス**

50570(0xc58a)	データ変換に失敗しました
50576(0xc590)	指定されたレコードが存在しません
50641(0xc5d1)	アクセスモードが正しく指定されていません
50642(0xc5d2)	順次呼び出しのファイルに乱アクセスを指定しました
50643(0xc5d3)	指定されたキーが存在しません
50644(0xc5d4)	オープン時のアクセスモードが不正です
50645(0xc5d5)	セレクト中のため、乱アクセスは指定できません
50646(0xc5d6)	オープンモードとの組み合わせが不正です
50647(0xc5d7)	データ変換に失敗しました
50648(0xc5d8)	カレントレコードが見つかりません
50649(0xc5d9)	メモリ確保に失敗しました
50650(0xc5da)	セカンダリ表のレコードが存在しません
50651(0xc5db)	EOFを検出しました
50652(0xc5dc)	第2プライマリ表以降に指定されたキーが存在しません
50653(0xc5dd)	相対番号に0が指定されました
54021(0xd305)	トランザクション開始処理でエラーです

**SELECT命令でのステータス**

50741(0xc635)	オープンモードにOUTPUT/EXTENDが指定されました
50742(0xc636)	メモリ確保に失敗しました
50743(0xc637)	メモリ解放に失敗しました
50744(0xc638)	ワークファイルが存在しません
50745(0xc639)	カラム情報が存在しません
50746(0xc63a)	アクセスモードが不正です
50747(0xc63b)	内部処理不正です
50748(0xc63c)	指定キータイプがファイルの属性と違います
50749(0xc63d)	指定キータイプが不正です
50750(0xc63e)	ORDER BY句で記述された項目が65個以上指定されました
50751(0xc63f)	ワークファイルの作成に失敗しました
50752(0xc640)	ORDER BY句で記述された項目の合計サイズが4096バイトを超えました
50753(0xc641)	内部SORT処理でエラーが発生しました
50754(0xc642)	内部SORT処理がメモリ確保に失敗しました
50755(0xc643)	内部SORT処理がディスクスペース確保に失敗しました

**SCRATCH命令でのステータス**

50761(0xc649)	SELECTされていません
50762(0xc64a)	ワークファイルが存在しません
50763(0xc64b)	オープンモードとの組み合わせが不正です
50764(0xc64c)	ワークファイル削除に失敗しました

## 4-4-3 各種RDB表使用時の注意事項

### (1) 基本表およびすべての仮想表

- RDB索引順編成としてRECORD KEY、ALTERNATE RECORD KEYで使用するキーは、あらかじめ表定義ユーティリティでインデックスに登録しておく必要があります。
- SELECT命令のWHERE句で集団項目を指定した場合、その集団項目は文字列として比較が行われます。ただし、集団項目でそれに従属する基本項目がただ1つの時は、その基本項目のデータ型にしたがった比較が行われます。

(例) 以下の形式で定義された表／データを使用してSELECTを実行

FILE SECTION.

FD F001

LABEL RECORD IS STANDARD

VALUE OF IDENTIFICATION "TABLE".

01 F001-REC.

02 COLM1 PIC S999 USAGE IS COMP-3.

02 COLM23.

03 COLM2 PIC 999.

03 COLM3 PIC S999 SIGN IS TRAILING SEPARATE.

02 COLM40.

03 COLM4 PIC S999 SIGN IS TRAILING.

02 COLM5 PIC S99 SIGN IS TRAILING.

WORKING-STORAGE SECTION.

01 VALUE1.

02 VALUE1-DATA PIC S999 SIGN IS TRAILING VALUE 200.

01 VALUE2.

02 VALUE21 PIC 999 VALUE 0.

02 VALUE22 PIC S999 SIGN IS TRAILING SEPARATE VALUE 0.

○SELECT F001 WHERE COLM40 IS < VALUE1.

とした場合は、200より小さい4列目(COLM4)を持つデータを抽出

○SELECT F001 WHERE VALUE2 IS < COLM23.

とした場合は、2列目(COLM2)と3列目(COLM3)を合わせて1つの文字列(COLM23)とし、  
"000000+"より大きいCOLM23を持つデータを抽出

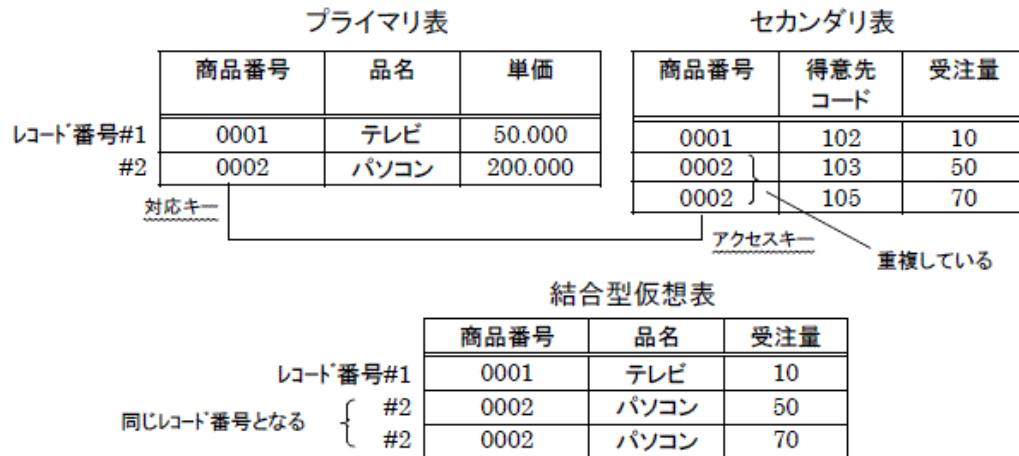
### (2) 選択型仮想表

- OS互換モードがA-VX5モードの場合、出力／拡張モードでのオープンはできません。
- 選択型仮想表にWRITE命令を実行した場合、仮想表に選択されなかった基本表の項目には数値属性ならば0、文字属性ならばスペースが書き出されます。本対象となる項目が二重キー許可のないインデックスとして定義されている場合、この仮想表に対しWRITE命令を行ってはいけません。キー値の重複が起こり、WRITE命令が失敗します。

### (3) 結合型仮想表

- 出力／拡張モードでのオープンはできません。
- 入出力両用モードでオープンした場合、WRITE/REWRITE/DELETE命令による更新はできません。
- RDB索引順編成として使用する場合、RECORD KEY、ALTERNATE RECORD KEYで指定できるキーはプライマリ表にインデックスとして定義されている列のみです。
- RDB索引順編成として使用する場合、二重キー検出のためREAD命令に時間がかかる場合があります。これを回避したい場合、COBOLファイルマッパーのオプション "NOREAD02(READ命令で二重キーを検出しない)" を指定してください。これにより、多少の性能改善が図れます。ただし、入出力状態に"02"ステータスは返却されません。

- RDB相対編成ファイルとして使用する場合、相対キーはプライマリ表のレコード番号となります。したがって、アクセスキーであるインデックスに重複するキー値が存在するような場合、同じ相対キーを持つレコードが複数存在することになります。



- 対応キーとアクセスキーは、データ型、桁数および小数点以下桁数が一致していなければなりません。キー属性の違うカラムを結合させたい場合(たとえばアクセスキー＝文字／対応キー＝外部10進)は、とりあえず定義上は両者を「文字」属性で統一し、COBOLソース上のレコード定義で「外部10進」属性で参照するなどの工夫が必要になります。

#### (4) 合併型仮想表

- 出力／拡張モードでオープンできません。
- RDB索引順編成として使用する場合、RECORD KEY、ALTERNATE RECORD KEYとして指定できるキーは、元となるすべてのプライマリ表に定義しておく必要があります。
- RDB相対編成としてアクセスできません。
- WRITE命令は実行できません。
- REWRITE命令では重複キーの検出が行われません。したがって、重複キーが発生するような更新を行った場合でも入手出力状態に“02”ステータスは返却されません。

## 5. IFASカスタムコントロール(参考)

IFAS FILE Ver2.0、IFASPRO RDB Ver2.0より、「IFASカスタムコントロール」機能の提供を中止しました。本章の内容は従来バージョンをご利用のお客様が旧製品の仕様を確認いただくために残しています。

カスタムコントロールはIFAS ファイルのopen/close、レコードのread/write機能などを提供しており、MS-Visual BasicからIFAS一般ファイルやIFAS RDB表へのアクセス機能を簡単に利用することができます。 ※32ビットAPからのみ利用可能です。

### 1つのコントロールで複数のIFAS ファイルをアクセス可能

複数のIFAS一般ファイルやIFAS RDB表を同時にアクセスできます。

### インターフェースの簡略化

COBOL、C言語と比較しインターフェースが簡略化されています。

たとえば、索引順編成ファイルの索引指定もデータ中のオフセットを指定せず、項目の名前で行うことができます。

### データ入手の簡便化

IFAS ファイル中のデータの型をString型のデータとして扱うことができます。ファイルへの読み書きはカスタムコントロールが型を自動的に合致させます。APでのデータの扱いが簡単にできます。

### IFAS ファイル情報を入手可能

IFAS一般ファイルの情報やIFAS RDB表の定義内容を簡易に取得できます。汎用性のあるAPの作成もできます。

### コントロールでIFAS ファイル利用方法の変更が可能

ファイルや表単位にログ収集の可否などIFAS FILE、IFASPRO RDBのファイルオプションがコントロールから指示できます。

### エラー処理の記述が簡易

エラーステートメントにより、エラー処理を一括して記述することができます。

## 5-1 機能概要

### 5-1-1 諸元

IFAS一般ファイルおよびIFAS RDB表はそれぞれ下表のような諸元となっています。

	IFAS一般ファイル	IFAS RDB表
最大レコードサイズ	64K バイト	32K-1 バイト
最大キーサイズ	256 バイト	255 バイト
最大複数索引数	255 個	
集合／合成キーの最大キーパート数 (1つの索引に指定できる項目数)	16個	
	(ただし、合計キーサイズが 256バイトまで)	(ただし、合計キーサイズが 255バイトまで)
最大列数(項目数)	—	2000 列



#### 一般索引順編成ファイルのファイル名について

一般索引順編成ファイルは、指定されたファイル名がデータファイル名となり、データファイル名に拡張子“.idx”を付加したものがインデックスファイル名となります。データファイル名に拡張子が付加されている場合は、その拡張子を“.idx”に変更し、インデックスファイル名とします。これは、IFAS システムが指定されたファイル名を自動的に変換しています。したがって、一般索引順編成ファイルのファイル名に拡張子“.idx”は指定できません。

## 5-1-2 データアクセス機能

### (1) 一般ファイルのアクセス

カスタムコントロールから以下の3種類のファイル編成が使用可能です。ただし、一般ファイルをアクセスするためには、RFDファイルが必須となります。RFDファイルのない一般ファイルは「RFD編集ツール」でRFDファイルを作成します。

- ・ 一般索引順編成ファイル
- ・ 一般相対編成ファイル
- ・ 一般順編成ファイル

### (2) RDB表のアクセス

カスタムコントロールからRDB表をアクセスする場合、以下の3種類のファイル編成として使用することが可能です。

- ・ RDB索引順編成ファイル
- ・ RDB相対編成ファイル
- ・ RDB順編成ファイル

### (3) IFAS ファイルのオープン

IFAS ファイルを使用可能な状態にします。オープンモード、アクセスモードの指定方法により、使用できる機能に違いがあります。また、オープンモードと一緒にIFAS ファイルに対する共有／排他(ファイルロックモード)の指定を行います。

#### オープンモード

##### ・ 出力モード

IFAS ファイルを初期化し、新規にデータを作成するためのモードです。  
レコードの書き出しのみ行うことができます。

##### ・ 参照モード

IFAS ファイルのデータを参照するためのモードです。  
レコードの読み込み、検索を行うことができます。

##### ・ 更新モード

IFAS ファイルのデータを参照・更新するためのモードです。  
レコードの読み込み、検索、更新などすべての操作を行うことができます。

##### ・ 追加モード

IFAS ファイルに対し、データを追加するためのモードです。  
レコードの書き出しのみ行うことができます。

#### ファイルロックモード

- ・ **排他モード**

IFAS ファイルをクローズするまでの間、他の利用者との共用を一切禁止します。他の利用者はIFAS ファイルが使用できる状態となるまでOpenメソッドが保留されます。

- ・ **共用モード**

IFAS ファイルに対して、他の利用者との共用を許可します。ファイル内のレコードについてはレコード排他制御(レコードロック)が働きます。他の利用者が共用モード以外で使用する場合、Openメソッドは使用できる状態となるまで保留されることがあります。

- ・ **保護モード**

IFAS ファイルをクローズするまでの間、他の利用者から更新されないように保護します。すなわち、このモードでオープンしたIFAS ファイルに対し、他の利用者は参照モードでしかオープンすることができません。その他のオープンモードで使用する場合、Openメソッドは使用できる状態となるまで保留されます。

#### アクセスモード

- ・ **索引順編成、順アクセス**

IFAS ファイル内のレコードをキー順に順次処理をすることができます。

- ・ **索引順編成、動アクセス**

IFAS ファイル内のレコードをキー順に順次処理したり、キーを指定して「乱呼び出し」することができます。

- ・ **相対編成、順アクセス**

IFAS ファイル内のレコードを相対キー順に順次処理をすることができます。

- ・ **相対編成、動アクセス**

IFAS ファイル内のレコードを相対キー順に順次処理したり、相対キーを指定して「乱呼び出し」することができます。

- ・ **順編成**

IFAS ファイル内のレコードを先頭レコードから順番に順次処理をすることができます。

## (4) レコードの読み込み／検索

レコードイメージの読み込みや読み込みたいレコードへ位置付ける機能を提供します。また、ある条件と合致するレコードをすべて探し出す検索機能があります。参照モードまたは更新モードでオープンしたIFAS ファイルに使用できます。

## (5) レコードの更新

レコードイメージの書き出し、再書き出しおよびレコードを削除する機能を提供します。レコードの書き出し機能は、出力モード、追加モードおよび更新モードの動アクセスでオープンしたIFAS ファイルに使用できます。レコードの再書き出し、削除機能は更新モードでオープンしたIFAS ファイルに使用できます。

## (6) トランザクション機能

アクセスするIFAS ファイルに対しトランザクション処理を行うことを宣言するプロパティを用意しています。IFAS RDB表の場合は表定義ユーティリティで宣言しておくこともできます。IFAS RDB表の指定は、表定義ユーティリティで行うことを推奨します。また、レコードの更新を確定し、トランザクション開始を宣言するためのコミット機能、トランザクション開始を宣言以降に行われた更新を取り消すためのロールバック機能もメソッドとして

提供しています。トランザクション処理中、更新したレコードのロックは解除されず、コミット機能またはロールバック機能を行うと解除されます。

トランザクション処理を行っているIFAS ファイルは、不意の電源ダウンなどが発生した場合、システム再起動時に更新処理途中であったトランザクションを自動的にロールバックするリカバリ機能を使用することができます。詳しくは「6-4-2 システム障害からの復旧」を参照してください。

## (7) 更新ログ収集機能

アクセスするIFAS ファイルに対し、更新ログの収集を宣言するプロパティを用意しています。IFAS RDB表の場合は表定義ユーティリティで宣言しておくこともできます。IFAS RDB表の指定は、表定義ユーティリティで行うことを推奨します。

更新ログの収集を行っていれば、ディスク障害などでIFAS ファイルが使用できなくなってしまう場合、バックアップファイルと更新ログファイルから最新の状態に復旧すること(ロールフォワード)が可能となります。

## 5-1-3 オープンモードと共用制御

共用モードまたは保護モードでオープンしたIFAS ファイルは、レコード操作を行うと同時にそのレコードに対し共用制御が働きます。

ロックされるレコード操作	ロックが解除されるレコード操作
トランザクション処理を行わないIFAS ファイル	
ロック指定の読み込み(Read) 位置付け(Start)	次の読み込み(Read) 次の位置付け(Start) 更新(Write, Rewrite, Delete) IFAS ファイルのクローズ(Close)
トランザクション処理を行うIFAS ファイル	
ロック指定の読み込み(Read) 位置付け(Start)	次の読み込み(Read) 次の位置付け(Start) 更新(Write, Rewrite, Delete) コミット(Commit) ロールバック(Rollback) IFAS ファイルのクローズ(Close)
更新(Write, Rewrite, Delete)	コミット(Commit) ロールバック(Rollback) IFAS ファイルのクローズ(Close)

カスタムコントロールでは、レコードの共用制御は1レコードロック方式を取っており、ロック指定の読み込みや位置付けを行った直後のカレントレコードしかロックされません。ロックされたレコードは、他の利用者からのアクセスが禁止されるため、他の利用者はそのレコードのロックが解除されるまでレコード操作が保留されます。ロックされたレコードのロックが解除されるのは、そのIFAS ファイルに対し別のレコード操作を行うか、またはIFAS ファイルをクローズした時です。

また、トランザクション処理を行っているIFAS ファイルの場合、ロック指定の読み込み、位置付け操作だけではなく、更新を行ったレコードもロックされます。読み込み、位置付け操作によりロックされたレコードは、そのIFAS ファイルに対し別のレコード操作を行うか、またはコミット/ロールバック/IFAS ファイルをクローズした時です。更新操作によりロックされたレコードはコミット、ロールバックまたはIFAS ファイルをクローズするまでロックが解除されません。

IFAS ファイルをカスタムコントロールからアクセスする場合のOS互換モードによる共用制御を以下の表に示します。

表における記号の説明

E	排他モード	RD	Readメソッド
P	保護モード	ST	Startメソッド
S	共用モード	—	その他のメソッド(ロック指定なしReadを含む)
◎	ファイルは共用されレコード排他が働く <sup>注1</sup>		
○	ファイルは共用されるがレコード排他は働かない <sup>注1</sup>		
●	ファイルは共用されレコード排他が働くが、レコード排他が働く機会はない <sup>注1</sup>		
空欄	ファイルは共用されない		

注1. 本共用制御の説明はトランザクション処理を行わない場合に適用されます。トランザクション処理を行っている場合、更新系の操作(Write, Rewrite, Delete)が行われたレコードはCommit/Rollback処理またはClose処理を行うまでレコード排他が働きます。

### オープンモードと共用制御 (A-VX II 互換モード)

先行 後行		参照 モード					更新 モード							出力 モード		追加 モード		
		E	P	S			E	P			S			E	E	P	S	
				—	RD	ST		—	RD	ST	—	RD	ST					
参照 モード	E																	
	P		○	○	○	○												
	S		○	○	○	○		○	◎	◎	○	◎	◎			●	●	
更新 モード	E																	
	P			○	注 2	注 4												
	S			○							○	◎	◎				●	
出力 モード	E																	
追加 モード	E																	
	P			○	注 3	注 5												
	S			○							○	●	●				●	

注2. 既定値では○(レコード排他は働かない)動作ですが、運用支援ツールのCOBOL設定において「INPUTモードのREAD命令でレコードロックする」とした場合、◎(レコード排他が働く)の動作になります。

注3. 既定値では○(レコード排他は働かない)動作ですが、運用支援ツールのCOBOL設定において「INPUTモードのREAD命令でレコードロックする」とした場合、●(レコード排他が働くが、レコード排他が働く機会はない)の動作になります。

注4. 既定値では○(レコード排他は働かない)動作ですが、運用支援ツールのCOBOL設定において「INPUTモードのSTART命令でレコードロックする」とした場合、◎(レコード排他が働く)の動作になります。

注5. 既定値では○(レコード排他は働かない)動作ですが、運用支援ツールのCOBOL設定において「INPUTモードのSTART命令でレコードロックする」とした場合、●(レコード排他が働くが、レコード排他が働く機会はない)の動作になります。

### オープンモードと共用制御 (A-V X 5 互換モード)

先行 後行		参照 モード					更新 モード							出力 モード		追加 モード		
		E	P	S			E	P			S			E	E	P	S	
				—	RD	ST		—	RD	ST	—	RD	ST					
参照 モード	E																	
	P		○	○	○	○												
	S		○	○	○	○		○	◎	○	○	◎	○			●	●	
更新 モード	E																	
	P			○	注 6	○												
	S			○		○					○	◎	○				●	
出力 モード	E																	
追加 モード	E																	
	P			○	注 7	○												
	S			○		○					○	●	○				●	

注6. 既定値では○(レコード排他は働かない)動作ですが、運用支援ツールのCOBOL設定において「INPUTモードのREAD命令でレコードロックする」とした場合、◎(レコード排他が働く)の動作になります。

注7. 既定値では○(レコード排他は働かない)動作ですが、運用支援ツールのCOBOL設定において「INPUTモードのREAD命令でレコードロックする」とした場合、●(レコード排他が働くが、レコード排他が働く機会はない)の動作になります。

## 5-2 動作環境の設定

本説では、カスタムコントロールからIFAS ファイルを使用する場合の各種動作環境の設定について説明します。

### 5-2-1 動作環境の設定

運用支援ツールの[COBOL設定]タブで、カスタムコントロールからの動作設定を行うことが可能です(運用支援ツールについては「7-4-1 運用支援ツール」を参照)。

なお、[COBOL設定]タブの各項目は、COBOL関連製品での共通の設定になっています。  
[COBOL設定]タブを変更した場合、COBOLプログラムや、他の製品(DBリンクキット等)の動作にも影響があるのでご注意ください。

以下に、動作環境の各項目について説明します。

#### (1) OS互換モード

[OS互換モード]ラジオボタンによって動作モードをA-VXⅡ互換とするかA-VX5互換とするかを選択できます。

下表にOS互換モードによる差分動作を示します。

OS互換モードによる差分動作

差分動作	A-VXⅡ	A-VX5
選択型仮想表を出力／更新モードでオープン時	“000”(正常)	“037”エラー
結合型仮想表を出力／更新モードでオープン時	実行時エラー <sup>注1</sup>	“037”エラー
結合型仮想表を更新モードでオープン時	実行時エラー <sup>注1</sup>	“000”(正常) <sup>注3</sup>
ファイルビジー時	ファイルビジー解除を待つ <sup>注2</sup>	“091”エラー
Open メソッド直後の Read(ifasGetPrior)メソッド時	“010”(EOF 検出)	先頭レコードを読む
Select メソッド直後の Read(ifasGetPrior)メソッド時	最終レコードを読む	先頭レコードを読む
Select メソッド直後の Start(ifasGetPrior)メソッド時	最終から 2 番目のレコードに位置付ける	“023”エラー
Start メソッド時	レコードロックする	レコードロックしない
Rewrite(ifasRewriteKey)メソッドで同一キーのレコードが 2 件以上存在時	同一キーの最初のレコードを更新する	“022”エラー
Delete(ifasDeleteKey)メソッドで同一キーのレコードが 2 件以上存在時	同一キーの最初のレコードを削除する	“022”エラー
Read ( ifasGetEqual / ifasGetGreaterThan / ifasGetGreaterEqual)メソッドで指定キーのインデックスが存在しない時	実行時エラー <sup>注1</sup>	“092”エラー
Start ( ifasStartEqual / ifasStartGreaterThan / ifasStartGreaterEqual)メソッドで指定キーのインデックスが存在しない時	実行時エラー <sup>注1</sup>	“092”エラー

出力モードでオープンしたファイルに Select メソッドを行った時	“047”エラー	“092”エラー
Select メソッドを行わずに Scratch メソッドを行った時	“000”(正常)	“092”エラー
相対編成に対して、Start(ifasStartGreaterThan／ifasStartGreaterEqual)メソッドを相対キー=0指定で行った時	先頭レコードに位置付けする	“023”エラー
デッドロック発生時	“09J”エラー	実行時エラー <sup>注1</sup>
ロールバックログ収集エラー時	“09K”エラー	実行時エラー <sup>注1</sup>
ログ収集エラー時	“09T”エラー	実行時エラー <sup>注1</sup>
更新ログファイルオーバーフロー時	“09O”エラー	実行時エラー <sup>注1</sup>
RDB 破壊検出時	“09M”エラー	実行時エラー <sup>注1</sup>
ロックテーブルオーバーフロー時	“09P”エラー	実行時エラー <sup>注1</sup>

注1 カスタムコントロールでは、“0F0”／“0FF”エラーとなります。COBOLアプリケーションや、その他のCOBOL関連製品ではエラーコードが異なります。詳細は各製品の説明書を参照してください。

注2 カスタムコントロールのプロパティの指定によっては“091”エラーが返却される場合があります。

注3 更新系のメソッド(Write／Rewrite／Delete)はエラーになります。

## (2) Openメソッド時のレコードサイズ

[OPEN命令でレコードサイズのチェックをしない]チェックボックスによって、OPEN時のレコードサイズチェックの緩和を指定できます。**カスタムコントロールでは直接影響しません。**チェックボックスをチェックしていない場合、レコードサイズとIFAS ファイルのレコードサイズは一致していなければなりません(一致していない場合はエラーとなります)。チェックボックスをチェックすると、レコードサイズはIFAS ファイルのレコードサイズ以上の値を指定すればOPENが実行されます(小さい値を指定した場合はエラーとなります)。

## (3) 二重キー検出動作

[READ命令で二重キーを検出しない]および[WRITE／REWRITE命令で二重キーを検出しない]チェックボックスによって、二重キーを検出しないようにできます。

チェックボックスをチェックすると、IFAS一般索引順編成ファイル／IFAS RDB索引順編成の、重複キー許可インデックスに対するRead／Write／Rewriteメソッドで二重キーが発生しても、IfasActionStatusプロパティに二重キー(“002”)ステータスが返却されないで正常(“000”)ステータスが返却されるようになります。

なお、重複キー不許可インデックスに対するWrite／Rewriteメソッドでは、本オプションによらずIfasActionStatusプロパティに二重キーエラー(“022”)が返却されます。

## (4) 二重キー検出範囲

[REWRITE命令で変更するキーのみ二重キーを検出する]チェックボックスによって、Rewriteメソッドでの重複キー許可インデックスへの二重キー検出範囲を指定することができます。チェックボックスをチェックすると、IFAS一般索引順編成ファイル／IFAS RDB表索引順編成の重複キー許可インデックスに対するRewriteメソッドで、データの変更があったキーに対してのみ二重キーチェックを行います。

チェックボックスをチェックしないと、すべてのキーに対して二重キーチェックを行います(データの変更がないキーも含まれます)。  
 どちらの場合も、二重キーが検出された場合は、**IfasActionStatus**プロパティに二重キー検出(“002”)ステータスが返却されます。

## (5) 共用制御動作

[INPUTモードのREAD命令でレコードロックする]および[INPUTモードのSTART命令でレコードロックする]チェックボックスによって、参照モードでレコード排他が働くようにすることができます。

チェックボックスをチェックすると、参照モードの**Read**／**Start**メソッドが実行されたレコードは次の命令までレコード排他が働くようになります。

## (6) 索引順編成ファイルのRewrite/Delete

[索引順編成ファイルのREWRITE/DELETEをカレントレコード対象とする]チェックボックスによって、IFAS一般索引順編成ファイル(動呼び出し)の**Rewrite (ifasRewriteKey)**／**Delete (ifasDeleteKey)**メソッドで処理対象となるレコードを選択できます。

チェックボックスをチェックしないと、指定されたキー値に基づいてレコードの**Rewrite**／**Delete**メソッドを実行します。二重キーが存在する場合は、最初のレコードが対象になります(下記例では、レコード1が処理対象)。なお、A-VX5互換モードの場合は、**IfasActionStatus**プロパティに“022”が返却されます。

チェックボックスをチェックすると、下記の条件を両方満たしている時、先の**Read**メソッドで読み込んだレコードが**Rewrite**／**Delete**メソッドの対象レコードとなります(下記例では、レコード3が処理対象)。A-VX5互換モードの場合でも**Rewrite (ifasRewriteKey)**／**Delete (ifasDeleteKey)**は、先の**Read**メソッド命令で読み込んだレコードが対象となります。

- ・ 直前のメソッドが**Read**(保護／共用モードの場合はロック指定あり**READ**)
- ・ 主キー値の変更がない

なお、条件を満たしていない場合は、チェックしていない場合と同じになります。

例 Bが主キーの部分である(この場合すべてB)

3回**Read**メソッド(**ifasGetNext**)実行している(レコード3を入手している)

B	レコード 1
B	レコード2
B	レコード3
B	レコード4

## (7) ファイル／表削除時のRFDファイル

[ファイル／表削除時、RFDファイルを削除する]チェックボックスによって、COBOL関連製品でIFAS ファイルを削除すると同時に、RFDファイルを削除することができます。**カスタムコントロール**では直接影響しません。

チェックボックスをチェックすると、IFAS ファイル削除時、RFDファイルを削除します。

なお、ファイル管理ユーティリティでIFAS一般ファイルを削除する場合は、RFDファイルは削除されません。また、表定義ユーティリティでIFAS RDB表を削除する場合は、表定義ユーティリティのオプションに従います。



## (8) COMP-5／BINARY互換モード

[COMP-5／BINARYの仕様をCOBOL85モードにする]チェックボックスによって、IFAS ファイルの列に指定するCOMP-5／BINARYの仕様を決定することができます。

チェックボックスをチェックすると、COMP-5／BINARYの仕様をCOBOL85に合わせます。

チェックボックスをチェックしないと、IFAS FILEの独自仕様になります。

カスタムコントロールでは、IFAS FILEの独自仕様のCOMP-5/BINARYを列に持つIFAS ファイルは使用できません。必ずチェックしてください。

## (9) CLOSEメソッドのトランザクション動作

[CLOSE命令時のトランザクション動作]ラジオボタンによって、Closeメソッドで暗黙的にCOMMITするかROLLBACKするかを選択できます。

なお、プログラム終了時にオープンされたままのファイルは自動的にクローズされますが、トランザクション処理中のファイルは、本オプションの設定にかかわらず、自動的にロールバックされてからクローズされます。

## (10) 主キーが重複キー許可属性の場合の動作

[RECORD KEYで指定したキーが重複キー許可属性の場合]ラジオボタンによって、Openメソッドのaccesskeyオプションで指定されたキーに重複キー許可属性があった場合、正常とするか実行時エラーとするかを選択できます。

## (11) Openメソッド時のキー更新

[ALTERNATE RECORD KEYを指定しない場合]ラジオボタンによって、OpenメソッドでIFAS ファイルに定義されているすべてのキーを更新するか、accesskeyオプションで指定されたキーのみ更新するかを選択できます。

## (12) Selectメソッドでのソート処理メモリサイズ

[SELECT命令でのソート処理用メモリサイズ]の項目で、IFAS RDB表に対するSelectメソッドで確保するメモリサイズをKBで指定します。

最小値はSelectメソッドで確保する最小メモリサイズを指定します。最小分のメモリが確保できなかった場合、Selectメソッドはエラーとなります。

最大値はSelectメソッドで確保する最大メモリサイズを指定します。最大分のメモリが確保できなかった場合は、サイズを小さくてメモリを確保します。

最小値/最大値が0の場合、以下の表のレコード数をもとにメモリを自動取得します。

- ・ 基本表の場合→基本表
- ・ 選択型仮想表→選択型仮想表を構成する基本表
- ・ 結合型仮想表→プライマリ表として構成する基本表
- ・ 合併型仮想表→第一プライマリ表として構成する基本表

なお、必要となるメモリの見積もりは、レコード数(conditionオプションの条件式を満たす件数)、および、ソートキー(sortkeyオプション)によって異なりますが、およそ以下の計算式で計算できます。

$$(a \times 2 + 4) \times b \times 1.5$$

a = **sortkey**オプションで指定された項目の合計サイズ

b = **condition**オプションがある場合は、指定された条件式を満足するレコード数  
**condition**オプションがない場合は、各表のレコード数

## 5-2-2 更新ログファイル設定

カスタムコントロールからIFAS ファイルを更新した場合の更新ログは、運用支援ツール(「7-4-1 運用支援ツール」を参照)の[更新ログ設定]タブにおける[更新ログファイルパス]に指定されているファイルに出力されます。なお、本項目はCOBOL関連製品での共通設定になっています。

更新ログファイルの運用に関しては、「6-3 更新ログファイルの運用」を参照してください。

## 5-3 利用方法

ここでは、MS-Visual Basicからカスタムコントロールを利用する方法について説明します。

### 5-3-1 各種資源

カスタムコントロールを使用するための各種資源を記述します。

資源種別	資源名
コンポーネント	ifas32 OLE Control module
クラス名	Ifas32Ctrl
付属モジュール	ifas32.bas IFAS FILE インストールディレクトリ¥Vb¥ifas32.bas

### 5-3-2 利用方法

#### (1) IFAS ファイルの作成

カスタムコントロールを使用する前に、IFAS ファイルを作成しておかなければなりません。

- IFAS一般ファイルは、「ファイル管理ユーティリティ」等で作成してください(「7-1-1 ファイル管理ユーティリティ」参照)。索引順編成ファイルの場合は、インデックスファイルも必要です。この時、インデックスファイル名はCOBOL仕様に合わせてください。なお、索引順編成ファイルの場合は、インデックスの作成も必要です。  
また、IFAS一般ファイルとは別に、「RFD編集ツール」にてRFDファイルを作成してください(「7-1-4 RFD編集ツール」参照)。RFDファイルは、データファイルと同じディレクトリ配下に、ファイル名.rfd(ファイル名に識別子がある場合は識別子を除いたファイル名.rfd)という名前で作成してください。なお、「RFD編集ツール」でRFDファイルを作成する時に設定する「サーバコンピュータ名」および「フォーマット名」は既定値を使用してください。
- IFAS RDB表は、「表定義ユーティリティ」で、RDB表を作成してください(「7-2-1 表定義ユーティリティ」参照)。  
RDB索引順編成の場合は、インデックスの作成も必要です。

#### (2) コンポーネント／付属モジュールの追加

MS-Visual Basicのプロジェクトに、コンポーネント「ifas32 OLE Control module」、および、付属モジュール「ifas32.bas」を追加して、本コントロールを貼り付けてください。  
本コントロールは複数個、貼り付けることができます。また、1つのコントロールで複数のIFASファイルをアクセスすることができます。

### (3) カレントアクセスキーとキーの省略値

#### カレントアクセスキー

カレントアクセスキーとは、索引順編成に対するReadメソッドのgetmode=ifasGetNext／ifasGetPrior／ifasGetFirst／ifasGetLast指定時に、それぞれ順方向／逆方向／最小値／最大値のレコードを読み込むためのキーのことを言います。

たとえば、

	カラム1	カラム2
レコード番号1	1	3
レコード番号2	2	2
レコード番号3	3	1

となるファイル(カラム1／カラム2ともキーとして定義されている)に対して、レコード番号2に位置付けされているとします。この場合、カレントアクセスキーがカラム1／カラム2のどちらに設定されているかによって、Read(getmode=ifasGetNext)で入手するレコードが異なります。

- ・ カレントアクセスキーがカラム1の場合は、レコード番号3を入手
- ・ カレントアクセスキーがカラム2の場合は、レコード番号1を入手

カレントアクセスキーは、IFAS ファイルをOpenする時に指定するaccesskeyのキーに初期設定されます。その後、下記の各メソッドで指定したaccesskeyのキーに変更されます。

- ・ Read (getmode=ifasGetEqual／ifasGetGreaterThan／ifasGetGreaterEqual指定)
- ・ Start

なお、Openメソッドのaccesskeyで指定した項目を主キー、それ以外のキーを副キーと呼ぶこともあります。

#### キー値の省略

索引順編成に対するRead／Rewrite／Delete／Startの各メソッドでは、accesskeyを省略することができます。省略した場合、同じIFAS ファイルに対する直前のOpen／Read／Rewrite／Delete／Startで指定したaccesskeyが有効になります。

なお、Openメソッドでは、accesskey省略不可です。

なお、各メソッドの指定に関しては、「5-5 メソッド解説」参照してください。

### (4) データ操作

カスタムコントロールでは、Open／Close／Read／Write等、IFAS ファイルをアクセスするための各種メソッドを提供しています。

ただし、MS-Visual Basic上でデータイメージの参照／更新をするためには、二段階の操作が必要になります。これは、IFAS ファイルのデータ型にはMS-Visual Basicで扱えないデータ型があるためです。データはいったんカスタムコントロール内のレコード領域に保持し、そのレコード領域に対してフィールド単位にデータアクセスし、必要ならばデータ型の変換を行います。

実例を示します。例内の各種メソッドについては、「5-5 メソッド解説」を参照してください。  
例は、以下のカラムを持つIFAS一般ファイルC:\IFAS\IFASFILE (Open済み) に対して、各機能を実施します。

カラム名	データ型(サイズ)
COLM1	COMP-1
COLM2	COMP-3 符号あり(4 桁)
COLM3	LEAD(5 桁)
COLM4	文字(10 桁)

注 COLM2がインデックスになっている

なお、メソッド内で使用するいくつかの変数の定義をします。

```
Dim FILE As String
Dim VarColumn1
Dim VarColumn2
Dim VarColumn3
Dim VarColumn4
Dim Accesskey As String
```

```
FILE = " C:\IFAS\IFASFILE"
VarColumn1 = "COLM1"
VarColumn2 = "COLM2"
VarColumn3 = "COLM3"
VarColumn4 = "COLM4"
Accesskey = "COLM2"
```

また、オブジェクト式はIfas321とします。

#### ・ データの出力

**SetColumnDataString**メソッドで各カラムにデータをセットした後、**Write**メソッドでレコードを出力。

```
Ifas321.SetColumnDataSiring FILE, VarColumn1,"1"
Ifas321.SetColumnDataSiring FILE, VarColumn2,"-30"
Ifas321.SetColumnDataSiring FILE, VarColumn3,"+200"
Ifas321.SetColumnDataSiring FILE, VarColumn4,"abcdefg"
Ifas321.Write FILE
```

#### ・ データの入力

**Read**メソッドでレコードを入力した後、**GetColumnDataString**メソッドで各カラムのデータ値入手(シンボルColumnImage nにCOLMnのデータが入る)。

```
Ifas321.Read FILE,ifasGetNext, ifasReadNoLock
Ifas321.GetColumnDataString FILE, VarColumn1,ColumnImage1,ColumnAttribute
Ifas321.GetColumnDataString FILE, VarColumn2,ColumnImage2,ColumnAttribute
Ifas321.GetColumnDataString FILE, VarColumn3,ColumnImage3,ColumnAttribute
Ifas321.GetColumnDataString FILE, VarColumn4,ColumnImage4,ColumnAttribute
```

### ・ 特定キーのデータの入力

SetIndexAccessKeyStringメソッドで、COLM2にキー値をセットした後、Readメソッドでレコードを入力(この後、各カラムのデータを入力)。

```
Ifas321.SetIndexAccessKeyString FILE, VarColumn2,"10"
```

```
Ifas321.Read FILE,ifasGetEqual, ifasReadLock, Accesskey
```

## (5) レコード単位の操作

カスタムコントロールでは、1回の操作でデータイメージの参照／更新を可能とする以下のメソッドを用意しています。

- ・ ReadRecord
- ・ ReadRecordString
- ・ WriteRecord
- ・ WriteRecordString
- ・ RewriteRecord
- ・ RewriteRecordString
- ・ DeleteRecord
- ・ DeleteRecordString
- ・ StartRecord
- ・ StartRecordString

実例を示します。例内の各種メソッドについては、「5-5 メソッド解説」を参照してください。例は、以下のカラムを持つIFAS一般ファイルC:\¥IFAS¥IFASFILE (Open済み) に対して、各機能を実施します。

カラム名	データ型(サイズ)
COLM1	COMP-1
COLM2	COMP-3 符号あり(4 桁)
COLM3	LEAD(5 桁)
COLM4	文字(10 桁)

注 COLM2がインデックスになっている

なお、メソッド内で使用するいくつかの変数の定義をします。

```
Dim FILE As String
Dim Accesskey As String
Dim record
Dim columnatri
Dim varrec(1 To 4)
Dim strrec(1 To 4) As String
Dim attribute(1 To 4) As Integer
```

```
FILE = " C:\¥IFAS¥IFASFILE"
Accesskey = "COLM2"
```

また、オブジェクト式はIfas321とします。

### ・データの出力

recordで示された配列に各カラムのデータイメージをセットし、WriteRecord／WriteRecordStringメソッドでファイルに出力する。

```
record = varrec
record(1) = 1
record(2) = "2"
record(3) = "10"
record(4) = "ABCD"
Ifas321.WriteRecord FILE, record
'COLM1=1／COLM2=2／COLM3=10／COLM4="ABCD"となる
```

```
record = strrec
record(1) = "2"
record(2) = "1"
record(3) = "101"
record(4) = "efghijk"
Ifas321.WriteRecordString FILE, record
'COLM1=2／COLM2=1／COLM3=101／COLM4="efghijk"となる
```

### ・データの入力

ReadRecord／ReadRecordStringメソッドを実行することにより、入力したレコードイメージがrecordで示された配列に各カラム単位にセットされる。

```
columnnatri = attribute
record = varrec
Ifas321.ReadRecord FILE, record, columnnatri, ifasGetNext
'record(1) = 2／record(2) = "1"／record(3) = "101"／record(4) = "efghijk"
'columnnatri(1)=ifasColumnInteger／columnnatri(2)=ifasColumnNumeric／
'columnnatri(3)=ifasColumnDecimal／columnnatri(4) = ifasColumnChar
'となる
```

```
record = strrec
Ifas321.ReadRecordString FILE, record, columnnatri, ifasGetNext
'record(1) = "1"／record(2) = "2"／record(3) = "10"／record(4) = "ABCD"となる
```

### ・特定キーのデータの入力

COLM2に対応する領域(record(2))にキー値をセットした後、ReadRecordStringメソッドを実行することにより、キー値に対応したレコードを入力する。また、入力したレコードイメージがrecordで示された配列に各カラム単位にセットされる。

```
record = strrec
record(2) = "2"
Ifas321.ReadRecordString FILE, record, columnnatri, ifasStartEqual, , Accesskey
'record(1) = "1"／record(2) = "2"／record(3) = "10"／record(4) = "ABCD"となる
```

## (6) IFAS ファイル定義情報の入手

カスタムコントロールでは、データ操作のメソッドの他に、IFAS ファイルの定義情報を入手するメソッドを提供しています。  
以下にIFAS ファイル定義情報の入手方法例を記述します。

IFAS一般ファイルC:\IFAS\IFASFILEは以下のカラム／インデックスで構成されているとします。

カラム名	データ型(サイズ)
COLM1	COMP-1
COLM2	COMP-3 符号あり(4 桁)
COLM3	LEAD(5 桁)
COLM4	文字(10 桁)

注 COLM2は単一キー(重複なし)となっている  
(COLM3,COLM4)は合成キー(重複あり)となっている

なお、メソッド内で使用するいくつかの変数定義をしておきます。

```
Dim FILE As String
FILE = " C:\IFAS\IFASFILE"
```

また、オブジェクト式はIfas321とします。

### ① IFAS ファイルの定義情報入手

GetTableDefメソッドにて、IFAS ファイルの定義情報を入手します。

```
Ifas321.GetTableDef FILE,table,column,record,index
```

各返却値は以下のとおりになります。

オプション	返却値
table	ifasTableTypeIfasIndex(一般索引順編成ファイル)
column	4(項目数)
record	20(レコードサイズ)
index	2(インデックス数)

### ② IFAS ファイルの索引定義情報入手

GetIndexDefメソッドにて、IFAS ファイルの索引定義情報を入手します。

```
Ifas321.GetIndexDef FILE,index,attri,column
```

各返却値は以下のとおりになります。なお、indexは1～2まで指定できます。

index=1の場合

オプション	返却値
attri	ifasIndexSingle(重複なし)
column	1(インデックスを構成する項目数)

index=2の場合

オプション	返却値
attri	ifasIndexDuplicate(重複あり)
column	2(インデックスを構成する項目数)



### ③ IFAS ファイルの索引構成定義情報入手

GetIndexColumnDefメソッドにて、IFAS ファイルの索引を構成している項目の定義情報を入手します。

Ifas321.GetIndexColumnDef FILE,index,column,name,attri,length,decpoint  
index=1の場合columnは1まで、index=2の場合columnは1～2まで指定します。  
各返却値は以下のとおりになります。

index=1 / column=1の場合

オプション	返却値
name	COLM2
attri	ifasColumnNumeric(内部10進数)
length	4(桁数)
decpoint	0(小数点桁数)

index=2 / column=1の場合

オプション	返却値
name	COLM3
attri	ifasColumnDecimal(外部10進数)
length	5(桁数)
decpoint	0(小数点桁数)

index=2 / column=2の場合

オプション	返却値
name	COLM4
attri	ifasColumnChar(文字)
length	10(桁数)
decpoint	0(小数点桁数)

### ④ IFAS ファイルの項目定義情報入手

GetColumnDefメソッドにて、IFAS ファイルを構成している項目の定義情報を入手します(GetColumnDefメソッドを使用するために前もってファイルをオープンしておきます)。

Ifas321.GetColumnDef FILE,column,name,attri,length,decpoint  
columnは1～4まで指定します。  
各返却値は以下のとおりになります。

オプション	返却値			
	column=1	column=2	column=3	column=4
name	COLM1	COLM2	COLM3	COLM4
attri	単精度整数	内部10進数	外部10進数	文字
length		4	5	10
decpoint		0	0	0

## (7) ファイルの指定

カスタムコントロールの多くのメソッドには、IFAS ファイル名を指定します。

IFAS ファイル名の指定法式を記述します。

- ・ **IFAS一般ファイルの場合**

[<サーバ名>]ファイル名

- ・ **IFAS RDB表の場合**

[<サーバ名>][データベース名.]表名

<サーバ名>は、クライアント・サーバ機能を使用する時に指定します(「6-1 クライアント・サーバ機能」参照)。  
<サーバ名>を省略した場合は、ローカルマシン上のIFAS ファイルが対象となります。

データベース名を省略した場合は、データベースの既定値が有効になります。

また、環境設定ファイルを用いて、アプリケーション毎に、ソースに記述されているファイル名／表名と実ファイル名／実表名の関連付けを行ったり、各種オプションを変更することができます。  
環境設定ファイルについては、「2-3 環境設定ファイル」を参照してください。

## (8) サンプルプログラム

IFAS FILEインストールディレクトリ¥VB¥SAMPLE¥ADDRESS配下に、カスタムコントロールの利用方法を理解していただくためのサンプルプログラム(アドレス帳)、Readme.txtを提供しています。  
カスタムコントロール利用時の参考としてください(詳細はReadme.txtを参照してください)。

### 5-3-3 ファイル編成と入出力命令

利用可能なファイル編成と入出力命令を以下に示します。

実行可能な入出力命令

ファイル編成			一般順編成 RDB順編成				一般相対編成 RDB相対編成				一般索引順編成 RDB索引順編成			
オープンモード メソッド			参照	出力	更新	追加	参照	出力	更新	追加	参照	出力	更新	追加
ファイル呼出方	順 ア ク セ ス	Read	○	×	○	×	○	×	○	×	○	×	○	×
		Write	×	○	×	○	×	○	×	○	×	○	×	○
		Rewrite	×	×	○	×	×	×	○	×	×	×	○	×
		Start	○	×	○	×	○	×	○	×	○	×	○	×
		Delete	×	×	○	×	×	×	○	×	×	×	○	×
		Select	○	×	○	×	○	×	○	×	○	×	○	×
		Scratch	○	×	○	×	○	×	○	×	○	×	○	×
		Commit	○	○	○	○	○	○	○	○	○	○	○	○
		Rollback	○	○	○	○	○	○	○	○	○	○	○	○
	動 ア ク セ ス	Read	—	—	—	—	○	×	○	—	○	×	○	—
		Write	—	—	—	—	×	○	○	—	×	○	○	—
		Rewrite	—	—	—	—	×	×	○	—	×	×	○	—
		Start	—	—	—	—	○	×	○	—	○	×	○	—
		Delete	—	—	—	—	×	×	○	—	×	×	○	—
		Select	—	—	—	—	○	×	○	—	○	×	○	—
		Scratch	—	—	—	—	○	×	○	—	○	×	○	—
		Commit	—	—	—	—	○	○	○	—	○	○	○	—
		Rollback	—	—	—	—	○	○	○	—	○	○	○	—

### 5-3-4 IFAS ファイル使用時の注意事項

#### (1) IFAS一般ファイル／IFAS RDB表共通

- IFAS一般索引順編成ファイル／RDB索引順編成表を使用する場合、各メソッドのアクセスキー（accesskeyオプション）で指定する列は、あらかじめ各種ツールでインデックスとして定義しておく必要があります。

#### (2) 選択型仮想表

- OS互換モードがA-VX5モードの場合、出力／追加モードでのオープンはできません。
- 選択型仮想表にWriteメソッドを実行した場合、仮想表に選択されなかった基本表の項目には数値属性ならば0、文字属性ならばスペースが書き出されます。本対象となる項目が二重キー許可のないインデックスとして定義されている場合、この仮想表に対しWriteメソッド命令を行ってはいけません。キー値の重複が起こり、Writeメソッド命令が失敗します。

### (3) 結合型仮想表

- 出力／追加モードでのオープンできません。
- 更新モードでオープンした場合、**Write**／**Rewrite**／**Delete**メソッドによる更新はできません。
- RDB索引順編成として使用する場合、各メソッドのアクセスキー(**accesskey**オプション)で指定するカラムはプライマリ表にインデックスとして定義されている列のみとなります。
- RDB索引順編成として使用する場合、二重キー検出のため**Read**メソッドに時間がかかる場合があります。これを回避したい場合、**DupKeyReadMode**プロパティに**ifasDupKeyReadSkip**(二重キーを検出しない)を設定してください。これにより、多少の性能改善が図れます。ただし、**IfasActionStatus**プロパティには“002”が返却されません。
- RDB相対編成ファイルとして使用する場合、相対キーはプライマリ表のレコード番号となります。したがって、アクセスキーであるインデックスに重複するキー値が存在するような場合、同じ相対キーを持つレコードが複数存在することになります。



### (4) 合併型仮想表

- 出力／追加モードでオープンできません。
- RDB索引順編成として使用する場合、各メソッドのアクセスキー(**accesskey**オプション)で指定する列は、元となるすべてのプライマリ表のキーとして定義しておく必要があります。
- RDB相対編成としてアクセスできません。
- Write**メソッドは実行できません。
- Rewrite**メソッドでは重複キーの検出が行われません。したがって、重複キーが発生するような更新を行った場合でも**IfasActionStatus**プロパティには“002”が返却されません。

## 5-4 プロパティ解説

カスタムコントロールは以下のようなプロパティを提供します。

IfasActionStatus	カスタムコントロールのステータスを返します。
IfasMinorStatus1	IFAS FILE、IFASPRO RDB本体の詳細ステータスを返却します。
IfasMinorStatus2	IFAS FILE、IFASPRO RDB本体の内部ステータスを返却します。
IfasErrState	エラーステートメントの動作を設定します。
IfasErrDupKeyState	二重キー検出時のエラーステートメントの動作を設定します。
IfasErrEofState	EOF検出時のエラーステートメントの動作を設定します。
ErrTableName	エラー発生メソッドで指定したIFAS ファイル名を返却します。
ErrMethod	エラーが発生したメソッド名を返却します。
DupKeyWriteMode	Write/Rewriteメソッドで二重キー検出の動作を設定します。
DupKeyReadMode	Readメソッドで二重キー検出の動作を設定します。
TransactionMode	IFAS ファイルのトランザクション処理の動作を設定します。
LogMode	IFAS ファイルの更新ログを保存する／しないの動作を設定します。
BufferNumber	IFAS FILEのバッファ数を設定します。
Separator	アクセスキー、ソートキーの区切り記号を設定します。
PatternMatch	Selectメソッドの検索条件式で一致する文字列のワイルドカードの文字(1文字)を設定します。
PatternMask	Selectメソッドの検索条件式でマスクして一致する文字列のワイルドカードの文字(1文字)を設定します。
BusyReturnMode	オープンの時ビジーリターンするかしないかを設定します。

なお、説明中の構文内で、[]で囲まれているオプションは省略可能であることを示しています。

## IfasActionStatusプロパティ

カスタムコントロールのステータスをString型3桁で返却します。値の取得のみ可能です。  
本プロパティの値(コード)により、各メソッドの実行結果を確認できます。なお、コードの詳細は  
「付録A-5 カスタムコントロールのステータス」を参照してください。

On Errorステートメントはこのプロパティのコードがエラーを意味する場合(ifasActionNormal="000"以外)に動作します(ifasErrStateプロパティがTrueである場合、IfasErrDupKeyState、IfasErrEofStateの設定も関係します)。

### 構文

---

Object.IfasActionStatus

### 指定項目

---

指定項目	内容
Object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。

## IfasMinorStatus1プロパティ

IFAS FILE、IFASPRO RDBの詳細ステータスをString型5桁で返却します。値の取得のみ可能です。なお、詳細ステータスについては「付録A-4 COBOL利用時のステータス」を参照してください。

詳細ステータスは、IfasActionStatus="0xx"(IFAS FILEエラー)のエラーが発生した場合に、本プロパティに返却します。

### 構文

---

Object.IfasMinorStatus1

### 指定項目

---

指定項目	内容
Object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。

## IfasMinorStatus2プロパティ

IFAS FILE、IFASPRO RDBの内部ステータスをString型5桁で返却します。値の取得のみ可能です。なお、内部ステータスについては「付録A-3 C言語関数のステータス」を参照してください。

IfasMinorStatus1=“01002”／“01006”／“01010”の場合に、詳細ステータスの左16ビットを内部ステータスとして、本プロパティに返却します。

---

### 構文

Object.IfasMinorStatus2

---

### 指定項目

指定項目	内容
Object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。



## IfasErrStateプロパティ

エラーステートメントの動作を設定します。

IfasActionStatusプロパティが“000”以外の場合は、本プロパティにTrueに設定していればエラーステートメントが動作します。

IfasActionStatusプロパティが“000”以外であっても、本プロパティにFalseに設定していればエラーステートメントは動作しません。

### 構文

---

```
object.IfasErrState [=value]
```

### 指定項目

---

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
value	エラーステートメントの動作を設定します。設定する値については次の「設定値」を参照してください。

### 設定値

---

設定値	内容
True	(既定値)エラーステートメントは動作します。
False	エラーステートメントは動作しません。

## IfasErrDupKeyStateプロパティ

二重キー検出時のエラーステートメントの動作を設定します。

IFAS一般索引順編成ファイル/IFAS RDB表索引順編成の、重複キー許可インデックスに対するRead/Write/Rewriteメソッドで、二重キーが検出された場合(**IfasActionStatus**プロパティが“002”)、本プロパティが**True**に設定されていればエラーステートメントが動作します。

二重キーが検出されても、本プロパティが**False**に設定されていればエラーステートメントは動作しません。

**IfasErrState**プロパティに**False**が設定されていれば、本プロパティに限らずどのようなステータスでもエラーステートメントは動作しません。

なお、エラーステートメントが動作しない場合でも**IfasActionStatus**プロパティに“002”を返却します。

重複キー不許可インデックスに対するWrite/Rewriteメソッドで、二重キーエラーが検出された場合(**IfasActionStatus**プロパティが“022”)のエラーステートメントは本プロパティに無関係に動作します。

### 構文

---

```
object.IfasErrDupKeyState [=value]
```

### 指定項目

---

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
value	エラーステートメントの動作を設定します。設定する値については次の「設定値」を参照してください。

### 設定値

---

設定値	内容
True	二重キー検出時にエラーステートメントは動作します。
False	(既定値)二重キー検出時にエラーステートメントは動作しません。

## IfasErrEofStateプロパティ

EOF検出時のエラーステートメントの動作を設定します。

EOFが検出された場合(**IfasActionStatus**プロパティが“010”)、本プロパティが**True**に設定されていればエラーステートメントが動作します。

EOFが検出されても、本プロパティが**False**に設定されていればエラーステートメントは動作しません。

**IfasErrState**プロパティに**False**が設定されていれば、本プロパティに限らずどのようなステータスでもエラーステートメントは動作しません。

なお、エラーステートメントが動作しない場合でも**IfasActionStatus**プロパティに“010”を返却します。

### 構文

---

```
object.IfasErrEofState [=value]
```

### 指定項目

---

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
value	エラーステートメントの動作を設定します。設定する値については次の「設定値」を参照してください。

### 設定値

---

設定値	内容
True	(既定値) EOF検出時にエラーステートメントは動作します。
False	EOF検出時にエラーステートメントは動作しません。

## ErrTableNameプロパティ

エラー発生メソッドで指定したIFAS ファイル名をString型で返却します。値の取得のみ可能です。

### 構文

---

```
object.ErrTableName
```

### 指定項目

---

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。

## ErrMethodプロパティ

エラーが発生したメソッド名をString型で返却します。値の取得のみ可能です。

### 構文

---

```
object.ErrMethod
```

### 指定項目

---

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。

## DupKeyWriteModeプロパティ

IFAS一般索引順編成ファイル／IFAS RDB索引順編成の重複キー許可インデックスに対するWrite／Rewriteメソッドで、二重キーを検出するかしないかの動作を設定します。

本プロパティにifasDupKeyWriteSkipをセットした場合、重複キー許可インデックスに対するWrite／Rewriteメソッドで二重キーが検出されても、IfasActionStatusプロパティに“002”が返却されず、正常 (IfasActionStatusプロパティに“000”が返却) となります。

本プロパティにifasDupKeyWriteExecをセットした場合、重複キー許可インデックスに対するWrite／Rewriteメソッドで二重キーが検出されればIfasActionStatusプロパティに“002”が返却されます。

なお、重複キー不許可インデックスに対するWrite／Rewriteメソッドで、二重キーエラーが検出された場合は本プロパティに関係なくIfasActionStatusプロパティに“022”が返却されます。

### 構文

```
object.DupKeyWriteMode [= value]
```

### 指定項目

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
value	モードを設定します。詳細は「設定値」を参照してください。

### 設定値

設定値	値	内容
ifasDupKeyWriteDef	0	(既定値)運用支援ツールのCOBOL設定に従う
ifasDupKeyWriteSkip	1	二重キーを検出しない
ifasDupKeyWriteExec	2	二重キーを検出する

### 規則

IFAS ファイルのオープン(Openメソッド)時に設定していた値が、そのIFAS ファイルのアクセスにおいて有効となります。オープン後にこのプロパティを変更しても動作は変更しません。

## DupKeyReadModeプロパティ

IFAS一般索引順編成ファイル／IFAS RDB表索引順編成の重複キー許可インデックスに対するReadメソッドで、二重キーを検出するかしないかの動作を設定します。

本プロパティにifasDupKeyReadSkipをセットした場合、重複キー許可インデックスに対するReadメソッドで二重キーが検出されても、IfasActionStatusプロパティに“002”が返却されず、正常（IfasActionStatusプロパティに“000”が返却）となります。

本プロパティにifasDupKeyReadExecをセットした場合、重複キー許可インデックスに対するReadメソッドで二重キーが検出されればIfasActionStatusプロパティに“002”が返却されます。

### 構文

---

```
object.DupKeyReadMode [= value]
```

### 指定項目

---

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
value	モードを設定します。詳細は「設定値」を参照してください。

### 設定値

---

設定値	値	内容
ifasDupKeyReadDef	0	(既定値)運用支援ツールのCOBOL設定に従う
ifasDupKeyReadSkip	1	二重キーを検出しない
ifasDupKeyReadExec	2	二重キーを検出する

### 規則

---

IFAS ファイルのオープン(Openメソッド)時に設定していた値が、そのIFAS ファイルのアクセスにおいて有効となります。オープン後にこのプロパティを変更しても動作は変わりません。

## TransactionModeプロパティ

IFAS ファイルのトランザクション処理をする／しないの動作を設定します。  
本プロパティにifasTransactionSkipが設定されていれば、トランザクション処理をしません。  
本プロパティにifasTransactionExecが設定されていれば、トランザクション処理をします。

### 構文

---

```
object.TransactionMode [= value]
```

### 指定項目

---

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
value	モードを設定します。詳細は「設定値」を参照してください。

### 設定値

---

設定値	値	内容
ifasTransactionDef	0	(既定値) IFAS一般ファイルの場合はトランザクション処理しない IFAS RDB表の場合は表の定義に従う
ifasTransactionSkip	1	トランザクション処理をしない
ifasTransactionExec	2	トランザクション処理をする

### 規則

---

IFAS ファイルのオープン(Openメソッド)時に設定していた値が、そのIFAS ファイルのアクセスにおいて有効となります。オープン後にこのプロパティを変更しても動作は変更しません。



## LogModeプロパティ

IFAS ファイルの更新ログを保存する／しないの動作を設定します。  
本プロパティにifasLogSkipが設定されていれば、更新ログを保存しません。  
本プロパティにifasLogExecが設定されていれば、更新ログを保存します。

### 構文

---

```
object.LogMode [= value]
```

### 指定項目

---

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
value	モードを設定します。詳細は「設定値」を参照してください。

### 設定値

---

設定値	値	内容
ifasLogDef	0	(既定値) IFAS一般ファイルの場合は更新ログを保存しない IFAS RDB表の場合は表の定義に従う
ifasLogSkip	1	更新ログを保存しない
ifasLogExec	2	更新ログを保存する

### 規則

---

IFAS ファイルのオープン(Openメソッド)時に設定していた値が、そのIFAS ファイルのアクセスにおいて有効となります。オープン後にこのプロパティを変更しても動作は変わりません。

## BufferNumberプロパティ

IFAS ファイルのバッファ数をInteger型で設定します。  
1バッファには1レコード分セットされます。

### 構文

---

```
object.BufferNumber [= value]
```

### 指定項目

---

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
value	(既定値=10)バッファ数を指定します。

### 規則

---

- (1) IFAS ファイルのオープン(**Open**メソッド)時に設定していた値が、そのIFAS ファイルのアクセスにおいて有効となります。オープン後にこのプロパティを変更しても動作は変更しません。
- (2) バッファ数は最小=10、最大=500までです。設定値が、10より小さい場合は10、500より大きい場合は500とします。

## Separatorプロパティ

アクセスキー／ソートキーに項目を複数指定する場合の、項目間の区切り記号をString型で設定します。

### 構文

---

```
object.Separator [= value]
```

### 指定項目

---

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
value	区切り記号を文字列式で設定します。

### 規則

---

既定値は“.”です。

## PatternMatchプロパティ

Selectメソッドの検索条件式に指定する(前、中、後に)一致する文字列のワイルドカードの文字(1文字)をString型で設定します。

指定例は、「5-5 メソッド解説」内のSelectメソッドの説明内を参照してください。

### 構文

---

```
object.PatternMatch [= value]
```

### 指定項目

---

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
value	ワイルドカードとなる文字(1文字)を文字列式で設定します。

### 規則

---

既定値は“\*”です。

## PatternMaskプロパティ

Selectメソッドの検索条件式に指定するマスクして一致する文字列のワイルドカードの文字(1文字)をString型で設定します。

指定例は、「5-5 メソッド解説」内のSelectメソッドの説明内を参照してください。

### 構文

---

```
object.PatternMask [= value]
```

### 指定項目

---

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
value	ワイルドカードとなる文字(1文字)を文字列式で設定します。

### 規則

---

既定値は“?”です。

## BusyReturnModeプロパティ

Openメソッドの時ビジーリターンする／しないの動作を設定します。

本プロパティでTrueを設定した場合、他で使用中のIFAS ファイルに対してOpenメソッドを行うと、ビジーリターンしてIfasActionStatusプロパティに“091”を返却します。

本プロパティでFalseを設定した場合、他で使用中のIFAS ファイルに対してOpenメソッドを行うと、IFAS ファイルが使用できるようになるまで待ちます。

### 構文

---

```
object.BusyReturnMode [= value]
```

### 指定項目

---

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
value	ファイルビジーの時の動作を設定します。設定する値については次の「設定値」を参照してください。

### 設定値

---

設定値	内容
True	Openメソッドで指定したIFAS ファイルが他で使用中の場合、ビジーリターンします。
False	(既定値) Openメソッドで指定したIFAS ファイルが使用できるようになるまで待ちます。

## 5-5 メソッド解説

カスタムコントロールは以下のようなメソッドを提供します。

Open	IFAS ファイルをオープンします。
Close	IFAS ファイルをクローズします。
Read	レコードイメージをIFAS ファイルから読み込みます。
Write	レコードイメージをIFAS ファイルに書き出します。
Rewrite	レコードイメージをIFAS ファイルに再書き出しします。
Delete	IFAS ファイルからレコードを削除します。
Start	IFAS ファイルのレコードに位置付けします。
Select	IFAS ファイルの部分集合の作成とソートを行います。
Scratch	<b>Select</b> メソッドで作成した部分集合を解放します。
Commit	現時点まで更新を確定しレコードロックを解除します。
Rollback	現時点までの更新を取り消しレコードロックを解除します。
Flush	IFAS ファイルに対して仮クローズを行います。
GetColumnData	項目値を取得します。
GetColumnDataString	項目値を取得します。
SetColumnData	項目値を設定します。
SetColumnDataString	項目値を設定します。
SetIndexAccessKey	索引順編成ファイルのアクセスキーの値を設定します。
SetIndexAccessKeyString	索引順編成ファイルのアクセスキーの値を設定します。
ReadRecord	レコードイメージをIFAS ファイルから読み込みます。
ReadRecordString	レコードイメージをIFAS ファイルから読み込みます。
WriteRecord	レコードイメージをIFAS ファイルに書き出します。
WriteRecordString	レコードイメージをIFAS ファイルに書き出します。
RewriteRecord	レコードイメージをIFAS ファイルに再書き出しします。
RewriteRecordString	レコードイメージをIFAS ファイルに再書き出しします。
DeleteRecord	IFAS ファイルからレコードを削除します。
DeleteRecordString	IFAS ファイルからレコードを削除します。
StartRecord	IFAS ファイルのレコードに位置付けします。
StartRecordString	IFAS ファイルのレコードに位置付けします。
GetTableDef	IFAS ファイルの定義情報を取得します。
GetIndexDef	IFAS ファイルの索引定義情報を取得します。
GetIndexColumnDef	IFAS ファイルの索引を構成する項目の情報を取得します。
GetIndexInfoDef	IFAS ファイルの索引を構成する項目の情報を取得します。
GetColumnDef	IFAS ファイルの項目定義情報を項目毎に取得します。
GetRecordDef	IFAS ファイルの項目定義情報を項目毎に取得します。
Term	カスタムコントロールの動作環境の終了宣言を行います。

なお、説明中の構文内で、[]で囲まれているオプションは省略可能であることを示しています。

## Openメソッド

IFAS ファイルをオープンします。

### 構文

```
object.Open tablename,openmode,accessmode [,accesskey]
```

### 指定項目

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。
openmode	指定方法の詳細は、「5-3-2 利用方法」を参照してください。 必ず指定します。オープンモードを整数または定数で指定します。詳細については「設定値」を参照してください。
accessmode	必ず指定します。アクセスモードを整数または定数で指定します。詳細については「設定値」を参照してください。
accesskey	索引順編成の場合は、キーとなる項目名を文字列で必ず指定してください(省略不可)。 相対編成の場合は最大相対キーを文字列で指定します。(パラメータを省略した場合は0を指定したことになります)。 順編成の場合はパラメータを省略してください(指定しても無視します)。

### 設定値

#### openmodeの設定値

設定値	値	内容
ifasOpenOutput	&H22	出力モード
ifasOpenExInput	&H21	排他参照モード
ifasOpenExUpdate	&H23	排他更新モード
ifasOpenExExtend	&H24	排他追加モード
ifasOpenShInput	&H01	共用参照モード
ifasOpenShUdate	&H03	共用更新モード
ifasOpenShExtend	&H04	共用追加モード
ifasOpenPrInput	&H11	保護参照モード
ifasOpenPrxUpdate	&H13	保護更新モード
ifasOpenPrExtend	&H14	保護追加モード

#### accessmodeの設定値

設定値	値	内容
ifasAccessSeq	&H00	IFAS RDB表順編成
ifasAccessRelSeq	&H01	IFAS RDB表相対編成、順アクセス



ifasAccessRelRnd	&H02	IFAS RDB表相対順編成、動アクセス
ifasAccessIndexSeq	&H80	IFAS RDB表索引順編成、順アクセス
ifasAccessIndexRnd	&H81	IFAS RDB表索引順編成、動アクセス
ifasAccessIfasSeq	&H08	IFAS一般ファイル順編成
ifasAccessIfasRelSeq	&H09	IFAS一般ファイル相対編成、順アクセス
ifasAccessIfasRelRnd	&H0A	IFAS一般ファイル相対編成、動アクセス
ifasAccessIfasIndexSeq	&H88	IFAS一般ファイル索引順編成、順アクセス
ifasAccessIfasIndexRnd	&H89	IFAS一般ファイル索引順編成、動アクセス

## 規則

---

- (1) Openメソッドが成功すると、IFAS ファイルが使用可能な状態になります。
- (2) openmode/accessmodeの設定値については、「5-1-2 データアクセス機能」/「5-1-3 オープンモードと共用制御」を参照してください。
- (3) 索引順編成の場合、accesskeyの指定は必須です(省略できません)。
- (4) 索引順編成で、accesskeyに合成キー(複数の項目からなるインデックス)を指定する場合は、Separatorプロパティで指定した区切り文字で項目名同士を連結した文字列を指定します。  
 例 Separator". "で、項目名aとbを連結  
 指定文字列は"a.b"

## Closeメソッド

OpenされているIFAS ファイルをクローズします。

### 構文

---

```
object.Close tablename
```

### 指定項目

---

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。 指定方法の詳細は、「5-3-2 利用方法」を参照してください。

### 規則

---

- (1) Closeメソッドが成功すると、IFAS ファイルが使用不可能な状態になります。

## Readメソッド

IFAS ファイルからレコードイメージを読み込みます。

### 構文

```
[relaccesskey =]  
object.Read tablename,getmode[,lockmode][,accesskey]
```

### 指定項目

指定項目	内容
relaccesskey	相対編成の場合にアクセスしたレコードの相対キーを返却します(Long型)。
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。
getmode	指定方法の詳細は、「5-3-2 利用方法」を参照してください。 必ず指定します。レコード読み込みモードを指定します。詳細については「設定値」を参照してください。
lockmode	ロックモードを指定します(省略可)。詳細については「設定値」を参照してください。
accesskey	索引順編成で特定のキーをもとに読み込む場合 (getmode=ifasGetEqual／ifasGetGreaterThan／ifasGetGreaterEqual)、キーとなる項目名を文字列で指定します(省略化)。 相対編成でgetmode=ifasGetEqual(特定相対キーの読み込み)場合は、読み込むレコードの相対キーを文字列で必ず指定します。特定相対キーの読み込み以外の場合は省略します(指定しても無視します)。 順編成の場合はパラメータを省略します(指定しても無視します)。

### 設定値

getmodeの設定値

設定値	値	内容
ifasGetNext	&H2	カレント情報に基づき順方向にレコードを読み込みます。 なお、索引順編成の場合は、カレントアクセスキーの順方向になります。
ifasGetPrior	&H3	カレント情報に基づき逆方向にレコードを読み込みます。 なお、索引順編成の場合は、カレントアクセスキーの逆方向になります。
ifasGetFirst	&H0	索引順編成の場合は、カレントアクセスキーの最小値のレコードを読み込みます。

ifasGetLast	&H1	順編成の場合は、先頭レコードを読み込みます。 索引順編成の場合は、カレントアクセスキーの最大値のレコードを読み込みます。
ifasGetEqual	&H6	順編成の場合は、最後のレコードを読み込みます。 索引順編成の場合は、指定キーの値と等しい値のレコードを読み込みます。 相対編成の場合は、指定相対キーを持つレコードを読み込みます。
ifasGetGreaterThan	&H4	索引順編成の場合は、指定キーの値よりも大きな値のレコードを読み込みます。
ifasGetGreaterEqual	&H5	索引順編成の場合は、指定キーの値よりも大きい、または等しい値のレコードを読み込みます。

lockmodeの設定値

設定値	値	内容
ifasReadNoLock	0	ロック指定なしRead(既定値)
ifasReadLock	1	ロック指定ありRead

## 規則

- (1) Readメソッドが成功すると、レコードイメージはカスタムコントロール内に読み込まれます。レコードの各項目値を入手するためには、GetColumnDataメソッドまたはGetColumnDataStringメソッドを使用してください。
- (2) Openメソッドのアクセスモードと、Readメソッドのレコード読み出しモードの組み合わせを記述します(○が許可)。

ファイル種別	IFAS一般ファイル					IFAS RDB表				
	順	相対		索引順		順	相対		索引順	
アクセスモード	順	順	動	順	動	順	順	動	順	動
ifasGetNext	○	○	○	○	○	○	○	○	○	○
ifasGetPrior	○	○	○	○	○	○	○	○	○	○
ifasGetFirst	○			○	○	○			○	○
ifasGetLast	○			○	○	○			○	○
ifasGetEqual			○		○			○		○
ifasGetGreaterThan					○					○
ifasGetGreaterEqual					○					○

- (3) 索引順編成で、getmode=ifasGetEqual/ifasGetGreaterThan/ifasGetGreaterEqualにて読み込む前に、SetIndexAccessKeyメソッドまたはSetIndexAccessKeyStringメソッドにて指定キーに値をセットしておく必要があります。
- (4) lockmodeにifasReadLockを指定した場合は、Readメソッドが成功すると入力したレコードにロックがかかります。  
lockmodeにifasReadNoLockを指定した場合は、Readメソッドが成功しても入力したレコードにロックがかかりません。  
詳細は、「5-1-3 オープンモードと共用制御」を参照してください。
- (5) 索引順編成で、accesskeyに合成キー(複数の項目からなるインデックス)を指定する場合は、Separatorプロパティで指定した区切り文字で項目名同士を連結した文字列を指定します。

## Writeメソッド

レコードイメージをIFAS ファイルに書き出します。

### 構文

---

```
[relaccesskey =]  
object.Write tablename[,accesskey]
```

### 指定項目

---

指定項目	内容
relaccesskey	相対編成の場合にアクセスしたレコードの相対キーを返却します(Long型)。
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。
accesskey	指定方法の詳細は、「5-3-2 利用方法」を参照してください。 索引順編成または順編成の場合はパラメータを省略します(指定しても無視します)。 相対編成で動アクセスの場合は、書き出すレコードの相対キーを文字列で必ず指定します。順アクセスの場合は省略します(指定しても無視します)。

