

Adabas

コマンドリファレンス

バージョン 8.1.3

June 2008

This document applies to Adabas Version 8.1.3 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © Software AG 1971-2008. All rights reserved.

The name Software AG™, webMethods™, Adabas™, Natural™, ApplinX™, EntireX™ and/or all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. Other company and product names mentioned herein may be trademarks of their respective owners.

目次

1 コマンドリファレンス	1
2 構文形式	3
3 概要	5
4 Adabas コマンドについて	7
コマンドの種類	8
トランザクションと ET ロジック	17
競合更新	22
非アクティビティタイムリミット	26
5 プログラミング全般に関する考慮事項	29
コマンド ID、フォーマット ID、グローバルフォーマット ID	30
ISN リスト処理	37
マルチフェッチ/プリフェッチ機能の使用	42
6 Adabas の呼び出し	51
Adabas が ACB ダイレクトコールと ACBX ダイレクトコールを区別する方 法	53
ACB インターフェイスダイレクトコールの指定	53
ACBX インターフェイスダイレクトコールの指定	54
ACB ダイレクトコールと ACBX ダイレクトコールの混在	58
7 Adabas コントロールブロックの構造 (ACB および ACBX)	59
Adabas コントロールブロック (ACB)	61
拡張 Adabas コントロールブロック (ACBX)	67
ACB と ACBX の相違点	77
コントロールブロックのロギング	82
8 Adabas バッファ記述 (ABD)	85
使用可能な ABD タイプ	86
ABD の構造	87
ABD フィールドの説明	88
ABD DSECT	91
ABD リスト	91
9 バッファの定義	93
10 バッファタイプ間の関係	95
11 フォーマットバッファ	99
フォーマットバッファ構文	100
フィールドの選択条件	101
レコードフォーマットの指定	102
フォーマットバッファでの LA (ロング英数字) フィールドのフィールド長の 指定	123
フォーマットバッファのパフォーマンスに関する考慮事項	123
12 レコードバッファ	125
レコードバッファでの SQL 空値インジケータの指定と読み込み	127
レコードバッファでの LA (ロング英数字) フィールドのフィールド長の指 定	129
13 フォーマットバッファとレコードバッファの例	131

例 1：基本フィールド使用（標準の長さおよびフォーマット）	133
例 2：基本フィールド使用（長さおよびフォーマットを変更）	134
例 3：ピリオディックグループを参照	135
例 4：ピリオディックグループ GB の第 1、第 2 オカレンスを参照	136
例 5：マルチプルバリュースフィールド MF の 6 番目の値を参照	137
例 6：マルチプルバリュースフィールド MF の 1、2 番目の値を参照	138
例 7：ピリオディックグループ GC の最高オカレンス番号と、マルチプルバ リュースフィールド MF に存在する値の数	139
14 プリフェッチバッファ	141
15 マルチフェッチバッファ	143
16 サーチバッファ	145
サーチバッファの形式	147
物理カップリングファイル	153
ソフトカップリング	155
17 バリュースバッファ	157
バリュースバッファの構文	158
SQL 空値とインジケータ	158
符号の取り扱い	159
18 サーチバッファとバリュースバッファの例	161
例 1：単一の検索式の使用	162
例 2：AND で結合した検索式の使用	162
例 3：OR で結合した検索式の使用	163
例 4：FROM-TO で結合した検索式の使用	164
例 5：BUT-NOT で結合した検索式の使用	164
例 6：マルチプルバリュースディスクリプタの使用	165
例 7：ピリオディックグループ内のディスクリプタの使用	165
例 8：サブディスクリプタの使用	166
例 9：英数字フォーマットのスーパーディスクリプタの使用	167
例 10：バイナリ形式のスーパーディスクリプタの使用	167
例 11：以前に生成した ISN リストの使用	167
例 12：値演算子の使用	168
例 13：値演算子および結合演算子の使用	169
例 14：物理カップリングファイルの使用	169
例 15：単一ソフトカップリング条件と単一検索条件の使用	170
例 16：単一のソフトカップリング条件と複数の検索条件の使用	170
例 17：複数のソフトカップリング条件と複数の検索条件の使用	171
例 18：複数のソフトカップリング条件と、物理カップリングによる複数の検索 条件の使用	171
19 ISN バッファ	173
20 ユーザーバッファ	175
21 パフォーマンスバッファ	177
22 コマンド	179
23 A1 コマンド：レコードの更新	181
機能および使用	182
ACB インターフェイスダイレクトコール：A1 コマンド	182

ACBX インターフェイスダイレクトコール：A1 コマンド	188
バッファ	193
その他の考慮事項	193
24 BT コマンド：トランザクションのバックアウト	199
機能および使用	200
ACB インターフェイスダイレクトコール：BT コマンド	200
ACBX インターフェイスダイレクトコール：BT コマンド	204
ISN バッファ	208
25 C1 コマンド：チェックポイントの書き込み	209
機能および使用	210
ACB インターフェイスダイレクトコール：C1 コマンド	210
ACBX インターフェイスダイレクトコール：C1 コマンド	213
26 C3 コマンド：チェックポイントの書き込み	217
機能および使用	218
ACB インターフェイスダイレクトコール：C3 コマンド	218
ACBX インターフェイスダイレクトコール：C3 コマンド	221
バッファ	223
27 C5 コマンド：プロテクションログへのユーザーデータの書き込み	225
機能および使用	226
ACB インターフェイスダイレクトコール：C5 コマンド	226
ACBX インターフェイスダイレクトコール：C5 コマンド	229
バッファ	231
28 CL コマンド：ユーザーセッションのクローズ	233
機能および使用	234
ACB インターフェイスダイレクトコール：CL コマンド	234
ACBX インターフェイスダイレクトコール：CL コマンド	238
バッファ	241
29 E1 コマンド：レコードの削除／ファイルのリフレッシュ	243
機能および使用	244
ACB インターフェイスダイレクトコール：E1 コマンド	244
ACBX インターフェイスダイレクトコール：E1 コマンド	248
30 ET コマンド：トランザクションの終了	253
機能および使用	254
ACB インターフェイスダイレクトコール：ET コマンド	255
ACBX インターフェイスダイレクトコール：ET コマンド	259
バッファ	262
31 HI コマンド：レコードのホールド	265
機能および使用	266
ACB インターフェイスダイレクトコール：H1 コマンド	266
ACBX インターフェイスダイレクトコール：H1 コマンド	269
32 L1/L4 コマンド：読み込み／読み込みおよびレコードのホールド	273
機能および使用	274
ACB インターフェイスダイレクトコール：L1 コマンドおよび L4 コマンド	275
ACBX インターフェイスダイレクトコール：L1 コマンドおよび L4 コマ ンド	285

バッファ	290
33 L2/L5 コマンド：物理順に読み込み	293
機能および使用	294
ACB インターフェイスダイレクトコール：L2 コマンドおよび L5 コマンド	294
ACBX インターフェイスダイレクトコール：L2 コマンドおよび L5 コマ ンド	300
バッファ	305
その他の考慮事項	306
34 L3/L6 コマンド：論理順に読み込み	307
機能および使用	308
ACB インターフェイスダイレクトコール：L3 コマンドおよび L6 コマンド	309
ACBX インターフェイスダイレクトコール：L3 コマンドおよび L6 コマン ド	319
バッファ	326
その他の考慮事項	331
35 L9 コマンド：ディスクリプタ値の読み込み	333
機能および使用	334
ACB インターフェイスダイレクトコール：L9 コマンド	334
ACBX インターフェイスダイレクトコール：L9 コマンド	342
バッファ	348
その他の考慮事項	352
36 LF コマンド：フィールド定義の読み込み	353
機能および使用	354
ACB インターフェイスダイレクトコール：LF コマンド	354
ACBX インターフェイスダイレクトコール：LF コマンド	357
バッファ	360
37 N1 コマンドと N2 コマンド：レコードの追加	369
機能および使用	370
ACB インターフェイスダイレクトコール：N1 コマンドおよび N2 コマンド	370
ACBX インターフェイスダイレクトコール：N1 コマンドおよび N2 コマン ド	375
バッファ	380
38 OP コマンド：ユーザーセッションのオープン	385
機能および使用	386
ユーザータイプ	387
ACB インターフェイスダイレクトコール：OP コマンド	388
ACBX インターフェイスダイレクトコール：OP コマンド	395
バッファ	401
ユーザーキューエレメント	403
タイムリミットの超過	404
コントロールブロックフィールドに返される値	404
39 RC コマンド：コマンド ID／グローバルフォーマット ID の解放	407
機能および使用	408
ACB インターフェイスダイレクトコール：RC コマンド	409
ACBX インターフェイスダイレクトコール：RC コマンド	413

40 RE コマンド：ET ユーザーデータの読み込み	419
機能および使用	420
ACB インターフェイスダイレクトコール：RE コマンド	420
ACBX インターフェイスダイレクトコール：RE コマンド	425
バッファ	428
41 RI コマンド：レコードの解放	431
機能および使用	432
ACB インターフェイスダイレクトコール：RI コマンド	432
ACBX インターフェイスダイレクトコール：RI コマンド	434
42 S1、S2、S4 コマンド：レコードの検索	439
機能および使用	440
ACB インターフェイスダイレクトコール：S1、S2、S4 の各コマンド	441
ACBX インターフェイスダイレクトコール：S1、S2、S4 の各コマンド	453
バッファ	459
43 S5 コマンド：カップリングリストの検索	461
機能および使用	462
ACB インターフェイスダイレクトコール：S5 コマンド	462
ACBX インターフェイスダイレクトコール：S5 コマンド	466
バッファ	470
44 S8 コマンド：ISN リストの処理	473
機能および使用	474
ACB インターフェイスダイレクトコール：S8 コマンド	475
ACBX インターフェイスダイレクトコール：S8 コマンド	479
バッファ	483
45 S9 コマンド：ISN リストのソート	485
機能および使用	486
ACB インターフェイスダイレクトコール：S9 コマンド	486
ACBX インターフェイスダイレクトコール：S9 コマンド	492
バッファ	497
46 プログラミング例	499
47 例で使用されているファイル定義	501
48 ACB の例	503
ACB アセンブラの例	504
ACB COBOL の例	511
ACB PL/I の例	518
ACB Fortran の例	525
目次	529

1 コマンドリファレンス

このドキュメントでは、データベース処理を実行する強力かつ柔軟な Adabas ダイレクトコールコマンドについて説明します。Natural などの第4世代データベース言語を使用しない場合は、Adabas ダイレクトコールコマンドが Adabas データベースへの直接インターフェイスとなります。



Note: このドキュメントでは、DD で始まるデータセット名は、DD 接頭辞を含まない VSE データセット名と区別できるように、DD と残りのデータセット名の間にはスラッシュを入れて示されています。スラッシュはデータセット名の一部ではありません。

『Adabas コマンドリファレンス』は、次の部分で構成されています。

● 概要	Adabas コマンドの概要、および Adabas コールを発行する場合のプログラミング全般に関する考慮事項を示します。
● Adabas の呼び出し	ACB インターフェイスと ACBX インターフェイスの Adabas コマンドをコールする手順について説明します。
● Adabas コントロールブロックの構造	Adabas コールの発行時に使用できる Adabas コントロールブロック (ACB および ACBX) の構造について説明します。
● Adabas バッファ記述 (ABD)	Adabas に ACBX インターフェイスコールを発行する場合の Adabas バッファ記述 (ABD) の構造とその使用方法について説明します。ABD は、Adabas 8 以降のリリースでサポートされています。
● バッファの定義	Adabas コールを発行する場合のバッファ定義の作成とその使用方法について説明します。
● コマンド	Adabas コール内で使用できる各 Adabas コマンドについて詳しく説明します。
● プログラミング例	各種ホスト言語による Adabas コールのプログラミング例を示します。

2 構文形式

このチャプターでは、ダイレクトコール、バッファ記述、およびバッファ指定に関してこのドキュメントで使用する構文形式について説明します。特別なバッファの表記については、本章で後述します。

表記規則	識別対象	説明	例
大文字、 太字	Adabas キーワ ード	太字の大文字で表示される構文要素は Adabas キーワードです。これらのキーワードを指定する際に、表示されているとおりに入力する必要があります。	<i>field</i> NC 構文要素 NC は Adabas キーワードです。
小文字、 斜体、標 準フォ ント	変数	斜体のかかった標準フォントの小文字で表示される構文要素は、ユーザーが入力すべき項目を表します。	<i>field</i> <i>i</i> [- <i>j</i>] 構文要素 <i>field</i> 、 <i>i</i> 、および <i>j</i> は変数です。これらの変数には値を指定する必要があります。
縦線 ()		縦線は、相互に排他的な選択肢の区切りを表します。 注意: 角カッコや中カッコを使用した、より複雑な構文では、相互排他的な選択肢が入れ子になって記述されます。	a b 上の例では、"a" または "b" を選択する必要があります。デフォルト値はありません。
角カッコ ([])	オプシ ョンの 要素 または 選 択 肢	角カッコは、選択型の要素を表します。角カッコ内で複数の要素が、入れ子になって記述されたり、縦線で区切られていたりするときには、その中の1つだけを使用できます。	<i>field</i> [, <i>length</i>] この例では、長さパラメータがオプションです。
中カッコ ({})	必須の 要素 または 選 択 肢	中カッコは、必須の要素を表します。角カッコ内で複数の要素が、入れ子になって記述されたり、縦線で区切られていたりするときに	{c s} この例では、"C" または "S" のいずれかを指定する必要があります。

構文形式

表記規則	識別対象	説明	例
		は、その中の1つだけを使用する必要があります。	
字下げ	サブパラメータ	字下げは、パラメータのサブパラメータを表します。	
省略記号 (...)	繰り返しの要素	省略記号は、繰り返して使用できる要素を表します。他の区切り文字を省略記号の前に記述した場合は、その区切り文字を使用して、繰り返しの要素を区切る必要があります。省略記号を角カッコまたは中カッコの外に置くと、角カッコまたは中カッコ内の全要素が繰り返し可能な要素とみなされます。	<p><i>field ...</i></p> <p>この例では、<i>field</i>は必須であり、オプションで繰り返しの要素にすることもできます。</p> <p><i>field1[, length1][,field2[,length2]...</i></p> <p>この例では、<i>field1</i>は必須であり、<i>length1</i>はオプションです。また、追加のフィールドおよび長さ (<i>field2</i> および <i>length2</i> で指定) をコンマで区切って指定することもできます。</p>
<i>b...</i>	空白	斜体の小文字 <i>b</i> は、単独またはグループ (<i>bbbbbbb</i> など) で使用した場合、1つまたは一連の空白を示します。各 <i>b</i> には、空白が1つ入ります。	このフィールドの値として "ISN <i>bbbbbb</i> " を指定すると、その ISN 値がソート順として使用されることとなります (<i>bbbbbb</i> には空白が入ります)。
その他の区切り文字と記号	必須の区切り文字	その他のすべての区切り文字と記号は、表示されたとおりに入力する必要があります。	<p><i>fmtsel redfmt .</i></p> <p>この例では、ピリオドが必須です。</p>

3 概要

『Adabas コマンドリファレンスマニュアル』のこの章では、Adabas コマンドの概要、および Adabas コールを発行するとき、プログラミング全般に関して考慮する必要がある事項を示します。Adabas コールの発行に使用する手順については、「[Adabas の呼び出し](#)」で説明します。

情報は次の項目で構成されています。

- [Adabas コマンドについて](#)
- [プログラミング全般に関する考慮事項](#)

4 Adabas コマンドについて

■ コマンドの種類	8
■ トランザクションと ET ロジック	17
■ 競合更新	22
■ 非アクティビティタイムリミット	26

このセクションでは、Adabas コマンドの概要を示します。コマンドは、機能ごとにデータベース問い合わせ、データベース更新、データストレージ読み込み、アソシエータ読み込み、論理トランザクション処理、および特殊処理コマンドに分類できます。

また、データ保護、リカバリ、およびユーザー再スタートに関連する Adabas コマンド機能についても説明します。トランザクションという概念を紹介し、ET ロジックオペレーションについて説明します。ET ロジック（レコードのホールド／解放、およびリソースのデッドロック回避）ユーザーと排他制御ユーザーを対象に競合更新について説明します。また、全ユーザータイプを対象に非アクティビティタイムリミットについても説明します。

このchapterでは、次のトピックについて説明します。

コマンドの種類

コマンドは、次の機能に分類することができます。


- データベース問い合わせコマンド (Sx)
- データストレージ読み込み (L1~L6)
- アソシエータ読み込み (L9、LF)
- データベース更新 (A1、E1、N1/N2)
- 論理トランザクション制御コマンド (ET/BT)
- 特殊コマンド

データベース問い合わせコマンド (Sx)

データベース問い合わせコマンド (S1/S4、S2、S5) は、指定した検索条件に応じて、レコードまたはレコードグループの ISN を検索して返します。その他のこの分類に属するコマンド (S8、S9) は、後から処理するための準備として、ISN 結果リストをソートします。

Sx コマンドで作成した ISN リストは、Adabas WORK データセットに格納しておく、後でユーザーセッション中に取り出すことができます。

これらのコマンドはほとんどの場合、実際にはデータベースを読み取りません。ISN はアソシエータのインバーテッドリストから直接読み込まれます。任意で、実行結果の ISN レコードをホールド状態にして、そのレコードが解放されるまで他のユーザーによる更新を防止できます。また、必要に応じて、先頭の ISN レコードに含まれているフィールド値をデータストレージから読み込むことができます。

 **Note:** Adabas データベースにおける非ディスクリプタ値の検索は、メインフレームの場合とオープンシステムの場合とで、フィールド内の空値省略に関して違いがあります。オープンシステムではフィールドが空値省略 (NU) の場合、非ディスクリプタ値を検索したときに空値を含むレコードを返しません。メインフレームシステムでは、非ディスクリプタ値を検索したときにフィールドの空値省略 (NU) は無視されます。このときフィー

ルドを非ディスクリプタ値の検索で使用する必要がある場合は、オープンシステムフィールドの空値省略 (NU) オプションを解除してこの問題を解決することをお勧めします。

このsectionでは、次のデータベース問い合わせコマンドについて説明します。

- S1/S4
- S2
- S5
- S8
- S9

S1/S4

S1/S4 コマンドでは、指定された検索条件を満たすレコードを選択します。

- ディスクリプタのみを使用している場合は、データストレージにアクセスすることなく、インバーテッドリストのみを使用して、問い合わせを解決します。
- 検索条件にディスクリプタを1つも使用していない場合は、データストレージの各レコードを読み込んで問い合わせを解決します。
- 検索条件にディスクリプタと非ディスクリプタの両方を使用している場合は、まずインバーテッドリストでディスクリプタを検索し、次にデータストレージを読み込んで非ディスクリプタと一致するすべての ISN を調べます。

S1/S4 コマンドでは、検索条件を満たすレコードの数と、それらのレコードの ISN を記載したリストを返します。オプションとして、ISN 結果リストの先頭にある ISN のレコードをデータストレージから読み込むこともできます。

S4 コマンドを使用すると、ISN リストの先頭にある ISN のレコードをホールド状態にすることができます。ホールドされたレコードが解放されるまで、他のユーザーはそのレコードを更新できません。

S2

S2 コマンドでは、選択したレコードの ISN をユーザー指定のディスクリプタのソート順で返すことを除いて、S1 コマンドと同じです。1~3個のディスクリプタを使用できます。昇順または降順を指定できます。

S5

S5 コマンドでは、指定した ISN を持つ一方のファイル中のレコードとカップリングされた他方のファイル中のレコードの ISN を選択します。

コマンドには、カップリングされた ISN を返すファイルと ISN、およびカップリングされたレコードを選択するファイルを指定します。Adabas は、ISN にカップリングされたレコードの数と、そのカップリングされた ISN のリストを返します。

S8

S8 コマンドでは、以前に S1/S4、S5、S8、S9 のいずれかのコマンドで作成した 2 つの ISN リストに論理演算（AND、OR、または NOT）を実行します。

- AND は、2 つの ISN リストのどちらにもある ISN のリストを作成します。
- OR は、2 つの ISN リストのどちらかにある ISN のリストを作成します。
- NOT は、第 1 ISN リストにあって第 2 ISN リストにない ISN のリストを作成します。

S9

S9 コマンドでは、以前に S1/S4、S2、S5、S8、S9 のいずれかのコマンドで作成した ISN リストをソートします。

ISN リストは、ISN（昇順に限る）または 1~3 個のユーザー指定ディスクリプタ（昇順または降順）でソートできます。

データストレージ読み込み（L1~L6）

L1~L6 コマンドは、データストレージから実際のレコードを読み込みます。指定したコマンドおよびそのオプションに応じて、レコードは格納されている順、データベース問い合わせコマンドのいずれかで作成した ISN リストの順、またはユーザー指定のディスクリプタによる論理順に、個別に読み込まれます。

ホールドオプションを使うと、専用のコマンドでレコードを解放するか、トランザクションが終了するまで、データベースレコードをロックできます。

この section では、次のデータストレージ読み込みコマンドについて説明します。

- L1/L4
- L2/L5

■ L3/L6

L1/L4

L1 コマンドでは、データストレージから単一レコードを読み込みます。コマンドには、ファイル番号、読み込むレコードの ISN、および値を取得するフィールドを指定します。Adabas は要求フィールドの値を返します。

L4 コマンドは、レコードをホールド状態にすることを除いて、L1 コマンドと同じです。レコードが解放されるまで、他のユーザーはレコードを更新できません。

マルチフェッチ／プリフェッチオプションは、セッションベースまたはコマンドベースでレコードをプリフェッチします。これにより、複数のレコードフェッチによって生じるオーバーヘッドが減少します。マルチフェッチオプションは、プラットフォームに依存しません。

GET NEXT オプションを使用すると、ISN を指定することなく、ISN リスト内の ISN で識別されるレコードを読み込むことができます。通常、ISN リストは以前の Sx コマンドによって作成されたものです。

レスポンスコードオプションを指定すると、他のユーザーがレコードをホールドしているために L4 コマンドがレコードをホールドできない場合、レスポンスコード 145 が返されます。それ以外の場合は待機状態になり、その状態は ISN（およびレコード）が解放されるか、または待機をもち続けているユーザーのトランザクションがタイムアウトするまで続きます。

ISN 順オプションを使用すると、レコードを ISN 順に読み込むことができます。指定した ISN が存在する場合は、該当する ISN のレコードが読み込まれ、存在しない場合は、次に高い ISN のレコードが読み込まれます。

L2/L5

L2 コマンドでは、ファイルからレコードをデータストレージに物理的に格納されている順で読み込みます。コマンドには、ファイル番号と値を取得するフィールドを指定します。Adabas は要求フィールドの値を返します。

L5 コマンドは、読み込んだレコードをホールド状態にすることを除いて、L2 コマンドと同じです。レコードが解放されるまで、他のユーザーはレコードを更新できません。マルチフェッチ／プリフェッチオプションおよびレスポンスコードオプション（上記の L1/L4 コマンドの説明を参照）は、L2/L5 コマンドにも適用できます。

L3/L6

L3 コマンドでは、データストレージからレコードをユーザー指定ディスクリプタの論理順で読み込みます。コマンドには、ファイル番号、順序制御に使用するディスクリプタ、および値を取得するフィールドを指定します。Adabas は要求フィールドの値を返します。

L3/L6 コマンドのコマンドオプション2によって、レコードを昇順または降順のどちらで読み込むかを指定します。また、コマンドオプション2を使用して、開始値も指定できます。

L6 コマンドは、読み込んだレコードをホールド状態にすることを除いて、L3 コマンドと同じです。レコードが解放されるまで、他のユーザーはレコードを更新できません。

マルチフェッチ/プリフェッチオプションおよびレスポンスコードオプション（上記のL1/L4 コマンドの説明を参照）の他に、開始値オプションを使用してユーザー指定の値またはISNで読み込みを開始することもできます。

アソシエータ読み込み (L9、LF)

L9 コマンドと LF コマンドでは、アソシエータインバーテッドリストまたはフィールド定義テーブル (FDT) から直接情報を読み込みます。このコマンドによって返される値は、指定したディスクリプタのインバーテッドリスト値、または指定したデータベースファイルのフィールド定義のいずれかです。

このsectionでは、次のアソシエータ読み込みコマンドについて説明します。

- L9
- LF

L9

L9 コマンドでは、あるディスクリプタのインバーテッドリスト内の各値と、その値を持つレコード数を返します。

コマンドには、ファイル番号と値を取得するディスクリプタ、コマンド処理を開始する値を指定します。値は昇順または降順で返されます。

LF

LF コマンドでは、ファイルのフィールド定義を返します。コマンドには、フィールド定義を取得するファイルを指定します。

ファイル内の全フィールドのフィールド定義が返されます。各フィールド定義は、フィールド名、レベル番号、標準フォーマット、標準長、および定義オプションで構成されています。

データベース更新 (A1、E1、N1/N2)

データベース更新コマンド (A1、E1、および N1/N2) は、データベースレコードを追加、変更、または削除し、関連するアソシエータリストを適宜更新します。新しいレコードに自分で ISN を割り当てることも、Adabas によって割り当てることもできます。

このsectionでは、次のデータベース更新コマンドについて説明します。

- A1
- E1
- N1/N2

A1

A1 コマンドでは、レコード内の1つ以上のフィールドの内容を更新します。コマンドには、更新するレコードのファイル番号と ISN の他、更新するフィールドと更新する値も指定します。

Adabas は、アソシエータおよびデータストレージに必要な修正を行います。アソシエータ更新が必要になるのは、ディスクリプタを更新する場合に限られます。

更新前にレコードをホールド状態にするホールドオプションを使用できます。

E1

E1 コマンドでは、レコードを削除したり、ファイルをリフレッシュしたりします。削除する場合はレコードのファイル番号と ISN を指定し、ファイルをリフレッシュする場合は ISN やコマンド ID ではなくファイルのみを指定します。

Adabas は、アソシエータおよびデータストレージに必要な修正を行います。

削除前にレコードをホールド状態にするためのホールドオプションを使用できます。

N1/N2

N1/N2 コマンドでは、ファイルに新しいレコードを追加します。コマンドには、レコードを追加するファイルの他、使用するフィールドとその値を指定します。Adabas は、アソシエータおよびデータストレージに必要な修正を行います。

N1 コマンドを使用する場合、新規レコードの ISN は Adabas が割り当てます。N2 コマンドを使用する場合、自分で ISN を指定する必要があります。

論理トランザクション制御コマンド (ET/BT)

Adabas 論理トランザクションは、実行中のデータベースオペレーションの論理的な開始 (BT) と終了 (ET) を定義します。論理トランザクションコマンドを使用する主な目的は、ユーザーまたは Adabas の異常終了時、正常に処理されなかった最後のトランザクションからユーザーを再開できるようにすることです。ET/BT コマンドでは、トランザクションの開始と終了を定義し、トランザクションの正常終了を妨げる状況が発生した場合にはこのトランザクションを処理する前の状態にリストアし、トランザクションシーケンスで書き込まれたプログラム指定のユーザーデータを読み込みます。

これらのコマンドを使用するプログラムを ET ロジックプログラムと呼びます。必須ではありませんが、ET ロジックを使用することをお勧めします。

この section では、次の論理トランザクション制御コマンドについて説明します。

- BT
- ET

BT

BT コマンドでは、現在処理中のトランザクションをバックアウトします。

トランザクションで実行した更新処理は削除され、トランザクションでホールド状態にされたレコードはすべて解放されます (ただし、マルチフェッチオプションによってホールドされたレコードを除きます。詳細については、「[マルチフェッチ処理](#)」セクションを参照してください)。

ET

ET コマンドでは、論理トランザクションの終了を示します。

ET コマンドを実行すると、トランザクションに関連したデータプロテクション情報がすべて物理的に格納されます。トランザクションで実行した更新処理を物理的にデータベースに適用する前に、現在の Adabas セッションが終了した場合、Adabas はこのデータプロテクション情報を使用して、このトランザクションによる更新処理を次の Adabas セッション開始時に適用します。