### 規則

---

- (1) **Write**メソッドが成功すると、カスタムコントロール内のレコードイメージがIFAS ファイルに書き出されます。なお、**Write**メソッド実行前に、あらかじめ、**SetColumnData**メソッドまたは**SetColumnDataString**メソッドにて、カスタムコントロール内のイメージに項目値を設定しておく必要があります。
- (2) 相対編成／順アクセスで、出力モードの場合、最初のWriteレコードの相対キーは1となり、以降、2、3、... となります。  
相対編成／順アクセスで、追加モードの場合、最初のWriteレコードの相対キーは、Open前の最大相対キー(存在するレコード内で最大の相対キー)の次の相対キーとなり、以降、その次の相対キーとなります。
- (3) 索引順編成／順アクセスの場合、**Open**メソッドで指定した**accesskey**のキー項目値は、昇順になるように設定しなければなりません。

## Rewriteメソッド

レコードイメージをIFAS ファイルに再書き出します(書き換え)。

### 構文

```
[relaccesskey =]  
object.Rewrite tablename,rewritemode[,accesskey]
```

### 指定項目

指定項目	内容
relaccesskey	相対編成の場合にアクセスしたレコードの相対キーを返却します(Long型)。
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。
rewritemode	指定方法の詳細は、「5-3-2 利用方法」を参照してください。 必ず指定します。更新モードを整数または定数で指定します。詳細については「設定値」を参照してください。
accesskey	索引順編成でrewritemode=ifasRewriteKey(特定アクセスキーの書き換え)の場合は、書き換えるレコードのキー値をセットした項目名を文字列で指定します(省略化)。rewritemode=ifasRewriteCurrent(カレントレコードの書き換え)の場合は省略します(指定しても無視します)。 相対編成でrewritemode=ifasRewriteKey(特定相対キーの書き換え)の場合は、書き換えるレコードの相対キーを文字列で必ず指定します。rewritemode=ifasRewriteCurrent(カレントレコードの書き換え)の場合は省略します(指定しても無視します)。 順編成の場合はパラメータを省略します(指定しても無視します)。

### 設定値

rewritemodeの設定値

設定値	値	内容
ifasRewriteCurrent	&H09	カレントレコードを書き換えます。
ifasRewriteKey	&H49	アクセスキーのレコードを書き換えます。

### 規則

- (1) Rewriteメソッドが成功すると、IFAS ファイルのレコードはカスタムコントロール内のレコードイメージに書き換わります。なお、Rewriteメソッド実行前に、あらかじめ、SetColumnDataメソッドまたはSetColumnDataStringメソッドにて、カスタムコントロール内のイメージの項目値を設定しておく必要があります。
- (2) Openメソッドのアクセスモードと、Rewriteメソッドの更新モードの組み合わせを記述

します(○が許可)。

ファイル種別	IFAS一般ファイル					IFAS RDB表				
	順	相対		索引順		順	相対		索引順	
アクセスモード	順	順	動	順	動	順	順	動	順	動
ifasRewriteCurrent	○	○	○	○	○	○	○	○	○	○
ifasRewriteKey			○		○			○		○

- (3) **rewriteMode=ifasRewriteCurrent**にて書き換える場合、あらかじめ**Read**メソッド(共用／保護モードの場合はロック指定あり)でレコードを読み込んでおかなければなりません。
- (4) 索引順編成で、**rewriteMode=ifasRewriteKey**にて書き換える場合、あらかじめ**SetIndexAccessKey**メソッドまたは**SetIndexAccessKeyString**メソッドにて指定キーに値をセットしておく必要があります。
- (5) 索引順編成／順アクセスの場合、カレントアクセスキーの値は変更できません。
- (6) 索引順編成で、**accessKey**に合成キー(複数の項目からなるインデックス)を指定する場合は、**Separator**プロパティで指定した区切り文字で項目名同士を連結した文字列を指定します。

## Deleteメソッド

IFAS ファイルからレコードを削除します。

### 構文

```
[relaccesskey =]  
object.Delete tablename,deletemode[,accesskey]
```

### 指定項目

指定項目	内容
relaccesskey	相対編成の場合にアクセスしたレコードの相対キーを返却します(Long型)。
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。
deletemode	指定方法の詳細は、「5-3-2 利用方法」を参照してください。 必ず指定します。削除モードを整数または定数で指定します。詳細については「設定値」を参照してください。
accesskey	索引順編成でdeletemode=ifasDeleteKey(特定アクセスキーの削除)の場合は、削除するレコードのキー値をセットした項目名を文字列で指定します(省略化)。deletemode=ifasDeleteCurrent(カレントレコードの削除)の場合は省略します(指定しても無視します)。 相対編成でdeletemode=ifasDeleteKey(特定相対キーの削除)の場合は、削除するレコードの相対キーを文字列で必ず指定します。deletemode=ifasDeleteCurrent(カレントレコードの削除)の場合は省略します(指定しても無視します)。 順編成の場合はパラメータを省略します(指定しても無視します)。

### 設定値

deletemodeの設定値

設定値	値	内容
ifasDeleteCurrent	&H29	カレントレコードを削除します。
ifasDeleteKey	&H69	アクセスキーのレコードを削除します。

### 規則

- (1) Deleteメソッドが成功すると、IFAS ファイルのレコードが削除されます。
- (2) Openメソッドのアクセスモードと、Deleteメソッドの削除モードの組み合わせを記述します(○が許可)。



ファイル種別	IFAS一般ファイル					IFAS RDB表				
	順	相対		索引順		順	相対		索引順	
アクセスモード	順	順	動	順	動	順	順	動	順	動
ifasDeleteCurrent	○	○	○	○	○	○	○	○	○	○
ifasDeleteKey			○		○			○		○

- (3) **deletemode=ifasDeleteCurrent**にて削除する場合、あらかじめ**Read**メソッド(共用／保護モードの場合はロック指定あり)でレコードを読み込んでおかなければなりません。
- (4) 索引順編成で、**deletemode=ifasDeleteKey**にて削除する場合、あらかじめ**SetIndexAccessKey**メソッドまたは**SetIndexAccessKeyString**メソッドにて指定キーに値をセットしておく必要があります。
- (5) 索引順編成で、**accesskey**に合成キー(複数の項目からなるインデックス)を指定する場合は、**Separator**プロパティで指定した区切り文字で項目名同士を連結した文字列を指定します。
- (6) ExcelのVBAで、**Delete**メソッドを使用することはできません。同じインターフェースを持つ**DeleteX**メソッドを使用してください。

## Startメソッド

IFAS ファイル内のレコードを位置付けします。

### 構文

---

```
[relaccesskey =]  
object.Start tablename,startmode[,accesskey]
```

### 指定項目

---

指定項目	内容
relaccesskey	相対編成の場合にアクセスしたレコードの相対キーを返却します(Long型)。
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。
startmode	指定方法の詳細は、「5-3-2 利用方法」を参照してください。 必ず指定します。位置付けモードを指定します。詳細については「設定値」を参照してください。
accesskey	索引順編成では、位置付けを行うキーの項目名を文字列で指定します(省略化)。 相対編成でstartmode=ifasStartEqual／ifasStartGreaterThan／ifasStartGreaterEqual(特定相対キーの位置付け)場合は、位置付けするレコードの相対キーを文字列で必ず指定します。特定相対キーの読み込み以外の場合は省略します(指定しても無視します)。 順編成の場合はパラメータを省略します(指定しても無視します)。

### 設定値

---

startmodeの設定値

設定値	値	内容
ifasStartNext	&H2	カレント情報に基づき順方向にレコードを位置付けます。 なお、索引順編成の場合は、指定キーの順方向になります。
ifasStartPrior	&H3	カレント情報に基づき逆方向にレコードを位置付けます。 なお、索引順編成の場合は、指定キーの逆方向になります。
ifasStartFirst	&H0	索引順編成の場合は、指定キーの最小値のレコードに位置付けます。 順編成／相対編成の場合は、先頭レコードに位置付けします。

ifasStartLast	&H1	索引順編成の場合は、指定キーの最大値のレコードに位置付けます。 順編成／相対編成の場合は、最後のレコードに位置付けます。
ifasStartEqual	&H6	索引順編成の場合は、指定キーの値と等しい値のレコードに位置付けます。 相対編成の場合は、指定相対キーを持つレコードに位置付けます。
ifasStartGreaterTan	&H4	索引順編成の場合は、指定キーの値より大きい値を持つレコードに位置付けます。 相対編成の場合は、指定相対キーより大きい相対キーを持つレコードに位置付けます。
ifasStartGreaterEqual	&H5	索引順編成の場合は、指定キーの値より大きいか、または等しい値を持つレコードに位置付けます。 相対編成の場合は、指定相対キーより大きいか、または、等しい相対キーを持つレコードに位置付けます。

## 規則

- (1) **Start**メソッドが成功すると、IFAS ファイル内の特定レコードに位置付けされます。
- (2) **Open**メソッドのアクセスモードと、**Start**メソッドの位置付けモードの組み合わせを記述します(○が許可)。

ファイル種別	IFAS一般ファイル					IFAS RDB表				
編成	順	相対		索引順		順	相対		索引順	
アクセスモード	順	順	動	順	動	順	順	動	順	動
ifasStartNext	○			○	○	○			○	○
ifasStartPrior	○			○	○	○			○	○
ifasStartFirst	○	○	○	○	○	○	○	○	○	○
ifasStartLast	○	○	○	○	○	○	○	○	○	○
ifasStartEqual		○	○	○	○		○	○	○	○
ifasStartGreaterTan		○	○	○	○		○	○	○	○
ifasStartGreaterEqual		○	○	○	○		○	○	○	○

- (3) 索引順編成で、startmode=ifasStartEqual／ifasStartGreaterTan／ifasStartGreaterEqualにて読み込む前に、SetIndexAccessKeyメソッドまたはSetIndexAccessKeyStringメソッドにて指定キーに値をセットしておく必要があります。
- (4) 索引順編成で、accesskeyに合成キー(複数の項目からなるインデックス)を指定する場合は、Separatorプロパティで指定した区切り文字で項目名同士を連結した文字列を指定します。

## Selectメソッド

条件を満足するレコードの部分集合の作成とソートを行います。

### 構文

```
sortrecordcount As Long =  
object.Select tablename[,condition][,sortkey]
```

### 指定項目

指定項目	内容
sortrecordcount	必ず指定します。選択条件を満たしたレコードの件数を返します。
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。 指定方法の詳細は、「5-3-2 利用方法」を参照してください。
condition	検索条件式を指定します。最大255バイトです。指定方法の詳細は「検索条件式の指定方法」を参照してください。
sortkey	このパラメータを省略すると全レコードを含んだ部分集合を作ります。 ソートキーにする項目名の文字列と、項目の値を昇順に並べるか降順に並べるかを指定します。項目名+ならば昇順、 項目名-ならば降順となります。最大16個の組み合わせをSeparatorプロパティで設定した文字列式で連結した文字列で指定します。 例 Separator". "で、項目名aを昇順、項目bを降順で連結 指定文字列は"a+.b-" このパラメータを省略するとソートしません。

### 検索条件式の指定方法

Selectメソッドのconditionオプションで指定する条件式は、次の形式で指定します。

なお、条件式内の項目名、定数、比較演算子、論理演算子、カッコは、**1文字以上の1バイト空白文字**で区切って指定します。

なお、指定方法内の{ }, [ ]は以下を意味します。

{ }	:	複数の条件式を記述する場合に{ }内のいずれか1つを必ず選択
[ ]	:	[ ]内は省略可

### 条件式

```

      AND
単純条件1 [ { OR } 単純条件2 ... { OR } 単純条件n ]
      XOR
      XOR
```

さらに、単純条件または条件式の否定形を指定できます。

**NOT 単純条件** :単純条件の否定

**NOT (条件式)** :条件式の否定

条件の評価の順を明確にするために、カッコ( )を使用することができます。

### 単純条件の指定方法

```

項目名 { =
        !=
        =< } 定数
        =>
        <
        >

```

- ・ 項目名と定数の指定位置は、逆でも指定できます。
- ・ 単純条件において、項目名により指定する項目と定数の字類は同一でなければなりません。
- ・ 比較の対象となる項目が数字項目の時は、定数の整数部、小数部の桁数が表定義時に指定した範囲内になければなりません。  
定数に1バイト空白文字が含まれる場合は、引用符(“)で項目値を囲んで指定します。引用符なしで空白文字が含まれる項目値を設定した場合は、条件式の解析を中断します。項目属性が、文字の場合は、定数を引用符で囲んで設定することをお勧めします。
- ・ 特殊文字または予約語を定数に指定する場合は、引用符(“)で囲みます。また引用符を定数に指定する場合は(”)のように2つ並べて指定します。
- ・ 次に特殊文字と予約語を示します。  
特殊文字    >、>=、=>、<、<=、=<、!=、=、”  
予約語      AND、OR、XOR、NOT

### 比較演算子

```

=          等しい
!=         等しくない
<          小さい
<=、=<    小さいか等しい
>          大きい
>=、=>    大きいか等しい

```

- ・ 比較の対象となる項目が日本語項目の時比較演算子として = または != だけが指定できます。

### マスク検索

項目の属性が文字または日本語の項目に対しては、前方一致、中間一致、後方一致、部分一致によるマスク検索が可能です。ただし、この場合に使用できる比較演算子は = のみです。

単純条件の定数に指定できます。

#### 前方一致の定数の形式

項目値の先頭が指定した文字列であるレコードを抽出します。

サーチする文字列の最後にPatternMatchプロパティで設定した文字を**1文字**指定します。

#### 中間一致の定数の形式

項目値に指定した文字列を含むレコードを抽出します。

サーチする文字列の前後にPatternMatchプロパティで設定した文字を**1文字ずつ**指定します。

#### 後方一致の定数の形式

項目値の末尾が指定した文字列であるレコードを抽出します。

サーチする文字列の先頭にPatternMatchプロパティで設定した文字を**1文字**指定します。

#### 部分一致の定数形式

項目値の特定位置の文字が、指定した文字と一致するレコードを抽出します。

一致する必要のない文字位置にPatternMaskプロパティで設定した文字を**バイト数分**指定します。

### 条件式の記述例

(1)住所 = 東京都\*

住所の先頭が“東京都”であるレコードを抽出します。

例 **東京都**港区××、**東京都**中央区××など。

“京都府東京都”は先頭が“**東京都**”でないため抽出されません。

(2)名前 = \*谷川\*

名前の中に“谷川”があるレコードを抽出します。

例 長**谷川**太郎、**谷川**花子など

(3)名前 = \*太郎

名前の最後が“太郎”であるレコードを抽出します。

例 山田**太郎**、鈴木**太郎**など。

“山田太郎衛門”は最後が“**太郎**”でないため抽出されません。

(4)名前 = ??田??

名前の3～4バイト目(日本語は1文字2バイト)が“田”であるレコードを抽出します。

例 山**田**太郎、原**田**次郎など。

“田川太郎”、“山本田朗”は、3～4バイト目が“**田**”でないため抽出されません。

(5)体重 =< 58

体重が、58以下のレコードを抽出します。

(6)住所 = 大阪府\* AND ( 体重 > 50 OR 体重 < 40 )

住所が大阪府xxxで、体重が50より多いかまたは40より少ないレコードを抽出します。

(7)( 身長 > 160 OR 身長 < 150 ) AND ( 体重 > 50 OR 体重 < 40 )

以下の条件を両方満たすレコードが抽出されます。

- ・ 身長は160より大きい、または150より小さい
- ・ 体重は50より多い、または40より少ない

### 規則

- (1) **Select**メソッドが成功すると、IFAS ファイル内から条件式を満たすレコードを、ソートキー順にソートした部分集合を作成します。  
条件式を省略すると全レコードを含んだ部分集合を作ります。  
ソートキーを省略した場合は、ソートされません。部分集合内のレコード順は不定となります。
- (2) **Select**メソッドで部分集合を作成すると、以後そのIFAS ファイルに対して、**Scratch**メソッドを実行するか、**Open**メソッドでオープンし直さないかぎり、部分集合に対す

る参照のみが可能になります。

- (3) **Select**メソッドで作成した部分集合に対しては、順アクセスのみが可能になります。また、部分集合に対する各メソッドのは以下のとおりになります。
- **Read**メソッド  
作成された部分集合に対して、ソートされた順(ソートキーが省略された場合は順番不定)に**Read (IfasGetNext／IfasGetPrior)**メソッドで入力できます。
  - **Start**メソッド  
作成された部分集合に対して、ソートされた順(ソートキーが省略された場合は順番不定)に**Start (IfasStartNext／IfasStartPrior)**メソッドで入力できます。
  - **Write**メソッド  
元のIFAS ファイルのみに追加され、**Select**メソッドで作成した部分集合には反映されません。
  - **Rewrite**メソッド  
作成された部分集合に対して、**Rewrite (IfasRewriteCurrent)**メソッドで書き換えができます。
  - **Delete**メソッド  
作成された部分集合に対して、**Delete (IfasDeleteCurrent)**メソッドでレコード削除ができます。
- (4) **Select**メソッドを複数回実行することにより、最初に作成された部分集合の部分集合(条件の絞り込み)を作成することができます。
- (5) 作成した部分集合に条件式を満たすレコードが1件もなかった場合は、次の**Read**メソッドでEOFを検出(**IfasActionStatus**プロパティが“010”)します。
- (6) ExcelのVBAで、**Select**メソッドを使用することはできません。同じインターフェースを持つ**SelectX**メソッドを使用してください。

## Scratchメソッド

Selectメソッドで作成した部分集合を解放します。

### 構文

---

```
object.Scratch tablename
```

### 指定項目

---

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。 指定方法の詳細は、「5-3-2 利用方法」を参照してください。

### 規則

---

- (1) Scratchメソッドが成功すると、Selectメソッドで確保したIFAS ファイルの部分集合を解放します。  
Scratchメソッド後は、元のIFAS ファイルに対して各種処理を行います。



## Commitメソッド

直前までのレコード更新要求を有効とし、トランザクション処理開始の宣言と、レコードロックの解除を行います。

### 構文

---

```
object.Commit
```

### 指定項目

---

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。

### 規則

---

- (1) **Commit**メソッドが成功すると、直前のトランザクション処理開始宣言以降、すべての更新要求が有効になります。合わせて、更新したレコードに対するのロック解除も行います。  
また、**新規にトランザクション処理開始を宣言**します。

## Rollbackメソッド

直前の静止点から現時点までのレコード更新要求を無効にし、トランザクション処理開始の宣言と、レコードロックの解除を行います。

### 構文

---

```
object.Rollback
```

### 指定項目

---

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。

### 規則

---

- (1) Rollbackメソッドが成功すると、直前のトランザクション処理開始宣言以降、すべての更新要求が無効になります。合わせて、更新したレコードに対するのロック解除も行います。  
また、**新規にトランザクション処理開始を宣言**します。
- (2) Rollbackメソッドを実行すると、カレント情報は不定となります。

## Flushメソッド

IFAS ファイルに対する追加／更新／削除を行った時、ファイル復旧を容易にするために、仮クローズ(ディスクへの書き出し)を行います。

### 構文

---

```
object.Flush tablename
```

### 指定項目

---

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。 指定方法の詳細は、「5-3-2 利用方法」を参照してください。

### 規則

---

- (1) Flushメソッドが成功すると、Openメソッドまたは最後に行ったFlushメソッド以降の更新メソッド(Write／Rewrite／Delete)に対して、仮クローズ処理を行います。  
仮クローズ処理を実施すると、バッファ上のデータがディスクへ書き出されます。
- (2) Flushメソッドを実行しても、カレント情報は影響ありません。
- (3) トランザクション指定のないIFAS ファイルに対してFlushメソッドを実行すると、レコード排他制御は解除されます。
- (4) トランザクション指定のIFAS ファイルに対してFlushメソッドを実行すると、参照によるレコード排他制御は解除されますが、更新によるレコード排他制御は解除されません。更新によるレコード排他制御はCommit／Rollbackメソッドにより解除されます。

## GetColumnDataメソッド

カスタムコントロール内のレコードイメージから項目値を取得します。

### 構文

```
object.GetColumnData tablename,column,columnimage,columnattribute
```

### 指定項目

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。 指定方法の詳細は、「5-3-2 利用方法」を参照してください。
column	必ず指定します(バリエーション型)。項目名をString型で指定するか、IFAS ファイルの先頭からの項目番号(1からn)をInteger型で指定してください。該当項目の項目値を取得します。
columnimage	必ず指定します(バリエーション型)。項目値を返します。
columnattribute	必ず指定します(Integer型)。項目の属性を返します。属性の内容は設定値を参照してください。

### 設定値

設定値	値	内容
ifasColumnChar	0	文字
ifasColumnJpnChar	1	日本語
ifasColumnInteger	2	単精度整数
ifasColumnLong	3	倍精度整数
ifasColumnDecimal	4	外部10進数
ifasColumnNumeric	5	内部10進数
ifasColumnBinary	6	2進数

### 規則

- (1) GetColumnDataメソッドにより、カスタムコントロール内のレコードイメージから、columnで指定した項目の項目値をcolumnimageに返却します。
- (2) 各属性毎の、項目値のMS-Visual Basicでの型は次のとおりです。  
文字、日本語、外部10進数、内部10進数、2進数=String  
単精度整数=Integer  
倍精度整数=Long
- (3) 数字のString型では、上位の“0”はゼロサプレスして返却します。  
小数点以下の値は項目の桁数分返却します。  
マイナスの値の場合は“-”を付加します。

100の場合(総桁数=5)は“100”を返却します。“00100”ではありません。  
100.100の場合(総桁数=6/小数点桁数=3)は“100.100”を返却します。“10  
0.1”ではありません。  
-100の場合は“-100”を返却します。

## GetColumnDataStringメソッド

カスタムコントロール内のレコードイメージから項目値を取得します。

### 構文

---

```
object.GetColumnDataString tablename,column,columnimage,columnattribute
```

### 指定項目

---

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。 指定方法の詳細は、「5-3-2 利用方法」を参照してください。
column	必ず指定します(バリエーション型)。項目名をString型で指定するか、IFAS ファイルの先頭からの項目番号(1からn)をInteger型で指定してください。該当項目の項目値を取得します。
columnimage	必ず指定します(String型)。項目値を返します。
columnattribute	必ず指定します(Integer型)。項目の属性を返します。属性の内容は設定値を参照してください。

### 設定値

---

設定値	値	内容
ifasColumnChar	0	文字
ifasColumnJpnChar	1	日本語
ifasColumnInteger	2	単精度整数
ifasColumnLong	3	倍精度整数
ifasColumnDecimal	4	外部10進数
ifasColumnNumeric	5	内部10進数
ifasColumnBinary	6	2進数

### 規則

---

- (1) GetColumnDataStringメソッドは、項目値を常にString型で返す以外は、GetColumnDataメソッドと同じです。

## SetColumnDataメソッド

項目値をカスタムコントロール内のレコードイメージへ設定します。

### 構文

---

```
object.SetColumnData tablename,column,columnimage
```

### 指定項目

---

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。
column	指定方法の詳細は、「5-3-2 利用方法」を参照してください。 必ず指定します(バリエーション型)。項目名をString型で指定するか、IFAS ファイルの先頭からの項目番号(1からn)をInteger型で指定してください。該当項目の項目値を設定します。
columnimage	必ず指定します(バリエーション型)。項目値を指定します。

### 規則

---

- (1) SetColumnDataメソッドにより、columnimageで指定した値を、columnで指定した項目の項目値としてカスタムコントロール内のレコードイメージへ設定します。
- (2) 各属性毎の、項目値のMS-Visual Basicでの型は次のとおりです。  
文字、日本語、外部10進数、内部10進数、2進数=String  
単精度整数=Integer  
倍精度整数=Long
- (3) 数字のString型では、上位の“0”はゼロサプレスしてなくてもかまいません。  
小数点以下の最後の“0”はなくてもかまいません。  
マイナスの値の場合は“-”を付加します。  
100の場合(総桁数=5)は“100”でも、“00100”でもかまいません。  
100.100の場合(総桁数=6/小数点桁数=3)は“100.100”でも、“100.1”でもかまいません。  
-100の場合は“-100”を指定します。

## SetColumnDataStringメソッド

項目値をカスタムコントロール内のレコードイメージへ設定します。

### 構文

---

```
object.SetColumnDataString tablename,column,columnimage
```

### 指定項目

---

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。
column	指定方法の詳細は、「5-3-2 利用方法」を参照してください。 必ず指定します(バリエーション型)。項目名をString型で指定するか、IFAS ファイルの先頭からの項目番号(1からn)をInteger型で指定してください。該当項目の項目値を設定します。
columnimage	必ず指定します(String型)。項目値を指定します。

### 規則

---

- (1) SetColumnDataStringメソッドは、項目値を常にString型で指定する以外はSetColumnDataメソッドと同じです。



## SetIndexAccessKeyメソッド

索引順編成のアクセスキー値を、カスタムコントロール内のレコードイメージへ設定します。

### 構文

```
object.SetIndexAccessKey tablename,column,columnimage
```

### 指定項目

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。
column	指定方法の詳細は、「5-3-2 利用方法」を参照してください。 必ず指定します(バリエーション型)。項目名をString型で指定するか、IFAS ファイルの先頭からの項目番号(1からn)をInteger型で指定してください。該当項目の項目値を設定します。
columnimage	必ず指定します(バリエーション型)。項目値を指定します。

### 規則

- (1) **SetIndexAccessKey**メソッドにより、**columnimage**で指定した値を、**column**で指定した項目のアクセスキーとし、カスタムコントロール内のレコードイメージへ設定します。
- (2) 各属性毎の、項目値のMS-Visual Basicでの型は次のとおりです。  
文字、日本語、外部10進数、内部10進数、2進数=String  
単精度整数=Integer  
倍精度整数=Long
- (3) 数字のString型では、上位の“0”はゼロサプレスしてなくてもかまいません。  
小数点以下の最後の“0”はなくてもかまいません。  
マイナスの値の場合は“-”を付加します。  
100の場合(総桁数=5)は“100”でも、“00100”でもかまいません。  
100.100の場合(総桁数=6/小数点桁数=3)は“100.100”でも、“100.1”でもかまいません。  
-100の場合は“-100”を指定します。

## SetIndexAccessKeyStringメソッド

索引順編成のアクセスキー値を、カスタムコントロール内のレコードイメージへ設定します。

### 構文

---

```
object.SetIndexAccessKeyString tablename,column,columnimage
```

### 指定項目

---

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。
column	指定方法の詳細は、「5-3-2 利用方法」を参照してください。 必ず指定します(バリエーション型)。項目名をString型で指定するか、IFAS ファイルの先頭からの項目番号(1からn)をInteger型で指定してください。該当項目の項目値を設定します。
columnimage	必ず指定します(String型)。項目値を指定します。

### 規則

---

- (1) SetIndexAccessKeyStringメソッドは、項目値を常にString型で指定する以外はSetIndexAccessKeyメソッドと同じです。

## ReadRecordメソッド

IFAS ファイルからレコードイメージを読み込み、属性に従った形式で値を返却します。

### 構文

```
[relaccesskey =]  
object.ReadRecord tablename,record,columnattribute,getmode [,lockmode][,accesskey]
```

### 指定項目

指定項目	内容
relaccesskey	相対編成の場合にアクセスしたレコードの相対キーを返却します(Long型)。
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。
record	指定方法の詳細は、「5-3-2 利用方法」を参照してください。 必ず指定します(バリエーション型)。レコードイメージを項目単位にセットする配列をセットします。配列はバリエーション型の配列とし、配列の要素数はレコード項目数分用意します。
columnattribute	必ず指定します(バリエーション型)。各項目の属性を返却する配列をセットします。配列はInteger型の配列とし、配列の要素数はレコード項目数分用意します。
getmode	必ず指定します。レコード読み込みモードを指定します。
lockmode	ロックモードを指定します(省略可)。
accesskey	索引順編成で特定のキーをもとに読み込む場合(getmode=ifasGetEqual/ifasGetGreaterThan/ifasGetGreaterEqual)、キーとなる項目名を文字列で指定します(省略化)。 相対編成でgetmode=ifasGetEqual(特定相対キーの読み込み)場合は、読み込むレコードの相対キーを文字列で必ず指定します。特定相対キーの読み込み以外の場合は省略します(指定しても無視します)。 順編成の場合はパラメータを省略します(指定しても無視します)。

### 規則

- (1) ReadRecordメソッドが成功すると、IFAS ファイルからレコードを1件読み込み、項目値および項目属性を指定された配列に返却します。
- (2) relaccesskey/getmode/lockmode/accesskeyは、Readメソッドで使用する項目と同じです。
- (3) 索引順編成で、getmode=ifasGetEqual/ifasGetGreaterThan/ifasGetGreaterEqualにて読み込む前に、recordで指定された配列のキー項目の所に、キー値をセットしておく必要があります。この時、配列にセットする形式は、SetIndexAccessKeyメソッド

ッドで指定する項目値の形式を同じです。

- (4) **record**で示す配列に返す項目の値は、**GetColumnData**メソッドで返す項目値の形式と同じです。
- (5) **columnattribute**で示す配列に返す値は、**GetColumnData**メソッドで返す項目の属性値と同じです。
- (6) 上記以外の規則は、**Read**メソッドと同じです。

## ReadRecordStringメソッド

IFAS ファイルからレコードイメージを読み込み、string型で値を返却します。

### 構文

```
[relaccesskey =]  
object.ReadRecordString tablename,record,columnattribute,getmode[,lockmode][,accesskey]
```

### 指定項目

指定項目	内容
relaccesskey	相対編成の場合にアクセスしたレコードの相対キーを返却します(Long型)。
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。
record	指定方法の詳細は、「5-3-2 利用方法」を参照してください。 必ず指定します(バリエーション型)。レコードイメージを項目単位にセットする配列をセットします。配列はString型の配列とし、配列の要素数はレコード項目数分用意します。
columnattribute	必ず指定します(バリエーション型)。各項目の属性を返却する配列をセットします。配列はInteger型の配列とし、配列の要素数はレコード項目数分用意します。
getmode	必ず指定します。レコード読み込みモードを指定します。
lockmode	ロックモードを指定します(省略可)。
accesskey	索引順編成で特定のキーをもとに読み込む場合(getmode=ifasGetEqual/ifasGetGreaterThan/ifasGetGreaterEqual)、キーとなる項目名を文字列で指定します(省略化)。 相対編成でgetmode=ifasGetEqual(特定相対キーの読み込み)場合は、読み込むレコードの相対キーを文字列で必ず指定します。特定相対キーの読み込み以外の場合は省略します(指定しても無視します)。 順編成の場合はパラメータを省略します(指定しても無視します)。

### 規則

- (1) ReadRecordStringメソッドは以下に示す点を除き、ReadRecordメソッドと同じです。
  - ・ 項目値をセットする配列はString型の配列である
  - ・ キー値としてセットする項目値は常にString型である(SetIndexAccessKeyStringメソッドで指定する項目値の形式と同じ)
  - ・ 項目値は常にString型で返す(GetColumnDataStringメソッドで返す項目値の形式と同じ)

## WriteRecordメソッド

属性に従った形式のレコードイメージをIFAS ファイルに書き出します。

### 構文

---

```
[relaccesskey =]  
object.WriteRecord tablename,record[,accesskey]
```

### 指定項目

---

指定項目	内容
Relaccesskey	相対編成の場合にアクセスしたレコードの相対キーを返却します(Long型)。
Object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
Tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。 指定方法の詳細は、「5-3-2 利用方法」を参照してください。
Record	必ず指定します(バリエーション型)。レコードイメージを項目単位にセットする配列をセットします。配列はバリエーション型の配列とし、配列の要素数はレコードの項目数分用意します。
Accesskey	索引順編成または順編成の場合はパラメータを省略します(指定しても無視します)。 相対編成で動アクセスの場合は、書き出すレコードの相対キーを文字列で必ず指定します。順アクセスの場合は省略します(指定しても無視します)。

### 規則

---

- (1) **WriteRecord**メソッドが成功すると、指定された配列に設定されている項目値がIFAS ファイルに書き出されます。
- (2) **Relaccesskey**／**accesskey**は、**Write**メソッドで使用する項目と同じです。
- (3) **Record**で示す配列に指定する項目値の形式は、**SetColumnData**メソッドで設定する項目値の形式と同じです。
- (4) 上記以外の規則は**Write**メソッドと同じです。

## WriteRecordStringメソッド

String型のレコードイメージをIFAS ファイルに書き出します。

### 構文

---

```
[relaccesskey =]  
object.WriteRecordString tablename,record[,accesskey]
```

### 指定項目

---

指定項目	内容
Relaccesskey	相対編成の場合にアクセスしたレコードの相対キーを返却します(Long型)。
Object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
Tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。 指定方法の詳細は、「5-3-2 利用方法」を参照してください。
Record	必ず指定します(バリエーション型)。レコードイメージを項目単位にセットする配列をセットします。配列はString型の配列とし、配列の要素数はレコードの項目数分用意します。
Accesskey	索引順編成または順編成の場合はパラメータを省略します(指定しても無視します)。 相対編成で動アクセスの場合は、書き出すレコードの相対キーを文字列で必ず指定します。順アクセスの場合は省略します(指定しても無視します)。

### 規則

---

- (1) WriteRecordStringメソッドは以下に示す点を除き、WriteRecordメソッドと同じです。
  - ・項目値をセットする配列はString型の配列である
  - ・recordで示す配列に指定する項目値の形式は常にString型である(SetColumnDataStringメソッドで指定する項目値の形式と同じ)

## RewriteRecordメソッド

属性に従った形式のレコードイメージをIFAS ファイルに再書き出します(書き換え)。

### 構文

```
[relaccesskey =]  
object.RewriteRecord tablename,record,rewritemode[,accesskey]
```

### 指定項目

指定項目	内容
Relaccesskey	相対編成の場合にアクセスしたレコードの相対キーを返却します(Long型)。
Object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。
record	指定方法の詳細は、「5-3-2 利用方法」を参照してください。 必ず指定します(バリエーション型)。レコードイメージを項目単位にセットする配列をセットします。配列はバリエーション型の配列とし、配列の要素数はレコードの項目数分用意します。
rewritemode accesskey	必ず指定します。更新モードを整数または定数で指定します。 索引順編成でrewritemode=ifasRewriteKey(特定アクセスキーの書き換え)の場合は、書き換えるレコードのキー値をセットした項目名を文字列で指定します(省略化)。rewritemode=ifasRewriteCurrent(カレントレコードの書き換え)の場合は省略します(指定しても無視します)。 相対編成でrewritemode=ifasRewriteKey(特定相対キーの書き換え)の場合は、書き換えるレコードの相対キーを文字列で必ず指定します。rewritemode=ifasRewriteCurrent(カレントレコードの書き換え)の場合は省略します(指定しても無視します)。 順編成の場合はパラメータを省略します(指定しても無視します)。

### 規則

- (1) RewriteRecordメソッドが成功すると、IFAS ファイルのレコードは、指定された配列で設定されている項目値に書き換わります。
- (2) relaccesskey/accesskeyは、Rewriteメソッドで使用する項目と同じです。
- (3) recordで示す配列に指定する項目値の形式は、SetColumnDataメソッドで設定する項目値の形式と同じです。
- (4) 上記以外の規則はRewriteメソッドと同じです。



## RewriteRecordStringメソッド

String型のレコードイメージをIFAS ファイルに再書き出します(書き換え)。

### 構文

```
[relaccesskey =]  
object.RewriteRecordString tablename,record,rewritemode[,accesskey]
```

### 指定項目

指定項目	内容
relaccesskey	相対編成の場合にアクセスしたレコードの相対キーを返却します(Long型)。
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。
record	指定方法の詳細は、「5-3-2 利用方法」を参照してください。 必ず指定します(バリエーション型)。レコードイメージを項目単位にセットする配列をセットします。配列はString型の配列とし、配列の要素数はレコードの項目数分用意します。
rewritemode accesskey	必ず指定します。更新モードを整数または定数で指定します。 索引順編成でrewritemode=ifasRewriteKey(特定アクセスキーの書き換え)の場合は、書き換えるレコードのキー値をセットした項目名を文字列で指定します(省略化)。rewritemode=ifasRewriteCurrent(カレントレコードの書き換え)の場合は省略します(指定しても無視します)。 相対編成でrewritemode=ifasRewriteKey(特定相対キーの書き換え)の場合は、書き換えるレコードの相対キーを文字列で必ず指定します。rewritemode=ifasRewriteCurrent(カレントレコードの書き換え)の場合は省略します(指定しても無視します)。 順編成の場合はパラメータを省略します(指定しても無視します)。

### 規則

- (1) RewriteRecordStringメソッドは以下に示す点を除き、RewriteRecordメソッドと同じです。
  - ・ 項目値をセットする配列はString型の配列である
  - ・ recordで示す配列に指定する項目値の形式は常にString型である(SetColumnDataStringメソッドで指定する項目値の形式と同じ)

## DeleteRecordメソッド

IFAS ファイルから属性に従った形式でレコードを削除します。

### 構文

```
[relaccesskey =]  
object.DeleteRecord tablename,deletemode[,accesskey][,record]
```

### 指定項目

指定項目	内容
relaccesskey	相対編成の場合にアクセスしたレコードの相対キーを返却します(Long型)。
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。
deletemode	指定方法の詳細は、「5-3-2 利用方法」を参照してください。 必ず指定します。削除モードを整数または定数で指定します。
accesskey	索引順編成でdeletemode=ifasDeleteKey(特定アクセスキーの削除)の場合は、削除するレコードのキー値をセットした項目名を文字列で指定します(省略化)。Deletemode=ifasDeleteCurrent(カレントレコードの削除)の場合は省略します(指定しても無視します)。 相対編成でdeletemode=ifasDeleteKey(特定相対キーの削除)の場合は、削除するレコードの相対キーを文字列で必ず指定します。deletemode=ifasDeleteCurrent(カレントレコードの削除)の場合は省略します(指定しても無視します)。
record	順編成の場合はパラメータを省略します(指定しても無視します)。 索引順編成で、特定アクセスキーレコードを削除する(deletemode=ifasDeleteKey)場合、キー値となる項目値を、項目単位にセットする配列をバリエーション型で指定します(省略化)。配列はバリエーション型の配列とし、配列の要素数はレコードの項目数分用意します。

### 規則

- (1) DeleteRecordメソッドが成功すると、IFAS ファイルのレコードが削除されます。
- (2) relaccesskey／deletemode／accesskeyは、Deleteメソッドで使用する項目と同じです。
- (3) recordで示す配列に指定する項目値の形式は、SetIndexAccessKeyメソッドで設定する項目値の形式と同じです。
- (4) 上記以外の規則はDeleteメソッドと同じです。

## DeleteRecordStringメソッド

IFAS ファイルからString型でレコードを削除します。

### 構文

```
[relaccesskey =]
object.DeleteRecordString tablename,deletemode[,accesskey][,record]
```

### 指定項目

指定項目	内容
Relaccesskey	相対編成の場合にアクセスしたレコードの相対キーを返却します(Long型)。
Object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
Tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。 指定方法の詳細は、「5-3-2 利用方法」を参照してください。
Deletemode	必ず指定します。削除モードを整数または定数で指定します。
Accesskey	索引順編成でdeletemode=ifasDeleteKey(特定アクセスキーの削除)の場合は、削除するレコードのキー値をセットした項目名を文字列で指定します(省略化)。deletemode=ifasDeleteCurrent(カレントレコードの削除)の場合は省略します(指定しても無視します)。 相対編成でdeletemode=ifasDeleteKey(特定相対キーの削除)の場合は、削除するレコードの相対キーを文字列で必ず指定します。deletemode=ifasDeleteCurrent(カレントレコードの削除)の場合は省略します(指定しても無視します)。
Record	順編成の場合はパラメータを省略します(指定しても無視します)。 索引順編成で、特定アクセスキーレコードを削除する(deletemode=ifasDeleteKey)場合、キー値となる項目値を、項目単位にセットする配列をバリエーション型で指定します(省略化)。配列はString型の配列とし、配列の要素数はレコードの項目数分用意します。

### 規則

- (1) DeleteRecordStringメソッドは以下に示す点を除き、DeleteRecordメソッドと同じです。
  - ・ 項目値をセットする配列はString型の配列である
  - ・ キー値としてセットする項目値は常にString型である(SetIndexAccessKeyStringメソッドで指定する項目値の形式と同じ)

## StartRecordメソッド

IFAS ファイル内のレコードを属性に従った形式で位置付けします。

### 構文

---

```
[relaccesskey =]  
object.StartRecord tablename,startmode[,accesskey][,record]
```

### 指定項目

---

指定項目	内容
relaccesskey	相対編成の場合にアクセスしたレコードの相対キーを返却します(Long型)。
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。
startmode	指定方法の詳細は、「5-3-2 利用方法」を参照してください。 必ず指定します。位置付けモードを指定します。
accesskey	索引順編成では、位置付けを行うキーの項目名を文字列で指定します(省略化)。 相対編成でstartmode=ifasStartEqual/ifasStartGreaterThan/ifasStartGreaterEqual(特定相対キーの位置付け)場合は、位置付けするレコードの相対キーを文字列で必ず指定します。特定相対キーの読み込み以外の場合は省略します(指定しても無視します)。 順編成の場合はパラメータを省略します(指定しても無視します)。
record	索引順編成で、startmode = ifasStartEqual/ifasStartGreaterThan/ifasStartGreaterEqualにて位置付けする場合、キー値となる項目値を、項目単位にセットする配列をバリエーション型で指定します(省略化)。配列はバリエーション型の配列とし、配列の要素数はレコードの項目数分用意します。

### 規則

---

- (1) StartRecordメソッドが成功すると、IFAS ファイル内の特定レコードに位置付けされます。
- (2) relaccesskey/startmode/accesskeyは、Startメソッドで使用する項目と同じです。
- (3) recordで示す配列に指定する項目値の形式は、SetIndexAccessKeyメソッドで設定する項目値の形式と同じです。
- (4) 上記以外の規則はStartメソッドと同じです。

## StartRecordStringメソッド

IFAS ファイル内のレコードをString型で位置付けします。

### 構文

```
[relaccesskey =]  
object.StartRecordString tablename,startmode[,accesskey][,record]
```

### 指定項目

指定項目	内容
relaccesskey	相対編成の場合にアクセスしたレコードの相対キーを返却します(Long型)。
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。
startmode	指定方法の詳細は、「5-3-2 利用方法」を参照してください。 必ず指定します。位置付けモードを指定します。
accesskey	索引順編成では、位置付けを行うキーの項目名を文字列で指定します(省略化)。 相対編成でstartmode=ifasStartEqual/ifasStartGreaterThan/ifasStartGreaterEqual(特定相対キーの位置付け)場合は、位置付けするレコードの相対キーを文字列で必ず指定します。特定相対キーの読み込み以外の場合は省略します(指定しても無視します)。 順編成の場合はパラメータを省略します(指定しても無視します)。
record	索引順編成で、startmode = ifasStartEqual/ifasStartGreaterThan/ifasStartGreaterEqualにて位置付けする場合、キー値となる項目値を、項目単位にセットする配列をバリエーション型で指定します(省略化)。配列はString型の配列とし、配列の要素数はレコードの項目数分用意します。

### 規則

- (1) StartRecordStringメソッドは以下に示す点を除き、StartRecordメソッドと同じです。
  - ・ 項目値をセットする配列はString型の配列である
  - ・ キー値としてセットする項目値は常にString型である(SetIndexAccessKeyStringメソッドで指定する項目値の形式と同じ)

## GetTableDefメソッド

IFAS ファイルの定義情報を取得します。

### 構文

```
object.GetTableDef tablename,tabletype,columnnumber,recordlength,indexnumber
```

### 指定項目

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 なお、 <b>ファイル名は絶対パス名</b> で指定してください。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。
tabletype	指定方法の詳細は、「5-3-2 利用方法」を参照してください。 必ず指定します(Integer型)。IFAS ファイルのタイプを返します。詳細は設定値を参照してください。
columnnumber	必ず指定します(Integer型)。IFAS ファイルの項目数を返します。
recordlength	必ず指定します(Long型)。IFAS ファイルのレコード長を返します。
indexnumber	必ず指定します(Integer型)。インデックスの定義がある場合インデックス数を返します。インデックスがない場合は0を返します。

### 設定値

設定値	値	内容
ifasTableTypeBase	0	基本表
ifasTableTypeSelect	1	1つの基本表からなる仮想表(選択型仮想表)
ifasTableTypeConnect	2	複数の基本表からなる仮想表(結合型／合併型仮想表)
ifasTableTypeIfasSeq	10	一般順編成ファイル
ifasTableTypeIfasRel	11	一般相対編成ファイル
ifasTableTypeIfasIndex	12	一般索引順編成ファイル

### 規則

- (1) GetTableDefメソッドにより、IFAS ファイルの各種情報を返却します。
- (2) 取得時にIFAS ファイルはオープンしていなくてもかまいません。

## GetIndexDefメソッド

IFAS ファイルの索引定義情報を取得します。

### 構文

---

```
object.GetIndexDef tablename,indexnumber,indexattribute,columnnumber
```

### 指定項目

---

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 なお、 <b>ファイル名は絶対パス名</b> で指定してください。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。
indexnumber	指定方法の詳細は、「5-3-2 利用方法」を参照してください。 必ず指定します(Integer型)。IFAS ファイルに定義されている索引の順次番号(1から順番)を指定します。
indexattribute	必ず指定します(Integer型)。索引の属性を返します。詳細は、設定値を参照してください。
columnnumber	必ず指定します(Integer型)。索引を構成する項目の総数を返します。

### 返却値

---

設定値	値	内容
ifasIndexSingle	0	重複なし
ifasIndexDuplicate	1	重複あり

### 規則

---

- (1) GetIndexDefメソッドにより、IFAS ファイルに定義されているindexnumber番目の索引定義情報を返却します。
- (2) 取得時にIFAS ファイルはオープンしていなくてもかまいません。

## GetIndexColumnDefメソッド

IFAS ファイルの索引に対して、索引を構成する項目の情報を取得します。

### 構文

```
object.GetIndexColumnDef    tablename,indexnumber,columnnumber,columnname,
                             columnattribute,columnlength,coulmndecpoint
```

### 指定項目

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。なお、 <b>ファイル名は絶対パス名</b> で指定してください。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。
indexnumber	指定方法の詳細は、「5-3-2 利用方法」を参照してください。 必ず指定します(Integer型)。IFAS ファイルに定義されている索引の順次番号(1から順番)を指定します。
columnnumber	必ず指定します(Integer型)。索引を構成している項目の先頭からの順次番号(1を先頭とする)を指定します。
columnname	必ず指定します(String型)。項目名を返します。
columnattribute	必ず指定します(Integer型)。索引の属性を返します。詳細は、設定値を参照してください。
columnlength	必ず指定します(Integer型)。項目の総桁数を返します。
columndecpoint	必ず指定します(Integer型)。項目の少数点以下桁数を返します。

### 返却値

設定値	値	内容
ifasColumnChar	0	文字
ifasColumnJpnChar	1	日本語
ifasColumnInteger	2	単精度整数
ifasColumnLong	3	倍精度整数
ifasColumnDecimal	4	外部10進数
ifasColumnNumeric	5	内部10進数
ifasColumnBinary	6	2進数

### 規則

- (1) GetIndexColumnDefメソッドにより、IFAS ファイルに定義されているindexnumber番目の索引に対して、columnnumber番目の項目に関する定義情報を返却します。
- (2) columnlength(総桁数)／columndecpoint(小数点以下桁数)は、カラム属性によって、返る値が異なります(表内で - は値不定です)。



カラム属性	columnlength	columndecpoint
文字	バイト数	0
日本語	バイト数	0
単精度整数	-	-
倍精度整数	-	-
外部10進数	総桁数	小数点以下桁数
内部10進数	総桁数	小数点以下桁数
2進数	総桁数	小数点以下桁数

- (3) 取得時にIFAS ファイルはオープンしていかなくてもかまいません。

## GetIndexInfoDefメソッド

IFAS ファイルの索引に対して、索引を構成する項目の情報を取得します。

### 構文

```
object.GetIndexInfoDef  tablename,indexnumber,indexattribute,columnnumber,
                        columnname,columnattribute,columnlength,coulmndecpoint
```

### 指定項目

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。なお、 <b>ファイル名は絶対パス名</b> で指定してください。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。 指定方法の詳細は、「5-3-2 利用方法」を参照してください。
indexnumber	必ず指定します(Integer型)。IFAS ファイルに定義されている索引の順次番号(1から順番)を指定します。
indexattribute	必ず指定します(Integer型)。索引の属性を返します。
columnnumber	必ず指定します(Integer型)。索引を構成する項目の総数を返します。
columnname	必ず指定します(バリエーション型)。項目名を項目単位に返却する配列をセットします。配列はString型の配列とし、配列の要素数は索引を構成する項目数分用意します。
columnattribute	必ず指定します(バリエーション型)。項目の属性を項目単位に返却する配列をセットします。配列はInteger型の配列とし、配列の要素数は索引を構成する項目数分用意します。
columnlength	必ず指定します(バリエーション型)。項目の総桁数を項目単位に返却する配列をセットします。配列はInteger型の配列とし、配列の要素数は索引を構成する項目数分用意します。
columndecpoint	必ず指定します(バリエーション型)。項目の少数点以下桁数を項目単位に返却する配列をセットします。配列はInteger型の配列とし、配列の要素数は索引を構成する項目数分用意します。

### 規則

- (1) GetIndexInfoDefメソッドにより、IFAS ファイルに定義されているindexnumber番目の索引情報を返却します。
- (2) indexnumber／indexattribute／columnnumberは、GetIndexDefメソッドで使用する項目と同じです。
- (3) columnname／columnattribute／columnlength／columndecpointで示す配列に返却する値の形式は、GetIndexColumnDefメソッドで使用する項目と同じです。
- (4) columnname／columnattribute／columnlength／columndecpointの各配列には、索引を構成する項目の情報を、索引を構成している項目順に返します。また、配列の要素数は索引を構成する項目数分必要ですが、索引を構成する項目は最大16

- 項目であるため、あらかじめ16個の要素を持つ配列を使用すれば問題ありません。
- (5) 上記以外の規則は**GetIndexDef**メソッドと同じです。

## GetColumnDefメソッド

IFAS ファイルの構成する項目の情報を取得します。

### 構文

```
object.GetColumnDef    tablename,columnnumber,columnname,  
                        columnattribute,columnlength,coulmndecpoint
```

### 指定項目

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。 指定方法の詳細は、「5-3-2 利用方法」を参照してください。
columnnumber	必ず指定します(Integer型)。IFAS ファイルを構成している項目の先頭からの順次番号(1を先頭とする)を指定します。
columnname	必ず指定します(String型)。項目名を返します。
columnattribute	必ず指定します(Integer型)。項目の属性を返します。詳細は、設定値を参照してください。
columnlength	必ず指定します(Integer型)。項目の総桁数を返します。
columndecpoint	必ず指定します(Integer型)。項目の少数点以下桁数を返します。

### 返却値

設定値	値	内容
ifasColumnChar	0	文字
ifasColumnJpnChar	1	日本語
ifasColumnInteger	2	単精度整数
ifasColumnLong	3	倍精度整数
ifasColumnDecimal	4	外部10進数
ifasColumnNumeric	5	内部10進数
ifasColumnBinary	6	2進数

### 規則

- (1) GetColumnDefメソッドにより、IFAS ファイルに定義されているcolumnnumber番目の項目に関する定義情報を返却します。
- (2) columnlength(総桁数)／columndecpoint(小数点以下桁数)は、カラム属性によって、返る値が異なります(表内で - は値不定です)。

カラム属性	columnlength	columndecpoint
文字	バイト数	0
日本語	バイト数	0
単精度整数	-	-
倍精度整数	-	-
外部10進数	総桁数	小数点以下桁数
内部10進数	総桁数	小数点以下桁数
2進数	総桁数	小数点以下桁数

- (3) 取得時にIFAS ファイルは**オープン**をされていなければなりません。

## GetRecordDefメソッド

IFAS ファイルの構成する項目の情報を取得します。

### 構文

---

```
object.GetRecordDef  tablename,columnname,columnattribute,columnlength,  
                      coulmndecpoint
```

### 指定項目

---

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。
tablename	必ず指定します。IFAS ファイル名を文字列で指定します。 IFAS一般ファイル名は [＜サーバ名＞]ファイル名の形式で指定します。 IFAS RDB表名は [＜サーバ名＞][データベース名.]表名の形式で指定します。 指定方法の詳細は、「5-3-2 利用方法」を参照してください。
columnname	必ず指定します(バリエーション型)。項目名を項目単位に返却する配列をセットします。配列はString型の配列とし、配列の要素数はレコードの項目数分用意します。
columnattribute	必ず指定します(バリエーション型)。項目の属性を項目単位に返却する配列をセットします。配列はInteger型の配列とし、配列の要素数はレコードの項目数分用意します。
columnlength	必ず指定します(バリエーション型)。項目の総桁数を項目単位に返却する配列をセットします。配列はInteger型の配列とし、配列の要素数はレコードの項目数分用意します。
columndecpoint	必ず指定します(バリエーション型)。項目の少数点以下桁数を項目単位に返却する配列をセットします。配列はInteger型の配列とし、配列の要素数はレコードの項目数分用意します。

### 規則

---

- (1) GetRecordDefメソッドにより、IFAS ファイルに定義されている項目に関する定義情報を返却します。
- (2) columnname／columnattribute／columnlength／columndecpointで示す配列に返却する値の形式は、GetColumnDefメソッドで使用する項目と同じです。
- (3) 上記以外の規則はGetColumnDefメソッドと同じです。

## Termメソッド

カスタムコントロールの動作環境の終了宣言を行います。

### 構文

---

```
object.Term
```

### 指定項目

---

指定項目	内容
object	必ず指定します。カスタムコントロールへの参照を表すオブジェクト式を指定します。

### 規則

---

- (1) **Term**メソッドにより、カスタムコントロールの終了宣言を行います。  
終了宣言を行うと、オープンしているすべてのIFAS ファイルをクローズしたり、カスタムコントロールが確保している資源を解放したりします。
- (2) アプリケーション終了時にこのメソッドを呼び出すことを推奨します。

## 6. 運用と保守

IFAS FILE、IFASPRO RDBの特徴であるクライアント・サーバファイルアクセス機能、障害時のファイル復旧機能について、その運用方法を説明します。

### 6-1 クライアント・サーバ機能

クライアント・サーバ形態で運用する場合、サーバマシンにIFAS FILE、IFASPRO RDBの製品(IFASPRO RDB Server Runtimeを除く)をインストールしてください。本項では、サーバ上のファイルをアクセスする方法について説明しています。

#### 6-1-1 サーバ名の指定方法

IFAS FILE、IFASPRO RDBのクライアント・サーバ形態では、クライアントアプリケーションプログラムがアクセスするサーバ名をIFAS FILE、IFASPRO RDBへ知らせる必要があります。

IFAS FILE、IFASPRO RDBで、クライアントマシンからサーバマシン上のファイルをアクセスする場合、下記に示すようファイル名にサーバ名を付加してください。

〈サーバ名〉ファイルパス

ソース上にファイル名を、直接“〈サーバ名〉ファイルパス”と記述するよりも、COBOLファイルマップパー、環境設定ファイルでファイル名の関連付けを行うことを推奨します。

#### 注意事項

- (1) クライアント・サーバでファイルをアクセスする場合、ファイル名はフルパスで指定してください。

正

```
<SERVER01>C:¥IFASFILE¥FILE01
<SERVER01>¥¥SERVER01¥IFASSHR¥FILE02
```

誤

```
<SERVER01>FILE01
<SERVER01>DIR¥FILE02
```

- (2) ネットワークドライブ上のファイルは、指定しないでください。ネットワークドライブを使ってサーバ上のファイルをアクセスした場合、IFASサーバを経由しないためファイルの排他制御を十分行うことができず、ファイルの更新が正しく行えない場合があります。
- (3) 以下の様な運用は可能ですが2フェーズコミットのような、サーバ間のトランザクション処理の同期制御はできません。APからのコミットやロールバック等の要求が、あるホストでエラーになった場合、APにはエラーが通知されますが、既に成功した他サーバへの要求を取り消すことはできません。
  - ・1つのクライアントAPが同時にサーバとクライアントローカルファイルへアクセスを行う
  - ・1つのクライアントAPが同時に複数のサーバへアクセスを行う(複数サーバでファイルをオープンする)
- (4) [Host List]に論理サーバ名に実際に存在するサーバの名前は極力を指定しないでください。IFASの提供するユーティリティ/コマンドにもこの指定は有効になり、誤動作の原因となります。(テスト有用時に留めてください)
- (5) ファイル名がUNC名の場合以下のようにしてしてください。なお、IFASサーバとファイルサーバは同一のマシンにしてください。

```
<SERVER01>¥¥SERVER01¥IFASSHR¥FILE03
IFASサーバ      ファイルサーバ
```

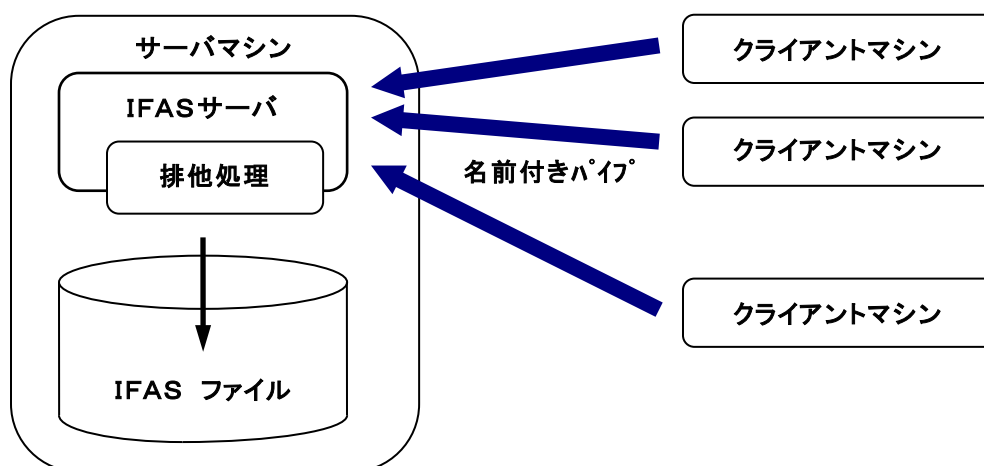


## ネットワークドライブアクセスとの違い

OSの機能によってネットワークマウントしたドライブ上のファイルは、IFAS FILE、IFASPRO RDBにとってはローカルファイルという認識になります。IFAS FILE、IFASPRO RDBのクライアント・サーバアクセス機能を使用する場合は、ネットワークドライブ上のファイルは指定しないでください。ネットワークドライブを使ってサーバ上のファイルをアクセスした場合は、IFASサーバを経由しないためファイルの排他制御を行うことができません。よってファイルの更新が正しく行えない場合があります。

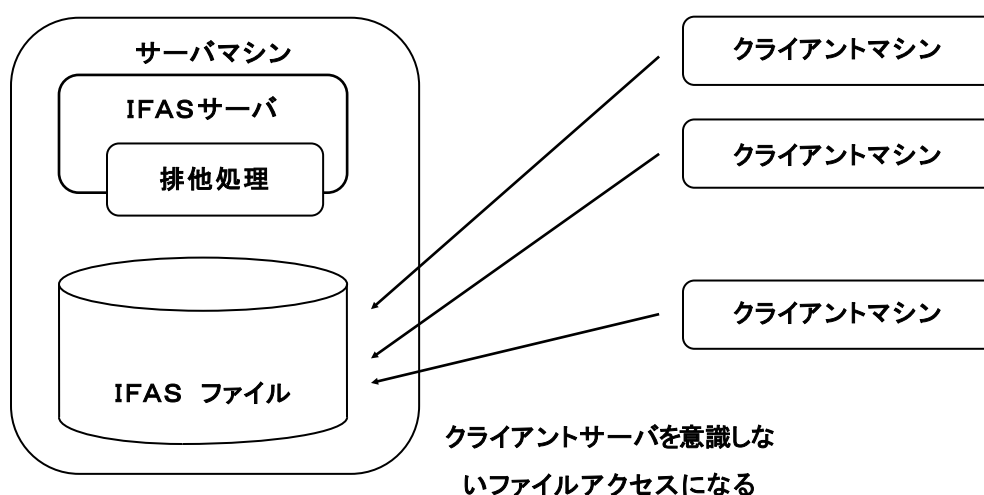
### クライアント・サーバアクセス

図のように各クライアントマシンからのアクセスをIFASサーバが一括して処理することで、複数クライアントからのデータ更新を可能にしています。



### ネットワークドライブやUNC名によるファイルアクセス

図のようにネットワークドライブを経由したアクセスやUNC名(¥¥サーバ名¥共有名¥ファイルパス)でアクセスを行った場合、複数のクライアントからそれぞれのローカルなファイルとして同時に更新が行われ、場合によってはデータの保証ができなくなることがあります。



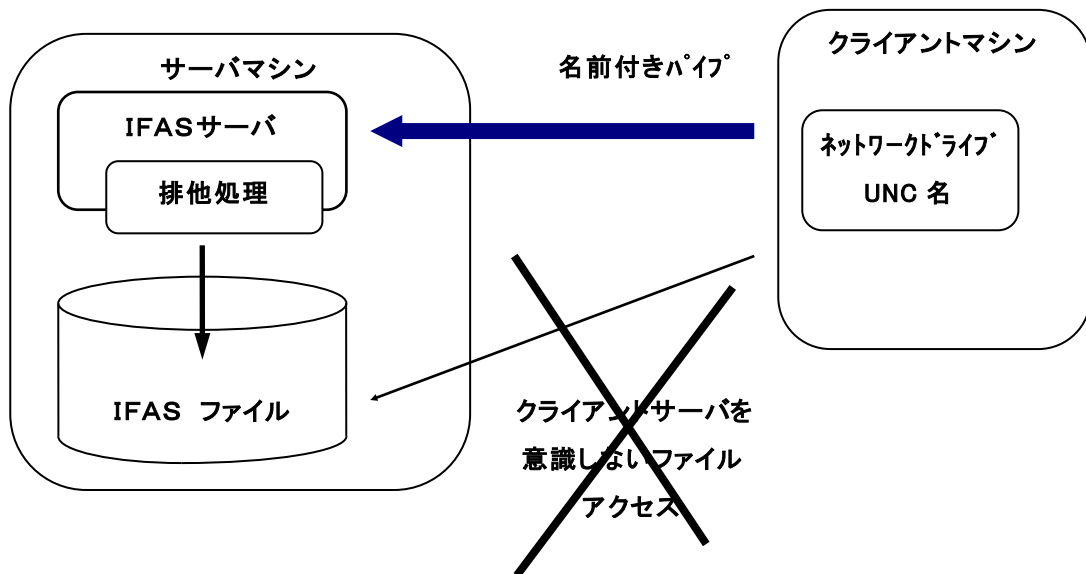
## 自動C/S設定

運用支援ツールの[その他]タブー[自動C/S設定を行う]を設定することで、自動的にネットワークドライブを経由したアクセスやUNC名でのアクセスがクライアント・サーバアクセスに変換されるようになります。なお、本機能は内部的に以下のファイル名変換を行うことで実現しています。

¥¥SERVER01¥SHR01¥FILE01 → <SERVER01>¥¥SERVER01¥SHR01¥FILE01

E:¥IFASFILE¥FILE01 → <SERVER01>¥¥SERVER01¥SHR01¥FILE01

(“E”はネットワークドライブでサーバSERVER01の共有名SHR01に接続している)



## クライアント・サーバのパラメータ

環境設定ファイルの[Control]セクションに下記の記述を行うことで、サーバ接続時のタイムアウト値を指定することができます。WAN環境でときどきファイルのオープンでクライアント・サーバ系のエラーが発生するような場合この値を大きくしてください。

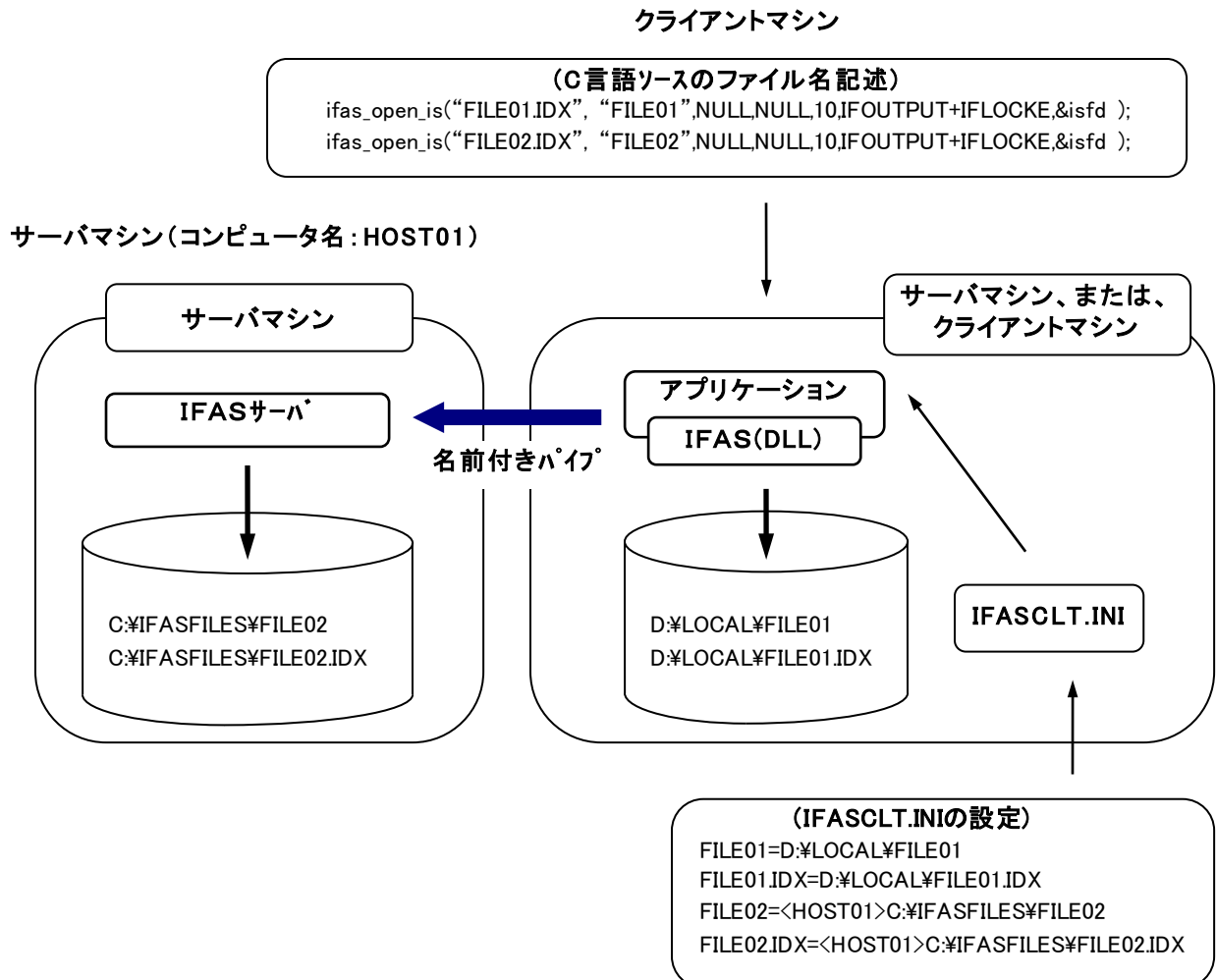
[Control]

ConnectTimeout=n

環境設定ファイルの詳細は「**2-3 環境設定ファイル**」を参照してください。

## 6-1-2 Cプログラムからの利用

ローカルファイルアクセスとクライアント・サーバアクセスの違いを理解しやすいように、まずローカルファイルの指定例を示し、次にサーバ上のファイルの指定例を示します。



### 例1. クライアント(または、サーバ)上のアプリケーションがローカルファイルをアクセス

クライアント(または、サーバ)上のアプリケーションがローカルファイルをアクセスする場合、次のようにファイル名を指定してください。

#### C言語ソース上のファイル名の記述

//IFAS一般ファイルのオープン

```
ifas_open_is("FILE01.IDX", "FILE01", NULL, NULL, 10, IFOUTPUT+IFLOCKE, &isfd );
```

#### 環境設定ファイルの設定

[Table List]

```
FILE01=D:¥LOCAL¥FILE01
```

```
FILE01.IDX=D:¥LOCAL¥FILE01.IDX
```

ソース上「FILE01/FILE01.IDX」と記述されているファイル名は、「D:¥LOCAL¥FILE01/D:¥LOCAL¥FILE01.IDX」に関連付けられます。

環境設定ファイルを用いずに直接、次のようにファイル名をソースに記述することもできます。なお、プログラムの保守性から環境設定ファイルの使用をお勧めします。

```
ifas_open_is(" D:¥LOCAL¥FILE01.IDX ",
" D:¥LOCAL¥FILE01", NULL, NULL, 10, IFOUTPUT + IFLOCKE, &isfd );
```

**例2. クライアント上のアプリケーションがサーバ上のファイルをアクセス**

クライアント上のアプリケーションがサーバ上のファイルをアクセスする場合、次のようにファイル名を指定してください。

**C言語ソース上のファイル名の記述**

```
//IFAS一般ファイルのオープン
```

```
ifas_open_is("FILE02.IDX", "FILE02", NULL, NULL, 10, IFOUTPUT+IFLOCKE, &isfd );
```

**環境設定ファイルの設定**

```
[Table List]
```

```
FILE02=<HOST01>C:¥IFASFILES¥FILE02
```

```
FILE02.IDX=<HOST01>C:¥IFASFILES¥FILE02.IDX
```

ソース上「FILE02/FILE02.IDX」と記述されているファイル名は、

「<HOST01>C:¥IFASFILES¥FILE02/<HOST01>C:¥IFASFILES¥FILE02.IDX」に関連付けられます。

ます。

環境設定ファイルを用いずに直接、次のようにファイル名をソースに記述することもできます。な

お、プログラムの保守性から環境設定ファイルの使用をお薦めします。

```
ifas_open_is("<HOST01>C:¥IFASFILES¥FILE02.IDX ",
              "<HOST01>C:¥IFASFILES¥FILE02", NULL, NULL, 10,
              IFOUTPUT+IFLOCKE, &isfd );
```

## 6-1-3 COBOLプログラムでの指定

### IFAS一般ファイルの指定例

ローカルファイルアクセスとクライアント・サーバアクセスの違いを理解しやすいように、まずローカルファイルの指定例を示し、次にサーバ上のファイルの指定例を示します。

**例1. クライアント(または、サーバ)上のアプリケーションがローカルマシン上のファイルをアクセス**

アプリケーションがアクセスするIFAS一般ファイルを、COBOLファイルマッパーに指定してください。

**COBOLソース上のファイル名の記述**

```
FD F0001
```

```
LABEL RECORD IS STANDARD
```

```
VALUE OF IDENTIFICATION IS "FILE01".
```

**COBOLファイルマッパーでの指定**

```
内部ファイルID      FILE01
```

```
デバイス      指定なし
```

```
ファイル種別  IFAS FILE
```

「IFAS FILE」を指定することでIFAS一般ファイルとしてアクセス可能になります。

実ファイル名 D:¥LOCAL¥FILE01

ソース上「FILE01」と記述されている内部ファイルIDは、「D:¥LOCAL¥FILE01」に変換されます。

**例2. クライアント上のアプリケーションがサーバ上のファイルをアクセス**

アプリケーションが接続するIFASサーバとIFAS一般ファイルを、COBOLファイルマッパーに指定してください。

**COBOLソース上のファイル名の記述**

```
FD F0002
```

```
LABEL RECORD IS STANDARD
```

```
VALUE OF IDENTIFICATION IS "FILE02".
```

**COBOLファイルマッパーでの指定**

```
内部ファイルID      FILE02
```

```
デバイス      指定なし
```

```
ファイル種別  IFAS FILE
```

「IFAS FILE」を指定することでIFAS一般ファイルとしてアクセス可能になります。

実ファイル名 <HOST01>C:¥IFASFILES¥FILE02  
 サーバ名「<HOSTS01>」を指定することでサーバ上の一般ファイルがアクセス可能になる。  
 ソース上「FILE02」と記述されている内部ファイルIDは、  
 「<HOST01>C:¥IFASFILES¥FILE02」に変換されます。

## IFAS RDB表の指定例

ローカルファイルアクセスとクライアント・サーバアクセスの違いを理解しやすいように、まずローカルファイルの指定例を示し、次にサーバ上のファイルの指定例を示します。

### 例1. クライアント(または、サーバ)上のアプリケーションがローカルマシン上のRDB表にアクセス

アプリケーションがアクセスするIFASRDB表を、COBOLファイルマッパーに指定してください。

#### COBOLソース上の表名の記述

```
FD F0001
  LABEL RECORD IS STANDARD
  VALUE OF IDENTIFICATION IS "DBTBL01".
```

#### COBOLファイルマッパーでの指定

```
内部ファイルID      DBTBL01
デバイス            RDB
ファイル種別        IFASPRO RDB
「IFASPRO RDB」を指定することでIFAS RDB表としてアクセス可能になります。
実ファイル名        DB01.DBTBL01
ソース上「DBTBL01」と記述されている内部ファイルIDは、「DB01.DBTBL01」に変換されます。
```

### 例2. クライアント上のアプリケーションがサーバ上のRDB表にアクセス

アプリケーションが接続するIFASサーバとIFAS RDB表を、COBOLファイルマッパーに指定してください。

#### COBOLソース上の表名の記述

```
FD F0002
  LABEL RECORD IS STANDARD
  VALUE OF IDENTIFICATION IS "DBTBL02".
```

#### COBOLファイルマッパーでの指定

```
内部ファイルID      DBTBL02
デバイス            RDB
ファイル種別        IFASPRO RDB
「IFASPRO RDB」を指定することでIFAS RDB表としてアクセス可能になります。
実ファイル名        <HOST01>DB02.DBTBL02
サーバ名「<HOSTS01>」を指定することでサーバ上のRDB表がアクセス可能になる。
ソース上「DBTBL02」と記述されている内部ファイルIDは、「<HOST01>DB02.DBTBL02」
に変換されます。
```

## 6-1-4 IFASカスタムコントロールでの指定(参考)

IFAS FILE Ver2.0、IFASPRO RDB Ver2.0より、「IFASカスタムコントロール」機能の提供を中止しました。本項の内容は従来バージョンをご利用のお客様が旧製品の仕様を確認いただくために残しています。