ET コマンドでは、トランザクションの処理中、ホールド状態になっているレコードをすべて解放します (ただし、マルチフェッチオプションによってホールドされたレコードを除きます。詳細については、「[マルチフェッチ処理](#)」セクションを参照してください)。また ET コマンドを使用して、Adabas システムファイルにユーザーデータの格納もできます。このデータは、OP または RE コマンドで読み込むことができます。

特殊コマンド

特殊コマンドは、Adabas データベース環境をメンテナンスするのに必要な維持管理の役割を多数行います。このグループのコマンドでは、次の機能を実行できます。

- セッションのオープンおよびクローズ（ただし、トランザクションの制御は行いません）
- データプロテクション情報とチェックポイントの書き込み
- レコードホールド状態の設定および解放

このsectionでは、次の特殊コマンドについて説明します。

- CL
- C1
- C3
- C5
- HI
- OP
- RC
- RE
- RI

CL

CL コマンドでは、ユーザーセッションを終了し、このユーザー用にホールドされていたレコードをすべて解放します。

- 現在のデータプロテクション情報をすべてデータプロテクションログに物理的に書き込みます。
- このユーザーに対して現在ホールド状態にあるレコードをすべて解放します。
- このユーザーに現在割り当てられているコマンド ID および対応する ISN リストをすべて解放します。
- Adabas システムファイルにユーザーデータを格納します（オプション）。

C1

C1 コマンドでは、チェックポイントを取得します。

C1 コマンドでは、現在のデータプロテクション情報をすべてデータプロテクションログに物理的に書き込み、チェックポイントエントリをデータプロテクションログおよびシステムチェックポイントファイルに書き込みます。このチェックポイントエントリは、後で更新を取り消したり、再適用したりするための参照ポイントとして使用できます。C1 コマンドでは、オプションで、バッファフラッシュを初期化することもできます。

C3

C3 コマンドは排他制御更新ユーザー（ET ロジックを使用していないユーザー）のみが発行できるコマンドで、その機能は Adabas チェックポイントファイルに SYNX-03 チェックポイントを書き込むことです。

- チェックポイントには、現在のデータプロテクションログおよびブロック番号が含まれています。
- このチェックポイントエントリを使用すると、データベース（または特定のファイル）をチェックポイント取得時の状態にリストアできます。排他制御更新処理するプログラムを再実行または再スタートするためには、事前にチェックポイントまで戻さなければなりません。

コマンドオプション2を指定すると、再スタート用にユーザーデータも Adabas チェックポイントファイルに格納されます。格納されたデータは、後続の OP コマンドまたは RE コマンドで読み取れます。

C5

C5 コマンドでは、ユーザーデータを Adabas データプロテクションログに書き込みます。

後で ADASEL ユーティリティを使用して、このデータを読み込むことができます。

HI

HI コマンドでは、レコードをホールド状態にします。ホールド状態にするレコードがあるファイルとその ISN を指定します。

レコードがホールド状態になると、解放されるまで他のユーザーは更新できません。

OP

OP コマンドは、下記に該当する場合に必須です。

- ADARUN パラメータ OPENRQ=YES を指定してニュークリアスを実行します。
- 排他ファイル制御（EXF）を実行します。
- 以前に ET コマンドで Adabas システムファイルに格納したユーザーデータを読み込みます。
- C3、CL、ET のいずれかのコマンドを使用して、ユーザーデータを Adabas システムファイルに格納します。
- 自分に特別な優先権を割り当てます。
- 自分をアクセスオンリーユーザー（更新コマンドが許可されていないユーザー）にします。
- ADARUN パラメータ TT または TNA_x で指定した値とは異なるトランザクションタイムリミットまたは非アクティビティタイムリミットを設定します。これらのタイムリミットはそれぞれ、ADARUN のパラメータ MXTT と MXTNA で設定した最大値に一致する必要があります。

- ユーザーセッション用に特殊なデータエンコードまたはアーキテクチャを指定します。

また、ホールド状態にできる最大レコード数、および同時にアクティブにできるコマンド ID の最大数を設定する場合にも使用できます。

RC

RC コマンドを使用すると、現在割り当てられている 1 つ以上のコマンド ID を解放したり、1 つまたはすべてのグローバルフォーマット ID を削除したりできます。

RE

RE コマンドでは、以前に CL コマンドまたは ET コマンドで Adabas システムファイルに格納したユーザーデータを読み込みます。

RI

RI コマンドでは、ホールド状態からレコードを解放します。

解放するレコードのファイルと ISN を指定します。また、現在自分がホールドしているすべてのレコードを解放するように要求することもできます。RI コマンドは、非 ET ユーザーのみが発行してください。

トランザクションと ET ロジック

このセクションでは、トランザクションの概念および ET ロジックユーザーについて説明します。また、トランザクションの通常終了処理および異常終了処理、ユーザー（ET）データの格納および取得など、ET ロジックオペレーションについても説明します。

このsectionでは、次のトピックについて説明します。

- 論理トランザクションとは
- トランザクションシーケンス番号
- ET トランザクションタイムリミット
- トランザクションのバックアウト (BT) コマンド
- 自動バックアウト
- エンドトランザクション (ET) コマンド
- ユーザー (ET) データ

■ Adabas ユーザー ID

論理トランザクションとは

論理トランザクションとは、データベース内の情報を論理的に矛盾のないものにするために、ひとまとめに実行する必要がある（ユーザーが定義した）最小業務単位です。論理トランザクションコマンドを使用するユーザーを ET ロジックユーザーと呼びます。

論理トランザクションは、論理的な業務単位を完了するために必要に応じてデータベースを読み込んだり、更新したりする 1 つ以上の Adabas コマンドで構成されています。論理トランザクションは、レコードをホールド状態にするコマンドで始まり、同じユーザーに対して ET、BT、CL、OP のいずれかのコマンドを発行した時点で終了します。

RE（ET データの読み込み）コマンドを使用すると、ET コマンドまたは CL コマンドで格納したユーザー再スタートデータを取得できます。

トランザクションシーケンス番号

Adabas は、プログラムから ET コマンドまたは CL コマンドが発行されると、ET コマンドまたは CL コマンドのコントロールブロックのコマンド ID フィールドにトランザクションシーケンス番号を返します。トランザクションシーケンス番号は、ユーザーセッション中に発行された ET コマンドの総数です。

ユーザーが発行した最初の ET コマンドは、トランザクションシーケンス番号 1 に設定されます。OP コマンドに続く最初の ET コマンドは、コマンド ID フィールドまたは Adabas コントロールブロックにトランザクションシーケンス番号 2 を返します。後続の各 OP コマンドは、同じユーザーが最後に発行した ET コマンドのトランザクションシーケンス番号を返します。

ET トランザクションタイムリミット

Adabas では、ET ロジックを使用するプログラムにトランザクションタイムリミットを設定できます。トランザクション時間の測定は、最初のコマンドがレコードをホールド状態にした時点から始まり、プログラムが ET、BT、CL のいずれかのコマンドを発行した時点で終了します。

タイムリミットは、ADARUN TT パラメータで設定します。特定のユーザーのトランザクションタイムリミットを変更するには、OP コマンドを使用します。このタイムリミットは、ADARUN MXTT パラメータで制御します。

トランザクションが所定のタイムリミットを超えると、Adabas は BT（トランザクションのバックアウト）コマンドを生成します。BT コマンドでは、トランザクションで実行した更新処理をすべて削除し、ホールドしたレコードをすべて解放します。

ユーザーが次にコマンドを発行すると Adabas からレスポンスコード 9 が返されて、現在のトランザクションがバックアウトされたことが示されます。ユーザーは、バックアウトされたトランザクションを初めから繰り返すことも、他のトランザクションを開始することもできます。

トランザクションのバックアウト (BT) コマンド

BT コマンドでは、現在処理中のトランザクションですでに実行した更新処理をすべて削除します。この処理は、プログラムエラーやタイムアウトが発生した場合、またはトランザクションを正常に完了できない場合に必要になります。

BT コマンドでは、暗黙で ET コマンドも実行します。つまり、マルチフェッチオプションで特に指定されない限り、トランザクションでホールドされたレコードをすべて解放します。詳細については、「[マルチフェッチ処理](#)」セクションを参照してください。

例えば、次のコマンドシーケンスを実行したとします。

```
FIND (S4)
  UPDATE (A1) (modify field XX to value 20)
  FIND (S4)
  UPDATE (A1) (modify field YY to value 50)
  END TRANSACTION (ET)
  FIND (S4)
  UPDATE (A1) (modify field XX to value 10)
  BACKOUT TRANSACTION (BT)
```

その結果、フィールド値は XX=20 および YY=50 となります。フィールド XX に対する 2 番目の更新が、BT コマンドによって削除されます。

自動バックアウト

自動バックアウトは ET ロジックユーザーに対してのみ実行され、異常終了したセッションで自動的にトランザクションをバックアウトします。Adabas は、内部 BT (トランザクションのバックアウト)、続いて自動再スタートを実行してから、レスポンスコード 9 を返して、最後のトランザクションがバックアウトされたことを示します。

自動バックアウトは、HALT で終了したニュークリアスセッションの最後に発生します。

また、自動バックアウトは、Adabas が異常終了した次の Adabas セッションの始めにも実行され、ET ロジックユーザーが前回のセッションで実行した不完全なトランザクションによる更新をすべて削除します。

レスポンスコード 9 でトランザクションのバックアウトが示された場合、ユーザーはトランザクションを最初から再発行するか、または別のトランザクションを開始するかを選択できます。

バックアウトされたトランザクションを再開する場合は、端末オペレータがトランザクションのデータを再入力するか、またはトランザクションの更新フェーズの最初から内部での再開を実行する必要があります。トランザクションの更新フェーズは、レコードをホールド状態にする最初のコマンドから始まります。

エンドトランザクション (ET) コマンド

ET コマンドは、論理トランザクションの終了時点で発行しなければなりません。ET コマンドが正常に実行されると、後続のユーザーまたは Adabas セッションの中断に関係なく、トランザクションで実行した更新処理がすべて物理的にデータベースに適用されます。

ET コマンドが正常に実行されなかった場合は、トランザクションで実行した更新処理が自動バックアウトルーチンでバックアウトされます（「[自動バックアウト](#)」を参照）。

マルチフェッチオプションで特に指定されない限り、ET コマンドはトランザクションでユーザーがホールドしたレコードをすべて解放します。Adabas は、ユニークなトランザクションシーケンス番号を返します（「[トランザクションシーケンス番号](#)」を参照）。この番号を使用すると、監査や再開目的でトランザクションを特定できます。

また ET コマンドを使用して、Adabas システムファイルにユーザーデータの格納もできます。このデータはユーザー再スタートのために使用でき、OP コマンドまたは RE コマンドで読み込むことができます。

ユーザープログラム	Adabas
FIND (S4) 、 UPDATE (A4)	レコードは Adabas バッファ内で更新されるが、必ずしもデータベースに書き込まれるわけではない
FIND (S1) 、 HOLD ISN (HI) 、 UPDATE (A4)	レコードは Adabas バッファ内で更新されるが、必ずしもデータベースに書き込まれるわけではない
トランザクションの終了 (ET)	トランザクションのデータプロテクション情報を Adabas ワークおよびデータプロテクションログに書き込み、トランザクションを終了します。
FIND (S4) 、 UPDATE (A4)	レコードは Adabas バッファ内で更新されるが、必ずしもデータベースに書き込まれるわけではない
FIND (S1) 、 HOLD ISN (HI) 、 UPDATE (A4)	レコードは Adabas バッファ内で更新されるが、必ずしもデータベースに書き込まれるわけではない
Adabas またはユーザーセッション中断	

次の Adabas セッションが始まるか、またはユーザーがタイムアウトした時点で、トランザクション1のどちらの更新も物理的にデータベースに書き込まれます（まだ書き込まれていなかった場合）。トランザクション2の更新は、このトランザクションでは ET コマンドが処理されなかったため、物理的に書き込まれません（またはバックアウトされます）。

ユーザー (ET) データ

ユーザーデータ (ET データ) は、ET コマンドまたは CL コマンドを使用して格納し、OP コマンドまたは RE (ET データの読み込み) コマンドを使用して読み込むことができます。

ユーザー ID ごとにユーザーデータのレコードが1つ保持されます。データは、ユーザーデータが指定される次の ET コマンドまたは CL コマンドが発行されるまで保持されます。

各ユーザーデータレコードは、トランザクションで書き込まれたチェックポイントを使用して Adabas データプロテクションログにも書き込まれます。このデータは、後で ADASEL ユーティリティを使用して読み込むことができます。

ユーザーデータは、主に、次の機能を実行するのに使用します。

- オペレーションの再開に必要なデータを格納します (例えば、入力メッセージデータ、トランザクション識別データ、トランザクションサマリデータなど)。
- 後続のトランザクションで使用する中間データを格納します (例えば、監査結果を追跡するため)。
- 他のユーザーとコミュニケーションします。データを格納したユーザーのユーザー ID を OP コマンドに指定すると、他のユーザーがそのデータを読み込むことができます。

また、次の機能を実行するには、ユーザーデータ機能を使用することをお勧めします。

- ユーザーデータレコードのインストラクション標準を確立し、ET コマンドを使用して更新トランザクションごとにデータを格納します。
- ユーザーがアプリケーションにログオンするとき、ユーザーデータを読み込んで、標準メッセージ形式で表示します。ユーザーには、そのユーザー ID に対応する最後の正常更新処理が通知されます。

ユーザーデータを提供する ET コマンドと、提供しない ET コマンドを別々に扱うことをお勧めします。この2種類の ET コマンドを組み合わせると、(一時的に格納されている) Adabas WORK データセットからプロテクションログにユーザーデータが繰り返しコピーされるため、余分なオーバーヘッドが発生します。

Adabas ユーザー ID

ユーザー ID は、Adabas がユーザー (ET) データを格納および取得し、ユーザーに特別な優先権を割り当てるのに使用する ID です。ユーザー ID は OP コマンドのアディション1フィールドに指定します。

ユーザーが指定するユーザー ID は、次の条件の両方を満たす必要があります。

- 文字 A (X'C1') から 9 (X'F9') で始まる必要があります。

- この番号は、どの端末を使用しても、ユーザー（ET）データがそのユーザーに関連付けられるように、ユニークである必要があります。ユーザー（ET）データは、次にユーザー（ET）データを指定した ET コマンドまたは CL コマンドを発行するまで保持されます。

ユーザー ID を指定しなかった場合や、無効なユーザー ID を指定した場合は、Adabas がユーザーセッションのデフォルトのユーザー ID を設定し、ユーザーが格納したユーザー（ET）データを現在のセッションが終了したときに削除します。

後で制限を受けることがないように、常にユーザー ID を指定するようにしてください。

競合更新

競合更新は、2人以上のユーザー（マルチユーザーモード）が同一 Adabas ファイル（複数の場合もあり）を更新するときに発生します。

このセクションでは、競合更新環境でデータ保全性を確保するのに使用される Adabas 機能について説明します。具体的には、レコードのホールドと解放の処理、リソースのデッドロックの回避、および排他制御更新について説明します。

このsectionでは、次のトピックについて説明します。

- ET ユーザーレコードホールド
- リソースのデッドロックの回避
- 排他制御更新

ET ユーザーレコードホールド

レコードホールド機能を使用すると、ET ユーザーはレコードをホールド状態にして更新処理を実行し、その処理の間は他のユーザーが同じレコード更新できないようにすることができます。Adabas は、レコードの ISN をホールドキューに格納して、レコードをホールドします。このため、レコードホールドは ISN ホールドとも呼ばれています。

レコードホールドは、ET ロジックユーザーに適用されます。Adabas は EXU（排他制御更新）ユーザーに対しては ISN をホールド状態にしません。詳細については、「[排他制御更新](#)」を参照してください。

このsectionでは、次のトピックについて説明します。

- レコードホールドコマンド
- ホールドオプションを使用したレコード更新

■ レコード解放

レコードホールドコマンド

ホールド付き検索コマンド (S4)、ホールド付き読み込みコマンド (L4、L5、L6)、ホールドオプションを指定した A1 コマンドまたは E1 コマンド、あるいはレコードホールドコマンド (HI) を発行すると、レコードはホールド状態になります。また、ET ロジックユーザーが N1/N2 コマンドを発行すると、追加されたレコードはホールド状態になります。

これらのどのコマンドが正常に完了しても、レコード (ISN) はホールド状態になります。レコードがすでに他のユーザーによってホールドされている場合、レコードホールドコマンドを発行したユーザーは、レコードが使用可能になるまで待機状態になります。使用可能になった時点で、Adabas はコマンドを実行します。

レコードホールドコマンドに R (リターンコード) または O (マルチフェッチ/リターンコード) オプションを指定して発行すると、ホールドしようとするレコードが他のユーザーにホールドされている場合、Adabas はユーザーを待機状態にするのではなく、レスポンスコード 145 を返します。

検索 (S1) または 読み込み (L1、L2、L3) コマンドを発行した場合は、レコードが他のユーザーにホールドされていてもレコードにアクセスできます。

ホールドオプションを使用したレコード更新

ホールド状態にしたレコードを更新または削除するには、A1 コマンドまたは E1 コマンドを発行します。

A1 コマンドは、要求したユーザーに対してレコードがホールド状態にある場合、またはホールド状態にないレコードのホールドを指定した場合にのみ実行できます。レコードが現在他のユーザーにホールドされている場合、ホールド要求は待機状態になります。ただし、レスポンスコード 145 を受け取るように要求することもできます。レコードがホールド状態でない場合は、レスポンスコード 144 が返され、レコードがホールド状態であったことが示されます。

E1 コマンド (ホールドオプションなし) をホールド状態にないレコードに対して発行した場合、そのレコードが他のユーザーに対してホールド状態でない限り、レコードはホールド状態になります。削除するレコードをホールド状態にしないと、E1 コマンドを発行する前に、他のユーザーがそのレコードを更新または削除してしまう可能性があります。

N1/N2 コマンドを発行したときに、ホールドキューに使用可能なスペースがなかった場合、レスポンスコード 145 が返されます。

レコード解放

ET ロジックユーザーがレコードをホールド状態から解放するには、各論理トランザクションの後に ET コマンドを使用するか、Adabas セッションの終わりに クローズ (CL) コマンドを使用します。現在のトランザクションで更新を行った場合は、ET ロジックを使用するプログラムでレコード解放 (RI) コマンドを発行してレコードのホールドを解放しないでください。解放すると、データ安全性が失われることがあります。

例えば、ET ユーザー 1 がレコードを更新してから、RI コマンドでそのレコードを解放し、その後もトランザクションが続くとします。その間に、ユーザー 2 が同じレコードを更新し、ET コマンドでセッションを終了します。この場合、エラーやタイムアウトのためにユーザー 1 のトランザクションがバックアウトされると、ユーザー 2 のトランザクションが ET コマンドで正常に完了していても、両者 (ユーザー 1 と 2) の行った更新は削除されます。

このため、ET ロジックを使用するユーザープログラムでは、論理トランザクションの完了時に、ET コマンドでレコードを解放する必要があります。非 ET ロジックユーザーは、A1 コマンドまたは E1 コマンドでレコードの更新や削除を行い、レコードを解放する必要があります。

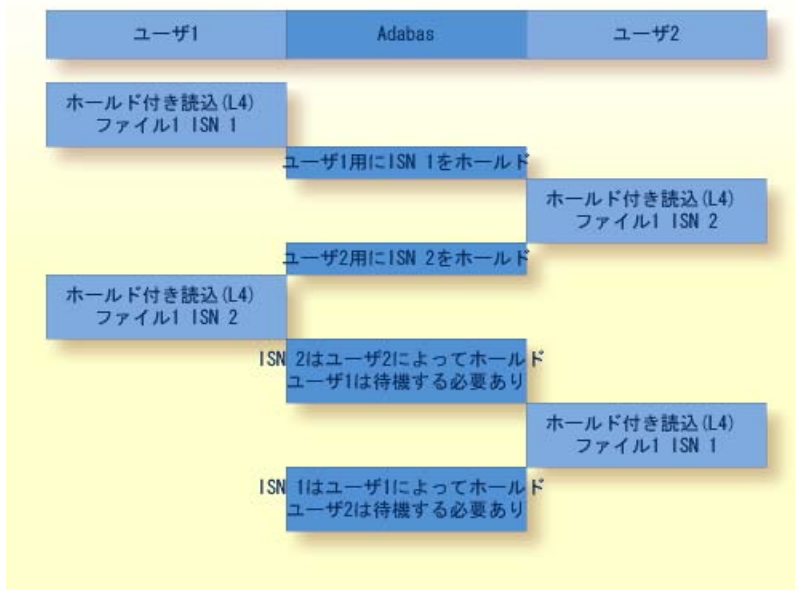
マルチフェッチオプションを使用すると、ET (または BT) コマンドの発行時に ISN バッファ内にゼロ以外のカウントとファイル番号/ISN を指定している場合に限り、対象となるレコード (およびその ISN) を選択してホールド状態から解放できます。詳細については、「[マルチフェッチ処理](#)」セクションを参照してください。

CL コマンドでは、このユーザーが ET ロジックを使用しているかどうかにかかわらず、このユーザーに対してホールド状態にあるレコードをすべて解放します。

リソースのデッドロックの回避

2人のユーザーが互いに相手のホールド状態にあるレコードを要求したために、両ユーザーとも待機状態になった場合には、リソースのデッドロックが発生します。

例



リソースデッドロックの例

Adabasは、ユーザー2を待機状態にする前にデッドロックの可能性を検出し、2番目のユーザーにレスポンスコード 145 を返します。

排他制御更新

ユーザーセッションの実行中に他のユーザーがファイルを更新できないように、1つ以上のAdabas ファイルに対する排他制御を要求できます。

排他制御を要求するには、OP コマンドを使用します。要求したファイルが他のユーザーまたはユーティリティにより更新モードでオープンされていない場合に限り、排他制御権が付与されます。要求したファイルがない場合、OP コマンドはレスポンスコード 48 を返します。

排他制御は、ファイルの競合更新を防ぐだけではありません。他のユーザーの動作に関係なくファイルをリストアできるため、リカバリ手順を単純化できます。

ユーザーが排他制御を持つファイルに対して、レコードホールドコマンドを使用する必要はありません（ただし、使用することは可能です）。Adabasは、排他ファイル制御で更新されるファイルに対しては、ホールドロジック処理を無効にします。

排他制御ユーザーは、非 ET ロジックユーザーとみなされるため、ET タイムアウト制限による制御を受けません。ただし、非 ET ロジックユーザーが ET コマンドを発行した場合、ユーザーは自動的に ET ロジックユーザーになり、何らかの理由でレコードをホールド状態にすると、トランザクションタイムアウト制限が課せられます。このようなユーザーは、終了するか、またはタイムアウトになるまで、排他制御を保持します。

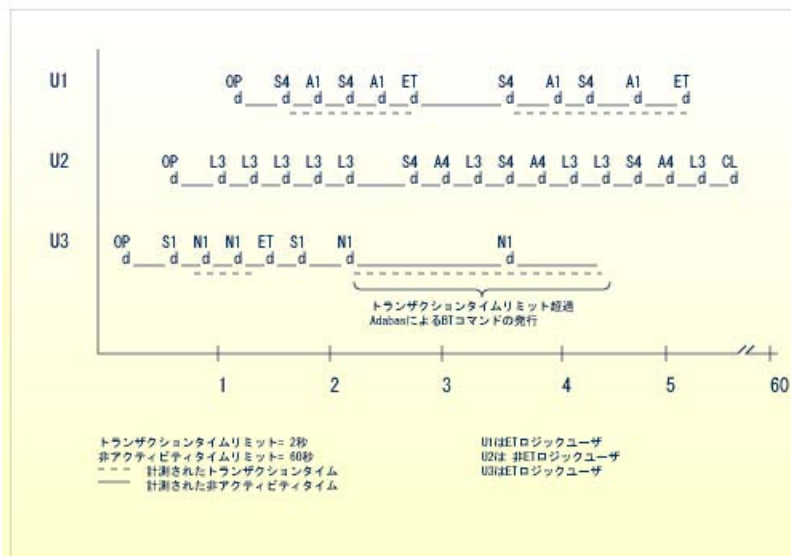
排他制御更新を行うユーザーは、C1 コマンドを使用して、チェックポイントの取得を要求できます。C1チェックポイントは、チェックポイント後に適用した更新を削除したり、チェックポイント前に適用した更新を再適用したりするための参照ポイントとなります。

非アクティビティタイムリミット

非アクティビティタイムリミットは、全ユーザー（アクセスオンリー、ETロジック、排他制御）に適用されます。ADARUN TNAA、TNAE、TNAX の各パラメータを使用すると、ユーザータイプごとに異なる非アクティビティタイムリミットを定義できます（詳細については、「Adabas の初期化 (ADARUN ステートメント)」を参照）。

また、ユーザーはOPコマンドで任意に非アクティビティタイムリミットを設定できます。最大値はADARUNMXTNAパラメータで制御します。詳細については、「[OP コマンド：ユーザーセッションのオープン](#)」を参照してください。

次の図は、ユーザーが非アクティビティタイムリミットを超えたときに Adabas が実行する処理をまとめたものです。



トランザクションと非アクティビティタイムリミット

このsectionでは、次のトピックでこれらの Adabas 処理について説明します。

- ET ロジックユーザー
- 排他ファイル制御ユーザー (EXF)

■ アクセスオンリーユーザー

ET ロジックユーザー

ET ロジックユーザーの場合、Adabas は次の処理を実行します。

- (必要に応じて) 現在のユーザートランザクションに対して、トランザクションのバックアウト (BT) コマンドを発行します。
- トランザクション中にホールドされたレコードをすべて解放します。
- UQE 内のユーザーファイルリストを削除します。
- このユーザーに対するコマンド ID をすべて削除します。

次の状況のいずれかが発生した場合、Adabas は次にコマンドが発行されたときにレスポンスコード 9 をユーザーに返します。

- タイムアウトが発生したとき、ユーザーは ET 状態ではありませんでした。
- タイムアウトが発生したとき、ユーザーは ET 状態でしたが、OP コマンドに空白以外のユーザー ID を指定していました。

タイムアウトが発生したときにユーザーが ET 状態であった場合、このユーザーが OP コマンドに空白以外のユーザー ID を指定していないと、UQE も削除されます。

排他ファイル制御ユーザー (EXF)

排他ファイル制御ユーザーの場合、Adabas は UQE 内のユーザーのファイルリストを削除します。これにより、ユーザーは排他制御を行っていたファイル (複数可) に対する排他制御権を失います。

また、ユーザーに対するコマンド ID はすべて解放され、ユーザーはアクセスオンリーユーザーになります。

アクセスオンリーユーザー

アクセスオンリーユーザーの場合、Adabas は UQE 内のユーザーのファイルリストを削除します。

5 プログラミング全般に関する考慮事項

■ コマンド ID、フォーマット ID、グローバルフォーマット ID	30
■ ISN リスト処理	37
■ マルチフェッチ／プリフェッチ機能の使用	42

このセクションでは、コールをプログラミングするときに考慮する必要のある概念について説明します。

内部 ID を指定すると、Adabas コマンドの実行時に重要な機能を実行できます。Adabas ダイレクトコールコマンドのコントロールブロックで、次の ID を指定できます。

- コマンド ID (ACBCID または ACBXCID) 。これは、ACB または ACBX に指定される空白でもゼロでもない値で、コマンド処理の内部 ID として機能します。コマンド ID を使用すると、同じフォーマットバッファを使用する一連のコマンドで解釈および変換が繰り返されるのを排除できます。
- フォーマット ID。これは、デコード結果のフォーマットバッファに固有の4バイトの内部 ID で、ユーザー固有のものにすることも、同じ Adabas ニュークリアスで実行している他のユーザーもグローバルに使用可能なもの (グローバルフォーマット ID) にすることもできます。