## IFAS一般ファイルの指定例

ローカルファイルアクセスとクライアント・サーバアクセスの違いを理解しやすいように、まずローカルファイルの指定例を示し、次にサーバ上のファイルの指定例を示します。

### 例1. クライアント(または、サーバ)上のアプリケーションがローカルファイルをアクセス

クライアント(または、サーバ)上のアプリケーションがローカルファイルをアクセスする場合、次のようにファイル名を指定してください。

**MS-VisualBasic(以下VB)ソース上のファイル名の記述**

```
//IFAS一般ファイルのオープン
```

```
Ifas321.Open "D:¥LOCAL¥FILE01",ifasOpenExUpdate,ifasAccessIfasIndexRnd,"KeyName"
```

**例2. クライアント上のアプリケーションがサーバ上のファイルをアクセス**

クライアント上のアプリケーションがサーバ上のファイルをアクセスする場合、次のようにファイル名を指定してください。

**MS-VisualBasic(以下VB)ソース上のファイル名の記述**

```
//IFAS一般ファイルのオープン
```

```
Ifas321.Open
```

```
"<HOST01>C:¥IFASFILE¥FILE02",ifasOpenExUpdate,ifasAccessIfasIndexRnd,"KeyName"
```

**IFAS RDB表の指定例**

ローカルファイルアクセスとクライアント・サーバアクセスの違いを理解しやすいように、まずローカルファイルの指定例を示し、次にサーバ上のファイルの指定例を示します。

**例1. クライアント(または、サーバ)上のアプリケーションがローカルマシン上のRDB表をアクセス**

クライアント(または、サーバ)上のアプリケーションがローカルファイルをアクセスする場合、次のようにファイル名を指定してください。

**MS-VisualBasic(以下VB)ソース上のファイル名の記述**

```
//IFASRDB表のオープン
```

```
Ifas321.Open "DB01.DBTBL01",ifasOpenExUpdate,ifasAccessIndexRnd,"KeyName"
```

**例2. クライアント上のアプリケーションがサーバ上のRDB表をアクセス**

クライアント上のアプリケーションがサーバ上のファイルをアクセスする場合、次のようにファイル名を指定してください。

**MS-VisualBasic(以下VB)ソース上のファイル名の記述**

```
//IFASRDB表のオープン
```

```
Ifas321.Open "<HOST01>DB02.DBTBL02",ifasOpenExUpdate,  
ifasAccessIndexRnd,"KeyName"
```

**6-1-5 注意事項****IFAS ファイルのセキュリティについて**

IFAS ファイルのセキュリティは、OSでの機能／設定に従います。たとえば、ファイルのアクセスについては、NTFSを使えば、ファイル(または上位ディレクトリ)に設定されたアクセス権属性で要求が許可／拒否されます(ファイルマネージャやエクスプローラで設定可能です)。クライアント／サーバでは、加えてネットワーク接続のセキュリティ設定(ドメインユーザマネージャやユーザマネージャで設定可能です)により、接続要求が許可／拒否されます。ファイルアクセスのセキュリティは、IFAS FILE、IFASPRO RDBに限らず、WindowsOS上で動作するAPに関わる一般的機能です。

**必要なアクセス権について****ログオンアカウント(OSログオン時のユーザ名)の権限**

IFAS FILE、IFASPRO RDBを使用する場合、ログオンアカウントに下記の権限が必要になります。なお、RDB表を利用しない場合は「3)」は不要です。

1) レジストリを参照更新する権限

2) IFAS FILE、IFASPRO RDBインストールディレクトリ配下のファイルやサブディレクトリを参照更新する権限

3) IFASPRO RDB管理ディレクトリ配下のファイルを参照更新する権限

IFASの各種設定情報、制御情報を参照更新するため、これらの権限が必要です。

## 6-2 IFAS ファイルのバックアップ

IFAS ファイル環境を他のシステムへ移行するため、または障害時のデータ復旧のために、IFAS ファイル環境(IFAS RDB表環境を含む)のバックアップをとっておくことを推奨します。

なお、バックアップは以下の操作を行った後に実施することを推奨します。

- ユーティリティを使用してIFAS ファイルの状態を変更した場合  
(IFAS ファイルの作成やインデックスの変更、コピーなど)
- IFAS RDB表環境の再構築を行った場合(リカバリ実行後を含む)
- システムの時刻/タイムゾーンを変更した場合  
(更新ログを収集している場合は、更新ログのリセットも同時に行ってください)

### 使用するバックアップツールについて

IFAS FILE、IFASPRO RDBはバックアップをするための専用ツールは実装しておりません。

IFASに実装される、ファイルコピーユーティリティや表定義ユーティリティ、またIFASのアクセス関数を使用した自作のアプリケーションを利用してバックアップは行わないでください。

ファイルコピーユーティリティ等は論理レコード単位でデータを複写するプログラムであり、バックアップ(物理的にファイルを待避する)ためのものではありません。

WindowsOS標準のエクスプローラやファイルマネージャ、またはバックアップ専用の市販アーカイバ等を使用するようにしてください。

### IFAS RDB表環境のバックアップ

IFAS RDB表環境のバックアップを行うには、RDB管理ディレクトリ(RDB管理ファイル群)および表のバックアップを行う必要があります。

以下にデータベースを構成するファイルを示します。

#### RDB管理ファイル

RDB管理ファイルはRDB管理ディレクトリのディレクトリパス配下にあります。ファイルには隠し属性が付加されています。

各種定義情報	ファイル名	
データベース定義情報	DBFILES	DBFILE. IDX
	DBFILE. ISF	DBFILE. FDX
表の定義情報	TABLES	TABLES. IDX
	TABLES. ISF	TABLES. FDX
列の定義情報	COLUMNS	COLUMNS. IDX
	COLUMNS. ISF	COLUMNS. FDX
インデックスの定義情報	INDEXES	INDEXES. IDX
	INDEXES. ISF	INDEXES. FDX
インデックスを構成するキーの定義情報	INDEX_KEYS	INDEX_KEYS. IDX
	INDEX_KEYS. ISF	INDEX_KEYS. FDX
表制約の定義情報	TABLES__CONSTRAINTS	TABLES__CONSTRAINTS. IDX
	TABLES__CONSTRAINTS. ISF	TABLES__CONSTRAINTS. FDX
表制約を構成する列の定義情報	KEY_COLUMN_USAGE	KEY_COLUMN_USAGE. IDX
	KEY_COLUMN_USAGE. ISF	KEY_COLUMN_USAGE. FDX
仮想表の定義情報	VIEW_DEFINITIONS	VIEW_DEFINITIONS. IDX
	VIEW_DEFINITIONS. ISF	VIEW_DEFINITIONS. FDX
仮想表構成する表の定義情報	VIEW_TABLE_USAGE	VIEW_TABLE_USAGE. IDX
	VIEW_TABLE_USAGE. ISF	VIEW_TABLE_USAGE. FDX
仮想表を構成する列の定義情報	VIEW_COLUMN_USAGE	VIEW_COLUMN_USAGE. IDX
	VIEW_COLUMN_USAGE. ISF	VIEW_COLUMN_USAGE. FDX

仮想表の属性定義情報	IFAS__VIEW__ATTRIBUTES	IFAS__VIEW__ATTRIBUTES. IDX
仮想表の関係式定義情報	IFAS__VIEW__RELATIONS	IFAS__VIEW__RELATIONS. IDX
仮想表の相関名定義情報	IFAS__VIEW__ALIASES	IFAS__VIEW__ALIASES. IDX

## 表

表はデータベースディレクトリのディレクトリパス配下にあります。

実表(表を構成するファイル群)のファイル名を以下に示します。

表名

表名. IDX

表名. ISF

表名. FDX

なお、上記と同じディレクトリパス配下に表に対応するRFDファイル(表名. rfd)がある場合、表と一緒にバックアップを行うことを推奨します。

## バックアップ可能な環境

IFAS ファイルのバックアップを行う場合はIFAS ファイルをアクセスしているアプリケーションが動作していない状態で行ってください。またバックアップ処理実行中はIFAS ファイルをアクセスするアプリケーションを起動しないようにしてください。

前述した“IFAS ファイルをアクセスするアプリケーション”とは、IFASに実装する以下のユーティリティプログラムも含まれます。

- ・ファイル管理ユーティリティ
- ・表定義ユーティリティ
- ・ファイルコピーユーティリティ
- ・インデックスメンテナンスユーティリティ
- ・ファイルロックリセットユーティリティ

また、IFAS以外の製品でもIFAS ファイルをアクセスするユーティリティプログラムがありますのでご注意ください。

アプリケーションで使用中のIFAS ファイルのバックアップが行われると、不完全なファイル状態でバックアップファイル中に保存される可能性があります。

## バックアップ支援機能

IFASは、利用者が確実に正常なバックアップを行えるように、以下のような機能を持つ「バックアップ支援ユーティリティ」を提供しています。

- ・ IFASアプリケーションが起動中か否かを監視する。
- ・ 起動中のIFASアプリケーションがいる場合その終了を待ちあわせて通知する。
- ・ バックアップ実行中はIFASアプリケーションが新たに起動されるのを抑制する。

なお、詳細については「7-4-2 バックアップ支援ユーティリティ」を参照してください。



## 6-3 更新ログファイルの運用

IFAS ファイル更新時に、更新ログの収集を指定することができます。更新ログは、IFAS FILEイニシエータ(サービス)により起動されたログライタというプロセスで収集されます。そのために、更新ログファイルはサービスのアカウントでアクセスされることになります。更新ログファイルには、このサービスのアカウントにも読み書き可能な権限を与えてください。

更新ログファイルには、IFAS ファイルに対して行われた、ファイル操作およびレコード操作による更新情報がすべて記録されます。更新ログを収集しておく、ディスク障害などによりIFAS ファイルが使用できなくなってしまう場合、IFAS ファイルのバックアップファイルと更新ログファイルを用いて、ロールフォワードユーティリティでIFAS ファイルを復旧することができます。

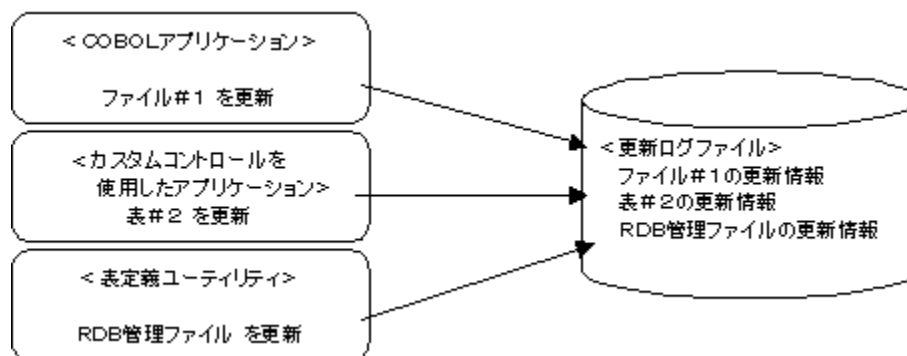
更新ログファイルの運用には、シングルログ運用とマルチログ運用の2種類の方法があります。

### 6-3-1 更新ログの収集

更新ログの収集はロールフォワードを行うことを前提としています。ロールフォワードは、ログファイルに記録されている更新情報を順番に再生することによりIFAS ファイルを復旧します。そのため、あるIFAS ファイルに対してその更新情報が記録されている更新ログファイルは1つだけ存在するようにしなければなりません。また、ロールフォワードはIFAS ファイルのバックアップファイルと更新ログファイルを使用して復旧を行いますので、バックアップ以降の更新情報がすべて更新ログファイルに収集されている必要があります。したがって、バックアップを行った時点で、それまで収集していた更新ログは不要となりますので、更新ログファイルのリセットすることをお勧めします。更新ログファイルのリセット方法については「6-3-4 更新ログファイルのリセット」で説明します。

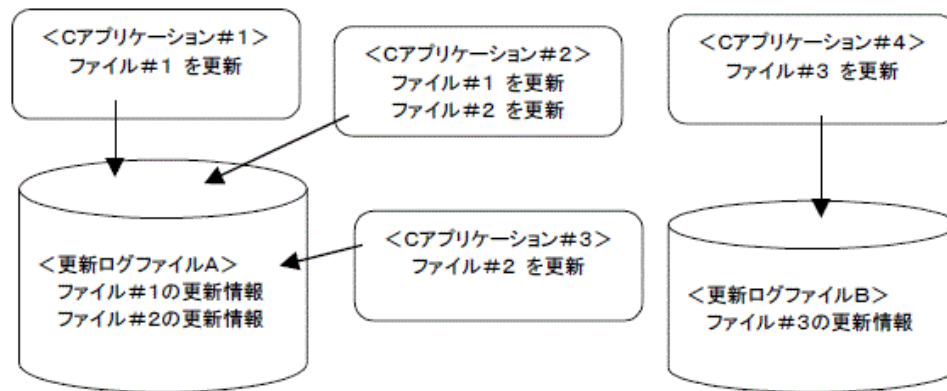
＜COBOLまたはカスタムコントロールなどで使用するIFAS ファイル＞

COBOLまたはカスタムコントロールで使用するIFAS ファイルの更新ログは運用支援ツールの[更新ログ設定]タブで指定したファイルに収集されます。



＜Cアプリケーションで使用するIFAS ファイル＞

Cアプリケーションで使用するIFAS ファイルの更新ログはLOGOPEN関数で指定したファイルに収集されます。したがって複数のアプリケーションから同じIFAS ファイルを更新する場合、すべてのアプリケーションで同一の更新ログファイルをオープンする必要があります。このため、Cアプリケーションだけでなく、COBOLやカスタムコントロールからも同一のIFAS ファイルを使用するような運用を行う場合は、運用支援ツールの[更新ログ設定]タブで指定したファイル名をLOGOPEN関数で指定してください。



## 6-3-2 シングルログによる運用

シングルログ運用とは、1つの更新ログファイルにIFAS ファイルの更新情報を記録していく運用方法です。指定された更新ログファイルがディスク上に存在しない場合、IFASで更新ログファイルを作成します。既に更新ログファイルが存在する場合、追加書き出ししていきます。

ディスクに空きスペースが存在するかぎり、更新情報を記録していくので、任意のタイミングでIFAS ファイルのバックアップを行い、更新ログファイルをリセットする必要があります(詳細は「6-3-4 更新ログファイルのリセット」を参照してください)。

更新ログファイルを既定値で使用する場合は、シングルログ運用となります。

## 6-3-3 マルチログによる運用

マルチログ運用とは、COBOLやカスタムコントロールで使用する更新ログファイルのみ使用できる運用方法です。更新ログファイルを2つ以上のファイルで構成し、そのファイル群にIFAS ファイルの更新情報を記録していきます。指定された更新ログファイルがディスク上に存在しない場合、IFASで更新ログファイルを作成します。既に更新ログファイルが存在する場合、追加書き出ししていきます。

1つの更新ログファイルが指定されたファイルサイズに達した時、次のファイルに自動的に切り替わり、最後のファイルまでいっぱいとなった場合、最初のファイルに戻り記録します(オーバーフロー)。このため、更新ログを出し続けることによりディスクを圧迫することはありません。

更新ログファイルがオーバーフローし、更新ログが消失した場合、その更新ログファイルを使用してロールフォワードを行っても、IFAS ファイルを最新の状態に復旧できなくなってしまいます。このため、更新ログファイルがオーバーフローする前に、IFAS ファイルのバックアップを行う必要があります。

マルチログ運用する場合は、「6-3-5 更新ログファイルの設計」を参考に、IFAS ファイルのバックアップから次のバックアップまでに行くと予想される更新量を算出し、更新ログファイルサイズを計算してください。

マルチログ運用を行うためには、運用支援ツールでファイル数とファイルサイズの設定を行う必要があります。運用支援ツールの[更新ログ設定]タブで任意のファイル名を指定し、ファイル数を2以上、ファイルサイズを1MB以上として設定することによりマルチログ運用の設定ができます。この時、**ファイル名に拡張子を付けてはいけません**。IFASがファイル番号を拡張子として付加し、更新ログファイルを作成します。また、ファイル数の最小値が2、1つの更新ログファイルサイズの最小値が1MBとなりますので、更新ログファイルを置くドライブには最低2MBの空き容量が必要となります。

下図の例は、更新ログファイル名に“CBLLOG”、ファイル数を3、ファイルサイズに1MB以上の値を指定した時の更新ログファイル使用例です。

更新ログファイル名、ファイル数またはファイルサイズを変更すると、それまで収集されていた更新ログは無効となりますので、設定の変更時は必ずIFAS ファイルのバックアップを採取してから行うようにしてください。



マルチログ運用を行う場合、更新ログファイル情報の出力など、いくつかのオプションが用意されています。これを利用して、更新ログファイル数／サイズの設定値の確認やオーバーフローによる更新ログの消失を防ぐことができます。更新ログが消失してしまった場合は、IFAS ファイルのバックアップ後、更新ログファイルのサイズを再計算し、設定を変更してください。

### 更新ログファイル切り替え情報

更新ログファイルが1番から2番、2番から3番…と切り替わるたびにその切り替え情報と、更新ログファイルのリセット情報をイベントログまたは指定されたテキストファイルに出力することができます。更新ログファイルがいつ切り替わったか、1番の更新ログファイルにはどの時点の更新情報が記録されているか、現在運用中のファイルは何かなどの情報を確認することができます。

### 更新ログの消失防止

最後の更新ログファイルが指定ファイルサイズに達してしまった時、その後の更新をエラーとすることができます。

また、最後の更新ログファイルに切り替わった場合や更新ログの消失が発生した場合、そのメッセージをイベントログまたは指定されたテキストファイルに出力するように指定することができます。

更新ログファイルをリセットする時期＝IFAS ファイルをバックアップする時期を判断するための情報として使用できます。

## 6-3-4 更新ログファイルのリセット

更新ログファイルのリセットはIFAS ファイルのバックアップと同期をとって行う必要があります。これは、いままでに何度も説明しているように、更新ログの収集がロールフォワードを行うことを前提としているからです。詳しくは「6-4-1 ディスク障害からの復旧」を参照してください。

IFAS ファイルのバックアップを行わずに更新ログファイルの実体を削除したり、マルチログ運用時にオーバーフローが発生したりすると、更新ログが消失してしまうため、その後再度更新ログを収集しても、消失した部分についてはロールフォワードで復旧することはできません。また、シングルログ運用の場合、更新ログファイルをリセットすることなく更新ログを収集し続けると、更新ログファイルサイズが大きくなり、ディスク容量を圧迫することになります。

このような状態を回避するために、バックアップ支援ユーティリティを用意しています。更新ログファイルのリセット機能は、このユーティリティから使用できるようになっています。

IFAS ファイルのバックアップ完了後、バックアップ支援ユーティリティの更新ログリセット機能を実行します。指定された更新ログファイルがシングルログ運用の場合、更新ログファイルが削除されます。マルチログ運用の場合、それまでに収集された更新ログが無効となり、ログ収集を再開すると、1番の更新ログファイルから更新情報が記録されるようになります。

ロールフォワードは、IFAS ファイルの最終更新時間以降に記録された更新情報を復旧の対象とします。システムの日時、タイムゾーンなどを変更すると、更新時間にずれが生じ、正しく復旧できなくなることがあります。したがって、システムの時刻を変更した場合も、更新ログファイルのリセットを行うようにしてください。

IFAS ファイルのバックアップ方法については「6-2 IFAS ファイルのバックアップ」を、バックアップ支援ユーティリティの使用方法については「7-4-2 バックアップ支援ユーティリティ」を参照してください。

## 6-3-5 更新ログファイルの設計

マルチログ運用を行う場合、1回のバックアップから次のバックアップまでの間に出力される更新ログの量を見積もり、オーバーフローによる更新ログの消失を防ぐよう設計しなければなりません。更新ログは、IFAS ファイルのオープン/クローズ、レコードの更新、IFAS ファイルの変更(ファイルの作成、削除、名前変更、インデックスの削除/追加など)、参照系以外のすべての操作で出力されます。しかし、すべての操作を前提とした場合の更新ログファイルの計算はとてもむずかしいものとなるため、ここでは、レコードの更新処理のみを前提とした簡易的な計算方法を説明します。この計算値に10～20%上乘せした値を更新ログファイルサイズとしてください。

$$Lf = ( (Rs \times 2 + 12) \times Rc ) / ( 1024 \times 1024 )$$
 [切り上げ]

Lf: ログファイルサイズ(MB)

Rs: IFAS ファイルのレコード長(バイト)

複数のIFAS ファイルの更新情報を1つの更新ログファイルに収集する場合、そのIFAS ファイルの中で一番大きいレコード長を使用する

Rc: 更新レコード数

複数のIFAS ファイルの更新情報を1つの更新ログファイルに収集する場合、すべてのIFAS ファイルの更新レコード数を合計した値)

## 6-4 障害時のファイル復旧

不意の電源ダウンやハードディスクの障害が発生すると、IFAS ファイルのデータが壊れたり、ファイルのアクセス自体ができなくなってしまうことがあります。ここでは、そのような障害発生時のファイル復旧方法について説明します。

### 6-4-1 ディスク障害からの復旧

ディスク障害等でIFAS ファイルのデータが破壊され、読み込みや書き込みができなくなった場合、IFAS ファイルが使用できなくなります。このようなIFAS ファイルを復旧する方法は、更新ログを収集している場合とそうでない場合で異なります。

#### 更新ログを収集していないIFAS ファイル

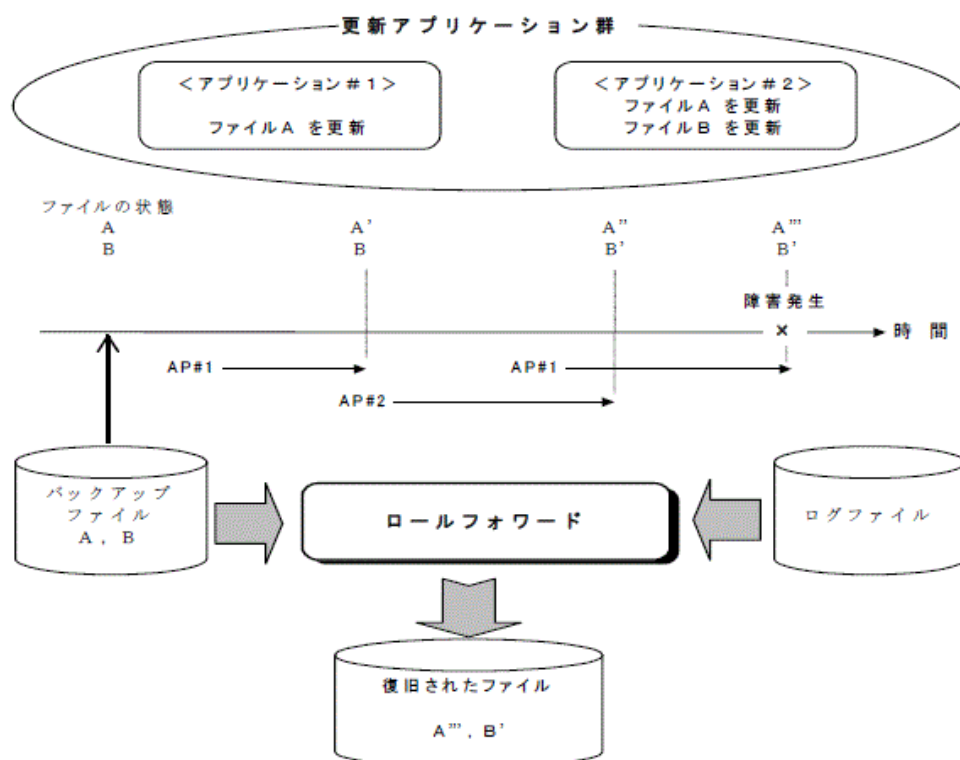
更新ログを収集していないIFAS ファイルは、IFASのユーティリティ等によるファイル復旧はできません。IFAS ファイルのバックアップを行っていた場合は、バックアップファイルからファイルを復元してください。その後、バックアップ時点からの更新を再度行っていただくことになります。バックアップファイルが存在しない場合は、ファイルの初期化から行うことになります。

#### 更新ログを収集しているIFAS ファイル

更新ログを収集しているIFAS ファイルは、IFAS ファイルのバックアップファイルと更新ログファイルを用いて、ロールフォワードを行うことにより、障害発生直前の状態に復旧することができます。したがって、重要なIFAS ファイルについては更新ログを収集することを推奨します。

更新ログの収集については「6-3 更新ログファイルの運用」を参照してください。また、更新ログを収集するための設定方法については、IFAS ファイルを使用する言語により異なりますので、該当する章を参照してください。

下図は障害が発生したIFAS ファイルをロールフォワードで復旧する過程を表しています。ロールフォワードによる詳しい復旧手順については「7-3-1 ロールフォワードユーティリティ」を参照してください。



### 更新ログファイル障害時の対処

ディスク障害により、更新ログファイルにIFAS ファイルの更新情報を書き出せなかった場合、更新ログが消失しているため、ロールフォワードを用いたIFAS ファイルの復旧が行えなくなります。このような時は、即座に障害のあった更新ログファイルに更新情報を記録しているIFAS ファイルのバックアップを行い、更新ログファイルを削除して運用を再開してください。マルチログ運用時は更新ログファイルのリセットも同時に行ってください。

なお、上記のような障害を防止するため、更新ログファイルとIFAS ファイルとは別々のドライブに確保することを推奨します。

## 6-4-2 システム障害からの復旧

不意の電源障害等によりシステム障害が発生した場合、IFAS ファイルに対する更新処理が途中で中断され、システム再起動後にはIFAS ファイルが未クローズ状態となり、使用できなくなります。このようなIFAS ファイルを復旧する方法は、トランザクション処理を行っている場合とそうでない場合で異なります。

### トランザクション処理を行っていないIFAS ファイル

システム障害時、ファイルの更新を行っていた場合は、システム再起動時にファイルが「未クローズロック」状態になり、IFAS ファイルの状態が不整合になっています。この場合は、ファイルロックリセットユーティリティで、未クローズロック状態を解除してください。ファイルロックリセットを行うことにより、IFAS ファイルは論理的に整合状態になりますが、システム障害の直前までのユーザのレコード更新がすべて反映されるわけではありませんので、ご注意ください。

システム再起動時、IFAS ファイルが「未クローズロック」状態にならなかった場合は、障害直前までレコード更新がすべて反映されている状態であり、そのまま使用することができます。IFAS ファイルの状態は、ファイル管理ユーティリティで参照することができます。

ファイルビジーの状態が「未クローズロック状態」にならないければ、ロックリセットユーティリティを使用してファイルの整合性を正しくすることはできません。通常、「未クローズロック状態」になるのはシステム再起動のときかイニシエータの再起動のときです。しかし、アプリケーションの強制終了したときなど、システム再起動を行わずにビジー状態を解除したい場合は、IFAS管理情報メンテナンスツールでビ

ジー状態を解除することができます。詳細は「7-4-4 IFAS管理情報メンテナンスツール」を参照してください。

## トランザクション処理を行っているIFAS ファイル

ロールバックログ<sup>注1</sup>を収集しているトランザクション処理対象のIFAS ファイルは、システム再起動後ロールバックログを用いたリスタートリカバリ処理が自動的に実行され、障害発生直前の状態に復旧されます。したがって、重要なIFAS ファイルについてはトランザクション処理対象ファイルとすることを推奨します。

トランザクション処理を行うための設定方法については、IFAS ファイルを使用する言語により異なりますので、該当する章を参照してください。

**注1.** ロールバックログとは、IFASによって自動的に収集されるものです。トランザクション処理を行うことを宣言したIFAS ファイルのみ収集されます。更新ログとはまったく別のものであるため、更新ログを収集していてもリスタートリカバリ／ディファードリカバリによる復旧はできません。

### リスタートリカバリによるIFAS ファイルの復旧

リスタートリカバリプログラムにより、システム再起動後にIFAS ファイルのロールバック処理が行われます。このリスタートリカバリプログラムは、IFAS FILEイニシエータから自動的に起動されるため、利用者はリスタートリカバリプログラムによるロールバック処理の実行を意識する必要はありません。リスタートリカバリプログラムによるロールバック処理が完了後には、IFAS ファイルが利用できる状態となります。

なお、リスタートリカバリによるIFAS ファイルの復旧状態は、ディファードリカバリユーティリティを起動させて確認することができます。ディファードリカバリユーティリティのリカバリファイル選択画面には、リスタートリカバリに失敗したIFAS ファイルの一覧が表示されます。何らかのファイル名が表示されている場合は、該当ファイルのロールバック処理が正しく行えなかったことを示しています。この場合、リスタートリカバリで復旧できなかった理由とそのIFAS ファイル名がシステムのイベントログに出力されていますので、本ログを参照することによりリスタートリカバリでの障害状況を知ることができます。

リスタートリカバリでロールバックできなかったIFAS ファイルについては、その原因を解消してからディファードリカバリを行い、ファイル復旧を行います。

#### 【注意事項】

- ① IFAS ファイルのオープン処理中に電源障害等でプロセスが終了した場合、IFASのファイル識別子管理ファイル上ではオープンされた状態になっていますがロールバックログにはまだオープン情報が出力されてない状態があり得ます。この場合、リスタートリカバリプログラムではファイル識別子管理ファイル上のロックが解除できないため、ロックの解除を別プロセスが行います。（ロック解除実行プロセスはリスタートリカバリプログラムが自動起動します）  
ロック解除実行プロセスが未クローズロックの解除を実行した場合は、その結果（実行ログ）を以下のファイルに出力します。実行ログは蓄積されるためファイルが大きくなることがあるので、適当な時期にファイルを削除してください。

IFAS FILE、IFASPRO RDBインストールディレクトリ¥AUTOFRST.LOG

- ② リスタートリカバリプログラムは、サービスのアカウントにてIFAS ファイルの復旧操作を行います。したがって、リカバリが必要なIFAS ファイルには、本アカウントからの書き込み権が設定されている必要があります。なお、本アカウントからは、すべてのIFAS ファイルにファイル操作が行えてしまうため、セキュリティの関係で本操作が行えないシステムにおいては、ディファードリカバリによるファイル復旧にて運用を行うようにしてください。

### ディファードリカバリによるファイルの復旧

リスタートリカバリ実行時に障害が発生し復旧されなかったIFAS ファイルは、ディファードリカバリユーティリティにより復旧することができます。

ディファードリカバリユーティリティを起動した直後、「リカバリすべきファイルがありません」というメッセージボックスが表示された場合は、システム起動時に行われたリスタートリカバリが成功している

ため、IFAS ファイルはすべて障害直前(最新の静止点)の状態に復旧されています。前述のメッセージが表示されず、復旧すべきIFAS ファイルの選択が可能な状態となった場合はリスタートリカバリが何らかの原因で失敗しています。復旧されなかったファイルは「リカバリ失敗」ファイルとなり、ディファードリカバリで復旧しないかぎり、このIFAS ファイルは使用することができなくなります。リスタートリカバリ／ディファードリカバリが失敗してしまう主な原因としては、次のことが考えられます。障害内容がその他の理由に起因しているような場合、または下記原因を解消してもディファードリカバリが失敗してしまう場合は速やかに製品窓口までご連絡ください。

＜リスタートリカバリが失敗する主な原因＞

- 復旧対象のIFAS ファイルは、「IFAS FILE Initiator」(ログオンアカウント)からの書き込み権が与えられていない
- 復旧対象のIFAS ファイルが置かれているネットワークドライブが接続されていない

＜ディファードリカバリが失敗する主な原因＞

- 復旧対象のIFAS ファイルに、ディファードリカバリを実行したアカウントに対する書き込み権がない
- 復旧対象のIFAS ファイルが置かれているネットワークドライブが接続されていない

※上記の場合、イベントログにはイベントID＝51021、51022、51023のいずれかが出力されます。

ディファードリカバリによる復旧手順については「7-3-2 ディファードリカバリユーティリティ」を参照してください。

### リカバリプログラムで出力するイベントログ

リスタートリカバリ／ディファードリカバリが何らかの原因でリカバリに失敗した場合、その理由とIFAS ファイル名がイベントログに出力されます。お問い合わせの際は、イベントIDとイベントの詳細に表示されているデータをワード形式で報告してください。

イベントID	理由(原因)
51001	リカバリ処理に失敗しました
51003	IFAS FILEサービス(イニシエータ)が開始されていません。
51005	リカバリプロセスの起動に失敗しました。手動でリカバリプロセスを起動してください。
51007	ロールバックログファイルの読み込みに失敗しました
51009	トランザクション処理を行うことはできません
51011	IFAS ファイルのリカバリに失敗しました(索引順編成ファイル)
51012	IFAS ファイルのリカバリに失敗しました(相対編成ファイル)
51013	IFAS ファイルのリカバリに失敗しました(順編成ファイル)
51021	IFAS ファイルが見つからないか、アクセス権がありません(索引順ファイル)
51022	IFAS ファイルが見つからないか、アクセス権がありません(相対ファイル)
51023	IFAS ファイルが見つからないか、アクセス権がありません(順ファイル)

## 6-4-3 RDB管理ファイルの復旧

RDB管理ファイルに障害が発生した場合も、IFAS ファイルと同様、その障害に応じた方法で復旧できます。管理ファイル復旧後は必ずRDB環境のチェックを行い、定義情報が正しく復旧されていることを確認してください。

### ディスク障害の場合

更新ログを収集していればロールフォワードにより復旧することが可能です。RDB管理ファイルの更新ログは、表定義ユーティリティで更新ログの収集を指定することにより、運用支援ツールの[COBOL設定]タブで指定された更新ログファイルに保存されます。復旧時は、バックアップファイルからRDB管理ファイルを復元後、ロールフォワードを起動します。ロールフォワードを行う時、ファイル名表示方法をRDB表名が表示されるように切り替えると、データファイル名の欄に「管理フ

イル群」という表示がされます。この「管理ファイル群」を選択してロールフォワードを実行してください。

更新ログを収集していない場合は、最新の状態に復旧することはできません。RDB管理ファイルの再作成、またはバックアップファイルからの復元後、定義が消えてしまったRDB表の再定義が必要となります。

復旧方法については「6-4-1 ディスク障害からの復旧」を参照してください。また、ロールフォワードユーティリティの使用法については「7-3-1 ロールフォワードユーティリティ」を参照してください。

### システム障害の場合

RDB管理ファイルはトランザクション処理対象のファイルです。RDB表の作成／削除、インデックスの作成／削除などの操作を行うと、自動的にトランザクション処理が行われます。したがって、RDB表の作成中にシステム障害が発生した場合、システム再起動時にリスタートリカバリによるRDB管理ファイルの復旧が自動的に行われます。

何らかの理由でリスタートリカバリが失敗した場合、RDB管理ファイルは「リカバリ失敗」状態となり、その後のアクセスができなくなります。この場合はディファードリカバリが必要となります。

復旧方法については「6-4-2 システム障害からの復旧」を参照してください。

## 6-5 RDB環境の移行

既存のデータベース環境を他システム等に移行したい場合の移行方法を、以下に示します。

### (1) RDB管理ディレクトリを移行する方法

- ・最初に移動先のディレクトリを作成します。次に移動元のRDB管理ディレクトリ配下のファイルすべてを移動先のディレクトリ下に移動します。この時、RDB管理ファイルには隠し属性が付加されていますのでご注意ください。
- ・表定義ユーティリティのメニュー[環境設定]→[変更]で、RDB管理ディレクトリに移動先のディレクトリを指定します。
- ・本処理でデータベースディレクトリが変更となる場合、(2)の作業も行ってください。

### (2) データベースを移行する方法

#### 既存の実データをともに移行する場合

- ・最初に移動先のディレクトリを作成します。次に移動元データベース配下のファイルすべてを移動先のディレクトリ下に移動します。
- ・表定義ユーティリティから当該データベースを選択し、メニュー[ファイル]→[変更]→[設定変更]で、データベースのディレクトリに移動先のディレクトリを指定します。
- ・表定義ユーティリティから当該データベースを選択し、メニュー[環境設定]→[再構築]で、再構築機能を行います。再構築機能に関する詳細は、「7-2-1 表定義ユーティリティ RDB環境のチェック」を参照してください。

#### 既存の実データを必要とせず定義環境のみを移行する場合

- ・移動先のディレクトリ配下に、実表がないことを確認してください。
- ・移動先のディレクトリは存在する必要はありません。
- ・表定義ユーティリティから当該データベースを選択し、メニュー[ファイル]→[変更]→[設定変更]で、データベースのディレクトリに移動先のディレクトリを指定します。
- ・表定義ユーティリティから当該データベースを選択し、メニュー[環境設定]→[再構築]で、再構築機能を行います。再構築機能に関する詳細は、「7-2-1 表定義ユーティリティ RDB環境のチェック」を参照してください。

### (3) RDB表を移行する方法

表定義ユーティリティのRFD出力機能によりRFDファイルを作成し、実表とともに保存してあるRDB表に対してのみ、以下の方法でRDB表を移行することができます。



### 既存の実データをともに移行する場合

- ・ 移動先のディレクトリ配下に、該当する実表がないことを確認してください。
- ・ 移動先のディレクトリは存在する必要はありません。
- ・ 移動元のRDB表とそれに対応するRFDファイルが同じディレクトリ直下にあることを確認してください。
- ・ 表定義ユーティリティで、移動元のRDB表に対するRFDファイルを選択したRFD入力機能を行います。この時、データベースがなければ移動先のディレクトリを指定します。RFD入力機能に関する詳細は、「7-2-1 表定義ユーティリティ RFDファイルを利用した表定義」を参照してください。
- ・ 表定義ユーティリティからRFD入力で作成したRDB表を選択し、インポート機能を利用して、移動元RDB表のデータ移行を行います。この時、移動元RDB表は索引順編成ファイルとして選択を行います。インポート機能に関する詳細は、「7-2-1 表定義ユーティリティ データのインポート」を参照してください。

### 既存の実データを必要とせず定義環境のみを移行する場合

- ・ 移動先のディレクトリ配下に、該当する実表がないことを確認してください。
- ・ 移動先のディレクトリは存在する必要はありません。
- ・ 表定義ユーティリティで、移動元のRDB表に対するRFDファイルを選択したRFD入力機能を行います。この時、データベースがなければ移動先のディレクトリを指定します。RFD入力機能に関する詳細は、「7-2-1 表定義ユーティリティ RFDファイルを利用した表定義」を参照してください。

## 6-6 未クローズファイル

IFAS ファイルまたはIFAS RDB表を更新処理しているアプリケーションが、何らかの原因で突然異常終了したような場合、ファイルがIFASの管理上オープンされたままの状態になる可能性があります。また、システム再起動後に、ファイルのオープンを行うと、1009(0x03f1) というエラーステータスが返却されてファイルがオープンできなくなる場合があります。このような状態にあるファイルは正しくクローズ処理が行われなかったファイルであり、ここでは「未クローズファイル」と称します。

### 未クローズファイルの保護

未クローズファイルは論理的にデータが正しくない状態(ファイルラベル情報と実際のデータ状態の不一致等)である可能性が高いため、Windowsシステムの再起動時にIFAS FILEイニシエータはファイル保護のために、正しく終了処理が行われなかったファイルにロックをかけてオープンできないようにしています。

未クローズファイルを再利用するためには「ファイルロックリセットユーティリティ」または「ファイル管理ユーティリティ」でロック解除を行う必要があります。ファイルロック解除機能については「7-1-1 ファイル管理ユーティリティ」「7-3-3 ファイルロックリセットユーティリティ」を参照してください。

### 未クローズ状態となる原因

IFAS FILE、IFASPRO RDBは終了処理機能を実装していますので、通常はアプリケーションが異常終了してもファイルは自動クローズされ、未クローズ状態になることはありません。

しかし以下のような操作や処理を利用者が行った場合は、未クローズファイルになる可能性が高くなります。また、OSの種類やIFAS製品の運用形態によって未クローズファイルとなる条件は異なります。

- (1) WS-EML(ワークステーションエミュレータ)からプログラム放棄する。
- (2) プロセスビューアからプログラムを終了する。
- (3) タスクマネージャからタスクを終了する。
- (4) タスクマネージャからプロセスを終了する。
- (5) CTRL+Cでプログラムを終了する。
- (6) ウィンドウを閉じて(×クリック)プログラムを終了する。
- (7) コントロールメニューから終了する。
- (8) システムをシャットダウンする。
- (9) システムをログオフする。
- (10) 電源断。

なお、上記操作や処理は、クライアント／サーバ環境で運用している場合はクライアント側の操作や処理という前提です。

記号の説明

(○:未クローズにならない ×:未クローズになる △:一部未クローズになる -:対象外)

IFAS製品	利用者の動作/処理									
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
スタンドアロン	○	○	○	○	○	○	○	△ 注1	△ 注2	×
クライアント/サーバ 接続	-	○	○	○	○	○	○	○	○	○

注1 GUIアプリケーションの場合、ファイルは未クローズになり、システムの再起動によってこれらのファイルは未クローズロックされます。

注2 GUIアプリケーションの場合、システムの再起動後にファイルは未クローズロックされます。

ファイルが未クローズ状態になることを未然に防ぐためには、上記の条件表を参考にして運用方法を選択してください。

## 未クローズロックの自動解除

IFAS FILE、IFASPRO RDBの以下の機能を利用することによって、システム再起動時に未クローズファイル自動的に復旧することができます。

### (1) トランザクションオプション

更新処理を行うファイルに対してトランザクションオプションを設定すると、そのファイルに対するロールバックログが収集され、システム再起動の際に自動的にリカバリ処理(ロールバック処理)が行われ、プログラム終了やシステムダウン直前のCOMMIT命令実行時のデータまで保証されます。

つまり、トランザクションオプションが指定されたファイルは、障害があつてシステム再起動した後も未クローズロック状態にはなることはありません。

ただし、トランザクションオプションを使用する場合は、トランザクション処理(COMMIT/ROLLBACK命令)を考慮したプログラム設計が必要になりますのでご注意ください。

トランザクションオプションおよびトランザクション機能については「3-1-3 トランザクション処理」「4-1-2 データアクセス機能 (3)トランザクション機能」等を参照してください。

### (2) 自動ロックリセット機能

運用支援ツールの「その他」タブで「立ち上げ時にロックリセット機能を実行する」をチェックしておくと、システム再起動時にすべての未クローズファイルを検出し、自動的にロックリセット処理を行います。これにより、利用者はユーティリティによって未クローズファイルをロック解除する必要がなくなりますが、逆に未クローズファイルがあつたことを知る事ができなくなります。

ロックリセットによるファイルの復旧はプログラム終了やシステムダウン直前までの更新命令によるデータまで保証できるとはかぎりませんので、完全なデータ復旧を望まれる場合はトランザクションオプションの利用を推奨します。

自動ロックリセット機能の設定方法については「7-4-1 運用支援ツール」を参照してください。

## 7. ユーティリティ

IFAS FILE、IFASPRO RDBは、ファイル/DBシステムとして利用者の運用性や保守性を向上するためにさまざまな用途に応じた各種ユーティリティプログラムを実装しています。

IFAS FILE、IFASPRO RDBのユーティリティプログラムは主に、IFAS ファイルの作成/保守やデータの複写/移行を目的としたファイル操作を行うユーティリティ、IFAS RDB表の定義やデータ移行を行うためのユーティリティ、システムやハードウェア障害によるデータ破壊からファイルを復旧するためのユーティリティ、システム運用をサポートするためのユーティリティに大別されます。

### 7-1 一般ファイルの管理

#### 7-1-1 ファイル管理ユーティリティ

ファイル管理ユーティリティは、IFAS FILE、IFASPRO RDBが認識しているIFAS ファイルのアクセス数／状態等の一覧表示、オプション指定を簡略化したファイル作成、同一ファイル編成間でのコピー／移動、ファイルのロックリセット、およびデータの全件削除等を、簡易的な操作で行うための統合ユーティリティです。クライアント／サーバ機能も搭載していますので、サーバ上のファイル操作も可能です。

#### 機能概要

ファイル管理ユーティリティは、以下の機能を提供します。

##### ファイル一覧表示機能

IFAS FILE、IFASPRO RDBが認識しているIFAS ファイルの情報を一覧表示します。表示情報は、ファイル名、作成編成、RFDファイルの有無、アクセス数、状態、使用編成、備考です。

表示情報は時間が経過すれば古くなるため、適時一覧表示のファイル情報を最新に更新してからファイル操作を行ってください。ファイル情報の更新方法は、「メニューバー／ツールバーの説明」を参照してください。

##### ファイル詳細情報表示機能

選択されたIFAS ファイルの詳細情報を表示します。表示情報は、ファイル名、パス名、ファイル作成編成、レコード追加方式(索引順編成ファイルのみ)、ファイル作成日、最終更新日、レコード長、ファイルサイズ、最大レコード番号、有効レコード数、インデックス情報(索引順編成ファイルのみ)です。レコード追加方式では、索引順編成ファイルのレコード追加時に削除領域の使用の有無を設定することができます。インデックス情報では、インデックス番号、キー重複可否、キー状態(整合／不整合)、キー属性(更新／非更新／実行時生成)、キー情報(オフセット／サイズ／タイプ)が表示されます。

##### ファイル作成機能

ファイル編成、ファイル名、レコードサイズを指定するだけで、簡単に指定ディレクトリにIFAS ファイルを作成します。

索引順編成ファイルを作成する場合は、プライマリインデックスなしのファイルを作成します。

##### ファイル削除機能

選択されたIFAS ファイルを削除します。

##### ファイルロックリセット機能

更新処理中にシステムダウン等が発生し、未クローズ状態のままになってしまったIFAS ファイルのファイルロックを解除します。

ファイルロックリセットユーティリティと同等の機能を持ちます。

##### データ全件削除機能

選択したIFAS ファイル内のレコードをすべて削除します。

索引順編成、順編成ファイルについては、ファイル容量も縮小されます。

**ファイルコピー／移動機能**

メニューバー／ツールバーの切り取り／コピー／貼り付け機能を使用して、同一のファイル編成／ファイル名で、コピーや移動を行います。他マシンとのやり取りはできません。

**ファイルコピーユーティリティ／インデックスメンテナンスユーティリティの連携機能**

選択したIFAS ファイルをコピー元ファイルとして、同一編成、異編成へのコピー、テキストファイルへのコピーを行うために、ファイルコピーユーティリティを起動します。

選択した索引順編成ファイルのインデックス操作を行うため、インデックスメンテナンスユーティリティを起動します。

起動したユーティリティは、ファイル管理ユーティリティを終了させても、連動して終了しません。各起動ユーティリティで終了してください。

**APDIRECTOR登録機能**

IFAS ファイルの情報をAPDIRECTORの資産データベースに登録します。

※ただし、現在はAPDIRECTORが対応していないため、本機能は利用できません。

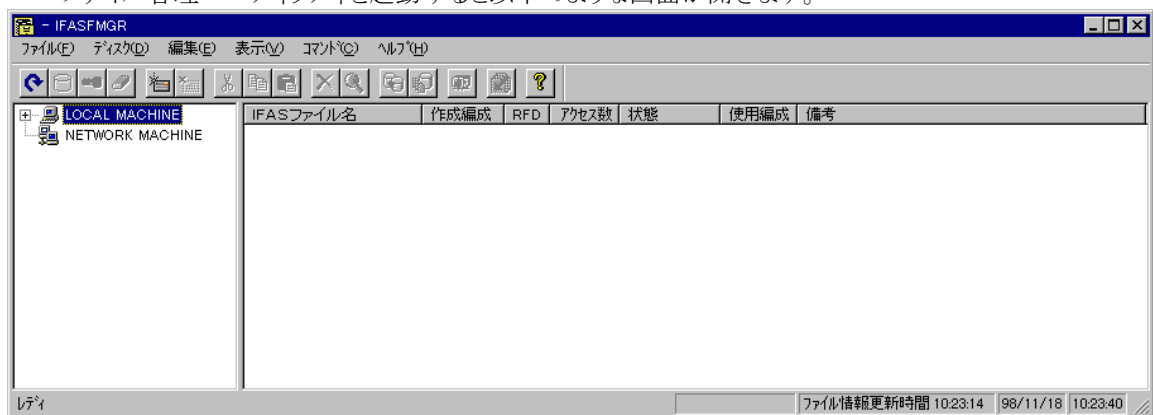
**ファイル管理情報登録機能**

ファイル識別子管理ファイル(DRSファイル)へIFAS ファイル情報を登録します。

IFASの再インストールなど、DRSファイルが失われた場合、ファイル管理ユーティリティでIFAS ファイル情報の取得／操作が行えなくなりますが、この機能で復旧させることができます。

## 表示画面の説明

ファイル管理ユーティリティを起動すると以下のような画面が開きます。

**ディレクトリ／一覧表示の説明**

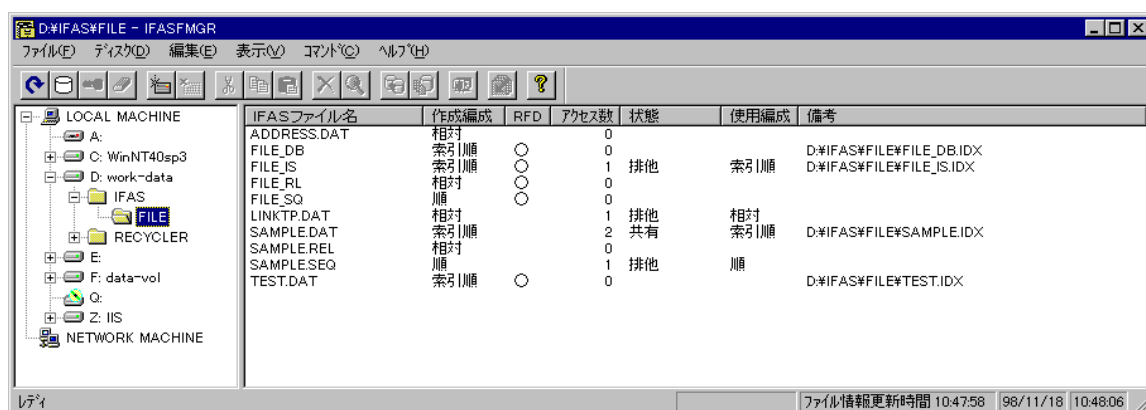
画面の左にディレクトリのツリービュー、右にIFAS ファイルの一覧を表示します。

起動直後のツリービューには「LOCAL MACHINE」と「NETWORK MACHINE」のツリーがあり、「LOCAL MACHINE」配下に起動マシンのディレクトリが表示され、「NETWORK MACHINE」にはサーバマシンと接続することにより、共有ディレクトリを表示することができます。

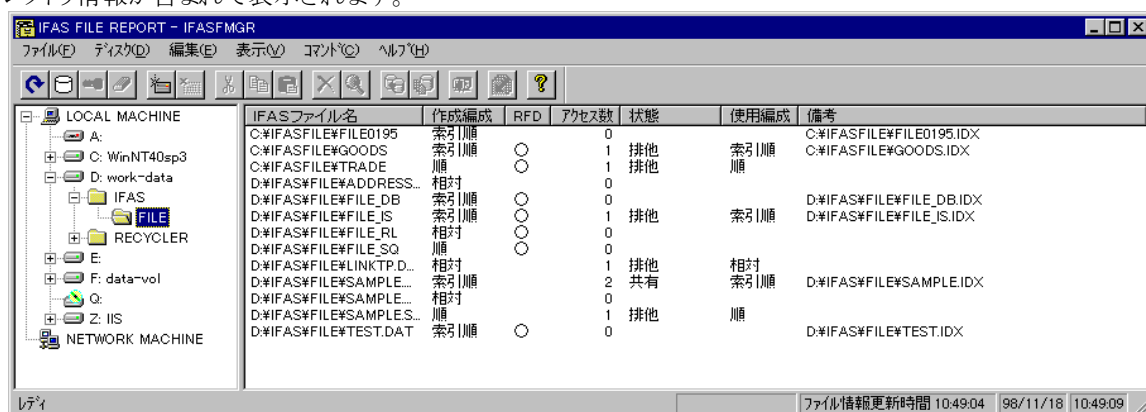
IFAS ファイルの一覧表示には、ディレクトリのツリービューと連動して表示するか、すべてのIFAS ファイルを一覧表示するかの二種類の表示方法があります。

表示方法の切り替えは、メニューバーの[表示]メニューの[ファイル情報をツリービューと連動する]の項目をチェックするか否かでを行います。

[ファイル情報をツリービューと連動する]をチェックした場合は、ファイル情報がツリービューと連動し、ツリービューで選択したディレクトリ下にあるIFAS ファイルの一覧を表示します。



[ファイル情報をツリービューと連動する]をチェックしない場合は、ファイル情報がツリービューと連動しなくなり、すべてのIFAS ファイルの一覧を表示します。この場合、一覧表示内のファイル名表示部分にはディレクトリ情報が含まれて表示されます。



IFAS ファイルの一覧表示は、IFAS ファイル名の項目でソートされて表示されていますが、列見出しをクリックすることにより、それぞれの項目についてソートすることができます。

ファイルロックリセット、データ全件削除等のファイル操作を行う場合は、一覧表示されているIFAS ファイル名を選択して実行します。また、ファイル作成、ファイルの貼り付け(コピー/移動)は、ツリービューのディレクトリを選択して実行します。

一覧表示項目

#### ① IFAS ファイル名

ディレクトリのツリービューと連動していた場合は、ファイル名のみ表示します。ツリービューと連動していない場合は、ディレクトリを含んでファイル名が表示されます。  
索引順編成ファイルの場合は、データファイルの備考欄にインデックスファイル名が表示され、インデックスファイルは一覧表示されません。

#### ② 作成編成

ファイルの初期化、作成した時の編成が表示されます。  
ただし、相対編成/順編成にインデックスファイルを追加作成した場合は、作成編成は索引順編成に変わります。

#### ③ RFD

ファイルにRFDファイルが存在するか否かを表示します。これは、ファイル名の拡張子を.RFDに変更したファイルが存在するかどうかを表示していますので、RFDファイルの内容とファイルの定義が一致しているかどうかは判断しません。

#### ④ アクセス数

ファイル管理ユーティリティを起動した時点、または、ファイル情報が更新された時点でアクセスしているユーザ数です。  
ステータスバーのファイル情報更新時間にて、ファイル情報が更新された時間がわかります。

#### ⑤ 状態

ファイルの状態を 排他/共有/未クローズ/リカバリ失敗/クローズ失敗等 で表示します。

排他

排他モードでファイルが使用中です。

共有

共有モードでファイルが使用中です。

未クローズ

更新処理中にシステムダウン等が発生し、未クローズ状態のままになってしまったIFAS ファイルです。

リカバリ失敗

リスタートリカバリが失敗したIFAS ファイルです。

クローズ失敗

クローズが失敗したIFAS ファイルです。

#### ⑥ 使用編成

ファイルが使用中の場合、使用しているファイル編成を表示します。

作成編成が索引順編成のIFAS ファイルは、相対編成、順編成として使用することができます。

#### ⑦ 備考

索引順編成ファイルの場合、インデックスファイル名が表示されます。

### タイトルバー

メニューバーの[表示]メニューの[ファイル情報をツリービューと連動する]のチェックがされている場合、現在選択しているディレクトリを表示します。

### ステータスバー

左側から

メニューやツールバーを選択する時などに簡単な説明を表示します。

表示しているマシンでバックアップ支援ユーティリティが起動している場合、その旨表示します。

ファイル一覧表示の情報の更新時間を表示します。

現在の日時を表示します。

## メニューバー／ツールバーの説明

### ファイル(F)



#### IFAS ファイル作成(N)

選択されたディレクトリを作成ディレクトリとしてIFAS ファイル作成のダイアログを表示します。ファイル形式等を指定し、「OK」ボタンをクリックしてIFAS ファイルを作成します。

指定する項目

形式

ファイル編成を選択します。

作成ディレクトリ

ディレクトリツリービューで選択したディレクトリが表示されています。違うディレクトリ下を作成する場合は修正してください。

ファイル名

作成するファイル名を指定します。

インデックスファイル名

相対編成／順編成の場合は指定できません。

索引順編成の場合は、インデックスファイル名を指定する必要がありますが、以下のインデックスファイル名をCOBOL仕様とする項目をチェックすると指定できなくなります。

インデックスファイル名をCOBOL仕様とする

チェックした場合、COBOL仕様に準じたインデックスファイル名が自動で作成されます。

ファイル名項目で指定されたファイル名の拡張子部分を”.IDX”と変更してインデックスファイル名が作成されます。

本設定は、ファイル管理ユーティリティの環境設定ファイルへ保存され、次回起動時にも有効となります。

レコード長

レコード長を指定します。0から65536バイト内で指定してください。0を指定した場合は、既定値256バイトを指定したことになります。

指定する項目が不足／誤っていた場合は、その旨の確認メッセージボックスを表示します。



### 削除(D)

選択されたIFAS ファイルを削除します。以下の確認用メッセージボックスを表示しますので、「OK」ボタンをクリックして該当IFAS ファイルを削除します。



### プロパティ(R)

選択されたIFAS ファイルの詳細情報を表示します。

該当ファイルが使用中や未クローズなどの状態の場合、使用中である旨の確認メッセージボックスを表示し、詳細表示をしません。

ファイル詳細	
ファイル名	: TEST.DAT
パス	: D:\IFAS\FILE\
<hr/>	
インデックス ファイル名	: TEST.IDX
パス	: D:\IFAS\FILE\
<hr/>	
ファイル作成編成	: 索引順
レコード追加方式	: 削除領域使用 <span>方式変更</span>
ファイル作成日	: 1998/07/03 15:06:28
最終更新日	: 1998/07/03 15:24:35
レコード長	: 256 バイト
ファイルサイズ	: 520000128 バイト
最大レコード番号	: 2000000
有効レコード件数	: 2000000

インデックス  
OK

表示内容

ファイル名 & パス

選択されたIFAS ファイル名とそのIFAS ファイルが存在するディレクトリを表示します。

インデックスファイル名 & パス

索引順編成の場合に、選択されたIFAS ファイルに対応するインデックスファイル名とインデックスファイルが存在するディレクトリを表示します。

ファイル作成編成

ファイルを作成した時のファイル編成を表示します。

ただし、相対編成／順編成ファイルに対して、追加でインデックスファイルを作成した場合、作成編成は索引順編成になります。

## レコード追加方式

索引順編成のレコード追加の場合に、レコード削除によってできた削除領域を使用するか否かを設定変更することができます。[方式変更]ボタンで選択し、[OK]ボタンで設定変更を行います。

## ファイル作成日

ファイルを作成した日時を表示します

ただし、新規にファイル作成した場合は、最終更新日より新しい日時となる場合があります。

## 最終更新日

該当ファイルを最後に更新した日時を表示します。

## レコード長

レコード長を表示します。

## ファイルサイズ

ファイルサイズを表示します。

## 最大レコード番号

順編成ファイルの場合は、有効レコード数と同じ値を表示します。

相対編成ファイルの場合は、格納可能なレコード数を表示します。

索引順編成ファイルの場合は、今までに格納された最大レコード番号を表示します。

## 有効レコード件数

有効なレコード数を表示します。

## 「インデックス」表示内容

このボタンがグレーアウトされている場合は、インデックスは存在しません(未作成状態)。

## ・インデックス番号

現在表示されているインデックスの番号を表示します。

## ・キー重複

インデックスが重複キーを許すかどうかを表示します。

## ・状態

レコードに対するインデックスの状態(整合／不整合)を表示します。

## ・属性

インデックスの更新属性を表示します。

## ・キーオフセット／キーサイズ／キータイプ

インデックスを構成するキーパートのレコード内のオフセット／サイズ／タイプを表示します。

## アプリケーションの終了(X)

ファイル管理ユーティリティを終了します。

## ディスク(D)



## 他マシン情報の取得(N)

サーバ上のファイル情報の取得／操作を可能にします。

※本機能を利用する場合は、"Administrator"ユーザで実行する必要があります。

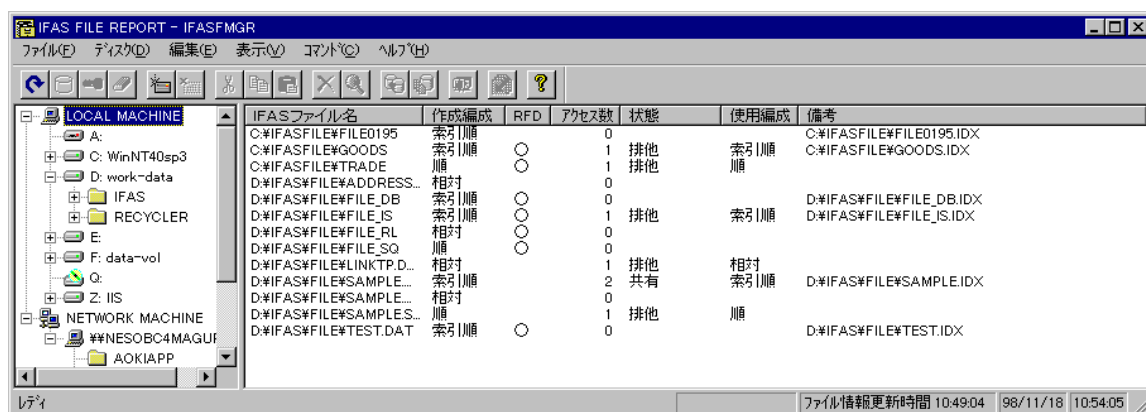
ツリービューで「NETWORK MACHINE」配下を選択すると、実行できるようになります。

以下のダイアログボックスにコンピュータ名を書き、「OK」ボタンをクリックすることで、指定コンピュータのディレクトリが表示されます。ただし、表示されるのは共有されているディレクトリとそのサブディレクトリです。

本接続はIFASのクライアント／サーバ機能を利用してサーバマシン上のファイル識別子管理ファイル(DRSファイル)情報を表示します。実際にOSのネットワークマウントが行われるわけではありません。

接続が正常に行われると「NETWORK MACHINE」配下にコンピュータ名が表示され、サーバマシンの共有ディレクトリ内のIFAS ファイルを操作することができるようになります。





### 他マシン情報の切り離し(C)

選択されている他マシン情報の接続を解除します。

### 編集(E)



### 切り取り(T)

選択されたIFAS ファイルを切り取ります。

この後[貼り付け(P)]を行うことで選択したIFAS ファイルを移動することができます。



### コピー(C)

選択されたIFAS ファイルをコピーします。

この後[貼り付け(P)]を行うことで選択したIFAS ファイルをコピーすることができます。



### 貼り付け(P)

[切り取り(T)]、または[コピー(C)]したIFAS ファイルをツリービューで選択したディレクトリに貼り付けます。

該当IFAS ファイルが使用中／既に選択ディレクトリに存在する等の処理エラーが発生した場合、その旨の確認メッセージボックスを表示します。

複数ファイルの貼り付け途中で処理エラーが発生した場合は、その旨の確認メッセージボックスを表示後、該当ファイルをスキップして次のファイルへの処理を続行します。

### 表示(V)

#### ツールバー(T)

ツールバーの表示／非表示を切り替えます。

#### ステータスバー(S)

ステータスバーの表示／非表示を切り替えます。

#### ファイル情報をツリービューと連動する(V)

ファイル情報の一覧表示をツリービューで選択したディレクトリのみ表示するか、選択したディレクトリに関わらずすべてのファイルを表示するかを切り替えます。

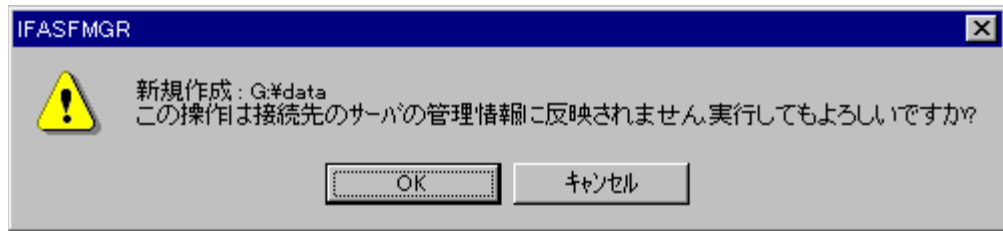
本設定は、ファイル管理ユーティリティの環境設定ファイルへ保存され、次回起動時にも有効となります。

#### ネットワークドライブへの操作を監視する(N)

ネットワークドライブ接続を行っていて、ネットワークドライブ上に存在するIFAS ファイルの操作を行おうとした場合、警告メッセージを表示するか否かを切り替えます。

ネットワークドライブ上のIFAS ファイルは本ユーティリティからはローカルファイルとして認識されます。

クライアントアプリケーションからクライアント・サーバ機能を介して使用しているサーバ上のIFAS ファイルを本ユーティリティで操作した場合、IFASサーバを経由しないアクセスになりますので、そのファイルを同時にアクセスしている他のクライアントアプリケーションとの排他制御が正しく行えません。そのため、ネットワークドライブのファイル操作(新規作成、ロックリセット、全件削除、コピー、移動、削除)に対して注意を促すために警告メッセージボックスを表示します。「キャンセル」ボタンをクリックすることで該当ファイル操作をキャンセルすることができます。



本設定は、ファイル管理ユーティリティの環境設定ファイルへ保存され、次回起動時にも有効となります。



#### ファイル情報更新(L)

IFAS ファイル情報を保持しているファイル識別子管理ファイル(DRSファイル)から IFAS ファイル情報を再読み込みして、一覧表示、および、ステータスバーの更新時間を更新します。

ファイル情報は、時間が経過すれば古くなるため、適時、本機能によりファイル一覧情報を更新してください。

#### コマンド(C)



#### ロックリセット(L)

選択したIFAS ファイルのファイルロックを解除します。

複数選択された場合は、選択された各IFAS ファイル毎に処理されます。

複数選択してのロックリセット途中で処理エラーが発生した場合、その旨の確認メッセージボックスを表示後、該当ファイルをスキップして次のファイルへの処理を続行します。

#### 一括ロックリセット(A)

ファイルロック状態の全IFAS ファイルのロックを解除します。

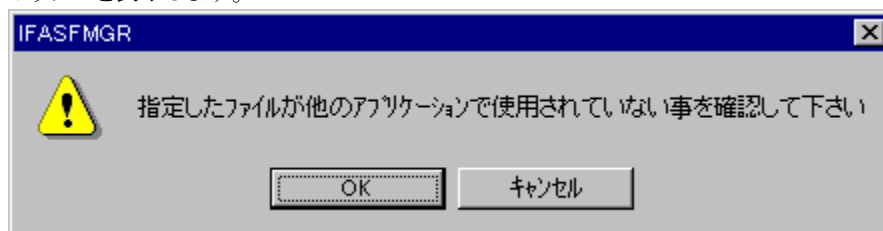
ロックリセットは各IFAS ファイル毎に処理されます。

途中で処理エラーが発生した場合、その旨の確認メッセージボックスを表示後、該当ファイルをスキップして次のファイルへの処理を続行します。



#### データ全件削除(D)

以下の確認メッセージボックスを表示します。「OK」ボタンをクリックして選択されたIFAS ファイル内のレコードをすべて削除します。複数ファイル選択した場合も最初の一回のみ確認メッセージボックスを表示します。



複数選択された場合は、選択された各IFAS ファイル毎に処理されます。

該当ファイルが使用中や未クローズなどの状態で処理エラーが発生した場合、その旨の確認メッセージボックスを表示し処理を中断します。

複数ファイル選択でのデータ全件削除途中で処理エラーが発生した場合、その旨の確認メッセージボックスを表示後、該当ファイルをスキップして次のファイルへの処理を続行します。



#### ファイルコピー起動(C)

選択したIFAS ファイルをコピー元としてファイルコピーユーティリティを起動します。

ファイルコピーユーティリティの詳細は、「7-1-2 ファイルコピーユーティリティ」の説明を参照してください。



#### インデックスメンテナンス起動(I)

メニューバーから実行する場合は以下のサブメニューを選択します。

##### ADDIDX(A)

選択したIFAS ファイルを用いて、インデックスメンテナンスユーティリティをインデックスの追加処理として起動します。

##### DELIDX(D)

選択したIFAS ファイルを用いて、インデックスメンテナンスユーティリティをインデックスの削除処理として起動します。

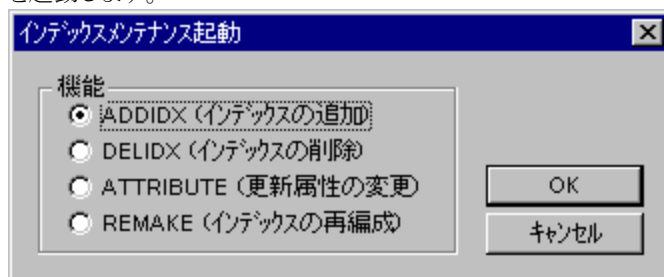
#### ATTRIBUTE(T)

選択したIFAS ファイルを用いて、インデックスメンテナンスユーティリティを更新属性の変更処理として起動します。

#### REMAKE(R)

選択したIFAS ファイルを用いて、インデックスメンテナンスユーティリティをインデックスの再編成処理として起動します。

ツールバーのボタンから起動した場合は、以下のインデックスメンテナンスの起動ダイアログが表示されます。処理したい機能を選択し、“OK”をクリックしてインデックスメンテナンスユーティリティを起動します。



該当ファイルが索引順編成ファイルでない場合、その旨の確認メッセージボックスを表示し処理を中断します。

インデックスメンテナンスユーティリティの詳細は、「7-1-3 インデックスメンテナンスユーティリティ」の説明を参照してください。



#### インデックス再作成(R)

選択されたIFAS ファイルのインデックス再作成を行います。

なお、再編成方式は“データレコードの詰め直し”が指定されます。

その他の再編成方式を行う場合、インデックスメンテナンスユーティリティで行ってください。

インデックスメンテナンスユーティリティの詳細は、「7-1-3 インデックスメンテナンスユーティリティ」の説明を参照してください。



#### APDIRECTOR登録(S)

選択されたIFAS ファイルをAPDIRECTORの資産データベースに登録します。

※ただし、現在はAPDIRECTORが対応していないため、本機能は利用できません。

#### ファイル管理情報登録(E)

選択されたディレクトリ配下のIFAS ファイルをファイル識別子管理ファイル(DRSファイル)へ登録します。サブディレクトリの情報を登録するかどうかを選択することができます。

#### ヘルプ(H)



#### IFASFMGRの使い方(H)

オンラインヘルプを表示します。

#### バージョン情報(A)

バージョン情報、著作権情報を表示します。

## 注意事項

1. IFAS FILE、IFASPRO RDBが認識しているIFAS ファイル  
IFAS FILE、IFASPRO RDB、一度使用されたIFAS ファイルの情報をファイル識別子管理ファイル(DRSファイル)に保持し、IFAS ファイルのファイル名管理やファイル排他制御等のために使用します。  
このため、IFAS FILE、IFASPRO RDBのAPI、ユーティリティを使用せずに複写または移動したIFAS ファイルは、一度参照／更新することで、DRSファイルにファイル情報が登録され、IFAS FILE、IFASPRO RDBが認識している状態となります。
2. インストール  
以前のバージョンをアンインストールして、本バージョンをインストールした場合、または、別ディレク

トリにインストールした場合、IFAS ファイル情報を保持するファイル識別子管理ファイル(DRSファイル)が初期化されます。既存IFAS ファイルをDRSファイルに登録する場合には、ファイル管理情報登録機能を使用してください。それ以外では、再度既存IFAS ファイルが参照等使用された時点で、DRSファイルにファイル情報が保持され、ファイル管理ユーティリティで一覧表示できるようになります。

このため、バージョンアップ直後にファイル管理ユーティリティを起動するとIFAS ファイルが1ファイルも表示されないことがあります。以降IFAS ファイルを参照等使用する毎に一覧表示できるIFAS ファイル数も増えることになります。

#### ☆注意☆

バージョンアップにより、DRSファイルのIFAS ファイル情報が初期化されるだけで、利用者のIFAS ファイルが初期化されることはありません。

3. ネットワークに接続しているドライブへの操作  
ネットワークドライブ上に存在するIFAS ファイルの操作を行った場合、ネットワークドライブ先のサーバに存在するIFAS FILE、IFASPRO RDBのファイル識別子管理ファイル(DRSファイル)にIFAS ファイル情報が反映されません。(他マシン情報の取得機能で接続された「NETWORK MACHINE」配下のディレクトリで実行されたIFAS ファイルの操作はサーバ上のDRSファイルに反映され、現在起動しているマシン内のDRSファイルには反映されません。)  
このため、ネットワークドライブ上のIFAS ファイルを操作する場合、警告メッセージを出力します。この警告メッセージは、メニュー[表示]の[ネットワークドライブへの操作を監視する]のチェックを外すことで表示しなくなります。
4. 二重起動  
ファイル管理ユーティリティは、二重に起動できません。既に起動している場合に、新規に起動し直したい場合は、既に起動しているファイル管理ユーティリティを終了させてから、再度起動してください。
5. IFAS FILE、IFASPRO RDB(API)のエラー  
ファイル操作等にて、IFAS FILE、IFASPRO RDB(API)からエラーが戻った場合、その旨の確認メッセージボックスを表示します。

## 7-1-2 ファイルコピーユーティリティ

ファイルコピーユーティリティは、IFAS ファイルのデータを別のIFAS ファイルにコピー(またはデータの追加)したり、MS-DOS形式のテキストファイルデータ、CSV形式データからIFAS ファイル(各編成ファイル)にデータを登録するための、簡易データ入出力用のユーティリティです。

### 機能概要

ファイルコピーユーティリティは、IFAS索引順編成ファイル、IFAS相対編成ファイル、IFAS順編成ファイル、MS-DOSテキストファイル、CSV形式ファイル、相互間のデータコピーを行うためのユーティリティです。出力ファイルに入力ファイルのデータを追加することも可能です。

#### IFAS ファイル(同編成)間のデータコピー

たとえば、削除レコードエリアができた索引順編成ファイルを新しい索引順編成ファイルにコピーすると、出力ファイルには削除レコードエリアが作られないためファイルサイズを縮小させることができます。

#### IFAS ファイル(異編成)間のデータコピー

たとえば、順編成ファイルから索引順編成ファイルを作成することができます。

#### IFAS ファイルとテキストファイル間のデータコピー

たとえば、索引順編成ファイルのデータをインデックスキー順、レコード番号順にテキストファイルにコピーして、エディタ等でデータ内容を確認することができます。また、テキストファイルからIFAS ファイルを作成することができます。テキストファイルを入力ファイルとする場合は複数ファイル一括入力指定が可能です。