フォーマット ID の使用については、このセクションで詳しく説明します。

このセクションでは、これ以外に、Adabas ワークから ISN を取得するための手順、マルチフェッチオプション、およびプリフェッチオプションについても説明します。マルチフェッチおよびプリフェッチオプションを使用すると、Adabas コールの完了に必要なシステムコマンドの数が減るため、大量のデータを順次処理するプログラムの実行時間を短縮できます。

このchapterでは、次のトピックについて説明します。

コマンド ID、フォーマット ID、グローバルフォーマット ID

Adabas コントロールブロック (ACBCID または ACBXCID) に指定されているコマンド ID は、Adabas コマンド実行中に重要な機能を果たします。コマンド ID は、自動生成されるか、またはユーザーが指定する空白でもゼロでもない値であり、次の機能を実行します。

- デコード結果のレコードフォーマットの内部 ID として機能し、フォーマットバッファのデコーディングの繰り返しを防ぎます。
- 後でアクセスできるように、Sx コマンドが生成した ISN リストにタグを付け、ISN リストのオーバーフローを防ぎます。
- レコードセットの順序読み込み処理にタグを付けて識別できるようにします。

必要に応じて、デコード結果のフォーマットバッファに固有の内部フォーマット ID を指定できます。この値は、アディション5フィールドの先頭バイトの上位 (左端) 2ビットにあるフラグで識別されます。フォーマット ID は、フラグの値に応じて、ユーザー固有のもの (個別のフォーマット ID) にすることも、同じ Adabas ニュークリアスで実行している他のユーザーも使用可能なもの (グローバルフォーマット ID) にすることもできます。

このsectionでは、次のトピックについて説明します。

- **コマンド ID、フォーマット ID、グローバルフォーマット ID の指定**

- 読み込みシーケンシャルコマンドで使用するコマンド ID
- 読み込み、更新および検索コマンドで使用するコマンド ID およびフォーマット ID
- コマンド ID とフォーマット ID に別々の ID 値を使用
- グローバルフォーマット ID の使用
- ISN リストで使用するコマンド ID
- コマンド ID の自動生成
- コマンド ID の解放
- コマンド ID の内部識別
- コマンド ID の使用例

コマンド ID、フォーマット ID、グローバルフォーマット ID の指定

次の表は、コマンド ID、フォーマット ID、およびグローバルフォーマット ID を指定するのに必要な Adabas コントロールブロック（ACB または ACBX）設定をまとめたものです。

ID	ACB または ACBX フィールド指定	
	コマンド ID (ACBCID または ACBXCID)	アディション 5 (ACBADD5 または ACBXADD5)
コマンド ID	このフィールドには空白でもゼロでもない値を指定してください。 X'FFFFFFFF' のコマンド ID 値を指定すると、X'00000001' から始まって 1 ずつ増加するコマンド ID が Adabas によって自動生成されます。	バイナリの "00" に設定される上位（左端）ビット
フォーマット ID	このフィールドには空白でもゼロでもない値を指定してください。 X'FFFFFFFF' のコマンド ID 値を指定すると、X'00000001' から始まって 1 ずつ増加するコマンド ID が Adabas によって自動生成されます。	先頭バイトの上位（左端）2 ビットをバイナリの "10" に設定します。5～8 番目のバイトをフォーマット ID に設定します。フォーマット ID を X'FE' および X'FF' で始めることはできません。 非メインフレーム Adabas 環境では、アディション 5 フィールドの先頭バイトを任意の小文字に設定します。
グローバルフォーマット ID	このフィールドには空白でもゼロでもない値を指定してください。 X'FFFFFFFF' のコマンド ID 値を指定すると、X'00000001' から始まって 1 ずつ増加するコマンド ID が Adabas によって自動生成されます。	先頭バイトの上位（左端）2 ビットをバイナリの "11" に設定します。残りのバイトをグローバルフォーマット ID に設定します。アディション 5 フィールドの 8 バイトはすべて、グローバルフォーマット ID として使用されます。 メインフレーム環境では、16 進の E2～E9（文字 S～Z）の範囲に先頭バイトを割り当てることができます。それ以外の範囲は、Software AG によって使用されるために予約されています。 非メインフレーム Adabas 環境では、16 進の 51～5A（文字 S～Z）の範囲に先頭バイトを割り当てることができます。



Notes:

1. アディクション5フィールドの先頭バイトの上位（左端）2ビットに B'00' を設定すると、コマンド ID は自動的にフォーマット ID として使用されます。
2. アディクション5フィールドの先頭バイトの上位（左端）2ビットに B'10' を設定すると、コマンド ID とフォーマット ID に別々の値が使用され、アディクション5フィールドの5~8バイトがフォーマット ID として使用されます。
3. アディクション5フィールドの先頭バイトの上位（左端）2ビットに B'11' を設定すると、コマンド ID とフォーマット ID に別々の値が使用され、フォーマット ID はグローバルフォーマット ID となり、アディクション5フィールドの8バイトはすべてグローバルフォーマット ID として使用されます。
4. Adabas はグローバルフォーマット ID の割り当てを確認しません。正当な範囲のグローバルフォーマット ID のみが割り当てられるようにするのは、ユーザーの責任です。Software AG は、ユーザーに有効なグローバルフォーマット ID の範囲を拡張することも、グローバルフォーマット ID 割り当ての問題を解決するためにその製品に変更を行うこともできません。

読み込みシーケンシャルコマンドで使用するコマンド ID

読み込みシーケンシャルコマンド（L2/L3、L5/L6、L9）には、コマンド ID を指定する必要があります。Adabas が正しい順序でユーザーにレコードを返すには、コマンド ID が必要です。これらのコマンド ID は、シーケンシャルコマンドのテーブルに保持されます。

別のフォーマット ID を使用しない限り（「[コマンド ID とフォーマット ID に別々の ID 値を使用](#)」セクションを参照）、これらのコマンドで使用するコマンド ID 値も、内部フォーマットバッファプールに入力され、保持されます。順次読み込み処理中にエンドオブファイル条件が検出されると、コマンド ID は解放されます。

読み込み、更新および検索コマンドで使用するコマンド ID およびフォーマット ID

読み込みコマンド（L1~L6、L9）および更新コマンド（A1/A4、N1/N2）では、読み込むフィールドまたは更新するフィールドを指定するフォーマットバッファが必要です。このフォーマットバッファは、Adabas によって解釈され、内部フォーマットバッファに変換されます。有効なコマンド ID を使用すると、同じフォーマットバッファを使用する一連のコマンドで解釈と変換が繰り返されるのを防ぐことができます。

読み込みコマンドまたは更新コマンドに有効なコマンド ID を指定すると、Adabas はそのコマンド ID が内部フォーマットバッファプールにあるかどうかチェックします。コマンド ID がある場合、そのコマンド ID に対応する内部フォーマットバッファが使用されるため、フォーマットバッファを再度解釈する必要がなくなります。



Note: 同じフォーマットバッファを使用する一連のレコードを読み込んだり、更新したりする場合、コマンド ID を使用すると、読み込みコマンドや更新コマンドの処理時間を大幅に短縮できます。

L9 コマンドから生じた内部フォーマットバッファ（およびフォーマット ID）は、別の L9 コマンドでのみ使用できます。また、L9 コマンドでは、L9 以外の内部フォーマットバッファ/フォーマット ID を使用することはできません。このことは、グローバルフォーマットバッファ（およびグローバルフォーマット ID）と L9 コマンドについても言えます。

同じフィールドを読み込んで更新する場合（例えば、L5 の後に A1 が続く場合）、両コマンドに同じコマンド ID を使用することをお勧めします（読み込みと更新に同じフォーマットバッファを使用する場合の制限については、A1/A4 コマンドと N1/N2 コマンドの説明を参照）。

S1/S2/S4 コマンドで第 1 検索レコードを読み込むオプションを使用し、かつコマンド ID を指定すると、コマンド ID とそれに対応する内部フォーマットバッファも内部フォーマットバッファプールに格納されます。

内部フォーマットバッファプールが一杯になり、プールに存在しないコマンド ID のコマンドを受け取ると、Adabas はプール内で最も長く使用されていないエントリを新たに解釈されたフォーマット ID に置き換えます。その後、削除したコマンド ID を使用するコマンドを受け取ると、そのコマンド ID 用に再度解釈されたフォーマットバッファが、プール内で次に長く使用されていないエントリと置き換えられます。このため、フォーマットバッファは、同じコマンド ID を使用する一連の読み込みコマンドまたは更新コマンド間で変更しないでください。なお、コマンド ID を使用しても、フォーマットバッファが再解釈されないことは保証されません。

コマンド ID とフォーマット ID に別々の ID 値を使用

コマンド ID とフォーマット ID に別々の値を使用できます。アディクション 5 フィールドの上位 2 ビットにバイナリの 00 を設定している限り、コマンド ID は自動的にフォーマット ID として使用されます。しかし、アディクション 5 フィールドの上位 2 ビットがバイナリの 10 であると、フィールドの 5~8 バイト（アディクション 5+4(4)）はフォーマット ID として使用されます。ID を X'FE' および X'FF' で始めることはできません。



Note: コマンド ID とは異なるフォーマット ID を識別するために、非メインフレーム Adabas 環境では、アディクション 5 フィールドの先頭バイトは小文字であるとみなされます。異種環境で別のフォーマット ID を使用するときは、システムで使用している全プラットフォームでもその ID を識別できることが重要です。


グローバルフォーマット ID の使用

オンライン環境で特にみられることですが、複数のユーザーが同じプログラムを使用して 1 つのファイルの同じフィールドを読み込み、更新し、フォーマットバッファも同じものを使用するということがよくあります。

- 個別のフォーマット ID オプションを使用すると、Adabas はユーザーごとに同じ内部フォーマットバッファを格納します。
- グローバルフォーマット ID オプションを使用すると、多くのユーザーが 1 つの内部フォーマットバッファを共有でき、Adabas が内部フォーマットバッファプールのエントリを書き換える必要がなくなります。このオプションにより、各ユーザーの指定するフォーマットバッ


ファは（フォーマット ID と端末 ID で識別するのではなく）フォーマット ID でのみ識別されます。コマンド ID をグローバルフォーマット ID として指定することはできません。また、L9 フォーマットとその ID は他の L9 コマンドでしか使用できないという制限は、グローバルフォーマットや ID にも適用されます。

グローバルフォーマット ID オプションを有効にするには、アディクション 5 フィールドの先頭バイトの上位 2 ビットにバイナリの 11 を設定します（「[コマンド ID、フォーマット ID、およびグローバルフォーマット ID の指定](#)」を参照）。これにより、アディクション 5 の 8 バイトはすべて、グローバルフォーマット ID として認識されます。

 **Note:** フォーマット ID をグローバルとして識別するために、非メインフレーム Adabas 環境では、アディクション 5 フィールドの先頭バイトは任意の数字か小文字であるとみなされます。異種環境でグローバルフォーマット ID を使用するときは、システムで使用しているどのプラットフォームでもその ID を識別できることが重要です。

グローバルフォーマット ID の先頭バイトは、16 進の E2～E9（文字 S～Z）の範囲に割り当てることができます。それ以外の範囲は、Software AG によって使用するために予約されています。

非メインフレーム Adabas 環境でグローバルフォーマット ID に許可されている値の範囲は、16 進の 51～5A（文字 S～Z）です。

 **Caution:** Adabas はグローバルフォーマット ID の割り当てを確認しません。正当な範囲のグローバルフォーマット ID のみが割り当てられるようにするのは、ユーザーの責任です。Software AG は、ユーザーに有効なグローバルフォーマット ID の範囲を拡張することも、グローバルフォーマット ID 割り当ての問題を解決するためにその製品に変更を行うこともできません。

グローバルフォーマット ID を削除するには、RC コマンドを使用し、アディクション 5 フィールドにそのグローバルフォーマット ID を指定します。

ISN リストで使用するコマンド ID

ISN リストを返すコマンド（S1、S2、S4、S5、S8、S9）にコマンド ID を指定した場合、後でそのコマンド ID 値を使用してリストを識別できます。

- S_x コマンドで SAVE ISN LIST オプションを使用する場合は、コマンド ID を指定する必要があります。SAVE ISN LIST オプションにより、ISN リストがすべて Adabas ワークに格納されます。リスト内の ISN は、後で S_x コマンドまたは L1/L4 コマンドの GET NEXT オプションを使用して読み取ることができます。
- SAVE ISN LIST オプションを使用せず、ISN バッファオーバーフローが発生した（ISN リスト全体が ISN バッファに収まらない）場合、コマンド ID 値を使用したときに限り、オーバーフローした ISN は Adabas ワークに格納されます。この場合、コマンド ID とそれに対応する ISN リストは、ISN をすべてユーザーに返した時点で解放されます。

コマンド ID の自動生成

コマンド ID 値に X'FFFFFFFF' を指定すると、コマンド ID を自動的に生成できます。これにより、X'00000001' から始まって 1 ずつ増加するコマンド ID が Adabas ニュークリアスによって自動生成されます。ただし、コマンド ID の自動生成が望ましくないこともあります。詳細については、「[読み込み、更新および検索コマンドで使用するコマンド/フォーマット ID](#)」セクション参照してください。

コマンド ID の解放

コマンド ID とその関連エントリ (ISN リスト) を解放するには、RC コマンドまたは CL コマンドを使用します。この他、Sx コマンド (S1、S2、S4、S5、S8、S9) の CID 解放オプションで解放することもできます。

RC コマンドには、内部フォーマットバッファプール、シーケンシャルコマンドテーブル、または ISN リストテーブルにあるコマンド ID のみを解放するオプションがあります。

CL コマンドでは、そのユーザーに対して現在アクティブであるコマンド ID をすべて解放します。

Sx コマンドに CID 解放オプションを指定すると、まずコマンド ID を解放してからコマンド処理を実行します。

コマンド ID の内部識別

各コマンド ID エントリは、コマンド ID 値と内部ユーザー ID で識別されます。このため、他のユーザーが使用中のコマンド ID 値を意識する必要はありません。ただし、同じコマンド ID 値を異なるコマンドで使用するのは避けてください。特に、L2/L5、L3/L6、L9 の各読み込みシーケンシャルコマンドと Sx コマンドでは同じコマンド ID を使用しないことをお勧めします。

コマンド ID の使用例

この section では、次のトピックについて説明します。

- [例 1：検索/読み込み処理](#)
- [例 2：GET NEXT オプションを使用した検索/読み込み処理](#)
- [例 3：読み込み/更新処理](#)

■ 例 4：読み込み／検索処理

例 1：検索／読み込み処理

レコードセットを選択し、読み込みます。読み込むレコードごとに同じフォーマットバッファを使用します。

```
FIND (S1)          CID=EX1A
READ (L1)          CID=EX1B
READ (L1)          CID=EX1B
```

例 2：GET NEXT オプションを使用した検索／読み込み処理

L1/L4 コマンドの GET NEXT オプションを使用してレコードセットを選択し、読み込みます。

```
FIND (S1)          CID=EX2A
READ (L1)          CID=EX2A
READ (L1)          CID=EX2A
READ (L1)          CID=EX2A
```

例 3：読み込み／更新処理

ファイルを順次読み込んで、更新します。読み込みと更新で同じフォーマットバッファを使用します。

```
READ PHYS SEQ (L5) CID=EX3A
UPDATE (A4)         CID=EX3A
READ PHYS SEQ (L5) CID=EX3A
UPDATE (A4)         CID=EX3A
```

例 4：読み込み／検索処理

ファイルを論理順に読み込みます。第 1 ファイルから読み込んだフィールドの値を使用して、第 2 ファイルに検索コマンドを発行します。次に、GET NEXT オプションを使用して、検索コマンドで取得したレコードを読み込みます。

```
READ LOG SEQ (L3)  CID=EX4A
FIND (S1)          CID=EX4B
READ (L1)          CID=EX4B
READ (L1)          CID=EX4B
READ LOG SEQ (L3)  CID=EX4A
FIND (S1)          CID=EX4B
READ (L1)          CID=EX4B
```

ISN リスト処理

このセクションでは、Adabas WORK データセットから ISN を読み込む手順について説明します。L1/L4 コマンドの GET NEXT オプションを使用して、ISN リスト中の ISN に対応するレコードを読み込む場合は、このセクションで説明する ISN 処理を Adabas が自動的に行うため、ユーザーがその手順を使用する必要はありません。

このセクションで使用する S_x コマンドとは、ISN リストを生成するコマンド (S1/S2/S4/S5/S8/S9) のことです。

このsectionでは、次のトピックについて説明します。

- ISN リストの格納
- ISN リストの取得
- ISN リスト処理例

ISN リストの格納

Adabas は次のいずれかの条件があてはまる場合、ISN を WORK データセットに格納します。

- S_x コマンドが発行され、空白／ゼロ以外のコマンド ID が指定され、SAVE ISN LIST オプションが指定されています。この場合、ISN 結果リスト全体が格納されます。
- S_x コマンドが発行され、空白／ゼロ以外のコマンド ID が指定され、SAVE ISN LIST オプションが指定されず、ISN 結果リストが ISN バッファの許容量を超えています。この場合、あふれた ISN のみが格納されます。

コマンド ID フィールドに空白かバイナリの 0 を指定して S_x コマンドを発行すると、Adabas は ISN を Adabas ワークに格納しません。

ISN リストの取得

Adabas WORK データセットに格納されている ISN を取得するには、最初の S_x コマンドに使用したのと同じコマンド ID 値を指定して S_x コマンドを発行します。アクティブなコマンド ID 値を指定して S_x コマンドを発行すると、Adabas は既存の ISN リストに ISN を要求しているものとみなします。Adabas は指定のコマンド ID で識別される ISN リストを探し出し、次の ISN グループを ISN バッファに入れます。ISN バッファに入るだけの ISN が返されます。

このsectionでは、次のトピックについて説明します。

- SAVE ISN LIST オプションを指定した場合
- SAVE ISN LIST オプションを指定しなかった場合

■ コントロールブロックの ISN 数フィールドの使用

SAVE ISN LIST オプションを指定した場合

ISN リストの作成に使用する Sx コマンドで SAVE ISN LIST オプションを指定した場合、Adabas は ISN 下限フィールドに指定された ISN を使用して次に返す ISN グループを判断します。

ISN 下限フィールドに指定された ISN より次に大きい ISN で始まるものが、次のグループになります。

- ISN 下限フィールドにバイナリの 0 を指定した場合、リスト内の先頭 ISN で始まるものが、次のグループになります。
- リストのどの ISN よりも大きい値を指定すると、レスポンスコード 25 が返されます。

S2 コマンドを使用して ISN リストを作成した場合、指定する ISN は ISN リストに存在している必要があります。このように SAVE ISN LIST オプションを使用すると、ユーザーは ISN リスト内を前後にスキップできます。このことは、画面を前後にスクロールする操作をプログラムで実現するのに役立ちます。

SAVE ISN LIST オプションを指定しなかった場合

ISN リストの作成に使用した Sx コマンドに SAVE ISN LIST オプションを指定しなかった場合、Adabas はリスト内の配置順で ISN を返します。Adabas ワークからの各 ISN グループをユーザーの ISN バッファに入力し終わると、Adabas はこの ISN リストを削除します。リストの識別に使用されたコマンド ID も、ISN の最後のグループがユーザーに返されたときに解放されます。この場合、指定した ISN 範囲を超えて処理が始まらない限り、ISN 下限フィールドは使用されません。

コントロールブロックの ISN 数フィールドの使用

コントロールブロック内の ISN 数フィールドを使用すると、ユーザーはリスト内の ISN の取得が完了した時点判断できます。

- 最初の Sx コマンドでは、検索条件を満たすレコードの総数をこのフィールドに返します。
- また、Adabas ワークから ISN を取得するのに使用する後続の各 Sx コマンドでは、このフィールドを使用して、ISN バッファに挿入された ISN の番号を格納します。

ISN リスト処理例

このsectionでは、次のトピックについて説明します。

- 例 1：GET NEXT オプションを指定した L1/L4 コマンドと S_xコマンドの併用
- 例 2：SAVE ISN LIST オプションの使用
- 例 3：ISN オーバーフローの取り扱い
- 例 4：ISN オーバーフロー取り扱いなし

例 1：GET NEXT オプションを指定した L1/L4 コマンドと S_xコマンドの併用

```
Sx command
L1/L4 command with 'GET NEXT' option
L1/L4 command with 'GET NEXT' option
L1/L4 command with 'GET NEXT' option
```

次の各例では、ISN バッファのサイズによって、さまざまな結果が得られることを示しています。ISN リストの配置は ISN 下限で定義します。

```
1. ISN Buffer Length = 0
   read the ISN list to the work area
```

```
2. L1/L4 with GET NEXT
   result: first ISN's record
```

```
1. ISN Buffer Length = 4
   read first ISN with S1
```

```
2. L1/L4 with GET NEXT
   result: second ISN's record
```

```
1. ISN Buffer Length = 12
   read first ISN with S1
   first 3 ISNs are returned in ISN buffer
```

```
2. L1/L4 with GET NEXT
   result: read fourth/fifth/sixth ISNs' records
```


例 2 : SAVE ISN LIST オプションの使用

SAVE ISN LIST オプションを使用した最初の Sx コール

```
Command = Sx
Command ID = SX01          (save-ISN-list option)
Command Option 1 = H
ISN Lower Limit = 0
ISN Buffer Length = 20
CALL ADABAS ...
```

実行結果の ISN 数 = 7 (格納リスト内の ISN の総数)

ISN 結果リスト : (ISN はすべてワークに格納されます)

```
8   12  14  15  24  31  33
```

実行結果の ISN バッファ :

```
8   12  14  15  24
```

後続の Sx コール

```
Command = Sx
Command ID = SX01
ISN Lower Limit = 24      (limit ISN choice to 24, +)
ISN Buffer Length = 20    (space for 5 ISNs from ISN list)
CALL ADABAS ...
```

実行結果の ISN 数 = 2 (ISN バッファに返された ISN の総数)

実行結果の ISN バッファ :

```
31  33  14  15  24.... remainder of ISN buffer unchanged....
```

後続の Sx コール

```
Command = Sx
Command ID = SX01
ISN Lower Limit = 0
ISN Buffer Length = 20
CALL ADABAS ...
```

実行結果の ISN 数 = 7

実行結果の ISN バッファ :

```
8   12  14  15  24
```


例 3 : ISN オーバーフローの取り扱い

最初の S_x コール (SAVE ISN LIST オプションは使用しない)

```
Command = Sx
Command ID = SX02
Command Option 1 = blank      (no option)
ISN Lower Limit = 0
ISN Buffer Length = 20
CALL ADABAS ...
```

実行結果の ISN 数 = 7 (ISN バッファに返された ISN とワークに格納されている ISN の総数)

ISN 結果リスト : (ISN 31 と 33 のみがワークに格納されています)

```
8   12  14  15  24  31  33
```

実行結果の ISN バッファ :

```
8   12  14  15  24
```

後続の S_x コール

```
Command = Sx
Command ID = SX02
ISN Lower Limit      (not used)
ISN Buffer Length = 20
CALL ADABAS ...
```

実行結果の ISN 数 = 2

実行結果の ISN バッファ :

```
31  33  14  15  24
```

31 と 33 の ISN を Adabas ワークから削除し、コマンド ID SX02 を解放します。最後の ISN がユーザーに返された後で SX02 が解放されたため、コマンド ID を SX02 とした後続の S_x コールは最初の S_x コールとして処理されます。

上記の例では ISN 下限を使用していませんが、ゼロ以外の値を指定すると、その値より大きい ISN のみを ISN バッファに返します (例 2 と同様)。

例 4 : ISN オーバーフロー取り扱いなし

コマンド ID を空白かゼロとした最初の S_x コール

```
Command = Sx
Command ID = blanks or binary zeros
Command Option 1 = blank      (no option)
ISN Lower Limit = 0          (no lower limit specified)
ISN Buffer Length = 20
CALL ADABAS ...
```

実行結果の ISN 数 = 7 (一致した ISN の総数)

ISN 結果リスト : ワークには何も格納されません。

```
8 12 14 15 24 31 33
```

実行結果の ISN バッファ :

```
8 12 14 15 24
```

後続の S_x コールでコマンド ID を空白かバイナリの 0、ISN 下限を 0 とすると、イニシャルコール時と同じ検索コマンドが再実行され、同じ結果が得られます。後続のコールでコマンド ID を空白かバイナリの 0、ISN 下限を 24 とすると、S_x コマンドが再実行されます。その結果、ISN 数=2 となり、ISN バッファには 31 と 33 の ISN が入ります。

マルチフェッチ／プリフェッチ機能の使用

大量のデータを順次処理するプログラムは、頻繁にストレージにアクセスする必要があり、実行時間が長くなります。Adabas マルチフェッチオプションとプリフェッチオプションは、Adabas コールの実行に必要なシステムコマンド数を減らして、このようなプログラムの実行時間を大幅に短縮します。

マルチフェッチオプションとプリフェッチオプションを使用すると、ほぼすべての標準アプリケーションで実行時間が短縮されます。ただし、具体的に得られるメリットはアプリケーションプログラムのタイプによって異なります。



Note: Adabas 8 の ACBX インターフェイスコールでは、マルチフェッチ機能はサポートされますが、プリフェッチ機能はサポートされません。ただし、**ACB インターフェイスダイレクトコール**用のプリフェッチ機能はサポートされています。したがって、アプリケーションに **ACB インターフェイスダイレクトコール**を使用している場合、これらのコールのみにプリフェッチ機能は、引き続き使用することができます。

このsectionでは、次のトピックについて説明します。

- [マルチフェッチとプリフェッチの比較](#)
- [マルチフェッチ／プリフェッチの起動](#)

- マルチフェッチ処理
- プリフェッチ処理
- プリフェッチまたはマルチフェッチしたレコードの ISN を変更した場合の影響

マルチフェッチとプリフェッチの比較

マルチフェッチ機能とプリフェッチ機能は、アプリケーションプログラムと Adabas ニュークリアス間のコミュニケーションオーバーヘッドを削減します。マルチフェッチ処理は、単一のコールで取得した複数のレコードを格納し、そのレコードをレコードバッファのユーザーに転送します。マルチフェッチを使用しないと、これと同じ結果を得るには複数の Adabas コールが必要になります。**ACB インターフェイスダイレクトコール**を実行すると、マルチフェッチしたレコードのレコードディスクリプタエレメント (RDE) が **ISN バッファ**に格納されます。**ACBX インターフェイスダイレクトコール**を実行すると、マルチフェッチしたレコードのレコードディスクリプタエレメント (RDE) が **マルチフェッチバッファ**に格納されます。

マルチフェッチ処理はプリフェッチに似ています。実際、マルチフェッチはプリフェッチ機能とその他の機能で構成されています。Adabas 8 以前のリリースではプリフェッチ処理とマルチフェッチ処理のどちらもサポートしていますが、新規に作成するプログラムでは、どの Adabas プラットフォームでも機能が共通しているマルチフェッチ (M) オプションを使用することをお勧めします。また、プリフェッチオプションは Adabas 8 以降のリリースではサポートされません。

マルチフェッチ／プリフェッチの起動

作成したアプリケーションで **ACB インターフェイスダイレクトコール**のみを使用している場合、プリフェッチ処理またはマルチフェッチ処理を呼び出すには2つの方法があります。どちらの方法で呼び出すかは、処理するために先読みしたレコードが保管されている場所、およびどんなバッファスペースを割り当てる必要があるかによって決まります。

- 1つ目の方法は、ADARUN ステートメントに PREFETCH=YES (マルチフェッチ処理の場合) または OLD (プリフェッチ処理の場合) パラメータを指定します。これは、最も効率のよい方法であり、アプリケーションプログラムを変更する必要がありません。これらのパラメータでは、バッチジョブを制御できます。

PREFETCH=YES または OLD を指定すると、Adabas はダブルバッファリング技法により、1つのレコードグループの処理中に次のレコードグループがフェッチされるようになります。

PREFETCH、PREFICMD、PREFIFIL、PREFNREC、PREFSBL、PREFTBL、PREFXCMD、PREFXFIL などプリフェッチ／マルチフェッチに関する ADARUN パラメータの詳細については、「Adabas の初期化 (ADARUN ステートメント)」を参照してください。