#### IFAS ファイルとCSV形式ファイル間のデータコピー

たとえば、MS-ACCESSなどで加工したデータをCSV形式ファイルに出力することによって、それを使用してIFAS一般ファイルに出力することができます。

### 一時ファイル機能

一時ファイルは「アプリケーション実行環境」および「COBOL アプリケーション実行環境」が提供するジョブ管理配下での動作時のみ使用可能なファイルです。したがって、バッチファイルでの実行時に使用され、GUIから使用することはできません。しかし、GUIで一時ファイルを使用する定義ファイルを作成することはできます。

ファイルコピーユーティリティには、32ビット版と64ビット版があり、それぞれ以下のような特徴および制約があります。

#### ① 32ビット版のファイルコピーユーティリティ(IFASCOPY.EXE)

- ・ MS-DOSテキストファイルを入力ファイル(コピー元ファイル)にする場合は複数ファイル一括入力(最大30個)が可能です。ただしMS-DOSテキストファイル間のコピーは行えません。
- ・ CSV形式ファイルの入出力に対するIFAS一般ファイルの入出力には、IFAS一般ファイルに対応したRFDファイルが必須となります。ただし、IFAS一般ファイル間、MS-DOSテキストファイル/IFAS一般ファイル間では、RFDファイルは参照されません。
- ・ CSV形式ファイル間、CSV形式ファイル/MS-DOSテキストファイル間のコピーは行えません。
- ・ 索引順編成ファイルを入力ファイル(コピー元ファイル)にする場合は、レコード入力順をインデックスキー昇順/降順、レコード番号昇順、とそれぞれ選択できます。
- ・ 同一編成ファイルどうしのコピーの場合は、入力ファイルのファイル属性をそのまま継承してコピーできます。(出力ファイルの定義を手動で行う必要がありません)
- ・ 画面で設定した定義情報を定義ファイルとして保存/再利用することができます。この定義ファイルはエディタ等で作成することも可能で、さらにコンソールコマンドから以下のようにして実行することもできます。
- ・ 一時ファイル機能を使用するためには、「**アプリケーション実行環境**」が必要です。

#### ② 64ビット版のファイルコピーユーティリティ(IFASCOPY64.EXE)

以下を除き、32ビット版のファイルコピーユーティリティと同じ機能です。

- ・ 一時ファイル機能を使用するためには、「**COBOL アプリケーション実行環境**」が必要です。
- ・ CSV形式ファイルの入出力はできません。
- ・ GUI(画面操作)機能は使用できません(コマンド機能としてのみ使用可)。なお、32ビット版のファイルコピーユーティリティ画面で作成した定義ファイルはIFASCOPY64.EXEで利用可能です。

## IFASCOPY 定義ファイル名

ファイルコピーユーティリティをアイコンから起動すると以下のような画面が開きます。



## メニューバー／ツールバーの説明

### ファイル(F)



#### 定義ファイルの新規作成(N)

新規にファイルコピーユーティリティのウィンドウを開きます。



#### 定義ファイルの読み込み(O)

画面上で設定した情報は、定義ファイルとして保存することができます。この定義ファイルを読み込むことで以前設定した定義情報を再定義する必要はありません。



#### 定義ファイルの上書き保存(S)

画面上で変更した定義情報を読み込んだ定義ファイルに上書き保存します。



#### 定義ファイルを名前を付けて保存(A)

画面で設定した定義情報を新規に定義ファイルとして保存します。



#### 入力データファイルを選択(D)

コピー元のデータファイルを選択します。画面上の入力ファイル形式の選択を行ってからファイルを選択してください。

**入力インデックスファイルを選択(I)**

コピー元のインデックスファイルを選択します。画面上の入力ファイル形式として索引順編成ファイルを選択した場合のみ指定可能です。

**入力ファイルを選択解除(C)**

選択した入力ファイル(コピー元ファイル)の選択を解除します。画面上の入力ファイル形式としてテキストファイルを選択した場合のみ指定可能です。

**出力データファイルを選択(D)**

コピー先のファイルを選択します。画面上の出力ファイル形式の選択を行ってからファイルを選択してください。

**出力インデックスファイルを選択(I)**

コピー先のインデックスファイルを選択します。画面上の出力ファイル形式として索引順編成ファイルを選択した場合のみ指定可能です。

**アプリケーションの終了(X)**

本ユーティリティを終了します。

**表示(V)****ツールバー(T)**

ツールバーを表示します。

**ステータスバー(S)**

ステータスバーを表示します。

**オプション(O)****入力オプション(I)**

入力ファイル(コピー元ファイル)に対するオプション情報を設定します。

**出力オプション(O)**

出力ファイル(コピー先ファイル)に対するオプション情報を設定します。

**出力インデックス定義(T)**

出力ファイル(コピー先ファイル)が索引順編成の場合、インデックス定義情報を設定します。

**COBOL仕様(C)**

索引順編成ファイルのインデックスファイル名を以下のように自動的に設定します。

データファイル名(~/xxxxx) → インデックスファイル名(~/xxxxx.idx)

データファイル名(~/xxxxx.yyy) → インデックスファイル名(~/xxxxx.idx)

**インデックス情報を継承(S)**

索引順編成どうしのファイルコピーの場合、入力ファイルと同じ属性のインデックスを出力ファイルに作成します。この場合、出力ファイルのインデックス定義をすることはできません。

**機能設定(F) 複写(C)**

出力ファイルを複写先として、新規作成または既存の同名ファイルを初期化します。

**機能設定(F) 追加(A)**

出力ファイルに対して入力ファイルのデータを追加します。

**相対編成→相対編成(R)**

相対編成ファイルどうしのファイルコピーの場合、レコード番号を引き継いでコピーするか、入力ファイルの削除レコードを無視して順次コピーするかを選択します。

**実行(E)****COPYの実行(E)**

データのコピー処理を開始します。

**COPYの中止(S)**

データのコピー処理を途中で中止します。

## ヘルプ(H)



## IFASCOPYの使い方(H)

オンラインヘルプを表示します。

## バージョン情報(A)

バージョン情報、著作権情報を表示します。

## 一時ファイル機能

ファイルコピーユーティリティは、「アプリケーション実行環境」および「COBOL アプリケーション実行環境」が提供するジョブ管理配下での動作時のみ使用可能な一時ファイルに対応しています。

一時ファイルが通常のファイルと異なる点は以下のとおりです。

- ・ バッチファイル単位に異なるファイル名が生成されます。
- ・ バッチファイル終了時に削除されます。
- ・ 一時ファイルはコマンド実行時に使用され、GUIからは使用することはできませんが、GUIから一時ファイル対応の定義ファイルを作成することができます。画面の「テンポラリ指定」チェックボタンをONにするとコピーの実行はできなくなりますが、定義ファイルの保存を行うことにより、一時ファイル対応の定義ファイルが作成されます。

一時ファイル使用時の注意点は以下のとおりです。

- ・ 一時ファイルについては、これをサポートしているアプリケーション、または、ジョブ管理配下で起動されたバッチプログラムのみで 사용할 ことができます。
- ・ CSV形式ファイルとIFASファイル間のコピーの場合、入出力いずれのファイルも一時ファイルとすることはできません。
- ・ 一時ファイルはクライアント／サーバ形態でのサーバ側ファイルとすることはできません。
- ・ 一時ファイル名の指定は絶対パス名で行ってください。
- ・ 一時ファイル名としてネットワークファイル名を指定することはできません。
- ・ 一時ファイル名は230バイト以内で指定してください。

## 固定長CSV形式ファイルの仕様

固定長CSVファイルはIFAS FILE、IFASPRO RDBが対応しているCSV形式ファイルの一形態で、そのファイルはリターンコード(¥n/¥r¥n)までを1レコードとして扱い、ヘッダを除く各レコードの長さは基本的には等しいものです。MS-DOS形式テキストファイルと似たような書式ですが、RFDファイルを使用して、ユーザレコード定義とおりの属性でIFAS一般ファイルを作成することができます。

## ヘッダ

固定長CSVファイルのヘッダ(項目名レコード)は各項目名を1バイトブランクで区切ったものです。

## 数値属性項目

「符号あり」の数値項目に関しては、符号領域として項目の左端に1バイトの領域を必要とします。正数の場合は0を表示し、負数の場合は-(マイナス)を表示します。入力データにもこの領域は必要となります。

COMP-1とCOMP-2は固定サイズとなり、その桁数は以下ようになります。

COMP-1	5桁固定(よって実際は符号バイトと合わせた6桁で表示)
COMP-2	10桁固定(よって実際は符号バイトと合わせた11桁で表示)

COMP-5はRFDファイルへの定義桁数によって以下の固定サイズになります。

1～4桁	5桁固定(「符号あり」の場合は+1桁、小数の場合にはさらに+1桁)
5～9桁	10桁固定(「符号あり」の場合は+1桁、小数の場合にはさらに+1桁)
10～18桁	19桁固定(「符号あり」の場合は+1桁、小数の場合にはさらに+1桁)



オプション指定によりエラー(終了)/パディングが選択できます。パディングキャラクタは項目の属性によって異なります。

数値属性項目	0をパディング
1バイト文字項目	Null文字をパディング
2バイト文字項目	Null文字を2個パディング

#### レコード長より長いレコードの処理

オプション指定によりエラー(終了)/切り詰めが選択できます。

#### 出力時のレコード中にNull文字を検出した場合

以降の項目属性において、数値属性の場合は0、文字属性の場合は1バイト空白、2バイト文字属性の場合は2バイト空白をパディングします。

## コマンドラインからの実行

ファイルコピーユーティリティは、操作コマンドを記述した定義ファイルをパラメータとしてコンソールコマンドから以下のように実行することができます。

32ビット版ファイルコピーユーティリティ

**IFASCOPY 定義ファイル名 [置換文字列1]・・・[置換文字列9]**

64ビット版ファイルコピーユーティリティ

**IFASCOPY64 定義ファイル名 [置換文字列1]・・・[置換文字列9]**

パラメータは必ずブランクで区切ってください。ファイル名にブランクが含まれる場合はパラメータをダブルクォーテーション(")で囲んでください。

置換文字列1～9については、「7-5 定義ファイルのパラメータ置換機能」を参照してください。

この定義ファイルは画面で定義した結果を「定義ファイルを名前を付けて保存」によって作成することも可能で、エディタ等で加工することもできます。

【コマンドラインの戻り値】(ERRORLEVEL)

0	正常終了しました
1	IFAS FILEのワーニングステータス41290が返りました
2～99	IFAS FILEの正常系ステータスで終了しました
100	IFAS FILEの異常系ステータスで終了しました
255	ユーティリティによるエラーで終了しました

## 定義ファイルの記述方法

コマンドライン実行を行うには、定義ファイルを作成しなくてはなりません。定義ファイルはIFASCOPYおよびIFASCOPY64を実行するために必要なオプションやキー記述を定義したテキストファイルです。このファイルは、GUIから自動作成することもできますし、ユーザがエディタ等を使用して直接作成することもできます。

定義ファイルの中は一回の動作に必要な、入出力ファイル形式・入出力ファイル名・機能コード・オプション・キー指定の設定をもって1つのセクションとし、複数セクション指定することができます。セクションは **[Execute Set]** を先頭に始まります。

定義ファイルの設定コマンドは以下のとおりです。(太字は既定値)

### 入出力ファイル形式

【書式】

InFileForm=入力ファイル形式 Ind/Rel/Text/Seq/Csv のうち、1つを指定(必須)  
 OtFileForm=出力ファイル形式 Ind/Rel/Text/Seq/Csv のうち、1つを指定(必須)  
 “Csv”を除く各形式の頭に[Tmp.]を付けると、その形式のファイルを一時ファイルとして扱います。

## 【指定内容】

Ind	IFAS索引順編成ファイル
Rel	IFAS相対編成ファイル
Text	MS-DOS形式テキストファイル
Seq	IFAS順編成ファイル
Csv	CSV形式ファイル
Tmp.Ind	IFAS索引順編成ファイル(一時ファイル)
Tmp.Rel	IFAS相対編成ファイル(一時ファイル)
Tmp.Text	MS-DOS形式テキストファイル(一時ファイル)
Tmp.Seq	IFAS順編成ファイル(一時ファイル)

## 入出力ファイル名

- ・IFAS索引順編成ファイルの場合

## 【書式】

UtilInFileNameIfDat=入力IFASデータファイルのフルパスファイル名(必須)  
 UtilInFileNameIfIdx=入力IFASインデックスファイルのフルパスファイル名(必須)  
 UtilOtFileNameIfDat=出力IFASデータファイルのフルパスファイル名(必須)  
 UtilOtFileNameIfIdx=出力IFASインデックスファイルのフルパスファイル名(必須)  
 (必ずデータファイル、インデックスファイルともに指定してください)

- ・MS-DOS形式テキスト／IFAS相対編成／IFAS順編成ファイルの場合

## 【書式】

UtilInFileName=入力ファイルのフルパスファイル名(必須)  
 UtilOtFileName=出力ファイルのフルパスファイル名(必須)

## 機能コード

## 【書式】

FuncCode=Copy/Appendのどちらかを選択(必須)

## オプション

## 【書式】

OtShortRec=出力レコードサイズより短いレコードの処置

## 【指定内容】

<b>Blank</b>	空白文字パディングを行う
Null	Null文字パディングを行う
Error	エラーとして処理を中止する

## 【書式】

OtLongRec=出力レコードサイズより長いレコードの処置

## 【指定内容】

<b>Cut</b>	超えた部分を切り捨てる
Next	超えた部分を次レコードとする(テキスト→索引のコピーのみ可)
Error	エラーとして処理を中止する

## 【書式】

InReturnCode=テキストファイル入力時のリターンコードの処置

## 【指定内容】

<b>Code</b>	コードとして扱う
Data	データとして扱う

## 【書式】

OtReturnCode=テキストファイル出力時のリターンコードの処置

## 【指定内容】

<b>Code</b>	コードとして扱う
Data	データとして扱う

## 【書式】

InRecMethod=IFAS索引順編成ファイル入力時の読み込み方法

## 【指定内容】

<b>Number</b>	レコード番号順に読み込む
Upkey?	インデックス番号?をキーに昇順で読み込む

DownKey? インデックス番号?をキーに降順で読み込む

【書式】

RecSize=出力レコードサイズ(既定値:0)

(0を指定した場合)

テキストファイル入力時 : レコードサイズを256バイトにします

索引順編成ファイル入力時 : レコードサイズを入力ファイルと同じにします

【書式】

IndexBlockSize=IFAS索引順編成ファイル出力時のインデックスブロックサイズ(既定値:1024)

【書式】

DupKeyError=IFAS索引順編成ファイルへの追加時に重複キーエラー検出時の処置

【指定内容】

**Abort** GUIの場合はユーザに問い合わせます。  
コマンドラインの場合は発生時に強制終了

**Skip** GUIの場合は問い合わせをスキップします。  
コマンドラインの場合は無視され強制終了します

【書式】

CsvErrorCheck=CSV形式ファイルの中身とRFDファイルの属性が異なっている時の処理

【指定内容】

**Abort** エラーとして処理を中断する

**Skip** 属性変更できる場合は変換し、できない場合はそのレコードをスキップする

【書式】

CsvFirstField=CSV形式ファイルの先頭行の処理

【指定内容】

**Data** 先頭行をデータとして処理する

**Header** 先頭行を項目名行として処理する

【書式】

CsvFieldCutChar=フィールド区切り文字

【指定内容】

**Comma** カンマ

**Semicolon** セミコロン

**Tab** タブ

**Blank** 空白

**Null** なし(固定長CSV形式ファイルとして処理を行う)

**?** 任意の1バイト文字

【書式】

CsvTextCutChar=テキスト区切り文字

【指定内容】

**Double** ダブルクォーテーション

**Single** シングルクォーテーション

**Null** なし

【書式】

CsvPaddingChar=1バイトパディングキャラクタ

【指定内容】

**Null** Null文字

**Blank** 空白

**?** 任意の1バイト文字

【書式】

CsvPaddingNChar=2バイトパディングキャラクタ

【指定内容】

**Null** Null文字

**Blank** 空白(2バイト)

**?** 任意の2バイト文字

【書式】

CsvLongRecord=固定長CSVファイル入力時のレコード長が長かった時の処理

【指定内容】

**Execute** 切り捨てて処理を継続する  
**Error** エラーとして処理を中断する

## 【書式】

CsvShortRecord=固定長CSVファイル入力時のレコード長が短かった時の処理

## 【指定内容】

**Execute** Nullパディングして処理を継続する  
**Error** エラーとして処理を中断する

## キー指定

## 【書式】

Key1=昇順番号,オフセット,サイズ,タイプ番号[,キー重複指定]

Key2=昇順番号,オフセット,サイズ,タイプ番号[,キー重複指定]

:

Keyn=昇順番号,オフセット,サイズ,タイプ番号[,キー重複指定]

左辺値のKey番号は必ず昇順に抜けないように振ってください。抜けている番号があった場合は、その前の番号までしか処理には反映されません。

## 【指定内容】

## 昇順番号

1から順に付けてください。

マルチキーパートのインデックス(合成キー)を作成する場合は同じ番号を付けてください。

昇順番号1はプライマリキー専用です。

## オフセット

レコードの先頭からのオフセットです。(バイト)

## サイズ

キーサイズを指定してください。(バイト)

プライマリレスファイルを作成したい場合は、昇順番号1のキーパートのサイズを0としてください。

例 Key1=1,0,0,0                      プライマリレスを指定  
      Key2=2,12,8,0,Dup              セカンダリキーを指定

## タイプ番号

キータイプの番号です。IFAS FILE関数との対応は以下のようになっています。

キータイプ	IFAS FILE関数指定	タイプ番号
文字	IFCHAR	0
DWORD (4バイト)	IFBIN4	1
WORD (2バイト)	IFBIN2	2
LONG (4バイト)	IFSBIN4	3
INT (2バイト)	IFSBIN2	4
FLOAT (4バイト)	IFFLOAT	5
DOUBLE (8バイト)	IFDOUBLE	6
パック10進数 (符号なし)	IFPDNS	7
パック10進数 (右端符号)	IFPDT	8
パック10進数 (左端符号)	IFPDL	9
アンパック10進数 (符号なし)	IFUDNS	10
アンパック10進数 (右重ね符号)	IFUDT	11
アンパック10進数 (左重ね符号)	IFUDL	12
アンパック10進数 (右分離符号)	IFUDTS	13
アンパック10進数 (左分離符号)	IFUDLS	14
符号なしBINARY (2バイト)	IFUBE2	15
符号なしBINARY (4バイト)	IFUBE4	16
符号なしBINARY (8バイト)	IFUBE8	17
符号ありBINARY (2バイト)	IFSBE2	18
符号ありBINARY (4バイト)	IFSBE4	19
符号ありBINARY (8バイト)	IFSBE8	20

符号なしCOMP-5 (2バイト)	IFUCP2	21
符号なしCOMP-5 (4バイト)	IFUCP4	22
符号なしCOMP-5 (8バイト)	IFUCP8	23
符号ありCOMP-5 (2バイト)	IFSCP2	24
符号ありCOMP-5 (4バイト)	IFSCP4	25
符号ありCOMP-5 (8バイト)	IFSCP8	26
2バイト文字	IFNCHAR	27

#### キー重複指定

重複するキーが存在する場合、Dup を指定してください。

(ただし昇順番号1のインデックスにDup指定はできません)

マルチキーパート(合成キー)の場合は、昇順番号の等しいキーのいずれか(複数可)についていれば重複ありとして作成します。

#### 一括処理

1つの定義ファイルに複数のセクションを記述することで一括処理をすることができます。

##### 【書式】

[Execute Set]

⋮

[Execute Set 2]

⋮

[Execute Set 3]

⋮

[Execute Set n]

セクションを2つ以上記述する場合、2つ目以降からは番号をふってください。同一番号のセクションが存在する場合は先に書かれている方が実行され、後のものは実行されません。

## エラーステータス

コマンドラインで実行中にエラーを検出した場合は、カレントディレクトリに IFASCOPY.ERR というファイルを作成し、エラーコードを書き出します。エラーを検出した処理に関しては、それ以前の正常に終了したセクションも含めて、使用したファイルのデータの保証はいたしません。また、コマンドラインの戻り値としてもステータスを返しますので、バッチファイルの中で認識することも可能です。

#### ステータス一覧

エラーコード	意味
41110	入力ファイルのオープンに失敗しました。
41111	入力テキストファイルの読み込みに失敗しました。
41112	出力レコードサイズより長いレコードを検出しました。
41113	出力レコードサイズより短いレコードを検出しました。
41114	出力テキストファイルへの書き出しに失敗しました。
41115	予期せぬエラーが発生しました。
41116	アプリケーションの実行に必要な領域がありません。
41117	定義ファイルの内容が不正です。
41118	定義ファイルが存在しません。
41119	定義ファイルのオープンに失敗しました。
41120	領域確保に失敗しました。
41121	定義ファイルにファイル形式指定がありません。
41122	定義ファイルにファイル名指定がありません。
41123	定義ファイルに機能コードがありません。
41125	定義ファイルで指定ファイル形式と入力ファイル形式が一致しません。

41127	定義ファイルのキー指定が不正です。
41128	定義ファイルの保存に失敗しました。
41131	サポートされていない形式の組み合わせでコピーを実行しようとしました。
41132	RFDファイルが存在しない、または不正です。
41133	一時ファイル機能は使用できません。以下の環境下で使用してください。 IFASCOPY.EXE : アプリケーション実行環境 IFASCOPY64.EXE : COBOL アプリケーション実行環境
41141	読み込もうとしたファイルは定義ファイルではありません。
41142	テンポラリファイルの作成に失敗しました。
41203	IFAS FILE関数エラーです。 このステータスとともに出力されるIFAS FILEステータスを参照してください。
41204	IFAS FILE関数エラーです。 このステータスとともに出力されるIFAS FILEステータスを参照してください。
41205	指定されたCSV形式ファイルがオープンできません。
41206	メモリ不足です。
41207	RFDの定義が不正です。
41208	フィールド区切り文字とテキスト区切り文字が同じです。
41209	CSV形式が不正です。
41210	入力と出力のカラム定義がアンマッチです。
41211	RFDファイルの定義内に未対応のデータタイプを検出しました。
41212	IFAS一般ファイルとRFDファイルの定義がアンマッチです。
41213	IFAS索引順編成ファイルに索引がありません。
41214	RFDファイルが存在しません。
41215	IFAS索引順編成ファイルに索引がありません。
41216	固定長CSV形式ファイルでレコード長より長い入力レコードを検出しました
41217	固定長CSV形式ファイルでレコード長より短い入力レコードを検出しました。
41280	CSV形式ファイルが存在しない。または、データの出力に失敗しました。
41281	指定されたCSV形式ファイルがオープンできません。
41282	予期せぬエラーが発生しました。
41290	定義のアンマッチです。ワーニングとして処理を継続しました。

これとは別に、IFASPROのエラーも返します。詳細は「付録A ステータスコード一覧」を参照してください。

## 定義ファイル作成例

### 1. IFAS索引順編成ファイルからMS-DOS形式テキストファイルにコピーする場合

[Execute Set]	
InFileForm=Ind	入力は索引順編成
OtFileForm=Text	出力はテキスト
UtilInFileNameIfDat=SUMPLE1.DAT	
UtilInFileNameIfIdx=SUMPLE1.IDX	
UtilOtFileName=SUMPLE1.TXT	
FuncCode=Copy	機能コードはCopy
OtShortRec=NULL	出力レコードより短いレコードはNullパディング
OtLongRec=Cut	出力レコードより長いレコードは超えた分を切り捨て
OtReturnCode=Code	出力時のリターンコードはコードとして処理
InRecMethod=Downkey3	読み込みはインデックス番号3のキーを降順に
RecSize=0	レコードサイズは入力ファイルと等しくする

### 2. MS-DOS形式テキストファイルからIFAS索引順編成ファイルにコピーする場合

[Execute Set]	
InFileForm=Text	入力はテキスト
OtFileForm=Ind	出力は索引順編成

UtilInFileName=SUMPLE21.TXT,SUMPLE22.TXT	テキストを複数指定
UtilOtFileNameIfDat=SUMPLE2.DAT	
UtilOtFileNameIfIdx=SUMPLE2.IDX	
FuncCode=Copy	機能コードはCopy
OtShortRec=NULL	出力レコードより短いレコードはNullパディング
OtLongRec=Error	出力レコードより長いレコードを検出次第エラーで中止
InReturnCode=Data	入力時のリターンコードはデータとして扱う
OtReturnCode=Code	出力時のリターンコードはコードとして扱う
RecSize=100	レコードサイズは100バイト
IndexBlockSize=1024	インデックスブロックサイズは1024バイト
Key1=1,0,10,0	オフセット(0),サイズ(10),タイプ(文字)
Key2=1,4,12,0	オフセット(4),サイズ(12),タイプ(文字)
	Key1とKey2でマルチキーパートのプライマリキー
Key3=2,0,12,1,Dup	オフセット(0),サイズ(12),タイプ(DWORD),重複あり

### 3. IFAS索引順編成ファイルからIFAS索引順編成ファイルにコピーする場合

[Execute Set]

InFileForm=Ind	入力は索引順編成
OtFileForm=Ind	出力は索引順編成
UtilInFileNameIfDat=INPUT3.DAT	
UtilInFileNameIfIdx=INPUT3.IDX	
UtilOtFileNameIfDat=OUTPUT3.DAT	
UtilOtFileNameIfIdx=OUTPUT3.IDX	
FuncCode=Copy	機能コードはCopy
OtShortRec=NULL	出力レコードより短いレコードはNullパディング
OtLongRec=Cut	出力レコードより長いレコードは超えた分を切り捨て
OtReturnCode=Code	出力時のリターンコードはコードとして扱う
InRecMethod=Number	読み込みはレコード番号順
RecSize=0	レコードサイズは入力ファイルに合わせる
IndexBlockSize=1024	インデックスブロックサイズは1024バイト
Key1=1,0,0,0	プライマリインデックスなし
Key2=2,12,4,0,Dup	オフセット(12),サイズ(4),タイプ(文字)
Key3=2,22,4,0,Dup	オフセット(22),サイズ(4),タイプ(文字)
	Key2とKey3でマルチキーパート(重複あり)指定
Key4=3,2,32,14,Dup	オフセット(2),サイズ(32),タイプ(アンパック10進数:左分離符号)
Key5=3,40,4,0,Dup	オフセット(40),サイズ(4),タイプ(文字)
	Key4とKey5でマルチキーパート(重複あり)指定

※IFAS索引順編成ファイルからIFAS索引順編成ファイルにコピーする場合、インデックスの継承オプションはありません。入力ファイルのすべてのインデックスを指定してください。

### 4. IFAS索引順編成ファイルからIFAS相対編成ファイルにコピーする場合

[Execute Set]

InFileForm=Ind	入力は索引順編成
OtFileForm=Rel	出力は相対編成
UtilInFileNameIfDat=INPUT4.DAT	
UtilInFileNameIfIdx=INPUT4.IDX	
UtilOtFileName=OUTPUT4.REL	
FuncCode=Copy	機能コードはCopy
OtShortRec=NULL	出力レコードより短いレコードはNullパディング
OtLongRec=Cut	出力レコードより長いレコードは超えた分を切り捨て
OtReturnCode=Code	出力時のリターンコードはコードとして扱う
InRecMethod=Numbe	読み込みはレコード番号順
RecSize=1000	レコードサイズは1000バイト

## 5. MS-DOS形式テキストファイルからIFAS索引順編成ファイルに追加する場合

[Execute Set]	
InFileForm=Text	入力テキスト
OtFileForm=Ind	出力は索引順編成
UtilInFileName=INPUT51.TXT,INPUT52.TXT テキストファイルを複数指定	
UtilOtFileNameIfDat=OUTPUT2.DAT	
UtilOtFileNameIfIdx=OUTPUT2.IDX	
FuncCode=Append	機能コードはAppend
OtShortRec=NULL	出力レコードより短いレコードはNullパディング
OtLongRec=Error	出力レコードより長いレコードを検出次第エラーで中止
InReturnCode=Data	入力時のリターンコードはデータとして扱う
OtReturnCode=Code	出力時のリターンコードはデータとして扱う
RecSize=100	レコードサイズは100バイト
IndexBlockSize=1024	インデックスブロックサイズは1024バイト
DupKeyError=Abort	重複キーの検出時は問い合わせ、もしくは強制終了

## 6. CSV形式ファイルからIFAS索引順編成ファイルにコピーする場合

[Execute Set]	
InFileForm=Csv	入力CSV形式
OtFileForm=Ind	出力は索引順編成
UtilInFileName=INPUT6.CSV	
UtilOtFileNameIfDat=OUTPUT6.DAT	
UtilOtFileNameIfIdx=OUTPUT6.IDX	
FuncCode=Copy	機能コードはコピー
CsvErrorCheck=Skip	不正レコードは強制またはスキップ
CsvFirstField=Data	先頭行をデータとする
CsvFieldCutChar=Comma	フィールド区切り文字はカンマ
CsvTextCutChar=Double	テキスト区切り文字はダブルクォーテーション
CsvPaddingChar=NULL	1バイトパディングキャラクタはNull文字
CsvPaddingNChar=NULL	2バイトパディングキャラクタはNull文字

## 7. 固定長CSV形式ファイルからIFAS相対編成ファイルにコピーする場合

[Execute Set]	
InFileForm=Csv	入力CSV形式
OtFileForm=Rel	出力は相対編成
UtilInFileName=INPUT7.CSV	
UtilOtFileNameIfDat=OUTPUT7.Rel	
FuncCode=Copy	機能コードはコピー
CsvErrorCheck=Skip	不正レコードは強制またはスキップ
CsvFirstField=Data	先頭行をデータとする
CsvFieldCutChar=NULL	固定長CSV指定
CsvLongRecord=Execute	定義より長いレコードがあれば切り捨てて処理
CsvShortRecord=Error	定義より短いレコードがあれば処理を中断

## 7-1-3 インデックスメンテナンスユーティリティ

インデックスメンテナンスユーティリティは、作成済みのIFAS索引順編成ファイルに対して、新たにインデックスを追加したり、既存のインデックスを削除したり、インデックスの再編成を行うためのインデックス保守用ユーティリティです。また、MIXファイル互換機能を使用するためのインデックス更新属性設定機能も提供します。



## 機能概要

インデックスメンテナンスユーティリティはIFAS索引順編成ファイルに対して、

1. インデックスの追加  
索引順編成ファイルに新たにサブインデックスを追加作成します。
2. インデックスの削除  
索引順編成ファイルから既存のインデックスを削除します。
3. インデックスの同時更新属性変更  
作成済みのインデックスの同時更新属性として、「同時更新」「非更新」「実行時生成」の3種類から選択して設定します。
4. 不整合インデックスの再編成  
データファイルと不整合状態になった「非更新」属性のインデックスを再編成してデータファイルと整合がとれる状態にします。

以上4機能の処理を行うためのユーティリティです。画面構成は機能毎に4種類のサブ画面があり、それぞれ複数同時に開いて処理定義を設定することが可能になっています。

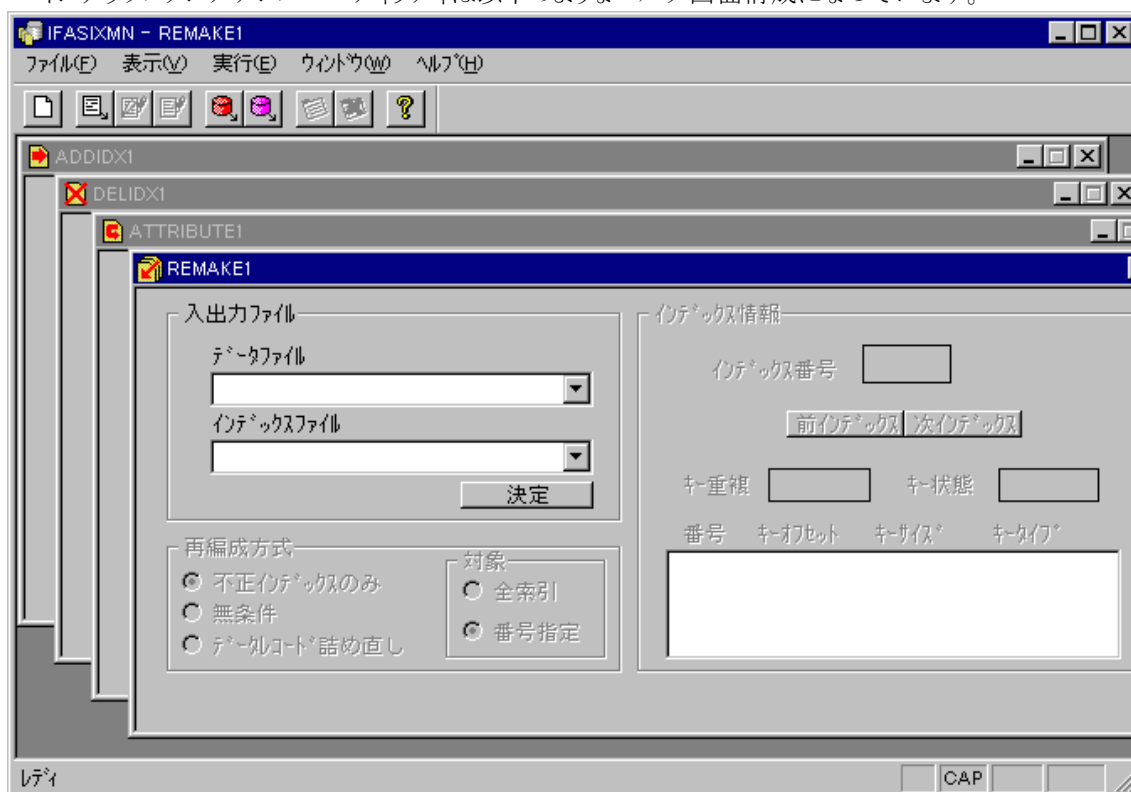
インデックスの追加によって、1つの索引順編成ファイルに対し最大255個までインデックスを作成することが可能です。

インデックスの追加によって作成したインデックスのインデックス番号は自動的に割り当てられます。インデックスの削除によって欠番になったインデックス番号は、インデックスの追加時に再利用されます。

サブ画面単位で設定した定義情報を定義ファイルとして保存／再利用することができます。この定義ファイルはエディタ等で作成することも可能で、さらにコンソールコマンドから以下のようにして実行することもできます。

### IFASIXMN 定義ファイル名

インデックスメンテナンスユーティリティは以下のようなマルチ画面構成になっています。



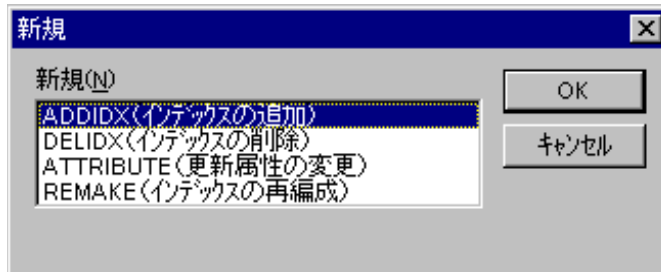
## メニューバー／ツールバーの説明

### ファイル(F)



#### 新規ウィンドウ作成(N)

新規にサブ画面を開きます。以下のダイアログボックスから目的の機能を選択します。



#### 定義ファイルの読み込み(O)

画面上で設定した情報は、定義ファイルとして保存することができます。この定義ファイルを読み込むことで以前設定した定義情報を再定義する必要はありません。



#### 定義ファイルの上書き保存(S)

画面上で変更した定義情報を読み込んだ定義ファイルに上書き保存します。



#### 定義ファイルを名前を付けて保存(A)

画面で設定した定義情報を新規に定義ファイルとして保存します。



#### データファイルを選択(D)

処理を行う索引順編成ファイルのデータファイルを選択します。



#### インデックスファイルを選択(I)

処理を行う索引順編成ファイルのインデックスファイルを選択します。

### アプリケーションの終了(X)

本ユーティリティを終了します。

### 表示(V)

#### ツールバー(T)

ツールバーを表示します。

#### ステータスバー(S)

ステータスバーを表示します。

### 実行(X)



#### 実行(E)

処理を実行します。



#### 実行の中止(S)

処理を途中で中止します。

### ウィンドウ(W)

#### ウィンドウを閉じる(C)

表示中のサブ画面を閉じます。

#### 重ねて表示(L)

表示中の複数のサブ画面を整列させます。

#### アイコンの整列(A)

アイコン化されたサブ画面を整列させます。

### ヘルプ(H)



#### IFASIXMNの使い方(H)

オンラインヘルプを表示します。

#### バージョン情報(A)

バージョン情報、著作権情報を表示します。

## インデックスの追加: ADDIDX画面

入力ファイル

データファイル  
A:\¥IFASKIT¥test¥Work. dat

インデックスファイル  
A:\¥IFASKIT¥test¥Work. IDX

決定

キーパート設定

オフセット 24    サイズ 2    タイプ IFCHAR

追加

追加インデックス情報

追加カウント 1    ☐ キー重複有り

前インデックス    次インデックス

番号	キーオフセット	キーサイズ	キータイプ
1	0	22	IFCHAR
2	24	2	IFCHAR

挿入    削除

インデックス削除

### [入力ファイル]

データファイル、インデックスファイルを選択して「決定」ボタンをクリックします。指定したファイルが正しいIFAS索引順編成ファイルであることが確認されると設定が可能になります。

### [キーパート設定]

キーパートの属性として、レコード内オフセット、キーパートサイズ、キータイプを指定します。

### [追加インデックス情報]

本枠内の情報が追加したいインデックス1つ分の定義となります。

「追加」ボタンをクリックすると[キーパート設定]で定義したキーパート情報がリストボックスセットされます。マルチキーパートのインデックスを定義する場合は[キーパート設定]→「追加」を繰り返します。この場合「追加」ボタンはキーパートリストの最後に新しいキーパートを追加します。キーパートリストの途中で新しいキーパートを挿入したい場合は、直前に挿入したいキーパートをクリックして反転させてから「挿入」ボタンをクリックします。1つのインデックスには最大16個までのキーパート設定が可能です。

リストボックス内に誤って追加したキーパートを削除したい場合は、削除するキーパートを反転させて「削除」ボタンをクリックします。

定義したインデックス情報をすべてクリアしたい場合は「インデックス削除」ボタンをクリックします。

インデックス定義が完了して、インデックス追加を実行したい場合はメニューバーまたはツールバーボタンで処理を実行します。複数のインデックス定義をまとめて行いたい場合は、「次インデックス」ボタンをクリックして2つめ以降のインデックスの定義を同様の方法で行います。

n個目の追加インデックスの定義をした後で「前インデックス」ボタンをクリックすると、n-1個目のインデックス定義画面に戻ることができます。

## インデックスの削除: DELIDX画面

**入力ファイル**

データファイル  
A:\¥IFASKIT¥test¥Work. dat

インデックスファイル  
A:\¥IFASKIT¥test¥Work. IDX

決定

**インデックス情報**

インデックス番号 4

前インデックス 次インデックス

キー重複 あり キー状態 整合

番号	キーオフセット	キーサイズ	キータイプ
1	32	22	IFCHAR

### [入力ファイル]

データファイル、インデックスファイルを選択して「決定」ボタンをクリックします。指定したファイルが正しいIFAS索引順編成ファイルであることが確認されると設定が可能になります。

### [インデックス情報]

「次インデックス」ボタン、「前インデックスボタン」をクリックして削除したいインデックス情報を表示させます。

メニューバーまたはツールバーボタンで処理を実行すると、情報が表示されているインデックスはインデックスファイルから削除されます。

## 更新属性の変更: ATTRIBUTE画面

**入力ファイル**

データファイル  
A:\¥IFASKIT¥test¥Work. dat

インデックスファイル  
A:\¥IFASKIT¥test¥Work. IDX

決定

**インデックス情報**

インデックス番号 2

前インデックス 次インデックス

キー重複 あり キー状態 整合

番号	キーオフセット	キーサイズ	キータイプ
1	24	2	IFCHAR

**属性変更**

属性状態 更新

更新 更新 非更新 実行時生成

### [入力ファイル]

データファイル、インデックスファイルを選択して「決定」ボタンをクリックします。指定したファイルが正しいIFAS索引順編成ファイルであることが確認されると設定が可能になります。

### [インデックス情報]

本枠内の情報が追加したいインデックス1つ分の定義となります。

「次インデックス」ボタン、「前インデックスボタン」をクリックして更新属性を変更したいインデックス情報を表示させます。

### [属性変更]

更新属性を、表示されている現在の状態から「更新」「非更新」「実行時生成」のいずれかに変更します。

メニューバーまたはツールバーボタンで処理を実行すると、情報が表示されているインデックスの更新属性は変更されます。

## インデックスの再編成:REMAKE画面

The REMAKE1 window is divided into several sections:

- 入出力ファイル (Input/Output Files):**
  - データファイル (Data File): A:\¥IFASKIT¥test¥Work. dat
  - インデックスファイル (Index File): A:\¥IFASKIT¥test¥Work. IDX
  - 決定 (Decide) button
- 再編成方式 (Reorganization Method):**
  - 不正インデックスのみ (Only Invalid Indexes): Selected
  - 無条件 (Unconditionally)
  - データファイルの詰め直し (Recompact Data File)
- 対象 (Target):**
  - 全索引 (All Indexes)
  - 番号指定 (Specify Number): Selected
- インデックス情報 (Index Information):**
  - インデックス番号 (Index Number): 2
  - 前インデックス (Previous Index) / 次インデックス (Next Index) buttons
  - キー重複 (Key Duplication): あり (Yes)
  - キー状態 (Key Status): 整合 (Consistent)
  - Table with 4 columns: 番号 (Number), キーオフセット (Key Offset), キーサイズ (Key Size), キータイプ (Key Type)
 

番号	キーオフセット	キーサイズ	キータイプ
1	24	2	IFCHAR

### [入力ファイル]

データファイル、インデックスファイルを選択して「決定」ボタンをクリックします。指定したファイルが正しいIFAS索引順編成ファイルであることが確認されると設定が可能になります。

### [再編成方式]

再編成する方式を以下の中から選択します。

- ・不正インデックスのみ  
不整合状態のインデックスのみ再編成を行います。
- ・無条件  
不整合状態でないインデックスも皆、再編成を行います。
- ・データファイルの詰め直し  
データファイルの削除レコード領域を無くしてレコードを並べ直した上で、すべてのインデックスを再編成します。

### [対象]

- ・全インデックス  
すべてのインデックスを再編成します。
- ・指定インデックス  
[インデックス情報]で表示しているインデックスを再編成します。

### [インデックス情報]

「次インデックス」ボタン、「前インデックスボタン」をクリックして再編成したいインデックス情報を表示させます。  
メニューバーまたはツールバーボタンで処理を実行すると、インデックス再編成を行います。

## コマンドラインからの実行

インデックスメンテナンスユーティリティは、操作コマンドを記述した定義ファイルをパラメータとしてコンソールコマンドから以下のように実行することができます。

### IFASIXMN 定義ファイル名 [置換文字列1]・・・[置換文字列9]

パラメータは必ずブランクで区切ってください。ファイル名にブランクが含まれる場合はパラメータをダブルクォーテーション(")で囲んでください。

置換文字列1～9については、「7-5 定義ファイルのパラメータ置換機能」を参照してください。

この定義ファイルは画面で定義した結果を「定義ファイルを名前を付けて保存」によって作成することも可能で、エディタ等で加工することもできます。

### 【コマンドラインの戻り値】(ERRORLEVEL)

0	正常終了しました
1～99	IFAS FILEの正常系ステータスで終了しました
100	IFAS FILEの異常系ステータスで終了しました
255	ユーティリティによるエラーで終了しました

## 定義ファイルの記述方法

コマンドライン実行を行うには、定義ファイルを作成しなくてはなりません。定義ファイルはIFASIXMNを実行するために必要なオプションやキー記述を定義したテキストファイルです。このファイルは、GUIから自動作成することもできますし、ユーザがエディタ等を使用して直接作成することもできます。定義ファイルの中は一回の動作に必要な、入力ファイル形式・入力ファイル名・機能コード・オプション・キー指定の設定をもって1つのセクションとし、複数セクション指定することができます。セクションは **[Execute Set]** を先頭に始まります。定義ファイルの設定コマンドは以下のとおりです。

### 入力ファイル形式

#### 【書式】

InFileForm=Ind      索引順編成を指定(必須)

### 入力ファイル名

#### 【書式】

UtilInFileNameIfDat=IFASデータファイルのフルパスファイル名(必須)

UtilInFileNameIfIdx=IFASインデックスファイルのフルパスファイル名(必須)

### 機能コード

#### 【書式】

FuncCode=実行可能機能 AddIndex/DelIndex/Remake/Attributeから選択(必須)

#### 【指定内容】

AddIndex	インデックス追加機能
DelIndex	インデックス削除機能
Remake	インデックス再編成機能
Attribute	インデックス属性変更機能

### オプション(インデックス再編成のみ)

#### 【書式】

ReorganizeMethod=再編成方式 IFCHK/IFNCHK/IFGABG のうち、1つを記述

#### 【指定内容】

IFCHK	インデックスの正当性をチェックして不正なインデックスであれば再編成します。
IFNCHK	無条件にインデックスの再編成を行います。
IFGABG	データファイルの削除レコードを無くし、全インデックスを再編成します。

#### 【書式】

ReorganizeMethodNumber=インデックス番号(0を指定した場合は全インデックス対象)

\* 前のオプションで IFGABG を指定すると、本パラメータは無効になります。

### キー指定

キー指定の方法は機能コードにより異なります。

#### (1)インデックスの追加

#### 【書式】

Key1=昇順番号,オフセット,サイズ,タイプ番号[,キー重複指定]

Key2=昇順番号,オフセット,サイズ,タイプ番号[,キー重複指定]

:

Keyn=昇順番号,オフセット,サイズ,タイプ番号[,キー重複指定]

#### 【指定内容】

: 昇順番号

1から順に付けてください。

マルチキーパートのインデックス(合成キー)を作成する場合は同じ番号を付けてください。

: オフセット

レコードの先頭からのオフセットです。(バイト)

: サイズ

キーサイズを指定してください。(バイト)

:タイプ番号

キータイプの番号です。IFAS FILE関数との対応は以下のようになっています。

キータイプ	IFAS FILE関数指定	タイプ番号
文字	IFCHAR	0
DWORD (4バイト)	IFBIN4	1
WORD (2バイト)	IFBIN2	2
LONG (4バイト)	IFSBIN4	3
INT (2バイト)	IFSBIN2	4
FLOAT (4バイト)	IFFLOAT	5
DOUBLE (8バイト)	IFDOUBLE	6
パック10進数 (符号なし)	IFPDNS	7
パック10進数 (右端符号)	IFPDT	8
パック10進数 (左端符号)	IFPDL	9
アンパック10進数 (符号なし)	IFUDNS	10
アンパック10進数 (右重ね符号)	IFUDT	11
アンパック10進数 (左重ね符号)	IFUDL	12
アンパック10進数 (右分離符号)	IFUDTS	13
アンパック10進数 (左分離符号)	IFUDLS	14
符号なしBINARY (2バイト)	IFUBE2	15
符号なしBINARY (4バイト)	IFUBE4	16
符号なしBINARY (8バイト)	IFUBE8	17
符号ありBINARY (2バイト)	IFSBE2	18
符号ありBINARY (4バイト)	IFSBE4	19
符号ありBINARY (8バイト)	IFSBE8	20
符号なしCOMP-5 (2バイト)	IFUCP2	21
符号なしCOMP-5 (4バイト)	IFUCP4	22
符号なしCOMP-5 (8バイト)	IFUCP8	23
符号ありCOMP-5 (2バイト)	IFSCP2	24
符号ありCOMP-5 (4バイト)	IFSCP4	25
符号ありCOMP-5 (8バイト)	IFSCP8	26
2バイト文字	IFNCHAR	27

:キー重複指定

重複するキーが存在する場合、Dup を指定してください。

マルチキーパート(合成キー)の場合は、昇順番号の等しいキーのいずれか(複数可)についていれば重複ありとして作成します。

(2)インデックスの削除

【書式】

Key1=インデックス番号

Key2=インデックス番号

:

Keyn=インデックス番号

(3)インデックスの属性変更

【書式】

Key1=インデックス番号,変更の指定Change/NoChange/DynamicMakeのいずれか

Key2=インデックス番号,変更の指定Change/NoChange/DynamicMakeのいずれか

:

Keyn=インデックス番号,変更の指定Change/NoChange/DynamicMakeのいずれか

【指定内容】

Change	指定インデックスを更新属性にします
NoChange	指定インデックスを非更新属性にします
DynamicMake	指定インデックスを実行時生成属性にします

### 一括処理

1つの定義ファイルに複数のセクションを記述することで一括処理をすることができます。

#### 【書式】

```
[Execute Set]
:
:
:
[Execute Set 2]
:
:
[Execute Set 3]
:
:
[Execute Set n]
```

セクションを2つ以上記述する場合、2つ目以降からは番号をふってください。同一番号のセクションが存在する場合は先に書かれている方が実行され、後のものは実行されません。

## エラーステータス

コマンドラインで実行中にエラーを検出した場合は、カレントディレクトリに IFASIXMN.ERR というファイルを作成し、エラーコードを書き出します。エラーを検出した処理に関しては、それ以前の正常に終了したセクションも含めて、使用したファイルのデータの保証はいたしません。また、コマンドラインの戻り値としてもステータスを返しますので、バッチファイルの中で認識することも可能です。

#### ステータス一覧

エラーコード	意味
41110	入力ファイルのオープンに失敗しました。
41111	入力テキストファイルの読み込みに失敗しました。
41114	出力テキストファイルへの書き出しに失敗しました。
41115	予期せぬエラーが発生しました。
41116	アプリケーションの実行に必要な領域がありません。
41117	定義ファイルの内容が不正です。
41118	定義ファイルが存在しません。
41119	定義ファイルのオープンに失敗しました。
41120	領域確保に失敗しました。
41121	定義ファイルにファイル形式指定がありません。
41122	定義ファイルにファイル名指定がありません。
41123	定義ファイルに機能コードがありません。
41125	定義ファイルで指定ファイル形式と入力ファイル形式が一致しません。
41127	定義ファイルのキー指定が不正です。
41128	定義ファイルの保存に失敗しました。
41141	読み込もうとしたファイルは定義ファイルではありません。
41142	テンポラリファイルの作成に失敗しました。

これとは別に、IFASPROのエラーも返します。詳細は「付録A ステータスコード一覧」を参照してください。



## 定義ファイル作成例

### (1) インデックスの追加

#### 例1

[Execute Set]	
InFileForm=Ind	
UtilInFileNameIfDat=Sumple.dat	データファイルはSumple.dat
UtilInFileNameIfIdx=Sumple.idx	インデックスファイルはSumple.idx
FuncCode=AddIndex	機能コードはインデックス追加
Key1=1,8,16,0	オフセット(8),サイズ(16),タイプ(文字)
Key2=1,32,8,6	オフセット(32),サイズ(8),タイプ(DOUBLE)
	Key1とKey2でマルチキーパート
Key3=2,40,4,0,Dup	オフセット(40),サイズ(4),タイプ(文字),重複あり

#### 例2 (誤った例)

[Execute Set]	
InFileForm=Ind	
UtilInFileNameIfDat=Sumple.dat	データファイルはSumple.dat
UtilInFileNameIfIdx=Sumple.idx	インデックスファイルはSumple.idx
FuncCode=AddIndex	機能コードはインデックス追加
Key1=1,8,16,0	オフセット(8),サイズ(16),タイプ(文字)
Key3=1,32,8,6	オフセット(32),サイズ(8),タイプ(DOUBLE)
	Key1とKey2でマルチキーパート
Key4=5,40,4,0,Dup	オフセット(40),サイズ(4),タイプ(文字),重複あり

- Key番号が昇順ではない (Key2が抜けている)
- 昇順番号が昇順ではない (2~4がない)

### (2) インデックスの削除

#### 例1

[Execute Set]	
InFileForm=Ind	
UtilInFileNameIfDat=Sumple.dat	データファイルはSumple.dat
UtilInFileNameIfIdx=Sumple.idx	インデックスファイルはSumple.idx
FuncCode=DelIndex	機能コードはインデックス削除
Key1=2	インデックス番号2のインデックスを削除
Key2=5	インデックス番号5のインデックスを削除

#### 例2 (誤った例)

[Execute Set]	
InFileForm=Ind	
UtilInFileNameIfDat=Sumple.dat	データファイルはSumple.dat
UtilInFileNameIfIdx=Sumple.idx	インデックスファイルはSumple.idx
FuncCode=DelIndex	機能コードはインデックス削除
Key1=2	インデックス番号2のインデックスを削除
Key3=5	インデックス番号5のインデックスを削除

- Key番号が昇順ではない (Key2が抜けている)

### (3) インデックスの再編成

[Execute Set]	
InFileForm=Ind	
UtilInFileNameIfDat=Sumple.dat	データファイルはSumple.dat
UtilInFileNameIfIdx=Sumple.idx	インデックスファイルはSumple.idx
FuncCode=Remake	機能コードはインデックス再編成
ReorganizeMethod=IFCHK	不正なキーならば再編成
ReorganizeMethodNumber=4	インデックス番号4のインデックスを対象とする

## (4)インデックスの属性変更

## 例1

[Execute Set]

InFileForm=Ind

UtilInFileNameIfDat=Sumple.dat

UtilInFileNameIfIdx=Sumple.idx

FuncCode=Attribute

Key1=2,Change

Key2=4,NoChange

Key3=6,DynamicMake

データファイルはSumple.dat

インデックスファイルはSumple.idx

機能コードはインデックス属性変更

インデックス番号2のインデックスを更新に変更

インデックス番号4のインデックスを非更新に変更

インデックス番号6のインデックスを実行時生成に変更

## 例2(誤った例)

[Execute Set]

InFileForm=Ind

UtilInFileNameIfDat=Sumple.dat

UtilInFileNameIfIdx=Sumple.idx

FuncCode=Attribute

Key1=1,Change

Key2=4,NoChange

データファイルはSumple.dat

インデックスファイルはSumple.idx

機能コードはインデックス属性変更

インデックス番号1のインデックスを更新に変更

インデックス番号4のインデックスを非更新に変更

・ インデックス番号1はプライマリキーのため属性変更はできない。

## 7-1-4 RFD編集ツール

RFD編集ツールは、IFAS一般ファイルに対するRFDファイルの作成／編集処理を、簡易的な操作で行うためのユーティリティです。

### 機能概要

RFD編集ツールは、以下の機能を提供します。

#### RFDファイル作成機能

指定とおりのIFAS一般ファイルに対するRFDファイルを作成します。

#### RFDファイル編集機能

既存のRFDファイルの定義情報を読み込み、編集して上書き保存または新規のRFDファイルを作成することができます。

### 表示画面の説明

RFD編集ツールを起動すると以下のような画面が開きます。

無題 - IFRFDUTL

ファイル(F) 表示(V) ヘルプ(H)

IFAS一般ファイル情報

ファイル種別(K) IFAS一般ファイル名(F)

順編成

参照...

定義済み列情報

列名 データ型 桁数

列情報定義

列名(N)

データ型(C)

COMP-1

桁数(P) 小数点以下(S)

0 0

追加(A)

削除(D)

変更(M)

↑ ↓

定義済みインデックス情報

重複キー キー情報

定義(D)

削除(R)

レディ

NUM

### ・IFAS一般ファイル情報

IFAS一般ファイル情報としてファイル種別を、順編成、相対編成、索引順編成から選択します。  
また、IFAS一般ファイルのフルパス名を指定します。IFAS一般ファイルは、[参照]ボタンにより選択もできます。

### ・列情報定義

IFAS一般ファイルを構成する列の情報を入力します。  
列名、データ型、総桁数、小数点以下桁数を入力し、[追加]ボタンをクリックすることにより、定義済み列情報へ追加します。

### ・定義済み列情報

定義された列情報の一覧を表示します。  
[削除]ボタンにて、定義済み列情報から選択された列情報を削除します。  
定義済み列情報を変更したい場合、変更したい列情報をダブルクリックします。  
定義済みの列情報が列情報定義に表示されますので、列情報を変更し、[変更]ボタンをクリックすることにより、列情報が変更されます。  
また、矢印ボタンにより、選択された列情報の列順序のみを変更することもできます。

### ・定義済みインデックス情報

IFAS一般ファイル情報のファイル種別が索引順編成の時、定義されたインデックス情報の一覧を表示します。定義された重複キーの有無と定義順に列名が羅列されたキー情報が表示されます。  
[定義]ボタンにてインデックスを定義するダイアログが表示されます。重複キーの有無とキー情報(列の選択)を指定してください。

インデックス - 新規作成

☐ 重複キー有り(E)

列一覧(S)

列名	データ型	桁数
COLUMNS1	COMP-1	
COLUMNS2	文字	5

追加(A)      削除(D)

選択済み列一覧(P)

列名	データ型	桁数
COLUMNS1	COMP-1	

OK      キャンセル      ヘルプ(H)

[削除]ボタンにてインデックスを削除します。  
定義済みインデックス情報を選択してダブルクリックすることにより、重複キーの有無を変更することができます。重複キー情報の表示を確認してください。  
キー情報を変更する場合は、削除した後に再定義してください。

## メニューバー／ツールバーの説明

### ファイル(F)



#### 新規作成(N)

新規にRFDファイルを作成するため、画面情報をクリアします。

**開く(O)**

既存のRFDファイルの情報を読み込み、画面上に定義情報を反映します。

A:\*TEMP\*IFASTEST.rfd - IFRFDUTL

ファイル(F) 表示(V) ヘルプ(H)

IFAS一般ファイル情報

ファイル種別(K) IFAS一般ファイル名(F)

索引順編成 A:\*TEMP\*IFASTEST 参照...

定義済み列情報

列名	データ型	桁数
COLUMNS1	COMP-1	
COLUMNS2	文字	5

追加(A) 削除(D) 変更(M)

列情報定義

列名(N)

データ型(O)

COMP-1

桁数(P) 小数点以下(S)

0 0

定義済みインデックス情報

重複キー	キー情報
なし	COLUMNS1

定義(Q)... 削除(R)

レディ NUM

**上書き保存(S)**

画面上で変更した定義情報を読み込んだRFDファイルに上書き保存します。保存時にサーバコンピュータ名とフォーマット名を入力します。

**名前を付けて保存(A)**

画面で設定した定義情報を新規にRFDファイルとして保存します。保存時にサーバコンピュータ名とフォーマット名を入力します。

**アプリケーションの終了(X)**

本ユーティリティを終了します。

**表示(V)****ツールバー(T)**

ツールバーを表示します。

**ステータスバー(S)**

ステータスバーを表示します。

## ヘルプ(H)



## IFRFDUTLの使い方(H)

オンラインヘルプを表示します。

## バージョン情報(A)

バージョン情報、著作権情報を表示します。

## 注意事項

- ・ 入力するRFDファイルにマルチレコードフォーマットの指定がある場合は、先頭のフォーマットを有効とします。
- ・ 入力するRFDファイルに仮想表の定義が存在した場合、これを無視します。
- ・ 入力するRFDファイルにコメント,REDEFINES句,RENAMES句および集団項目のレコードが存在する場合、このレコードは無視されます。
- ・ 入力するRFDファイルに「編集項目あり」のレコードが存在する場合、このレコードは「編集項目なし」のレコードとして処理されます。
- ・ 入力したマルチレコードフォーマットの定義があるRFDファイルに上書きすると、先頭のフォーマット以外は削除されます。
- ・ 入力した仮想表の定義があるRFDファイルに上書きすると、仮想表の定義は削除されます。
- ・ 別フォーマット形式を持つRFDファイル(従来形式)は、入力→上書きするとRFDファイルに変換されます。
- ・ 同じ列名が存在する場合、順序番号を付加して列名が重複しないように表示します。
- ・ IFAS一般ファイル情報で指定したIFAS一般ファイル名のディレクトリと異なるディレクトリ配下にRFDファイルを保存しようとすると、確認ダイアログが表示されます。IFAS FILE、IFASPRO RDBの機能では、対応するIFAS一般ファイルと同じディレクトリの配下にRFDファイルが存在しているという前提で動作します。
- ・ IFAS一般ファイルのファイル種別が索引順編成の時、1つ以上のインデックス定義が必須となります。インデックスが1つでも定義されていないとRFDファイルへの書き込みはできません。
- ・ RFDファイルで指定できる最大レコードサイズは、64K-1バイトです。

## 7-1-5 移行ユーティリティ

IFAS FILE、IFASPRO RDBは、他システムからExpressサーバ/PCクライアント<sup>\*1</sup>へのシステム移行を支援する目的で、他システムのファイル資産(順編成ファイル/相対編成ファイル/索引順編成ファイル)をIFAS一般ファイルへ移行する機能を提供しています。このファイル移行機能は、コマンドラインから起動する移行ユーティリティにより実現しています。

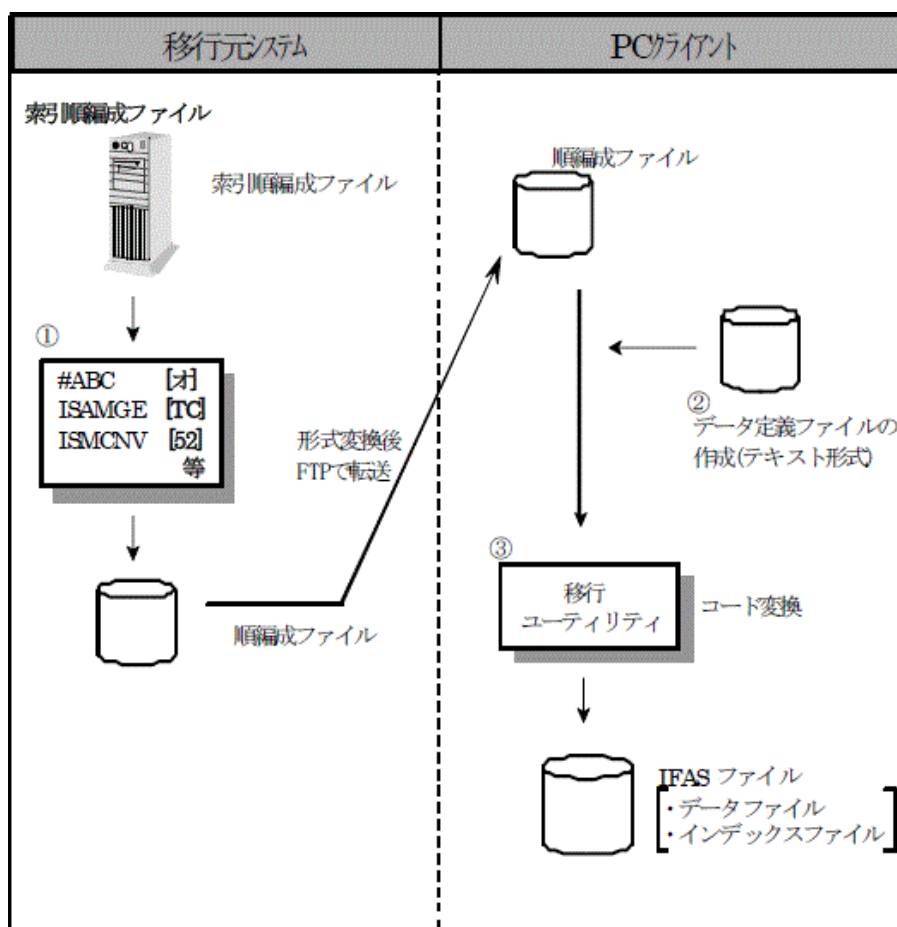
移行ユーティリティは、他システム上で移行元ファイルを順編成ファイルに変換した後、Expressサーバ/PCクライアント上にバイナリ転送したファイルを入力し、フィールド単位に必要なコード変換を行いながらIFAS一般ファイルを構築します。

他システム上で各編成ファイルを利用していたCOBOLプログラムをファイル定義に関してExpressサーバ/PCクライアント上のIFAS一般ファイル向けに変更することで、IFAS一般ファイルが利用できるようになります。

<sup>\*1</sup> 他システムとは、オフコン(A-VX II (10))/TC(A-VX5)/N5200 (PTOS)の他 ACOS-2 も含まれます。また、移行元システムということもあります。

## データ移行作業

以下の図は、索引順編成ファイルの移行を例として、IBM形式フロッピーディスクを入力ファイルとしたデータ移行作業の手順を示しています。



データ移行作業手順例

## 入力ファイルの準備

移行ユーティリティへの入力ファイルは、移行元システム上で作成します。

移行元システム上の移行したいファイルを、付属のファイル変換ユーティリティ(ISAMGE等)を利用して順編成ファイルに変換します。この時、**文字コードは端末内部コード(E)、またはJIPS(E)でなければなりません。**

変換後のファイルをFTPクライアントユーティリティによりExpressサーバ/PCクライアント上にバイナリ転送します。

## データ定義ファイルの準備

移行するファイルのレコード属性などを記述したデータ定義ファイルを作成します。データ定義ファイルは、テキストファイルで、以下の項目を指定します(書式は「**データ定義ファイルの作成**」を参照してください)。

### ・入出力ファイル情報

他システムからファイル転送した移行元のファイル名、および移行先のIFAS一般ファイルのファイル名をフルパス名で指定します。

### ・データ項目の属性定義

データ項目の属性(データ名、データタイプ、桁数)をレコードのフィールド毎に指定します。

マルチレコードフォーマットの場合は、レコードフォーマット毎にデータ項目の属性を指定します。

### ・インデックスの定義

IFAS一般ファイルとして索引順編成ファイルを選択した時、データ項目として指定したフィールドをインデックスとして定義する場合に指定します。

## 移行ユーティリティの実行

移行ユーティリティをコマンドラインより起動して、ファイル変換を実施します。この時、移行元のファイルのレコード形式を指定したデータ定義ファイル名を指定します<sup>\*1</sup>。

以下にコマンドの入力形式を示します。

IFASFCNV. EXE データ定義ファイル名

なお、以下に示すコマンドラインオプションにてオンラインマニュアルを参照できます。

IFASFCNV /? (または -?)

## 移行ユーティリティの終了

移行ユーティリティは、データ移行の結果をコンソールメッセージにて通知します。

### ・「正常に移行できました」

すべてのデータレコードを正常にIFAS一般ファイルに移行しました。

### ・「移行はできました. エラーとなったレコードがあるので確認してください」

データ移行処理中に何らかのエラーを検出したが、そのエラーが致命的とならなかった時にこのメッセージが出力されます。

本メッセージの前に、警告メッセージとレコード番号が出力されているので原因解析後、再度移行ユーティリティを実行してください。

---

<sup>\*1</sup> データ定義ファイル名を省略すると、バージョン情報と起動方法がコンソールに出力されます。



## データ定義ファイルの作成

データ定義ファイルはテキスト形式で記述し、複数のセクションからなります。各セクションは以下の形式です。

[セクション名]

キーワード1名=値1

キーワード2名=値2

(以下続く)

### 1. File Infoセクション

入力／出力ファイル情報を指定します。

#### 【書式】

(1)IFAS ファイル編成タイプに索引順編成ファイル(MIX互換ファイル)を指定する

[File Info]

InFile= 移行元のファイルのフルパスファイル名 (DISKの場合)

ドライブ名:データセット名 (FDの場合)

IFASFileType= IFAS ファイル編成タイプ

IdxSeq: 索引順編成ファイル (既定値)

IdxSeqM: MIX互換ファイル

IFASDataFile= 移行先のIFASデータファイルのフルパスファイル名

IFASIndexFile= 移行先のIFASインデックスファイルのフルパスファイル名

(2)IFAS ファイル編成タイプに相対編成ファイル／順編成ファイルを指定する

[File Info]

InFile= 移行元のファイルのフルパスファイル名 (DISKの場合)

ドライブ名:データセット名 (FDの場合)

IFASFileType= IFAS ファイル編成タイプ

Rel: 相対編成ファイル

Seq: 順編成ファイル

IFASDatafile= 移行先のIFASデータファイルのフルパスファイル名

### 2. (レコードフォーマット名)Data Type Infoセクション

各レコードフォーマットについてデータ項目の属性を指定します。

#### 【書式】

[(レコードフォーマット名) Data Type Info]

Data1= データ項目名, データタイプ, 桁数[, 符号]

Data2= データ項目名, データタイプ, 桁数[, 符号]

:

Datan= データ項目名, データタイプ, 桁数[, 符号]

#### 【パラメータ説明】

(レコードフォーマット名): マルチレコードフォーマットを定義する場合のみ必要です。  
シングルレコードフォーマットの場合は指定してはいけません。

データ項目名: 基本項目名(最小単位のデータ項目)、または集団項目名  
を31文字以内で指定します。

データタイプ: データ項目のデータ属性です。()内は短縮形です。

文字列: Char(Chr)

日本語: CharN(ChrN)

外部10進数: DecimalZoned(DecZ)

内部10進数: Decimal(Dec)

単精度固定2進数: Binary(Bin)

倍精度固定2進数: BinaryDouble(BinD)

集団項目: Group(Grp)

桁数: データ項目のサイズを指定します。データ項目名が集団項目名の時は、従属する基本項目数を指定します。  
各データタイプにより指定可能な最大桁数が異なります。

文字列: 2048文字  
日本語: 1024文字  
外部10進数: 18桁  
内部10進数: 18桁  
単精度固定2進数: 2バイト固定  
倍精度固定2進数: 4バイト固定  
集団項目: ∞

符号: データ項目が符号付きの外部10進数と内部10進数の時、符号位置を指定します。  
ただし、内部10進数はSTのみ指定可能です。

右重ね符号: ST  
左重ね符号: SL  
右分離符号: STS  
左分離符号: SLS

## 【指定例】

データ項目名	データの表現形式	SIGN句の形式	データ項目の定義形式
ABC	X(3)	無	Data1=ABC,Chr,3
	N(3)	無	Data1=ABC,ChrN,3
	9(3)	無	Data1=ABC,DecZ,3
	S9(3)	無	Data1=ABC,DecZ,3,ST
	S9(3)	TRAILING	Data1=ABC,DecZ,3,ST
	S9(3)	LEADING	Data1=ABC,DecZ,3,SL
	S9(3)	TRAILING SEPARATE CHARACTER	Data1=ABC,DecZ,3,STS
	S9(3)	LEADING SEPARATE CHARACTER	Data1=ABC,DecZ,3,SLS
	9(3)V99	無	Data1=ABC,DecZ,5
	9(3) COMP-3	無	Data1=ABC,Dec,3
	S9(3) COMP-3	無	Data1=ABC,Dec,3,ST
	S9(3) COMP-3	TRAILING	Data1=ABC,Dec,3,ST
	COMP-1	無	Data1=ABC,Bin,2
	COMP-2	無	Data1=ABC,BinD,4

## 3. Index Infoセクション

インデックスキーの定義をデータ項目名で指定します。

本セクションは、IFAS ファイル編成タイプが索引順編成ファイル(MIX互換ファイル)の場合のみ有効です。なお、COBOLでの利用を前提とする場合、PrimaryKeyキーは指定しないでください。

## 【書式】

[Index Info]  
[PrimaryKey= プライマリキーとして定義するデータ項目名]  
Key1= キーとして定義するデータ項目名[, キー重複指定]  
Key2= キーとして定義するデータ項目名[, キー重複指定]  
:  
Keyn= キーとして定義するデータ項目名[, キー重複指定]

## 【パラメータ説明】

データ項目名: 基本項目名(最小単位のデータ項目)、または集団項目名を31文字以内で指定します。

キー重複指定: インデックスキー値の重複を許可する場合に Duplicate(Dup)と指定します。()内は省略形です。

#### 4. Multi Record Infoセクション

マルチレコードフォーマット情報を記述します。本セクションは、マルチレコードフォーマットの入力レコードをデータ移行する際、必須となります。

レコードフォーマットを選択する比較条件には、『定数値との等号比較』と『データ項目間の大小比較』があります。これらの比較条件のうち、どちらか一方を指定してください。

##### 【書式】

##### (1) 定数値との等号比較を指定する

[Multi Record Info]

Format=	レコードフォーマット名-1, ..., レコードフォーマット名-256
FlagDataName=	レコードフォーマット名, データ項目名
FlagDataValue1=	データ値, レコードフォーマット名
FlagDataValue2=	データ値, レコードフォーマット名
:	:
FlagDataValuen=	データ値, レコードフォーマット名

##### (2) データ項目間の大小比較を指定する

[Multi Record Info]

Format=	レコードフォーマット名-1, ..., レコードフォーマット名-256
Condition1=	(データ項目名-1 比較演算子 データ項目名-2), レコードフォーマット名
Condition2=	(データ項目名-1 比較演算子 データ項目名-2), レコードフォーマット名
:	:
Conditionn=	(データ項目名-1 比較演算子 データ項目名-2), レコードフォーマット名

##### 【パラメータ説明】

Format(共通):	マルチレコードフォーマットとして定義するレコードフォーマット名を指定します。最大256個まで指定できます。ここで指定したレコードフォーマット名で、[(レコードフォーマット) Data Type Info]セクションを記述してください。
FlagDataName: (『定数値との等号比較』)	レコードフォーマットを選択する際、FlagDataValueキーで指定する定数値と比較するデータ項目名をレコードフォーマット名とペアで指定します。
FlagDataValue: (『定数値との等号比較』)	FlagDataNameで指定したデータ項目の値と等号比較する定数値を指定します。また、その比較条件を評価した結果、“真”となった時選択するレコードフォーマット名を指定します。
Condition: (『データ項目間の大小比較』)	2つのデータ項目の大小関係を比較演算子で表し、その比較条件を評価した結果、“真”となった時選択するレコードフォーマット名を指定します。なお、比較するデータ項目のデータタイプは同一でなければなりません。

【FlagDataValueの定数値の表記方法】

表記方法	指定規則	指定可能なデータタイプ	備考
文字列表記	2重引用符(“)で文字列を囲みます。	Char CharN	データサイズに満たない時、文字列の後ろに空白文字を自動的に補います。
数字表記	データ値を10進整数値で指定します。符号は数値の前に付けます。	Dec DecZ Bin BinD	データサイズに満たない時、数値の前に0を自動的に補います。 データタイプ=Binの時、指定範囲は-32768～+32767となります。 データタイプ=BinDの時、指定範囲は-2147483648～+2147483647となります。
16進表記	16進コード文字列の先頭に0xを付けます。	Grp以外のすべてのデータタイプ	データサイズに満たないとエラーになります。

【FlagDataValueの定数値の指定例】

データ項目の定義形式	データ値	FlagDataValue キーの指定形式
Data1=ABC,Chr,3	ABC	“ABC”
	A△△	“A”
	△△A	“△△A”
Data1=ABC,ChrN,3	あいう	“あいう”
	あ△△	“あ”
	あ△う	“あ△う”
Data1=ABC,DecZ,3	123	123
	012	12
Data1=ABC,DecZ,3,ST	123	123(または+123)
Data1=ABC,DecZ,3,SL		
Data1=ABC,DecZ,3,STS	-789	-789
Data1=ABC,DecZ,3,SLS		
Data1=ABC,Dec,3	012	12(または+12)
Data1=ABC,Dec,3,ST		
Data1=ABC,Bin,2	-001	-1
Data1=ABC,BinD,4		
Data1=ABC,DecZ,5	12345	0xf1f2f3f4f5
Data1=ABC,DecZ,3,ST	-123	0xf1f2d3
Data1=ABC,Bin,2	-2	0xfffe

【Conditionの比較演算子の表記方法】

比較演算子	意味	使用例	表記方法
=	等しい	ABCがXYZと等しい	ABC=XYZ
!=	等しくない	ABCがXYZと等しくない	ABC!=XYZ
>	より大きい	ABCがXYZより大きい	ABC>XYZ
>=	より大きいか、等しい	ABCがXYZより大きいか、等しい	ABC>=XYZ
<	より小さい	ABCがXYZより小さい	ABC<XYZ
<=	より小さいか、等しい	ABCがXYZより小さいか、等しい	ABC<=XYZ

## 5. Option Infoセクション

コード変換のための文字コードや漢字シフトコードの有無(入力コードがJIPS(E)の場合)を記述します。

### 【書式】

[Option Info]

ConvertCode=           入力／出力コード  
ApplyShiftCode=       漢字シフトコード情報

### 【パラメータ説明】

ConvertCode:           入力コードと出力コードを選択します。  
                  INTE-SJIS:       内部コード(E)からシフトJISコードに変換します。(既定値)  
                  JIPSE-SJIS:      JIPS(E)コードからシフトJISコードに変換します。  
漢字シフトコード情報:   入力コード中に漢字シフトコード(KI/KO)を含むか否かを指定します。本オプションは、ConvertCode=JIPSE-SJISの時のみ有効です。  
                  ON:           漢字シフトコードを含む(既定値)  
                  OFF:          漢字シフトコードを含まない

## 注意事項

### 1. データ移行作業について

#### (1) 移行元システムでの順編成ファイルへの変換

削除記録を含む相対編成ファイルをCOPYユーティリティで順編成ファイルに変換する場合、フィジカル処理コピーしてください。

ロジカル処理コピーした場合、格納されたレコードの並びによってはIFAS一般ファイル(相対編成ファイル)に移行した時の相対レコード番号が元のファイルと異なることがあります。

### 2. データ定義ファイルの作成について

#### (1) 全般

- ① 次のセクションのキー名指定で、添字の番号は昇順に抜けないように指定してください。番号が抜けた場合、それ以降の指定は無視されます。  
 [Data Type Info]セクションのDataキー  
 [Index Info]セクションのKeyキー  
 [Multi Record Info] セクションのFlagDataValueキー  
 [Multi Record Info] セクションのConditionキー
- ② [Data Type Info]セクションでデータ項目の属性定義を指定する際、同一セクション(同一レコードフォーマット)内でデータ項目名が重複してはいけません。ただし、『FILLER』は例外です。
- ③ 複数の連続するデータ項目を1つのインデックスキーとする場合、[Data Type Info]セクションで、対象となるデータ項目の上位レベルのデータ項目名を集団項目に定義してください。この時、定義する集団項目名が階層化している集団項目を従属する場合、そのメンバ数には、下位レベルの集団項目名は数に含まれませんので注意が必要です。

#### 例

```
03    Example.
05    Ex-1.
      07    Ex-1a      PIC      9(01).
      07    Ex-1a      PIC      X(01).
05    Ex-2            PIC      X(01).
03    Next            PIC      9(01).
```

上記例で、Example をインデックスキーとするには、次のように記述します。

```
[Data Type Info]
Data1=Example,Grp,3
Data2=Ex-1,Grp,2
Data3=Ex-1a,DecZ,1
Data4=Ex-1b,Chr,1
Data5=Ex-2,Chr,1
Data6=Next,DecZ,1
[Index Info]
Key1=Example
```

- ④ [File Info]セクションのIFASFileTypeキーにIdxSeqM(MIX 互換ファイル)を指定した場合には、[Index Info]セクションのPrimaryKeyキーの指定はできません。
- ⑤ COBOLの記述句に関して以下の注意があります。  
 (i) OCCURS句は、繰り返しの数に応じてデータ項目を定義してください。

例.03 Example PIC X(10) OCCURS 3 TIMES.



03	Ex-1	PIC	X(10).
03	Ex-2	PIC	X(10).
03	Ex-3	PIC	X(10).

```
[Data Type Info]
Data1=Ex-1,Char,10
Data2=Ex-2,Char,10
Data3=Ex-3,Char,10
```

(ii) SYNCHRONIZED句は、桁詰めによって生じる空きバイトについてもデータ定義をする必要があります。

- ⑥ COBOLでの利用を前提とする場合、[File Info]セクションのIFASFileTypeキーにIdxSeqM (MIX互換ファイル)を指定するか、[Index Info]セクションにPrimaryKeyキーの指定をしないようにしてください。

## (2) マルチレコードフォーマット

- ① 再定義していないデータ項目(群)、つまり各レコードフォーマット共通部分の属性定義は、データ項目名も含め必ず共通の指定を行ってください。
- ② プライマリキーとして指定できるデータ項目名は、各レコードフォーマット共通部分のみです。
- ③ [Multi Record Info]セクションのFlagDataValueキーに重ね符号付き外部10進数の値を指定する時、その入力データ値の符号(16進コード)にA、B、E、Fを使用している場合、指定されたレコードフォーマットを正しく選択できません。この場合は「16進表記」で値を指定してください。
- ④ [Multi Record Info]セクションのFlagDataNameキーおよびFlagDataValueキーとConditionキーとは、同時に指定できません。
- ⑤ [Multi Record Info]セクションのConditionキーで比較条件式を記述する場合、同一データ項目名は指定できません。また、異なるデータタイプであってもいけません。
- ⑥ レコードフォーマットを選択する比較条件の評価順序は、次のとおりです。
  - (i) 条件-1を評価する。
  - (ii) 条件-1が“真”ならば、指定されたレコードフォーマットを選択して、データの移行を行う。
  - (iii) 条件-1が“偽”ならば、条件-2を評価する。
 以降、(ii)(iii)を繰り返す。

## 3. 移行ユーティリティの実行について

### (1) 全般

入力レコードサイズは、指定されたデータ項目のサイズ(符号バイトを含む)の合計となります。なお、最終レコードのレコードサイズが上記合計に満たない場合は、足りない分がゼロパディングされます。

### (2) マルチレコードフォーマット

マルチレコードフォーマットのレコードをIFAS一般ファイルに移行する際にレコードフォーマットの選択ができない場合、その時入力したレコードは通常スキップされます。ただし、最初を入力レコードであればデータ移行を中止します。

## データ定義ファイルの作成例

### 1. シングルレコードフォーマット

#### (1) COBOLデータ記述項

この例では、MIXファイルを使用し、データ項目名“KEY”“NAME”をインデックスキーとします。

01 TIME-CARD.			
02 KEY.			
03 NO	PIC	9(6).	
03 NAME.			
05 1ST-NAME	PIC	N(5).	
05 2ND-NAME	PIC	N(5).	
02 SHOZOKU	PIC	N(10).	
02 HIZUKE.			
03 YEAR	PIC	9(4).	
03 MONTH	PIC	9(2).	
03 DAY	PIC	9(2).	
02 TIME.			
03 HH	PIC	9(2).	
03 MM	PIC	9(2).	
03 SS	PIC	9(2).	
02 MEMO.			
03 FILLER	PIC	X(20).	

#### (2) データ定義ファイル

データレベルをわかりやすくするためにインデントを付けて記述しましたが、揃えて記述してもかまいません。

[Data Type Info]

Data1=KEY,Grp,3

Data2=NO,DecZ,6

Data3=NAME,Grp,2

Data4=1ST-NAME,ChrN,5

Data5=2ND-NAME,ChrN,5

Data6=SHOZOKU,CharN,10

Data7=HIZUKE,Grp,3

Data8=YEAR,DecZ,4

Data9=MONTH,DecZ,2

Data10=DAY,DecZ,2

Data11=TIME,Grp,3

Data12=HH,DecZ,2

Data13=MM,DecZ,2

Data14=SS,DecZ,2

Data15=FILLER,Chr,20

[Index Info]

Key1=KEY

Key2=NAME,Dup

[File Info]

InFile=b:TIMECARD



```
IFASFileType=IdxSeqM
IFASDataFile=a:¥user¥data
IFASIndexFile=a:¥user¥index
```

```
[Option Info]
ConvertCode=INTE-SJIS
```

## 2. マルチレコードフォーマット

### (1) COBOLデータ記述項

この例では、MIXファイルを使用し、データ項目名“KEY”“NAME”をインデックスキーとします。  
また、データ項目名“PERSONAL”の値により2つのレコードフォーマットを選択します。

```
01 TIME-CARD.
  02 KEY.
    03 NO                      PIC      9(6).
    03 NAME.
      05 1ST-NAME              PIC      N(5).
      05 2ND-NAME              PIC      N(5).
  02 SHOZOKU                   PIC      N(10).
  02 PERSONAL                  PIC      S9(1).
  02 HIZUKE.
    03 H-YEAR                  PIC      9(4).
    03 H-MONTH                 PIC      9(2).
    03 H-DAY                   PIC      9(2).
  02 BIRTHDAY                  REDIFINES      HIZUKE.
    03 B-YEAR                  PIC      9(4).
    03 B-MONTH                 PIC      9(2).
    03 B-DAY                   PIC      9(2).
  02 TIME.
    03 HH                      PIC      9(2).
    03 MM                      PIC      9(2).
    03 SS                      PIC      9(2).
    03 FILLER                  PIC      X(2).
  02 BODY                      REDIFINES      TIME.
    03 HEIGHT                  PIC      999V9.
    03 WEIGHT                  PIC      999V9.
  02 MEMO.
    03 FILLER                  PIC      X(20).
  02 ETC                      REDIFINES      MEMO.
    03 HOBBY                   PIC      N(10).
```

### (2) データ定義ファイル

```
[Multi Record Info]
Format=TIMECARD,PERSON
FlagDataName=TIMECARD,PERSONAL
FlagDataValue1=1,TIMECARD
FlagDataValue2=0xd1,PERSON
; PERSONAL=1 の時、TIMECARD を選択
; PERSONAL=-1 の時、PERSON を選択
```

```
// TIMECARDのレコードフォーマット
[TIMECARD Data Type Info]
Data1=KEY,Grp,3
```

```

Data2=NO,DecZ,6
Data3=NAME,Grp,2
  Data4=1ST-NAME,ChrN,5
  Data5=2ND-NAME,ChrN,5
Data6=SHOZOKU,ChrN,10
Data7=PERSONAL,DecZ,1,ST
; Data1からData7までが”TIMECARD”と”PERSON”の共通定義部

```

```

Data8=HIZUKE,Grp,3
  Data9=H-YEAR,DecZ,4
  Data10=H-MONTH,DecZ,2
  Data11=H-DAY,DecZ,2
Data12=TIME,Grp,4
  Data13=HH,DecZ,2
  Data14=MM,DecZ,2
  Data15=SS,DecZ,2
  Data16=FILLER,Chr,2
Data17=FILLER,Chr,20

```

```
// PERSONのレコードフォーマット
```

```
[PERSON Data Type Info]
```

```

Data1=KEY,Grp,3
  Data2=NO,DecZ,6
  Data3=NAME,Grp,2
    Data4=1ST-NAME,ChrN,5
    Data5=2ND-NAME,ChrN,5
Data6=SHOZOKU,ChrN,10
Data7=PERSONAL,DecZ,1,ST
; Data1からData7までが”TIMECARD”と”PERSON”の共通定義部

```

```

Data8=BIRTHDAY,Grp,3
  Data9=B-YEAR,DecZ,4
  Data10=B-MONTH,DecZ,2
  Data11=B-DAY,DecZ,2
Data12=BODY,Grp,2
  Data13=HEIGHT,DecZ,4
  Data14=WEIGHT,DecZ,4
Data15=HOBBY,ChrN,10

```

```
[Index Info]
```

```
Key1=KEY
```

```
Key2=NAME,Dup
```

```
[File Info]
```

```
InFile=b:TIMECARD
```

```
IFASFileType=IdxSeqM
```

```
IFASDataFile=a:¥user¥data
```

```
IFASIndexFile=a:¥user¥index
```

```
[Option Info]
```

```
ConvertCode=INTE-SJIS
```

## 7-2 RDB表の管理

利用者が簡単にRDB環境の構築や保守を行うためのユーティリティプログラムを提供します。

### 7-2-1 表定義ユーティリティ

IFAS RDB表を管理するRDB管理ファイルの作成、データベース/RDB表/インデックスの構築や保守等を行うためのRDB表の管理ツールとして、表定義ユーティリティを提供します。

以降、表定義ユーティリティの説明内での管理ファイルとはRDB管理ファイルを、RDB表／表とはIFAS RDB表を指します。

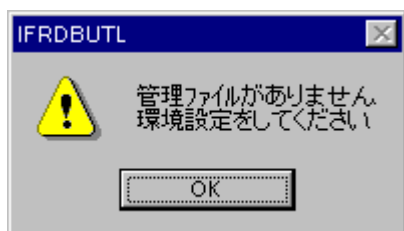
表定義ユーティリティでは、以下のような処理を行うことが可能です。

環境設定の変更	RDB導入時の管理ファイル作成、各種設定の変更
RDB環境の参照	仮想表の構成情報表示、サーバ上RDB環境の表示
RDB環境の構築	データベース/RDB表/インデックスの作成と削除、別名保存、コピー、移動
RDB環境の変更	名前/オプション/列情報の変更
RDB環境のチェック	データベース環境のチェックと再構築
RDB表の各種操作	データ削除、インポート/エクスポート、RFD入出力、APDIRECTORへの登録 (※ただし、現在はAPDIRECTORが対応していないため、APDIRECTORへの登録は利用できません。)

#### 【起動】

IFAS FILEプログラムグループから「表定義ユーティリティ」を選択して起動します。

IFASPRO RDBをインストールした直後に起動した場合、管理ファイルが作成されていないため、次のようなメッセージが表示されます。



この場合、[環境設定の変更]を行って、管理ファイルのディレクトリ名を設定してください。

「表定義ユーティリティ」を起動することにより、管理ファイルへのアクセスが行われます。

#### 【終了】

いずれかの方法で終了できます。

メニュー[ファイル]→[アプリケーションの終了]を指定する。

[コントロールメニュー ボックス]の[閉じる]を選択する。

### 環境設定の変更

RDB導入時の管理ファイル作成や各種動作環境の設定値を変更します。

#### 【設定変更方法】

- ① メニュー[環境設定]→[変更]を指定します。

- ② 環境設定の変更ダイアログにておのおのの設定情報を指定してください。

以下の項目について利用者が任意に設定を変更できます。

### RDB管理ディレクトリの設定

データベース機能を利用するには、管理ファイル群(各種定義情報を格納したファイル群)を作成するディレクトリパスを設定しなければなりません。このディレクトリとその配下の管理ファイル群を総称して**RDB管理ディレクトリ(IDIR)**と呼びます。

既定値は、「IFASPRO RDBインストールディレクトリパス¥IDIR」です。

### 作業領域の設定

作業領域を示すディレクトリパスです。

管理ディレクトリとは異なるディレクトリ名を指定してください。

既定値は、「IFASPRO RDBインストールディレクトリパス¥TMP」です。

### データベース名の既定値

COBOL、カスタムコントロールインタフェースで表を作成したときにデータベース名が省略された場合、この名前をデータベース名とします。

既定値は、「\*RDB」です。

### エラーログの設定

エラーログを保存するかどうかを指定します。また、エラーログを保存するとしたときの出力先も指定できます。

既定値は、「保存しない」です。

「保存する」とした場合、出力先のファイルパスも指定してください。

### 管理ファイルの更新ログの設定

表を新規に作成した場合など管理ファイルを更新したとき、その管理ファイルの更新イメージを更新ログファイルに保存するかどうかを指定します。保存した更新ログはロールフォワード機能で使うことができます。更新ログの出力先である更新ログファイルは、運用支援ツールにて設定されます。

既定値は、「保存しない」です。

### 表オープン時の更新ログのデフォルトオプション設定

表を更新したとき、その更新イメージを更新ログファイルに保存するかどうかを表オープン時のデフォルトオプションとして表定義上に保持することができます。これは表定義ユーティリティで作表成時に指定しますが、この既定値をここで指定することができます。保存した更新ログはロールフ

ワード機能で 사용할 수 있습니다. 更新 로그의 出力先である更新 로그 파일は、運用 支援 ツールにて設定されます。

既定値は、「保存しない」です。

### 表オープン時のトランザクション処理のデフォルトオプション設定

表をアクセスしたときに、トランザクション処理するかどうかを表オープン時のデフォルトオプションとして表定義上に保持することができます。これは表定義ユーティリティで作成時に指定しますが、この既定値をここで指定することができます。トランザクション処理を行うとコミット/ロールバック機能が有効になります。また、プログラム異常終了時のロールバックやシステム再起動時のリスタートリカバリによるロールバックが行われるようになります。

既定値は、「トランザクション処理しない」です。

### 表削除時のRFDファイル自動削除オプション設定

表削除時にRFDファイルが存在する場合、RFDファイルを自動的に削除するかどうかを指定することができます。実行時指定を設定した場合、表削除でRFDファイルが存在するごとにRFDファイルを削除するかどうかの問い合わせが行われます。

既定値は、「削除しない」です。

## RDB環境の参照

データベース/表/インデックス定義情報についての表示を行います。



### [ウィンドウの説明]

#### ・IDIRウィンドウ(左側)

IFAS RDBの管理ファイル内に定義されたデータベース/表/インデックスの名前をツリー表示します。  
データベースなどの作成/削除を行うには、ここに表示されたツリーのアイテムを選択する必要があります。

#### ・情報ウィンドウ(右側)

IDIRウィンドウで選択されたアイテムに関する情報を表示します。  
IDIRウィンドウでデータベースが選択された場合は、データベースに含まれる表の情報が、また、表が選択された場合は、表を構成する列の情報が表示されます。

なお、次のような場合は、表示情報を更新する必要があります。

- ・表定義ユーティリティ以外の手段でデータベースや表、インデックスを作成/削除したとき。
- ・同じ管理ファイルへ他の表定義ユーティリティからの定義変更が行われたとき。

メニュー[表示]→[最新の情報に更新]を指定してください。

## 構成列情報の参照

選択/結合型仮想表の各列の元となる列の情報を表示します。

### 【構成列情報の参照方法】

- ① メニュー[表示]→[詳細表示]をチェックしておくことで、選択/結合型仮想表を選択した場合、情報ウィンドウに元となる列の情報が追加されます。

## 構成表情報の参照

選択/結合/合併型仮想表の元となる構成表の情報を表示します。

また、結合型仮想表では、結合表の関係式も表示します。

### 【構成表情報の参照方法】

- ① 選択/結合/合併型いずれかの仮想表をクリックし、仮想表名を反転表示させます。
- ② 以下のいずれかの操作で、選択した仮想表に対応する参照ダイアログが表示されます。
  - ・メニュー[ファイル]→[参照]を指定する
  - ・マウスの右ボタンをクリック、ポップアップメニュー[参照]を指定する

## サーバマシン上RDB環境の参照

ローカルマシンからサーバマシン上のデータベース/表/インデックス定義情報を参照することができます。

本指定時、参照機能のみが動作します。

### 【サーバマシン上RDB環境の参照方法】

- ① メニュー[環境設定]→[端末設定]を指定します。
- ② 端末設定ダイアログにてサーバコンピュータ名を指定してください。

なお、ローカルマシンの情報に戻す場合は、必ず端末設定ダイアログにて「マイコンピュータ」を選択してください。サーバが指定された場合、表定義ユーティリティを再起動し直しても終了時の指定サーバ情報は記憶されます。

指定サーバマシンの情報入手に失敗した場合、端末設定前の表示情報に戻されます。

## RDB環境の構築

データベース/表/インデックス定義の作成や削除を行います。

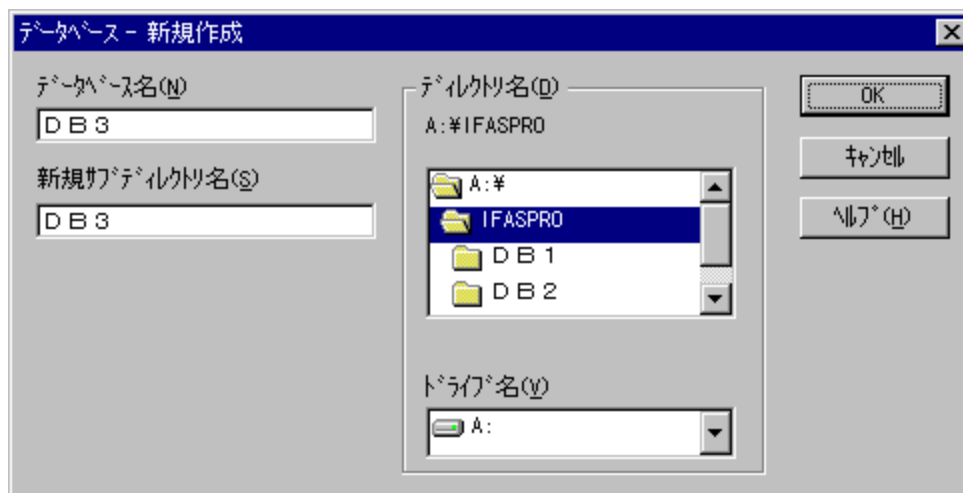
## データベースの作成／削除

### データベースを作成するには

- ・データベース名は32文字以内で指定し、「ピリオド(.)」「ダブルクォーテーション(”)」[ブランク(‘△’)]を含んではなりません。また、システムでディレクトリとして作成できない名前も使用できません。
- ・データベース名にIFASPRO RDBの予約語は使用できません。
- ・データベース名は管理ファイル内で一意でなければなりません。
- ・データベースのディレクトリは管理ファイル内で一意でなければなりません。

### 【データベース作成方法】

- ① 管理ディレクトリをクリックし、管理ディレクトリ名を反転表示させます。
- ② 以下のいずれかの操作で、データベースの新規作成ダイアログが現れます。
  - ・メニュー[ファイル]→[新規作成]→[データベース]を指定する
  - ・ツールバーのデータベース作成ボタンをクリックする
  - ・マウスの右ボタンをクリック、ポップアップメニュー[データベースの新規作成]を指定する
- ③ データベース名およびデータベースを作成するディレクトリを指定してください。指定されたサブディレクトリが存在しなければ、新規に作成します。



### データベースを削除するには

- 削除したいデータベース配下の表を元に、別のデータベースで仮想表が定義されてはいけません。

#### 【データベース削除方法】

- ① 削除するデータベースをクリックし、データベース名を反転表示させます。
- ② 以下のいずれかの操作で、削除ダイアログが現れます。
  - ・メニュー[ファイル]→[削除]を指定する
  - ・ツールバーの削除ボタンをクリックする
  - ・マウスの右ボタンをクリック、ポップアップメニュー[削除]を指定する

## データベースのコピー

既存のデータベースをもとに、別のデータベースを作成することができます。データベース配下の表は既存のデータベースと同じ定義情報とデータを持ちます。

#### 【データベースのコピー方法】

- ① コピーしたいデータベースをクリックし、データベース名を反転表示させます。
- ② 以下のいずれかの操作を行います。
  - ・メニュー[ファイル]→[コピー]を指定する
  - ・マウスの右ボタンをクリック、ポップアップメニュー[コピー]を指定する
- ③ 管理ディレクトリをクリックし、管理ディレクトリ名を反転表示させます。
- ④ 以下のいずれかの操作で、データベースの新規作成ダイアログが現れます。
  - ・メニュー[ファイル]→[貼り付け]を指定する
  - ・マウスの右ボタンをクリック、ポップアップメニュー[貼り付け]を指定する
- ⑤ データベース名およびデータベースを作成するディレクトリを指定してください。指定されたサブディレクトリが存在しなければ、新規に作成します。

## 表の作成／削除

### 表と列を作成するには

- ・ 表名は30文字以内で指定し、「ピリオド(.)」「ダブルクォーテーション(”)」 「ブランク(‘△’)」を含んではなりません。また、システムでファイルとして作成できない名前も使用できません。
- ・ 表名にIFASPRO RDBの予約語は使用できません。
- ・ 表名に大文字、小文字の区別はありません。
- ・ 列名は30文字以内で指定し、「ピリオド(.)」「ダブルクォーテーション(”)」 「ブランク(‘△’)」を含んではなりません。
- ・ 列名にIFASPRO RDBの予約語は使用できません。
- ・ 列名は表内で一意でなければなりません。
- ・ 1つの表で定義できる列数は最大2000個までです。
- ・ 1つの表で列の長さの合計がSQLのデータ型で最大32K－1バイトまで指定できます。
- ・ 列のデータ型にCOMP-5を指定した場合、列の桁数は以下のように変換されます。

指定桁数が1～4桁	: 作成される列の桁数は4桁
指定桁数が5～9桁	: 作成される列の桁数は9桁
指定桁数が10～18桁	: 作成される列の桁数は18桁

### 列名規則について

IFASPRO RDBでは表定義情報を内部に持つSQLエンジンに依存した形式で保持しています。したがって列のデータ型も内部SQLエンジンに依存したものになります。ここで表定義ユーティリティにおいては列のデータ型をCOBOL形式で指定しますが、実際にはSQLの形式に変換して保持します。このときSQLの形式にマッチしないデータ型の場合、COBOLでのデータ型を識別するためのデータ識別子を自動的に付加します。データ識別子は2文字のプリフィックスとして付加されます。したがってファイルメンテナンスキット、COBOL85データファイル変換ツール等を使用して作成した表にアクセスする場合は、データ識別子を含んだ列名を指定する必要があります。COBOLデータ型とデータ識別子の関係は次表を参照してください。



IFASPRO RDBでのデータ型と列名規則

COBOL	IFASPRO RDB	
データ型	SQLデータ型	列名規則 (先頭文字で識別する)
単精度固定2進数 COMP-1	SMALLINT	なし
倍精度固定2進数 COMP-2	INTEGER	なし
符号付き内部10進数 COMP-3	DECIMAL	3__ 5__ N__ n__ B__ b__ C__ c__ 以外
符号なし内部10進数 COMP-3		3__
符号なし外部10進数	UNSIGNED NUMERIC	なし
符号付き外部10進数 TRAILING	NUMERIC	R__ r__ L__ l__ E__ e__ 以外
符号付き外部10進数 TRAILING SEPARATE		R__ または r__
符号付き外部10進数 LEADING		L__ または l__
符号付き外部10進数 LEADING SEPARATE		E__ または e__
符号付き2進数 BINARY	DECIMAL	N__ または n__
符号なし2進数 BINARY		B__ または b__
符号付き2進数 COMP-5		C__ または c__
符号なし2進数 COMP-5		5__
文字 X(n)	CHARACTER	なし
文字(2バイト文字固定) N(n)	NATIONAL CHARACTER	なし

## 《SMALLINT型》

－32768～＋32767の整数値を格納できます。

## 《INTEGER型》

－2147483648～＋2147483647の整数値を格納できます。

## 《DECIMAL型》

符号なし／符号付きパック10進数を最大18桁(小数点以下の桁数を含む)まで格納できます。

## 《UNSIGNED NUMERIC型》

符号なしアンパック10進数を最大18桁(小数点以下の桁数を含む)まで格納できます。

## 《NUMERIC型》

符号付きアンパック10進数を最大18桁(小数点以下の桁数を含む)まで格納できます。

## 《CHARACTER型》

1バイト文字列を最大30000文字まで格納できます。

## 《NATIONAL CHARACTER型》

2バイト文字列を最大15000文字まで格納できます。

作成したい表の種別毎(ごと)に、「基本表の作成」、「選択型仮想表の作成」、「結合型仮想表の作成」、「合併型仮想表の作成」を参照してください。

また、定義済みの表を利用して、新規の表を作成することもできます。方法については、「表の別名保存」を参照してください。

### 表を削除するには

- ・ 削除したい表を元に、仮想表が定義されていないけません。

#### 【表削除方法】

- ① 削除したい表をクリックし、表名を反転表示させます。
- ② 以下のいずれかの操作で、削除ダイアログが現れます。
  - ・ メニュー[ファイル]→[削除]を指定する
  - ・ ツールバーの削除ボタンをクリックする
  - ・ マウスの右ボタンをクリック、ポップアップメニュー[削除]を指定する

## 基本表の作成

#### 【基本表作成方法】

- ① 表を作成したいデータベースをクリックし、データベース名を反転表示させます。
- ② 以下のいずれかの操作で、基本表の新規作成ダイアログが現れます。
  - ・ メニュー[ファイル]→[新規作成]→[基本表]を指定する
  - ・ ツールバーの基本表の作成ボタンをクリックする
  - ・ マウスの右ボタンをクリック、ポップアップメニュー[基本表の新規作成]を指定する
- ③ 表名および列情報定義を指定してください。

列名	SQL列名	データ型	SQLデータ型	桁数
商品コード	商品...	COMP-1	SMALL INT	20
商品名	商品名	文字(2BYTE...	NCHAR	20
商品単価	3_商...	COMP-3符号なし	DECIMAL	9,2
仕入単価	3_仕...	COMP-3符号なし	DECIMAL	9,2

## 選択型仮想表の作成

#### 【選択型仮想表作成方法】

- ① 表を作成したいデータベースをクリックし、データベース名を反転表示させます。
- ② 以下のいずれかの操作で、選択型仮想表の新規作成ダイアログが現れます。
  - ・ メニュー[ファイル]→[新規作成]→[選択型仮想表]を指定する
  - ・ ツールバーの選択型仮想表の作成ボタンをクリックする

- ・ マウスの右ボタンをクリック、ポップアップメニュー[選択型仮想表の新規作成]を指定する
- ③ 表名および選択する列情報を指定してください。



## 結合型仮想表の作成

### 結合型仮想表を作成するには

- ・ 1つの結合型仮想表を構成する表の選択可能な数は63個までとなります。関連名を指定して同じ表を選択した場合は、別表として扱います。
- ・ 選択済み表名に表示されている順番が、結合する表の優先順位となります。
- ・ 結合型仮想表で指定するアクセスキーと対応キーは、データ型、桁数および小数点以下桁数が一致していなければなりません。
- ・ 結合型仮想表でアクセスキーとして指定できる列は、インデックスとして定義されていなければなりません。
- ・ 結合型仮想表で対応キーとして指定できる列は、アクセスキーに指定した列の表よりも優先順位の高い表の列のみとなります。
- ・ 結合型仮想表をセカンダリ表として定義する場合、アクセスキーとして指定できる列はその結合型仮想表のプライマリ表の列のみとなります。
- ・ 表関係指定をしないで、結合型仮想表の結合種別に連鎖を指定することはできません。セカンダリ表別指定時の連鎖も含みます。

### 【結合型仮想表作成方法】

- ① 表を作成したいデータベースをクリックし、データベース名を反転表示させます。
- ② 以下のいずれかの操作で、結合型仮想表の新規作成ダイアログが現れます。
  - ・ メニュー[ファイル]→[新規作成]→[結合型仮想表]を指定する
  - ・ ツールバーの結合型仮想表の作成ボタンをクリックする
  - ・ マウスの右ボタンをクリック、ポップアップメニュー[結合型仮想表の新規作成]を指定する
- ③ 表名、結合する表および選択列情報を指定してください。また、アクセス／対応キーの指定がある場合は、表関係指定のダイアログより指定します。



## 合併型仮想表の作成

### 合併型仮想表を作成するには

- ・ 1つの合併型仮想表を構成する表の選択可能な数は63個までとなります。同じ表を複数指定することはできません。
- ・ 合併型仮想表を構成する表は基本表のみであり、すべての構成表はデータ型、桁数、小数点以下数、列の並びが一致していなければなりません。

### 【合併型仮想表作成方法】

- ① 表を作成したいデータベースをクリックし、データベース名を反転表示させます。
- ② 以下のいずれかの操作で、合併型仮想表の新規作成ダイアログが現れます。
  - ・ メニュー[ファイル]→[新規作成]→[合併型仮想表]を指定する
  - ・ ツールバーの合併型仮想表の作成ボタンをクリックする
  - ・ マウスの右ボタンをクリック、ポップアップメニュー[合併型仮想表の新規作成]を指定する

- ③ 表名、構成する表および列名情報を指定してください。



## 表の別名保存

定義済みの表の定義情報を、別名で保存することにより、新規の表を作成します。

### 【別名保存方法】

- ① 利用したい表をクリックし、表名を反転表示させます。
- ② 以下のいずれかの操作で、選択した表の種別に合った別名保存のダイアログが現れます。
  - ・ メニュー[ファイル]→[変更]→[別名保存]を指定する
  - ・ マウスの右ボタンをクリック、ポップアップメニュー[別名保存]を指定する
- ③ 新表名および定義済み情報の変更を指定してください。

## 表のコピー

既存の表をもとに、別の表を作成することができます。別名保存機能と違い、コピーされた表は、既存の表と同じインデックス定義情報とデータを持ちます。

### 【表のコピー方法】

- ① コピーしたい表をクリックし、表名を反転表示させます。
- ② 以下のいずれかの操作を行います。
  - ・ メニュー[ファイル]→[コピー]を指定する
  - ・ マウスの右ボタンをクリック、ポップアップメニュー[コピー]を指定する
- ③ コピー先のデータベースをクリックし、データベース名を反転表示させます。
- ④ 以下のいずれかの操作を行います。
  - ・ メニュー[ファイル]→[貼り付け]を指定する
  - ・ マウスの右ボタンをクリック、ポップアップメニュー[貼り付け]を指定する
- ⑤ コピー先のデータベースに同じ名前の表が存在している場合、新しい表名を指定するか、[キャンセル]ボタンで処理を終了してください。

## 表の移動

あるデータベース配下にある既存の表を、別のデータベース配下に移動することができます。

### 【表の移動方法】

- ① 移動したい表をクリックし、表名を反転表示させます。
- ② 以下のいずれかの操作を行います。
  - ・ メニュー[ファイル]→[切り取り]を指定する
  - ・ マウスの右ボタンをクリック、ポップアップメニュー[切り取り]を指定する
- ③ 移動先のデータベースをクリックし、データベース名を反転表示させます。
- ④ 以下のいずれかの操作を行います。
  - ・ メニュー[ファイル]→[貼り付け]を指定する
  - ・ マウスの右ボタンをクリック、ポップアップメニュー[貼り付け]を指定する
- ⑤ 移動先のデータベースに同じ名前の表が存在している場合、新しい表名を指定するか、[キャンセル]ボタンで処理を終了してください。

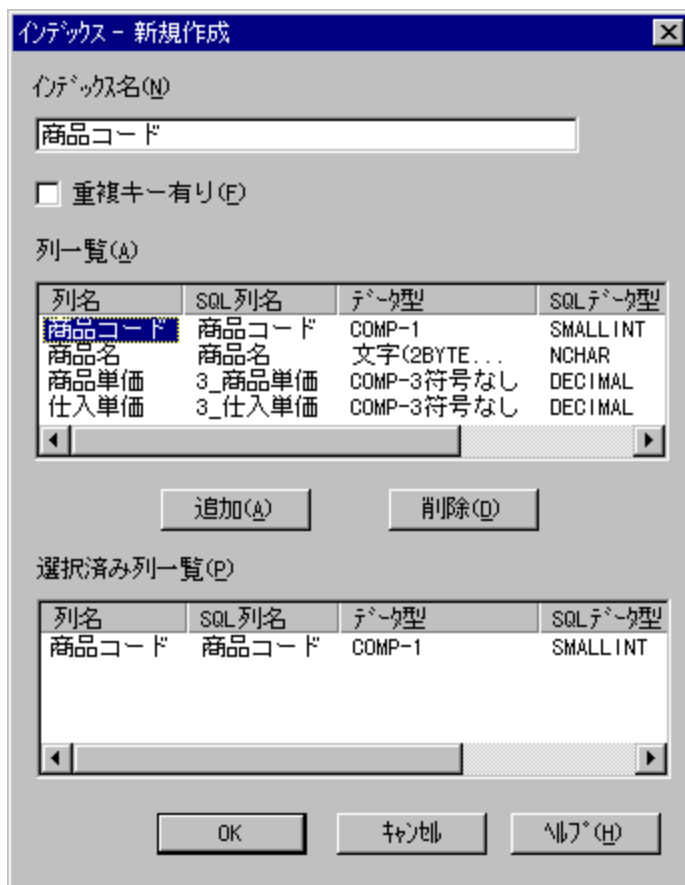
## インデックスの作成／削除

### インデックスを作成するには

- ・ インデックス名は30文字以内で指定し、「ピリオド(.)」「ダブルクォーテーション(”)」 「ブランク(‘△’)」を含んではなりません。
- ・ インデックス名にIFASPRO RDBの予約語は使用できません。
- ・ インデックス名はデータベース内で一意でなければなりません。
- ・ キーとする列名は最大16列まで指定できます。
- ・ 1つの列をインデックスとして定義する場合、キーの長さは最大255バイトまで指定できます。
- ・ 複数の列をインデックスとして定義する場合、各列のキーの長さの合計と列数を足したものが最大255バイトまで指定できます。
- ・ 1つのインデックス定義に対し、同一の列名は重複指定できません。
- ・ 1つの列を複数のインデックス定義で指定可能です。
- ・ 1つの基本表に定義できるインデックス数は最大255個まで指定できます。

### 【インデックス作成方法】

- ① インデックスを作成したい表をクリックし、表名を反転表示させます。
- ② 以下のいずれかの操作で、インデックスの新規作成ダイアログが現れます。
  - ・ メニュー[ファイル]→[新規作成]→[インデックス]を指定する
  - ・ ツールバーのインデックス作成ボタンをクリックする
  - ・ マウスの右ボタンをクリック、ポップアップメニュー[インデックスの新規作成]を指定する
- ③ インデックス名およびインデックスキーとする列名を選択してください。インデックスキーは単一キー、集合キーおよび合成キーのどれでも指定できます。なお、インデックス名は、自動作成された名前が表示されますので、指定し直す必要はありません。



### 【インデックス削除方法】

- ① 削除したいインデックスをクリックし、インデックス名を反転表示させます。
- ② 以下のいずれかの操作で、削除ダイアログが現れます。
  - ・メニュー[ファイル]→[削除]を指定する
  - ・ツールバーの削除ボタンをクリックする
  - ・マウスの右ボタンをクリック、ポップアップメニュー[削除]を指定する

## RDB環境の変更

定義済みの定義情報を変更します。

### 【変更方法】

変更する情報により、方法が異なります。  
詳細は、変更情報毎(ごと)に以下に記述します。



## データベースの変更

### データベース名の変更

- ① 変更したいデータベースをクリックし、データベース名を反転表示させます。
- ② 以下のいずれかの操作で、データベースの名前変更ダイアログが現れます。
  - ・メニュー[ファイル]→[変更]→[名前変更]を指定する
  - ・マウスの右ボタンをクリック、ポップアップメニュー[名前変更]を指定する
- ③ 変更するデータベース名を指定してください。

なお、以下の場合、データベース名の変更はできません。

- ・データベース配下に仮想表が定義されているとき
- ・データベース配下の表を元に仮想表が定義されている

### データベースを作成するディレクトリ名の変更

- ① 変更したいデータベースをクリックし、データベース名を反転表示させます。
- ② 以下のいずれかの操作で、データベースの設定変更ダイアログが現れます。
  - ・メニュー[ファイル]→[変更]→[設定変更]を指定する
  - ・マウスの右ボタンをクリック、ポップアップメニュー[設定変更]を指定する
- ③ [ディレクトリ]タブで変更するディレクトリ名を指定してください。



## 基本表の変更

### 基本表名の変更

- ① 変更したい基本表をクリックし、基本表名を反転表示させます。



- ② 以下のいずれかの操作で、表の名前変更ダイアログが現れます。
  - ・ メニュー[ファイル]→[変更]→[名前変更]を指定する
  - ・ マウスの右ボタンをクリック、ポップアップメニュー[設定変更]を指定する
- ③ 変更する基本表名を指定してください。

なお、以下の場合、基本表名の変更はできません。

- ・ 該当表を元に仮想表が定義されているとき

### 列情報の変更

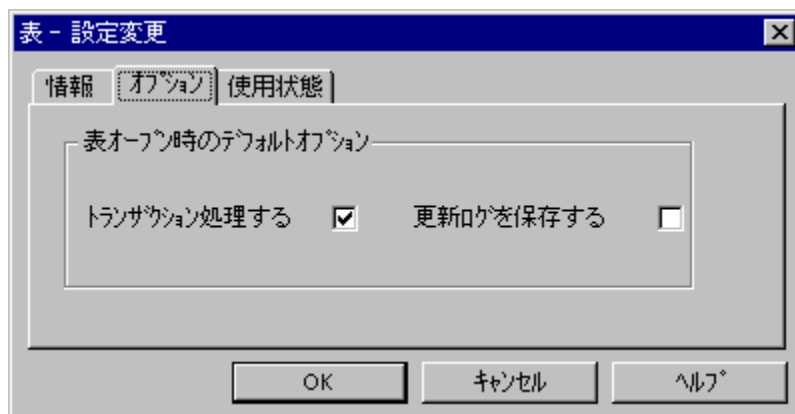
- ① 変更したい基本表をクリックし、基本表名を反転表示させます。
- ② 以下のいずれかの操作で、基本表の列情報変更ダイアログが現れます。
  - ・ メニュー[ファイル]→[変更]→[列情報変更]を指定する
  - ・ マウスの右ボタンをクリック、ポップアップメニュー[列情報変更]を指定する
- ③ 列名の変更および選択列情報の追加／削除／変更を行ってください。

なお、以下の場合、列情報の変更はできません。

- ・ データが存在するとき
- ・ 該当表を元に仮想表が定義されているとき
- ・ 列名変さらにて、変更対象表の元列名と同じ列名へ変更しようとしたとき

### 表オープン時のデフォルトオプションの変更

- ① 変更したい基本表をクリックし、基本表名を反転表示させます。
- ② 以下のいずれかの操作で、表の設定変更ダイアログが現れます。
  - ・ メニュー[ファイル]→[変更]→[設定変更]を指定する
  - ・ マウスの右ボタンをクリック、ポップアップメニュー[設定変更]を指定する
- ③ [オプション]タブの表オープン時のデフォルトオプションを変更し、[OK]ボタンを押してください。変更を取りやめる場合は、[キャンセル]ボタンを押します。



## 選択型仮想表の変更

### 表オープン時のデフォルトオプションの変更

- ① 変更したい選択型仮想表をクリックし、選択型仮想表名を反転表示させます
- ② 以下のいずれかの操作で、表の設定変更ダイアログが現れます。
  - ・ メニュー[ファイル]→[変更]→[設定変更]を指定する
  - ・ マウスの右ボタンをクリック、ポップアップメニュー[設定変更]を指定する
- ③ [オプション]タブの表オープン時のデフォルトオプションを変更し、[OK]ボタンを押してください。変更を取りやめる場合は、[キャンセル]ボタンを押します。

## 合併型仮想表の変更

### 表オープン時のデフォルトオプションの変更

- ① 変更したい合併型仮想表をクリックし、合併型仮想表名を反転表示させます。
- ② 以下のいずれかの操作で、表の設定変更ダイアログが現れます。
  - ・ メニュー[ファイル]→[変更]→[設定変更]を指定する
  - ・ マウスの右ボタンをクリック、ポップアップメニュー[設定変更]を指定する
- ③ [オプション]タブの表オープン時のデフォルトオプションを変更し、[OK]ボタンを押してください。変更を取りやめる場合は、[キャンセル]ボタンを押します。

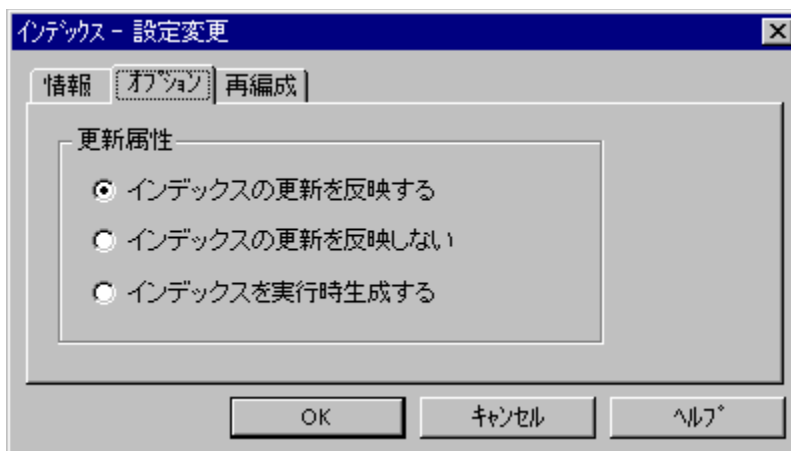
## インデックスの変更

### インデックス名の変更

- ① 変更したいインデックスをクリックし、インデックス名を反転表示させます。
- ② 以下のいずれかの操作で、インデックスの名前変更ダイアログが現れます。
  - ・ メニュー[ファイル]→[変更]→[名前変更]を指定する
  - ・ マウスの右ボタンをクリック、ポップアップメニュー[名前変更]を指定する
- ③ 変更するインデックス名を指定してください。

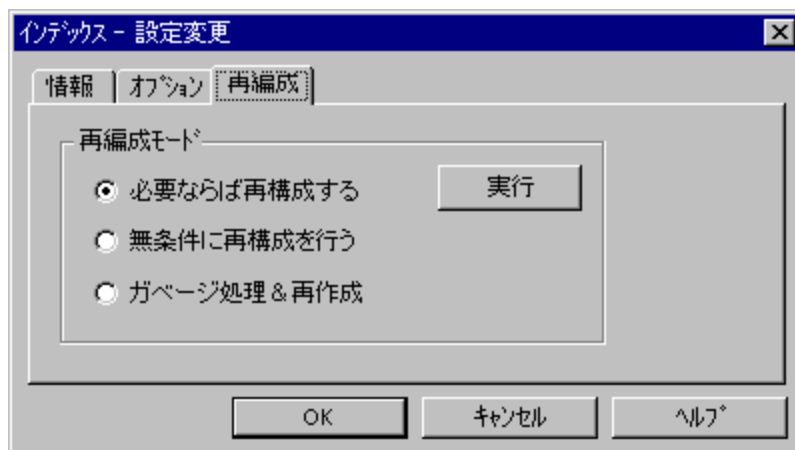
### インデックスの更新属性の変更

- ① 変更したいインデックスをクリックし、インデックス名を反転表示させます。
- ② 以下のいずれかの操作で、インデックスの設定変更ダイアログが現れます。
  - ・ メニュー[ファイル]→[変更]→[設定変更]を指定する
  - ・ マウスの右ボタンをクリック、ポップアップメニュー[設定変更]を指定する
- ③ [オプション]タブの更新属性を選択してください。



### インデックスの再編成

- ① 変更したいインデックスをクリックし、インデックス名を反転表示させます。
- ② 以下のいずれかの操作で、インデックスの設定変更ダイアログが現れます。
  - ・ メニュー[ファイル]→[変更]→[設定変更]を指定する
  - ・ マウスの右ボタンをクリック、ポップアップメニュー[設定変更]を指定する
- ③ [再編成]タブの再編成モードを選択し、実行ボタンをクリックしてください。

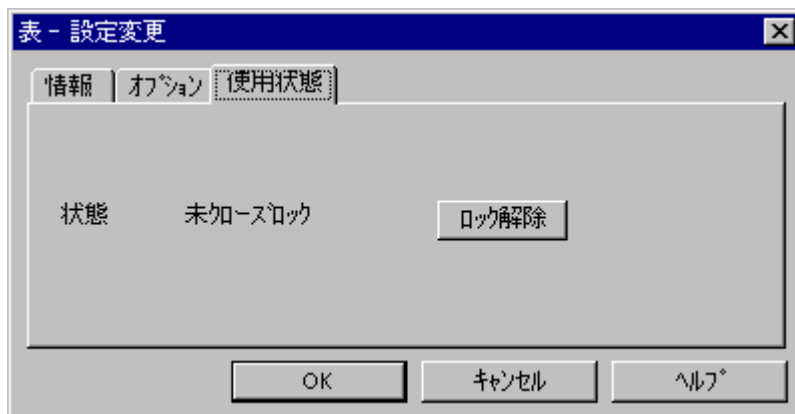


## RDB表のロックリセット

基本表の未クローズロック状態の解除を行います。

### 【基本表のロックリセット方法】

- ① 対象となる基本表をクリックし、基本表名を反転表示させます。
- ② 以下のいずれかの操作で、表の設定変更ダイアログが現れます。
  - ・ メニュー[ファイル]→[変更]→[設定変更]を指定する
  - ・ マウスの右ボタンをクリック、ポップアップメニュー[設定変更]を指定する
- ③ [使用状態]タブの[ロック解除]ボタンが有効の場合、[ロック解除]ボタンを押してください。処理を取りやめる場合は、[キャンセル]ボタンを押します。



## RDB環境のチェック

障害復旧やデータベース環境の移行作業完了後、データベース環境が正しい状態にあるかを再構築機能でチェックすることができます。

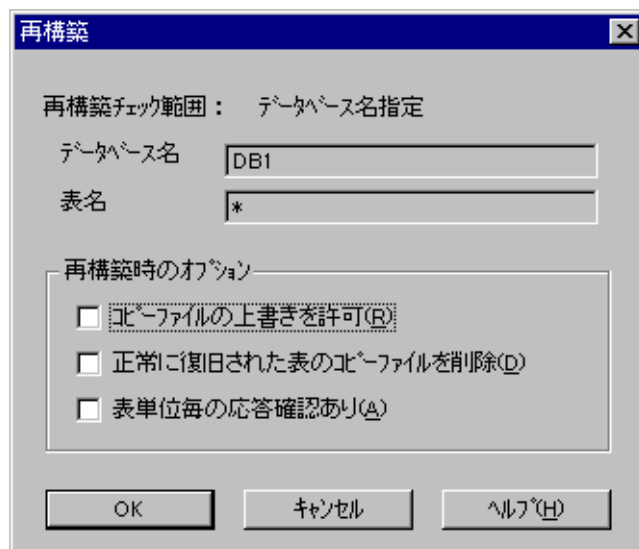
再構築機能とは、管理ファイル上の定義情報とディレクトリや実表(表を構成するファイル群)との整合性をチェックするもので、チェック結果が不整合の場合、定義情報に合ったディレクトリや実表を再構築します。管理ファイルや実表が不正な状態では正常に実行できません。

ただし、**実表内のレコードが正しい列情報を持っているかのチェックは行いません。**

また、再構築後に、既存の実表はコピーファイルとして、“**コピー**”が先頭に付加されたファイル名にて、保存されます。コピーファイルは、不要になった時点で必ず削除するようにしてください。

### 【再構築方法】

- ① 再構築処理を行いたい範囲(管理ディレクトリ、データベース、基本表)をクリックし、範囲名を反転表示させます。
- ② メニュー[環境設定]→[再構築]を指定すると、再構築ダイアログが現れます。再構築が行われる範囲をチェックし、再構築時のオプションを指定してください。再構築処理を行う範囲は、以下のようになっています。
  - ・ 管理ディレクトリを選択－管理ファイル下に定義されている全データベースと全基本表
  - ・ データベースを選択－選択されたデータベースとその配下の全基本表
  - ・ 基本表を選択－選択された基本表のみ
- ③ [OK]ボタンを押すと、チェックと再構築処理が実行されます。再構築結果の確認は、**【再構築の結果を確認】**を参照してください。



なお、再構築時には、以下の処置を指定できます。

- ・ コピーファイルの上書きを許可  
コピーファイルが既に存在するとき、上書きを許可します。  
本指定がなくコピーファイルが既に存在する場合、再構築処理はエラーとなります。  
実行前のコピーファイルが必要な場合、コピーファイルを移動してください。
- ・ 正常に復旧された表のコピーファイルを削除  
「再構築-結果表示」ダイアログ内で「～正常に復旧されました」と表示された表に対するコピーファイルを削除できます。  
エラー時や本指定がないためにコピーファイルを削除しなかった場合、不要になった時点でコピーファイルを削除するようにしてください。
- ・ 表単位毎(ごと)の応答確認あり  
定義情報とディレクトリ/実表が不整合である表毎(ごと)に、再構築を行うかどうかの応答を行います。

### 【再構築の結果を確認】

チェックと再構築が終了すると、再構築の結果を表示するダイアログが表示されます。

チェックの不整合により、再構築処理が行われたデータベース、表に対する処理結果のみを表示します。



## エラー時のメッセージと意味

### 「～復旧されました.(警告あり)」

表の再構築はできましたが、元の実表とおりの復旧はできませんでした。  
そのままで表の使用は可能ですが、警告メッセージを確認してください。

### 「～表の復旧処理をスキップしました。」

チェックにより不整合を検出しましたが、応答により表の復旧処理をスキップしました。

### 「～メモリ不足により処理を中止しました。」

メモリ不足のため、処理を中断しました。  
エラーとなった表の実表を再構築実行前の状態に戻し、再実行してください。  
このとき、該当する表に対するコピーファイルも削除してください。

### 「～定義情報の入手でエラーになりました。」

管理ファイルのアクセスに失敗しました。  
管理ファイルの状態が正常であることをチェックしてください。

### 「～表のチェックでエラーになりました。」

実表のアクセスに失敗しました。  
実表の状態が正常であることをチェックしてください。

### 「～ディレクトリの復旧でエラーになりました。」

ディレクトリの作成ができません。  
指定データベースのディレクトリの作成環境をチェックしてください。

### 「～既存表の保存でエラーになりました。」

実表の保存に失敗しました。

エラーとなった表の実表を再構築実行前の状態に戻し、再実行してください。  
このとき、該当する表に対するコピーファイルも削除してください。

「～表の復旧でエラーになりました。」

実表に対するファイルの再作成ができませんでした。または、元のデータファイルが不正で実表の再構築ができませんでした。  
エラー原因を取り除き、エラーとなった表の実表を再構築実行前の状態に戻し、再実行してください。  
このとき、該当する表に対するコピーファイルも削除してください。

「～インデックスの復旧でエラーになりました。」

インデックスの作成に失敗しました。  
エラーとなったインデックスを定義から削除し、エラーとなった表の実表を再構築実行前の状態に戻し、再実行してください。  
このとき、該当する表に対するコピーファイルも削除してください。

注) 再構築エラー後の定義削除時、実表が存在すると定義の削除ができない場合があります。該当する実表を削除または移動してから、定義の削除を行ってください。  
再構築エラー時、実行前の状態に実表に戻す場合、元の実表が保存前か保存後の状態であるかの確認をしてから削除・ファイル名変更等の処置を行ってください。

## 警告メッセージと意味

「元データの復旧はできませんでした。」

表の再構築はできましたが、元の実表とおりの復旧はできませんでした。  
そのままだでも表の使用は可能ですが、警告メッセージを確認してください。

「次のインデックスが実表に存在しました.: 定義されていないものがあります。」

定義に存在しないインデックスは、再構築対象になりません。  
実表内に存在したインデックス情報が表示されますので、必要なインデックスは定義を追加してください。

<表示されるインデックス情報>

インデックス毎(ごと)に、以下の情報が表示されます。

インデックスキーの数

重複キーの有無

インデックスキー情報

オフセット 実レコードの先頭からのオフセット(各列のデータサイズを順に足した値)

サイズ データサイズ(バイト)

タイプ データタイプをSQLデータ型で表示、IFASPRO RDBで扱われないタイプについては「NONTYPE(nnnn)」(nnnn=IFASデータタイプ)で表示

以下に、インデックス情報の解析例を示します。

以下の結果が表示された場合で、実表のインデックスキーを解析してみます。



1. オフセットが何列目のものに対するものかを判断する。

$$\begin{aligned} \text{3列目のオフセット} &= \text{1列目のデータサイズ} + \text{2列目のデータサイズ} \\ &= (6/2+1) + (5/2+1) = 4 + 3 = 7 = \text{表示情報のオフセット} \end{aligned}$$

2. データサイズとデータタイプも表示情報と一致することによって、実表に定義されていたインデックスは、3列目の情報をキー情報として持つことがわかる。

オフセット計算時に使用する各列のデータサイズとデータタイプの関係は、次表を参照してください。

## IFASPRO RDBでのデータタイプとデータサイズ

COBOL	IFASPRO RDB	
データ型	データタイプ <SQLデータ型>	データサイズ (桁数=n)
単精度固定2進数 COMP-1	SMALLINT	2バイト
倍精度固定2進数 COMP-2	INTEGER	4バイト
符号付き内部10進数 COMP-3	DECIMAL	n/2+1バイト (切り捨て) 注)
符号なし内部10進数 COMP-3		
符号付き2進数 BINARY		
符号なし2進数 BINARY		
符号付き2進数 COMP-5		
符号なし2進数 COMP-5		
符号なし外部10進数	UNSIGNED NUMERIC (UNUNERIC)	nバイト
符号付き外部10進数 TRAILING	NUMERIC	nバイト
符号付き外部10進数 TRAILING SEPARATE		
符号付き外部10進数 LEADING		
符号付き外部10進数 LEADING SEPARATE		
文字 X(n)	CHARACTER (CHAR)	nバイト
文字(2バイト文字固定) N(n)	NATIONAL CHARACTER (NCHAR)	2 * nバイト

注) 運用支援ツールの[COBOL設定]タブで「COMP-5/BINARYの仕様をCOBOL85モードにする」が設定されている場合、データ型COMP-5のデータサイズは必ず以下のようになります。

桁数4のとき      3バイト  
 桁数9のとき      6バイト  
 桁数18のとき     10バイト

## RDB表の各種操作

RDB表に対するデータ操作、移行処理の補助操作を行います。

## データの全件削除

選択した基本表のデータを全(すべ)て削除します。

## 【データの全件削除方法】

- ① データを削除したい基本表をクリックし、表名を反転表示させます。
- ② メニュー[ツール]→[データ全件削除]を指定します。



## データのインポート

CSVファイル、IFAS ファイルから基本表へデータを入力します。  
 CSVファイルでは、固定長CSVファイルを扱うことができます。

### 固定長CSV形式ファイルの仕様

固定長CSVファイルはIFAS FILE、IFASPRO RDBが対応しているCSV形式ファイルの一形態で、そのファイルはリターンコード(¥n/¥r¥n)までを1レコードとして扱い、ヘッダを除く各レコードの長さは基本的には等しいものです。MS-DOS形式テキストファイルと似たような書式ですが、RFDファイルを使用して、ユーザレコード定義とおりの属性でIFAS一般ファイルを作成することができます。

#### ・ヘッダ

固定長CSVファイルのヘッダ(項目名レコード)は各項目名を1バイト空白で区切ったものです。

#### ・数値属性項目

「符号あり」の数値項目に関しては、符号領域として項目の左端に1バイトの領域を必要とします。正数の場合は0を表示し、負数の場合は-(マイナス)を表示します。入力データにもこの領域は必要となります。

COMP-1とCOMP-2は固定サイズとなり、その桁数は以下ようになります。

COMP-1	5桁固定(よって実際は符号バイトと合わせた6桁で表示)
COMP-2	10桁固定(よって実際は符号バイトと合わせた11桁で表示)

COMP-5はRFDファイルへの定義桁数によって以下の固定サイズになります。

1～4桁	5桁固定(「符号あり」の場合は+1桁、小数の場合にはさらに+1桁)
5～9桁	10桁固定(「符号あり」の場合は+1桁、小数の場合にはさらに+1桁)
10～18桁	19桁固定(「符号あり」の場合は+1桁、小数の場合にはさらに+1桁)

#### ・レコード長より短いレコードの処理

オプション指定によりエラー(終了)/パディングが選択できます。パディングキャラクタは項目の属性によって異なります。

数値属性項目	0をパディング
1バイト文字項目	Null文字をパディング
2バイト文字項目	Null文字を2個パディング

#### ・レコード長より長いレコードの処理

オプション指定によりエラー(終了)/切り詰めが選択できます。

#### ・出力時のレコード中にNull文字を検出した場合

以降の項目属性において、数値属性の場合は0、文字属性の場合は1バイト空白、2バイト文字属性の場合は2バイト空白をパディングします。

### 【データのインポート方法】

- ① データを入力したい基本表をクリックし、表名を反転表示させます。
- ② メニュー[ツール]→[データのインポート]を指定すると、インポートダイアログが現れます。
- ③ 入力したいデータを持つCSVファイルやIFAS ファイルの情報、動作を指定してください。入力ファイルがCSVファイルの場合、[オプション]ボタンを押し、オプション情報も指定してください。
- ④ [OK]ボタンでの実行後、エクスポート結果として、正常時には出力データのレコード数が、エラー時にはエラーステータスが表示されます。エラーステータスは、「インポート／エクスポートのコマンド実行」内の「IFCPYCOMのエラーステータス」を参照してください。

## データのエクスポート

RDB表からCSVファイル、IFAS ファイルへデータを出力します。

### 【データのエクスポート方法】

- ① 出力したいデータを持つ表をクリックし、表名を反転表示させます。
- ② メニュー[ツール]→[データのエクスポート]を指定すると、エクスポートダイアログが現れます。
- ③ 出力したいCSVファイルやIFAS ファイルの情報、動作を指定してください。
- ④ [OK]ボタンでの実行後、エクスポート結果として、正常時には出力データのレコード数が、エラー時にはエラーステータスが表示されます。エラーステータスは、「インポート／エクスポートのコマンド実行」内の「IFCPYCOMのエラーステータス」を参照してください。

**エクスポート - RDB表**

指定ファイルの情報

出力ファイル名: B:Temp\kst\従業員.txt 参照

出力ファイルの種類: CSV形式ファイル

ファイル形式

フィールド区切り記号(F): |

テキスト区切り記号(Q): "

☐ 先頭行をフィールド名として使用する(T)

定義と実データ不整合時

☒ 不整合エラーで中止する(E)

☐ 強制的に定義に合わせて実行する(Q)

更新ログオプション

☐ 更新ログを保存する

OK キャンセル ヘルプ(H)

## RFDファイルを利用した表定義

既存のRFDファイルから表定義情報を入力し、表定義/インデックス定義を行う機能をRFD入力機能としてサポートしています。

### 【RFDファイルを利用して表定義する方法】

- ① メニュー[ツール]→[RFD入力]を指定すると、ファイルダイアログが現れます。
- ② 利用したいRFDファイルを選択し、[開く]ボタンを押します。
- ③ RFD入力ダイアログが表示されますので、オプションを指定しRFDファイルより入手した情報で表の定義を行う場合は[OK]ボタンを、取り消したい場合は[キャンセル]ボタンを押してください。このとき、データベース名および表名を変更することもできます。
- ④ 指定データベースが存在しない場合、「データベースの新規作成」と同等のダイアログが表示され、データベースも作成します。ただし、データベース名は再入力できません。
- ⑤ 同名の表が存在する場合、表の再作成を問い合わせるダイアログが表示されます。

なお、②で複数のRFDファイルを選択した場合、③のRFD入力ダイアログは表示されず、RFDファイルを開くと同時に処理が実行されます。このとき、「表オープン時のオプション」は、環境設定で指定された既定値が使用されます。

また、仮想表に対するRFD入力で構成表が存在しない場合、エラーとなります。構成表が存在する状態でRFD入力を行ってください。

**RFD入力-表定義**

データベース名: DB1

表名: 基本表1

表オープン時のオプション

トランザクション処理する ☐ 更新ログを保存する ☐

OK キャンセル

## RDB表のRFDファイル出力

RDB表の表定義情報をRFDファイルへ出力します。

管理ディレクトリ配下、データベース配下、表単体の単位でRFD出力機能を行うことができます。

### 【RDB表のRFDファイル出力方法】

- ① RFDファイルを出力したいアイテム(管理ディレクトリ、データベースまたは表)をクリックし、アイテム名を反転表示させます。
- ② メニュー[ツール]→[RFD出力]を指定します。
- ③ 既にRFDファイルが存在する場合、RFDファイルの再作成を問い合わせるダイアログが表示されます。

なお、RFDファイルは、表が存在するデータベースディレクトリ配下に「表名+“.rfd”」というファイル名で作成されます。

## コマンドラインからの実行

表定義ユーティリティの以下の機能は、DOSプロンプトから実行することができます。

- ・インポート/エクスポート機能
- ・管理ファイルのロック解除機能
- ・インデックスの更新属性変更機能
- ・インデックスの再作成機能
- ・データの全件削除機能

これらの機能は、コマンド実行時のみ、IFASのクライアント・サーバ機能に対応しています。ただし、サーバ名の指定方法が異なる機能があるため、注意してください。

なお、インポート/エクスポート機能のコマンド実行については、実行形式が異なるため、「インポート/エクスポートのコマンド実行」を参照してください。

### 【書式】

IFRDBUTL [/スイッチ]

### 【戻り値】

ユーティリティは実行後に ERRORLEVEL に以下の数値を返します。

0	: 正常
100	: パラメータ指定エラー
101	: 内部エラー(メモリ不足を含む)
102	: IFAS FILEエラー
103	: SQLエラー

### 【スイッチ】

以下にスイッチの種類と指定方法を示します。

なお、複数の選択肢がある場合、{ } で囲み、指定候補は | で分割表示します。

機能によっては指定が必須のスイッチと、省略が可能なものがあります。

スイッチの小文字部分の記述は、省略可能です。

スイッチを複数指定する場合、ブランクで区切ってください。

### ・サーバ名の指定

対象となるサーバ名を指定します。

省略した場合、実行端末に対する処理と認識します。

[指定方法]

/Server=サーバ名

### ・データベース名の指定

対象となるデータベース名を指定します。

省略した場合、データベース名の既定値が使用されます。

[指定方法]

/DB=DB名

### ・表名の指定

対象となる表名を指定します。

[指定方法]

/Table=表名

### ・インデックス名の指定

対象となるインデックス名を指定します。

[指定方法]

/Index=インデックス名

### ・ロック解除機能

何らかの障害が発生し、管理ファイルが未クローズロックになった場合に強制的にロック解除処理を行います。

ただし、管理ファイルのロールバック処理は行われませんので、管理ファイルの整合性がとれなくなる可能性があります。使用には十分注意してください。

[指定方法]

/Recovery

[例] IFRDBUTL /R

### ・インデックスの更新属性変更機能

指定されたインデックスの更新属性を変更します。

インデックス名の指定が必須です。

[指定方法]

/Attr={ SYNC | ASYNC | EXEC }

SYNC	:	インデックスの更新を反映する
ASYNC	:	インデックスの更新を反映しない
EXEC	:	インデックスを実行時生成する

[例] IFRDBUTL /A=SYNC /DB=DB1 /T=TABLE1 /I=INDEX1

DB1.TABLE1のINDEX1を“更新を反映する”に変更する。

### ・インデックスの再作成機能

指定されたインデックスを再作成します。

表名またはインデックス名の指定が必須です。

[指定方法]

/REMake={ AUTO | FORCE | GARBAGE }

AUTO	:	インデックスツリーのチェックを行い、必要に応じて再構成します
------	---	--------------------------------

FORCE : 無条件にインデックスの再構成を行います  
 GARBAGE : データファイルのガベージ処理と、インデックスの再構成を行います

[例] IFRDBUTL /REM=FORCE /DB=DB1 /T=TABLE1  
 DB1.TABLE1に定義されたインデックスを全(すべ)て再作成します。

### ・データの全件削除機能

データファイルからデータを全(すべ)て削除します。  
 表名の指定が必須です。

[指定方法]  
 /Cleardata

[例] IFRDBUTL /C /DB=DB1 /T=TABLE1  
 DB1.TABLE1のデータをすべて削除します。

## インポート／エクスポートのコマンド実行

以下の組み合わせのデータコピーをDOSプロンプトから、パラメータに定義ファイル名を指定して、コマンドラインプログラムとして実行することができます。

RDB表 ↔ RDB表  
 RDB表 ↔ CSV形式ファイル  
 RDB表 ↔ IFAS一般ファイル  
 IFAS一般ファイル ↔ CSV形式ファイル

上記以外の組み合わせはエラーとなります。

本機能は、操作コマンドを記述した定義ファイルをパラメータとしてコンソールコマンドから以下のように実行することができます。

### IFCPYCOM [定義ファイル名] [置換文字列1]・・・[置換文字列9]

パラメータは必ずブランクで区切ってください。ファイル名にブランクが含まれる場合はパラメータをダブルクォーテーション(")で囲んでください。

置換文字列1～9については、「7-5 定義ファイルのパラメータ置換機能」を参照してください。

### 【例】

定義ファイル=SAMPLE.TXTの場合  
 IFCPYCOM SAMPLE.TXT または IFCPYCOM.EXE SAMPLE.TXT

定義ファイル=SAMPLE DEFINE FILEの場合  
 IFCPYCOM "SAMPLE DEFINE FILE"

### 【戻り値】

コマンド実行後に ERRORLEVEL に以下の数値を返します。

0 : 正常終了  
 1 : 強制実行時のワーニング値  
 2～99 : IFAS FILE関数のワーニング戻り値  
 100 : IFAS FILE関数のエラー／コピールーチンのエラー  
 255 : 上記以外エラー

※ERRORLEVEL での判別は 0,1 (正常終了+強制ワーニング)と それ以外(エラー)で行うのが一般的です。

バッチ例

```
IFCPYCOM.EXE A:¥DEFFILE.TXT
IF ERRORLEVEL 2 GOTO ERR      → エラー処理へ
IF ERRORLEVEL 1 GOTO NORMAL   → ワーニングも正常処理へ
IF ERRORLEVEL 0 GOTO NORMAL   → 正常処理へ
```

## 定義ファイルの記述方法

インポート／エクスポートのコマンドライン実行を行うには、定義ファイルを作成しなくてはなりません。定義ファイルはIFCPYCOMを実行するために必要なオプションを定義したテキストファイルです。このファイルは、ユーザがエディタ等を使用して直接作成してください。定義ファイルの中は一回の動作に必要な入出力ファイル形式・入出力ファイル名・機能コード・オプションの設定をもって1つのセクションとし、複数セクション指定することができます。セクションは **[Execute Set]**を先頭に始まります。

空白、タブを除いた行の先頭に“;”がある行はコメント行として読み飛ばされます。

本コマンドはIFASのクライアント・サーバ機能を使用しており、これに対応しています。しかし、クライアント・サーバ機能の対象ファイルはIFAS一般ファイル／RDB表に限定されておりますので、CSV形式ファイルの入出力においてファイルの先頭に〈サーバ名〉を付けてもエラーとなります。

IFAS一般ファイルの入出力時に、同一ディレクトリ上にRFDファイルが存在しない場合、処理はエラーとなります。

ただし、以下の組み合わせの処理でRFDファイルがない場合は、自動的にRFDファイルを作成して処理が行われます。

- RDB表→IFAS相対編成ファイル(ローカルアクセス形態のみ)
- RDB表→IFAS順編成ファイル(ローカルアクセス形態のみ)

定義ファイルの設定コマンドは以下のとおりです。

### 入出力ファイル形式

#### 【書式】

InFileForm=入力ファイル形式 Rdb/Ind/Rel/Seq/Csv のうち、1つを指定(必須)

OtFileForm=出力ファイル形式 Rdb/Ind/Rel/Seq/Csv のうち、1つを指定(必須)

#### 【指定内容】

Rdb	RDB表
Ind	IFAS索引順編成ファイル
Rel	IFAS相対編成ファイル
Seq	IFAS順編成ファイル
Csv	CSV形式ファイル

### 入出力ファイル名

RDB表の場合

#### 【書式】

UtilInFileName=入力RDB表のDB名.表名(必須)

UtilOtFileName=出力RDB表のDB名.表名(必須)

DB名は大文字／小文字の区別あり、表名は大文字／小文字の区別なし

RDB表以外の場合

#### 【書式】

UtilInFileName=入力ファイルのフルパスファイル名(必須)

UtilOtFileName=出力ファイルのフルパスファイル名(必須)

IFAS索引順編成ファイルの場合、データファイルを指定すること

## 機能コード

### 【書式】

FuncCode = Copy/Appendのどちらかを指定(必須)

## オプション

オプションの設定コマンドです(太字は既定値、\*nは注意事項)。

### 【書式】

ErrorCheck=エラーレコード処理

### 【指定内容】

Abort エラーとして処理を中断する  
Skip 属性変更できる場合は強制的にいれ、それ以外の場合はそのレコードをスキップして処理を続行する(属性エラーは終了)

### 【書式】

LogCheck=表の更新ログ出力の有無

### 【指定内容】

Yes 出力する  
No 出力しない

本オプションが省略された場合、指定表のデフォルト状態で実行

### 【書式】

CsvFirstField=CSV形式ファイルの先頭行の処理 (\*1)

### 【指定内容】

Data 先頭行をデータとして処理する  
Header 先頭行をフィールド行として処理する(出力の場合 ヘッダ行を作成)

### 【書式】

CsvLongRecord=ファイル／表の定義より長いレコード処理 (\*3)

### 【指定内容】

Execute 切り捨てる  
Error エラーとして処理を中止する

### 【書式】

CsvShortRecord=ファイル／表の定義より短いレコード処理 (\*3)

### 【指定内容】

Execute Null文字パディングを行う  
Error エラーとして処理を中止する

### 【書式】

CsvFieldCutChar=フィールド区切り文字 (\*1\*2)

### 【指定内容】

Comma カンマ  
Semicolon セミコロン  
Tab タブ  
Blank スペース  
? 任意の1バイトキャラクタ  
Null なし(固定長CSV指定)

### 【書式】

CsvTextCutChar=テキスト区切り文字 (\*1\*4)