- もう1つの方法は、L1/L4、L2/L5、L3/L6、L9、BT または ET コマンドに M または O オプション (マルチフェッチ処理の場合) を指定するか、L1/L4、L2/L5、L3/L6 または L9 コマンドに P オプション (プリフェッチ処理の場合) を指定します。コマンドレベルオプション M、O、および P の使用については、各コマンドのドキュメントの他、「**マルチフェッチ処理**」と「**プリフェッチ処理**」を参照してください。

作成したアプリケーションで **ACBX インターフェイスダイレクトコール**を使用する場合、または **ACB インターフェイスダイレクトコール**と **ACBX インターフェイスダイレクトコール**を混在させる場合、マルチフェッチ処理を呼び出すには、L1/L4、L2/L5、L3/L6、L9、BT、または ET コマンドで M または O オプションを指定する必要があります。コマンドレベルオプション M および O の使用については、各コマンドのドキュメントの他、「**マルチフェッチ処理**」を参照してください。

 **Note:** プリフェッチ処理は、**ACBX インターフェイスダイレクトコール**を使用するアプリケーションではサポートされません。

マルチフェッチ処理

マルチフェッチ処理は、非メインフレームプラットフォーム上の対応する処理と互換性があり、異機種環境の各種プラットフォームで使用できます。

マルチフェッチ処理は、次の Adabas コマンドで使用できます。

- I または N オプションを指定した L1/L4 (ISN での読み込み、GET NEXT を指定した検索)
- L2/L5 (物理的な読み込み)
- L3/L6 (ディスクリプタ順に論理的な読み込み)
- L9 (ヒストグラム)
- BT (トランザクションのバックアウト)
- ET (トランザクションの終了)

マルチフェッチは、どの読み出しコール (L_x) に指定した場合でも、ひとまとまりのレコードをレコードバッファに返し、これらのレコードの記述をコール元の ISN バッファ (**ACB インターフェイスダイレクトコール**の場合) またはコール元のマルチフェッチバッファ (**ACBX インターフェイスダイレクトコール**の場合) に返します。レコードの最大数は次の値で制限されます。これらの値は、Adabas コントロールブロック (ACB または ACBX) のいずれか、または ABD に適宜指定されます。

- コールへの入力としてユーザーが定義した最大値
- レコードバッファ長
- ISN バッファ長 (**ACB インターフェイスダイレクトコール**)
- マルチフェッチバッファ長 (**ACBX インターフェイスダイレクトコール**)

この section では、次のトピックについて説明します。

- **読み込み (L_x) マルチフェッチ処理**

■ BT/ET のマルチフェッチ処理

読み込み (Lx) マルチフェッチ処理

読み込みコマンドのダイレクトコールを行うときにマルチフェッチ処理を使用する場合は、コールの前に次のようにフィールドを設定しておく必要があります。

設定内容	設定対象		
	ACB インターフェイスダイレクトコール	ACBX インターフェイスダイレクトコール	
	ACB	ACBX	ABD
サポートされるコマンドタイプおよびオプション (「マルチフェッチ処理」セクションのコマンドリストを参照)	コマンドコード (ACBCMD)	コマンドコード (ACBXCMD)	---
返される値の最大数。すべての値をマルチフェッチする場合は 0	ISN 下限 (ACBISL)	ISN 下限 (ACBXISL)	---
"M" または 0 に設定 (次の注を参照)	コマンドオプション 1 (ACBCOP1)	コマンドオプション 1 (ACBXCOP1)	---
レコードバッファの長さ	レコードバッファ長 (ACBRBL)	---	レコードバッファ ABD のバッファサイズ (ABDXSIZE)
ISN バッファの長さ	ISN バッファ長 (ACBIBL)	---	---
マルチフェッチバッファの長さ	---	---	マルチフェッチバッファ ABD のバッファサイズ (ABDXSIZE)



Notes:

1. コマンドオプション M は、マルチフェッチオプションを使用することを示します。コマンドオプション O は、L4/L5/L6 コマンドの場合、マルチフェッチオプション (M) と既存のコマンドオプション R (すでに別のユーザーがホールドしている ISN を要求するとレスポンスコード 145 が返される) の両方を選択します。
2. L1 コマンドの場合は、マルチフェッチコマンドオプション M または O とともに、コマンドオプション I (ISN 順) または N (GETNEXT オプション) のどちらかを指定する必要があります。指定しないと、レスポンスコード 22 が発生します。

返されるレコードバッファと、ISN バッファまたはマルチフェッチバッファの内容は、次のとおりです。

Record Buffer: *record1, record2, ... , recordn*

レコードは通常どおりにレコードバッファに戻されます。複数のレコードが返された場合、そのいずれのレコードもレコードバッファ内で隣接しています。

これらのレコードの記述は ISN バッファに戻されます。ISN バッファの最初のフルワードは、後続の要素数を示します（符号付き整数、4 バイト）。このカウントの後に、16 バイトのレコードディスクリプタ要素が続きます。

ISN or Multifetch Buffer: *RDE count*{*RDE1* }...

レコードディスクリプタ要素（RDE）の構造を次の表に示します。

フォーマット	長さ	内容
すべてのフィールドは符号なし、右詰	4 バイト	レコードバッファ内のこのレコードの長さ。レコードによって長さが異なることがあります。
	4 バイト	このレコードの Adabas レスポンス。ゼロ以外のレスポンスの場合、レコードはレコードバッファに格納されません。
	4 バイト	このレコードの ISN。
	4 バイト	(L9 のみ) ISN 数：このディスクリプタの値のカウント

最初のレコードの処理中にエラーが検出されると、レスポンスコードが適切な Adabas コントロールブロック（ACBRSP または ACBXRSP）のレスポンスコードフィールドに戻されます。

2 番目以降のレコードの処理中にエラーが検出されると、レスポンスコードが ISN バッファまたはマルチフェッチバッファの対応するレコードディスクリプタ要素に戻されます。


BT/ET のマルチフェッチ処理

Adabas はデフォルトでは、BT/ET コマンドを発行しているユーザーに対して現在ホールドされている ISN をすべて解放します。マルチフェッチオプションを指定すると、現在のトランザクションでホールドされているレコードのサブセットのみが解放されます。[ACB ダイレクトコールインターフェイス](#)を使用している場合、ホールド状態から解放されるレコードは ISN バッファに指定されます。[ACBX ダイレクトコールインターフェイス](#)を使用している場合、ホールド状態から解放されるレコードはマルチフェッチバッファに指定されます。ISN バッファまたはマルチフェッチバッファ内の最初のフルワードは、次の 8 バイトの要素の数になります。

コマンドレベルのマルチフェッチ機能を有効にするには、BT/ET コマンドコールの前に、Adabas コントロールブロックのフィールドに次の値を設定しておく必要があります。

ET コマンドまたは BT コマンドのダイレクトコールを行うときにマルチフェッチ処理を使用する場合は、コールの前に次のようにフィールドを設定しておく必要があります。

設定内容	設定対象		
	ACB インターフェイスダイレクトコール	ACBX インターフェイスダイレクトコール	
	ACB	ACBX	ABD
"BT" または "ET"	コマンドコード (ACBCMD)	コマンドコード (ACBXCMD)	---
"M"	コマンドオプション 1 (ACBCOP1)	コマンドオプション 1 (ACBXCOP1)	---
ISN バッファの長さ	ISN バッファ長 (ACBIBL)	---	---
マルチフェッチバッファの長さ	---	---	マルチフェッチバッファ ABD のバッファサイズ (ABDXSIZE)

 **Note:** ADARUN PREFETCH=YES でマルチフェッチを設定すると、ET/BT コマンドには "P" オプション (プリフェッチ) が自動的に使用されます。それ以外のコマンドには "M" オプション (マルチフェッチ) が自動的に使用されます。

ISN バッファまたはマルチフェッチバッファには、次の値が含まれている必要があります。

ISN or Multifetch Buffer: *ISN descriptor count* {*ISN descriptor element* (See table below)} ...


ISN ディスクリプタエレメントの構造を次の表に示します。

フォーマット	長さ	内容
バイナリ、右詰	4 バイト	Adabas ファイル番号
	4 バイト	ISN

プリフェッチ処理

プリフェッチは、**ACB** **ダイレクトコールインターフェイス**を使用してシーケンシャルコマンド (GET NEXT を指定した L1/L4、L2/L5、L3/L6、L9) をコールするプログラムで効果を発揮します。**ACBX** **ダイレクトコール**では使用できません。

プリフェッチを使用すると、一連の読み込みシーケンシャルコマンドが、1 回の Adabas コールで実行されます。1 回のコールでデータベースから一度に複数のレコードを読み込みます。この結果、リージョン間コミュニケーションのオーバーヘッドを大幅に削減し、ユーザープログラムと Adabas ニュークリアス処理をオーバーラップさせることができます。

 **Note:** ホールドオプション (L4/5/6 コマンド) を使用している場合、レコードがプリフェッチバッファエリアに読み込まれると、そのレコードはホールド状態になります。つまり、

全レコードの処理が完了する前に ET コマンドを発行すると、（未処理のレコードも含め）すべてのレコードが解放されます。ET コマンドの ISN ホールドオプションまたは HI コマンドを使用すると、このようなレコードをホールド状態に戻すことができます。

プリフェッチオプション処理から特定のコマンドまたはファイルを除外するには、ADARUN PREFXFIL パラメータまたは PREFXCMD パラメータを使用して、除外するファイルまたはコマンドを指定します。

このsectionでは、次のトピックについて説明します。

- コマンドオプション "P"を指定したプリフェッチ処理の呼び出し
- プリフェッチプログラミングのその他の考慮事項

コマンドオプション "P"を指定したプリフェッチ処理の呼び出し

コマンド固有の P オプションを指定してプリフェッチを有効にすると、Adabas はユーザープログラム内に定義された ISN バッファを先読みレコード用の中間ストレージエリアとして使用します。ISN バッファの各レコードには、16 バイトのヘッダーが付きます。

バイト	使用法
1~2	レコード長（長さの定義を含む） 長さ=ゼロはデータの終わりを示します。
3~4	ニュークリアスレスポンスコード
5~8	ニュークリアスの内部 ID（レスポンスコードがゼロでも 3 でもない場合、右端 2 バイトにサブコードが返されます）
9~12	レコードの ISN
13~16	ISN 数（L9 コマンドのみ）

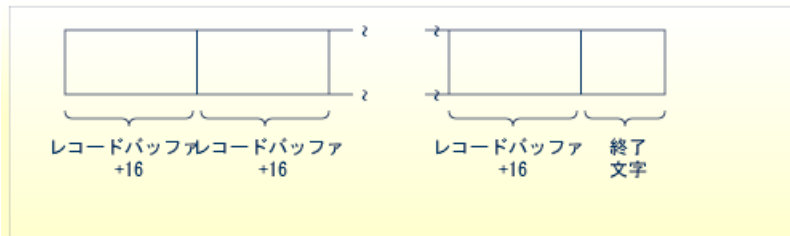
最初のレコードは、Adabas によってレコードバッファに返されます（16 バイトのヘッダーなし）。2 番目以降のレコードは、ユーザーが ISN バッファから取り出して処理する必要があります。エンドオブファイルが発生した場合、ISN バッファの最後のレコードのヘッダーには Adabas レスポンスコード 3 が含まれ、2 バイトの終了文字にはバイナリの 0 が含まれています。

プリフェッチプログラミングのその他の考慮事項

プリフェッチオプションを使用する場合には、次の事項を考慮してください。

- レコードバッファサイズは、想定される最大サイズの非圧縮レコードを格納するのに十分な大きさに設定します。
- エンドオブファイル条件が検出されるまでファイルの順次読み込みを継続しない場合は、ファイル処理の完了時に RC コマンドを発行してコマンド ID を解放します。
- ファイルの処理中は、コマンド ID を変更しないでください。

- コマンドオプションPを使用してプリフェッチ処理を実行する場合、ISNバッファサイズはレコードバッファ長に16バイトを加えた値の倍数に、さらに終了文字用の2バイトを足した大きさにします。



プリフェッチ プログラムのためのISN バッファサイズ

プリフェッチまたはマルチフェッチしたレコードの ISN を変更した場合の影響

プリフェッチまたはマルチフェッチしたレコードを更新した場合、次の処理が適用されます。

- ⚠ **Important:** ファイルでレコードを読み込んだり、検索したりしているときには、その同じファイル（特にアクセスされているディスクリプタ）を変更しないでください。
- 更新の保護が適用されるのはアクティブなセッションのみであり、それ以外のセッションで実行される処理は保護の対象に含まれません。
- プリフェッチまたはマルチフェッチしたレコードの ISN を変更した場合、そのレコードはプログラムがそのレコードに到達したときに（L1 コマンドによって）再度フェッチされます。
- プリフェッチまたはマルチフェッチしたレコードの ISN を削除した場合、そのレコードはスキップされます。
- プリフェッチまたはマルチフェッチしたレコードの ISN を挿入した場合、そのレコードはスキップされます。このため、ファイルでレコードを読み込んだり、検索したりしている間は、その同じファイルに変更を加えないことをお勧めします。

6 Adabas の呼び出し

- Adabas が ACB ダイレクトコールと ACBX ダイレクトコールを区別する方法 53
- ACB インターフェイスダイレクトコールの指定 53
- ACBX インターフェイスダイレクトコールの指定 54
- ACB ダイレクトコールと ACBX ダイレクトコールの混在 58

このchapterでは、Adabas を呼び出して Adabas コマンドを実行するための手順について説明します。Adabas ダイレクトコールは、ホスト言語（例えば、アセンブラ、COBOL、Fortran、C、PL/I）で用意されている標準のコール手順を使用します。



Note: 「[プログラミング例](#)」では、各種ホスト言語による Adabas コールの例をプログラミング例とともに示しています。

Adabas でサポートされているコントロールブロックインターフェイスごとに1つ、計2種類の Adabas ダイレクトコールがあります。

- **ACB** ダイレクトコールインターフェイスは、Adabas バージョン8より前の Adabas リリースで、従来からのダイレクトコールインターフェイスとして使用されています。このフォーマットのダイレクトコールには、従来の Adabas コントロールブロック（ACB）を使用する必要があります。Adabas 8より前の Adabas リリースをお使いの場合は、アプリケーションによるダイレクトコールが実行される際、ACB ダイレクトコールインターフェイスが使用されます。
- **ACBX** ダイレクトコールインターフェイスは、Adabas 8以降の Adabas リリースに導入されている拡張ダイレクトコールインターフェイスです。この形式のダイレクトコールでは、**拡張版**の Adabas コントロールブロック（ACBX）を使用する必要があります。Adabas 8（以降）がインストールされている場合は、アプリケーションでこのフォーマットのダイレクトコールを使用できます。それ以外の場合は、使用できません。

Adabas バージョン8では、ACB ダイレクトコールインタフェースと ACBX ダイレクトコールインタフェースの両方が完全にサポートされています。

- ACB ダイレクトコールインターフェイスを使用する既存のアプリケーションプログラムは、変更なしで従来と同じ方法で実行できます。
- さらに、ACBX ベースまたは ACB ベースのどちらのダイレクトコールインターフェイスをアプリケーションプログラムで使用するのかを、コールごとに決めることができます。1つのプログラムが両方のインターフェイスを使用できます。

コントロールブロックおよび関連バッファには、実行する Adabas コマンド、そのコマンドに必要な付加情報（パラメータまたはオペランド）を指定します。該当するコントロールブロック（ACBまたはACBX）へのポインタが常に、Adabas コールに指定する最初のオペランドである必要があります。

このchapterでは、次のトピックについて説明します。

Adabas が ACB ダイレクトコールと ACBX ダイレクトコールを区別する方法

すべてのアプリケーションプログラムは、ACB および ACBX のどちらのダイレクトコールも行えます。コントロールブロック（ACB または ACBX）は、ACB または ACBX インターフェイスのどちらを使用しても、Adabas コール最初のパラメータです。Adabas 8 は、コントロールブロックのオフセット 2 にある文字 "F" で始まる値によって、コールに使用するコントロールブロックを判断します。ACB のオフセット 2 はコマンドコードフィールド（ACBCMD）ですが、有効な F* Adabas コマンドがないため、ACB を使用する有効なダイレクトコールではオフセット 2 に文字 "F" で始まる値が含まれません。ACBX 内のオフセット 2 は、新しい ACBX を識別する新しいバージョンのフィールド（ACBXVER）です。

オフセット 2 上の "F" の有無により、Adabas 8 がダイレクトコールを解釈する方法が異なります。オフセット 2 に "F" が指定されている場合、Adabas は、コントロールブロックと残りのダイレクトコールパラメータを ACBX コールとして解釈します。オフセット 2 に "F" が指定されていない場合、Adabas は、コントロールブロックと残りのダイレクトコールパラメータを ACB コールとして解釈します。何らかの理由で残りのコントロールブロックフィールドおよびダイレクトコールパラメータが、オフセット 2 の "F" の有無により示されたコールのタイプに正しく指定されない場合は（ACB パラメータが ACBX コールに指定された場合など）、エラーとなるかコールの結果が予想どおりにならない場合があります。ACB または ACBX を使用してダイレクトコールを指定する方法の詳細については、「[ACB インターフェイスダイレクトコールの指定](#)」または「[ACBX インターフェイスダイレクトコールの指定](#)」を参照してください。

ACB インターフェイスダイレクトコールの指定

ACB インターフェイスを使用してダイレクトコールを行う場合は、次のような構文を使用します（これは COBOL の例です）。

```
CALL 'ADABAS' USING acb-control-block-name  
  
    [format-buffer]  
  
    [record-buffer]  
  
    [search-buffer]  
  
    [value-buffer]  
  
    [ISN-buffer]
```

ACB ダイレクトコールの場合、Adabas はこの構文に示されている順にバッファが指定されているものと見なします。コールに必要なバッファがない場合は、バッファを指定する必要はありません。ただし、フォーマットバッファは必要ないものの、他のバッファのいずれか（例えば、

レコードバッファ)が必要な場合は、レコードバッファより前にダミー (または空白) のフォーマットバッファを指定する必要があります。同じく、コールで ISN バッファのみが必要な場合は、ダミーのフォーマットバッファ、レコードバッファ、サーチバッファ、およびバリュースバッファも指定する必要があります。

次の表に示す斜体の文字は、実際の構文では置き換えられる項目です。ACB コントロールブロックのフォーマットおよび Adabas バッファの詳細については、「[Adabas コントロールブロック \(ACB\)](#)」および「[バッファの定義](#)」を参照してください。異なる ABD タイプ間の関係の詳細については、「[バッファタイプ間の関係](#)」を参照してください。

置き換えられる文字	実際に割り当てられる内容
<i>acb-control-block-name</i>	コールに使用する Adabas コントロールブロック (ACB) へのポインタ。
<i>format-buffer</i>	コールに使用する フォーマットバッファ の名前またはポインタ。単一の ACB ダイレクトコールには、フォーマットバッファを 1 つのみ指定できます。
<i>ISN-buffer</i>	コールに使用する ISN バッファ の名前またはポインタ。単一の ACB ダイレクトコールには、ISN バッファを 1 つのみ指定できます。
<i>record-buffer</i>	コールに使用する レコードバッファ の名前またはポインタ。単一の ACB ダイレクトコールには、レコードバッファを 1 つのみ指定できます。
<i>search-buffer</i>	コールに使用する サーチバッファ の名前またはポインタ。単一の ACB ダイレクトコールには、サーチバッファを 1 つのみ指定できます。
<i>value-buffer</i>	コールに使用する バリュースバッファ の名前またはポインタ。単一の ACB ダイレクトコールには、バリュースバッファを 1 つのみ指定できます。

ACBX インターフェイスダイレクトコールの指定

アプリケーションにダイレクトコールを指定する方法は、新しい ACBX インターフェイスを使用する場合には、従来の ACB インターフェイスと違った方法になります。また、コールはメインフレームアプリケーションおよびオープンシステムアプリケーションによってさまざまに異なります。この section では、次のトピックについて説明します。

- **メインフレームアプリケーションでの ACBX インターフェイスダイレクトコールの指定**

■オープンシステムアプリケーションでのACBXインターフェイスダイレクトコールの指定

メインフレームアプリケーションでのACBXインターフェイスダイレクトコールの指定

アプリケーションにダイレクトコールを指定する方法は、新しいACBXインターフェイスを使用する場合には、従来のACBインターフェイスと違った方法になります。メインフレームアプリケーションでACBXインターフェイスを使用してダイレクトコールを行う場合は、次のような構文を使用します（これはCOBOLの例です）。

```
CALL 'ADABAS' USING acbx-control-block-name

    reserved-fullword

    reentrancy-token

    [format-buffer-ABD record-buffer-ABD [multifetch-buffer-ABD]]...

    [search-buffer-ABD]

    [value-buffer-ABD]

    [ISN-buffer-ABD]

    [performance-buffer-ABD]

    [user-buffer-ABD]
```

各ABDは関連するバッファより前に直接記述するか、または各ABDにそのバッファへのポインタを含めます。これでバッファを効率的に表現できます。

ABDは、ACBXインターフェイスダイレクトコールのどの場所にも指定できますが、異なるタイプのABD同士を対にする必要がある場合には、ABDの対が順番になるように指定します。例えば、コールの中にフォーマットバッファのABDとレコードバッファのABDがそれぞれ3つずつ存在する場合、最初のフォーマットバッファのABDとレコードバッファABDが対になり、続いて2番目、3番目のABD同士が対になります。

指定したABDの対の数が一致しない場合は、不足分のダミーのABD（バッファ長はゼロ）が生成されます。例えば、フォーマットバッファのABDを3つ指定し、レコードバッファのABDを2つ指定した場合、3番目のフォーマットバッファのABDと対になるように、ダミーのレコードバッファのABDが1つ生成されます。ダミーのレコードバッファのABDと2番目のフォーマットバッファのABDを対にする必要がある場合は、3番目のフォーマットバッファと対になるレコードバッファのABDの前にダミーのレコードバッファのABDを指定する必要があります。

レコードバッファ内のデータを、フォーマットバッファ内のフォーマット指定では記述していないコマンドの場合は、フォーマットバッファセグメントの指定は必要ありません。指定した場合

は無視されます。このような指定が可能なコマンドは、OP コマンドの他に数種類しかありません。

次の表に示す斜体の文字は、実際の構文では置き換えられる項目です。拡張 Adabas コントロールブロック (ACBX) のフォーマット、Adabas バッファ記述 (ABD)、および Adabas バッファの詳細については、「[拡張 Adabas コントロールブロック \(ACBX\)](#)」、「[Adabas バッファ記述 \(ABD\)](#)」、および「[バッファの定義](#)」を参照してください。異なるバッファタイプ間の関係の詳細は、「[バッファタイプ間の関係](#)」を参照してください。

置き換えられる文字	実際に割り当てられる内容
<i>acbx-control-block-name</i>	コールに使用する拡張 Adabas コントロールブロック (ACBX) に対するポインタ。
<i>format-buffer-ABD</i>	コールに使用する フォーマットバッファ セグメントを定義する フォーマットバッファ ABD の名前またはポインタ。各 フォーマットバッファ セグメントは、ピリオドで終了し、単独の フォーマットバッファ として完結した有効なものである必要があります。単一の ACBX ダイレクトコールには、複数の フォーマットバッファ ABD を指定できます。
<i>ISN-buffer-ABD</i>	コールに使用する ISN バッファ セグメントを定義する ISN バッファ ABD の名前またはポインタ。単一の ACBX ダイレクトコールには、 ISN バッファ ABD を1つのみ指定できます。
<i>multifetch-buffer-ABD</i>	コールに使用する マルチフェッチバッファ セグメントを定義する マルチフェッチバッファ ABD の名前またはポインタ。単一の ACBX ダイレクトコールには、複数の マルチフェッチバッファ ABD を指定できます。
<i>performance-buffer-ABD</i>	Adabas Review で使用する パフォーマンスバッファ セグメントを定義する パフォーマンスバッファ ABD の名前またはポインタ。 パフォーマンスバッファ セグメントは、Adabas Review 用に予約されています。
<i>record-buffer-ABD</i>	コールに使用する レコードバッファ セグメントを定義する レコードバッファ ABD の名前またはポインタ。単一の ACBX ダイレクトコールには、複数の レコードバッファ ABD を指定できます。
<i>reentrancy-token</i>	ADALNK リエントランシー トークン。これは、ADALNK がスタティクなデータエリアのアドレスを格納するコール元プログラムのストレージのフルワードです。このフルワードは、最初の Adabas コールの前にゼロに設定しておく必要があります。フルワードの値は、プログラムの実行中、どのダイレクトコールでも変わりません。
<i>reserved-fullword</i>	バイナリの 0 が含まれているフルワード。このフルワードは、Adabas 用に予約されており、最初の Adabas コールの前にバイナリの 0 に設定しておく必要があります。
<i>search-buffer-ABD</i>	コールに使用する サーチバッファ セグメントを定義する サーチバッファ ABD の名前またはポインタ。単一の ACBX ダイレクトコールには、 サーチバッファ ABD を1つのみ指定できます。
<i>user-buffer-ABD</i>	コールに使用する ユーザーバッファ セグメント (拡張) を定義する ユーザーバッファ ABD の名前またはポインタ。 ユーザーバッファ 拡張 (UBX) は、ユーザー出口 LNKUEX1 (リンクルーチンプリコール出口) および LNKUEX2 (リンクルーチンポストコール出口) に渡されるユーザーデー

置き換えられる文字	実際に割り当てられる内容
	タに使用されます。ACBX ダイレクトコールには、単一のユーザーバッファ ABD を指定できます。
<i>value-buffer-ABD</i>	コールに使用するバリューストックを定義するバリューストック ABD の名前またはポインタ。単一の ACBX ダイレクトコールには、バリューストック ABD を 1 つのみ指定できます。

オープンシステムアプリケーションでの ACBX インターフェイスダイレクトコールの指定

アプリケーションにダイレクトコールを指定する方法は、新しい ACBX インターフェイスを使用する場合には、従来の ACB インターフェイスと違った方法になります。オープンシステムアプリケーションで ACBX インターフェイスを使用してダイレクトコールを行う場合は、次のような構文を使用します（これは COBOL の例です）。

```
CALL 'ADABAS' USING acbx-control-block-name
```

```
ABD-count
```

```
ABD-list-pointer
```

次の表に示す斜体の文字は、実際の構文では置き換えられる項目です。拡張 Adabas コントロールブロック (ACBX) のフォーマット、Adabas バッファ記述 (ABD)、ABD リスト、および Adabas バッファの詳細については、「[拡張 Adabas コントロールブロック \(ACBX\)](#)」、「[Adabas バッファ記述 \(ABD\)](#)」、「[ABD リスト](#)」、および「[バッファの定義](#)」を参照してください。

置き換えられる文字	実際に割り当てられる内容	条件
<i>acbx-control-block-name</i>	コールに使用する拡張 Adabas コントロールブロック (ACBX) に対するポインタ。	必須。
<i>ABD-count</i>	ダイレクトコールの ABD リストに含まれている ABD ポインタの数。	ABD およびその関連するバッファをダイレクトコールで使用する場合にのみ必須です。
<i>ABD-list-pointer</i>	ダイレクトコールの ABD リストへのポインタ。ABD リストには、ACBX ダイレクトコールが使用するすべての ABD へのポインタ参照が含まれています。ABD リストの詳細については、「 ABD リスト 」を参照してください。	ダイレクトコールにバッファが必要な場合にのみ必須です。