### 【指定内容】



Double	ダブルクォーテーション
Single	シングルクォーテーション
Null	なし

## 【書式】

CsvPaddingChar=1バイトパディングキャラクタ (\*1\*4)

## 【指定内容】

Null	Null文字
Blank	スペース
?	任意の1バイトキャラクタ

## 【書式】

CsvPaddingNChar=2バイトパディングキャラクタ (\*1\*4)

## 【指定内容】

Null	Null文字
Blank	スペース(2バイト)
?	任意の2バイトキャラクタ

- \*1 CSV形式ファイルの入出力のみ有効
- \*2 CsvFieldCutChar と CsvTextCutChar が等しい場合はエラーになります
- \*3 固定長CSVの場合のみ有効
- \*4 固定長CSVの場合は無効

## 一括処理

1つの定義ファイルに複数のセクションを記述することで一括処理をすることができます。

## 【書式】

```
[Execute Set]
:
:
[Execute Set 2]
:
:
[Execute Set n]
```

セクションを2つ以上記述する場合、2つ目以降からは番号をふってください。同一番号のセクションが存在する場合は、先に書かれているほうが実行され、後のものは実行されません。

## 定義ファイル作成例

RDB表からCSV形式ファイルにコピーする場合

[Execute Set]	
InFileForm=Rdb	入力 は RDB 表
OtFileForm=Csv	出力 は CSV 形式ファイル
UtilInFileName=DB.TEST	DB 名 = DB、表名 = TEST
UtilOtFileName=A:¥TEST.CSV	フルパスファイル名 = A:¥TEST.CSV
FuncCode=Copy	機能コードはコピー
ErrorCheck=Abort	エラー時は処理中止
LogCheck=No	更新ログの出力しない
CsvFirstField=Data	先頭行からデータを出力
CsvFieldCutChar=Comma	フィールド区切り文字はカンマ
CsvTextCutChar=Double	テキスト区切り文字はダブルクォーテーション
CsvPaddingChar=Null	1バイトパディングキャラクタはNull文字
CsvPaddingNChar=Null	2バイトパディングキャラクタはNull文字

## IFCPYCOMエラー処理

IFCPYCOMコマンドラインで実行中にエラーを検出した場合は、カレントディレクトリにIFCPYCOM.ERR というファイルを作成し、エラーコードと発生したセクション番号を書き出します。エラーを検出した処理に関しては、それ以前の正常に終了したセクションも含めて、使用したファイルのデータの保証はいたしません。また、コマンドラインの戻り値を返しますので、バッチファイルの中で認識することも可能です。

### IFCPYCOMのエラーステータス

エラーコード	意味
41110	入力ファイルのオープンに失敗しました。
41111	入力テキストファイルの読み込みに失敗しました。
41112	出力レコードサイズより長いレコードを検出しました。
41113	出力レコードサイズより短いレコードを検出しました。
41114	出力テキストファイルへの書き出しに失敗しました。
41115	予期しないエラーが発生しました。
41116	アプリケーションの実行に必要な領域がありません。
41117	定義ファイルの内容が不正です。
41118	定義ファイルが存在しません。
41119	定義ファイルのオープンに失敗しました。
41120	領域確保に失敗しました。
41121	定義ファイルにファイル形式指定がありません。
41122	定義ファイルにファイル名指定がありません。
41123	定義ファイルに機能コードがありません。
41125	定義ファイルで指定ファイル形式と入力ファイル形式が一致しません。
41127	定義ファイルのキー指定が不正です。
41128	定義ファイルの保存に失敗しました。
41131	サポートされていない形式の組み合わせでコピーを実行しようとしてしました。
41132	RFDファイルが存在しない、または不正です。
41141	読み込もうとしたファイルは定義ファイルではありません。
41142	テンポラリファイルの作成に失敗しました。
41203	IFAS FILE関数エラーです。 このステータスとともに出力されるIFAS FILEステータスを参照してください。
41204	IFAS FLE関数エラーです。 このステータスとともに出力されるIFAS FILEステータスを参照してください。
41205	指定されたCSV形式ファイルがオープンできません。
41206	メモリ不足です。
41207	RFDの定義が不正です。
41208	フィールド区切り文字とテキスト区切り文字が等しい。
41209	CSV形式が不正です。
41210	入力と出力のカラム定義が合っていません。
41211	RFDファイルの定義内に未対応のデータタイプを検出しました。
41212	IFAS ファイルとカラム定義が合っていません。
41213	IFAS索引順編成ファイルに索引がありません。
41214	RFDファイルが存在しません。
41215	IFAS索引順編成ファイルに索引がありません。
41216	固定長CSV形式ファイルでレコード長より長い入力レコードを検出しました。
41217	固定長CSV形式ファイルでレコード長より短い入力レコードを検出しました。
41280	CSV形式ファイルが存在しない。または、データの出力に失敗しました。
41281	指定されたCSV形式ファイルがオープンできません。
41282	予期しないエラーが発生しました。
41290	定義が合っていません。ワーニングとして処理を継続しました。

これとは別に、IFAS FILEのエラーも返します。詳細は「付録A ステータスコード一覧」を参照してください。

## IFASPRO RDBの予約語一覧

IFASPRO RDBの予約語一覧です。この予約語はデータベース名、表名、列名、インデックス名に使用できません。

ABSOLUTE	ADA	ADBS	ADD
AFTER	AJ	ALL	ALLOCATE
ALTER	AND	ANY	APPEND
ARE	AS	ASC	ASSERTION
AT	AUTHORIZATION	AVG	BASE
BEFORE	BEGIN	BETWEEN	BINARY
BIT	BIT_LENGTH	BJ	BOTH
BY	CASCADE	CASCADED	CASE
CAST	CATALOG	CHAR	CHARACTER
CHARACTER_LENGTH	CHAR_LENGTH	CHEC	CLOSE
CLUSTER	COALESCE	COBOL	COLLATE
COLLATION	COLUMN	COMD	COMMIT
COMMON	COMPOUND	CONNECT	CONNECTION
CONSTRAINT	CONSTRAINTS	CONTINUE	CONVERT
CORRESPONDING	COUNT	CREATE	CURRENT
CURRENT_DATE	CURRENT_TIME	CURSOR	DATABASE
DATE	DAY	DBFILE	DEALLOCATE
DEC	DECIMAL	DECLARE	DEFAULT
DEFERRABLE	DEFERRED	DELETE	DESC
DESCRIBE	DESCRIPTOR	DIAGNOSTICS	DICTIONARY
DIR	DIRECT	DISCONNECT	DISPLACEMENT
DISPLAY	DISTINCT	DOMAIN	DOUBLE
DROP	EFN	ELSE	END
END_EXEC	ESCAPE	EXCEPT	EXCEPTION
EXCLUDE	EXEC	EXECUTE	EXISTS
EXPSIZE	EXTERNAL	EXTRACT	FALSE
FETCH	FIELD	FILE	FIRST
FLOAT	FOR	FORCE	FOREIGN
FORMAT	FORTRAN	FOUND	FRA
FRACTION	FREESPACE	FROM	FULL
GET	GLOBAL	GO	GOTO
GRANT	GROUP	GUEST	HASH
HASHED	HAVING	HOLDMODE	HOURL
ID	IDENTITY	IGNORE	IMMEDIATE
IN	INCLUDE	INDEX	INDEXSTORAGE
INDICATOR	INDIRECT	INITIALLY	INITSIZE
INNER	INPUT	INSENSITIVE	INSERT
INT	INTEGER	INTERSECT	INTERVAL
INTO	IRDS	IS	ISOLATION
JOIN	JOURNAL	KEY	KEYCLUSTER
KEYRANGE	LANGUAGE	LAST	LEFT
LEVEL	LIBRARY	LIKE	LINE
LOCAL	LOWER	MATCH	MAX
MAXRANGENO	MIN	MINUTE	MODULE
MONTH	MUMPS	NAME	NAMES
NATIONAL	NATURAL	NCHAR	NEW
NEXT	NO	NOCASCADE	NONE
NOT	NULL	NULLIF	NUMERIC
OBJECT	OCTET_LENGTH	OF	OFF
OLD	ON	ONLY	OPEN
OPERATION	OPTION	OR	ORDER
OTHERS	OUTER	OUTPUT	OVERLAPS
PARTIAL	PASCAL	PERCI	PF
PLI	POSITION	PRECISION	PREPARE

PRESERVE	PRIMARY	PRIOR	PRIVILEGES
PROCEDURE	PUBLIC	QUADRUPLE	RANDOM
RANGE	RATE	RDB	RDIR
READ	REAL	REFERENCES	REFERENCING
REFFILE	RELATIVE	REPLACE	REPORT
RESIDENT	RESTRICT	RETURN	REVOKE
RIGHT	RIQS	ROLLBACK	ROWS
SCHEMA	SCMOWN	SCROLL	SECOND
SECTION	SELECT	SEQUENCE	SET
SHARE	SHORTNAME	SIMPLE	SIZE
SMALLINT	SOME	SPECIAL	SQL
SQLCODE	SQLERROR	SQLSTATE	STANDARD
STATISTICS	STATUS	STORAGE	STORE
SUBSTRING	SUM	SYNONYM	SYSTEM
TABLE	TEMPORARY	THEN	TIME
TIMESTAMP	TIMEZONE_HOUR	TIMEZONE_MINUTE	TINYINT
TO	TQF	TRANSACTION	TRANSLATE
TRANSLATION	TRIGGER	TRUE	TYPE
UNION	UNIQUE	UNKNOWN	UNSIGNED
UNUMERIC	UPDATE	UPPER	USAGE
USER	USING	UTC	VALUE
VALUES	VARCHAR	VARDEC	VARNCHAR
VARNUM	VARUNUM	VARYING	VIEW
VOLUME	WHEN	WHenever	WHERE
WITH	WORK	WRITE	WRITEPROTECT
YEAR	ZONE		

## 7-3 IFAS ファイルの復旧

IFAS ファイルに障害が発生し、使用できなくなってしまった場合の復旧手段として、IFAS ファイル復旧ユーティリティ群を提供します。

### 7-3-1 ロールフォワードユーティリティ

ロールフォワードユーティリティは、ディスク障害等によりデータが破壊されたIFAS ファイルを障害発生直前の状態までロールフォワード(復旧)するアプリケーションです。

バックアップファイルから復元したIFAS ファイルの最終更新時間以降の更新情報を、更新ログファイルから検索し、復旧を行います。したがって、ロールフォワードユーティリティを使用してIFAS ファイルを復旧する場合、更新情報(更新ログ)の収集を行っていることが前提となります。

更新ログを収集するためには、IFAS ファイルの使用形態により以下の方法があります。詳しくは使用する言語に該当する章を参照してください。

- Cプログラムから使用する場合…IFAS FILEのLOGOPEN/LOGCLOSE関数を使用します。
- COBOLプログラムから使用する場合…COBOLファイルマッパーで定義します。
- IFASカスタムコントロール(参考)から使用する場合…LogModeプロパティを使用します。
- RDB表の場合…表定義ユーティリティで既定値を定義できます。

## 操作方法

### 【起動前の準備】

IFAS ファイルを更新ログ収集開始時点の状態に復元します。IFAS ファイルの復元はバックアップファイルから行います。

IFAS ファイルがクローズ処理未完で正常にオープンできない状態(IFAS FILE関数のステータスコード=1009)の場合には、ファイルロックリセットユーティリティを実行してから、ロールフォワード処理を行ってください。

### 【起動】

IFAS FILEプログラムグループから「ロールフォワード」を選択して起動します。  
以下のようなメニュー画面が表示されます。



メニュー画面の各表示項目は次のようになっています。

- ログファイル名

「復旧手順 1. 更新ログファイルの選択」で指定されたファイル名を表示します。

#### ●データファイル名

IFAS ファイルのデータファイル名を表示します。

[RDB表名で表示]を選択している場合、「データベース名. 表名」という形式でRDB表名を表示します。

#### ●インデックスファイル名

ロールフォワードするIFAS ファイルが索引順編成ファイルの時のみ表示します。

データファイルと対になるIFAS ファイルのインデックスファイル名を表示します。

#### ●状態

ロールフォワード処理の状態をIFAS ファイル単位に表示します。

正常 : ロールフォワード処理が正常に完了しました。

復旧 : ファイルはロールフォワード処理中です。

異常 : エラーが発生しました。

中断 : ロールフォワード処理が中断されました。

待機 : ロールフォワード処理の実行待ちです。

(※IFAS ファイルの選択直後はこの状態です)

#### ●ステータスバー

総数 : ロールフォワード対象となるIFAS ファイルの総数を表示します。

選択 : 選択したIFAS ファイルの数を表示します。

正常 : ロールフォワード処理が正常終了したIFAS ファイルの数を表示します。

異常 : ロールフォワード処理が異常終了したIFAS ファイルの数を表示します。

中断 : ロールフォワード処理の中止を選択した時、ロールフォワード処理が中断されたIFAS ファイルの数を表示します。

### 【復旧手順】

以下の手順で操作してください。

#### 1. 更新ログファイルの選択

ツールバーの[ログファイル]ボタンをクリックするか、または[ファイル]メニューの[ログファイルを開く]を選びます。「ログファイル選択」ダイアログが表示されるので、IFAS ファイルの更新ログを収集している更新ログファイルを指定します。

更新ログファイルは、復旧したいIFAS ファイルの使用法により、次のそれぞれの方法で指定することができます。

#### ●COBOLプログラム／カスタムコントロールなどで使用

「ログファイル選択」ダイアログのコンボボックスから更新ログファイルを選択し、[OK]ボタンをクリックします。

(注) マルチログ運用を行っている場合は、必ずコンボボックスから選択してください。

#### ●Cプログラムで使用

「ログファイル選択」ダイアログの[参照]ボタンをクリックすると、「ログファイルを開く」ダイアログが開きます。ここで、更新ログファイルを選択し、[開く]ボタンをクリックします。

#### 2. ファイル名表示状態の切り替え

RDB表を使用している場合、IFAS ファイル名の一覧を実ファイル名で表示するか、RDB表名で表示するかを切り替えることができます。この操作は指定した更新ログファイルにRDB表の更新が記録されている場合のみ有効となります。

#### ●実ファイル名で表示するには

ツールバーの[ファイル名で表示]ボタンをクリックするか、または[表示]メニューの[ファイル名で表示]を選びます。ロールフォワードユーティリティ起動直後は、この状態です。また、指定し

た更新ログファイルに一般ファイルの更新情報しか記録されていない場合も自動的にこの表示状態となります。

#### ●RDB表名で表示するには

ツールバーの[RDB表名で表示]ボタンをクリックするか、または[表示]メニューの[RDB表名で表示]を選びます。指定した更新ログファイルにRDB表の更新情報が記録されている場合のみ切り替えが可能となります。

この表示状態を選択した場合、以下のような形式で表示されます。

管理ファイル群	: RDBの管理情報が格納されているファイル群(RDB管理ファイル)のことです。
DB(データベース)名. 表名	: RDBの基本表です。
実ファイル名	: RDB表として定義されていない、一般ファイルです。

### 3. ロールフォワード対象ファイルの選択

復旧したいIFAS ファイルを指定します。IFAS ファイルを選択するには、選択した更新ログファイルに記録されているすべてのIFAS ファイルを指定する方法と、復旧したいIFAS ファイルを個別に指定する方法があります。

#### ●IFAS ファイルの全選択

ツールバーの[全選択]ボタンをクリックするか、または[ファイル]メニューの[全てを選択]を選びます。

メニュー画面にロールフォワード対象のIFAS ファイルの一覧が表示され、状態欄の表示が「待機」となります。

#### ●IFAS ファイルの個別選択

ツールバーの[選択]ボタンをクリックするか、または[ファイル]メニューの[選択]を選びます。

「ロールフォワードするファイルを選択」ダイアログが表示され、ロールフォワード対象のIFAS ファイルの一覧が表示されます。

ロールフォワードするIFAS ファイルを選択し、[OK]ボタンでメニュー画面に戻ります。



メニュー画面に戻ると、選択したIFAS ファイルの一覧が表示されます。

メニュー画面上には、選択したファイルの一覧が表示され、状態欄の表示が「待機」となります。



#### 4. ロールフォワードの実行

ツールバーの[実行]ボタンをクリックするか、または[ロールフォワード]メニューの[実行]を選択して、IFAS ファイルのロールフォワード処理を開始してください。

##### ○復旧処理状況

ロールフォワード処理の状況はメイン画面の状態欄およびのステータスバーに表示されます。必要に応じて、ツールバーの[更新]ボタンをクリック、または[表示]メニューの[最新情報に更新]でロールフォワード処理の状況をリフレッシュしてください。

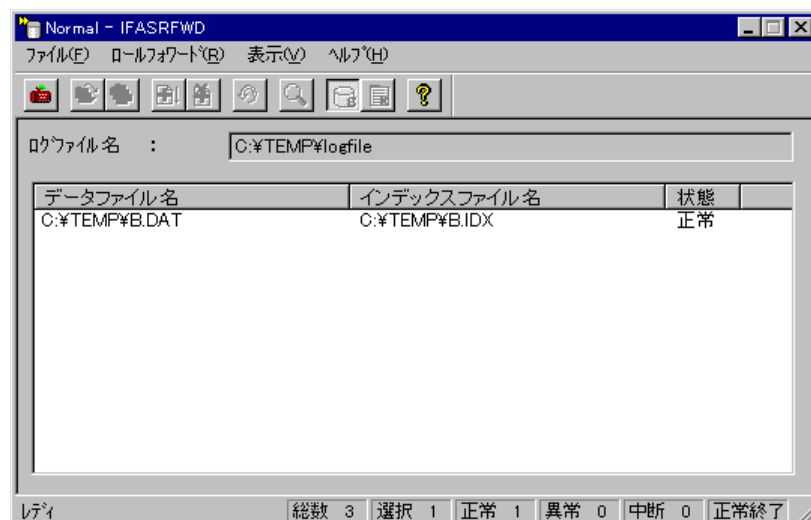
##### ○ロールフォワードの実行中止

ツールバーの[中止]ボタンをクリックするか、または[ロールフォワード]メニューの[中止]を選択すると、IFAS ファイルのロールフォワード処理を中止することができます。この結果、IFAS ファイルの内容は不正な状態となります。ロールフォワード処理を再開する場合は、IFAS ファイルをバックアップファイルから復元し、「復旧手順 1. 更新ログファイルの選択」から行います。

#### 5. ロールフォワード処理の実行結果確認

メニュー画面でロールフォワード処理が正常に終了したIFAS ファイルの状態欄には、「正常」が表示されます。

ステータスバーには、ロールフォワードの実行状態が表示され、指定したIFAS ファイルのロールフォワード処理がすべて正常に終了すると「正常終了」と表示されます。復旧したIFAS ファイルは、ファイル障害が発生する直前の最新コミット時点の状態に復旧されます。





詳細な情報を参照したい場合は、メニュー画面に表示されているIFAS ファイルを選択し、ツールバーの[詳細]ボタンをクリックするか、[表示]メニューの[詳細情報を表示]を選びます。詳細情報には、ロールフォワード処理の対象となった更新ログの開始/終了日時を表示します。IFAS ファイルが最新の状態に復旧されたことの確認の1つとして利用できます。



ロールフォワード処理の対象となる更新ログが見つからなかった場合、IFAS ファイルの状態欄には「正常」が表示され、詳細情報に「ロールフォワードの対象となるログレコードが見つかりませんでした」というメッセージが表示されます。「注意事項」を参照してください。

復旧処理中に異常が発生すると、IFAS ファイルの状態欄には「異常」が表示され、詳細情報にエラー内容が表示されます。IFAS FILE関数でエラーとなった場合はステータスコードも同時に表示されますので、「付録A-3 C言語関数のステータス」を参照し、エラーの原因を取り除いてから、再度ロールフォワードを実行してください。再実行する場合は、IFAS ファイルをバックアップファイルから復元し、「復旧手順 1. 更新ログファイルの選択」から行います。

#### 6. ロールフォワードの終了

[ファイル]メニューの[アプリケーションの終了]、または[コントロールメニューボックス]の[閉じる]を選択し、ロールフォワードユーティリティを終了させます。

別のIFAS ファイルを復旧する場合は、「復旧手順 1. 更新ログファイルの選択」に戻って再度更新ログファイルを指定してください。

## 注意事項

### 1. 「ロールフォワードの対象となるログレコードが見つかりませんでした」

ロールフォワード処理が正常終了した場合、詳細情報で上記のようなメッセージが表示されることがあります。このメッセージが表示された場合、次のようなことが考えられます。IFAS ファイルの状態を確認してください。

- バックアップから現在まで、IFAS ファイルの更新が行われていない  
→IFAS ファイルは最新の状態です。
- IFAS ファイルのバックアップからの復元を行っていない
- ロールフォワード処理を一度中断したが、再開する前にバックアップからの復元を行っていない
- 前回のバックアップ以降に日付/時刻/タイムゾーンの設定が変更された

### 2. 日時/時刻/タイムゾーンの設定変更に対する影響

ロールフォワードは、IFAS ファイルの最終更新時間を元に対象となる更新ログを検索し、復旧処理を行います。IFAS ファイルのバックアップ以降にシステムの時間設定を変更すると、更新ログファイルに記録された時刻とファイルの最終更新時間にずれが生じ、最新の状態に戻すことができなくなってしまうおそれがあります。

システムの時間設定を変更した場合、更新ログの収集を行っているIFAS ファイルのバックアップを取るようにしてください。

### 3. ログ消失した更新ログファイルのロールフォワード

マルチログで運用している場合、更新ログファイルが1サイクルする前にバックアップを取らないと、ログが消失してしまいます。このような更新ログファイルを用いてロールフォワード処理を行うと、ロールフォワード処理自体は正常終了しますが、ログが消失した部分については復旧されません。

#### 4. 非同時更新属性のインデックスが存在する索引順編成ファイルの復旧

一般索引順編成ファイルに非同時更新属性のインデックスが存在する場合、ロールフォワード処理を行うと、そのIFAS ファイルのインデックスは不整合状態になります。

復旧後、インデックスメンテナンスユーティリティでインデックスの再編成を行ってください。

## 7-3-2 ディファードリカバリユーティリティ

ディファードリカバリユーティリティは、システムダウンまたはユーザAPの異常終了等により未クローズとなったIFAS ファイルを最終更新時点までロールバック(復旧処理)するアプリケーションです。本アプリケーションを使用したIFAS ファイルの復旧は、トランザクション処理を行っていることが前提となります。ディファードリカバリユーティリティは、次のような機能を持っています。

#### リスタートリカバリ機能

システムの再起動時にIFAS FILEイニシエータより起動され、トランザクション処理を行っていた未クローズ状態のIFAS ファイルを、コミットされた最終更新の状態まで自動的にロールバックします。

したがって、アプリケーションとしての操作は行う必要がありません。

#### ディファードリカバリ機能

リスタートリカバリに失敗したIFAS ファイルの復旧処理を行います。

IFAS FILEプログラムグループから起動して、未クローズ状態のIFAS ファイルの一覧から、ユーザがファイルを選択して、復旧処理を行います。

## 操作方法

### 【起動】

IFAS FILEプログラムグループから「ディファードリカバリ」を選択して起動します。

以下のようなメニュー画面が表示されます。



メニュー画面の各表示項目は次のようになっています。

#### ●データファイル名

リカバリするIFAS ファイルのデータファイル名を表示します。RDB表を使用している場合でも、すべて実ファイル名の表示となります。

#### ●インデックスファイル名

リカバリするIFAS ファイルが索引順編成ファイルの時のみ表示します。  
データファイルと対になるIFAS ファイルのインデックスファイル名を表示します。

#### ●状態

復旧処理の状態をIFAS ファイル単位に表示します。

- 正常 : IFAS ファイルは使用できる状態です。
- 復旧 : IFAS ファイルは復旧処理中です。
- 異常 : エラーが発生しました。(イベントログにエラー情報が記録されます)
- 中断 : 復旧処理が中断されました。
- 削除 : IFAS ファイルが存在しません。(削除された可能性があります)
- 待機 : 復旧処理の実行待ちです。(※IFAS ファイルの選択直後はこの状態です)

#### ●ステータスバー

- 総数 : 復旧対象となるIFAS ファイルの総数を表示します。
- 選択 : 選択したIFAS ファイルの数を表示します。
- 正常 : 復旧処理が正常終了したIFAS ファイルの数を表示します。
- 復旧 : IFAS ファイルは復旧処理中です。
- 異常 : 復旧処理が異常終了したIFAS ファイルの数を表示します。
- 中断 : 復旧処理の中止を選択した時、復旧処理を実行していないIFAS ファイルの数を表示します。
- 削除 : 復旧処理対象のIFAS ファイルのうち、存在しないファイルの数を表示します。

### 【復旧手順】

以下の手順で操作してください。

#### 1. リカバリ対象ファイルの選択

復旧したいIFAS ファイルを指定します。IFAS ファイルの指定方法には、リスタートリカバリが失敗したIFAS ファイルをすべて指定する方法と、復旧したいIFAS ファイルを個別に指定する方法があります。

#### ●IFAS ファイルの全選択

ツールバーの[全選択]ボタンをクリックするか、または[ファイル]メニューの[全てを選択]を選びます。  
メニュー画面に復旧が必要なIFAS ファイルの一覧が表示され、状態欄の表示が「待機」となります。

#### ●IFAS ファイルの個別選択

ツールバーの[選択]ボタンをクリックするか、または[ファイル]メニューの[選択]を選びます。  
「リカバリするファイルを選択」ダイアログが表示され、リカバリが必要なIFAS ファイルの一覧が表示されます。  
リカバリするIFAS ファイルを選択し、[OK]ボタンでメニュー画面に戻ります。



メニュー画面に戻ると、選択したIFAS ファイルの一覧が表示されます。  
メニュー画面上には、選択したファイルの一覧が表示され、状態欄の表示が「待機」となります。



## 2. ディファードリカバリの実行

ツールバーの[実行]ボタンをクリックするか、または[リカバリ]メニューの[実行]を選択して、IFAS ファイルの復旧処理を開始してください。

### ○復旧処理状況

復旧処理の状況はメイン画面の状態欄およびのステータスバーに表示されます。必要に応じて、ツールバーの[更新]ボタンをクリック、または[表示]メニューの[最新情報に更新]で復旧処理の状況をリフレッシュしてください。

### ○ディファードリカバリの実行中止

ツールバーの[中止]ボタンをクリックするか、または[リカバリ]メニューの[中止]を選択すると、IFAS ファイルの復旧処理を中止することができます。

この結果、IFAS ファイルの内容は不正な状態のままとなります。復旧処理を再開する場合は、ディファードリカバリユーティリティを再起動してください。

### 3. ディファードリカバリ処理の実行結果確認

メニュー画面で復旧処理が正常に終了したIFAS ファイルの状態欄には、「正常」が表示されます。

ステータスバーには、ディファードリカバリの実行状態が表示され、指定したIFAS ファイルの復旧処理がすべて正常に終了すると「正常終了」と表示されます。復旧したIFAS ファイルは、システム障害が発生する直前の最新コミット時点(静止点)の状態に復旧されます。



### 4. ディファードリカバリの終了

[ファイル]メニューの[アプリケーションの終了]、または[コントロールメニューボックス]の[閉じる]を選択し、ディファードリカバリユーティリティを終了させます。

## 注意事項

### 1. IFAS ファイルへのアクセス権

ディファードリカバリユーティリティは、復旧処理対象となるIFAS ファイルへのアクセス権がない場合、復旧処理に失敗します。

ディファードリカバリユーティリティは、システム起動時にIFAS FILEイニシエータからリスタートリカバリ機能として自動的に起動され、サービスのアカウントで実行されます。したがって、IFAS FILEイニシエータに対し、必ずログオンアカウントを設定してください(詳細は「2-1 イニシエータ」を参照してください)。また、IFAS ファイルには設定したアカウントへの書き込みのアクセス権を設定するようにしてください。

### 2. IFAS ファイルの削除

復旧処理に失敗したIFAS ファイルを、<ファイルマネージャ>等から削除しないでください。復旧処理が必要だという情報だけが残し、以後ディファードリカバリユーティリティが異常終了するようになります。

IFAS ファイルの削除にはファイル管理ユーティリティを使用してください。

## 7-3-3 ファイルロックリセットユーティリティ

IFAS ファイルを更新中のプログラムが動作中、システムが電源ダウン等で停止してしまった場合、プログラムからのデータ更新がディスクに反映されていない可能性があるため、そのIFAS ファイルは論理的に矛盾した状態(ファイルラベルの管理情報と実レコード数との不一致等)になっている可能性があります。そのためIFAS FILEはシステム再起動時に前回のシステム停止前にクローズされていなかったフ

ファイルにロックをかけて通常のアプリケーションからはオープンできないようにしています。(ただしROLL BACKオプションを指定してオープンしていたファイルはシステム再起動時に自動的にリカバリするのでロックはかかりません)

この未クローズファイルのロックを解除するためのツールとしてファイルロックリセットユーティリティを提供します。

★注意★

ロックリセットすることによって、ファイル管理情報と実レコード数の矛盾は解消されますが、システムダウン直前までのデータ更新がファイルに反映されるようになるわけではありません。

## 操作方法

### 【起動】

ファイルロックリセットユーティリティを起動すると以下のような画面が開きます。



### 【実行手順】

ファイル名を選択して[実行]ボタンをクリックします。

IFAS相対編成、IFAS順編成ファイルの場合はデータファイル名のみ指定してください。

## コマンドラインからの実行

ファイルロックリセットユーティリティは、未クローズファイルのロックを解除したいファイル名をパラメータとしてコンソールコマンドから以下のように実行することができます。

- ・ IFAS索引順編成ファイルの場合  
**IFASFRST [データファイル名] [インデックスファイル名]**
- ・ IFAS相対編成ファイル、IFAS順編成ファイルの場合  
**IFASFRST [ファイル名]**

パラメータは必ずブランクで区切ってください。ファイル名にブランクが含まれる場合はパラメータをダブルクォーテーション(")で囲んでください。

実行結果はコマンドラインの戻り値としてステータスが返りますのでバッチファイルの中で認識することも可能です。

## 複数ファイルの一括ロックリセット

複数の未クローズロック状態のファイルを一括してロックリセットするためには、前述したコマンドをバッチファイルに複数行記述して実行する方法もありますが、ファイル管理ユーティリティを利用するとファイル一覧表示上でどのファイルが未クローズロック状態かが確認可能な上、複数のファイルをマウスで選択して(または無条件に未クローズロック状態のファイルすべて)一括してロックリセットすることが可能です。

詳細は「7-1-1 ファイル管理ユーティリティ」を参照してください。

## RDB表のロックリセット

未クローズ状態のIFASRDB表のロックリセットは表定義ユーティリティで行うことも可能です。  
詳細は「7-2-1 表定義ユーティリティ」を参照してください。

## 7-4 運用管理

### 7-4-1 運用支援ツール

IFAS FILE、IFASPRO RDBではロックテーブルの初期サイズ、バッファサイズ等のシステム毎の資源や、COBOLアプリケーションのオプションのようなアプリケーション毎のIFAS FILE、IFASPRO RDBの設定をユーザが容易に行えるように運用支援ツール(IFASSYSG.EXE)を用意しています。

#### 機能概要

運用支援ツールには以下の項目のタブがあります。

##### ロック制御

ロック制御テーブルの設定をします。

##### バッファ制御

最大共有バッファのサイズを設定します。

##### ロールバックログ制御(サーバ版のみ)

ロールバックログ出力ディレクトリを設定します。

##### DRS制御

ファイル識別子管理ファイル名および格納ディレクトリを設定します。

##### COBOL設定

COBOLオプションを設定します。

##### 更新ログ設定

COBOL／カスタムコントロールで使用する更新ログファイルに関する設定をします。

##### その他

自動ロックリセット、自動C／S設定、エラーログファイルの最大サイズを設定します。

#### “ロック制御”タブの設定パラメータ

ロック制御の設定を変更するには簡易設定、詳細設定の2通りの設定方法が存在します。

##### (1) 簡易設定

以下の①～③のパラメータの値を変更することにより、ロックテーブル初期サイズをシステムの利用状況に合った大きさで確保することができます。IFAS FILE、IFASPRO RDBは、各パラメータの値を基に数種類あるロックテーブルのそれぞれのサイズを決定します。④、⑤は特に設定する必要はありませんが、状況に応じて設定してください。

なお、①から③はロックテーブル初期サイズを設定し易くするために便宜的に設けたものであり、実際には、あるパラメータ項目が設定値をオーバーしてもロックテーブルが必ず拡張するとはかぎりません。逆に、パラメータ項目が設定値より少なくとも拡張する時があります。

##### ① 同時実行スレッド数(正整数、初期値:10)

システムで同時に実行され得る(IFAS ファイルを処理する)スレッド数の最大値です。同時に実行される可能性のあるスレッド数(最大)を指定してください。

##### ② 1スレッドの同時オープンファイル数(正整数、初期値\*1:10)

1スレッドで同時にオープンされ得るファイル数の最大値(のシステム内の全スレッドの平均値)です。

##### ③ 1ファイルの同時ロックレコード数(正整数、初期値\*2:400)

ファイルあたりの同時ロックレコード数(平均値)です。レコードロック指定の場合、ロックを解放する関数(CLOSE関数、COMMIT関数等)を実行しないかぎりロックは蓄積され続け、その分だけロックテーブルを消費します。また、レコードロック指定の場合はレコードだけではなく索引ブロックもロックしますので、索引を使用する処理の場合には索引レベル数も本パラメータ設定時に考慮してください。

##### ④ 最大使用可能メモリを指定する(正整数、初期値:指定なし)



自動拡張の制限値です。システム全体で利用できるテーブルサイズをMB単位で示します。初期値の使用メモリサイズがこの値を超えている場合は使用メモリサイズ分確保されます。このパラメータを設定しない場合は、システムの仮想メモリサイズまで拡張されます。

⑤ ロック待ち制限時間を指定する(正整数、初期値:指定なし)

既に排他ロックされている資源に対する、待ち時間の最大値を秒単位で指定します。このパラメータを設定しない場合は、ロックが解けるまで永久の待ちとなります。

\*1 IFASPRO RDBの場合は512です。

\*2 IFASPRO RDBの場合は1000です。

## (2) 詳細設定

テーブル数、待ち時間指定など詳細に設定します。ロック制御の構造に関する高度な知識を必要としますので、通常は使用しないでください。

## ロック制御設定の変更方法

以下にロックテーブルサイズを変更する方法を述べます。なお、ロックテーブルサイズの変更は簡易設定と詳細設定の2つの方法がありますが、ここでは通常使用する簡易設定についてのみ説明します。

- (1) “運用支援ツール: **IFASSYSQ.EXE**”を実行します。
- (2) 「運用支援ツール」ダイアログボックスが開きますので“**ロック制御**”タブを選択してください。
- (3) 必要な項目を変更してください。  
ここで変更した値を初期値(下記ダイアログの値)に戻したい場合は“**初期値**”ボタンを押してください。
- (4) 次に“**適用**”ボタンを選択するとロックテーブルサイズの変更を確定します。“**OK**”ボタンを選択するとロックテーブルサイズの変更を確定後、プログラムを終了します。必要な項目を変更してください。
- (5) IFAS FILE、IFASPRO RDBを使用しているユーザプログラムの終了を待って、IFAS FILEイニシエータによるシステムの再初期化を行ってください。IFAS FILEイニシエータの詳細については、「**2-1 イニシエータ**」を参照してください。

運用支援ツール

DRS制御 | COBOL設定 | 更新ログ設定 | その他

ロック制御 | バッファ制御 | ロールバックログ制御

設定方法: ☒ 簡易設定 ☐ 詳細設定

使用可能メモリ: 141988 KB

同時実行スロット数	10	=>	10
1スロットの同時オープンファイル数	10	=>	10
1スロットの同時ロックコート数	400	=>	400
使用メモリ(KB)	420	=>	420

最大使用可能メモリを指定する ☐

ロック待ち制限時間を指定する ☐

説明

**注** 簡易設定で設定された数値は詳細設定時に引き継がれます(各設定値の名称は違っていますが設定値は簡易設定での値をもとに計算されたものです)。しかし、詳細設定を行うと設定値を簡易設定に引き継ぐことはできません。詳細設定の後に簡易設定を行うと詳細設定の前に簡易設定で設定していた値が表示されます。

## “バッファ制御”タブの設定パラメータ

### 最大バッファサイズの見積もり

最大共有バッファサイズは以下に示す式によって、大まかに見積もることができます。

最大共有バッファサイズ(Mバイト) =

$$(((Bis \times Bin) + (Bds \times Bsn)) \times Son \times Sen) + 999999) \div 10^6$$

**Bis** : オープン時に指定する平均インデックスブロックサイズ(バイト)

**Bin** : オープン時に指定する平均インデックスバッファ数(個)

**Bds** : オープン時に指定する平均レコードサイズ(バイト)

**Bdn** : オープン時に指定する平均データファイルバッファ数(個)

**Son** : 1スレッドあたりの同時オープンファイル数(個)

**Sen** : 同時実行スレッド数(個)

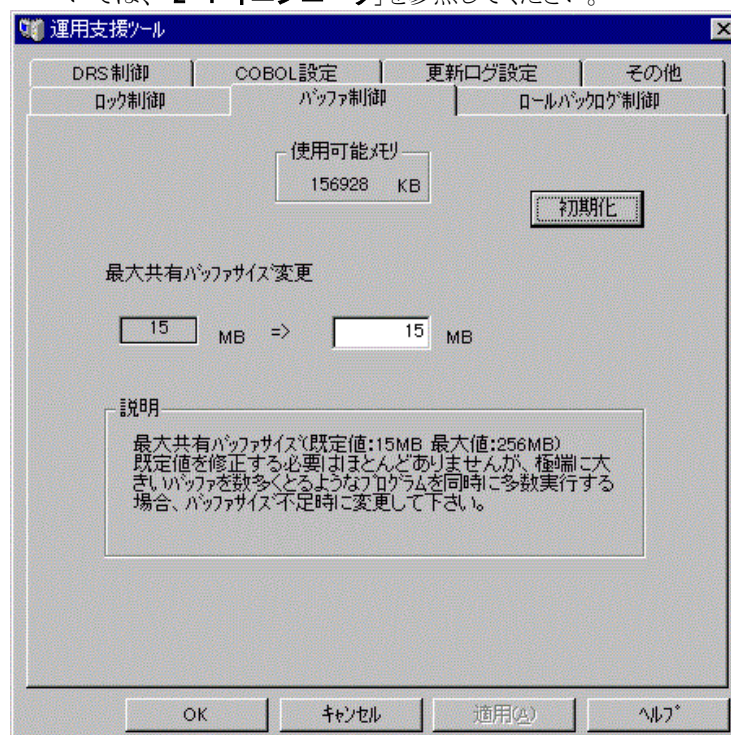
最大共有バッファサイズの上限值を変更する場合は、本式によって見積もった値より、多少大きめの値を指定するようにしてください。

なお、最大共有バッファサイズはMバイト単位で1～256Mバイト(既定値:15Mバイト)まで指定できます。

## 最大共有バッファサイズの変更方法

以下に最大共有バッファのサイズの変更方法を述べます。

- (1) “運用支援ツール:IFASSYS.G.EXE”を実行します。
- (2) 「運用支援ツール」ダイアログボックスが開きますので“**バッファ制御**”タブを選択します。
- (3) ダイアログボックス内の最大共有バッファサイズに変更したい値を入力します。この時、最大共有バッファサイズを既定値(15Mバイト)に戻したい場合は、“**初期化**”ボタンを選択します。
- (4) 次に“**適用**”ボタンを選択すると最大共有バッファサイズの変更を確定し、“**OK**”ボタンを選択すると共有バッファサイズの変更を確定後、プログラムを終了します。
- (5) IFAS FILE、IFASPRO RDBを使用しているユーザプログラムの終了を待って、IFAS FILEイニシエータによるシステムの再初期化を行ってください。IFAS FILEイニシエータの詳細については、「**2-1 イニシエータ**」を参照してください。



## “ロールバックログ制御”タブの設定項目

設定パラメータは以下のとおりです。通常は、本設定は変更しないでください。

### ① サーバ名とディレクトリパス

ロールバックログを収集するサーバ名と、どのディレクトリパスにロールバックログを出力するかを指定してください。

ただし、サーバ名として以下の名前は指定できません。

RollbackLogDirectoryName

RestartSequenceNumber

UseServerDirectory

## ロールバックログ設定の変更方法

以下にロールバックログ設定の変更方法を述べます。

- (1) “運用支援ツール: IFASSYSQ.EXE”を実行します。
- (2) 「運用支援ツール」ダイアログボックスが開きますので“**ロールバックログ制御**”タブを選択します。
- (3) 該当する項目を設定します。  
ロールバックログを収集するサーバ名と、どのディレクトリパスにロールバックログを出力するかを入力して、“**更新・追加**”ボタンを押してください。  
※“**ロールバックログを指定ディレクトリ下に出力する**”チェックボックスはチェックしないでください。  
起動時、もしくは起動後に最後に更新を行った設定に戻りたい場合は“**適用時に戻す**”ボタンを押してください。
- (4) 次に“**適用**”ボタンを選択するとロールバックログ設定の変更を確定し、“**OK**”ボタンを選択するとロールバックログ設定の変更を確定後、プログラムを終了します。





## “DRS制御”タブの設定項目

設定パラメータは以下のとおりです。

- ドライブ名とDRSファイル名  
IFAS ファイルを格納する切替ディスクのドライブ名と、対応するDRSファイル名を指定してください。
- デフォルトDRSファイル  
上記で設定した切り替えディスク以外のドライブに作られるIFAS ファイルに対応するDRSファイルを表示しています。通常は変更の必要はありませんが、変更する場合は“**デフォルトDRSの変更**”ボタンを押してから設定してください。

## DRS設定の変更方法

以下にDRS設定の変更方法を述べます。

- (1) “運用支援ツール:IFASSYS.G.EXE”を実行します。
- (2) 「運用支援ツール」ダイアログボックスが開きますので“**DRS制御**”タブを選択します。
- (3) 該当する項目を設定します。  
切替ディスクであるドライブ名と、DRSファイル名を入力して、“**更新・追加**”ボタンを押してください。  
デフォルトDRSファイルを変更する場合は“**デフォルトDRSの変更**”ボタンを押してからデフォルトDRS入力欄に設定してください。  
起動時、もしくは起動後に最後に更新を行った設定に戻りたい場合は“**適用時に戻す**”ボタンを押してください。
- (4) 次に“**適用**”ボタンを選択するとファイル識別子管理ファイル設定の変更を確定し、“**OK**”ボタンを選択するとファイル識別子管理ファイル設定の変更を確定後、プログラムを終了します。

## “COBOL設定”タブの設定パラメータ

IFAS一般ファイルおよびIFAS RDB表をCOBOLで使用する時に、本プログラムを使用してオプションを設定することができます。

設定パラメータは以下のとおりです。各項目の内容は「4-2-2 動作環境の設定」、「5-2-1 動作環境の設定」を参照してください

### ① OS互換モード

COBOLにおける動作モードをA-VX II 互換とするかA-VX5互換とするか選択可能です。OS互換モードはCOBOL関連プロダクトの共通オプションです。DBリンクキット等にも影響があります。OS互換モードによってCOBOLだけでなく、カスタムコントロール、smart/cs、ファイルメンテナンスキットの動作も変更される可能性があります。IFAS FILEのインストール時の初期設定はA-VX II 互換です。

### ② OPEN命令でレコードサイズのチェックの有無

OPEN命令で、フラグがOFFならば指定レコードサイズはIFAS ファイル(表)のレコードサイズと一致していなければエラーとなり、ONならば指定レコードにはIFAS ファイル(表)のレコードサイズ以上の値を指定しなければエラーとなります。

なお、どちらの場合も、OUTPUT以外のモードでは指定レコードサイズには0を指定することが可能です。既定値はOFFです。

### ③ READ命令での二重キー検出の有無

READ命令で重複キー許可インデックスに二重キー発生時、02(二重キー)ステータスが返却されるか、00(正常)ステータスが返却されるかを設定します。既定値では02(二重キー)ステータスが返却されます。

### ④ WRITE/REWRITE命令での二重キー検出の有無

WRITE/REWRITE命令で重複キー許可インデックスに二重キー発生時、02(二重キー)ステータスが返却されるか、00(正常)ステータスが返却されるかを設定します。既定値では02(二重キー)ステータスが返却されます。

### ⑤ REWRITE命令で変更するキーのみ二重キー検出するかの有無

REWRITE命令で重複キー許可インデックスがある場合、フラグがONならば変更のあったインデックスのみ二重キーのチェックを行い、OFFならばすべてのインデックスに対して二重キーのチェックを行います(変更のないキーに対しても二重キーのチェックを行う)。

どちらの場合も、二重キーが検出された場合は、02ステータスが返却されます。

既定値はOFFです。

### ⑥ INPUTモードのREAD命令でのレコードロックの有無

INPUTモードのREAD命令で、レコードはロックをするかを設定します。既定値では、レコードロックはされません。

本設定をレコードロックありとしてもREAD命令にWITH UNLOCKを指定した場合には、レコードはロックされません。

### ⑦ INPUTモードのSTART命令でのレコードロックの有無(A-VX II 互換モードのみ)

INPUTモードのSTART命令で、レコードはロックをするかを指定します。既定値ではレコードはロックはされません。

A-VX5互換モードではSTART命令でレコードがロックされないので本設定はできないようになります。

### ⑧ 索引順編成ファイルのREWRITE/DELETEをカレントレコード対象にするか否か

主キーに2重キーがある一般索引順編成ファイルを動的呼び出しでオープンしている場合、フラグがONだとREWRITE/DELETE命令を以下の条件でカレントレコードを対象として処理を行います。

- ・直前の命令がREAD命令

- ・処理要求する主キー値がカレントレコードと一致する

フラグがOFFの場合は、REWRITE/DELETE命令を要求する主キー値で検索し、最初に検出されたレコードが処理対象となります。既定値はOFFです。

### ⑨ ファイル/表削除時のRFDファイル削除

COBOL関連製品でファイル/表を削除する場合、フラグがOFFならば、RFDファイルを削除しません。フラグがONならば、RFDファイルを削除します。

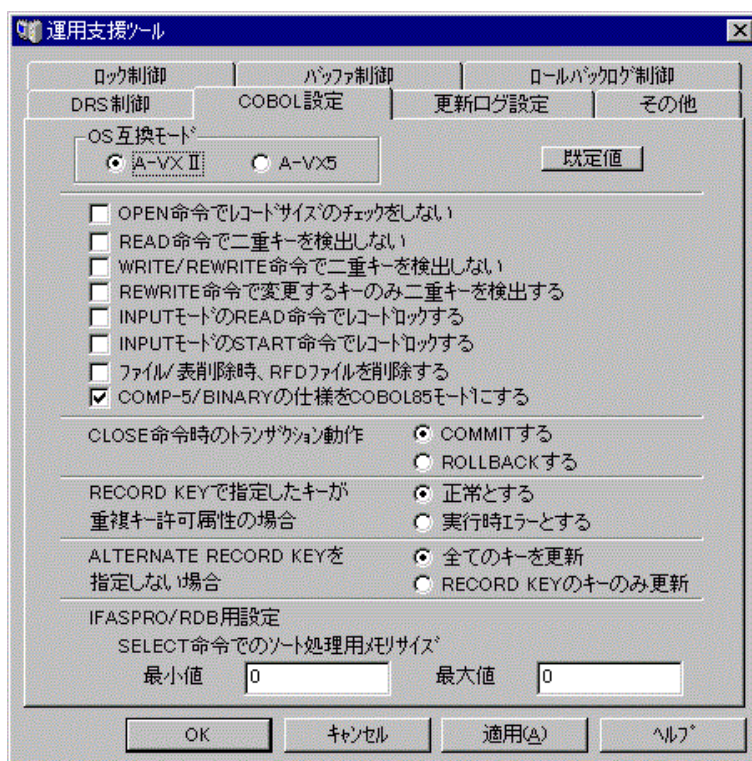
なお、ファイル管理ユーティリティで、ファイルを削除する場合は、RFDファイルは削除されません。また、表定義ユーティリティで表を削除する場合は、表定義ユーティリティのオプションに従います。既定値はOFFです。

- ⑩ COMP-5/BINARYの仕様をCOBOL85モードにするか否か  
ファイル/表のカラムにCOMP-5/BINARY指定があった場合、フラグがONならば、言語仕様をCOBOL85に合わせます。  
フラグがOFFの場合は、IFAS FILE、IFASPRO RDBの独自仕様にします。OFFにした場合、COBOL85から該当ファイル/表を使用することはできません。  
既定値はONです。
- ⑪ CLOSE命令時のトランザクション動作  
トランザクション処理(手動COMMIT)を行っている場合に、「COMMITする」ならばCLOSE命令で暗黙的にCOMMIT処理を行い、「ROLLBACKする」ならばCLOSE命令で暗黙的にROLLBACK処理を行います。既定値は「COMMITする」です。  
なお、プログラム終了時にオープンされたままのファイルは自動的にクローズされますが、トランザクション処理中のファイルは、本オプションの設定にかかわらず、自動的にロールバックされてからクローズされます。
- ⑫ RECORD KEYで指定したキーが重複キー許可属性時のエラー検出  
COBOLでMIX(複数索引)ファイルをOPENする時にRECORD KEYで指定したインデックスに重複キー許可属性がある場合に、正常とするか実行時エラーとするかを設定します。既定値は「正常とする」です。なお、本フラグはCOBOLにおいてのみ有効です。
- ⑬ ALTERNATE RECORD KEYを指定しない時の更新キー  
COBOLでMIX(複数索引)ファイルをOPENする時にALTERNATE RECORD KEYを指定しなかった場合に、すべてのキーを更新するかRECORD KEYの指定キーのみ更新するかを設定します。既定値は「全てのキーを更新」です。  
本設定を「RECORD KEYのキーのみ更新」にした場合、ALTERNATE RECORD KEYが指定されていないCOBOLアプリケーションでMIXファイルを更新するとRECORD KEYで指定したインデックス以外がすべて不整合状態となります。この場合はインデックスメンテナンスユーティリティで再編成が必要となります。なお、本フラグはCOBOLにおいてのみ有効です。
- ⑭ SELECT命令でのソート処理メモリサイズ  
IFAS RDB表に対するSELECT命令で確保するメモリサイズをKBで指定します。  
最小値はSELECT命令で確保する最小メモリサイズを指定します。最小分のメモリが確保できなかった場合、SELECT命令はエラーとなります。  
最大値はSELECT命令で確保する最大メモリサイズを指定します。最大分のメモリが確保できなかった場合は、サイズを小さくしてメモリを確保します。  
最小値/最大値が0の場合、以下の表のレコード数をもとにメモリを自動取得します。  
基本表の場合→基本表  
選択型仮想表→選択型仮想表を構成する基本表  
結合型仮想表→プライマリ表  
合併型仮想表→第一プライマリ表  
既定値は、最小値=0、最大値=0です。

## COBOL設定の変更方法

以下にCOBOL設定を変更する方法を述べます。

- (1) “運用支援ツール: IFASSYSQ.EXE”を実行します。
- (2) 「運用支援ツール」ダイアログボックスが開きますので“**COBOL設定**”タブを選択します。
- (3) 該当する項目を設定します。  
ここで変更した設定を既定値に戻したい場合は“**既定値**”ボタンを押してください。
- (4) 次に“**適用**”ボタンを選択するとCOBOL設定の変更を確定し、“**OK**”ボタンを選択するとCOBOL設定の変更を確定後、プログラムを終了します。



## “更新ログ設定”タブの設定項目

設定パラメータは以下のとおりです。

- ・ 更新ログファイルパス  
更新ログを出力するファイル名をフルパスで指定してください。
- ・ 更新ログファイル数  
更新ログファイル数を指定します。2以上を指定した場合、マルチ更新ログファイル指定となります。
- ・ 更新ログファイルサイズ  
更新ログファイルサイズをMB単位で指定します。1以上を指定した場合、マルチ更新ログファイル指定となります。
- ※ マルチ更新ログは、ファイル名に拡張子がなく、ログファイル数が2以上で、ログファイルサイズが1以上指定された場合に有効となります。いずれかの条件が満たされなかった場合には、設定更新時に警告メッセージボックスが開きます。
- ・ ファイル切り替え情報  
更新ログ設定をマルチログにて運用時、更新ログファイルが更新ログファイルサイズに達して次の更新ログファイルに切り替わった場合、履歴情報をイベントログ／テキストファイルへ出力します。なお、更新ログファイルをリセットした場合、リセット情報をイベントログ／テキストファイルへ出力します。  
「テキストファイルに出力」を指定した場合、ファイル切り替え情報を出力するテキストファイルのファイルパスを指定してください。
- ・ ログ消失防止指定  
更新ログ設定をマルチログにて運用時、更新ログの消失を警告するメッセージの出力と、最後の更新ログファイルが更新ログファイルサイズに達してしまった場合の動作を選択できます。IFAS ファイルのバックアップと更新ログファイルのリセットを行う時期の判断に利用できます。  
「更新をエラーとする」チェックボックスをONにした場合、最後の更新ログファイルが更新ログファイルサイズに達してしまった時点で、それ以降の更新がエラーとなります。この場合、更新アプリケーションには、ファイル入出力状態＝“90”、詳細ステータス＝1704(0x06a8)が返ります。  
「イベントログに警告／エラー情報を出力」「テキストファイルに警告／エラー情報を出力」チェックボックスをONにした場合、イベントログ／テキストファイルに以下のメッセージが出力されます。  
○最後の更新ログファイルに切り替わった時、更新ログファイルの空き容量がわずかであることを表示警告メッセージが出力されます。



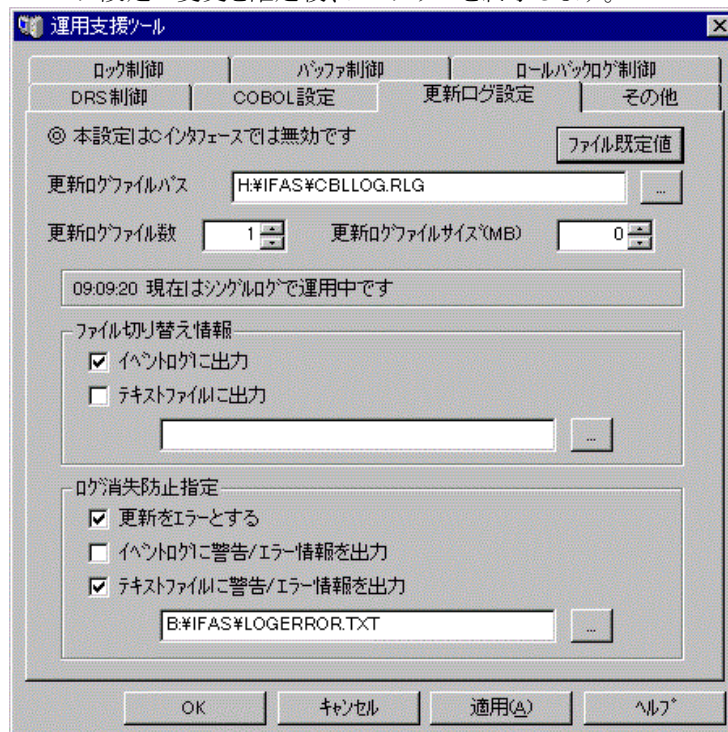
○「更新をエラーとする」チェックボックスがOFFの場合、最後の更新ログファイルが更新ログファイルサイズに達すると、その後、更新ログファイルの切り替え時に更新ログが消失したことを表すエラーメッセージが出力されます。

「テキストファイルに出力」を指定した場合、ログ消失情報を出力するテキストファイルのファイルパスを指定してください。

## 更新ログ設定の変更方法

以下に更新ログ設定を変更する方法を述べます。

- (1) “運用支援ツール：IFASSYSG.EXE”を実行します。
- (2) 「運用支援ツール」ダイアログボックスが開きますので“**更新ログ設定**”タブを選択します。
- (3) 該当する項目を設定します。  
ここで変更した更新ログファイルパス、更新ログファイル数、更新ログファイルサイズを既定値に戻したい場合は“**ファイル既定値**”ボタンを押してください。
- (4) 次に“**適用**”ボタンを選択すると更新ログ設定の変更を確定し、“**OK**”ボタンを選択すると更新ログ設定の変更を確定後、プログラムを終了します。



## “その他”タブの設定項目

設定パラメータは以下のとおりです。

### ・立ち上げ時のロックリセット機能実行の有無

電源ダウン等でプロセスが終了する時にクローズされなかったIFAS ファイル(以降未クローズファイルと述べる)は、ファイル内の管理情報とデータの整合性がとれていない等、ファイルが壊れている可能性があるため、通常はそのままではアクセスできない様に、IFAS管理情報ファイル上でロック(以降、未クローズロックと述べる)されています。

自動ロックリセット機能を実行すると、立ち上げ時(IFAS FILEイニシエータ起動時)に未クローズファイルを自動検出し、未クローズロックを解除してIFAS ファイル内の管理情報とデータの整合性をとります。なお、自動ロックリセット機能が実行された場合以下のファイルに実行ログを出力しますので、適当な時期にログの削除を行ってください。

ログファイル名: IFAS FILE、IFASPRO RDBインストールディレクトリ¥AUTOFRST.LOG

主なログ情報:

- ・ロックリセット処理を行ったファイル名
- ・ロックリセット処理をエラーが起きた時の詳細情報



### ・ 自動C/S設定

ファイル名にUNC名(¥¥サーバ名¥共有名¥ファイルパス)またはネットワークドライブのファイルパスが指定されている場合、自動的にIFASのC/Sアクセスが行われるようになります。

#### 【注意】

- ・IFASのC/Sアクセスを行う場合の一般的な注意事項

IFAS一般ファイルをC/Sアクセスする場合、下記のようにサーバ名、ファイル名を明記する必要があります。

” <サーバ名>サーバ上のファイルパス”

しかし、「自動C/S設定を行う」チェックボックスが無効、かつ、「ファイル名にUNC名/ネットワークドライブ」を指定してしまった場合、IFASのC/S機能が使われないため、IFASサーバを経由せずにファイルアクセスが行われてしまいます。その場合、複数のクライアントからファイルアクセスが行われると、データの保証が出来なくなることがあります。

### ・ エラーログファイルの最大サイズ

エラーログファイルの最大サイズを指定します。KB単位で設定します。既定値は4096です。

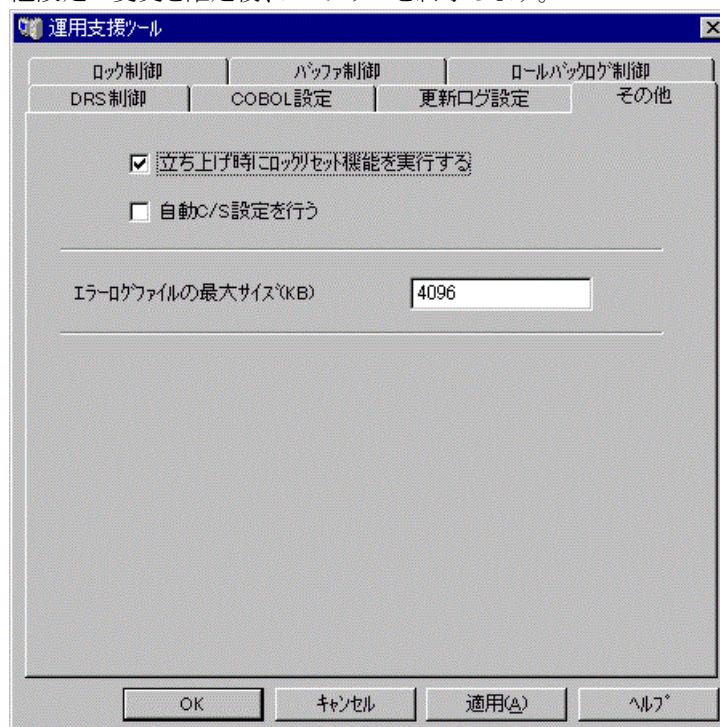
エラーログファイル(ERROR01.LOG、ERROR02.LOG、ERROR03.LOG)は3世代で管理されており、ここで設定されるエラーログの最大サイズを超えるタイミングで切り替えが発生します。

エラーログファイルはIFAS FILE、IFASPRO RDBのインストールディレクトリ¥rbk配下に作成されるテキストファイルで、メモ帳等で参照することが可能です。

## その他設定の変更方法

以下にその他設定を変更する方法を述べます。

- (1) “運用支援ツール: IFASSYSQ.EXE”を実行します。
- (2) 「運用支援ツール」ダイアログボックスが開きますので“その他”タブを選択します。
- (3) 該当する項目を設定します。
- (4) 次に“適用”ボタンを選択するとその他設定の変更を確定し、“OK”ボタンを選択するとその他設定の変更を確定後、プログラムを終了します。



## 7-4-2 バックアップ支援ユーティリティ

ユーザはIFAS FILE、IFASPRO RDB運用中にIFAS ファイルのバックアップを採取する場合があると思いますが、24時間運用等を行っている場合、バックアップのためのファイルが更新されないタイミングを作る必要があります。それには、ユーザアプリケーションを終了させたりしなければならず、運用方法によっては簡単にできない場合もあります。そこでIFAS FILE、IFASPRO RDBでは、ユーザが任意のタイミングで安全にIFAS ファイル環境のバックアップを行えるように、IFAS FILE、IFASPRO RDBの動作を抑制するユーティリティを提供します。

### 機能概要

バックアップ支援ユーティリティ IFASBKSPは、ユーザが任意のタイミングで安全にIFAS ファイル環境のバックアップを行えるように、バックアップツール起動中はIFAS FILE、IFASPRO RDBの動作を抑制するためのアプリケーションです。

なお、本アプリケーションはバックアップ機能を実装していません。ファイルをバックアップ(ファイルの物理コピー)するアプリケーションまたはコマンドが別途必要になります。

本アプリケーションを使用する場合は、IFAS FILEイニシエータが起動していなければなりません。

また、IFAS FILE関数の実行が抑制されますので、ファイルコピーユーティリティ等のIFAS FILE関数を使用しているアプリケーションでバックアップを取ることはできません。

バックアップ支援ユーティリティには以下の機能があります。

ファイルオープン相当の関数(ファイル名指定関数)の実行を抑制します。

更新ログをリセットします。

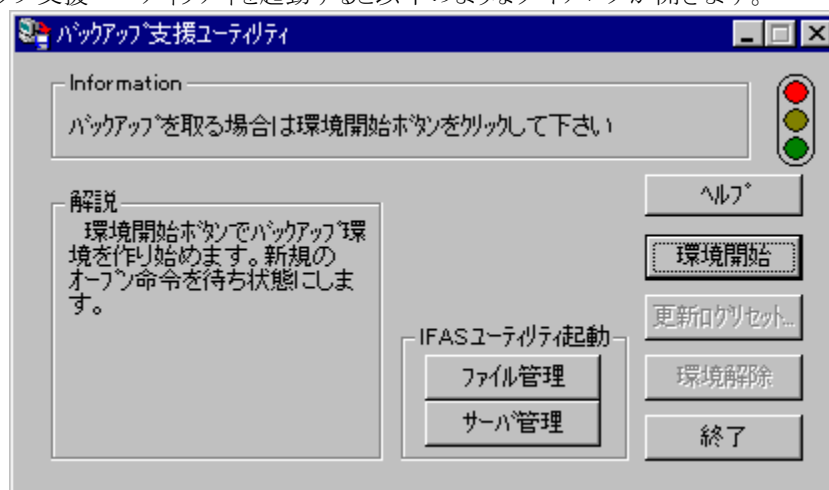
コンソールコマンドから以下のようにして実行すると、バックアップ採取処理をバッチファイルで実行することができます。

#### IFASBKSP バックアップバッチファイル名 [/待ち時間] [/I]

MS-DOSのバッチコマンドを使用して必要なファイルのバックアップを採る場合、バックアップの採取後、更新ログファイルのリセットを行うコマンドを提供します。実行により不必要になった更新ログを廃棄します。

#### IFASLFRS /スイッチ ファイル名

バックアップ支援ユーティリティを起動すると以下のようなダイアログが開きます。



### フィールド／ボタンの説明

#### Information

ユーティリティの現在の状況、現在オープンしているファイル数などを表示します。

#### 解説

「Information」に合わせて現在行われている処理や、ユーザがバックアップ処理を行うためにどういうアクションをすればいいかを表示します。

#### [ヘルプ]ボタン

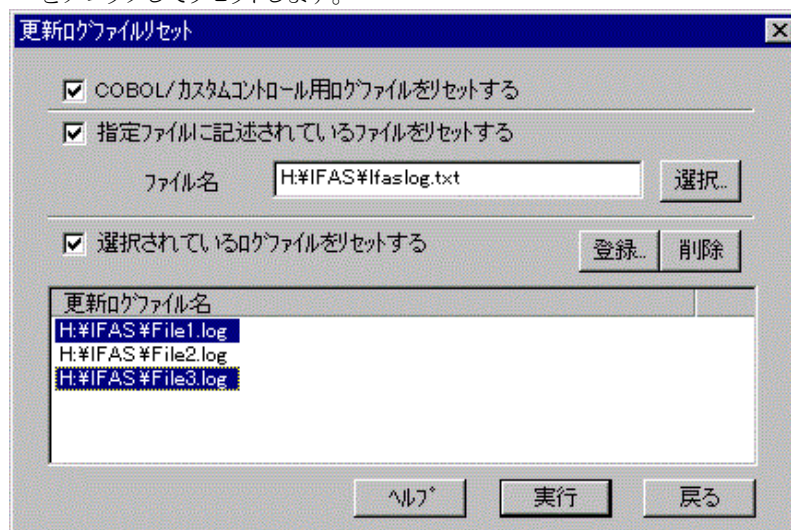
バックアップ支援ユーティリティのヘルプを表示します。

**[環境開始]ボタン**

バックアップ環境を構築します。これにより、IFAS FILE、IFASPRO RDBのオープン系の関数は実行を抑制されます。

**[更新ログリセット]ボタン**

更新ログリセット用のサブ画面を開きます。以下のダイアログボックスで目的の更新ログファイルをチェックしてリセットします。

**COBOL/カスタムコントロール用ログファイルをリセットする**

運用支援ツールの[更新ログ設定]タブで指定されている更新ログファイルをリセットします。  
(COBOL/カスタムコントロールで利用する場合)

**指定ファイルに記述されているファイルをリセットする**

ボックスに指定されているテキストファイルに記述されている更新ログファイルをすべてリセットします。

**選択されているログファイルをリセットする**

使用される可能性のある更新ログファイルを登録しておくことができます。リストに登録されているファイルのうち、実際にリセットしたいファイル選択して実行してください。ファイルは複数選択可能です。

**[環境解除]ボタン**

バックアップ環境を解除します。これにより、実行を抑制されていたIFAS FILE関数は実行されます。

**【注意】**

抑制を解除されたアプリケーションの起動順序はコンピュータのCPU割り当てに依存します。  
(止まった順に起動するとはかぎりません)

**[終了]ボタン**

アプリケーションを終了します。

## コマンドラインからの実行

バックアップ支援ユーティリティは、実際にバックアップを実行するバッチファイルをパラメータとしてコンソールコマンドから以下のように実行することができます。

**【書式】**

**IFASBKSP バックアップバッチファイル名 [/待ち時間] [/I]**

**【指定内容】**

バックアップバッチファイル名

実際にファイルコピーを行うコマンドなどを記述したバッチファイルです。

DOSコマンドのCOPYやMOVEなどを利用します。

バックアップ終了後に更新ログファイルをリセットする場合にはバッチファイルの最後に更新ログファイルリセットコマンドを記述してください。

/待ち時間

バッチコマンドの最大起動待ち時間です。秒単位で指定します。1以上の値を指定してください。(デフォルトは300で)

本コマンドでは、コマンドが実行された時に既にオープンされているファイルが存在した場合、それらがクローズされるのを待ちます。さらに、コマンド実行後のオープン命令も待たされることとなります。

コマンドが実行されたマシンで使用されているIFAS ファイルがすべてクローズされるのを待って、ユーザ指定のバックアップバッチコマンドを実行し、バッチの終了後、アプリケーションの続行(待たされているオープン命令の実行)を行います。

本パラメータはその時の待ち時間を指定するものです。

/I

本パラメータ(大文字・小文字の区別なし)が指定された場合、バックアップバッチが異常終了した時でもファイルオープン抑制は解除されます。

本パラメータを指定して実行し、IFASBKSPのバッチ実行が終了した時にバックアップバッチが異常終了した場合には、実行ディレクトリにIFASBKSP.ERRというファイルが作成されます。ファイル内にはエラーコードが出力されます。IFASBKSP.ERRは、次のエラーが発生するたびに上書きされ、正常終了時に削除されます。

#### 注意

- ・待ちが解除されたアプリケーションの起動順序はコンピュータのCPU割り当てに依存します。(止まった順に起動するとはかぎりません)
- ・パラメータは必ずブランクで区切ってください。ファイル名にブランクが含まれる場合はパラメータをダブルクォーテーション(")で囲んでください。
- ・本コマンドでは更新ログのリセットは行われません。

#### 【コマンドラインの戻り値】(ERRORLEVEL)

0	正常終了しました
1	強制終了されたアプリケーションが存在する可能性があります (ワーニングです。処理は実行されます)
255	ユーティリティによるエラーで終了しました

## エラーステータス

コマンドラインで実行中にエラーを検出した場合は、カレントディレクトリに IFASBKSP.ERR というファイルを作成し、エラーコードを書き出します。

#### ・ステータス一覧

41400	領域確保に失敗
41401	IFAS FILEが正常にインストールされていない
41402	該当ファイルは存在しない
41403	指定バッチファイルの起動に失敗した
41404	指定バッチファイルの実行でエラーが発生した
41405	DLLのロードに失敗した
41406	ゲート処理でエラーが発生した
41407	イベント発行処理でエラーが発生した
41408	セキュリティ処理でエラーが発生した
41409	待ち時間が終了した
41410	パラメータエラー

上記とは別にIFAS FILE関数のエラーも返却することがあります。

## 更新ログリセットコマンド

バックアップ採取後、不要になった更新ログをリセットします。コンソールコマンドから以下のような書式で実行することもできますし、バッチファイルで実行する場合には、バックアップ支援ユーティリティのコマンド実行のパラメータに指定するバッチファイルの最後に記述して使用します。

## 【書式】

## IFASLFRS /スイッチ [ファイル名]

## 【指定内容】

## スイッチ

以下にスイッチの種類と指定方法を記述します。3つのスイッチのうち、いずれか1つを必ず指定してください。

/CBL	COBOL用ログファイルをリセットします。運用支援ツールの[更新ログ設定]で指定している更新ログファイルをリセットします。
/TEXT ファイル名	指定ファイルに記述している更新ログファイルをリセットします。指定ファイルには、更新ログファイル名のフルパスを、テキスト形式で記述します。更新ログファイル名は1行に1ファイル記述し、複数ファイル指定する場合は、複数行に記述してください。主に、Cインターフェースで使用する更新ログファイル名を記述してください。
/C 更新ログファイル名	指定している更新ログファイルをリセットします。更新ログファイルは、主に、Cインターフェースで使用する更新ログファイル名を記述してください。

## 【戻り値】(ERRORLEVEL)

0	正常終了しました
10	複数ファイルでエラー発生
11	シングルログファイルが存在しません 更新ログファイル名が正しいか確認してください。
12	シングルログファイルBUSY 更新ログファイルが使用されています 更新ログファイルをリセットすることはできません
13	シングルフォーマットエラー 更新ログファイル名が正しいか確認してください
14	シングルログファイルアクセスエラー エラーログを参照してください
15	シングルログファイル削除エラー エラーログを参照してください
16	TEXTファイルのフォーマットエラー TEXTファイルの内容を確認してください
30	スイッチ情報不正 正しいスイッチを指定してください
31	レジストリ情報入手に失敗 エラーログを参照してください
32	レジストリ情報更新に失敗 エラーログを参照してください
33	TEXTファイル操作でエラー エラーログを参照してください

## バックアップバッチファイル例

MS-DOSのCOPYコマンドを利用したファイルコピー&更新ログのリセット

```
COPY B:¥UNYOU¥981119¥DATA C:¥BACKUP¥981119¥DATA
COPY B:¥UNYOU¥981119¥DATA.IDX C:¥BACKUP¥981119¥DATA.IDX
COPY B:¥UNYOU¥981119¥WORK.REL C:¥BACKUP¥981119¥WORK.REL
COPY B:¥UNYOU¥981119¥LOGFILE.LOG C:¥BACKUP¥981119¥LOGFILE.LOG
IFASLFRS /TEXT B:¥UNYOU¥981119¥LOGFILE.LOG
```

## 7-4-3 サーバ管理ユーティリティ

サーバ管理ユーティリティは、IFAS FILEまたはIFASPRO RDBで動作する、IFAS ファイルをアクセス中の実行アプリケーションの各種情報を一覧表示するユーティリティです。

※サーバOSでのみ利用可能です。また、IFASPRO RDBのランタイム製品では利用できません。

本ユーティリティは次のような場合に使用します。

IFAS ファイルのバックアップ時など、IFASを使用するアプリケーションが動作していないことを確認したい場合

クライアントアプリケーションが正しくC/Sアクセスが行われていることを確認したい場合

### 機能概要

サーバ管理ユーティリティで一覧表示される情報は以下のとおりです。

アプリケーションの実行要求をしたクライアントマシン名

アプリケーションのプロセスID

アプリケーションのスレッドID

アプリケーションに対応したサーバプロセスのプロセスID

アプリケーションがオープンしているIFAS ファイルの数

アプリケーションがサーバに接続した日付 & 時間

サーバ管理ユーティリティを起動すると以下のようなダイアログが開きます。



### メニューバーの説明

#### ファイル(F)

##### アプリケーションの終了(X)

サーバ管理ユーティリティを終了します。

#### 表示(V)

##### ステータスバー(S)

ステータスバーの表示／非表示を切り替えます。

##### クライアント順(C)

情報一覧をクライアント名順にソート表示します。

##### クライアントPID順(P)

情報一覧をクライアントプロセスID順にソート表示します。

##### クライアントTID順(T)

情報一覧をクライアントスレッドID順にソート表示します。

##### サーバPID順(V)



情報一覧をサーバプロセスID順にソート表示します。

#### ファイル数順(F)

情報一覧をファイル数順にソート表示します。

#### 日付順(D)

情報一覧を日付順にソート表示します。

#### 情報のリフレッシュ(R)

一覧表示、および、ステータスバーの更新時間を更新します。

情報は、時間が経過すれば古くなるため、適時、本機能により情報を更新してください。

## ステータスバーの説明

### 表示数

現在表示中(接続中)のクライアント数を表示します。

### 情報取得日時

現在表示中の情報の取得日時(情報のリフレッシュを行った日時)を表示します。

## 7-4-4 IFAS管理情報メンテナンスツール

IFAS管理情報メンテナンスツール“IFASDRSM”は、IFAS ファイル識別子管理ファイル(DRSファイル)に登録されている情報を変更することにより、運用/保守の効率化を図るためのコマンド形態のアプリケーションです。

なお、本アプリケーションはIFASの管理情報を直接変更することになりますので、使用する状況によっては他のアプリケーションに影響を与えることになります。(例:ファイルのオープンでエラーになる)本アプリケーションをご使用になる場合は、十分に注意してください。

また、本アプリケーションを使用するためにはIFAS FILEイニシエータが起動していなければなりません。

## 機能概要

本アプリケーションには下記の機能があります。

### (1) IFAS ファイルのビジーロック情報リセット

IFASを使用しているアプリケーションが強制終了された場合等、ファイルのクローズ処理が行われずに異常終了すると、ファイルのビジーロック情報(IFASの管理情報)が残ったままとなります。

通常、IFAS FILEイニシエータを再起動しないとビジーロック情報は解除されませんが、本機能によってIFAS FILEイニシエータを再起動することなくビジーロック情報を解除することが可能となります。

アプリケーションが異常終了した後に本機能を使用すると、そのアプリケーションで更新中であったIFAS ファイルに未クローズロックがかかります。この未クローズロックはファイル管理ユーティリティのロックリセット機能かロックリセットユーティリティで解除してください。

ただし、本機能を使用してもIFAS FILEイニシエータが管理している情報等によりプログラムからのアクセスが可能にならない場合があります。その場合はIFAS FILEイニシエータの再起動が必要となります。

### (2) IFAS ファイルのビジーロック情報リセットと未クローズロックリセット

(1)の機能と同じ処理を行った後、未クローズロックがかかっている場合、自動的にロックリセットユーティリティを実行することもできます。

## (3) IFAS ファイルの管理情報の削除

IFAS ファイルをIFAS以外の手段(エクスプローラ等)で削除しますと、管理ファイルには削除されたIFAS ファイルについての管理情報が残されたままとなります。

このような情報が多量に管理ファイル内に存在しますと、管理ファイルの情報を検索する場合(オープン処理やクローズ処理時)に余分な時間がかかることになります。

このような場合、本機能を使用して不要な情報を削除することにより、アクセス性能を上げることができます。

## (4) 登録されているIFAS ファイルの一覧の出力

登録されているIFAS ファイルの情報はファイル管理ユーティリティで参照することができますが、管理情報が登録されていてもIFAS ファイルの実体が存在しない場合にはファイル管理ユーティリティではその情報を表示しません。

(3)の機能を使用して不要な管理情報を削除したい場合など、本機能により一覧を出力することで、管理情報ファイルに登録されているすべてのIFAS ファイル名を知ることができます。

## 操作方法

本アプリケーションは、MS-DOSプロンプトやエクスプローラ等から、パラメータにオプションを指定して、コマンドラインプログラムとして実行します。

なお、管理情報が変更される処理の場合、対象となるIFAS ファイルが他のプロセスで使用中の時には、変更処理は実行されず本アプリケーションはエラーとなります。

## 【書式】

**IFASDRSM -O:オプション -F:ファイル名 [-M] [-D] [-NOCHK]**

パラメータは必ずブランクで区切ってください。ファイル名にブランクが含まれる場合はファイル名をダブルコーテーション(")で囲んでください。

## 【パラメータ説明】

### オプション

IFAS管理情報メンテナンスツールで実行する機能を指定します。サポートしている機能は以下のとおりです。

- C** … 指定されたIFAS ファイルのビジーロック情報をリセットします。ファイルの管理状態が更新途中であった場合には未クローズロックがかかります。ただし、トランザクション指定でオープンされているファイルは本処理の対象外となります。また、他のプロセスで使用中の場合も、本処理は実行されません。
- R** … 指定されたファイルのビジーロック情報をリセット後、ファイルの管理状態が更新途中であった場合にはロックリセットが実行されます。ただし、トランザクション指定でオープンされているファイルは本処理の対象外となります。また、他のプロセスで使用中の場合も、本処理は実行されません。なお、本オプションを実行時にロックリセット処理でエラーが発生した場合は、ロックリセットユーティリティ(IFASFRST.EXE)を使用してください。
- E** … 指定されたファイルの管理情報を管理ファイルから削除します。ただし、トランザクション指定でオープンされているファイルは本処理の対象外となります。また、他のプロセスで使用中の場合も、本処理は実行されません。
- L** … 管理情報に登録されているIFAS ファイルの一覧を指定されたファイルに出力します。出力されるファイルはCSV形式ファイルで下記の情報が出力されます。

“ファイル名”, “編成”, “状態”, “対のファイル名”
--------------------------------

ファイル名	: 登録されているファイル名
編成	: ファイルの編成編成を出力します。



	索引順編成／相対順編成／順編成
状態	: ファイルの状態を出力します。 使用中／未クローズ／クローズ失敗
対のファイル名	: 索引順編成の場合、一対となっているデータファイル名かインデックスファイル名を出力します。相対編成と順編成の場合は何も出力されません。

## ファイル名

Lオプションの場合は出力先のファイル名を指定します。  
それ以外のオプションの場合は、管理情報を変更する管理ファイルに登録されているIFAS ファイル名を指定します。  
ただし、“-M”(一括処理機能)を指定する場合は、IFAS ファイル名の一覧を記述したファイル名を指定してください。

## 拡張機能

IFAS管理情報メンテナンスツールでは拡張機能を使用することができます。サポートしている機能は以下のとおりです。

- M** …………… 一括処理機能を使用します。この機能を指定する場合は、ファイル名にはファイル名一覧のファイル(処理を行う対象となるIFAS ファイル名の一覧を記述したCSVファイルまたはTEXTファイル)を指定して下さい。ファイル名一覧を作成する場合は、行の先頭にファイル名を“(ダブルコーテーション)で囲んでください。それ以外の記述は無視されます。Lオプションで出力したファイルをそのまま利用することもできます。
- D** …………… 詳細エラー出力機能を使用します。この機能を指定すると、本アプリケーションの終了時にエラーステータスとエラーメッセージを端末に出力します。正常に終了した場合は「コマンドは正常に終了しました。」というメッセージを出力します。
- NOCHK** … 対象となるIFAS ファイルがトランザクション指定でオープンされていたり他のプロセスで使用中でも、管理情報の変更を実施します。

## 【戻り値】(ERRORLEVEL)

- 0 正常終了しました  
0以外 エラーが発生しました  
返却される値はエラーステータスの下2桁です。エラーの内容はエラーステータスを参照してください。

## エラー処理

コマンドラインで実行中にエラーを検出した場合は、エラーステータスの下2桁をコマンドラインの戻り値として返却します。“-D”(詳細エラー出力機能)が指定されている場合には、エラーステータスとエラーメッセージを端末に出力します。

また、エラーの内容によってはIFASのエラーログファイルに詳細情報を出力します。IFASのエラーログはデフォルトでは下記の場所にあります。

IFASのインストールディレクトリ¥RBK¥ERRORxx. LOG (xx:01~99)

## エラーステータス

・ステータス一覧

エラーコード 意味

53601	パラメータ不正
53602	指定されたファイル名が不正

53603	指定されたIFAS ファイルは存在しない
53604	指定されたIFAS ファイルは使用中
53605	指定されたIFAS ファイルは処理対象外
53610	作業用メモリの確保に失敗した
53611	IFAS FILEイニシエータが動作していない等、実行環境が不正で動作できない
53612	管理情報ファイルのアクセスエラー
53613	CSVファイルのアクセスエラー
53620	ロックリセット実行処理でエラー

## 使用例

IFAS管理情報メンテナンスツールを使用するバッチファイルの記述例を以下に示します。  
処理の内容は、ファイル名一覧をOUTFILE. CSVに出力してその結果を利用してIFAS ファイル  
のリセットを実行しています。

```
ECHO OFF
ECHO リスト出力を行います。
IFASDRSM -O:L -F:OUTFILE.CSV -D
IF ERRORLEVEL 1 GOTO :ERROR1
ECHO ビジーリセット&ロックリセット実行を行います。
IFASDRSM -O:R -F:OUTFILE.CSV -M -D
IF ERRORLEVEL 1 GOTO :ERROR2
ECHO 処理を完了しました。
GOTO :EOF
:ERROR1
ECHO ERRORLEBEL = %ERRORLEVEL%
ECHO リスト出力でエラーが発生しました。
GOTO :EOF
:ERROR2
ECHO ERRORLEBEL = %ERRORLEVEL%
ECHO リセット実行でエラーが発生しました。
GOTO :EOF
:EOF
```

## 7-5 定義ファイルのパラメータ置換機能

定義ファイルをパラメータとするユーティリティのコマンド実行は、パラメータで指定した文字列で定義ファイル中の特定の文字列を置き換えた状態で実行することができます。これらのコマンドは、パラメータに置換文字列1～9が指定されると、定義ファイルの文字列置換を行います。置換文字列が1つも指定されていない場合には、定義ファイルの文字列置換は行いません。

定義ファイルの文字列置換は以下のような規則で行われます。(コメント行を除く)

置換前	置換後
%0	定義ファイル名のフルパス(通常使われることはありません)
%1～%9	置換文字列1～置換文字列9 (引数で指定されていない場合、その"%n"は削除されます)
%%	%
%	“%”が削除されます

置換文字列を指定したコマンドライン実行を行うと“%”が特殊文字として扱われるため、置換させたくない“%”(入出力ファイル名中の“%”等)が定義ファイルに含まれている場合に正しく実行されません。(置換文字列を指定しないコマンドライン実行、およびGUIによる実行で、このような定義ファイルを使う分には、何の問題ありません)

このような定義ファイルを基にして、置換文字列を指定したコマンドライン実行に使用する定義ファイルを作成する際には、置換させたくない“%”を“%%”に書き換えてください。

GUIで“%”を含む入出力ファイル名を指定して定義ファイルの自動作成を行うと、“%”が含まれた定義ファイルが作成されます。この定義ファイルを基にして、置換文字列を指定したコマンドライン実行に使用する定義ファイルを作成する際には、特に注意してください。

また、サーバ名付きのファイル名をパラメータとする場合は、全体<サーバ名>+ファイル名)を“ ”(ダブルクォーテーション)で囲むようにしてください。

なお、この機能は指定された定義ファイルの内容を変更するものではありません。

文字列置換された定義ファイルは、テンポラリファイルに一時的に保存します。ただし、置換文字列を指定したコマンドライン実行を行っても、定義ファイル中に処理対象となる“%”が存在しなければ、テンポラリファイルは作成されません。作成されたテンポラリファイルは、コマンドユーティリティの実行完了時に削除されます。

このテンポラリファイルは、テンポラリディレクトリに作成され、ファイル名は、“ifs????.tmp”(????は16進数)となります。

ここでいうテンポラリディレクトリとは、環境変数“TMP”、“TEMP”で指定されたディレクトリのことです。両方で指定されている場合には“TMP”の指定が優先されます。

これらの環境変数でテンポラリディレクトリが指定されていない場合には、カレントディレクトリ、またはOSがインストールされているディレクトリにテンポラリファイルが作成されます。

## 7-6 その他

### 7-6-1 IFASプロパティ表示

IFAS FILE、IFASPRO RDBをインストールすると、インストール物件に関するさまざまな情報をOSのレジストリとして登録します。これらの情報は、IFAS FILE、IFASPRO RDBの運用におけるトラブルが発生した際の調査情報として大変重要なものです。IFAS FILE、IFASPRO RDBでは、これらの情報を表示するコンソールコマンドを提供しています。

#### 機能概要

IFASプロパティ表示コマンドでは、次の情報を取得することができます。

- 基本情報  
IFAS FILE、IFASPRO RDBのバージョン情報やエラーログファイル情報など、IFAS FILE、IFASPRO RDBが動作する上で基本的な事柄を表示します。
- ロックテーブル関連情報  
IFAS FILE、IFASPRO RDBが独自に行っている排他制御のテーブルに関する情報を表示します。
- 資源情報  
ロックテーブルを除く、資源情報(バッファ関連、ロールバック関連)を表示します。
- COBOL(A-VX II)関連情報  
現在設定されているCOBOL(A-VX II)オプション情報を表示します。
- COBOL(A-VX5)関連情報  
現在設定されているCOBOL(A-VX5)オプション情報を表示します。

#### 【注意】

IFAS FILE、IFASPRO RDBでは、これらの多くの情報をもっております。そのため、本コマンド実行時にはコンソールがスクロールしてしまい、すべての情報を画面上で入手するのはむずかしいと思われます。そこで本コマンド使用時には、MS-DOSのリダイレクト機能を使用してファイルに出力することをお薦めします。

#### コマンドの実行方法

IFASプロパティ表示コマンドは、コンソールコマンドから以下のようにして実行することができます。

##### IFASINFO [レベル]

パラメータは必ずブランクで区切ってください。

##### レベル

###### 1 : 基本情報

出力例

2014/10/17 00:00:00 (注1)

```

CobolVersion          : 1.00
PathName               : C:\Program Files(x86)\NEC\IFAS FILE
RegisteredOwner       : OwnerName
RegisteredOrganization : NEC
SI                     : 141017
IfasInstallType        : Standard
ErrorLogFilePathName  : C:\Program Files(x86)\NEC\IFAS FILE\RBK\ERROR.LOG
IfasRdb                : ** KEY_NOT_FOUND **
```

ErrorLogGenerationCount : 3  
 ErrorLogFileSize : 4096  
 ClientSetupFlag : SERVER

(注1) 情報取得した日付

## 2 : ロックテーブル関連情報

出力例

2014/10/17 00:00:00  
 LRC¥MultiExecuteThreadCount : 10  
 LRC¥ParallelOpenFileCount : 10  
 LRC¥SimultaneousRecordLockCount : 400  
 LRC¥LockWaitingTime : 0  
 LRC¥PctTableCount : 100  
 LRC¥SctTableCount : 4000  
 LRC¥TctTableCount : 4000  
 LRC¥TtbTableCount : 100  
 LRC¥MaxMemorySize : 0  
 PIO¥SharedBufferSize : 15

## 3 : 資源情報

出力例

2014/10/17 00:00:00  
 RCV¥RollbackLogDirectoryPathName : C:¥Program Files(x86)¥NEC¥IFAS FILE¥RBK  
 RCV¥UseServerDirectory : NO USE  
 DRS¥DrsDatabaseFilePathName : C:¥Program Files(x86)¥NEC¥IFAS FILE¥DRS  
 SVC.DB

## 4 : COBOL(A-VX II)関連情報

出力例

2014/10/17 00:00:00  
 CBL¥LogFilePathName : C:¥Program Files(x86)¥NEC¥IFAS FILE¥CBL  
 LOG.RLG  
 CBL¥AVX2¥FileBusy : 0  
 CBL¥AVX2¥OpenPrior : 0  
 CBL¥AVX2¥SelectPrior : 0  
 CBL¥AVX2¥ScratchCurrent : 0  
 CBL¥AVX2¥StartLock : 0  
 CBL¥AVX2¥DupRewriteDelete : 0  
 CBL¥AVX2¥ReadDupkey : 0  
 CBL¥AVX2¥WriteDupkey : 0  
 CBL¥AVX2¥TransactingClose : 0  
 CBL¥AVX2¥DupPrimarykey : 0  
 CBL¥AVX2¥ViewOpen : 0  
 CBL¥AVX2¥NonekeyRead : 0  
 CBL¥AVX2¥OutputSelect : 0  
 CBL¥AVX2¥NoSelectScratch : 0  
 CBL¥AVX2¥Rollback : 0  
 CBL¥AVX2¥Log : 0  
 CBL¥AVX2¥TransactionStatus : 0  
 CBL¥AVX2¥InputReadLock : 0  
 CBL¥AVX2¥InputStartLock : 0  
 CBL¥AVX2¥KeyAttribute : 0  
 CBL¥AVX2¥OnlyPrimarykey : 0  
 CBL¥AVX2¥NumberOfBuffer : 10  
 CBL¥AVX2¥OpenRecordCheck : 0

CBL¥AVX2¥SelectMinMem	: 0
CBL¥AVX2¥SelectMaxMem	: 0
CBL¥AVX2¥RfdMode	: 0
CBL¥AVX2¥RewriteDupKey	: 0

## 5 : COBOL(A-VX5)関連情報

出力例

2014/10/17 00:00:00

CBL¥AVX5¥FileBusy	: 1
CBL¥AVX5¥OpenPrior	: 1
CBL¥AVX5¥SelectPrior	: 1
CBL¥AVX5¥ScratchCurrent	: 1
CBL¥AVX5¥StartLock	: 1
CBL¥AVX5¥DupRewriteDelete	: 1
CBL¥AVX5¥ReadDupkey	: 0
CBL¥AVX5¥WriteDupkey	: 0
CBL¥AVX5¥TransactingClose	: 0
CBL¥AVX5¥DupPrimarykey	: 0
CBL¥AVX5¥ViewOpen	: 1
CBL¥AVX5¥NonekeyRead	: 1
CBL¥AVX5¥OutputSelect	: 1
CBL¥AVX5¥NoSelectScratch	: 1
CBL¥AVX5¥Rollback	: 0
CBL¥AVX5¥Log	: 0
CBL¥AVX5¥TransactionStatus	: 1
CBL¥AVX5¥InputReadLock	: 0
CBL¥AVX5¥InputStartLock	: 0
CBL¥AVX5¥KeyAttribute	: 0
CBL¥AVX5¥OnlyPrimarykey	: 0
CBL¥AVX5¥NumberOfBuffer	: 10
CBL¥AVX5¥OpenRecordCheck	: 0
CBL¥AVX5¥SelectMinMem	: 0
CBL¥AVX5¥SelectMaxMem	: 0
CBL¥AVX5¥RfdMode	: 0
CBL¥AVX5¥RewriteDupKey	: 0

99 : 1～5のすべての情報

## 8. 関連製品

### 8-1 ファイルメンテナンスキット

ファイルメンテナンスキットの日本語ファイルメンテナンスでIFAS ファイルを新規作成/更新することが可能です。

諸元および動作設定についてはCOBOLからの利用の場合と同様です。また、動的ファイルオプションおよびファイルの関連付けの順位についてもほぼ同様ですが、COBOLファイルマッパーの記述部分はありません。

IFAS RDB表を新規作成する場合は、あらかじめデータベースを構築しておく必要があります。

なお、ファイルメンテナンスキット実行中に発生したIFAS FILEに関するエラーは、COBOLでの入出力状態、および、詳細ステータスを参照してください(「付録A ステータスコード一覧」参照)。

ファイルメンテナンスキットの詳細については、使用されるIFAS FILE、IFASPRO RDBのバージョンと対応したファイルメンテナンスキットのマニュアルを参照してください。

### 8-2 smart/cs

smart/csからIFAS ファイルを利用することが可能です。

諸元および動作設定についてはCOBOLからの利用の場合と同様です。また、動的ファイルオプションおよびファイルの関連付けの順位についてもほぼ同様ですが、COBOLファイルマッパーの記述部分はありません。

なお、smart/cs実行中に発生したIFASに関するエラーは、COBOLでの入出力状態、および、詳細ステータスを参照してください(「付録A ステータスコード一覧」参照)

smart/csの詳細については、使用されるIFAS FILE、IFASPRO RDBのバージョンと対応したsmart/csのマニュアルを参照してください。

### 8-3 RDB/FILEアクセスコントロール

RDB/FILEアクセスコントロールはExpress5800/600シリーズ上のオフコン資産(A-VXのファイル/RDB)やExpress5800/100シリーズ上のWindows系COBOL資産(Oracle, SQLServer, IFASPRO/RDBのファイル/RDB)をVisualBasicアプリケーションから利用可能とするカスタムコントロール(ActiveX)です。

RDB/FILEアクセスコントロールの詳細については、使用されるIFAS FILE、IFASPRO RDBのバージョンと対応したRDB/FILEアクセスコントロールのマニュアルを参照してください。

# 付録A ステータスコード一覧

## A-1 ステータスの見方

IFAS FILE、IFASPRO RDBは、それぞれC、COBOL、IFASカスタムコントロール(参考)からのプログラムインターフェースを持っています。(Cプログラムインターフェースは一般ファイルアクセスのみ)  
各言語インターフェースによってIFAS FILE、IFASPRO RDBのステータスの種類、返却のされかた、見え方は異なりますのでご注意ください。

### Cプログラムの場合

ステータスはすべて関数の戻り値として利用者プログラムに返却されます。  
詳細は「付録A-3 C言語関数のステータス」を参照してください。

### COBOLプログラムの場合

入出力状態としてプログラムに返却されるものと、プログラム異常終了した場合の関連製品エラーメッセージに表示される詳細ステータスがあります。

入出力状態コードの詳細については「付録A-4-1 入出力状態」を参照してください。

関連製品エラーメッセージに表示される詳細ステータスは値によって以下のように分類されます

50000～50999、54000～54099 の場合

「付録A-4-1 詳細ステータス」を参照してください。

上記以外の値の場合

C言語関数のステータスが16進で表示されますので「付録A-3 C言語関数のステータス」を参照してください。

### IFASカスタムコントロールの場合(参考)

IFAS FILE Ver2.0、IFASPRO RDB Ver2.0より、「IFASカスタムコントロール」機能の提供を中止しました。本項の内容は従来バージョンをご利用のお客様が旧製品の仕様を確認いただくために残しています。

IfasActionStatusにString型3桁でステータスが返却されます。

- ステータスの値が100以上の場合

カスタムコントロールの内部処理エラーステータスになりますので「付録A-5 カスタムコントロールのステータス」を参照してください。

- ステータスの値が100より小さい場合

下2桁がCOBOLの入出力状態と同じ値がセットされますので「付録A-4-1 入出力状態」を参照してください。

なお、IfasActionStatusの値が "0FF"、"0F0" の場合は、それぞれ説明書の "0xffff"、"0xff0" に相当します。

この場合、IfasMinorStatus1にString型5桁で詳細ステータスが返却されます。

詳細ステータスはその値によって以下のように分類されます。

- 0xC350～0xC737、0xD2F0～0xD353 の場合

「付録A-4-1 詳細ステータス」を参照してください。

- 上記以外の値の場合

C言語関数のステータスが返却されますので「A-3 C言語関数のステータス」を参照してください。

IfasMinorStatus1の値が、"01002"、"01006"、"01010"、の場合のみ、IfasMinorStatus2に内部詳細ステータスが返却されます。内部詳細ステータスについては「A-3 C言語関数のステータス」のそれぞれ "1002(0x03ea)"、"1006(0x03ee)"、"1010(0x03f2)" の説明を参照してください。



## A-2 イニシエータのステータス

IFAS FILEイニシエータはIFASシステムの初期化ルーチンで、システムの立ち上げ時に起動します。イニシエータ起動の失敗には以下のような原因が考えられます。

以下に記載するステータスコードをダイアログにて通知します。

また日本語によるメッセージをイベントログ(カテゴリ:アプリケーション)に出力します。

10000 (0x2710)	共有メモリの作成に失敗した
10001 (0x2711)	共有メモリのマッピングに失敗した
10002 (0x2712)	レジストリキーのオープンに失敗した
10003 (0x2713)	レジストリキー値の入手に失敗した
10004 (0x2714)	レジストリキーのクローズに失敗した
10005 (0x2715)	レジストリキー値のサイズが大きすぎる
10006 (0x2716)	サービス制御マネージャの接続に失敗した
10007 (0x2717)	サービス制御関数の登録に失敗した
10008 (0x2718)	サービス状態情報の更新に失敗した
10009 (0x2719)	ディレクトリサービスの初期化に失敗した
10010 (0x271a)	ロック制御サービスの初期化に失敗した
10011 (0x271b)	IFAS FILE、IFASPRO RDBが正しくインストールされていない
10012 (0x271c)	IFASサーバの起動に失敗した

## A-3 C言語関数のステータス

Cプログラムから利用するIFAS FILE関数を実行した場合に、戻り値として返却するステータスコードについてその値と意味について説明します。

本説明書ではステータスコード値を以下のように表記して説明しています。

```

XXXXXXXX = YYYY(0xZZZZ)
      XX      : ステータスコードシンボル(Cプログラムでのコーディングで利用
      XX      : できます)
      XX
      XX
      YYY     : 10進数でのステータスコード値
      Y
      ZZZ     : 16進数でのステータスコード値
      Z
  
```

C言語関数のステータスコードは32ビットの領域を必要とし、基本的に右16ビットと左16ビットに分類して値を返却します。

通常のステータスコードは右16ビットに返却され左16ビットには値を返却しません(0を返却します)が、以下のステータスコードの場合は左16ビットに内部詳細ステータスを伴って返却されますので、その場合10進表示では非常に大きな値が返却されたように見えます。16進表示に変換して右4桁と左4桁に分類してみるようにしてください。

```

IFSYSERR = 1002(0          OSレベルのシステムエラー
x????03ea)
IFITNLER = 1006(0x????     IFAS FILEの内部処理エラー
03ee)
IFCLSVER = 1010(0          IFASクライアントの通信エラー
x????03f2)
  
```

上記16進ステータスの「????」の部分が内部詳細ステータスコードを示しています。

なお、10進表示部分は正確に表記できないので説明書上は右16ビットの値のみ記載しています。内部詳細ステータスの意味は各ステータスコードの説明を参照してください。

またステータスコードはその値によって以下の3種類の意味のステータスに分類されます。この分類によってプログラム側での処置が異なりますので注意してください。

- ・ **正常系ステータスコード** 0(0x0000)～99(0x0063)  
要求した機能は正常に終了しましたが、警告や通知のステータスとして返却します
- ・ **異常系ステータスコード** 100(0x0064)～999(0x03e7)  
要求した機能は失敗しましたが、ファイルに対する処理は継続して行うことができます。
- ・ **回復不可能エラーステータスコード** 1000(0x03e8)～  
要求した機能は失敗し、以降このファイルに対するアクセスは不可能になります。利用者はこのファイルをクローズしなければなりません。

### 正常系ステータスコード ～99(0x0063)

正常系ステータスコードは、要求した処理が正常に終了した場合に返却するステータスであり、警告／通知を意味するステータスとして返却します。

```

IFNORMAL = 0(0x0000)
 意味      処理は正常に終了しました。
  
```

IFNMDPKY = 2(0x0002)

**意味** GETの場合  
サブインデックスの重複キーのレコードがまだ次に存在します。  
PUT、PUTXの場合  
処理したレコードのサブインデックスに重複キーを検出しました。

IFNMLNLK = 4(0x0004)

**意味** OPEN関数は正常終了しましたがロック制御は行いません。

IFNMLNRB = 5(0x0005)

**意味** OPEN関数は正常終了しましたが、オープンされたファイルは以下の理由によりトランザクション処理対象外のため、ロールバックで復旧されません。  
・出力モードでオープンした場合

IFNMUMFL = 6(0x0006)

**意味** 選択したインデックスはデータファイルと不整合状態になっています。

IFWRDPKY = 50(0x0032)

**意味** 重複キーを検出しましたが無視して処理を実行しました。

## 異常系ステータスコード 100(0x0064)～999(0x03e7)

異常系ステータスコードは、主にパラメータの指定や論理的な誤りによって要求した処理が失敗した場合にその理由として通知します。利用者は誤りを訂正後、ファイル処理を継続して行うことができます。

### ファイル作成、ファイルオープンのエラー

IFALDYFN = 101(0x0065)

**意味** 指定したファイル名のファイルは既に存在します。  
**処置** 指定するファイル名を変更して再実行します。

IFNFNDFN = 201(0x00c9)

**意味** 指定したファイル名のファイルは存在しません。  
**処置** 指定するファイル名を変更して再実行します。  
OPEN関数の場合はCREATE関数を実行してから再実行します。

IFDUPOPEN = 202(0x00ca)

**意味** 指定したファイル名のファイルは同じプログラムで既にオープンされています。  
**処置** 指定するファイル名を変更して再実行します。

IFNOIFFL = 203(0x00cb)

**意味** オープン要求したファイルはIFAS形式のファイルではありません。  
**処置** ファイルを初期化してから再実行します。

IFNOINIT = 204(0x00cc)

**意味** オープン要求したファイルは初期化されていません。  
**処置** ファイルを初期化してから再実行します。

IFFLBUSY = 205(0x00cd)

意味 オープン要求したファイルは現在他の利用者が使用中です。  
処置 しばらく待ってから再実行します。

IFRSZERR = 206(0x00ce)

意味 入力、更新モードでオープンした時、指定したレコードサイズが実際のファイル属性のレコードサイズより小さい値です。  
処置 レコードサイズを訂正して再実行します。

IFOPBLER = 207(0x00cf)

意味 ファイルオブションブロックの内容が不正です。  
処置 ファイルオブションブロックの指定を訂正して再実行します。

IFUMCHFL = 209(0x00d1)

意味 以下のような原因でエラーとなりました。  
・インデックスファイルとデータファイルの組み合わせが不正です。  
・インデックスファイルとデータファイルを別サーバに指定しています。  
・ファイルが別のマシンに移動／コピーされています。  
処置 ファイルの組み合わせを訂正して再実行します。

IFOPCTER = 210(0x00d2)

意味 利用者が同時にオープンできるファイル数を超過しました。  
処置 なし。

IFBADHND = 211(0x00d3)

意味 ファイル記述子が不正です。  
処置 ファイル記述子を訂正して再実行します。

IFNOWRIT = 213(0x00d5)

意味 書き込み不可属性のファイルです。  
処置 ファイル属性を訂正して再実行します。

IFFORGER = 214(0x00d6)

意味 ファイル編成が誤っています。  
または、索引順編成ファイルのデータファイルとインデックスファイルを逆に指定しています。  
処置 指定ファイル名を訂正して再実行します。

IFUSERND = 215(0x00d7)

意味 順／相対編成ファイルとして更新された索引順編成ファイルです。  
処置 インデックスファイル属性を訂正して再実行します。

IFRBKBSY = 216(0x00d8)

意味 オープンしようとしているファイルはリカバリ処理中です。  
処置 しばらく待ってから再実行します。

IFRSZOVR = 217(0x00d9)

意味 クライアントとサーバでIFAS FILEのバージョンが異なることによるレコードサイズエラーのためオープンできません。  
処置 クライアントIFAS FILEのバージョンをサーバ側と合わせてください。

IFBKPSY = 220(0x00dc)

意味 IFAS ファイルのバックアップ中(バックアップ支援ユーティリティ起動中)です。  
処置 バックアップの終了を待って再実行します。

## レコード参照、レコード検索のエラー

IFRCBUSY = 302(0x012e)

**意味** アクセスしようとしたレコードは他の利用者がロックしています。  
本ステータスは「運用支援ツール」でロック待ち制限時間を設定した場合のみ返却されます。

**処置** しばらく待ってから再実行します。

IFNCURIX = 303(0x012f)

**意味** 検索用のインデックスが設定されていないのに、キー検索によるアクセス関数を実行しました。

**処置** インデックスの設定を行ってから再実行します。

IFNCURRC = 304(0x0130)

**意味** カレントレコードが設定されていないのに、カレントレコード指定によるアクセス関数を実行しました。

**処置** レコード再位置付けを行ってから再実行します。

IFLGCEOF = 305(0x0131)

**意味** ファイルの論理的終端を検出しました。

**処置** レコード再位置付けを行ってから再実行します。

IFNFNDRC = 306(0x0132)

**意味** 検索要求したキー値またはレコード番号に対応するレコードは存在しません。

**処置** 指定キーまたはレコード番号を訂正して再実行します。

IFNFNDKD = 307(0x0133)

**意味** 指定したキー記述構造体に相当するインデックスが存在しません。

**処置** キー記述構造体を訂正して再実行します。

IFNRWPMK = 308(0x0134)

**意味** PUTX関数においてプライマリインデックスのキー値を書き換えるような要求をしました。

**処置** データイメージを訂正して再実行します。

IFDEDLOK = 309(0x0135)

**意味** デッドロックが発生しました。

**処置** トランザクション処理中の場合  
ROLLBACK関数を実行しトランザクションを取り消します。  
トランザクション処理中でない場合  
COMMIT関数を実行しロック資源を解放します。

IFNPRMIX = 310(0x0136)

**意味** 選択されているインデックスはプライマリインデックスではありません。

**処置** POSITION関数によってプライマリインデックスを選択します。

IFALDYKD = 311(0x0137)

**意味** 指定したキー記述構造体に相当するインデックスが既に存在します。

**処置** キー記述構造体を訂正して再実行します。

IFDLPMKD = 312(0x0138)

**意味** プライマリインデックスを削除しようとしてしました。

**処置** キー記述構造体を訂正して再実行します。

IFAICTER = 313(0x0139)

意味 最大インデックス数を超えてインデックス追加要求を行いました。  
処置 不要なインデックスを削除して再実行します。

IFLKLTOV = 314(0x013a)

意味 1スレッドあたりのロック制限値を超えました。  
処置 トランザクション処理中の場合  
ROLLBACK関数を実行しトランザクションを取り消します。  
トランザクション処理中でない場合  
COMMIT関数を実行しロック資源を解放します。

IFRDMXRN = 316(0x013c)

意味 ファイルの最大レコード番号より大きいレコード番号のレコード参照要求がされました。  
処置 指定レコード番号を訂正して再実行します。

### レコード追加、レコード更新、レコード削除のエラー

IFTXRBFL = 401(0x0191)

意味 IFROLLBACK指定でオープンしたファイルに対しPUT関数、PUTX関数、DELETE関数を実行しましたが、その前にTXSTART関数が実行されていません。  
処置 TXSTART関数を実行して再実行します。

IFDUPKEY = 402(0x0192)

意味 プライマリインデックスまたは重複キー許可指定のないインデックスにおいて重複キーを検出しました。  
処置 レコードイメージを訂正して再実行します。

IFALDYRN = 405(0x0195)

意味 指定レコード番号のレコードは既に存在します。  
処置 指定レコード番号を訂正して再実行します。

IFWTMXRN = 406(0x0196)

意味 ファイルの最大レコード番号より大きいレコード番号のレコード出力要求がされました。  
処置 指定レコード番号を訂正して再実行します。

### トランザクション処理、更新ログ収集処理のエラー

IFALDYLG = 701(0x02bd)

意味 更新ログファイルは既にオープンされています。  
処置 必要ありません。

IFNOPNLG = 702(0x02be)

意味 更新ログファイルがオープンされていません。  
処置 更新ログファイルをオープンして再実行します。

IFNENDTX = 703(0x02bf)

意味 トランザクション処理が終了していない状態で、CLOSE関数、LOGCLOSE関数またはTXSTART関数を実行しました。  
処置 トランザクション処理を終了して再実行します。

IFNOPNRB = 704(0x02c0)

**意味** IFROLLBACK指定のファイルが1つもオープンされていない状態でTXSTART関数を実行しました。

**処置** ロールバック対象ファイルをオープンして再実行します。

IFCTLGCL = 705(0x02c1)

**意味** 更新ログ収集対象ファイルがオープン中の状態で、LOGCLOSE関数を実行しました。

**処置** 更新ログ収集対象ファイルをクローズして再実行します。

IFOLFMLG = 706(0x02c2)

**意味** 旧フォーマットの更新ログファイルが指定されました。

**処置** IFAS ファイルのバックアップを行った後、更新ログファイルを削除してください。

## その他のエラー

IFOPMDER = 801(0x0321)

**意味** 以下のような理由でオープンモードエラーとなりました。

- ・OPEN関数においてアクセスモードとロックモードの組み合わせが不正です
- ・オープン時のアクセスモードに対して実行できないアクセス関数を実行しました

**処置** オープンモードの組み合わせを訂正してOPEN関数を再実行します。  
オープン時のアクセスモードに適した関数を実行するか、関数に適したアクセスモードで再オープンします。

IFKDSCER = 802(0x0322)

**意味** キー記述構造体の内容が不正です。

**処置** キー記述構造体の指定を訂正して再実行します。

IFRCVCAL = 803(0x0323)

**意味** 再帰呼び出しを行いました。

**処置** 現在処理中の関数が終了してから再実行します。

IFPARMER = 804(0x0324)

**意味** 関数のパラメータエラーを検出しました。

**処置** パラメータ指定を訂正して再実行します。

IFFNAMER = 805(0x0325)

**意味** ファイル名の指定が誤っています。

**処置** ファイル名指定を訂正して再実行します。

IFNOSUPP = 806(0x0326)

**意味** サポートされていない関数を呼び出しました。

**処置** なし。

## 回復不可能エラーのステータスコード 1000(0x03e8)～

回復不可能なエラーステータスコードは、IFAS FILEの内部処理中に発生した予期しないエラー、またはシステムでエラーが発生した場合に返却するステータスコードで、継続してこのファイルに対してアクセス要求を行えない状態にあります。

利用者はこのファイルに対して、CLOSE関数を実行しなければなりません。

IFFLIOER = 1001(0x03e9)

意味 ファイルのI/Oエラーを検出しました。

処置 終了処理を行ってからアプリケーションを終了させてください。

IFSYSERR = 1002(0x???03ea)

意味 システムのエラーを検出しました。

詳細 本ステータスは内部詳細ステータスを伴って返却されます。形式はステータス領域の右16ビットに“1002(0x03ea)”を、左16ビットに内部詳細ステータスを返却します。

内部詳細ステータスに以下のような値が返却された場合は、IFAS作業領域のほとんどをロック制御テーブルに占められている可能性があるため、運用支援ツールのロック制御タブにおいて最大使用可能メモリを現在より小さく設定してください。(設定されていない場合は無制限となりますので10～100MB程度に設定してください。)

または、システムのメモリが不足している可能性がありますので、システムの仮想メモリのサイズを大きくしてください。

533(0x0215)

IFAS ファイルをアクセスするための作業領域が確保できません。

上記ステータス以外の内部詳細ステータスが返却された場合はIFAS FILEの内部障害が発生した可能性があるため速やかに製品窓口までご連絡ください。

IFABTLCK = 1003(0x03eb)

意味 このファイルは1度リカバリ処理に失敗したファイルです。

処置 初期化してデータを作り直すか、リカバリーユーティリティを実行してファイルを復旧します。

IFMMEXER = 1004(0x03ec)

意味 メモリの確保に失敗しました。

処置 終了処理を行ってからアプリケーションを終了させてください。

IFERRFIL = 1005(0x03ed)

意味 エラーが発生したファイルに対しアクセス要求を行いました。

処置 該当ファイルをクローズしてください。

IFITNLER = 1006(0x???03ee)

意味 IFAS FILEの内部処理エラーを検出しました。

詳細 本ステータスは内部詳細ステータスを伴って返却されます。形式はステータス領域の右16ビットに“1006(0x03ee)”を、左16ビットに内部詳細ステータスを返却します。

内部詳細ステータスに以下のような値が返却された場合は、ファイル内のファイル管理情報が不正になっています。エラーが発生したファイルを作り直してください。なお、索引順編成ファイルの場合はインデックスメンテナンスでインデックスの再編成を行うと修復する場合があります。

506(0x01fa)

インデックス内のブロック番号が正しくありません。

520(0x0208)

最大ブロック番号を超えるアクセスを行いました。

上記ステータス以外の内部詳細ステータスが返却された場合はIFAS FILEの内部障害が発生した可能性があるため速やかに製品窓口までご連絡ください。

IFFLEXER = 1007(0x03ef)

意味 ファイルサイズの拡張に失敗しました。

処置 該当ファイルをクローズし、ディスクの空き容量を確認してください。



IFIXLVER = 1008(0x03f0)

意味 インデックスツリーレベルがオーバーフローしました。

処置 インデックスブロックサイズを拡張してファイルを再作成してください。

IFNOCLSF = 1009(0x03f1)

意味 このファイルは終了処理が行われなかったファイルです。

処置 初期化してデータを作り直すか、リセットユーティリティを実行してファイルを復旧します。

IFCLSVR = 1010(0x???03f2)

意味 IFAS FILEクライアントのエラーが発生しました。

詳細 本ステータスは内部詳細ステータスを伴って返却されます。形式はステータス領域の右16ビットに“1010(0x03f2)”を、左16ビットに以下のような内部詳細ステータスを返却します。

52003(0xcb23)	同時接続ホスト数が最大値を超えました
52004(0xcb24)	ホスト管理テーブルのアクセスエラーが発生しました
52005(0xcb25)	同時オープンファイル数が最大値を超えました
52006(0xcb26)	ファイル管理テーブルのアクセスエラーが発生しました
52052(0xcb54)	メモリの確保に失敗しました
52053(0xcb55)	Cインターフェース用DLLロードに失敗しました
52054(0xcb56)	COBOLインターフェース用DLLロードに失敗しました
52055(0xcb57)	以下のような原因で接続エラーが発生しました ・サーバ名の指定が誤っている ・サーバマシンの電源が落ちている ・その他
52056(0xcb58)	サーバが起動していないため接続エラーが発生しました
52057(0xcb59)	サーバ使用権がないため接続エラーが発生しました
52058(0xcb5a)	セマフォタイムアウトにより接続エラーが発生しました
52059(0xcb5b)	その他の原因により接続エラーが発生しました
52099(0xcb83)	送受信エラーが発生しました
52100(0xcb84)	インデックスファイルとデータファイルが別ホストです
52101(0xcb85)	SG情報が不正です
52102(0xcb86)	同時起動スレッド数が最大値を超えました
52103(0xcb87)	スレッド管理テーブルのアクセスエラーが発生しました
52104(0xcb88)	更新ログファイル名が不正です
52200(0xcbe8)	クラスタシステムにおいてフェイルオーバが発生しました
52201(0xcbe9)	クラスタシステムにおいてクラスタダウンが発生しました
52202(0xcbea)	Windows95/98/Me未対応機能です
52203(0xcbeb)	レコードサイズが不正です

IFSHBFR = 1011(0x03f3)

意味 共有バッファの確保に失敗しました。

処置 IFAS FILEを使用するプログラムをすべて終了し、共有バッファサイズの設定をやり直してください。

IFVERERR = 1012(0x03f4)

意味 旧バージョンのIFAS FILEとの競合が発生しました。

処置 IFAS FILEのインストールを正しくやり直してください。

IFTIMLIM = 1013(0x03f5)

意味 試用版IFAS FILEの期限が切れています。

処置 正式版IFAS FILEをインストールしてください。

IFNODISC = 1014(0x03f6)

意味 ディスク媒体が準備されていません。  
処置 ディスク媒体を準備して再実行してください。

IFPRTECT = 1015(0x03f7)

意味 ディスク媒体が書き込み禁止になっています。  
処置 書き込み禁止を解除するか媒体を交換してください。

IFFAILOV = 1016(0x03f8)

意味 クラスタシステムにおいてフェイルオーバが発生しました。  
処置 アプリケーションを終了して再実行します。

IFFACLSF = 1017(0x03f9)

意味 このファイルは終了処理に失敗したファイルです。  
処置 初期化してファイルを作り直すか、ファイルロックリセットを行ってください。

IFDRDBER = 1101(0x044d)

意味 ファイル識別子管理ファイルのI/Oエラーが発生しました。  
処置 IFAS FILEを使用するプログラムをすべて終了し、ファイル識別子管理ファイル(drs svc.db)を削除して、IFAS FILE初期化プロセスを再起動してください。

IFINTPRC = 1201(0x04b1)

意味 IFAS FILE初期化プロセスが動作していません。  
処置 IFAS FILE初期化プロセスを起動してください。

IFLRCERR = 1202(0x04b2)

意味 ロック処理で回復不可能なエラーを検出しました。  
処置 IFAS FILEを使用するプログラムをすべて終了し、IFAS FILE初期化プロセスを再起動してください。

IFLKTOVF = 1203(0x04b3)

意味 ロックテーブルオーバーフローが発生しました。  
処置 IFAS FILEを使用するプログラムをすべて終了し、運用支援ツールでロックテーブルサイズを最適値に変更してください。

IFSGIFER = 1204(0x04b4)

意味 IFAS FILEのSG情報が不正です。  
処置 運用支援ツールで設定の矛盾を訂正してください。

IFPRTCER = 1205(0x04b5)

意味 IFAS FILEを使用できない環境です。  
処置 IFAS FILEを正しくインストールしてください。

IFLGIOER = 1701(0x06a5)

意味 更新ログファイルのI/Oエラーが発生しました。  
処置 同一の更新ログファイルを使用するプログラムをすべて終了し、IFAS ファイルのバックアップを行った後、更新ログファイルを削除してください。

IFLGMMER = 1702(0x06a6)

意味 ログマネージャでメモリの確保に失敗しました。  
処置 プログラムを終了して再起動してください。

IFLGFNER = 1703(0x06a7)

意味 更新ログファイル名が不正です。  
処置 更新ログファイル名を訂正して再実行します。

**IFLGSTOP** = 1704 (0x06a8)

**意味** 更新ログファイルがオーバーフローしました。

**処置** IFAS ファイルのバックアップを行った後、バックアップ支援ユーティリティで更新ログファイルをリセットしてください。

**IFRBCLER** = 1801 (0x0709)

**意味** ロールバックログ収集エラーが発生しました。

**処置** プログラムを終了し、ロールバック対象ファイルのロックリセットを行います。

**IFRBMGER** = 1802 (0x070a)

**意味** ロールバックマネージャでエラーが発生しました。

**処置** プログラムを終了しロールバック対象ファイルのロックリセットを行います。

## A-4 COBOL利用時のステータス

### A-4-1 入出力状態

00	意味	処理は正常に終了しました。
02	意味	処理は正常に終了しましたが、重複キーを検出しました。
10	意味 処置	ファイルの終端(EOF)を検出しました。 レコード再位置付けを行ってから、再実行してください。
14	意味 処置	順呼び出しのREAD/START要求において、最大レコード番号を超える要求が されました(EOFと同等です)。 レコード再位置付けを行ってから、再実行してください。
21	意味 処置	WRITE要求されたレコードの主キーが昇順になっていません。 REWRITE要求で主キーを更新しようとした。 レコードイメージを訂正して再実行してください。
22	意味 処置	WRITE/REWRITE要求において、重複キー許可指定のないインデックスにおい て重複キーを検出しました。 乱呼び出しのREWRITE/DELETE要求において、同一キーのレコードが複数存 在します。 WRITE要求で指定されたレコード番号のレコードは既に存在します。 レコードイメージまたは指定レコード番号を訂正して再実行してください。
23	意味 処置	READ/REWRITE/DELETE/START要求で指定されたレコードが存在しませ ん。 乱呼び出しのREAD/START要求において、最大レコード番号を超える要求が されました。 指定キーまたはレコード番号を訂正して再実行してください。
24	意味 処置	ファイルサイズの拡張に失敗しました。 WRITE/REWRITE/DELETE要求において、最大レコード番号を超える要求がさ れました。 該当ファイルをクローズし、ディスクの空き容量を確認してください。 レコード番号を訂正して再実行してください。
30	意味 処置	書き込み不可ファイルに対するアクセス要求です。 フロッピーディスクがセットされていません フロッピーディスクにWRITE PROTECTがかかっています。 ファイル属性を訂正して再実行してください。 フロッピーディスクを正しくセットしてください。
35	意味 処置	指定されたファイルが存在しません。 指定されたファイルに対応するRFDファイルが存在しません。 ファイル名、DB名または表名を変更して再実行してください。RDB表の場合は 実ファイルが存在することも確認してください。 RFDファイルを作成してから再実行してください。

37	意味	書き込み不可ファイルに対するアクセス要求です。 仮想表に対して使用できないオープンモードが指定されました。
	処置	ファイル属性を訂正して再実行してください。 オープンモードを訂正して再実行してください。
39	意味	IFAS ファイルではありません。 インデックスファイルとデータファイルの組み合わせが不正です。 ファイルのレコード長が定義情報と一致しません。 レコードサイズの指定が不正です。 指定されたキーがIFAS ファイルに存在しません。 RFDファイルのフォーマットが不正です。
	処置	ファイルを初期化してから再実行してください。 レコードサイズまたはキー情報を訂正して再実行してください。 RFDファイルを再作成してから再実行してください。
41	意味	指定されたファイルは既にオープンしています。
	処置	ファイル名を訂正して再実行してください。
42	意味	オープンされていないファイルにアクセス要求がされました。 オープン時に指定したオープンモードでは、使用できない要求です。
	処置	ファイル名を訂正して再実行してください。 オープンモードに適した要求を行うか、要求に適したオープンモードで再オープンしてください。
43	意味	READ要求を行っていないファイルに対し、順呼び出しのREWRITE/DELETE要求がされました。 SELECT中のファイルに対し、乱呼び出しのREWRITE/DELETE要求がされました。
	処置	READ要求を行ってから再実行してください。 SELECT中は順呼び出しを行わないように訂正して再実行してください。
46	意味	カレントレコードが設定されていないファイルに対し、順呼び出しのREAD/START要求がされました。
	処置	レコードの再位置付けを行ってから再実行してください。
47	意味	オープンされていないファイルにREAD/START/SELECT/SCRATCH要求がされました。 オープン時に指定したオープンモードでは、READ/START/SELECT/SCRATCH要求を行うことはできません。
	処置	ファイル名を訂正して再実行してください。 オープンモードに適した要求を行うか、要求に適したオープンモードで再オープンしてください。
48	意味	オープンされていないファイルにWRITE要求がされました。 オープン時に指定したオープンモードでは、WRITE要求を行うことはできません。
	処置	ファイル名を訂正して再実行してください。 オープンモードに適した要求を行うか、要求に適したオープンモードで再オープンしてください。

49	意味	オープンされていないファイルにREWRITE/DELETE要求がされました。 オープン時に指定したオープンモードでは、REWRITE/DELETE要求を行うことはできません。
	処置	ファイル名を訂正して再実行してください。 オープンモードに適した要求を行うか、要求に適したオープンモードで再オープンしてください。
91	意味	IFAS ファイルまたはDBシステムがビジーです。 IFASシステムがバックアップ中です。
	処置	しばらく待ってから再実行してください。 バックアップの終了を待って再実行してください。
92	意味	READ/START要求で指定された副キーがファイルに存在しません。 現在のオープンモードではSELECT要求を行うことはできません。 SELECTしていないファイルに対してSCRATCH要求を行うことはできません。
	処置	キー情報を訂正して再実行してください。 オープンモードに適した要求を行うか、要求に適したオープンモードで再オープンしてください。 SELECTを行ってからSCRATCHを行うように訂正して再実行してください。
96	意味	ファイル編成が誤っています。
	処置	ファイル名を訂正して再実行してください。
99	意味	ファイル名の指定が誤っています。
	処置	ファイル名を訂正して再実行してください。
9F	意味	フェイルオーバーが発生しました。 フェイルオーバー処理中です。
	処置	アプリケーションを終了してから再実行してください。
9J	意味	デッドロックが発生しました。
	処置	トランザクション処理中の場合、 ROLLBACK要求を行い、トランザクションを取り消してください。 トランザクション処理中でない場合、 ファイルをクローズしてから再オープンしてください。
9K	意味	ロールバックログ収集エラーです。 ロールバックマネージャのエラーです。
	処置	プログラムを終了し、ロールバック対象ファイルのロックリセットを行います。
9M	意味	RDB管理ファイルのアクセスエラー(DB破壊)です。
	処置	RDB管理ファイルが壊れている可能性があるので、RDB表を使用するプログラムをすべて終了し、RDB管理ファイルを復旧してください。
9O	意味	更新ログファイルに空きスペースがないため、ファイルの更新ができません。
	処置	IFAS ファイルのバックアップを行った後、バックアップ支援ユーティリティで更新ログファイルをリセットしてください。
9P	意味	ロックテーブルのオーバーフローが発生しました。
	処置	IFAS ファイルを使用するプログラムをすべて終了し、運用支援ツールでロックテーブルサイズを最適値に変更してください。
9T	意味	更新ログファイルのI/Oエラーです。
	処置	同一の更新ログファイルを使用するプログラムをすべて終了し、IFAS ファイルのバックアップを行った後、更新ログファイルを削除してください。

0xffff0	意味	パラメータエラーです。 オープンモード／アクセスモードなどの指定で、指定できない値が指定されている可能性があります。
	処置	各要求で指定する内容を訂正して再実行してください。
0xfffff	意味	内部エラーです。
	処置	詳しくは「付録A-4-2 詳細ステータス」を参照してください。

## A-4-2 詳細ステータス

### 50010(0xc35a)

意味	IFAS ファイル名が不正です。
処置	IFAS ファイル名を正しく設定してください。

### 50011(0xc35b)

意味	指定したレコード長が指定可能最大値を超えています。
処置	正しい値を指定してください。

### 50012(0xc35c)

意味	指定レコード長が既存ファイルのレコード長と一致しません。
処置	指定レコード長は既存ファイルのレコード長を指定してください。

### 50013(0xc35d)

意味	オープンモードまたはアクセスモードが不正です。
処置	オープンモードまたはアクセスモードを正しく指定してください。

### 50014(0xc35e)

意味	主キー情報に二重キー指定があります。
処置	二重キー指定のキーは主キーとして指定できません。主キーを変更してください。

### 50015(0xc35f)

意味	ファイルテーブルの登録に失敗しました。
処置	メモリ確保に失敗しました。終了処理を行ってからアプリケーションを終了させてください。

### 50016(0xc360)

意味	キーテーブル用のメモリ確保に失敗しました。
処置	終了処理を行ってからアプリケーションを終了させてください。

### 50018(0xc362)

意味	最大キー用のメモリ確保に失敗しました。
処置	終了処理を行ってからアプリケーションを終了させてください。

### 50019(0xc363)

意味	最大キー用のメモリ解放に失敗しました。
処置	終了処理を行ってからアプリケーションを終了させてください。

### 50020(0xc364)

意味	キー情報が不一致です(プライマリインデックスがあるファイルに対して、サブインデックスを主キーとして指定しました)。
処置	主キーにはプライマリインデックスを指定してください。

**50022(0xc366)**

**意味** キー変換に失敗しました(二重キー／キーサイズの指定が不正です)。  
**処置** キーに正しい値を指定してください。

**50023(0xc367)**

**意味** キー変換に失敗しました。  
**処置** キーが正しく指定されていない可能性があります。正しい値を指定してください。

**50024(0xc368)**

**意味** 主キー情報の比較に失敗しました。  
**処置** キーが正しく指定されていない可能性があります。正しい値を指定してください。

**50025(0xc369)**

**意味** キーテーブルの作成に失敗しました。  
**処置** メモリ確保に失敗しました。終了処理を行ってからアプリケーションを終了させてください。

**50026(0xc36a)**

**意味** 指定キー情報がファイルに見つかりません。  
**処置** 正しいキーを指定してください。

**50027(0xc36b)**

**意味** 主キー値の更新に失敗しました。  
**処置** キーが正しく指定されていない可能性があります。正しい値を指定してください。

**50028(0xc36c)**

**意味** カレントキー情報の更新に失敗しました。  
**処置** キーが正しく指定されていない可能性があります。正しい値を指定してください。

**50029(0xc36d)**

**意味** 指定キー情報のインデックス属性が不正です  
**処置** インデックスが不整合となっています。インデックスメンテナンスでインデックスを再編成してください。

**50030(0xc36e)**

**意味** ファイルテーブルの削除に失敗しました。  
**処置** メモリ解放に失敗しました。終了処理を行ってからアプリケーションを終了させてください。

**50032(0xc370)**

**意味** 直前キー用のメモリ解放に失敗しました。  
**処置** 終了処理を行ってからアプリケーションを終了させてください。

**50033(0xc371)**

**意味** キー情報用のメモリ解放に失敗しました。  
**処置** 終了処理を行ってからアプリケーションを終了させてください。

**50034(0xc372)**

**意味** 現在のオープンモードで、本要求は実行できません。  
**処置** オープンモードを正しく指定して、IFAS ファイルを再オープンしてください。

**50035(0xc373)**

**意味** カレントレコードが存在しません。  
**処置** READ(NEXT/PRIOR以外)要求等でレコードを確保した後、再度実行してください。



**50036(0xc374)**

- 意味** 実際のレコードサイズより小さいレコードサイズが指定されました。  
**処置** レコードサイズは、IFAS ファイルのレコードサイズ以上の値を指定してください。

**50040(0xc378)**

- 意味** ロックモードが不正です。  
**処置** 正しい値を指定してください。

**50041(0xc379)**

- 意味** アクセスモードが不正です。  
**処置** 正しい値を指定してください。

**50042(0xc37a)**

- 意味** 指定したレコード長が指定可能最大値を超えています。  
**処置** 正しい値を指定してください。

**50043(0xc37b)**

- 意味** キー情報が不正です。  
**処置** キーが正しく指定されていない可能性があります。正しい値を指定してください。

**50044(0xc37c)**

- 意味** アクセスモードが不一致です。  
**処置** 正しい値を指定してください。

**50045(0xc37d)**

- 意味** カレントキー情報の更新が不正です。  
**処置** キーが正しく指定されていない可能性があります。正しい値を指定してください。

**50046(0xc37e)**

- 意味** 主キー値の更新に失敗しました。  
**処置** キーが正しく指定されていない可能性があります。正しい値を指定してください。

**50047(0xc37f)**

- 意味** セレクト中のため、乱アクセスは指定できません。  
**処置** 順アクセス指定で再実行してください。

**50050(0xc382)**

- 意味** 指定したレコード長が指定可能最大値を超えています。  
**処置** 正しい値を指定してください。

**50051(0xc383)**

- 意味** 指定レコード内のキー値の順序が不正です。  
**処置** 順アクセスの場合、主キー値は昇順になるように設定しなければなりません。正しく指定してください。

**50052(0xc384)**

- 意味** 主キー値の更新に失敗しました。  
**処置** キーが正しく指定されていない可能性があります。正しい値を指定してください。

**50053(0xc385)**

- 意味** 現在のオープンモードで、本要求は実行できません。  
**処置** オープンモードを正しく指定して、IFAS ファイルを再オープンしてください。

**50054(0xc386)**

**意味** 実際のレコードサイズより小さいレコードサイズが指定されました  
**処置** レコードサイズは、IFAS ファイルのレコードサイズ以上の値を指定してください。

**50061(0xc38d)**

**意味** 指定したレコード長が指定可能最大値を超えています。  
**処置** 正しい値を指定してください。

**50062(0xc38e)**

**意味** キー情報が不正です。  
**処置** キーが正しく指定されていない可能性があります。正しい値を指定してください。

**50063(0xc38f)**

**意味** アクセスモードが不一致です。  
**処置** 正しい値を指定してください。

**50064(0xc390)**

**意味** カレントレコードが読み込まれていません。  
**処置** READ(NEXT/PRIOR以外)要求等でレコードを確保した後、再度実行してください。

**50066(0xc392)**

**意味** 主キー値を更新しようとしてしました。  
**処置** 順アクセスの場合主キー値は更新できません。正しい組み合わせで再実行してください。

**50067(0xc393)**

**意味** セレクト中のため、キーアクセスは指定できません。  
**処置** 順アクセス指定で再実行してください。

**50068(0xc394)**

**意味** 現在のオープンモードで、本要求は実行できません。  
**処置** オープンモードを正しく指定して、IFAS ファイルを再オープンしてください。

**50069(0xc395)**

**意味** カレントレコードが存在しません。  
**処置** READ(NEXT/PRIOR以外)要求等でレコードを確保した後、再度実行してください。

**50070(0xc396)**

**意味** 実際のレコードサイズより小さいレコードサイズが指定されました。  
**処置** レコードサイズは、IFAS ファイルのレコードサイズ以上の値を指定してください。

**50071(0xc397)**

**意味** キー情報が不正です。  
**処置** キーが正しく指定されていない可能性があります。正しい値を指定してください。

**50072(0xc398)**

**意味** アクセスモードが不正です。  
**処置** 正しい値を指定してください。

**50073(0xc399)**

**意味** カレントレコードが読み込まれていません。  
**処置** READ(NEXT/PRIOR以外)要求等でレコードを確保した後、再度実行してください。

**50074(0xc39a)**

意味 セレクト中のため、キーアクセスは指定できません。  
処置 順アクセス指定で再実行してください。

**50075(0xc39b)**

意味 現在のオープンモードで、本要求は実行できません。  
処置 オープンモードを正しく指定して、IFAS ファイルを再オープンしてください。

**50076(0xc39c)**

意味 カレントレコードが存在しません。  
処置 READ(NEXT/PRIOR以外)要求等でレコードを確保した後、再度実行してください。

**50080(0xc3a0)**

意味 アクセスモードが不正です。  
処置 正しい値を指定してください。

**50081(0xc3a1)**

意味 キー情報が不正です。  
処置 キーが正しく指定されていない可能性があります。正しい値を指定してください。

**50082(0xc3a2)**

意味 カレントキー情報の更新が不正です。  
処置 キーが正しく指定されていない可能性があります。正しい値を指定してください。

**50083(0xc3a3)**

意味 セレクト中のため、乱アクセスは指定できません。  
処置 順アクセス指定で再実行してください。

**50084(0xc3a4)**

意味 現在のオープンモードで、本要求は実行できません。  
処置 オープンモードを正しく指定して、IFAS ファイルを再オープンしてください。

**50085(0xc3a5)**

意味 カレントレコードが存在しません。  
処置 READ(NEXT/PRIOR以外)要求等でレコードを確保した後、再度実行してください。

**50090(0xc3aa)**

意味 ファイル名が不正です。  
処置 正しい値を指定してください。

**50091(0xc3ab)**

意味 指定したレコード長が指定可能最大値を超えています。  
処置 正しい値を指定してください。

**50092(0xc3ac)**

意味 オープン/アクセスモードが不正です。  
処置 正しい値を指定してください。

**50093(0xc3ad)**

意味 主キー情報に二重キー指定があります。  
処置 二重キー指定のキーは主キーとして指定できません。主キーを変更してください。

**50094(0xc3ae)**

**意味** キー変換に失敗しました(二重キー／キーサイズの指定が不正です)。  
**処置** キーに正しい値を指定してください。

**50095(0xc3af)**

**意味** キー変換に失敗しました。  
**処置** キーが正しく指定されていない可能性があります。正しい値を指定してください。

**50096(0xc3b0)**

**意味** キー変換に失敗しました(二重キー／キーサイズ／キータイプの指定が不正です)。  
**処置** キーに正しい値を指定してください。

**50097(0xc3b1)**

**意味** 主キー情報の比較に失敗しました。  
**処置** キーが正しく指定されていない可能性があります。正しい値を指定してください。

**50098(0xc3b2)**

**意味** キーテーブルの作成に失敗しました。  
**処置** メモリ確保に失敗しました。終了処理を行ってからアプリケーションを終了させてください。

**50099(0xc3b3)**

**意味** 指定キー情報がファイルに見つかりません。  
**処置** 主キーに正しい値を設定してください。

**50100(0xc3b4)**

**意味** ファイルテーブルの登録に失敗しました。  
**処置** メモリ確保に失敗しました。終了処理を行ってからアプリケーションを終了させてください。

**50101(0xc3b5)**

**意味** キーテーブル用のメモリ確保に失敗しました。  
**処置** 終了処理を行ってからアプリケーションを終了させてください。

**50102(0xc3b6)**

**意味** カレントキー情報の更新に失敗しました。  
**処置** キーが正しく指定されていない可能性があります。正しい値を指定してください。

**50103(0xc3b7)**

**意味** レコードのデータタイプが不正です。  
**処置** 正しい値を指定してください。

**50104(0xc3b8)**

**意味** レコードが存在するためファイルが削除できません。  
**処置** レコードを削除する等の処理を行ってください。

**50110(0xc3be)**

**意味** ロックモードが不正です。  
**処置** 正しい値を指定してください。

**50111(0xc3bf)**

**意味** アクセスモードが不正です。  
**処置** 正しい値を指定してください。

**50112(0xc3c0)**

意味 指定したレコード長が指定可能最大値を超えています。  
処置 正しい値を指定してください。

**50113(0xc3c1)**

意味 アクセスモードが不正です。  
処置 正しい値を指定してください。

**50114(0xc3c2)**

意味 セレクト中のため、乱アクセスは指定できません。  
処置 順アクセス指定で再実行してください。

**50115(0xc3c3)**

意味 現在のオープンモードで、本要求は実行できません。  
処置 オープンモードを正しく指定して、IFAS ファイルを再オープンしてください。

**50116(0xc3c4)**

意味 カレントレコードが存在しません。  
処置 READ(NEXT/PRIOR以外)要求等でレコードを確保した後、再度実行してください。

**50117(0xc3c5)**

意味 実際のレコードサイズより小さいレコードサイズが指定されました。  
処置 レコードサイズは、IFAS ファイルのレコードサイズ以上の値を指定してください。

**50120(0xc3c8)**

意味 指定したレコード長が指定可能最大値を超えています。  
処置 正しい値を指定してください。

**50121(0xc3c9)**

意味 現在のオープンモードで、本要求は実行できません。  
処置 オープンモードを正しく指定して、IFAS ファイルを再オープンしてください。

**50122(0xc3ca)**

意味 実際のレコードサイズより小さいレコードサイズが指定されました。  
処置 レコードサイズは、IFAS ファイルのレコードサイズ以上の値を指定してください。

**50130(0xc3d2)**

意味 指定したレコード長が指定可能最大値を超えています。  
処置 正しい値を指定してください。

**50131(0xc3d3)**

意味 アクセスモードが不正です。  
処置 正しい値を指定してください。

**50132(0xc3d4)**

意味 カレントレコードが存在しません。  
処置 READ(NEXT/PRIOR以外)要求等でレコードを確保した後、再度実行してください。

**50133(0xc3d5)**

意味 アクセスモードが不正です。  
処置 正しい値を指定してください。

**50134(0xc3d6)**

意味 セレクト中のため、キーアクセスは指定できません。

処置 順アクセス指定で再実行してください。

**50135(0xc3d7)**

意味 現在のオープンモードで、本要求は実行できません。

処置 オープンモードを正しく指定して、IFAS ファイルを再オープンしてください。

**50136(0xc3d8)**

意味 カレントレコードが存在しません。

処置 READ(NEXT/PRIOR以外)要求等でレコードを確保した後、再度実行してください。

**50137(0xc3d9)**

意味 実際のレコードサイズより小さいレコードサイズが指定されました。

処置 レコードサイズは、IFAS ファイルのレコードサイズ以上の値を指定してください。

**50140(0xc3dc)**

意味 アクセスモードが不正です。

処置 正しい値を指定してください。

**50141(0xc3dd)**

意味 カレントレコードが存在しません。

処置 READ(NEXT/PRIOR以外)要求等でレコードを確保した後、再度実行してください。

**50142(0xc3de)**

意味 アクセスモードが不正です。

処置 正しい値を指定してください。

**50143(0xc3df)**

意味 セレクト中のため、キーアクセスは指定できません。

処置 順アクセス指定で再実行してください。

**50144(0xc3e0)**

意味 現在のオープンモードで、本要求は実行できません。

処置 オープンモードを正しく指定して、IFAS ファイルを再オープンしてください。

**50145(0xc3e1)**

意味 カレントレコードが存在しません。

処置 READ(NEXT/PRIOR以外)要求等でレコードを確保した後、再度実行してください。

**50150(0xc3e6)**

意味 アクセスモードが不正です。

処置 正しい値を指定してください。

**50151(0xc3e7)**

意味 セレクト中のため、乱アクセスは指定できません。

処置 順アクセス指定で再実行してください。

**50152(0xc3e8)**

意味 現在のオープンモードで、本要求は実行できません。

処置 オープンモードを正しく指定して、IFAS ファイルを再オープンしてください。

**50160(0xc3f0)**

**意味** 現在のオープンモードで、本要求は実行できません。  
**処置** オープンモードを正しく指定して、IFAS ファイルを再オープンしてください。

**50161(0xc3f1)**

**意味** メモリ解放に失敗しました。  
**処置** 終了処理を行ってからアプリケーションを終了させてください。

**50162(0xc3f2)**

**意味** メモリ解放に失敗しました。  
**処置** 終了処理を行ってからアプリケーションを終了させてください。

**50163(0xc3f3)**

**意味** 仮想インデックスファイルが存在しません。  
**処置** ファイルを再オープンしてから、再実行してください。

**50164(0xc3f4)**

**意味** 列情報が存在しません。  
**処置** 正しい列を指定してください。

**50165(0xc3f5)**

**意味** アクセスモードが不正です。  
**処置** 正しい値を指定してください。

**50168(0xc3f8)**

**意味** プライマリインデックスを設定したファイルでSELECT要求は使用できません。  
**処置** 処理手順を見直してください。

**50169(0xc3f9)**

**意味** ソート項目に17個以上の項目が指定されました。  
**処置** ソート項目の指定は16個までです。

**50170(0xc3fa)**

**意味** SELECTされていません。  
**処置** 処理手順を見直してください。

**50171(0xc3fb)**

**意味** 仮想インデックスファイルが存在しません。  
**処置** ファイルを再オープンしてから、再実行してください。

**50172(0xc3fc)**

**意味** 現在のオープンモードで、本要求は実行できません。  
**処置** オープンモードを正しくして、IFAS ファイルを再オープンしてください。

**50173(0xc3fd)**

**意味** プライマリインデックスを設定したファイルでSCRATCH要求は使用できません。  
**処置** 処理手順を見直してください。

**50180(0xc404)**

**意味** アクセスモードが不正です。  
**処置** 正しい値を指定してください。

**50181(0xc405)**

**意味** 作業用領域のメモリ確保に失敗しました。  
**処置** 終了処理を行ってからアプリケーションを終了させてください。

**50182(0xc406)**

**意味** カレントレコードが存在しません。  
**処置** READ(NEXT/PRIOR以外)要求等でレコードを確保した後、再度実行してください。

**50190(0xc40e)**

**意味** ファイル名が不正です。  
**処置** 正しい値を指定してください。

**50191(0xc40f)**

**意味** インデックスの指定が不正です。  
**処置** 正しい値を指定してください。

**50192(0xc410)**

**意味** キー変換に失敗しました(二重キー/キーサイズの指定が不正です)。  
**処置** キーに正しい値を指定してください。

**50193(0xc411)**

**意味** RFDファイルのオープンに失敗しました。  
**処置** RFD編集ツール等でRFDファイルを再作成した後、再実行してください。

**50194(0xc412)**

**意味** RFDファイルが存在しません。  
**処置** RFD編集ツール等でRFDファイルを作成した後、再実行してください。

**50195(0xc413)**

**意味** RFDファイルのフォーマットが不正です。  
**処置** RFD編集ツール等でRFDファイルを再作成した後、再実行してください。

**50196(0xc414)**

**意味** RFDファイルの入力に失敗しました。  
**処置** RFD編集ツール等でRFDファイルを再作成した後、再実行してください。

**50197(0xc415)**

**意味** RFDファイルの内部処理でエラーが発生しました。  
**処置** RFD編集ツール等でRFDファイルを再作成した後、再実行してください。

**50198(0xc416)**

**意味** RFDファイルのクローズに失敗しました。  
**処置** RFD編集ツール等でRFDファイルを再作成した後、再実行してください。

**50199(0xc417)**

**意味** メモリ確保に失敗しました。  
**処置** 終了処理を行ってからアプリケーションを終了させてください。

**50280(0xc468)**

**意味** ファイル名が不正です。  
**処置** 正しい値を指定してください。

**50281(0xc469)**

**意味** RFDファイルのオープンに失敗しました。  
**処置** RFD編集ツール等でRFDファイルを再作成した後、再実行してください。



**50282(0xc46a)**

意味 RFDファイルが存在しません。

処置 RFD編集ツール等でRFDファイルを作成した後、再実行してください。

**50283(0xc46b)**

意味 RFDファイルのフォーマットが不正です。

処置 RFD編集ツール等でRFDファイルを再作成した後、再実行してください。

**50284(0xc46c)**

意味 RFDファイルの入力に失敗しました。

処置 RFD編集ツール等でRFDファイルを再作成した後、再実行してください。

**50285(0xc46d)**

意味 メモリ解放に失敗しました。

処置 終了処理を行ってからアプリケーションを終了させてください。

**50286(0xc46e)**

意味 RFDファイルのクローズに失敗しました。

処置 RFD編集ツール等でRFDファイルを再作成した後、再実行してください。

**50287(0xc46f)**

意味 IFAS ファイル用に作成されたRFDファイルではありません。

処置 RFD編集ツール等でRFDファイルを再作成した後、再実行してください。

**50289(0xc471)**

意味 RFDファイルの内部処理でエラーが発生しました。

処置 RFD編集ツール等でRFDファイルを再作成した後、再実行してください。

**50290(0xc472)**

意味 ファイル名が不正です。

処置 正しい値を指定してください。

**50291(0xc473)**

意味 RFDファイルのオープンに失敗しました。

処置 RFD編集ツール等でRFDファイルを再作成した後、再実行してください。

**50292(0xc474)**

意味 RFDファイルが存在しません。

処置 RFD編集ツール等でRFDファイルを作成した後、再実行してください。

**50293(0xc475)**

意味 RFDファイルのフォーマットが不正です。

処置 RFD編集ツール等でRFDファイルを再作成した後、再実行してください。

**50294(0xc476)**

意味 RFDファイルの入力に失敗しました。

処置 RFD編集ツール等でRFDファイルを再作成した後、再実行してください。

**50295(0xc477)**

意味 メモリ解放に失敗しました。

処置 終了処理を行ってからアプリケーションを終了させてください。

**50296(0xc478)**

**意味** RFDファイルのクローズに失敗しました。  
**処置** RFD編集ツール等でRFDファイルを再作成した後、再実行してください。

**50297(0xc479)**

**意味** IFAS ファイル用に作成されたRFDファイルではありません。  
**処置** RFD編集ツール等でRFDファイルを再作成した後、再実行してください。

**50299(0xc47b)**

**意味** RFDファイルの内部処理でエラーが発生しました。  
**処置** RFD編集ツール等でRFDファイルを再作成した後、再実行してください。

**50300(0xc47c)**

**意味** メモリ確保に失敗しました。  
**処置** 終了処理を行ってからアプリケーションを終了させてください。

**50301(0xc47d)**

**意味** ファイル情報を格納するエリアのサイズが不正です。  
**処置** 正しいサイズが返却されます。正しい値を指定して再実行してください。

**50310(0xc486)**

**意味** RFDファイルのオープンに失敗しました。  
**処置** RFD編集ツール等でRFDファイルを再作成した後、再実行してください。

**50311(0xc487)**

**意味** RFDファイルが存在しません。  
**処置** RFD編集ツール等でRFDファイルを作成した後、再実行してください。

**50312(0xc488)**

**意味** RFDファイルのフォーマットが不正です。  
**処置** RFD編集ツール等でRFDファイルを再作成した後、再実行してください。

**50313(0xc489)**

**意味** RFDファイルの入力に失敗しました。  
**処置** RFD編集ツール等でRFDファイルを再作成した後、再実行してください。

**50314(0xc48a)**

**意味** RFDファイルの内部処理でエラーが発生しました。  
**処置** RFD編集ツール等でRFDファイルを再作成した後、再実行してください。

**50315(0xc48b)**

**意味** RFDファイルのクローズに失敗しました。  
**処置** RFD編集ツール等でRFDファイルを再作成した後、再実行してください。

**50316(0xc48c)**

**意味** メモリ確保に失敗しました。  
**処置** 終了処理を行ってからアプリケーションを終了させてください。

**50521(0xc559)**

**意味** アクセスモードが正しく指定されていません。  
**処置** アクセスモードを正しく指定してください。

**50522(0xc55a)**

**意味** オープンモードが正しく指定されていません。  
**処置** オープンモードを正しく指定してください。

**50523(0xc55b)**

**意味** 指定レコードサイズと列情報の合計サイズが一致しません。  
**処置** レコードサイズを正しく指定してください。

**50524(0xc55c)**

**意味** インデックス情報が正しく指定されていません。  
**処置** 基本表のインデックス情報を正しく定義してください。

**50525(0xc55d)**

**意味** 列情報が正しく指定されていません。  
**処置** 仮想表の列情報を正しく定義してください。

**50526(0xc55e)**

**意味** 指定された表は既にオープンしています。  
**処置** 同じ表を2度オープンすることはできません。表名を正しく指定してください。

**50527(0xc55f)**

**意味** 主キーとして指定されたキーが存在しません。  
**処置** キー情報を正しく指定するか、基本表にインデックスを定義してください。

**50528(0xc560)**

**意味** オープン情報の登録に失敗しました。  
**処置** メモリ確保に失敗しました。終了処理を行ってからアプリケーションを終了させてください。

**50530(0xc562)**

**意味** メモリ確保に失敗しました。  
**処置** 終了処理を行ってからアプリケーションを終了させてください。

**50531(0xc563)**

**意味** 表定義情報の入手でアクセスエラーです。  
**処置** RDB管理ファイルが壊れている可能性があります。RDB表を使用するプログラムをすべて終了し、RDB管理ファイルを復旧してください。

**50533(0xc565)**

**意味** 指定された表が存在しません。  
**処置** 表名を正しく指定してください。

**50534(0xc566)**

**意味** RDB管理ファイルのアクセスエラーです。(DB破壊)  
**処置** RDB管理ファイルが壊れている可能性があります。RDB表を使用するプログラムをすべて終了し、RDB管理ファイルを復旧してください。

**50535(0xc567)**

**意味** 指定されたDBが存在しません。  
**処置** DB名を正しく指定してください。

**50536(0xc568)**

**意味** データ変換に失敗しました。  
**処置** 列のデータタイプが不正となっている可能性があります。列情報を正しく定義してください。

**50537(0xc569)**

**意味** 仮想表に対して指定できないオープンモードが指定されました。  
**処置** オープンモードを正しく指定してください。

**50538(0xc56a)**

**意味** 仮想表に対して指定できないオープンモードが指定されました。  
**処置** オープンモードを正しく指定してください。

**50540(0xc56c)**

**意味** アクセスキーに対応するインデックスが見つかりません。  
**処置** アクセスキーとして指定したインデックスが削除されている可能性があります。結合型仮想表の定義情報を確認し、インデックスを再定義してください。

**50541(0xc56d)**

**意味** アクセスキーと対応キーの関係が不正です。  
**処置** 結合型仮想表の関係式を正しく定義してください。

**50542(0xc56e)**

**意味** 表定義ユーティリティで作成された仮想表ではないため、アクセスできません。  
**処置** 表定義ユーティリティで仮想表を再定義してください。

**50543(0xc56f)**

**意味** 相対編成アクセスできない仮想表です。  
**処置** 相対編成以外のファイル編成を使用してください。

**50544(0xc570)**

**意味** 各プライマリ表のレコードサイズが一致しません。  
**処置** 合併型仮想表を構成する基本表の列情報を確認し、再定義してください。

**50545(0xc571)**

**意味** 第2プライマリ表以降に主キーとして指定されたキーが存在しません。  
**処置** 合併型仮想表を構成する基本表のインデックス情報を確認し、インデックスを定義してください。

**50546(0xc572)**

**意味** RFDファイルのオープンに失敗しました。  
**処置** 表定義ユーティリティでRFDファイルを再作成した後、再実行してください。

**50547(0xc573)**

**意味** RFDファイルが存在しません。  
**処置** 表定義ユーティリティでRFDファイルを作成した後、再実行してください。

**50548(0xc574)**

**意味** RFDファイルのフォーマットが不正です。  
**処置** 表定義ユーティリティでRFDファイルを再作成した後、再実行してください。

**50549(0xc575)**

**意味** RFDファイルの入力に失敗しました。  
**処置** 表定義ユーティリティでRFDファイルを再作成した後、再実行してください。

**50550(0xc576)**

**意味** RFDファイルの内部処理でエラーが発生しました。  
**処置** 表定義ユーティリティでRFDファイルを再作成した後、再実行してください。

**50551(0xc577)**

**意味** RFDファイルのクローズに失敗しました。  
**処置** 表定義ユーティリティでRFDファイルを再作成した後、再実行してください。

**50561(0xc581)**

**意味** アクセスモードが正しく指定されていません。  
**処置** アクセスモードを正しく指定してください。

**50562(0xc582)**

**意味** ロックモードが正しく指定されていません。  
**処置** ロックモードを正しく指定してください。

**50563(0xc583)**

**意味** 実際のレコードサイズより小さいレコードサイズが指定されました。  
**処置** レコードサイズを正しく指定してください。

**50564(0xc584)**

**意味** 順次呼び出しのファイルに乱アクセスを指定しました。  
**処置** 順アクセスを指定して再実行してください。

**50565(0xc585)**

**意味** 指定されたキーが存在しません。  
**処置** キー情報を正しく指定してください。

**50566(0xc586)**

**意味** オープン時のアクセスモードが不正です。  
**処置** 要求に適したアクセスモードを指定して再オープンしてください。

**50567(0xc587)**

**意味** セレクト中のため、乱アクセスはできません。  
**処置** 順アクセスを指定して再実行してください。

**50569(0xc589)**

**意味** オープンモードとの組み合わせが不正です。  
**処置** 要求に適したオープンモードを指定して再オープンしてください。

**50570(0xc58a)**

**意味** データ変換に失敗しました。  
**処置** 列のデータタイプが不正となっている可能性があります。列情報を正しく定義してください。

**50571(0xc58b)**

**意味** カレントレコードが見つかりません。  
**処置** READ/START要求を行った後、本要求を行うようにしてください。

**50572(0xc58c)**

**意味** セカンダリ表のレコードが存在しません。  
**処置** 指定されたキー値のレコードはプライマリ表には存在しますが、対応キーと一致するレコードがセカンダリ表に見つかりませんでした。  
指定キーまたは指定レコード番号を訂正して再実行してください。