ACB ダイレクトコールと ACBX ダイレクトコールの混在

同じアプリケーションに ACB ダイレクトコールと ACBX ダイレクトコールを自由に混在させることができます。

TSO 環境またはバッチ環境では、同じアプリケーションで ACB と ACBX の両方のダイレクトコールインタフェースを使用して Adabas 8 非リエントラント (ADALNK) ダイレクトコールを呼び出した場合、コールに使用するワークエリアが ADALNK モジュール自体に含まれるため、ユーザーコンテキストが保持されます。一方、同じアプリケーションで ACB と ACBX の両方のダイレクトコールインタフェースを使用してリエントラント (ADALNKR) ダイレクトコールを呼び出す場合は、ユーザーコンテキストを自分で保持する必要があります。このことを行わない場合は、アプリケーションによって誤った結果が出力されることがあります。ADALNKR ダイレクトコールのユーザーコンテキストを保持する方法については、「ADALNKR への ACB および ACBX インターフェイスダイレクトコールの混在」を参照してください。

7 Adabas コントロールブロックの構造 (ACB および ACBX)

■ Adabas コントロールブロック (ACB)	61
■ 拡張 Adabas コントロールブロック (ACBX)	67
■ ACB と ACBX の相違点	77
■ コントロールブロックのロギング	82

Adabas では 2 種類のコントロールブロックがサポートされています。

- *Adabas* コントロールブロック (ACB) は、Adabas バージョン 8 より前の Adabas リリースに導入されている従来からのコントロールブロックです。Adabas 8 より前の Adabas リリースを使用している場合、アプリケーションによるダイレクトコールには ACB が使用されます。Adabas 8 では ACB が完全にサポートされているため、Adabas 8 をインストールすれば、既存のアプリケーションを更新する必要はありません。
- Adabas 8 以降のリリースでは、拡張 *Adabas* コントロールブロック (ACBX) を使用できます。ACBX は、Adabas 8 から増量されたバッファサイズと、Adabas 8 で導入されたセグメントバッファに対応しています。Adabas 8 (以降) がインストールされている場合、アプリケーションからのダイレクトコールに ACBX を使用できます。それ以外の場合は、使用できません。

コントロールブロックの各フィールドの用法については、「[コマンド](#)」で各 Adabas コマンドの説明を参照してください。今後の Adabas リリースとユーザープログラムとの互換性を確保するため、特定のコマンドによって使用されていないすべてのコントロールブロックフィールドは、そのフィールドタイプに応じてゼロまたは空白に設定する必要があります。

コントロールブロック内の各フィールドの位置は固定です。また、コントロールブロックの値はすべて、フィールドに定義されているデータタイプで入力する必要があります。例えば、ISN フィールドはバイナリ形式であるため、このフィールドの値はバイナリ形式で入力する必要があります。



Notes:

1. Adabas など Software AG プログラム製品は一部のコントロールブロックフィールドを内部で使用しており、ユーザーには意味のないフィールド値を返すことがあります。このようなフィールドの用法および値は、リリースに応じて異なるため、ユーザープログラムでは使用しないでください。このマニュアルで説明しているフィールドおよび値のみを使用することをお勧めします。また、未使用のコントロールブロックフィールドは、それぞれのフィールドタイプに応じてゼロまたは空白で初期化しておく必要があります。
2. Adabas 依存の Software AG 製品の中には、レスポンスコードやサブコードなどのコントロールブロック値を返すものがあります。製品固有のコントロールブロック値の説明については、該当する製品のマニュアルを参照してください。

このchapterでは、次のトピックについて説明します。

Adabas コントロールブロック (ACB)

Adabas コントロールブロック (ACB) の長さは 80 バイトです。このsectionでは、次のトピックについて説明します。

- ACB フォーマット
- ACB フィールド
- ACB DSECT
- ACB の例

ACB フォーマット

次の表は、ACB のフォーマットの説明です。ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

DSECT 名	フィールド	コントロールブロックの位置	オフセット	長さ (バイト)	フォーマット
ACBTYPE	コールタイプ	1	00	1	バイナリ
予約	(予約)	2	01	1	バイナリ
ACBCMD	コマンドコード	3~4	02	2	英数字
ACBCID	コマンド ID	5~8	04	4	英数字/バイナリ
ACBFNR	ファイル番号	9~10	08	2	バイナリ
ACBRSP	レスポンスコード	11~12	0A	2	バイナリ
ACBISN	ISN	13~16	0C	4	バイナリ
ACBISL	ISN 下限	17~20	10	4	バイナリ
ACBISQ	ISN 数	21~24	14	4	バイナリ
ACBFBL	フォーマットバッファ長	25~26	18	2	バイナリ
ACBRBL	レコードバッファ長	27~28	1A	2	バイナリ
ACBSBL	サーチバッファ長	29-30	1C	2	バイナリ
ACBVBL	バリュウバッファ長	31-32	1E	2	バイナリ
ACBIBL	ISN バッファ長	33~34	20	2	バイナリ
ACBCOP1	コマンドオプション 1	35	22	1	英数字
ACBCOP2	コマンドオプション 2	36	23	1	英数字
ACBADD1	アディション 1	37~44	24	8	英数字/バイナリ
ACBADD2	アディション 2	45~48	2C	4	英数字/バイナリ
ACBADD3	アディション 3	49~56	30	8	英数字
ACBADD4	アディション 4	57~64	38	8	英数字
ACBADD5	アディション 5	65~72	40	8	英数字/バイナリ

DSECT 名	フィールド	コントロールブロックの位置	オフセット	長さ (バイト)	フォーマット
ACBCMDT	コマンドタイム	73~76	48	4	バイナリ
ACBUSER	ユーザーエリア	77~80	4C	4	該当なし

ACB フィールド

コントロールブロックフィールドおよびバッファの内容は、Adabas コマンド (コール) を発行する前にセットしておかなければなりません。また、Adabas は各コマンドの実行後、特定のフィールドおよびバッファに 1 つ以上の値またはコードを返します。

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。


ACB の各フィールドについては、この section で説明します。[ACB フォーマット](#) の表示順に記載されています。この説明は、ほとんどの Adabas コマンドに当てはまりますが、一部の Adabas コマンドでは、この説明とは異なる目的でコントロールブロックフィールドが使用されます。これらのフィールドが各 Adabas コマンドによってどのように使用されるかの詳細については、「[コマンド](#)」を参照してください。

コールタイプ (ACBTYPE)

Adabas コントロールブロック (ADACB) の先頭バイトは、Adabas API がどの処理を実行するかを判断するのに使用されます。詳細については、「[アプリケーションと Adabas のリンク](#)」を参照してください。

論理リクエストの値は次のとおりです。

16 進数	説明
X'00'	1 バイトファイル番号 (ファイル番号 1~255) または DBID
X'30'	2 バイトファイル番号 (ファイル番号 1~65535) または DBID
X'40'	空白以上の値。この値は、以前の Adabas リリースとの互換性を維持するため、論理アプリケーションコールとして受け入れられます。

 **Note:** ただし、X'44'、X'48'、および X'4C' のコールは、Software AG 用に予約されているため受け入れられません。

ADACB の先頭バイトのうち、上記以外の値はすべて、Software AG 用に予約されています。

アプリケーションでは ADACB の先頭バイトの値を各コールでリセットできるため、単一のアプリケーションに 1 バイトファイル番号と 2 バイトファイル番号 (DBID) 要求を混在させることができます。この場合、コールタイプごとに ADACB のファイル番号 (ACBFNR) とレスポンスコード (ACBRSP) のフィールドを適切に構成する必要があります。詳細については、これらのフィールドの説明を参照してください。

2バイトファイル番号を使用するアプリケーションでは、ADACBの先頭バイトにX'30'、ACBRSPフィールドに論理データベースID、ACBFNRフィールドにファイル番号を設定することをお勧めします。そうすることによって、アプリケーションは、使用中のファイル番号の値に関係なく、2バイトのバイナリ整数としてデータベースIDとファイル番号の両方を処理することができます。

Software AG の Natural 言語でアプリケーションを作成すると、Natural が適切な値を指定するため、Adabas ACB の 1 バイト目にこの値を指定する必要はありません。


コマンドコード (ACBCMD)

コマンドコードは、実行するコマンドを定義し、OP、A1、BT のように 2 文字の英数字で構成されます。

コマンド ID (ACBCID)

コマンド ID フィールドは、後続のコマンドで使用する論理読み込み順序、検索結果、および (オプション) デコードされたフォーマットを識別するために、多くの Adabas コマンドにより使用されます。用途に応じて英数字またはバイナリのコマンド ID を使い分けたり、Adabas に新しいバイナリコマンド ID を生成するように要求することができます。コマンド ID の詳細については、「[プログラミング全般に関する考慮事項](#)」を参照してください。ET、CL および一部の OP コマンドの場合、Adabas は、バイナリのトランザクションシーケンス番号をコマンド ID フィールドに返します。

ファイル番号 (ACBFNR)

 **Note:** カップリングされた対のファイルで演算するコマンドの場合、このフィールドは、ISN またはデータが返されるプライマリファイルを指定します。

ファイル番号は 1 または 2 バイトです。

1 バイトのファイル番号および DBID

1~255 (1 バイト) のファイル番号に対応するアプリケーションプログラムから Adabas コマンドを発行する場合は、コントロールブロックを次のように作成します。

位置	処理内容
1	ADACB の先頭バイトに X'00' を配置します。
9	ADACB の ACBFNR フィールドの 2 番目 (右端) のバイトにファイル番号を配置します。ACBFNR フィールドの 1 バイト目 (左側) は、論理 (データベース) ID または番号を格納するために使用します。

ACBFNR フィールドの先頭バイトが 0 (B'0000 0000) の場合、Adabas API は DDCARD 入力データに指定された ADARUN カードのデータベース ID の値、またはオフセット X'80' でリンクルーチンにアセンブルされたデフォルトのデータベース ID 値を使用します。

2 バイトのファイル番号および DBID

Adabas は、論理リクエストで 255 より大きなファイル番号を使用できます。256~5000 (2 バイト) のファイル番号に対応するアプリケーションプログラムから Adabas コマンドを発行する場合は、コントロールブロックを次のように作成します。

位置	処理内容
1	ADACB の先頭バイトに X'30' を配置します。
9	ファイル番号に ACBFNR の 2 バイトを使用し、データベース (論理) ID に ACBRSP の 2 バイトを使用します。

ACBRSP フィールドが 0 の場合、Adabas API は DDCARD 入力データに指定された ADARUN カードのデータベース ID の値、またはオフセット X'80' でリンクルーチンにアセンブルされたデフォルトのデータベース ID 値を使用します。

レスポンスコード (ACBRSP)

レスポンスコードフィールドには、2 バイトのデータベース ID が使用されます。

また、Adabas コマンドの完了時に必ず値が設定されます。通常、正常終了はゼロのレスポンスコードで示されます。レコードシーケンスまたは ISN シーケンスを処理する繰り返し可能なコマンドの場合には、レスポンスコードはエンドオブファイルまたはエンドオブ ISN リストを示します。ゼロ以外のレスポンスコードは、『Adabas メッセージおよびコードマニュアル』に定義されています。

ISN (ACBISN)

フィールドコマンドに応じて 4 バイトの ISN 値を指定します。Adabas は、必要に応じてコマンドが生成した ISN リストの先頭 ISN か、またはコマンドが読み込んだレコードの ISN を返します。

ISN 下限 (ACBISL)

ISN 下限には、処理の開始位置となる ISN リストの開始点または範囲を指定します。OP コマンドでは、オプションとしてユーザー指定の非アクティビティタイムアウトの値を、このフィールドに指定できます。また、OP コマンドはこのフィールドに Adabas リリース情報も返します (アディクション 5 フィールドの説明を参照)。マルチフェッチオプションを使用した場合、このフィールドはプリフェッチレコードの最大カウント (オプション) を保持します。ゼロにすると制限なしになります。

ISN 数 (ACBISQ)

コマンドから返される ISN のカウントです。カウントは、ISN リストの ISN 数を合計した値、またはこの操作で大規模な ISN プールから ISN バッファ内に入る ISN 数の合計値です。OP コマンドは、このフィールドを使用してオプションのユーザー指定トランザクションタイムリミットを指定し、ISN 数フィールドにシステム情報やコールタイプ情報のフラグを返します (アディクション5フィールドの説明を参照)。また、セキュリティバイバリューを使用する Sx コマンドは、検索条件を満たす ISN が複数存在した場合、このフィールドに 1 を設定します。

バッファ長: フォーマット、レコード、サーチ、バリュー、および ISN (ACBFBL、ACBRBL、ACBSBL、ACBVBL、および ACBIBL)

フォーマット、レコード、サーチ、バリュー、ISN の各バッファ長フィールドには、関連するバッファのサイズを指定します。バッファサイズは通常、トランザクション全体で同じサイズに保たれます。一部の ISN 関連操作では、ISN バッファサイズ値によって、コマンドによる ISN の処理方法が決まります。例えば、ISN バッファ長をゼロにすると、ISN 結果リストが Adabas ワークエリアに格納されます。バッファが不要な Adabas コマンドの場合は、対応するバッファ長値にゼロを設定する必要があります。また、バッファ長が制限されている場合もあります (例えば、マルチフェッチオプション)。詳細については、それぞれのコマンドの説明を参照してください。

コマンドオプション1 およびコマンドオプション2 (ACBCOP1 および ACBCOP2)

コマンドオプション1フィールドおよびコマンドオプション2フィールドには、処理オプションを指定できます (ISN ホールド、コマンドレベルのプリフェッチ制御、返される ISN など)。

アディクション1 (ACBADD1)

ISN リストを作成するディスクリプタの分類、またはカップリングした対のファイルの2番目のファイル番号など、その他のコマンド関連パラメータをアディクション1フィールドに指定することが必要な場合があります。

アディクション2 (ACBADD2)

An、Ln、Nn、S1/2/4 のどのコマンドでも、左端 (上位) 2 バイトに圧縮レコード長を返し、右端 (下位) 2 バイトにレコードバッファで選択されたフィールドの非圧縮長を返します。OP (オープン) および RE (ET データの読み込み) コマンドは、このフィールドにトランザクションシーケンス番号を返します。Entire Net-Work がインストールされている場合、一部のレスポンスコードでは、アディクション2フィールドの左端2バイトに、障害が発生した Entire Net-Work ノードのノード ID が返されます。

コマンドの結果がニュークリアスレスポンスコードである場合、アディクション2フィールドの下位 (右端) 2 バイト (47 と 48) にレスポンスコードの原因を識別する 16 進のサブコードが返されることがあります。例えば、セッションを OP コマンドで開始していないものの ADARUN ステートメントに OPENRQ=YES を指定している場合、レスポンスコード 9、サブコード 66 が返され、この 2 バイトには 10 進数の 66 に相当する 16 進値 0042 が設定されます。レスポンス

コードとサブコードの詳細については、『*Adabas* メッセージおよびコードマニュアル』の「ニュークリアスレスポンスコード」の説明を参照してください。

アディション 3 (ACBADD3)

アディション 3 フィールドには、パスワードで保護されたファイルにアクセスするのに必要なユーザーのパスワードを設定します。ファイルが実際にパスワードで保護されている場合、コマンドの実行中にフィールド内のパスワードがスペース (空白) に置き換えられた後、ユーザープログラムに制御が戻されます。

アディション 4 (ACBADD4)

アディション 4 フィールドには、暗号化 (サイファ化) されたデータベースのデータファイルを読み書きする命令に合わせたサイファコードを設定する必要があります。複数のコマンド ID を必要とするコマンドでは、サイファコードが不要であれば、コマンド ID の 1 つをこのフィールドに指定します。

ニュークリアスによって処理される Adabas コールは、Adabas リリース (バージョンおよび改訂) レベル番号およびデータベース ID をアディション 4 フィールドの下位 (右端) 3 バイトに次のような *vrnnnn* 形式で返します。

v Adabas バージョン番号を示します。

r Adabas リリース番号を示します。

nnnn コールを処理した Adabas データベースの番号 (16 進) を示します。

例えば、"741111" は、データベース 4369 の Adabas バージョン 7.4 ニュークリアスがコールを処理したことを示します。

アディション 5 (ACBADD5)

アディション 5 フィールドの 1 バイト目の上位 (左端) 2 ビットは、ユニークなフォーマット ID かグローバルフォーマット ID のどちらが選択されているかを制御します。下位 (右端) 4 バイトまたは 8 バイトは、選択に応じてユニークなフォーマット ID かグローバルフォーマット ID のいずれかを格納することができます この機能の詳細については、「[グローバルフォーマット ID の使用](#)」を参照してください。RC (コマンド ID の解放) コマンドでは、削除するグローバルフォーマット ID をこのフィールドに指定できます。コールが完了すると、OP コマンドはオプションで指定された非アクティビティタイムアウトやトランザクションタイムアウトの値をアディション 5 フィールドに返します。

コマンドタイム (ACBCMDT)

コマンドタイムフィールドは、ニュークリアスがコマンドを処理するのに要した経過時間を返すために、Adabas が使用します。スレッドが Adabas I/O オペレーションなどのリソースを待機していた時間は含まれません。時間は 16 マイクロ秒単位で、"Adabas スレッドタイム" と呼ばれ、バイナリ形式です。

ユーザーエリア (ACBUSER)

ユーザーエリアフィールドは、ユーザープログラムで使用できるように予約されています。論理ユーザーコールを発行したとき、Adabas はユーザーエリアをいっさい読み書きしません。

今後の Adabas リリースとの互換性のため、未使用のコントロールブロックフィールドはフィールドデータタイプに合わせた空値を設定しておくことをお勧めします。

ACB DSECT

ACB DSECT は、分散 Adabas SRCE ライブラリのメンバ ADACB にあります。

ACB の例

さまざまなホスト言語のコントロールブロック構造を示すプログラミング例は、このマニュアルの「[プログラミング例](#)」に記載されています。

- [アセンブラの例](#)
- [COBOL の例](#)
- [PL/I の例](#)
- [FORTRAN の例](#)

拡張 Adabas コントロールブロック (ACBX)

拡張 Adabas コントロールブロック ACBX は、Adabas コマンドのバッファサイズの増量に対応しています。ACB では 80 バイトでしたが、192 バイトに増えています。既存の非拡張 Adabas コントロールブロック (ACB) も引き続きサポートされ、既存のアプリケーションも機能しますが、Adabas 8 で導入された拡張機能を活用する場合には、新しい ACBX を使用する必要があります。特に、Adabas 8 の長い (32K を超える) バッファまたはセグメント化されたバッファ (フォーマットバッファとレコードバッファの複数ペア、またはフォーマット、レコードおよびマルチフェッチの3つのバッファの複数セット) 機能を使用している場合、ACBX を使用する必要があります。

ACBX を使用しない場合は、お使いのアプリケーションプログラムは既存のダイレクトコールインターフェイス (ACB) を使用した Adabas コールと新しいインターフェイス (ACBX) を使用したコールを自由に切り替えることができます。

- ACBX フォーマット
- ACBX フィールド
- ACBX DSECT

ACBX フォーマット

次の表は、ACBX のフォーマットの説明です。ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

DSECT フィールド名	フィールド	コントロールブロックの位置	オフセット	長さ (バイト)	フォーマット
ACBXTYP	コールタイプ	1	00	1	バイナリ
ACBXRSV1	予約 1	2	01	1	バイナリ
ACBXVER	バージョンインジケータ	3~4	02	2	バイナリ
ACBXLEN	ACBX 長	5~6	04	2	バイナリ
ACBXCMD	コマンドコード	7~8	06	2	英数字
ACBXRSV2	予約 2	9~10	08	2	バイナリ
ACBXRSP	レスポンスコード	11~12	0A	2	バイナリ
ACBXCID	コマンド ID	13~16	0C	4	英数字/バイナリ
ACBXDBID	データベース ID	17~20	10	4	数値
ACBXFNR	ファイル番号	21~24	14	4	数値
ACBXISNG	8 バイト ISN	25~32	18	8	使用不可
ACBXISN	ISN	29~32	1C	4	バイナリ
ACBXISLG	8 バイト (ISN 下限)	33~40	20	8	使用不可
ACBXISL	ISN 下限	37~40	24	4	バイナリ
ACBXISQG	8 バイト (ISN 数)	41~48	28	8	使用不可
ACBXISQ	ISN 数	45~48	2C	4	バイナリ
ACBXCOP1	コマンドオプション 1	49	30	1	英数字
ACBXCOP2	コマンドオプション 2	50	31	1	英数字
ACBXCOP3	コマンドオプション 3	51	32	1	英数字
ACBXCOP4	コマンドオプション 4	52	33	1	英数字
ACBXCOP5	コマンドオプション 5	53	34	1	英数字
ACBXCOP6	コマンドオプション 6	54	35	1	英数字
ACBXCOP7	コマンドオプション 7	55	36	1	英数字
ACBXCOP8	コマンドオプション 8	56	37	1	英数字
ACBXADD1	アディション 1	57~64	38	8	英数字/バイナリ
ACBXADD2	アディション 2	65~68	40	4	バイナリ
ACBXADD3	アディション 3	69~76	44	8	英数字/バイナリ

DSECTフィールド名	フィールド	コントロールブロックの位置	オフセット	長さ (バイト)	フォーマット
ACBXADD4	アディクション 4	77~84	4C	8	英数字
ACBXADD5	アディクション 5	85~92	54	8	英数字/バイナリ
ACBXADD6	アディクション 6	93~100	5C	8	英数字/バイナリ
ACBXRSV3	予約 3	101~104	64	4	バイナリ
ACBXERRG	バッファ内のエラーオフセット (64 ビット)	105~112	68	8	使用不可
ACBXERRA	バッファ内のエラーオフセット (32 ビット)	109~112	6C	4	バイナリ
ACBXERRB	エラー文字フィールド	113~114	70	2	英数字
ACBXERRC	エラーサブコード	115~116	72	2	バイナリ
ACBXERRD	エラーバッファ ID	117	74	1	英数字
ACBXERRE	予約済み (将来的に使用される 予定)	118	75	1	使用不可
ACBXERRF	エラーバッファシーケンス番号	119~120	76	2	数値
ACBXSUBR	サブコンポーネントレスポンス コード	121~122	78	2	バイナリ
ACBXSUBS	サブコンポーネントレスポンス サブコード	123~124	7A	2	バイナリ
ACBXSUBT	サブコンポーネントエラーテキ スト	125~128	7C	4	英数字
ACBXCMP	圧縮レコード長	129~136	80	8	バイナリ
ACBXLDEC	非圧縮レコード長	137~144	88	8	バイナリ
ACBXCMDT	コマンドタイム	145~152	90	8	バイナリ
ACBXUSER	ユーザーエリア	153~168	98	16	該当なし
ACBXRSV4	予約 4	169~193	A8	24	操作不可

ACBX フィールド

コントロールブロックフィールドおよびバッファの内容は、Adabas コマンド (コール) を発行する前にセットしておかなければなりません。また、Adabas は各コマンドの実行後、特定のフィールドおよびバッファに 1 つ以上の値またはコードを返します。

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

ACBX の各フィールドについては、この section で説明します。[ACBX フォーマット](#) の表示順に記載されています。この説明は、ほとんどの Adabas コマンドに当てはまりますが、一部の Adabas コマンドでは、この説明とは異なる目的でコントロールブロックフィールドが使用され

ます。これらのフィールドが各 Adabas コマンドによってどのように使用されるかの詳細については、「[コマンド](#)」を参照してください。

コールタイプ (ACBXTYP)

Adabas コントロールブロック (ADACBX) の先頭バイトは、Adabas API がどの処理を実行するかを判断するのに使用されます。詳細については、『*Adabas オペレーションマニュアル*』の「[アプリケーションと Adabas のリンク](#)」を参照してください。

Adabas コマンドを発行するときに、このフィールドをバイナリの 0 に設定します。これは、ACBXTUSR のように、論理ユーザーコールが行われていることを示します。

ACBXTYPE の値の X'04'、X'08'、X'0c'、X'10'、X'14'、X'18'、X'1c'、X'20'、X'24'、X'28'、X'2c'、X'34'、X'38'、X'3c'、X'44'、X'48'、および X'4c' は、Software AG で使用するために予約されています。そのため、アプリケーションプログラムで使用することはできません。

Software AG の Natural 言語で作成されたアプリケーションは、Natural が適切な値を指定するので、Adabas ACBX の 1 バイト目にこの値を指定する必要はありません。

予約 1 (ACBXRSV1)

このフィールドは予約されています。このフィールドにはゼロを設定します。

バージョンインジケータ (ACBXVER)

バージョンインジケータは、Adabas コントロールブロックが新しい ACBX フォーマットを使用するのか、従来の ACB フォーマットを使用するのかを示します。このフィールドを文字 "F" で始まる値 (例えば "F2") に設定すると、Adabas コントロールブロックは ACBX フォーマットで指定されているものとして扱われます。このフィールドをそれ以外の値に設定すると、Adabas コントロールブロックは従来の ACB フォーマットで指定されているものとして扱われます。

ACBX 長 (ACBXLEN)

ACBX 長フィールドは、Adabas に渡される ACBX 構造の長さ (ACBXQLL と同じく現時点では 192) に設定する必要があります。

コマンドコード (ACBXCMD)

コマンドコードは、実行するコマンドを定義し、OP、A1、BT のように 2 文字の英数字で構成されます。

予約 2 (ACBXRSV2)

このフィールドは予約されています。このフィールドにはゼロを設定します。

レスポンスコード (ACBXRSP)

このフィールドには、Adabas コマンドの完了時に値が設定されます。通常、正常終了はゼロのレスポンスコードで示されます。レコードシーケンスまたは ISN シーケンスを処理する繰り返し可能なコマンドの場合には、レスポンスコードはエンドオブファイルまたはエンドオブ ISN リストを示します。ゼロ以外のレスポンスコードについては、「Adabas メッセージおよびコード」を参照してください。

コマンド ID (ACBXCID)

コマンド ID フィールドは、後続のコマンドで使用する論理読み込み順序、検索結果、および（オプション）デコードされたフォーマットを識別するために、多くの Adabas コマンドにより使用されます。用途に応じて英数字またはバイナリのコマンド ID を使い分けたり、Adabas に新しいバイナリコマンド ID を生成するように要求することができます。コマンド ID の詳細については、「[プログラミング全般に関する考慮事項](#)」を参照してください。ET、CL および一部の OP コマンドの場合、Adabas は、バイナリのトランザクションシーケンス番号をコマンド ID フィールドに返します。

データベース ID (ACBXDBID)

このフィールドを使用して、データベース ID を指定します。Adabas コールはこのデータベースに送られます。


このフィールドは 4 バイトのバイナリフィールドですが、現時点では 2 バイトのデータベース ID のみがサポートされています。したがって、データベース ID はフィールドの下位部分（右端のバイト）に指定し、その前にバイナリの 0 を付加する必要があります。

このフィールドがバイナリの 0 に設定されている場合は、Adabas API は DDCARD 入力データで指定された ADARUN カードのデータベース ID か、リンクされているカリクルーチンによりロードされた LNKGBLS モジュールで指定されているデフォルトのデータベース ID 値のいずれかを使用します。

ファイル番号 (ACBXFNR)

このフィールドを使用して、Adabas コールで送る必要のあるファイル数を指定します。

このフィールドは 4 バイトのバイナリフィールドですが、ファイル番号はフィールドの下位部分（右端のバイト）に指定し、その前にバイナリの 0 を付加する必要があります。

 **Note:** カップリングされた対のファイルで演算するコマンドの場合、このフィールドは、ISN またはデータが返されるプライマリファイルを指定します。

ISN (ACBXISNG/ACBXISN)

ISN フィールドにはコマンドに必要な Adabas ISN 値を指定し、必要に応じてコマンドが読み込んだレコードの ISN か、またはコマンドが生成した ISN リストの最初の ISN が返されます。

ACBXISN フィールドは、8 バイトの ACBXISNG フィールドに埋め込まれた 4 バイトのバイナリフィールドで、まだ使用されていません。ACBXISNG フィールドの上位部分には、バイナリの 0 を設定します。

ISN 下限 (ACBXISLG/ACBXISL)

ISN 下限フィールドには、処理の開始位置となる ISN リストの開始点または範囲を指定します。

OP コマンドでは、オプションとしてユーザー指定の非アクティビティタイムアウトの値を、このフィールドに指定できます。また、OP コマンドはこのフィールドに Adabas リリース情報を返します。

マルチフェッチ オプションを使用した場合、このフィールドにはプリフェッチされるレコードの最大カウント (オプション) が保持されます。ゼロにすると制限なしになります。

ACBXISL フィールドは、8 バイトの ACBXISLG フィールドに埋め込まれた 4 バイトのバイナリフィールドで、まだ使用されていません。ACBXISLG フィールドの上位部分には、バイナリの 0 を設定します。

ISN 数 (ACBXISQG/ACBXISQ)

ISN 数フィールドは、検索 (Sx) コマンドが返した ISN のカウントです。このカウントは、ISN リストの ISN 数を合計した値、またはこの操作で大規模な ISN プールから ISN バッファセグメントに入る ISN 数の合計値です。検索コマンドでセキュリティバイバリューを使用すると、複数の ISN が検索条件を満たした場合、ISN 数フィールドが "1" に設定されます。

OP コマンドの場合、オプションとしてユーザー固有のトランザクションタイムリミットをこのフィールドに指定できます。OP コマンドは、このフィールドにシステム情報およびコールタイプ情報を返します。

ACBXISQ フィールドは、8 バイトの ACBXISQG フィールドに埋め込まれた 4 バイトのバイナリフィールドで、まだ使用されていません。ACBXISQG フィールドの上位部分には、バイナリの 0 を設定します。

1 から 8 までのコマンドオプション (ACBXCOP1~ACBXCOP8)

コマンドオプション1からコマンドオプション8までのフィールドには、処理オプションを指定できます (ISN ホールド、コマンドレベルのプリフェッチ制御、返される ISN など)。Adabas 8.1 では、コマンドオプション1フィールドとコマンドオプション2フィールドのみがサポートされています。それ以外のコマンドオプションフィールドは、今後の Adabas リリースでの拡張のために用意されています。

アディクション 1 (ACBXADD1)

ISN リストを作成するディスクリプタの分類、またはカップリングした対のファイルの2番目のファイル番号など、その他のコマンド関連パラメータをアディクション1フィールドに指定することが必要な場合があります。

アディクション 2 (ACBXADD2)

OP (オープン) および RE (ET データの読み込み) コマンドは、このフィールドにトランザクションシーケンス番号を返します。

アディクション 3 (ACBXADD3)

アディクション3フィールドには、パスワードで保護されたファイルにアクセスするのに必要なユーザーのパスワードを設定します。このフィールドは、コマンドの実行中、常に空白にリセットされます。

アディクション 4 (ACBXADD4)

暗合化 (サイファ化) された Adabas ファイルのレコードをコマンドで読み書きする場合は、アディクション4フィールドをそのファイルのサイファコードに設定する必要があります。複数のコマンド ID を必要とするコマンドでは、サイファコードが不要であれば、コマンド ID の1つをこのフィールドに指定します。

このフィールドは、コマンドの実行中、常に空白にリセットされます。

シングルユーザーモードを実行していない (ADARUN MODE=SINGLE と指定していない) ニュークリアスで Adabas コールを処理すると、アディクション4フィールドの下位 (右端) 3 バイトに Adabas リリース (バージョンおよび改訂) レベル番号およびデータベース ID が次のようなフォーマット *vrnnnn* で返されます。

- v Adabas バージョン番号を示します。
- r Adabas リリース番号を示します。
- nnnn コールを処理した Adabas データベースの番号 (16 進) を示します。

例えば、"811111" は、データベース 4369 の Adabas バージョン 8.1 ニュークリアスがコールを処理したことを示します。

アディション 5 (ACBXADD5)

アディション 5 フィールドの 1 バイト目の上位 (左端) 2 ビットは、ユニークなフォーマット ID かグローバルフォーマット ID のどちらが選択されているかを制御します。下位 (右端) 4 バイトまたは 8 バイトは、選択に応じてユニークなフォーマット ID かグローバルフォーマット ID のいずれかを格納することができます RC (コマンド ID の解放) コマンドでは、削除するグローバルフォーマット ID をこのフィールドに指定できます。コールが完了すると、OP コマンドはオプションで指定された非アクティビティタイムアウトやトランザクションタイムアウトの値をアディション 5 フィールドに返します。

アディション 6 (ACBXADD6)

このフィールドは現時点では使用されません。バイナリの 0 に設定する必要があります。

予約 3 (ACBXRSV3)

このフィールドは予約されています。このフィールドはバイナリの 0 に設定する必要があります。

バッファ内のエラーオフセット (64 ビット) (ACBXERRG)

バッファのエラーオフセット (64 ビット) およびバッファのエラーオフセット (32 ビット) フィールドは、存在する場合、ダイレクトコール中にエラーが検出されたバッファのオフセットを示します。

バッファのエラーオフセット (64 ビット) フィールドの ACBXERRG は使用できませんが、今後のリリースで使用できるようになる可能性があります。現時点では、バッファのエラーオフセット (32 ビット) フィールドの ACBXERRA を使用してください。

ACBXERRx フィールドは、レスポンスコードがダイレクトコールから返されたときにのみ設定されます。ACBXERRA、ACBXERRD、ACBXEFFE の各フィールドは、レスポンスコードがバッファ処理に関連しているときにのみ設定されます。

バッファ内のエラーオフセット (32 ビット) (ACBXERRA)

バッファのエラーオフセット (64 ビット) およびバッファのエラーオフセット (32 ビット) フィールドは、存在する場合、ダイレクトコール中にエラーが検出されたバッファのオフセットを示します。

バッファのエラーオフセット (64 ビット) フィールドの ACBXERRG は使用できませんが、今後のリリースで使用できるようになる可能性があります。現時点では、バッファのエラーオフセット (32 ビット) フィールドの ACBXERRA を使用してください。

ACBXERRx フィールドは、レスポンスコードがダイレクトコールから返されたときにのみ設定されます。ACBXERRA、ACBXERRD、および ACBXERRF フィールドは、レスポンスコードがバッファ処理に関連しているときにのみ設定されます。

エラー文字フィールド (ACBXERRB)

このフィールドは、エラー検出時に処理されていたフィールドがあれば、そのフィールドの2バイトの Adabas ショートネームを示します。

ACBXERRx フィールドは、レスポンスコードがダイレクトコールから返されたときにのみ設定されます。

エラーサブコード (ACBXERRC)

このフィールドには、ダイレクトコール処理中に発生したエラーのサブコードが格納されます。

ACBXERRx フィールドは、レスポンスコードがダイレクトコールから返されたときにのみ設定されます。Entire Net-Work がインストールされている場合、レスポンスコードによってはこのフィールドに障害が発生したノードのノード ID が返されます。

エラーバッファ ID (ACBXERRD)

このフィールドには、ACBXERRA フィールドが参照するバッファの ID (ABDID フィールドから取得したもの) が格納されるため、複数のバッファが関与している場合にも、エラーが発生しているバッファを特定できます。

ACBXERRx フィールドは、レスポンスコードがダイレクトコールから返されたときにのみ設定されます。ACBXERRA、ACBXERRD、および ACBXERRF フィールドは、レスポンスコードがバッファ処理に関連しているときにのみ設定されます。

予約 (ACBXERRE)

このフィールドは将来の使用のために予約されています。現時点ではこのフィールドを使用しないでください。

エラーバッファシーケンス番号 (ACBXERRF)

ACBXERRA フィールドおよび ACBXERRD フィールドを参照するエラーがバッファセグメントに含まれている場合、このフィールドにはそのセグメントの2バイトのシーケンス番号が入ります。

ACBXERRx フィールドは、レスポンスコードがダイレクトコールから返されたときにのみ設定されます。ACBXERRA、ACBXERRD、および ACBXERRF フィールドは、レスポンスコードがバッファ処理に関連しているときにのみ設定されます。

サブコンポーネントレスポンスコード (ACBXSUBR)

このフィールドには、Adabas アドオン製品が Adabas コマンドをインターセプトしたときに発生したエラーからのレスポンスコードが入ります。

サブコンポーネントレスポンスサブコード (ACBXSUBS)

このフィールドには、Adabas アドオン製品が Adabas コマンドをインターセプトしたときに発生したエラーからのレスポンスサブコードが入ります。

サブコンポーネントエラーテキスト (ACBXSUBT)

このフィールドには、Adabas アドオン製品が Adabas コマンドをインターセプトしたときに発生したエラーのエラーテキストが入ります。

圧縮レコード長 (ACBXLCMP)

このフィールドには、レコード読み書き時の圧縮レコード長が返されます。

これは、正常に完了した Adabas コールが処理した圧縮データの長さです。論理データストレージレコードが複数の物理データレコードにスパンしている場合、関連する物理レコードをすべて結合した長さがわからなくなることがあります。この場合、Adabas はこのフィールドの下位ワードに高い値を返します。

非圧縮レコード長 (ACBXLDEC)

このフィールドには、非圧縮レコード長が返されます。これは、正常に完了したコールが処理した非圧縮データの長さです。複数のレコードバッファセグメントを指定した場合は、バッファセグメントをすべて合計した長さになります。

コマンドタイム (ACBXCMDT)

コマンドタイム (スレッドタイムとも言う) フィールドは、ニュークリアスがコマンドを処理するのに要した経過時間を返すために、Adabas が使用します。これには、コマンドを実行する Adabas スレッドが Adabas I/O オペレーションまたは他のリソースで待機する時間は含まれませんが、コードを実行できるよう、スレッドがプロセッサを待機する時間は含まれます。経過時間は 1/4096 ミリ秒単位で計測され、バイナリ形式で表されます。

ユーザーエリア (ACBXUSER)

ユーザーエリアフィールドは、ユーザープログラムで使用できるように予約されています。論理ユーザーコールを発行したとき、Adabas はユーザーエリアをいっさい読み書きしません。

予約 4 (ACBXRSV4)

このフィールドは、Adabas での使用のために予約されています。ユーザープログラムでは、この ACBX を使用して最初の Adabas コールを実行する前にこのフィールドをバイナリの 0 に設定し、以後同じ値のままにしておく必要があります。

ACBX DSECT

ACBX DSECT は、分散 Adabas SRCE ライブラリのメンバ ADACBX にあります。また、この section には、参考のために **ACBX フォーマット** 表も記載されています。

ACB と ACBX の相違点

ACBX は多くの点で ACB と異なります。ACBX には ACB にはないフィールドがいくつかあり、一部の ACBX フィールドのサイズは ACB の同じフィールドよりも大きくなっています。ACBX におけるこれらの拡張は、基本的な構造を長期間にわたって変更することなく将来行われる予定の Adabas への機能強化に対応できるように、ACBX 構造に十分な柔軟性を持たせるために行われました。

この section では、ACB と ACBX の相違点について説明します。

- **コントロールブロック長**
- **バッファ長フィールド**
- **コマンドオプションフィールド、アディショナルフィールド、および予約フィールド**

- 単位の相違
- フィールド長の相違
- ACBX に追加されたフィールド
- ACB 多目的フィールドの変化
- 構造およびオフセットの相違点

コントロールブロック長

ACBX の長さは 192 (または X'C0') バイトで、ACB の長さは 80 バイトです。

バッファ長フィールド

バッファ長フィールドは ACB に含まれていますが ACBX には含まれていません。ACBX ダイレクトコールインタフェースを使用している場合は、このフィールドは個々の Adabas バッファ記述 (ABD) に用意されます。したがって ACBX には、ACB に含まれる ACBFBL、ACBIBL、ACBRBL、ACBSBL および ACBVBL に対応するバッファフィールドが含まれません。コールに関連付けられた ABD が代わりに使用されます。1 つの ABD は、単独の Adabas バッファセグメントを表します。詳細は、「[Adabas バッファ記述](#)」で説明しています。

コマンドオプションフィールド、アディションフィールド、および予約フィールド

ACBXの方が、コマンドオプション、アディション、予約済みコントロールブロックのフィールドの数が多くなっています。

- ACBX には、8 個のコマンドオプションフィールドが含まれていますが、ACB では 2 個です。
- ACBX には、6 個のアディションフィールドが含まれていますが、ACB では 5 個です。
- ACBX には、4 個の予約フィールドが含まれていますが、ACB では 1 個です。

予約された ACBX フィールドはバイナリの 0 に設定する必要があります。予約された 4 フィールド (ACBXRSV4) は、バイナリの 0 に初期化する必要があります、常にその状態を保つようにします。

単位の相違

コマンドタイム (スレッドタイム) を計測する単位は ACB と ACBX で異なります。ACB では、コマンドタイム (ACBCMDT) を 16 マイクロ秒単位で計測します。ACBX では、コマンドタイム (ACBXCMDT) を 1/4096 マイクロ秒単位で計測します。

フィールド長の相違

多くのコントロールブロックフィールドで ACBX の方が長さが長くなっています。次の表は、これらの変更点を要約したものです。

フィールドタイトル	長さ	
	ACB	ACBX
ファイル番号	2	4
データベース ID	2	4
ISN	4	4
ISN 下限	4	4
ISN 数	4	4
圧縮レコード長	4	8
非圧縮レコード長	4	8
コマンドタイム	4	8
ユーザーエリア	4	16
フォーマットバッファ長	2 4 (ABD 内)	
レコードバッファ長	2 4 (ABD 内)	
サーチバッファ長	2 4 (ABD 内)	
バリューストックバッファ長	2 4 (ABD 内)	

ACBX に追加されたフィールド

ACBX には次のフィールドが追加されています。

ACBX DSECT 名	説明
ACBXADD6	アディクション 6
ACBXCOP3	コマンドオプション 3
ACBXCOP4	コマンドオプション 4
ACBXCOP5	コマンドオプション 5
ACBXCOP6	コマンドオプション 6
ACBXCOP7	コマンドオプション 7
ACBXCOP8	コマンドオプション 8
ACBXDBID	データベース ID です。ACB では、データベース ID は X'30' コールではレスポンスコードフィールド (ACBRSP) 内、その他の論理コールでは ACBFNR の 1 バイト目に格納されます。
ACBXERRA	バッファ内へのエラーオフセット (32 ビット) です。
ACBXERRB	エラー文字フィールド (フィールド名) です。

ACBX DSECT 名	説明
ACBXERRC	エラーサブコードです。
ACBXERRD	複数のバッファが関連する場合のエラーバッファ ID です。
ACBXERRE	複数のバッファが関連する場合のエラーバッファシーケンス番号です。
ACBXERRG	バッファ内へのエラーオフセット (64 ビット) です。このフィールドは現在サポートされていません。
ACBXLCMP	圧縮レコード長 (レコード全体が読み込まれていない場合はレコードの一部) です。ACB では、圧縮レコード長はアディショナル 2 フィールド (ACBADD2) に格納されます。
ACBXLDEC	非圧縮レコード長です。ACB では、非圧縮レコード長はアディショナル 2 フィールド (ACBADD2) に格納されます。
ACBXLEN	ACBX の長さ (現在 192) です。
ACBXRSV2	予約済みです。このフィールドの値はゼロに設定されている必要があります。
ACBXRSV3	予約済みです。このフィールドの値はゼロに設定されている必要があります。
ACBXRSV4	Adabas が使用するように予約済みです。
ACBXSUBR	Adabas アドオン製品により使用されるサブコンポーネントレスポンスコードです。
ACBXSUBS	Adabas アドオン製品により使用されるサブコンポーネントレスポンスサブコードです。
ACBXSUBT	Adabas アドオン製品により使用されるサブコンポーネントエラーテキストです。
ACBXVER	C'F2' に設定した場合、このフィールドは Adabas に新しい拡張 ACB (ACBX) が使用されていることを示します。

ACB 多目的フィールドの変化

複数の用途を持つ ACB フィールドが、ACBX の追加のフィールドに分割されているケースが多く見られます。

- ACB では、X'30' コール of データベース ID を格納するためにレスポンスコードフィールド (ACBRSP) が使用されます。その他の論理コールの場合、1 バイト of データベース ID は、ファイル番号フィールド (ACBFNR) の 1 バイト目に格納されていました。ACBX では、この目的のためにデータベース ID フィールド (ACBXDBID) が提供されています。
- ACB では、特定の Adabas レスポンスコード of エラー情報を保持するために ACBADD2 フィールドが使用されます。ACBX では、この目的でエラー情報フィールド (ACBXERR* の一連のフィールド) が用意されています。
- ACB では正常にコールを行うために、処理されたデータの圧縮および非圧縮レコード長を返すために ACBADD2 フィールドが使用されます。ACBX では正常にコールを行うために、圧縮レコードフィールド (ACBXLCMP) に、Adabas により処理された圧縮データの長さが含まれます。また非圧縮レコードフィールド (ACBXLDEC) には、非圧縮データの長さが含まれます。

構造およびオフセットの相違点

次の表に示すように、ACBXフィールドのオフセットおよびシーケンスは、対応するACBフィールドとは全般的に異なります。

オフセット	ACB DSECT フィールド名	ACBX DSECT フィールド名
00	ACBTYPE (コールタイプ)	ACBXTYPE (コールタイプ)
01	予約	ACBXRSV1 (予約 1)
02	ACBCMD (コマンドコード)	ACBXVER (ACBX バージョンインジケータ)
04	ACBCID (コマンド ID)	ACBXLEN (ACBX 長)
06	(ACBCID の続き)	ACBXCMD (コマンドコード)
08	ACBFNR (ファイル番号)	ACBXRSV2 (予約 2)
0A	ACBRSP (レスポンスコード -- X'30' コールでデータベース ID に使用される)	ACBXRSP (レスポンスコード)
0C	ACBISN (ISN)	ACBXCID (コマンド ID)
10	ACBISL (ISN 下限)	ACBXDBID (データベース ID)
14	ACBISQ (ISN 数)	ACBXFNR (ファイル番号)
18	ACBFBL (フォーマットバッファ長)	ACBXISNG (8 バイト ISN)
1A	ACBRBL (レコードバッファ長)	(ACBXISNG の続き)
1C	ACBSBL (サーチバッファ長)	ACBXISN (ISN -- ACBXISNG に含まれる)
1E	ACBVBL (バリュースバッファ長)	(ACBXISN および ACBXISNG の続き)
20	ACBIBL (ISN バッファ長)	ACBXISLG (8 バイト ISN 下限)
22	ACBCOP1 (コマンドオプション 1)	(ACBXISLG の続き)
23	ACBCOP2 (コマンドオプション 2)	(ACBXISLG の続き)
24	ACBADD1 (アディクション 1)	ACBXISL (ISN 下限 -- ACBXISLG に含まれる)
28	(ACBADD1 の続き)	ACBXISQG (8 バイト ISN 数)
2C	ACBADD2 (アディクション 2)	ACBXISQ (ISN 数 -- ACBXISQG に含まれる)
30	ACBADD3 (アディクション 3)	ACBXCOP1 (コマンドオプション 1)
31	(ACBADD3 の続き)	ACBXCOP2 (コマンドオプション 2)
32	(ACBADD3 の続き)	ACBXCOP3 (コマンドオプション 3)
33	(ACBADD3 の続き)	ACBXCOP4 (コマンドオプション 4)
34	(ACBADD3 の続き)	ACBXCOP5 (コマンドオプション 5)
35	(ACBADD3 の続き)	ACBXCOP6 (コマンドオプション 6)
36	(ACBADD3 の続き)	ACBXCOP7 (コマンドオプション 7)
37	(ACBADD3 の続き)	ACBXCOP8 (コマンドオプション 8)
38	ACBADD4 (アディクション 4)	ACBXADD1 (アディクション 1)
40	ACBADD5 (アディクション 5)	ACBXADD2 (アディクション 2)

オフセット	ACB DSECT フィールド名	ACBX DSECT フィールド名
44	(ACBADD5 の続き)	ACBXADD3 (アディクション 3)
48	ACBCMDT (コマンドタイム)	(ACBXADD3 の続き)
4C	ACBUSER (ユーザーエリア)	ACBXADD4 (アディクション 4)
54	---	ACBXADD5 (アディクション 5)
5C	---	ACBXADD6 (アディクション 6)
64	---	ACBXRSV3 (予約 3)
68	---	ACBXERRG (バッファ内のエラーオフセット、64 ビット -- 現在サポートされていません)
6C	---	ACBXERRA (バッファ内のエラーオフセット、32 ビット)
70	---	ACBXERRB (エラー文字フィールド)
72	---	ACBXERRC (エラーサブコード)
74	---	ACBXERRD (エラーバッファ ID)
75	---	ACBXERRE (エラーバッファシーケンス番号)
78	---	ACBXSUBR (サブコンポーネントレスポンスコード)
7A	---	ACBXSUBS (サブコンポーネントレスポンスサブコード)
7C	---	ACBXSUBT (サブコンポーネントエラーテキスト)
80	---	ACBXLCMP (圧縮レコード長)
88	---	ACBXLDEC (非圧縮レコード長)
90	---	ACBXCMDT (コマンドタイム)
98	---	ACBXUSER (ユーザーエリア)
A8	---	ACBXRSV4 (予約 4)

コントロールブロックのロギング

コマンドログ (CLOG) には、CLOGLAYOUT=5 と CLOGLAYOUT=8 の 2 つのフォーマットがあります。ADARUN パラメータ CLOGLAYOUT によって、使用するフォーマットを区別します。



Note: Adabas 8 までのバージョン向けのユーザープログラムには CLOGLAYOUT=5 を使用することをお勧めします。Adabas 8 以降向けのユーザープログラムには、CLOGLAYOUT=8 を使用してください。

Adabas ニュークリアス起動パラメータに CLOGLAYOUT=5 を指定すると、Adabas コントロールブロックフィールドがすべて基本コマンドログエリアに記録されます。CLOGLAYOUT=8 を使用すると、各バッファとも CLOGLAYOUT=5 の場合とほとんど同じように書き出されますが、各バッファの接頭辞として、対応する Adabas バッファ記述 (ABD) が付加される点が異なる

ります。各セグメントバッファ（フォーマット、レコード、またはマルチフェッチ）は別々に書き込まれ、一意に識別されます。

CLOGLAYOUT パラメータについては、「CLOGLAYOUT コマンドログレイアウト」を参照してください。この2つのコマンドログフォーマットの詳細については、「コマンドログフォーマット」を参照してください。

8 Adabas バッファ記述 (ABD)

▪ 使用可能な ABD タイプ	86
▪ ABD の構造	87
▪ ABD フィールドの説明	88
▪ ABD DSECT	91
▪ ABD リスト	91

ACBX インターフェイスを使用して、バッファ指定を必要とする Adabas コールを行う場合は、Adabas バッファ記述 (ABD) を使用する必要があります。従来の **ACB インターフェイス**を使用して Adabas コールを指定する場合は、ABD を使用しないでください。**ACB インターフェイス**を使用して、バッファ指定が必要な Adabas コールを行う場合は、Adabas コール自体に直接バッファまたはバッファへのポインタを指定します。ACBX と ACB のインターフェイスダイレクトコールの詳細については、「[Adabas の呼び出し](#)」を参照してください。

Adabas 8 では ACBX インターフェイスでセグメントバッファ (フォーマットバッファとレコードバッファの複数ペア、またはフォーマット、レコード、マルチフェッチの3つのバッファの複数セット) をサポートしているため、ACBX コールのバッファの総数は固定されておらず、制限もありません。個別のバッファは ACBX のフィールド自身により記述されなくなりました (ACB ではバッファ長は ACB で定義されます)。その代わりに、各バッファは独自の Adabas バッファ記述 (ABD) 構造を持ちます。ABD には、バッファの種類、場所、サイズ、およびその他の関連情報が記述されます。

メインフレームシステムアプリケーションでは、ABD のアドレスを直接 Adabas コールに指定します。オープンシステムアプリケーションでは、ABD のアドレスを Adabas コールに関連付けられた **ABD リスト**に指定します。

このchapterでは、ABD および ABD リストの構造について説明します。バッファ自体の定義については、「[バッファの定義](#)」を参照してください。

使用可能な ABD タイプ

ACBX インターフェイスダイレクトコールで ABD を使用すると、ダイレクトコールで連続したバッファおよび連続していないバッファを使用できます。Adabas 8 では、次の 8 種類のバッファについて ABD を定義できます。

- フォーマットバッファ
- レコードバッファ
- マルチフェッチバッファ
- サーチバッファ
- バリューストックバッファ
- ISN バッファ
- ユーザーバッファ
- パフォーマンスバッファ (Adabas Review 専用に予約済み)。

各 Adabas バッファセグメントは単一の ABD で表されますが、同一のプログラムで1つの種類に対して複数の ABD を定義できます。各 ABD のオフセット 4 (ABDID) は、ABD で定義されたバッファの種類を識別します。

ACBX インターフェイスコールでは、ABD とバッファ指定とが1対1で対応します。つまり、指定するバッファごとに対応する ABD が必要です。バッファは、ABD 自体に指定したり、間接参照で参照できます。

ABD は、**ACBX インターフェイスダイレクトコール**のどの場所にも指定できますが、異なるタイプの ABD 同士を対にする必要がある場合には、ABD の対が順番になるように指定します。例えば、コールの中にフォーマットバッファの ABD とレコードバッファの ABD がそれぞれ3つずつ存在する場合、最初のフォーマットバッファの ABD とレコードバッファ ABD が対になり、続いて2番目、3番目の ABD 同士が対になります。

指定した ABD の対の数が一致しない場合は、不足分のダミーの ABD (バッファ長はゼロ) が生成されます。例えば、フォーマットバッファの ABD を3つ指定し、レコードバッファの ABD を2つ指定した場合、3番目のフォーマットバッファの ABD と対になるように、ダミーのレコードバッファの ABD が1つ生成されます。ダミーのレコードバッファの ABD と2番目のフォーマットバッファの ABD を対にする必要がある場合は、3番目のフォーマットバッファと対になるレコードバッファの ABD の前にダミーのレコードバッファの ABD を指定する必要があります。

レコードバッファ内のデータを、フォーマットバッファ内のフォーマット指定では記述していないコマンドの場合は、フォーマットバッファセグメントの指定は必要ありません。指定した場合は無視されます。このような指定が可能なコマンドは、OP コマンドの他に数種類しかありません。

異なるバッファタイプ間の関係の詳細は、「**バッファタイプ間の関係**」を参照してください。

ABD の構造

次の表は、ABD の構造の説明です。

DSECT フィールド名	フィールド	コントロールブロックの位置	オフセット	長さ (バイト)	フォーマット
ABDXLEN	ABD の長さ	1~2	00	2	バイナリ
ABDXVER	バージョンインジケータ	3~4	02	2	バイナリ
ABDXID	バッファタイプ ID	5	04	1	英数字
ABDXRSV1	予約 1	6	05	1	バイナリ
ABDXLOC	バッファロケーションフラグ	7	06	1	英数字/バイナリ
ABDXRSV2	予約 2	8	07	1	バイナリ
ABDXRSV3	予約 3	9	08	4	バイナリ
ABDXRSV4	予約 4	13	0C	4	バイナリ

DSECT フィールド名	フィールド	コントロールブロックの位置	オフセット	長さ (バイト)	フォーマット
ABDXSIZE	バッファサイズ (割り当てられた長さ)	17~24	10	8	バイナリ
ABDXSEND	送信データ長	25~32	18	8	バイナリ
ABDXRECV	受信データ長	33~40	20	8	バイナリ
ABDXRSV5	予約 5	41~44	28	4	バイナリ
ABDXADR	間接アドレスポインタ (ABDLOC='C' の場合)	45~48	2C	4	英数字
---	バッファ (ABDLOC='C' または 'X'00' の場合)	49~n	30	ユーザー定義	該当なし

ABD フィールドの説明

ABD の各フィールドについては、この section で説明します。ABD 構造に表示される順に記載されています。

ABD 長 (ABDXLEN)

必須。このフィールドは、ABD の長さを指定するのに使用します。現時点では、このフィールドの値は 48 にする必要があります。

バージョンインジケータ (ABDXVER)

必須。このフィールドは、ABD 構造のバージョンを表します。このフィールドの値が 'C'G2' の場合、バッファ定義が新しい拡張 ABD 構造であることを示します。

バッファタイプ ID (ABDXID)