**50573(0xc58d)**

**意味** ロック無視READの宣言／解除に失敗しました。  
**処置** 終了処理を行ってからアプリケーションを終了させてください。

**50574(0xc58e)**

**意味** メモリ確保に失敗しました。  
**処置** 終了処理を行ってからアプリケーションを終了させてください。

**50575(0xc58f)**

**意味** ファイルの終端(EOF)を検出しました。  
**処置** レコード再位置付けを行ってから再実行してください。

**50576(0xc590)**

**意味** 指定されたレコードが存在しません。  
**処置** 指定キーまたは指定レコード番号を訂正して再実行してください。

**50577(0xc591)**

**意味** 第2プライマリ表以降に指定されたキーが存在しません。  
**処置** 合併型仮想表を構成する基本表のインデックス情報を確認し、インデックスを定義してください。

**50581(0xc595)**

**意味** 実際のレコードサイズより小さいレコードサイズが指定されました。  
**処置** レコードサイズを正しく指定してください。

**50582(0xc596)**

**意味** オープン時のアクセスモードが不正です。  
**処置** 要求に適したアクセスモードを指定して再オープンしてください。

**50583(0xc597)**

**意味** 表種別が不正です。  
**処置** 本要求が行えない仮想表です。仮想表の種類を確認し、正しい要求を行うようにしてください。

**50584(0xc598)**

**意味** 主キーが昇順ではありません。  
**処置** レコードイメージを訂正して再実行してください。

**50585(0xc599)**

**意味** オープンモードとの組み合わせが不正です。  
**処置** 要求に適したオープンモードを指定して再オープンしてください。

**50586(0xc59a)**

**意味** データ変換に失敗しました。  
**処置** 列のデータタイプが不正となっている可能性があります。列情報を正しく定義してください。

**50601(0xc5a9)**

**意味** アクセスモードが正しく指定されていません。  
**処置** アクセスモードを正しく指定してください。

**50602(0xc5aa)**

**意味** 実際のレコードサイズより小さいレコードサイズが指定されました。  
**処置** レコードサイズを正しく指定してください。

**50603(0xc5ab)**

**意味** 順次呼び出しのファイルにキーアクセスを指定しました。  
**処置** 順アクセスを指定して再実行してください。

**50604(0xc5ac)**

**意味** 指定されたキーが存在しません。  
**処置** キー情報を正しく指定してください。

**50605(0xc5ad)**

**意味** オープン時のアクセスモードが不正です。  
**処置** 要求に適したオープンモードを指定して再オープンしてください。

**50606(0xc5ae)**

**意味** 表種別が不正です。  
**処置** 本要求が行えない仮想表です。仮想表の種類を確認し、正しい要求を行うようにしてください。

**50607(0xc5af)**

**意味** セレクト中のため、キーアクセスは指定できません。  
**処置** 順アクセスを指定して再実行してください。

**50608(0xc5b0)**

**意味** カレントレコードが存在しません。  
**処置** READ/START要求を行った後、本要求を行うようにしてください。

**50609(0xc5b1)**

**意味** 主キーを更新しようとしてしました。  
**処置** レコードイメージを訂正して再実行してください。

**50610(0xc5b2)**

**意味** オープンモードとの組み合わせが不正です。  
**処置** 要求に適したオープンモードを指定して再オープンしてください。

**50611(0xc5b3)**

**意味** データ変換に失敗しました。  
**処置** 列のデータタイプが不正となっている可能性があります。列情報を正しく定義してください。

**50612(0xc5b4)**

**意味** 同一キーのレコードが複数存在するため更新できません。  
**処置** レコードイメージを訂正して再実行してください。

**50613(0xc5b5)**

**意味** 指定されたレコードは存在しません。  
**処置** 指定キーまたはレコード番号を訂正して再実行してください。

**50614(0xc5b6)**

**意味** メモリ確保に失敗しました。  
**処置** 終了処理を行ってからアプリケーションを終了させてください。

**50621(0xc5bd)**

**意味** アクセスモードが正しく指定されていません。  
**処置** アクセスモードを正しく指定してください。

**50622(0xc5be)**

**意味** 順次呼び出しのファイルにキーアクセスを指定しました。  
**処置** 順アクセスを指定して再実行してください。

**50623(0xc5bf)**

**意味** 指定されたキーが存在しません。  
**処置** キー情報を正しく指定してください。

**50624(0xc5c0)**

**意味** オープン時のアクセスモードが不正です。  
**処置** 要求に適したアクセスモードを指定して再オープンしてください。

**50625(0xc5c1)**

**意味** 表種別が不正です。  
**処置** 本要求が行えない仮想表です。仮想表の種類を確認し、正しい要求を行うようにしてください。

**50626(0xc5c2)**

**意味** セレクト中のため、キーアクセスは指定できません。  
**処置** 順アクセスを指定して再実行してください。

**50627(0xc5c3)**

**意味** カレントレコードが存在しません。  
**処置** READ/START 要求を行った後、本要求を行うようにしてください。

**50628(0xc5c4)**

**意味** オープンモードとの組み合わせが不正です。  
**処置** 要求に適したオープンモードを指定して再オープンしてください。

**50629(0xc5c5)**

**意味** データ変換に失敗しました。  
**処置** 列のデータタイプが不正となっている可能性があります。列情報を正しく定義してください。

**50630(0xc5c6)**

**意味** 同一キーのレコードが複数存在するため削除できません。  
**処置** レコードイメージを訂正して再実行してください。

**50631(0xc5c7)**

**意味** 指定されたレコードは存在しません。  
**処置** 指定キーまたはレコード番号を訂正して再実行してください。

**50632(0xc5c8)**

**意味** メモリ確保に失敗しました。  
**処置** 終了処理を行ってからアプリケーションを終了させてください。

**50641(0xc5d1)**

**意味** アクセスモードが正しく指定されていません。  
**処置** アクセスモードを正しく指定してください。

**50642(0xc5d2)**

**意味** 順次呼び出しのファイルに乱アクセスを指定しました。  
**処置** 順アクセスを指定して再実行してください。

**50643(0xc5d3)**

**意味** 指定されたキーが存在しません。  
**処置** キー情報を正しく指定してください。



**50644(0xc5d4)**

**意味** オープン時のアクセスモードが不正です。  
**処置** 要求に適したアクセスモードを指定して再オープンしてください。

**50645(0xc5d5)**

**意味** セレクト中のため、乱アクセスはできません。  
**処置** 順アクセスを指定して再実行してください。

**50646(0xc5d6)**

**意味** オープンモードとの組み合わせが不正です。  
**処置** 要求に適したオープンモードを指定して再オープンしてください。

**50647(0xc5d7)**

**意味** データ変換に失敗しました。  
**処置** 列のデータタイプが不正となっている可能性があります。列情報を正しく定義してください。

**50648(0xc5d8)**

**意味** カレントレコードが見つかりません。  
**処置** READ/START要求を行った後、本要求を行うようにしてください。

**50649(0xc5d9)**

**意味** メモリ確保に失敗しました。  
**処置** 終了処理を行ってからアプリケーションを終了させてください。

**50650(0xc5da)**

**意味** セカンダリ表のレコードが存在しません。  
**処置** 指定されたキー値のレコードはプライマリ表には存在しますが、対応キーと一致するレコードがセカンダリ表に見つかりませんでした。  
指定キーまたは指定レコード番号を訂正して再実行してください。

**50651(0xc5db)**

**意味** ファイルの終端(EOF)を検出しました。  
**処置** レコード再位置付けを行ってから再実行してください。

**50652(0xc5dc)**

**意味** 第2プライマリ表以降に指定されたキーが存在しません。  
**処置** 合併型仮想表を構成する基本表のインデックス情報を確認し、インデックスを定義してください。

**50653(0xc5dd)**

**意味** 相対番号に0が指定されました。  
**処置**

**50661(0xc5e5)**

**意味** アクセスモードが正しく指定されていません。  
**処置** アクセスモードを正しく指定してください。

**50662(0xc5e6)**

**意味** オープンモードが正しく指定されていません。  
**処置** オープンモードを正しく指定してください。

**50663(0xc5e7)**

**意味** 表作成モードが正しく指定されていません。  
**処置** 表作成モードを正しく指定してください。

**50664(0xc5e8)**

**意味** インデックスの定義が不正です。インデックス数またはキーパート数が指定可能な上限を超えています。  
**処置** インデックスの定義内容を訂正して再実行してください。

**50665(0xc5e9)**

**意味** オープン情報の登録に失敗しました。  
**処置** 終了処理を行ってからアプリケーションを終了させてください。

**50666(0xc5ea)**

**意味** メモリ確保に失敗しました。  
**処置** 終了処理を行ってからアプリケーションを終了させてください。

**50667(0xc5eb)**

**意味** 表定義情報の登録または削除でアクセスエラーです。  
**処置** 内部SQLエンジンでエラーが発生している可能性があります。IFASPRO RDBはSQLエラーが発生した場合にエラーログを出力します。エラーログファイルはIFASPRO RDBのインストールディレクトリ¥rbk配下に“error\*.log”という名前で作成されるテキストファイルで、メモ帳等で参照することが可能です(\*は“01”または“02”となります)。  
 エラーログにはSQLSTATEコードとfNativeErrorコードが出力されていますので、それぞれのコードから詳細情報を得ることができます。エラーログは下記のように出力されますのでSQLSTATEコードとfNativeErrorコードに注目してください。  
 また、ダンプ内に「IFxxxx」というコードがある場合は、その値を「付録A-3 C言語関数のステータス」で参照してください。  
 なお、SQLSTATEコードとfNativeErrorコードの詳細については、「IFASPRO RDB SQLヘルプ」を参照してください。

```
DATE: 14/10/17 TIME: 00:00:00 PID: FFF872B1 TID: FFFAB551 COMMAND: C:\¥ifas¥¥test_rdb.exe
CALL_FUNC: ifInk_rdb_cli_ExeSql CALLED_FUNC (STATUS): ExcexDirect (=1)
[ SQLSTATE=S1907, fNativeError=00000010 ]
0x0088f498 00 00 00 20 00 00 00 40 00 00 00 60 53 31 39 30 ... ..@...`S190
0x0088f4a8 37 00 00 00 10 00 00 00 44 55 50 4b 45 59 20 20 7... .. DUPKEY
0x0088f4b8 3a 53 59 53 54 45 4d 20 45 52 52 4f 52 2e 20 27 :SYSTEM ERROR. '
```

**50668(0xc5ec)**

**意味** 指定された表は既に存在します。  
**処置** 表作成モードまたは表名を訂正して再実行してください。

**50669(0xc5ed)**

**意味** 指定されたDBは存在しません。  
**処置** DB名を正しく指定してください。

**50670(0xc5ee)**

**意味** 列数が指定できる最大値を超えました  
**処置** 列の定義内容を訂正して再実行してください。

**50671(0xc5ef)**

**意味** レコード長が指定できる最大値を超えました。  
**処置** レコード長を訂正して再実行してください。

**50672(0xc5f0)**

- 意味** 同一キーが二重定義されました。
- 処置** 同じ定義内容のインデックスを複数作成することはできません。インデックスの定義内容を訂正して再実行してください。

**50673(0xc5f1)**

- 意味** RDB管理ファイルのアクセスエラーです。(DB破壊)
- 処置** RDB管理ファイルが壊れている可能性があります。RDB表を使用するプログラムをすべて終了し、RDB管理ファイルを復旧してください。

**50674(0xc5f2)**

- 意味** 仮想表のため削除／再作成はできません。
- 処置** 本要求は仮想表に対して行うことはできません。基本表の表名を指定してください。

**50675(0xc5f3)**

- 意味** 仮想表に定義されているため削除／再作成はできません。
- 処置** 仮想表の構成表として定義されている基本表は削除できません。表定義ユーティリティで仮想表を削除してから再実行してください。

**50676(0xc5f4)**

- 意味** 列名が不正です。
- 処置** 列名に使用できない文字が含まれています。「7-2-1 表定義ユーティリティ」で説明されている列名規則を参照し、列名を訂正してください。

**50677(0xc5f5)**

- 意味** 列名が重複して指定されました。
- 処置** 同じ列名が複数定義されています。列名を訂正して再実行してください。

**50678(0xc5f6)**

- 意味** 列サイズが不定です。
- 処置** 列定義で指定された桁数が最大値を超えています。列の定義内容を訂正して再実行してください。

**50679(0xc5f7)**

- 意味** データが存在するため削除できません。
- 処置** 本要求はデータが存在する基本表に対して行うことはできません。表作成モードまたは表名を訂正して再実行してください。

**50691(0xc603)**

- 意味** 指定されたDBは存在しません。
- 処置** DB名を正しく指定してください。

**50692(0xc604)**

- 意味** 指定された表は存在しません。
- 処置** 表名を正しく指定してください。

**50693(0xc605)**

- 意味** 表定義ユーティリティで作成された仮想表ではないため、アクセスできません。
- 処置** 表定義ユーティリティで仮想表を再定義してください。

**50694(0xc606)**

- 意味** 表定義情報の入手でアクセスエラーです。
- 処置** RDB管理ファイルが壊れている可能性があります。RDB表を使用するプログラムをすべて終了し、RDB管理ファイルを復旧してください。

**50701(0xc60d)**

- 意味** 表定義情報エリアのサイズが不正です。  
**処置** 表定義情報エリアが不足しています。サイズを大きくして再実行してください。

**50702(0xc60e)**

- 意味** 表定義情報の入手でアクセスエラーです。  
**処置** RDB管理ファイルが壊れている可能性があります。RDB表を使用するプログラムをすべて終了し、RDB管理ファイルを復旧してください。

**50703(0xc60f)**

- 意味** インデックス情報が不正です。  
**処置** インデックスの定義情報が不正となっている可能性があります。インデックス情報を正しく定義してください。

**50704(0xc610)**

- 意味** 指定されたDBは存在しません。  
**処置** DB名を正しく指定してください。

**50705(0xc611)**

- 意味** 指定された表は存在しません。  
**処置** 表名を正しく指定してください。

**50706(0xc612)**

- 意味** 表定義ユーティリティで作成された仮想表ではないため、アクセスできません。  
**処置** 表定義ユーティリティで仮想表を再定義してください。

**50721(0xc621)**

- 意味** アクセスモードが正しく指定されていません。  
**処置** アクセスモードを正しく指定してください。

**50722(0xc622)**

- 意味** オープンモードとの組み合わせが不正です。  
**処置** 要求に適したオープンモードを指定して再オープンしてください。

**50723(0xc623)**

- 意味** メモリ確保に失敗しました。  
**処置** 終了処理を行ってからアプリケーションを終了させてください。

**50724(0xc624)**

- 意味** カレントレコードが見つかりません。  
**処置** READ/START要求を行った後、本要求を行うようにしてください。

**50725(0xc625)**

- 意味** ファイルの終端(EOF)を検出しました。  
**処置** レコード再位置付けを行ってから再実行してください。

**50741(0xc635)**

- 意味** オープンモードとの組み合わせが不正です。  
**処置** 要求に適したオープンモードを指定して再オープンしてください。

**50742(0xc636)**

- 意味** メモリ確保に失敗しました。  
**処置** 終了処理を行ってからアプリケーションを終了させてください。

**50743(0xc637)**

- 意味 メモリ解放に失敗しました。  
処置 終了処理を行ってからアプリケーションを終了させてください。

**50744(0xc638)**

- 意味 ワークファイルが存在しません。  
処置 表を再オープンしてから再実行してください。

**50745(0xc639)**

- 意味 列情報が存在しません。  
処置 正しい列を指定してください。

**50746(0xc63a)**

- 意味 アクセスモードが不正です。  
処置 正しい値を指定してください。

**50747(0xc63b)**

- 意味 データ変換に失敗しました。  
処置 列のデータタイプが不正となっている可能性があります。列情報を正しく定義してください。

**50749(0xc63d)**

- 意味 データタイプが不正です。  
処置 比較する列のデータタイプが一致しません。検索情報を見直してください。

**50750(0xc63e)**

- 意味 ソート項目に65個以上の項目が指定されました。  
処置 ソート項目の指定は64個までです。ソート項目の指定を見直してください。

**50751(0xc63f)**

- 意味 ワークファイルの作成に失敗しました。  
処置 表を再オープンしてから再実行してください。

**50752(0xc640)**

- 意味 ソート項目に指定した項目の合計サイズが4096バイトを超えました。  
処置 ソート項目の指定を見直してください。

**50753(0xc641)**

- 意味 内部ソート処理でエラーが発生しました。  
処置 終了処理を行ってからアプリケーションを終了させてください。

**50754(0xc642)**

- 意味 メモリ確保に失敗しました。  
処置 終了処理を行ってからアプリケーションを終了させてください。

**50755(0xc643)**

- 意味 ディスクスペースの確保に失敗しました。  
処置 スペース不足のためワークファイルが使用できません。  
IFASPRO RDBインストール先のディスク空き容量を確認してください。

**50761(0xc649)**

- 意味 SELECTされていません。  
処置 処理手順を見直してください。

**50762(0xc64a)**

**意味** ワークファイルが存在しません。  
**処置** 表を再オープンしてから再実行してください。

**50763(0xc64b)**

**意味** オープンモードとの組み合わせが不正です。  
**処置** 要求に適したオープンモードを指定して再オープンしてください。

**50764(0xc64c)**

**意味** ワークファイルの削除に失敗しました。  
**処置** 表を再オープンしてから再実行してください。

**50770(0xc652)**

**意味** DBディレクトリパスが256バイトを超えました。  
**処置** 指定するディレクトリ名が256バイト以内となるように見直してください。

**50771(0xc653)**

**意味** DBディレクトリパスの変換に失敗しました。  
**処置** ディレクトリ名のフルパスが256バイトを以内となるように見直してください。

**50772(0xc654)**

**意味** DB配下に表が存在するため、DBが削除できません。  
**処置** DB配下の表を削除した後、再実行してください。

**50773(0xc655)**

**意味** 指定されたDBは存在しません。  
**処置** DB名を正しく指定してください。

**50774(0xc656)**

**意味** 指定されたDBは既に存在します。  
**処置** DB作成モードまたはDB名を訂正して再実行してください。

**50775(0xc657)**

**意味** RDB管理ファイルのアクセスエラーです。(DB破壊)  
**処置** RDB管理ファイルが壊れている可能性があります。RDB表を使用するプログラムをすべて終了し、RDB管理ファイルを復旧してください。

**50776(0xc658)**

**意味** DBディレクトリの作成に失敗しました。  
**処置** 同一ディレクトリ名が既に存在していないか、ディスクの空き容量があるかなど、ディスクの状況を確認してください。

**50777(0xc659)**

**意味** DBディレクトリの削除に失敗しました。  
**処置** ディレクトリが存在しない可能性があります。ディスクの状況を確認してください。

**50778(0xc65a)**

**意味** DB定義情報の登録／削除でアクセスエラーです。  
**処置** 内部SQLエンジンでエラーが発生している可能性があります。詳細ステータス50667(0xc5eb)の処置欄を参照してください。

**50780(0xc65c)**

**意味** 表定義ユーティリティで作成された仮想表ではないため、アクセスできません。  
**処置** 表定義ユーティリティで仮想表を再定義してください。

**50781(0xc65d)**

**意味** 指定されたDBは存在しません。  
**処置** DB名を正しく指定してください。

**50782(0xc65e)**

**意味** 指定された表は存在しません。  
**処置** 表名を正しく指定してください。

**50783(0xc65f)**

**意味** RDB管理ファイルのアクセスエラーです。(DB破壊)  
**処置** RDB管理ファイルが壊れている可能性があります。RDB表を使用するプログラムをすべて終了し、RDB管理ファイルを復旧してください。

**50784(0xc660)**

**意味** インデックス定義情報の登録／削除でアクセスエラーです。  
**処置** 内部SQLエンジンでエラーが発生している可能性があります。詳細ステータス50667(0xc5eb)の処置欄を参照してください。

**50785(0xc661)**

**意味** 基本表ではないためインデックス操作ができません。  
**処置** 基本表の表名を指定してください。

**50786(0xc662)**

**意味** メモリ確保に失敗しました。  
**処置** 終了処理を行ってからアプリケーションを終了させてください。

**50787(0xc663)**

**意味** インデックス情報が正しく指定されていません。  
**処置** インデックス情報を正しく指定してください。

**50788(0xc664)**

**意味** インデックスが二重定義されています。  
**処置** 指定されたインデックス名または同一の列定義されたインデックスが既に定義されています。インデックス名または列情報を訂正してください。

**50789(0xc665)**

**意味** 指定されたインデックスが存在しません。  
**処置** 指定されたインデックス名または同一の列定義されたインデックスは定義されていません。インデックス名または列情報を訂正してください。

**50795(0xc66b)**

**意味** 表情報を格納するエリアのサイズが不正です。  
**処置** 表情報エリアが不足しています。サイズを大きくして再実行してください。

**50796(0xc66c)**

**意味** 指定されたDBは存在しません。  
**処置** DB名を正しく指定してください。

**50797(0xc66d)**

**意味** RDB管理ファイルのアクセスエラーです。(DB破壊)  
**処置** RDB管理ファイルが壊れている可能性があります。RDB表を使用するプログラムをすべて終了し、RDB管理ファイルを復旧してください。

**50850(0xc6a2)**

意味 表定義情報入手でアクセスエラーです。  
処置

**50851(0xc6a3)**

意味 RDB管理ファイルのアクセスエラーです。(DB破壊です)  
処置

**50852(0xc6a4)**

意味 指定されたDBは存在しません。  
処置

**50853(0xc6a5)**

意味 指定された表が存在しません。  
処置

**50860(0xc6ac)**

意味 表作成モードが正しく指定されていません。  
処置

**50861(0xc6ad)**

意味 表定義情報入手でアクセスエラーです。  
処置

**50862(0xc6ae)**

意味 RDB管理ファイルのアクセスエラーです。(DB破壊です)  
処置

**50863(0xc6af)**

意味 基本表のため、削除できません。  
処置

**50864(0xc6b0)**

意味 仮想表に定義されているため削除できません。  
処置

**50865(0xc6b1)**

意味 システム表のアクセスに失敗しました。  
処置

**50866(0xc6b2)**

意味 仮想表のタイプが正しく指定されていません。  
処置

**50867(0xc6b3)**

意味 仮想表の構成表を示すDBが存在しません。  
処置

**50868(0xc6b4)**

意味 仮想表の構成表が存在しません。  
処置

**50869(0xc6b5)**

意味 メモリ確保に失敗しました。  
処置



**50870(0xc6b6)**

意味 相関名指定でなければならない表に対して相関名が指定されていません  
処置

**50871(0xc6b7)**

意味 構成するカラムが存在しません。  
処置

**50872(0xc6b8)**

意味 指定されたDBが存在しません。  
処置

**50873(0xc6b9)**

意味 カラム数が指定できる最大値を超えました。  
処置

**50874(0xc6ba)**

意味 レコード長が指定できる最大値を超えました。  
処置

**50875(0xc6bb)**

意味 カラム名が不正です。  
処置

**50876(0xc6bc)**

意味 カラム名が重複して指定されました。  
処置

**50877(0xc6bd)**

意味 カラムサイズが不定です。  
処置

**50878(0xc6be)**

意味 構成表が多過ぎます(63個以下)  
処置

**50879(0xc6bf)**

意味 構成表は複数必要です  
処置

**50880(0xc6c0)**

意味 合併型仮想表の構成表は基本表でなければなりません  
処置

**50881(0xc6c1)**

意味 合併型仮想表のカラム数と構成表のカラム数が一致していません  
処置

**50882(0xc6c2)**

意味 構成表に同じ表が指定されています  
処置

## 50883(0xc6c3)

意味 すべての構成表のカラム情報は一致していなければなりません  
処置

## 50884(0xc6c4)

意味 仮想表構成情報構造体が指定されていません  
処置

## 50885(0xc6c5)

意味 仮想表項目情報構造体が指定されていません  
処置

## 50886(0xc6c6)

意味 結合型の構成表に合併型仮想表は指定できません  
処置

## 50887(0xc6c7)

意味 構成表名の指定正しくありません  
処置

## 50888(0xc6c8)

意味 相関名の指定正しくありません  
処置

## 50889(0xc6c9)

意味 アクセスキーパート数が不正です  
処置

## 50890(0xc6ca)

意味 結合タイプが正しく指定されていません  
処置

## 50891(0xc6cb)

意味 関係式数不正が不正です  
処置

## 50892(0xc6cc)

意味 関係式のカラムはタイプが一致していなければなりません  
処置

## 50893(0xc6cd)

意味 関係式の構成が正しくありません  
処置

## 50894(0xc6ce)

意味 関係式の左辺はインデックスでなければなりません  
処置

## 50895(0xc6cf)

意味 指定された表は既に存在します  
処置

## 50896(0xc6d0)

意味 構成表に存在しない表です  
処置

**50897(0xc6d1)**

**意味** メモリ解放失敗しました  
**処置**

**54002(0xd2f2)**

**意味** IFASPRO RDBがインストールされていません。  
**処置** RDB表を使用する場合はIFASPRO RDBをご購入ください。

**54003(0xd2f3)**

**意味** 表名変換でエラーです。  
**処置** 表名が指定されていないか、使用できない文字を指定しています。表名を正しく指定して再実行してください。

**54004(0xd2f4)**

**意味** ファイル名変換でエラーです。  
**処置** ファイル名が指定されていないか、使用できない文字を指定しています。ファイル名を正しく指定して再実行してください。

**54005(0xd2f5)**

**意味** 更新ログファイルのオープンでエラーです。  
**処置** エラー理由を取り除き、再実行してください。

**54006(0xd2f6)**

**意味** RDB管理ファイルの一括オープンでエラーです。  
**処置** RDB管理ファイルが作成されていない可能性があります。

**54007(0xd2f7)**

**意味** IFASPRO RDBがインストールされていません。  
**処置** RDB表を使用する場合はIFASPRO RDBをご購入ください。

**54008(0xd2f8)**

**意味** 表名変換でエラーです。  
**処置** 表名が指定されていないか、使用できない文字を指定しています。表名を正しく指定して再実行してください。

**54009(0xd2f9)**

**意味** ファイル名変換でエラーです。  
**処置** ファイル名が不正です。正しい値を指定してください。

**54010(0xd2fa)**

**意味** 表定義オプション入手でエラーです。  
**処置** DB名または表名が正しく指定されていない可能性があります。指定内容を確認してください。

**54011(0xd2fb)**

**意味** 更新ログファイルのオープンでエラーです。  
**処置** エラー理由を取り除き、再実行してください。

**54012(0xd2fc)**

**意味** RDB管理ファイルの一括オープンでエラーです。  
**処置** RDB管理ファイルが作成されていない可能性があります。

**54013(0xd2fd)**

意味 commitでエラーです。  
処置 エラー理由を取り除き、再実行してください。

**54014(0xd2fe)**

意味 rollbackでエラーです。  
処置 エラー理由を取り除き、再実行してください。

**54015(0xd2ff)**

意味 トランザクション開始処理でエラーです。  
処置 エラー理由を取り除き、再実行してください。

**54016(0xd300)**

意味 トランザクション開始処理でエラーです。  
処置 エラー理由を取り除き、再実行してください。

**54017(0xd301)**

意味 トランザクション開始処理でエラーです。  
処置 エラー理由を取り除き、再実行してください。

**54018(0xd302)**

意味 commitでエラーです。  
処置 エラー理由を取り除き、再実行してください。

**54019(0xd303)**

意味 rollbackでエラーです。  
処置 エラー理由を取り除き、再実行してください。

**54020(0xd304)**

意味 トランザクション開始処理でエラーです。  
処置 エラー理由を取り除き、再実行してください。

**54021(0xd305)**

意味 トランザクション開始処理でエラーです。  
処置 エラー理由を取り除き、再実行してください。

**54050(0xd322)**

意味 IFASPRO RDBがインストールされていません。  
処置 RDB表を使用する場合はIFASPRO RDBをご購入ください。

**54051(0xd323)**

意味 表名変換でエラーです。  
処置 表名が指定されていないか、使用できない文字を指定しています。表名を正しく指定して再実行してください。

**54052(0xd324)**

意味 ファイル名変換でエラーです。  
処置 ファイル名が不正です。正しい値を指定してください。

**54053(0xd325)**

意味 更新ログファイルのオープンでエラーです。  
処置 エラー理由を取り除き、再実行してください。

**54054(0xd326)**

意味 RDB管理ファイルの一括オープンでエラーです。  
処置 RDB管理ファイルが作成されていない可能性があります。

**54055(0xd327)**

**意味** キー情報が指定されていません。  
**処置** 正しい値を指定してください。

**54056(0xd328)**

**意味** インデックス名の変換に失敗しました。  
**処置** インデックス名が指定されていないか、使用できない文字を指定しています。インデックス名を正しく指定して再実行してください。

**54057(0xd329)**

**意味** 表定義オプション入手に失敗しました。  
**処置**

**54058(0xd32a)**

**意味** DBシステムがビジーです。  
IFASシステムがバックアップ中です  
**処置** しばらく待ってから再実行してください。  
バックアップの終了を待って再実行してください。

**54060(0xd32c)**

**意味** IFASPRO RDBがインストールされていません。  
**処置** RDB表を使用する場合はIFASPRO RDBをご購入ください。

**54061(0xd32d)**

**意味** DB名変換でエラーです。  
**処置** DB名が指定されていないか、使用できない文字を指定しています。DB名を正しく指定して再実行してください。

**54062(0xd32e)**

**意味** DBディレクトリパス名変換でエラーです。  
**処置** DBディレクトリパス名が指定されていないか、使用できない文字を指定しています。  
DBディレクトリパス名を正しく指定して再実行してください。

**54063(0xd32f)**

**意味** RDB管理ファイルの一括オープンでエラーです。  
**処置** RDB管理ファイルが作成されていない可能性があります。

**54064(0xd330)**

**意味** キー情報が指定されていません。  
**処置** 正しい値を指定してください。

**54065(0xd331)**

**意味** DBシステムがビジーです。  
IFASシステムがバックアップ中です  
**処置** しばらく待ってから再実行してください。  
バックアップの終了を待って再実行してください。

**54070(0xd336)**

**意味** IFASPRO RDBがインストールされていません。  
**処置** RDB表を使用する場合はIFASPRO RDBをご購入ください。

**54071(Oxd337)**

**意味** 表名変換でエラーです。

**処置** 表名が指定されていないか、使用できない文字を指定しています。表名を正しく指定して再実行してください。

**54072(Oxd338)**

**意味** RDB管理ファイルの一括オープンでエラーです。

**処置** RDB管理ファイルが作成されていない可能性があります。

**54073(Oxd339)**

**意味** ファイル名変換でエラーです。

**処置** ファイル名が不正です。正しい値を指定してください。

**54074(Oxd33a)**

**意味** DBシステムがビジーです。

IFASシステムがバックアップ中です

**処置** しばらく待ってから再実行してください。

バックアップの終了を待って再実行してください。

**54080(Oxd340)**

**意味** IFASPRO RDBがインストールされていません。

**処置** RDB表を使用する場合はIFASPRO RDBをご購入ください。

**54081(Oxd341)**

**意味** DB名変換でエラーです。

**処置** DB名が指定されていないか、使用できない文字を指定しています。DB名を正しく指定して再実行してください。

**54082(Oxd342)**

**意味** モードの指定が不正です。

**処置** 正しい値を指定してください。

**54083(Oxd343)**

**意味** RDB管理ファイルの一括オープンでエラーです。

**処置** RDB管理ファイルが作成されていない可能性があります。

**54084(Oxd344)**

**意味** DBシステムがビジーです。

IFASシステムがバックアップ中です

**処置** しばらく待ってから再実行してください。

バックアップの終了を待って再実行してください。

**54085(Oxd345)**

**意味** IFASPRO RDBがインストールされていません

**処置**

**54086(Oxd346)**

**意味** データベース名の変換でエラーです

**処置**

**54087(Oxd347)**

**意味** システムファイル一括オープンでエラーです

**処置**

**54088(0xd348)**

**意味** DBシステムがビジーです  
IFASシステムがバックアップ中です

**処置**

**54090(0xd34a)**

**意味** RDB管理ファイルの一括オープンでエラーです。

**処置** RDB管理ファイルが作成されていない可能性があります。

**54091(0xd34b)**

**意味** DBシステムがビジーです。  
IFASシステムがバックアップ中です

**処置** しばらく待ってから再実行してください。  
バックアップの終了を待って再実行してください。

**54095(0xd34f)**

**意味** DBシステムがビジーです。  
IFASシステムがバックアップ中です

**処置** しばらく待ってから再実行してください。  
バックアップの終了を待って再実行してください。

**54096(0xd350)**

**意味** DBシステムがビジーです。  
IFASシステムがバックアップ中です

**処置** しばらく待ってから再実行してください。  
バックアップの終了を待って再実行してください。

## A-5 IFASカスタムコントロールのステータス (参考)

IFAS FILE Ver2.0、IFASPRO RDB Ver2.0より、「IFASカスタムコントロール」機能の提供を中止しました。本付録の内容は従来バージョンをご利用のお客様が旧製品の仕様を確認いただくために残しています。

IFASカスタムコントロールのステータスについて説明します。

カスタムコントロールのステータスは、**IfasActionStatus**プロパティにString型3桁で設定されます。

ステータス一覧とその意味を以下に記述します。

種別	シンボル	コード	意味
000	ifasActionNormal	000	正常
IFAS FILEエラー	なし	0xx	IFASアクセス処理でのエラー <sup>注</sup>
OCXエラー	ifasActionNotOpenFile	101	表がオープンされていない
(論理エラー)	ifasActionNoTReadRec	102	レコードが読み込まれていない
	ifasActionSelectKeyName	103	キー指定が不正
	ifasActionRecNumber	104	レコード番号不正
	ifasActionColumnName	105	項目名が不正
	ifasActionColumnNum	106	項目番号が不正
	ifasActionIllegalData	107	データが不正
	ifasActionAccesKey	108	アクセスキーが不正
	ifasActionIllegalSortType	109	SORT順指定が不正
	ifasActionIllegalSortMode	110	SORTモードが不正
	ifasActionSelectCondition	111	検索条件式が不正
	ifasActionOpenMode	112	オープンモードが不正
	ifasActionLockMode	113	ロックモードが不正
	ifasActionDeleteMode	114	削除モードが不正
	ifasActionRewriteMode	115	更新モードが不正
	ifasActionStartMode	116	位置付けモードが不正
	ifasActionGetMode	117	レコード読み出しモードが不正
	ifasActionProperty	118	プロパティの値が不正
OCXエラー	ifasActionAllocateMemory	201	メモリが確保できなかった
(致命的なエラー)	ifasActionDeallocateMemory	202	メモリの解放に失敗した
	ifasActionOverflow	203	オーバフロー
	ifasActionDllerror	204	DLLのロードに失敗
未サポート機能	ifasActionTableType	301	表の型が不明
要求時	ifasActionColumnAttribute	302	項目値の属性が不明
IFAS ファイルオプションの設定エラー	ifasActionFileOption	401	ファイルオプションの設定エラー

**注** IFAS FILEエラーの場合、COBOL利用時の入出力状態が、“0xx”のxxに設定されます。COBOLの入出力状態に関しては「付録A-4-1 入出力状態」を参照してください。  
 なお、“0FF”/“0F0”が設定されている場合もありますが、おのおの0xffff/0xff0を指します。  
 また、IfasMinorStatus1/IfasMinorStatus2の各プロパティには、それぞれ、「詳細ステータス」/「内部ステータス」が設定されます。合わせて確認してください。



## 付録B IFASコマンド一覧

### IFASBKSP. EXE(バックアップ支援ユーティリティ)

#### 【機能】

運用中のIFAS ファイルのバックアップを採取するためにIFAS FILE関数の動作を抑制します。

#### 【書式】

IFASBKSP バッチファイル名 [/待ち時間] [/I]

#### 【パラメータ】

バッチファイル名:

バックアップ処理を記述したバッチファイルです。バックアップ後に更新ログのリセット処理を行う場合には、最後に更新ログリセットコマンドの記述を加えてください。ファイル名にブランクが含まれる場合はパラメータをダブルクォーテーション(")で囲んでください。

詳細は「7-4-2 バックアップ支援ユーティリティ」を参照してください。

/待ち時間:

本コマンドで抑制するIFAS FILE関数の待ち時間(秒)です。パラメータを省略すると300になります。

/I:

パラメータ(大文字・小文字の区別なし)が指定された場合、バックアップバッチが異常終了した時でもファイルオープン抑制は解除されます。

#### 【戻り値】(ERRORLEVEL)

0	正常終了しました
1～99	強制終了されたアプリケーションが存在する可能性があります(ワーニングですが処理は実行されます)
255	ユーティリティによるエラーで終了しました

### IFASCOPY. EXE、IFASCOPY64. EXE(ファイルコピーユーティリティ)

#### 【機能】

IFAS一般ファイル、MS-DOSテキストファイル、CSV形式ファイル間のコピーを行います。IFASCOPY.EXE(32ビット版)とIFASCOPY64.EXE(64ビット版)では利用可能な機能に差があります。詳細は「7-1-2 ファイルコピーユーティリティ」を参照してください。

#### 【書式】

IFASCOPY 定義ファイル名 [置換文字列1] … [置換文字列9]

IFASCOPY64 定義ファイル名 [置換文字列1] … [置換文字列9]

## 【パラメータ】

定義ファイル名:

処理の設定を記述したテキストファイルです。

ファイル名にブランクが含まれる場合はパラメータをダブルクォーテーション(")で囲んでください。詳細は「7-1-2 ファイルコピーユーティリティ」を参照してください。

置換文字列1～9:

定義ファイル内の特定の文字を指定文字列と置換します。詳細は「7-5 定義ファイルのパラメータ置換機能」を参照してください。

## 【戻り値】(ERRORLEVEL)

0	正常終了しました
1	IFAS FILEのワーニングステータス41290が返却されました
2～99	IFAS FILEの正常系ステータスで終了しました
100	IFAS FILEの異常系ステータスで終了しました
255	ユーティリティによるエラーで終了しました

## IFASDCLR. EXE(データクリアコマンド)

## 【機能】

IFAS一般ファイルのデータ数を0件にします。

## 【書式】

IFASDCLR IFAS ファイル名 [インデックスファイル名]

## 【パラメータ】

IFAS ファイル名:

IFAS ファイルのファイル名を記述します。索引順編成の場合、データファイル名を記述します。

ファイル名にブランクが含まれる場合はパラメータをダブルクォーテーション(")で囲んでください。

インデックスファイル名:

対象がIFAS索引順編成の場合、インデックスファイル名を記述します。IFAS索引順編成ファイルで本パラメータを省略した場合、自動的にデータファイル名の拡張子をIDXに変えたものをインデックスファイル名として処理を実行します。

ファイル名にブランクが含まれる場合はパラメータをダブルクォーテーション(")で囲んでください。

- ※ IFASのC/S機能を利用してサーバ上のファイルのデータクリアを行う場合は、サーバ上のファイルパスの前にサーバのコンピュータ名を"<コンピュータ名>"で括って記述してください。その際、コマンドのリダイレクト機能が働かないようにファイル名全体を"(ダブルクォーテーション)で括ってください。

例 IFASDCLR "<コンピュータ名>ファイルパス"

## 【戻り値】(ERRORLEVEL)

0	正常に終了しました
1～99	IFAS FILEの正常系ステータスで終了しました

100 IFAS FILEの異常系ステータスで終了しました  
 255 ユーティリティによるエラーで終了しました

#### 【詳細ステータス】

本コマンドが異常終了した場合、カレントディレクトリにエラーファイル(拡張子が.ERR)を作成し、詳細ステータスを返却します。

45001 パラメータ不足  
 45002 ファイル編成が索引順、相対、順ではない  
 45003 先がデータファイルではない  
 45004 後がインデックスファイルではない

上記以外の場合は、IFAS FILE関数ヘルプを参照してください。

## IFASDRSM. EXE(IFAS管理情報メンテナンスツール)

#### 【機能】

IFASのDRSファイル内の管理情報の保守を行います。ファイルビジーのリセットや登録されているファイルの管理情報の削除や一覧出力等を行えます。

#### 【書式】

IFASDRSM -O:オプション -F:ファイル名 [-M] [-D] [-NOCHK]

#### 【パラメータ】

オプション:

メンテナンス機能を指定します。

C: ビジーリセット  
 R: ビジーリセットと未クローズロックリセット  
 E: ファイル管理情報の削除  
 L: ファイル管理情報の一覧出力

ファイル名:

オプションが”L”の場合、ファイル管理情報一覧の出力先ファイル名(CSV形式)です。  
 オプションが”L”以外の場合、メンテナンス対象のIFAS ファイル名です。  
 ただし、下記拡張オプション”-M”が指定された場合は、メンテナンス対象のIFAS ファイル名の一覧を格納したCSVファイルまたはテキストファイルのファイル名です。  
 ファイル名に空白が含まれる場合はパラメータをダブルクォーテーション(”)で囲んでください。

拡張オプション:

-M: 複数のIFAS ファイルを一括してメンテナンスします。  
 -D: 詳細エラー情報を表示します。  
 -NOCHK: 対象となるIFAS ファイルがトランザクション指定でオープンされていたり他のプロセスで使用中でも、管理情報の変更を実施します。

詳細は「7-4-4 IFAS管理情報メンテナンスツール」を参照してください。

#### 【戻り値】(ERRORLEVEL)

0 正常に終了しました

0以外 エラーが発生しました

## IFASERAS. EXE(ファイル削除コマンド)

### 【機能】

IFAS一般ファイルを削除します。

### 【書式】

IFASERAS IFAS ファイル名 [インデックスファイル名]

### 【パラメータ】

IFAS ファイル名：

IFAS ファイルのファイル名を記述します。索引順編成の場合、データファイル名を記述します。

ファイル名にブランクが含まれる場合はパラメータをダブルクォーテーション(")で囲んでください。

インデックスファイル名：

対象がIFAS索引順編成の場合、インデックスファイル名を記述します。IFAS索引順編成ファイルで本パラメータを省略した場合、自動的にデータファイル名の拡張子をIDXに変えたものをインデックスファイル名として処理を実行します。

ファイル名にブランクが含まれる場合はパラメータをダブルクォーテーション(")で囲んでください。

- ※ IFASのC/S機能を利用してサーバ上のファイルを削除する場合は、サーバ上のファイルパスの前にサーバのコンピュータ名を"<">"で括って記述してください。その際、コマンドのリダイレクト機能が働かないようにファイル名全体を"(ダブルクォーテーション)で括ってください。

例 IFASERAS "<コンピュータ名>ファイルパス"

### 【戻り値】(ERRORLEVEL)

0	正常に終了しました
1～99	IFAS FILEの正常系ステータスで終了しました
100	IFAS FILEの異常系ステータスで終了しました
255	ユーティリティによるエラーで終了しました

### 【詳細ステータス】

本コマンドが異常終了した場合、カレントディレクトリにエラーファイル(拡張子が.ERR)を作成し、

詳細ステータスを返却します。

45001	パラメータ不足
45002	ファイル編成が索引順、相対、順ではない
45003	先がデータファイルではない
45004	後がインデックスファイルではない

上記以外の場合は、IFAS FILE関数ヘルプを参照してください。

## IFASFATR. EXE(レコード追加方式変更コマンド)

### 【機能】

IFAS索引順編成ファイルのレコード追加方式を変更します。

### 【書式】

IFASFATR /スイッチ データファイル名 [インデックスファイル名]

### 【パラメータ】

スイッチ:

以下にスイッチの種類と指定方法を記述します。2つのスイッチのうち、いずれか1つを必ず指定してください。

/A ファイル内の削除領域を使用せず、レコードの追加はファイルの最後尾に行われます。

/V ファイル内の削除領域を使用するため、レコードの追加は削除領域から順に行われます。

データファイル名:

IFAS索引順編成ファイルのデータファイル名を指定してください。

ファイル名にブランクが含まれる場合は、ファイル名をダブルクォーテーション(")で囲んでください。

インデックスファイル名:

IFAS索引順編成ファイルのインデックスファイル名を指定してください。本パラメータが省略された場合、自動的にデータファイル名の拡張子を.IDXに変えたものをインデックスファイル名として処理を実行します。

ファイル名にブランクが含まれる場合は、ファイル名をダブルクォーテーション(")で囲んでください。

※ IFASのC/S機能を利用してサーバ上のファイルのレコード追加方式の変更を行う場合は、サーバ上のファイルパスの前にサーバのコンピュータ名を"<>"で括って記述してください。

その際、コマンドのリダイレクト機能が働かないようにファイル名全体を"(ダブルクォーテーション)で括ってください。

例 IFASFATR /V "<コンピュータ名>データファイルパス"

### 【戻り値】(ERRORLEVEL)

0	正常に終了しました
1～99	IFAS FILEの正常系ステータスで終了しました
100	IFAS FILEの異常系ステータスで終了しました
255	ユーティリティによるエラーで終了しました

### 【詳細ステータス】

本コマンドが異常終了した場合、カレントディレクトリにエラーファイル(拡張子が.ERR)を作成し、詳細ステータスを返却します。

45001	パラメータ不足
45002	ファイル編成が索引順、相対、順ではない
45003	先がデータファイルではない
45004	後がインデックスファイルではない

上記以外の場合は、IFAS FILE関数ヘルプを参照してください。

## IFASFCNV. EXE(移行ユーティリティ)

### 【機能】

他システムで作成された順編成ファイルからIFAS ファイルを作成します。

### 【書式】

IFASFCNV データ定義ファイル名

### 【パラメータ】

データ定義ファイル名：

処理の設定を記述したテキストファイルです。

ファイル名にブランクが含まれる場合はパラメータをダブルクォーテーション(")で囲んでください。詳細は「7-1-5 移行ユーティリティ」を参照してください。

### 【戻り値】(ERRORLEVEL)

0	正常に終了しました(ワーニングも含む)
1	ユーティリティによるエラーで終了しました

## IFASFRST. EXE(ロックリセットユーティリティ)

### 【機能】

未クローズロックのかかったIFAS一般ファイルのロックを外します。

### 【書式】

IFASFRST IFAS ファイル名 [インデックスファイル名]

### 【パラメータ】

IFAS ファイル名：

IFAS ファイルのファイル名を記述します。索引順編成の場合、データファイル名を記述します。

ファイル名にブランクが含まれる場合はパラメータをダブルクォーテーション(")で囲んでください。

インデックスファイル名：

対象がIFAS索引順編成の場合、インデックスファイル名を記述します。

ファイル名にブランクが含まれる場合はパラメータをダブルクォーテーション(")で囲んでください。

### 【戻り値】(ERRORLEVEL)

0	正常に終了しました
1～99	IFAS FILEの正常系ステータスで終了しました
100	IFAS FILEの異常系ステータスで終了しました
255	ユーティリティによるエラーで終了しました

## IFASINFO. EXE(IFASプロパティ表示コマンド)

### 【機能】

IFAS FILE、IFASPRO RDBのインストール情報等を表示します。

### 【書式】

IFASINFO レベル

### 【パラメータ】

レベル:

表示する情報番号を記述します。

- 1 : 基本情報
- 2 : ロックテーブル関連情報
- 3 : 資源情報
- 4 : COBOL(A-VX II)関連情報
- 5 : COBOL(A-VX5)関連情報
- 99 : 1～5のすべての情報

詳細は「7-6-1 IFASプロパティ表示」を参照してください。

### 【戻り値】(ERRORLEVEL)

なし

## IFASIXMN. EXE(インデックスメンテナンスユーティリティ)

### 【機能】

IFAS索引順編成ファイルに対し、インデックスの追加、削除、属性変更、インデックス再作成を行います。

### 【書式】

IFASIXMN 定義ファイル名 [置換文字列1] … [置換文字列9]

### 【パラメータ】

定義ファイル名:

処理の設定を記述したテキストファイルです。詳細は「7-1-3 インデックスメンテナンスユーティリティ」を参照してください。

ファイル名にブランクが含まれる場合はパラメータをダブルクォーテーション(")で囲んでください。

置換文字列1～9:

定義ファイル内の特定の文字を指定文字列と置換します。詳細は「7-5 定義ファイルのパラメータ置換機能」を参照してください。

### 【戻り値】(ERRORLEVEL)

- |      |                           |
|------|---------------------------|
| 0    | 正常終了しました                  |
| 1～99 | IFAS FILEの正常系ステータスで終了しました |
| 100  | IFAS FILEの異常系ステータスで終了しました |
| 255  | ユーティリティによるエラーで終了しました      |

## IFASLFRS. EXE(更新ログリセットコマンド)

### 【機能】

バックアップ採取後などに、不要になった更新ログファイルをリセットします。

### 【書式】

IFASLFRS /スイッチ [ファイル名]

### 【パラメータ】

スイッチ:

以下にスイッチの種類と指定方法を記述します。3つのスイッチのうち、いずれか1つを必ず指定してください。

/CBL

COBOL用ログファイルをリセットします。運用支援ツールの[更新ログ設定]で指定している更新ログファイルをリセットします。

/TEXT ファイル名

指定ファイルに記述している更新ログファイルをリセットします。指定ファイルには、更新ログファイル名のフルパスを、テキスト形式で記述します。更新ログファイル名は1行に1ファイル記述し、複数ファイル指定する場合は、複数行に記述してください。

/C 更新ログファイル名

主に、Cインターフェースで使用する更新ログファイル名を記述してください。

指定している更新ログファイルをリセットします。更新ログファイルは、主に、Cインターフェースで使用する更新ログファイル名を記述してください。

ファイル名:

第1パラメータが /TEXT または /C の場合に指定します。

ファイル名に空白が含まれる場合はパラメータをダブルクォーテーション(")で囲んでください。

### 【戻り値】(ERRORLEVEL)

0	正常終了しました
10	複数ファイルでエラー発生
11	シングルログファイルが存在しません 更新ログファイル名が正しいか確認してください。
12	シングルログファイルBUSY 更新ログファイルが使用されています。 更新ログファイルをリセットすることはできません。
13	シングルフォーマットエラー 更新ログファイル名が正しいか確認してください。
14	シングルログファイルアクセスエラー エラーログを参照してください。
15	シングルログファイル削除エラー エラーログを参照してください。
16	TEXTファイルのフォーマットエラー TEXTファイルの内容を確認してください。



- |    |                                   |
|----|-----------------------------------|
| 30 | スイッチ情報不正<br>正しいスイッチを指定してください。     |
| 31 | レジストリ情報入手に失敗<br>エラーログを参照してください。   |
| 32 | レジストリ情報更新に失敗<br>エラーログを参照してください。   |
| 33 | TEXTファイル操作でエラー<br>エラーログを参照してください。 |

## IFASLTBI. EXE(ロックテーブル使用状況表示コマンド)

### 【機能】

IFAS FILE、IFASPRO RDBが独自の排他制御等で使用するロックテーブルの使用状況を表示します。  
ロック制御に関するトラブル対応時の開発者向けコマンドです。内容をコピーしてお問い合わせください。

### 【書式】

IFASLTBI

### 【パラメータ】

なし

### 【戻り値】(ERRORLEVEL)

なし

## IFCPYCOM. EXE(RDB表用コピーコマンド)

### 【機能】

RDB表に対するデータのインポート／エクスポートを行います。

### 【書式】

IFCPYCOM 定義ファイル名 [置換文字列1] … [置換文字列9]

### 【パラメータ】

定義ファイル名:

処理の設定を記述したテキストファイルです。詳細は「**7-2-1 表定義ユーティリティ**」を参照してください。

ファイル名にブランクが含まれる場合はパラメータをダブルクォーテーション(")で囲ってください。

置換文字列1～9:

定義ファイル内の特定の文字を指定文字列と置換します。詳細は「**7-5 定義ファイルのパラメータ置換機能**」を参照してください。

### 【戻り値】(ERRORLEVEL)

- |   |                                     |
|---|-------------------------------------|
| 0 | 正常終了しました                            |
| 1 | IFASPRO RDBのワーニングステータス41290が返却されました |

2～99	IFASPRO RDBの正常系ステータスで終了しました
100	IFASPRO RDBの異常系ステータスで終了しました
255	ユーティリティによるエラーで終了しました

## IFRDBUTL. EXE(表定義ユーティリティ)

### 【機能】

RDB表に対し、インデックスの更新属性変更／再作成、データの全件削除、RDB管理ファイルのロック解除を行います。

### 【書式】

IFRDBUTL /スイッチ

### 【パラメータ】

スイッチ：

/Recovery RDB管理ファイルのロック解除を行います。  
 /Attr={SYNC|ASYNC|EXEC}  
     SYNC : インデックスの更新を反映する  
     ASYNC : インデックスの更新を反映しない  
     EXEC : インデックスを実行時生成する  
 /REMake={AUTO|FORCE|GARBAGE}  
     AUTO : インデックスのチェックを行い再作成します。  
     FORGE : 無条件にインデックスの再作成を行います。  
     GARBAGE : データファイルのガベージ処理とインデックスの再作成を行います。  
 /Cleardata データファイルからデータをすべて削除します。  
 /DB=DB名 データベース名を指定します。省略時は既定値が使用されます。  
 /Table=表明 表名を指定します。  
 /Index=インデックス名 インデックス名を指定します。  
 指定例などの詳細は「7-2-1 表定義ユーティリティ」を参照してください。

### 【戻り値】(ERRORLEVEL)

0	正常終了しました
100	パラメータ指定が不正です
101	内部エラー(メモリ不足など)が発生しました
102	IFAS FILEエラーが発生しました
103	SQLエラーが発生しました

## 付録C RFDファイルについて

RFDファイルとは、データベースファイルやIFAS ファイルのファイル情報やレコード情報を定義／記憶したレコードフォーマットファイルです。ファイルが存在するディレクトリおよびデータベースディレクトリ配下に**ファイル名(または表名)+拡張子".rfd"**の名称で作成されます。

IFAS FILE、IFAPRO RDBの機能でRFDファイルを使用しているものを以下に示します。

コンポーネント	機能	備考
コピーユーティリティ (IFASCOPY.EXE)	CSV形式<->IFAS一般ファイルのデータコピー	必須
RDB表用コピーコマンド (IFCPYCOM.EXE)	IFAS一般ファイルからRDB表へのインポート	必須
	RDB表からIFAS一般ファイルへのエクスポート	必須 (注1)
	CSV形式<->IFAS一般ファイルのデータコピー	必須
表定義ユーティリティ (IFRDBUTL.EXE)	IFAS一般ファイルからのインポート	必須
	RFDファイルを利用した表定義	必須
IFASカスタムコントロール (IFAS32.OCX) (注2)	IFAS一般ファイルをOCXで利用	必須

IFAS ファイルに対するRFDファイルの作成には、以下の機能を使用します。

- IFAS一般ファイルのRFDファイル — RFD編集ツール(IFRFDUTL.EXE)の新規作成／編集
- IFAS RDB表のRFDファイル — 表定義ユーティリティ(IFRDBUTL.EXE)のRFD出力

(注1) IFAS一般ファイルのファイル編成が相対編成、順編成で、ローカルアクセス形態での利用の場合、RFDファイルがなければ自動的にRFDファイルを作成して処理します。

(注2) IFAS FILE Ver2.0、IFASPRO RDB Ver2.0より、「IFASカスタムコントロール」機能の提供を中止しました。本内容は従来バージョンをご利用のお客様が旧製品の仕様を確認いただくために残しています。

## 付録D バージョン間機能差異

IFAS FILE、IFASPRO RDBのVer1.0とVer2.0で対応している機能の差異の一覧です。

機能	Ver1.0	Ver2.0
サポートOS (※)	Windows Server 2008、Windows Vista以前はサポート対象外	Windows Server 2008 R2、Windows 8以前(Windows 7は除く)はサポート対象外
カスタムコントロール	利用可能	利用不可 Visual Basicから利用する場合、別途製品の「RDB/FILEアクセスコントロール」を利用することで代替
リスタートリカバリおよび更新ログの収集	本機能利用時、IFAS FILEイニシエータのサービスのアカウントに“Administrator”ユーザーの設定が必須	本機能利用時、IFAS FILEイニシエータのサービスのアカウントは、“Administrators”グループに所属する任意のユーザーを設定可能

(※) 最新のサポート状況はホームページでご確認ください。

# 索引

## A

ADDINDEX\_IS関数 ..... 59

## B

BufferNumberプロパティ ..... 186  
 BUILD\_IDXFILE\_RL関数 ..... 93  
 BUILD\_IDXFILE\_SQ関数 ..... 109  
 BusyReturnModeプロパティ ..... 190

## C

CHAIN ..... 12  
 CLOSE\_IS関数 ..... 46  
 CLOSE\_RL関数 ..... 77  
 CLOSE\_SQ関数 ..... 100  
 Closeメソッド ..... 194  
 COBOL設定 ..... 130, 158  
 COBOLファイルマッパー ..... 23, 24, 127  
 COMMIT関数 ..... 120  
 Commitメソッド ..... 209  
 CREATE\_IS関数 ..... 40  
 CREATE\_RL関数 ..... 72  
 CREATE\_SQ関数 ..... 96  
 CSV ..... 268, 272, 280

## D

DELETE\_IS関数 ..... 54  
 DELETE\_RL関数 ..... 84  
 DeleteRecordStringメソッド ..... 227  
 DeleteRecordメソッド ..... 226  
 Deleteメソッド ..... 200  
 DELINDEX\_IS関数 ..... 61  
 DupKeyReadModeプロパティ ..... 183  
 DupKeyWriteModeプロパティ ..... 182

## E

ERASE\_IS関数 ..... 65  
 ERASE\_RL関数 ..... 90  
 ERASE\_SQ関数 ..... 106  
 ErrMethodプロパティ ..... 181  
 ErrTableNameプロパティ ..... 180

## F

FLUSH関数 ..... 112  
 Flushメソッド ..... 211

## G

GET\_IS関数 ..... 49  
 GET\_RL関数 ..... 80  
 GET\_SQ関数 ..... 102  
 GetColumnDataStringメソッド ..... 214  
 GetColumnDataメソッド ..... 212  
 GetColumnDefメソッド ..... 236  
 GetIndexColumnDefメソッド ..... 232  
 GetIndexDefメソッド ..... 231  
 GetIndexInfoDefメソッド ..... 234  
 GETINFO\_IS関数 ..... 67  
 GETINFO\_RL関数 ..... 92  
 GETINFO\_SQ関数 ..... 108  
 GETRECNO関数 ..... 113  
 GetRecordDefメソッド ..... 238  
 GetTableDefメソッド ..... 230

## I

IFAS FILEイニシエータ ..... 17  
 IfasActionStatusプロパティ ..... 174  
 IFASBKSP ..... 425  
 IFASCOPY ..... 425  
 IFASDCLR ..... 426  
 IFASDRSM ..... 367, 427  
 IFASERAS ..... 428  
 IfasErrDupKeyStateプロパティ ..... 178  
 IfasErrEofStateプロパティ ..... 179  
 IfasErrStateプロパティ ..... 177  
 IFASFATR ..... 429  
 IFASFCNV ..... 430  
 IFASFRST ..... 430  
 IFASINFO ..... 431  
 IFASIXMN ..... 431  
 IFASLFRS ..... 432  
 IFASLTBI ..... 433  
 IfasMinorStatus1プロパティ ..... 175  
 IfasMinorStatus2プロパティ ..... 176  
 IFASデータ型 ..... 14  
 IFAS管理情報メンテナンスツール ..... 367  
 IFCPYCOM ..... 335, 338, 433  
 IFRDBUTL ..... 434

## J

JOIN ..... 12

**L**

LOGCLOSE関数.....	118
LogModeプロパティ.....	185
LOGOPEN関数.....	117

**M**

MIX互換機能.....	141
MIXファイル.....	140

**O**

OPEN_IS関数.....	42
OPEN_RL関数.....	73
OPEN_SQ関数.....	97
Openメソッド.....	192

**P**

PatternMaskプロパティ.....	189
PatternMatchプロパティ.....	188
POSITION_IS関数.....	56
POSITION_RL関数.....	86
PUT_IS関数.....	47
PUT_RL関数.....	78
PUT_SQ関数.....	101
PUTX_IS関数.....	52
PUTX_RL関数.....	82
PUTX_SQ関数.....	103

**R**

RDB環境の構築.....	310
<b>RDB環境の参照</b> .....	309
RDB環境のチェック.....	324
RDB環境の変更.....	319
RDB管理ファイル.....	255
RDB表のロックリセット.....	323
ReadRecordStringメソッド.....	221
ReadRecordメソッド.....	219
Readメソッド.....	195
REBUILD_IDXFILE_IS関数.....	70
REMAKEINDEX_IS関数.....	69
RENAME_IS関数.....	66
RENAME_RL関数.....	91
RENAME_SQ関数.....	107
RewriteRecordStringメソッド.....	225
RewriteRecordメソッド.....	224
Rewriteメソッド.....	198
RFD出力機能.....	332
RFD入力機能.....	331
RFDファイル.....	163, 291, 331, 332, 435
RFD編集ツール.....	163, 291

ROLLBACK関数.....	121
Rollbackメソッド.....	210

**S**

Scratchメソッド.....	208
Selectメソッド.....	204
Separatorプロパティ.....	187
SetColumnDataStringメソッド.....	216
SetColumnDataメソッド.....	215
SetIndexAccessKeyStringメソッド.....	218
SetIndexAccessKeyメソッド.....	217
SKIP_IS関数.....	63
SKIP_RL関数.....	88
SKIP_SQ関数.....	104
smart/cs.....	375
StartRecordStringメソッド.....	229
StartRecordメソッド.....	228
Startメソッド.....	202
STOPEND関数.....	114

**T**

Termメソッド.....	239
TransactionModeプロパティ.....	184
TXSTART関数.....	119

**W**

WRAPUP関数.....	115
WriteRecordStringメソッド.....	223
WriteRecordメソッド.....	222
Writeメソッド.....	197

**あ**

アクセスキー.....	12
アクセスモード.....	31

**い**

移行ユーティリティ.....	294
<b>一時ファイル機能</b> .....	269, 272
イベントログ.....	255
インデックス削除.....	319
インデックス作成.....	318
インデックスの変更.....	322
インデックスファイル.....	7
インデックスブロック.....	7
<b>インデックスブロックサイズ</b> .....	32
<b>インデックスブロックの占有率</b> .....	32
インデックスメンテナンス.....	281
インポート.....	329, 334

## う

運用支援ツール..... 352

## え

エクスポート..... 330, 334

エラーログ..... 361

## お

オプションブロック..... 42, 73, 97

## か

カスタムコントロール項目属性..... 16

仮想表..... 12, 150

合併型仮想表..... 13, 151

合併型仮想表の作成..... 314, 316

合併型仮想表の変更..... 322

環境設定..... 29, 307

管理ディレクトリ..... 308

## き

キーエントリ..... 6

キー記述構造体 40, 42, 56, 59, 61, 67, 93,  
109

キー記述ブロック..... 7

キーパート構造体... 40, 42, 56, 59, 93, 109

基本合併型仮想表..... 13

基本表..... 11, 150

基本表の作成..... 314

基本表の変更..... 320

共有バッファ..... 20

共用制御..... 125, 155

## く

クラス名..... 163

## け

結合..... 12

結合型仮想表..... 12, 150

結合型仮想表の作成..... 314, 315

## こ

更新属性..... 140

更新ログ..... 37, 124, 155, 249, 252

更新ログファイル..... 253

更新ログファイル設定..... 133, 162

**更新ログ未収集オプション**..... 32

コミット..... 37

コンポーネント..... 163

コンポーネント追加..... 163

## さ

サーバ管理ユーティリティ..... 366, 367

再構築機能..... 324

索引順編成ファイル..... 5, 6

## し

システム障害..... 253, 256

システム設定..... 17

**実行時生成**..... 140

順編成ファイル..... 10

シングルログ運用..... 250

## す

ステータスコード..... 137, 147, 378

## せ

セカンダリインデックス..... 7

セカンダリ表..... 12

セキュリティ..... 246

選択型仮想表..... 12, 150

選択型仮想表の作成..... 314

選択型仮想表の変更..... 321

選択合併型仮想表..... 13

## そ

関連名..... 12

相対編成ファイル..... 10

## た

対応キー..... 12

## ち

重複キー..... 9

## つ

ツリー構造..... 8

## て

定義ファイル. 272, 273, 285, 329, 334, 363

ディスク障害..... 38, 252, 255

ディファードリカバリ..... 38, 254, 346

データ型..... 15

データ定義ファイル..... 297

データの全件削除..... 328

データファイル..... 9

データベース..... 11

データベース削除..... 311

データベース作成..... 310

データベースのコピー..... 311

データベースの変更..... 320

データベース名 .....	308
デッドロック .....	36

## と

動作環境の設定 .....	130, 158
同時更新 .....	140
トランザクション処理 .....	36, 124, 154, 253

## な

内部ファイルID .....	127
----------------	-----

## に

入出力状態 .....	135, 144, 151
-------------	---------------

## は

バックアップ支援ユーティリティ .....	362
バッファ数 .....	33

## ひ

非同時更新 .....	140
表削除 .....	314
表定義ユーティリティ .....	307
表の移動 .....	318
表のコピー .....	317
表の別名保存 .....	314, 317
表名の関連付け .....	127

## ふ

ファイル移行 .....	294
ファイルオプション .....	127
ファイル共有ロック .....	31
ファイルコピー .....	270
ファイル情報構造体 .....	67, 92, 108
ファイルの指定 .....	127, 170
ファイル排他ロック .....	31
ファイルビジー .....	36
ファイル復旧 .....	252, 341
ファイルマッパー 指定例 .....	129
ファイル名 .....	123, 153
ファイル名の関連付け .....	127
ファイルメンテナンスキット .....	375
ファイルロック .....	35
ファイルロックリセット .....	350
不整合状態 .....	142

付属モジュール .....	163
付属モジュール追加 .....	163
プライマリインデックス .....	7
プライマリ表 .....	12
プロパティ .....	173

## ま

マルチレコードフォーマット .....	305
マルチログ運用 .....	250

## み

未クローズ .....	257
未クローズファイル .....	257, 350
未クローズロック .....	258
未クローズロック状態 .....	253, 323, 350
未使用管理ブロック .....	7

## め

メソッド .....	191
------------	-----

## よ

予約語 .....	339
-----------	-----

## ら

ラベルブロック .....	7
---------------	---

## り

リスタートリカバリ .....	38, 254
リモートデスクトップ .....	1

## れ

レコード番号 .....	6, 9, 10, 11
レコードロック .....	31, 36
列名規則 .....	312
連鎖 .....	12

## ろ

ロールバック .....	37, 254
ロールバックオプション .....	32
ロールフォワード .....	38, 252, 341
ロックテーブル .....	20
ロックモード .....	35



**IFAS FILE 、IFASPRO RDB Ver2.0**

**利用の手引**

2019年10月 第4版

日 本 電 気 株 式 会 社  
東京都港区芝五丁目7番1号  
TEL (03)3454-1111 (大代表)

©NEC Corporation 2014, 2019

日本電気株式会社の許可なく複製・改変などを行うことはできません。  
本書の内容に関しては、将来予告なしに変更することがあります。