必須。このフィールドは、次の表に示すように、ABD で記述されるバッファのタイプを表すのに使用します。

ID 設定	バッファのタイプ
C'F'	フォーマット
C'T'	ISN
C'M'	マルチフェッチ
C'P'	パフォーマンス (Adabas Review 専用予約済み)
C'R'	レコード
C'S'	サーチ

ID 設定	バッファのタイプ
C'U'	ユーザー
C'V'	バリュー

予約 1 (ABDXRSV1)

このフィールドは予約フィールドで、バイナリのゼロに設定されている必要があります。

バッファロケーションフラグ (ABDXLOC)

必須。このフィールドは、バッファの位置を間接アドレスで定義するか、または ABD 自体の末尾に定義するかを表すのに使用します。このフィールドを "I" (CI) に設定した場合、間接アドレッシングが指定され、間接アドレスポインタフィールドに指定されているアドレス

(ABDADDR) を使用するものと見なされます。この場合、バッファはプライマリアドレススペース内の 31 ビットアドレス可能ストレージに存在する必要があります。

このフィールドを空白 (C') にするか、またはこのフィールドに 16 進数の 0 を含める場合は、バッファを ABD の直後に置く必要があります。

予約 2 (ABDXRSV2)

このフィールドは予約フィールドで、バイナリのゼロに設定されている必要があります。

予約 2 (ABDXRSV3)

このフィールドは予約フィールドで、バイナリのゼロに設定されている必要があります。

予約 2 (ABDXRSV4)

このフィールドは予約フィールドで、バイナリのゼロに設定されている必要があります。

バッファサイズ (ABDXSIZE)

必須。このフィールドは、割り当てられているとおりに、バッファのサイズ (バイト単位) を指定するのに使用します。サイズを 0 にすると、ダミーバッファとなり、バッファを指定しなかった場合と同じように扱われます。

送信データ長 (ABDXSEND)

必須。このフィールドは、Adabas に送信するデータの長さ (バイト単位) を指定するのに使用します。このフィールドには、バッファサイズフィールド (ABDSIZE) を超える値を設定することはできません。バッファが Adabas に送信されるのは、そのバッファが、発行されるコマンドのタイプに合った入力バッファである場合のみです。



Note: 現時点では、このフィールドには最大バッファサイズ (ABDXSIZE フィールド) に指定するのと同じ値を指定する必要があります。このことは Adabas 8 での一時的な制限であり、将来のリリースで解決される予定です。

受信データ長 (ABDXRECV)

このフィールドには、Adabas に返すデータの長さ (バイト単位) を指定します。Adabas ルーターは、コール処理の最後にこの値を設定します。このフィールドには、バッファサイズフィールド (ABDSIZE) を超える値を設定することはできません。Adabas からバッファを受信するのは、そのバッファが、発行されるコマンドのタイプに合った出力バッファである場合のみです。

予約 5 (ABDXRSV5)

このフィールドは予約フィールドで、バイナリのゼロに設定されている必要があります。

間接アドレスポインタ (ABDXADR)

バッファロケーションフラグフィールド (ABDLOC) を 'T' (間接バッファ) に設定した場合は、このフィールドに実際のバッファのアドレスを指定します。現在、32KB を超えるデータを Adabas バッファに指定できます。

実際のバッファ

バッファロケーションフラグフィールド (ABDLOC) を ' ' (空白) に設定した場合は、このフィールドに実際のバッファが含まれている必要があります。ACBX インターフェイスを使用すると、32KB を超えるデータを Adabas 8 バッファに指定できます。バッファ定義の詳細については、「[バッファの定義](#)」を参照してください。

ABD DSECT

ABD DSECT は、分散 Adabas SRCE ライブラリのメンバ ADABDX にあります。

ABD リスト

ABD リストは、ダイレクトコールに使用する Adabas バッファ記述 (ABD) へのポインタ参照が記載されているファイルです。ABD リストは、オープンシステム ACBX ダイレクトコールにのみ使用されます。ABD リストでは、ダイレクトコールに必要なバッファセグメントごとに ABD ポインタが 1 つ必要です。

ABD リストには、フォーマット、レコード、マルチフェッチ、サーチ、バリュー、ISN、ユーザー、パフォーマンスの計 8 種類のバッファに関して ABD へのポインタを含めることができます。ABD リストに同じタイプの ABD を複数指定できます。

リストに指定する ABD の順番に特に決まりはありません。異なるタイプの ABD 同士を対にする必要がある場合には、ABD の対が順番になるように指定します。例えば、3 つのフォーマット ABD と 3 つのレコード ABD をリストに記載した場合、リストの最初のフォーマット ABD はリストの最初のレコード ABD に対応し、2 番目のフォーマット ABD は 2 番目のレコード ABD に対応し、3 番目のフォーマット ABD は 3 番目のレコード ABD に対応します。リストで対応関係にある ABD の数が等しくない場合（例えば、フォーマット ABD が 3 つあるのにレコード ABD が 2 つしかない場合）、Adabas は欠如している ABD についてダミーの ABD を生成します（この例の場合、ダミーのレコード ABD が生成されます）。

それぞれのタイプの ABD またはバッファ間に関する関係については、「[バッファタイプ間の関係](#)」を参照してください。

9 バッファの定義

ダイレクトコールで **ACB** [ダイレクトコールインターフェイス](#)を使用する場合、フォーマット、レコード、サーチ、バリュー、ISN の計 5 種類のバッファを定義できます。これらのバッファはそれぞれアプリケーションの別の場所で指定され、ACB ダイレクトコールで（ポインタ参照により）間接的に参照されます。

Adabas 8 では、ダイレクトコールで **ACBX** [ダイレクトコールインタフェース](#)を使用する場合、**Adabas** [バッファ記述 \(ABD\)](#) を使用してフォーマット、レコード、マルチフェッチ、パフォーマンス、サーチ、バリュー、ISN、およびユーザーの計 8 種類のバッファセグメントとその関連するバッファ定義を定義できます。各 Adabas バッファセグメントは単一の ABD で表されますが、ABD のタイプによっては同じプログラム内に複数の ABD を定義できます（例えば、同じプログラムで使用する複数のフォーマット ABD を定義できます）。各 ABD には、単一のバッファ定義が関連付けられます。この場合、ポインタ参照によって間接的に関連付けられるか、または ABD 自体で直接関連付けられます。ABD の構造を含む、ABD の詳細については、「[Adabas バッファ記述 \(ABD\)](#)」を参照してください。

このchapterでは、次のトピックについて説明します。

バッファタイプ間の関係	さまざまなバッファタイプおよび各バッファ間の関係、および関連する ABD 間の関係について説明します (ACBX インターフェイス ダイレクトコールを行う場合)。
フォーマットバッファ	フォーマットバッファおよびその構文について説明します。
レコードバッファ	レコードバッファおよびその構文について説明します。
フォーマットバッファとレコードバッファの例	フォーマットバッファとレコードバッファのペアの例を示します。
マルチフェッチバッファ	マルチフェッチバッファおよびその構文について説明します。
サーチバッファ	サーチバッファおよびその構文について説明します。
バリューバッファ	バリューバッファおよびその構文について説明します。

バッファの定義

● サーチバッファとバリュースタックの例	サーチバッファとバリュースタックのペアの例を示します。
● ISN バッファ	ISN バッファおよびその構文について説明します。
● ユーザーバッファ	ユーザーバッファおよびその構文について説明します。
● パフォーマンスバッファ	パフォーマンスバッファについて説明します。

10 バッファタイプ間の関係

次の構文は、ダイレクトコールに指定できる各種バッファ間の関係を表しています。この構文は、どのバッファ指定が他のバッファ指定の存在に依存しているかを把握するのに役立ちます。



Notes:

1. **ACBX** インターフェイスダイレクトコールを指定している場合は、対応する **Adabas** バッファ記述 (**ABD**) も指定する必要があります。また、**ACBX** インターフェイスダイレクトコールで、あるバッファ指定に他のバッファ指定が必要になる場合（例えば、フォーマットバッファにはレコードバッファが必要です）、バッファは指定した順番でペアになります（例えば、最初に指定したフォーマットバッファ **ABD** は最初に指定したレコードバッファ **ABD** とペアになります）。次の構文は、コールまたは **ABD** リストに **ABD** をどの順番で指定するか判断するのに役立ちます。
2. **ACB** インターフェイスダイレクトコールを指定している場合、この構文のマルチフェッチ、パフォーマンス、ユーザーの各バッファは適用されません。また、フォーマット、レコード、サーチ、バリュー、**ISN** の順にバッファを指定する必要があります。この順番の中で前に位置するバッファが必要なく、後に位置するバッファが必要である場合、必要なバッファの前にあるバッファは、たとえ空白であっても、指定する必要があります。例えば、**ACB** インターフェイスダイレクトコールで **ISN** バッファが必要であり、それ以外のバッファは必要ない場合、**ISN** バッファの前にフォーマット、レコード、サーチ、バリューのダミーバッファをそれぞれ指定する必要があります。

```
[format-buffer record-buffer... [multifetch-buffer]]...  
[search-buffer value-buffer]  
[ISN-buffer]  
[user-buffer] ...  
[performance-buffer]
```

次の表では、この構文の要素について説明します。

エレメント	説明	条件
<i>format-buffer</i>	コールに使用するフォーマットバッファセグメントです。各フォーマットバッファセグメントは、ピリオドで終了し、単独のフォーマットバッファとして完結した有効なものである必要があります。	<p>Adabas読み込みコマンドまたは更新コマンドの実行中に処理するフィールドを指定する必要がある場合にのみ必須です。</p> <p>必須の場合、ACBXインターフェイスダイレクトコールに複数のフォーマットバッファを指定できます。ACBインターフェイスダイレクトコールには、1つのフォーマットバッファのみ指定できます。</p> <p>コールにフォーマットバッファを指定する場合は、対応するレコードバッファも指定する必要があります。ACBXインターフェイスダイレクトコールの場合、レコードバッファを指定しないと、フォーマットバッファのペアとなるダミーのレコードバッファ（長さゼロ）が作成されます。ACBインターフェイスダイレクトコールの場合、レコードバッファを指定しないと、処理エラーが発生します。</p> <p>ACBXインターフェイスダイレクトコールの場合は、対応するマルチフェッチバッファを指定することもできます。</p>
<i>ISN-buffer</i>	コールに使用する ISN バッファセグメントです。	<p>ストレージ内にISNを格納するためのエリアを確保する必要がある場合、または（ACBインターフェイスダイレクトコールでは）マルチフェッチまたはプリフェッチしたレコードのレコードディスクリプタエレメント（RDE）を格納するためのエリアを確保する必要がある場合にのみ必須です。</p> <p>必須の場合、コールにISNバッファを1つのみ指定します。</p>
<i>multifetch-buffer</i>	ACBXインターフェイス コールに使用する マルチフェッチバッファ セグメント。このバッファは、 ACBXインターフェイスダイレクトコール にのみ使用できます。	<p>ACBXインターフェイスダイレクトコールでのみ使用されます。また、マルチフェッチしたレコードのレコードディスクリプタエレメント（RDE）を格納するためのエリアをストレージ内に確保する必要がある場合にのみ必須です。</p> <p>必須の場合、ACBXインターフェイスダイレクトコールに複数のマルチフェッチバッファを指定できます。</p> <p>マルチフェッチバッファを指定した場合は、対応するフォーマットバッファおよびレコードバッファも指定する必要があります。対応するバッファを指定しないと、マルチフェッチバッファに対応するダミーのフォーマットバッファおよびレコードバッファ（長さゼロ）が作成されます。</p>

エレメント	説明	条件
<i>performance-buffer</i>	ACBX インターフェイスコールに使用するパフォーマンスバッファセグメント。このバッファは、 ACBX インターフェイスダイレクトコールでのみ使用でき、Adabas Review でのみ使用されます。	必須ではありません。Adabas Review による ACBX インターフェイスダイレクトコールでのみ使用されます。詳細については、Adabas Review のドキュメントを参照してください。
<i>record-buffer</i>	コールに使用する レコードバッファ セグメントです。	<p>コールに必要なレコードデータまたはコール用に収集したレコードデータを格納するためのエリアをストレージに確保する場合にのみ必須です。</p> <p>必須の場合、ACBX インターフェイスダイレクトコールに複数のレコードバッファを指定できます。ACB インターフェイスダイレクトコールには、1つのレコードバッファのみ指定できます。</p> <p>コールにレコードバッファを指定する場合は、対応するフォーマットバッファも指定する必要があります。ACBX インターフェイスダイレクトコールの場合、フォーマットバッファを指定しないと、レコードバッファのペアとなるダミーのフォーマットバッファ（長さゼロ）が作成されます。ACB インターフェイスダイレクトコールの場合、フォーマットバッファを指定しないと、処理エラーが発生します。</p> <p>ACBX インターフェイスダイレクトコールの場合は、対応するマルチフェッチバッファを指定することもできます。</p>
<i>search-buffer</i>	コールに使用する サーチバッファ セグメントです。	<p>コール対象レコードの選択時に検索条件が必要な場合にのみ必須です。</p> <p>コールにサーチバッファを指定する場合は、対応するバリュースearchバッファも指定する必要があります。単一のダイレクトコールに指定可能なサーチバッファとバリュースearchバッファのペアの数は1つだけです。</p>
<i>user-buffer</i>	コールに使用する ユーザーバッファ セグメント（拡張）です。ユーザーバッファ拡張（UBX）は、Adabas ニュークリアスユーザー出口11と4、および Adalink ユーザー出口1と2（Adabas 7のユーザー出口AとB）に渡されるユーザーデータに使用されます。	ACBX インターフェイスダイレクトコールでのみ使用されます。また、コールで Adabas ニュークリアスユーザー出口11と4、および Adalink ユーザー出口1と2（Adabas 7のユーザー出口AとB）への入力が必要になる場合にのみ必須です。単一のダイレクトコールに指定できるユーザーバッファは1つだけです。

バッファタイプ間の関係

エレメント	説明	条件
<code>value-buffer</code>	コールに使用するバリューバッファセグメントです。	<p>コール対象レコードの選択時に検索条件が必要な場合にのみ必須です。</p> <p>コールにバリューバッファを指定する場合は、対応するサーチバッファも指定する必要があります。単一のダイレクトコールに指定可能なサーチバッファとバリューバッファのペアの数は1つだけです。</p>

11 フォーマットバッファ

▪ フォーマットバッファ構文	100
▪ フィールドの選択条件	101
▪ レコードフォーマットの指定	102
▪ フォーマットバッファでの LA（ロング英数字）フィールドのフィールド長の指定	123
▪ フォーマットバッファのパフォーマンスに関する考慮事項	123

フォーマットバッファには、Adabas 読み込みコマンドまたは更新コマンドの実行中に処理するフィールドを指定します。フォーマットバッファには、バッファ自体の末尾の (.) も含め、関連プログラムに含まれている最長のフィールド定義を保持できるだけの十分な長さが必要です。

フォーマットバッファはレコードバッファとペアで使用され、さらに **ACBX インターフェイスダイレクトコール** ではマルチフェッチバッファともペアで使用されます。フォーマットバッファを指定する場合は、レコードバッファも指定する必要があります。レコードバッファを指定しないと、フォーマットバッファのペアとなるダミーのレコードバッファ（長さゼロ）が作成されます。

ACBX ダイレクトコールインタフェースを使用する場合、Adabas ダイレクトコールに複数のフォーマットバッファを指定できます。

ACBX インターフェイスダイレクトコール では、マルチフェッチバッファセグメントも指定できます。詳細については、「**レコードバッファ**」および「**マルチフェッチバッファ**」を参照してください。

それぞれのタイプの ABD またはバッファ間に関する関係については、「**バッファタイプ間の関係**」を参照してください。

このchapterでは、次のトピックについて説明します。

フォーマットバッファ構文

フォーマットバッファは2つの部分に分かれています。バッファのオプションフィールド選択条件と、想定されるレコードフォーマットです。

多様なレコードフォーマットをもつファイルに対しては、フォーマットバッファに複数のレコードフォーマットを指定できます。どのレコードフォーマットを使用するかは、ファイルのフィールド値によって決まります（例えば、レコードタイプ）。複数のレコードフォーマットを指定する場合、各レコードフォーマットの前にフォーマット選択条件を指定する必要があります。

フォーマットバッファの構文は次のとおりです。

```
{[field-selection-criteria1] record-format1}{,[field-selection-criteria2] record-format2}]...
```

各フォーマットバッファエントリは、コンマ (,) で区切る必要があります。各エントリ間に1つ以上の空白があってもかまいません。最後のエントリの後には、コンマを付けないでください。

フォーマットバッファはピリオドで終了する必要があります。

次のsectionでは、この構文のコンポーネントについて説明します。

■ フィールドの選択条件

■ レコードフォーマットの指定

フィールドの選択条件

フィールド選択条件は、フォーマットバッファではオプションです。フィールド選択条件を使用すると、レコードフォーマットをフィールドの特定の値に制限できます。フィールド選択条件の構文は次のとおりです。

```
(field-name operator value1 [, value2]...)
```

field-name

フィールド選択条件に使用するフィールド名は、読み込まれる Adabas ファイルの FDT にあるフィールドの有効な名前である必要があります。次の値を指定することはできません。

- グループまたはピリオディックグループの名前
- MU、PE、LA、LB のいずれかのオプションを使用しているフィールド
- サブフィールド、スーパーフィールド、サブディスクリプタ、またはスーパーディスクリプタ
- 照合ディスクリプタ
- ハイパーディスクリプタ
- フォネティックディスクリプタ

また、NU または NC/NN オプションで指定したフィールドは空値以外の値である必要があります。空値であると、選択条件が偽になります。

operator

次の表は、フォーマットバッファフィールド選択条件に使用できる演算子のリストです。

演算子	説明
EQ	等しい
=	等しい
NE	等しくない
LT	より小さい
<	より小さい
GT	より大きい
>	より大きい
LE	より小さいまたは等しい
GE	より大きいまたは等しい

value

値は、整数または英数字にする必要があります。

EQまたは=演算子を使用する場合は、一連の値をコンマで区切って指定できます。英数字は引用符で囲みます（例えば 'value'）。

次に例を示します。

```
(SA = 1) record-format-1, (SA = 2,3,4) record-format-2, (SA GE 4) record-format-3.
```

フィールド選択条件は次のとおりです。

- SA フィールドの値が "1" の場合、レコードフォーマット 1 が使用されます。
- SA フィールドの値が "2"、"3"、または "4" の場合、レコードフォーマット 2 が使用されます。
- SA フィールドの値が "4" 以上の場合、レコードフォーマット 3 が使用されます。

最初に一致した条件が使用されます。いずれの条件も一致しない場合、レスポンスコードが返されます。上記の例では、SA フィールドの値が "4" の場合、フィールド選択条件の最後の2つが一致します。ただし、レコードフォーマット 2 が最初に一致する条件であるため、レコードフォーマット 3 ではなくレコードフォーマット 2 が常に使用されます。同じく、SA フィールドの値が "0" の場合、レスポンスコードが返されます。

レコードフォーマットの指定

レコードフォーマットは必須であり、Adabas ダイレクトコールでどのフィールドを読み込むか、または更新するかを示すのに使用します。

レコードフォーマットの構文は次のとおりです。

```
{ field [, length ] [, data-format ]  
  field-name - field-name  
  nX  
  'text' }
```

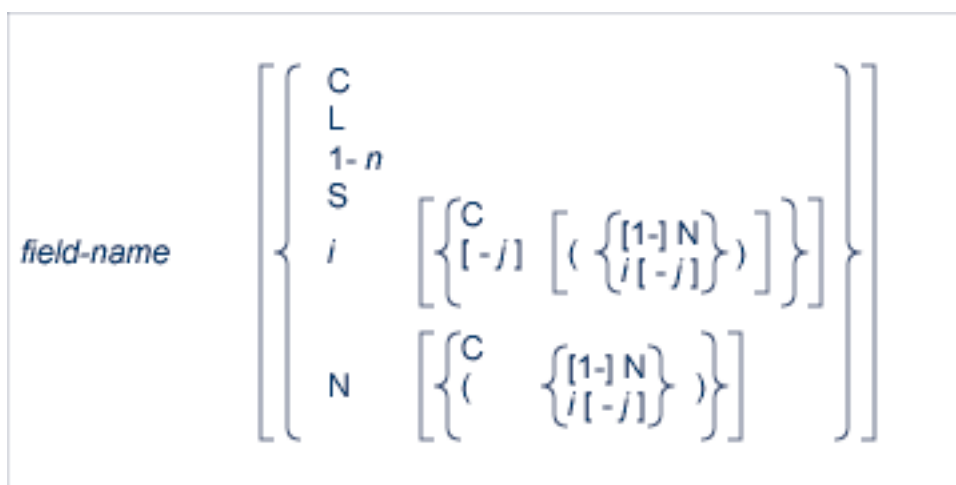
レコードフォーマットの構文

この構文の各エレメントについては、次の表に示すセクション（このsection内）を参照してください。

構文要素	参照先
<i>field</i>	field 構文
<i>length</i>	長さおよびデータフォーマット
<i>data-format</i>	長さおよびデータフォーマット
<i>field-name - field-name</i>	フィールドの連続表記
<i>nX</i>	スペース表記 (<i>nX</i>)
' <i>text</i> '	テキスト挿入表記

field 構文

レコードフォーマット構文の *field* の構文は次のとおりです。



フィールド構文

このsectionでは、このフィールド構文に関連する次のトピックを取り上げています。

- *field-name* 指定
- インデックスまたは範囲の表記 (*i[-j]* 表記)
- プリオディックグループ参照
- マルチプルバリューフィールド
- プリオディックグループ内のマルチプルバリューフィールド
- カウントインジケータ (C)
- 長さインジケータ (L)
- 最高オカレンス／値インジケータ (N)

■ SQL 有意性インジケータ (S)

field-name 指定

field-name は値、範囲、またはカウントを要求したり、新しい値を提供したりするフィールドまたはグループの名前です。指定した名前は2文字で、Adabas ダイレクトコールで読み込むかまたは更新されるファイルの FDT に存在している必要があります。この名前でエレメンタリフィールド、サブフィールド、スーパーフィールド、マルチプルバリュフィールド、グループ、またはピリオディックグループを参照できます。

フィールド名でグループを参照すると、グループ内の全フィールドが参照されます。グループ名を使用すると、コマンド処理に必要な時間を短縮できます。グループ名は、グループにマルチプルバリュフィールドまたは可変長フィールド（標準長なし）が含まれている場合には指定できません。

読み込みコマンドで同一名を複数指定すると、フィールド値が複数回返されます。

更新コマンドでは同一名を複数指定できません（ただし以降で説明するようにマルチプルバリュフィールドの場合は除く）。

サブフィールド、スーパーフィールド、サブディスクリプタ、またはスーパーディスクリプタの名前は読み込みコマンドには指定できますが、更新コマンドには指定できません。

次の表は、指定可能な単一値のフィールドタイプとコマンドタイプとの関係を示しています。

コマンドタイプ	フィールド	サブフィールド	スーパーフィールド	DE	SUBDE	SUPERDE	COLDE
読み込み (Ln) ¹	○	○	○	○	○	○	○ ³
追加 (Nn)	○	×	×	○	×	×	×
検索 (Sn)	○	○	○	○	○	○	○
更新 (A1/4) ²	○	×	×	○	×	×	×

 **Notes:**

1. 圧縮形式のレコードを要求する読み込みコマンドとともに C フォーマットを使用できます。
2. 更新コマンドに指定したフィールドは繰り返すことができません。また、フィールドには "1-N" 形式の指定を含めることができず、グループに含まれているグループおよびエレメント（フィールドまたはグループ）を指定できません。
3. 照合ディスクリプタ (COLDE) は、ユーザー出口でデコードオプションが指定されている場合に限り、L9 コマンドのフォーマットバッファに指定できます。この場合、インデックス値ではなく、元のフィールド値が返されます。

インデックスまたは範囲の表記 (*i*[-*j*] 表記)

以下は、マルチプルバリュースフィールドまたはピリオディックグループのオカレンスを選択するフィールド名形式です。

```
field-name i[-j]
```

上記の意味は次に示すとおりです。

<i>i</i>	ピリオディックグループまたはマルチプルバリュースのオカレンス。
<i>i-j</i>	ピリオディックグループまたはマルチプルバリュースのオカレンス範囲。

ピリオディックグループ名には、数字など適切な接尾辞を付ける必要があります（詳細については、「[カウントインジケータ \(C\)](#)」および「[最高オカレンス/値インジケータ \(N\)](#)」参照）。ピリオディックグループ名をフィールド名のみで指定することはできません。


マルチプルバリュースフィールドを指定するには、特定の値（インデックス）を明示的に指定するか、または各値を順番に参照して Adabas がその順番に基づいてインデックスを割り当てることができるようにします。詳細については、「[マルチプルバリュースフィールド](#)」を参照してください。

例	選択される内容
ZZ3	マルチプルバリュースフィールド ZZ の 3 番目の値。ピリオディックグループ ZZ の第 3 オカレンスの値。ピリオディックグループの第 3 のオカレンス内に存在する（単一またはマルチプルバリュースフィールド）ZZ の最初のオカレンスの値。
ZZ3-6	マルチプルバリュースフィールド ZZ の 3~6 番目の値。ピリオディックグループ ZZ の第 3~6 のオカレンスの値。ピリオディックグループの第 3~6 のオカレンス内に存在する（単一またはマルチプルバリュース）フィールド ZZ の最初のオカレンスの値。

ピリオディックグループ参照

ピリオディックグループ（またはピリオディックグループ内のフィールド）を参照する場合、どのオカレンスを参照するかを指定する必要があります。これは、1~3 桁のインデックス（値 1~191、先行のゼロ可）を名前に付けて表示します。

以下に例を示します。

 **Note:** この section で説明する例では、「[例で使用されているファイル定義](#)」に記載した Adabas ファイル定義を引用しています。

1. ピリオディックグループのオカレンスはすべて、グループ名とオカレンス番号で識別できません。

例	選択される内容
GB3	ピリオディックグループ GB の 3 番目のオカレンス（フィールド BA3、BB3、BC3）。

2. 1つ以上のピリオディックグループ内のフィールドを選択すると、フィールド名とオカレンス（値）が使用されます。グループ名は暗黙のうちに使用されます。

例	選択される内容
BB06	フィールド BB の 6 番目のオカレンス。

3. ピリオディックグループ（またはピリオディックグループ内のフィールド）のある範囲のオカレンスを参照するには、ピリオディックグループまたはフィールド名を入力し、その後に最初と最後のオカレンス番号をハイフンでつないで入力します。オカレンス番号は昇順にする必要があります。降順の範囲は許可されていません。

例	選択される内容
GB2-4	ピリオディックグループ GB の第 2 から第 4 オカレンスおよびこれらのオカレンス内のすべてのフィールドを参照します（BA2、BB2、BC2、BA3、BB3、BC3、BA4、BB4、BC4）。
BA2-4,BC2-4	BA と BC の第 2 から第 4 オカレンスを参照します（BA2、BA3、BA4、BC2、BC3、BC4）。

マルチプルバリュースフィールド

フォーマットバッファにマルチプルバリュースフィールドを指定するには、特定の値（インデックス）を明示的に指定するか、または各値を順番に参照して Adabas がその順番に基づいてインデックスを割り当てることができるようにします。この 2 つの方法は、マルチプルバリュースフィールドから得られたサブディスクリプタまたはスーパーディスクリプタにも適用されます。



Note: この section で説明する例では、「[例で使用されているファイル定義](#)」に記載した Adabas ファイル定義を引用しています。

1. インデックス：マルチプルバリュースフィールドの特定の値を参照するには、フィールド名の後に1～3桁のインデックス（値1～191、先行のゼロ可）を入力します。

例	選択される内容
MF2	マルチプルバリュースフィールド MF の 2 番目の値を参照します。
MF1,MF10	マルチプルバリュースフィールド MF の 1 番目の値と 10 番目の値を参照します。

マルチプルバリュースフィールドのある範囲の値を参照するには、最初と最後の値をハイフンでつなぎ、フィールド名の直後に指定します。降順範囲は指定できません。

例	選択される内容
MF1-3	マルチプルバリュースフィールド MF の 1 番目から 3 番目までの値を参照します。

2. 順番：フィールド名を繰り返してマルチプルバリュースフィールドを参照するには、まず最初の値を指定し、次に2番目の値を指定します。以後、同じように指定します。

例	選択される内容
MF,MF	マルチプルバリュースフィールド MF の 1 番目の値と 10 番目の値を参照します。
AA,MF,AB,MF,AC,MF	フィールド AA、AB、AC とともにマルチプルバリュースフィールド MF の 1 番目から 3 番目の値が参照されます。

マルチプルバリュースフィールド値を参照するのに、同一フォーマットバッファで上記2つの方法を両方とも使用できます。オカレンスインデックスを指定しない場合は、最終オカレンスより1つ大きく、左から右に進むオカレンスが割り当てられます。最後のインデックスとして、「N」または「N-1」が指定されており、このインデックスが現在の最高オカレンスを参照している場合、インデックスを明示的に指定しない新しいフィールド指定では、最後の指定として同一オカレンスを参照します。最高オカレンス／値インジケータ N/1-N/NC の詳細については、「[最高オカレンス／値インジケータ \(N\)](#)」を参照してください。

更新コマンドで、マルチプルバリュースフィールドへのすべてのフォーマットバッファ参照をインデックスではなく順番で指定した場合、指定したオカレンスのみがレコードに残り、それ以外のオカレンスは削除されます。マルチプルバリュースフィールドへのいずれかの参照をインデックスで指定した場合は、指定したオカレンスのみを変更され、それ以外のオカレンスは変わりません。

ピリオディックグループ内のマルチプルバリューフィールド


マルチプルバリューフィールドがピリオディックグループ内に存在している場合、マルチプルバリューフィールド名を指定すると、ピリオディックグループ名が暗黙的に使用されます。このため、グループ名を指定する必要はなく、グループオカレンスまたはオカレンス範囲のみを指定できます。1つのピリオディックグループオカレンスまたはオカレンス範囲内のマルチプルバリューフィールドまたはフィールド範囲を選択するための一般形式は次のとおりです。

```
field-name i[-j] (i[- j])
```

上記の意味は次に示すとおりです。

<i>i</i>	ピリオディックグループのオカレンス
<i>i-j</i>	ピリオディックグループのオカレンス範囲
(<i>i</i>)	マルチプルバリューフィールド値
(<i>i-j</i>)	マルチプルバリューフィールド値の範囲

以下に例を示します。

 **Note:** このsectionで説明する例では、「[例で使用されているファイル定義](#)」に記載した Adabas ファイル定義を引用しています。

1. ピリオディックグループ内のマルチプルバリューフィールドを参照する場合、フィールド名の後にピリオディックグループのオカレンス番号を入力し、その後求めるマルチプルバリューフィールド値または値の範囲をカッコで囲んで指定します。

例	選択される内容
CB2(5)	フィールド CB を含むピリオディックグループの第 2 オカレンス内のマルチプルバリューフィールド CB の 5 番目の値を参照します。
CB2(1-5)	フィールド CB を含むピリオディックグループの第 2 オカレンス内のマルチプルバリューフィールド CB の 1 番目から 5 番目の値を参照します。

2. ある範囲のピリオディックグループオカレンス内の同じマルチプルバリューフィールド、または同じ範囲のマルチプルバリューフィールドを参照する場合、フィールド名の後にピリオディックグループオカレンスの範囲を入力し、その後マルチプルバリューフィールド値の範囲をカッコで囲んで指定します。

例	選択される内容
CB3-5(2)	フィールド CB を含むピリオディックグループの第3から第5までの各オカレンスに存在するマルチプルバリュースフィールド CB の2番目の値を選択します。
CB1-2(1-4)	フィールド CB を含むピリオディックグループの第1、第2オカレンス内のマルチプルバリュースフィールド CB の1番目から4番目の値を参照します。

カウントインジケータ (C)

ピリオディックグループのオカレンス数、またはピリオディックグループに含まれていないマルチプルバリュースフィールドの値の個数は、ピリオディックグループまたはマルチプルバリュースフィールド名に文字 C を付けて参照できます。

`field-name C count for multiple-value field or periodic group "field-name"`

例	選択される内容
GBC	ピリオディックグループ GB の最高オカレンスカウントを参照します。
MFC	ピリオディックグループに含まれないマルチプルバリュースフィールド MF に存在する値の個数を参照します。

個数は、ユーザーが特に長さやフォーマットを指定（「長さおよびデータフォーマット」を参照）しなければ、1バイトのバイナリ数としてレコードバッファに返されます。

Adabas ファイルで Adabas 8 拡張 MU/PE の制限を使用する場合は、2バイトのオカレンスカウント値（例えば、FB='MUC,2,B'）を処理できるように、プログラムが2バイト以上のレコードバッファフィールドにオカレンスカウントを読み込むことを確認してください。拡張 MU/PE が制限されたファイルの MU フィールドまたは PE グループのオカレンスカウント用のレコードバッファに1バイトフィールドのみを指定した場合、Adabas はレスポンスコード 55、サブコード 9 を返します。

ピリオディックグループ内のマルチプルバリュースフィールドに値がいくつあるか確認するには、マルチプルバリュースフィールド名の直後にピリオディックグループオカレンスと文字 C を入力します。

`field-name i C count for multiple-value field "field-name" within a group occurrence`

例	選択される内容
CB4C	ピリオディックグループの第4オカレンス内マルチプルバリュースフィールド CB の値の個数を参照します。


更新コマンドに対して C 指定をすると、カウントは無視され、カウントフィールドがレコードバッファに占める桁数がスキップされます。

ユーザーは、マルチプルバリュースフィールドまたはピリオディックグループのカウントフィールドを直接更新できません。これらのカウントフィールドは、マルチプルバリュースフィールド値

およびピリオディックグループオカレンスを追加または削除するときに、Adabasによって自動的に更新されます。

長さインジケータ (L)

フォーマットバッファインジケータ *L* を使用すると、LA または LB 英数字フィールド値またはワイド文字フィールド値の実際の長さを取得または指定できます。このフォーマットバッファエレメントは、長さインジケータと呼ばれます。

 **Note:** 現在、長さインジケータは、LA または LB フィールドのフォーマットバッファ指定にのみ使用できます。他のフィールドにおける長さインジケータの使用のサポートは、Adabas の今後のリリースで導入される予定です。

長さインジケータは、フィールド名に続いて文字 *L* を使用して指定します。例えば、`FB='ACL,4,B.'` と指定すると、AC フィールドの長さが返されます。フィールドがマルチプルバリュールフィールドであるか、ピリオディックグループ内にある場合は、フィールド名および文字 *L* の後に関連するオカレンスインデックスが続きます。例えば、`FB='ACL2,4,B.'` と指定すると、マルチプルバリュールフィールド AC の 2 番目の値が返されます。圧縮フィールド長は、4 バイトのバイナリ形式で返されます。他の長さおよび形式は指定できません。

- [MU/PE フィールドにおける長さインジケータの使用](#)
- [読み込みコマンドにおける長さインジケータの使用](#)
- [更新コマンドにおける長さインジケータの使用](#)

MU/PE フィールドにおける長さインジケータの使用

長さインジケータを MU または PE フィールドで使用する場合、オカレンスインデックスを指定する必要があります。オカレンスインデックスの範囲を指定することもできます。ただし、1-N オカレンスインデックスは指定できません。以下に例を示します。

1. 次の例では、ピリオディックグループフィールド AC の 2 番目のオカレンスの 5 個目の値の長さが返されます。

```
FB='ACL2(5).'
```

2. 次の例では、マルチプルバリュールフィールド AC の最初の 10 個の値の長さが返されます。

```
FB='ACL1-10.'
```

3. 1-N 表記は MU または PE フィールドの長さインジケータでは許可されていないため、次の例は正しくありません。

```
FB= 'ACL1-N.'
```

4. 長さインジケータはオカレンスインデックスを持たない MU フィールドをサポートしないため、次の例も正しくありません。

```
FB= 'MCL.'
```

また、MU または PE フィールドの同一のフォーマットバッファ内で、長さインジケータと [アスタリスク長さ表記](#) による値の要求を組み合わせる場合、値の要求には長さの要求に対応する範囲を使用する必要があります。長さの要求と値の要求は同一のフォーマットバッファセグメントでも、異なるフォーマットバッファセグメントのどちらかで指定しても構いません。例として、XX が、MU オプション付きの LA または LB フィールドである場合を以下に示します。

1. 次の有効な例では、XX フィールドの最初の 2 個の値の長さを実際の値を要求しています。

```
FB= 'XXL1-2,XX1-2,*.'
```

```
FB= 'XXL1,XXL2,XX1,* ,XX2,*.'
```

2. 次の無効な例では、XX フィールドの最初の 2 個の値の長さを実際の値を要求しようとしています。ただしこの例では、MU フィールドに対して長さの要求に指定した範囲と、値の要求に指定した範囲が、対応する形式で指定されて異なるため、無効となります。

```
FB= 'XXL1,XXL2,XX1-2,*.'
```

```
FB= 'XXL1-2,XX1,* ,XX2,*.'
```

3. 次の 2 つのフォーマットバッファは、XX フィールドの 3 番目および 4 番目の値の長さとその実際の値を要求しています。

```
FB= 'XXL3,XXL4.'
```

```
FB= 'XX3,* ,XX4,*.'
```

4. 次の無効なフォーマットバッファは、XX フィールドの3番目および4番目の値の長さとその実際の値を要求していますが、長さの要求に指定した範囲と、対応する形式で指定されていないため、エラーになります。

```
FB='XXL3,XXL4.'
```

```
FB='XX3-4,*.'
```

読み込みコマンドにおける長さインジケータの使用

読み込みコマンドのフォーマットバッファ内のフィールドに長さインジケータを指定する場合、レコードバッファにフィールド値を格納するために必要なバイト数（パディングがなく、長さに関する情報を含まないバイト数）が、レコードバッファ内の対応するフィールド位置に返されます。レコードバッファに必要なスペースは、フィールドフォーマット、データベース、ファイルおよびユーザーの UES に関連する定義により異なります。

同一のフォーマットバッファ内で、LA または LB フィールドの長さを取得する長さインジケータを指定する際に、そのフィールドの基本フォーマットと異なる（変換された）フォーマットで、実際のフィールド値を要求することはできません。例えば、文字の LB フィールド L1 がフォーマット A で格納されている場合、フォーマットバッファ指定 `FB='L1L,4,B,L1,* ,W.'` は、L1 フィールドの長さの要求に加え、Unicode（フォーマット W）に変換された L1 フィールドの値を要求しているため、正しくありません。長さ要素はフィールドの長さをフィールド固有のフォーマットで指定しますが、要求されたフォーマット（Unicode）で返される値の長さが変換により異なるため、このような制限があります。

文字 LB フィールド L1（フォーマット A）が 40,000 バイトの EBCDIC 値を含んでいる場合を次の例に示します。

1. L1 のフォーマットバッファ指定が次のようであると仮定します。

```
FB='L1L,4,B.'
```

レコードバッファには、4 バイトバイナリ長の L1 フィールドの値が含まれます。

```
X'00009C40'
```


2. L1 のフォーマットバッファ指定が次のようであると仮定します。

```
FB='L1L,4,B,L1,* ,A.'
```

レコードバッファには4バイトバイナリ長のL1フィールドの値がレコードバッファエリアの先頭に含まれ、次に、40,000文字の実際のL1データが含まれます。

読み込み操作、長さインジケータ、および **NB** (空白圧縮なし) オプション

フォーマットバッファ内のフィールドが、対応する長さインジケータを使用して指定されていて (例えば `FB='L1L,4,B,L1.'` など)、フィールドが空白圧縮の対象外 (**NB** オプションが **FDT** 内のフィールドに指定されている) の場合、返される長さは、値が格納されたときに指定されたバイト数となります。ただし、フィールドが空白圧縮の対象となっている場合、返される長さは左端から意味を持つバイト数のみが返され、それ以上の値は空白で埋められます。

読み込み操作、長さインジケータ、および **NU** (空値省略) オプション

フォーマットバッファ内のフィールドが、対応する長さインジケータを使用して指定されていて (例えば `FB='L1L,4,B,L1.'` など)、フィールドが空値省略 (**FDT** 内でこのフィールドに **NB** オプションが指定されている) でフィールド値が完全に空白の場合、返されるフィールド値の長さはゼロになります。フィールドが空値省略ではない場合、返されるフィールド値の長さは、1つの空白の長さ (英数字フィールドで1バイト、ワイド文字フィールドで2バイト) になります。

更新コマンドにおける長さインジケータの使用

更新コマンドのフォーマットバッファに長さインジケータが指定されている場合、レコードバッファの対応する値がレコードバッファ内のフィールドの実際の値の長さを表します。基本フィールドの長さインジケータ1個のみが指定でき、このインジケータはフォーマットバッファで **アスタリスク (*) 表記** を伴う必要があります。

長さインジケータは、レコードバッファに可変長を指定するなどのフォーマット要素 (アスタリスク表記または長さゼロ表記による) より先にフォーマットバッファセグメント内で出現する必要があります。つまり、長さインジケータの場所は常に同じで、このフォーマットで指定されるどのフィールドの値からも影響を受けません。

最高オカレンス/値インジケータ (N)

文字 **N** を使用すると、最高値または最高オカレンスの番号がわからなくても、マルチプルバリュフィールドを構成している複数の値から最高値を選択したり、ピリオディックグループの最高オカレンスを選択できます。



Note: 1-N 表記は、LB フィールドではサポートされません。

1-N を使用すると、マルチプルバリュフィールドを構成している値をすべて選択したり、ピリオディックグループのオカレンスをすべて選択できます。ピリオディックグループ内のマルチ

プルバリュースフィールドの場合、グループオカレンスを示す 1-N という指定をフィールドオカレンスの指定を組み合わせることはできません。

NCは、そのフィールドを含むピリオディックグループの最高オカレンス内の現存するマルチプルバリュースフィールドの値の数を選択します。

<i>field-name</i> N	ピリオディックグループ <i>field-name</i> の最高オカレンスまたはマルチプルバリュースフィールド <i>field-name</i> の最高値
<i>field-name</i> NC	<i>field-name</i> を含むピリオディックグループの最高オカレンスにあるマルチプルバリュースフィールド <i>field-name</i> の値の数
<i>field-name</i> NC	マルチプルバリュースフィールド <i>field-name</i> の値をすべて選択するか、またはピリオディックグループ <i>field-name</i> のオカレンスをすべて選択します。
<i>field-name</i> N (N)	<i>field-name</i> を含むピリオディックグループの最高オカレンスにあるマルチプルバリュースフィールド <i>field-name</i> の最高値
<i>field-name</i> N(1-N)	<i>field-name</i> を含むピリオディックグループの最高オカレンスにあるマルチプルバリュースフィールド <i>field-name</i> のすべての値
<i>field-name</i> N(<i>i</i> [- <i>j</i>])	<p><i>field-name</i> を含むピリオディックグループの最高オカレンスにあるマルチプルバリュースフィールド <i>field-name</i> の値または値の範囲</p> <p>マルチプルバリュース LB フィールドおよびピリオディックグループ内の LB フィールドでは、フィールドの特定のオカレンスを指定する必要があります。次の例では、マルチプルバリュース LB フィールド L2 の最初の値が選択されます。</p> <pre>FB= 'L21.'</pre> <p>オカレンスインデックスのない基本フィールドを指定することはできません。例えば、次のフォーマットバッファ指定は有効ではありません。</p> <pre>FB= 'L2.'</pre>
<i>field-name</i> <i>i</i> [- <i>j</i>] (N)	<p><i>field-name</i> を含むピリオディックグループのオカレンスまたはオカレンスの範囲内にあるマルチプルバリュースフィールド <i>field-name</i> の最高値</p> <p>マルチプルバリュース LB フィールドおよびピリオディックグループ内の LB フィールドでは、フィールドの特定のオカレンスを指定する必要があります。次の例の場合、2 番目の PE グループインスタンスにある LB フィールド L3 の 5 番目の値が選択されます。</p>

	FB='L32(5).'
<i>field-name</i> <i>i</i> [- <i>j</i>] (1-N)	<i>field-name</i> を含むピリオディックグループのオカレンスまたはオカレンスの範囲内にあるマルチプルバリュースフィールド <i>field-name</i> の全値。

例	選択される内容
MFN	マルチプルバリュースフィールド MF の最高（最後）の値。マルチプルバリュースフィールド MF に 4 つの値がある場合は、4 番目の（最後に入力された）値が選択されます。
GBN	ピリオディックグループ GB の最高オカレンス番号。
MFNC	MF を含むピリオディックグループの最高オカレンスにあるマルチプルバリュースフィールド MF の値の個数。
MF1-N	マルチプルバリュースフィールド MF の全値。
GB1-N	ピリオディックグループ GB の全オカレンス。
MFN(1-N)	ピリオディックグループの最高オカレンスにあるマルチプルバリュースフィールド MF の全値。
MFN(N)	MF を含むピリオディックグループの最高オカレンスにあるマルチプルバリュースフィールド MF の最高値。
MFN(4)	MF を含むピリオディックグループの最高オカレンスにあるマルチプルバリュースフィールド MF の 4 番目の値。
CB3-5(N)	CB を含むピリオディックグループの第 3 から第 5 のオカレンスにあるマルチプルバリュースフィールド CB の最高値。
CB2(1-N)	CB を含むピリオディックグループの第 2 のオカレンスにあるマルチプルバリュースフィールド CB の全値。

SQL 有意性インジケータ (S)

S または空値インジケータ、およびレコードバッファ内の空値インジケータ値は、フィールドの値（ゼロまたは空白を含む）が有効であるか、または有効でない（未定義）かを示します。S インジケータは、NC オプションで定義されたエレメンタリフィールドにのみ適用でき、NU オプション付きフィールドには適用できません。

`field-name S SQL significance indicator`

空値インジケータの説明については、「[レコードバッファ](#)」セクションを参照してください。また、SQL 空値フィールドの定義方法については、『*Adabas ユーティリティマニュアル*』の ADACMP ユーティリティの説明を参照してください。

以下に示したレコードバッファの空値インジケータ値は、長さが 2 バイトの標準長で、フォーマットが固定小数点です。この長さとはフォーマットは変更しないでください。

更新タイプのコマンドの場合、対応する空値インジケータ値が X'FFFF' で、かつフィールドに対して NN オプションが指定されていないと、空値がフィールドに格納されます。それ以外の場合、空値インジケータは X'0000' である必要があります。この値は、S インジケータと組み合わせることによって、レコードバッファ内の他の場所に格納されたフィールド値が格納されることを示します。

S インジケータは更新タイプのコマンドには不要です。S インジケータが指定されていない場合にも、S インジケータが指定され、対応する空値インジケータ値がゼロに設定されているときと同じように、フィールド値を更新します（フィールド内の空値は値である）。次の例を参照してください。

読み込みタイプのコマンドでは、NN オプションなしで NC フィールドが定義されたときに、S インジケータが必要です。READ コマンドが NC 指定されたフィールドを検出し、そのフィールドに実際に空値が含まれていた場合、S インジケータがなければレスポンスコード 55 が返されます。

例

この例では、"フィールド名 S" インジケータをレコードバッファ内の 2 バイトの空値インジケータとともに使用してレコードを更新または追加し、フィールド AA を SQL 空値に設定し、フィールド AA の値を無視します。

フォーマットバッファ	AAS , AA , 2 ,	読み込むフィールドの名前
レコードバッファ	FFFF 123F	Adabas から返されるフィールド値

NC フィールドを含むグループまたはフィールド範囲を指定する場合の SQL 有意性（シグニフィカンス）インジケータの使用例については、「[有意性（シグニフィカンス）インジケータおよび連続フィールドの指定](#)」を参照してください。

"フィールド名 S" インジケータは、フォーマットバッファ内の任意の位置に指定できます。つまり、対応するフィールドエレメントより先に置く必要はありません。更新タイプのコマンドの場合、フォーマットバッファに同じフィールドを参照するエレメントを複数含めることはできません。

AAS. 読み込み／更新コマンドに有効です。

AA. 読み込み／更新コマンドに有効です。

AAS,AA,AAS. 読み込みコマンドにのみ有効です。

つまり、更新タイプのコマンドには、同じフィールドを参照するフォーマットバッファエレメントを複数指定することはできません。

AA,AA. レスポンスコード 44 が返されます。

AA,AAS,AA. レスポンスコード 44 が返されます。

長さおよびデータフォーマット

長さおよびフォーマットパラメータは、FDT に定義されている標準値とは異なる長さやフォーマットでフィールド値を指定したり、返したりする場合に使用します。長さまたはフォーマットパラメータを省略した場合、フィールドの標準の長さおよびフォーマットでフィールド値を指定する必要があります。返されるフィールド値も、標準の長さおよびフォーマットとなります。

```
[ , length ] [ , data-format ]
```

「許可されているデータ長とフォーマット」表に変換可能なフォーマットと長さを示します。サブフィールド、サブディスクリプタ、スーパーフィールド、スーパーディスクリプタ、またはハイパーディスクリプタに対してフォーマット変換は指定できません。

指定する長さは、そのフォーマットで値が入るだけの大きさで、かつ許容最大長を超えない値にする必要があります。

長さを 0 に指定するか、または *field-name* が可変長フィールド（標準長なし）として定義されているフィールドを参照する場合は、Adabas がレコードバッファ内に返す値の前には、値の長さを含む（長さバイトも含む）1 バイトのバイナリフィールドが付加されます。更新コマンドの場合、この長さバイトはユーザーがレコードバッファの先頭に指定する必要があります。

フォーマットエレメントに長さゼロを指定すると、レコードバッファの LB フィールド値に使用できるスペースは可変で、かつ実際の LB フィールド値によって異なります。この場合、レコードバッファの LB フィールド値の先頭 4 バイトは、この 4 バイト自体を含む LB フィールドの実際の長さ（LB フィールド値の長さに 4 を加えた値）を格納するのに使用されます。

指定するフォーマットは、フィールドの標準フォーマットと変換可能なフォーマットでなければなりません。

- パック／アンパック 10 進値とバイナリとの変換は、0 から 2,147,483,647 の値の範囲内に限られます。
- 数字形式から英数字に変換すると、アンパック値は左詰め、先行ゼロなし、末尾の空白付きとなります。例えば、3 バイトのパック値 10043F は F1F0F0F4F3404040 に変換されます。変換のときに、値の桁落ちが発生する場合があります。

このsectionでは、さらに次のトピックを取り上げています。

- **アスタリスク (*) を使った長さ表記**
- **編集マスク記号（読み込み時のみ有効）**

■ 例

アスタリスク (*) を使った長さ表記

LA および LB フィールドにのみ、フォーマット要素に長さではなくアスタリスク (*) を指定できます。アスタリスクを標準の英数字フィールドおよびワイド文字フィールドに指定することはできません。アスタリスクは、レコードバッファの LB フィールド値に使用できるスペースが可変となり、LB フィールドの実際の値によって異なることを示します。ただし、長さゼロの指定とは異なり、レコードバッファでは LB フィールド値の前に 4 バイト長フィールドは存在しません。LB フォーマット要素に対応するレコードバッファエリアには、LB フィールドの値のみが含まれます。実際の LB フィールド値の長さは読み込みコマンド用に取得される必要があります、新しいフォーマットバッファ長さインジケータ L を使用する更新コマンド用に指定される必要があります。長さインジケータの詳細については、このマニュアルの「長さインジケータ (L)」を参照してください。

次の例では、LB フィールド L1 のレコードバッファには L1 フィールドの値のみが含まれます。次に 10 バイトが割り当てられた AA フィールドの値が続きます。

```
FB='L1,* ,AA,10,A.'
```

次の例では、LB マルチバリュフィールド L2 のレコードバッファには、L2 フィールドの最初の 10 個の値が含まれます。

```
FB='L21-10,*.'
```

レコードバッファは、そのフォーマット要素のアスタリスクの長さ設定が含まれている場合は、フィールド全体が使用する十分な空きスペースが確保されるため、必ずしも必要ではありません。ただし、読み込みコマンドの処理中に次の両方の条件が合致した場合、フィールド値が切り捨てられます。

- レコードバッファの空きスペースがフィールド値に対して不十分である。
- アスタリスク表記付きのフィールドがフォーマットバッファの最後に指定されている。

これらの条件に合致する場合、エラーは返されません。上記の 2 番目の例 (FB='L21-10,*.') でこの条件が合致した場合、Adabas は読み取る 10 個の値を対応するレコードバッファセグメントの長さに合わせて切り落とします (右から順番に切り落とされます。つまり、最後の値が最初に切り落とされます。まだ残りの空きスペースが不十分な場合、最後から 2 番目が切り落とされ、以下同様に続きます)。極端な場合、このフィールド用にまったく空きスペースがない場合は、値はゼロバイトに切り落とされます。

上記の最初の例では (FB='L1,* ,AA,10,A.')、レコードバッファセグメントが非常に短い場合、固定長が指定されたフィールドまたは長さがゼロ (0) のフィールドには切り落としが許可されていないため、切り落としは発生しません。切り落としは発生しませんが、ニュークリアスはレスポンスコード 53 (レコードバッファが非常に小さい) を返します。

Adabas ニュークリアスにより実行される読み取りコマンドでのみ、アスタリスク表記が指定された値が切り落とされます。更新コマンドでは切り落としは発生しません。さらに、ADACMP ユーティリティは、アスタリスク表記が指定された値を切り落としません。

編集マスク記号（読み込み時のみ有効）

編集マスクは、COBOL プログラミング言語の標準編集マスク規則に従って使用します。

編集マスクは、数字として定義されたフィールドに対してのみ指定できます。Adabas が編集フィールドに返すデータは、フィールドの標準形式にかかわらず、すべてアンパック 10 進数に変換されます。編集フィールドに返すことができる最大桁数（編集文字以外）は 15 桁です。

編集フォーマットが指定されたフィールドの場合、長さパラメータはフィールド値と編集文字が入るだけの大きさである必要があります。

フォーマット	生成される編集マスク
E1	ZZZZZZZZZZZZZZZZ
E2	ZZZZZZZZZZZZZZ9-
E3	ZZZZZZZZZZ99.99.99
E4	ZZZZZZZZZZ99/99/99
E5	Z.ZZZ.ZZZ.ZZZ.ZZZ,ZZ
E6	Z,ZZZ,ZZZ,ZZZ,ZZZ.ZZ
E7	Z,ZZZ,ZZZ,ZZZ,ZZ9.99-
E8	Z.ZZZ.ZZZ.ZZZ.ZZ9.99-
E9	*,***,***,***,**9.99-
E10	*.***.***.***.**9,99-
E11	ユーザー定義編集マスク
E12	ユーザー定義編集マスク
E13	ユーザー定義編集マスク
E14	ユーザー定義編集マスク
E15	ユーザー定義編集マスク



Note: 年月日編集フォーマット E3 と E4 は、年の上 2 桁をスペースに置き換えますが、2000 年の場合にはこの編集を行いません。

例

フォーマットバッファ	フィールド値	編集後の値
XC,15,E1.	009877	bbbbbbbbbb9877
XC,8,E4.	301177	30/11/77
XB,5,E7.	-366	3.66-
XB,7,E9.	542	**5.42
Y2,10,E4.	20000229	2000/02/29

許可されているデータ長とフォーマット

「許可されているデータ長とフォーマット」表に変換可能なフォーマットと長さを示します。サブフィールド、サブディスクリプタ、スーパーフィールド、スーパーディスクリプタ、またはハイパーディスクリプタに対してフォーマット変換は指定できません。

フォーマット	最大長(バイト)	データタイプ	変換可能フォーマット
A	253	英数字。左詰め。	W
W	253 ¹	ワイド文字。左詰め。	A
A、W	16,381 ^{1,2}	ロング英数字 (LA) オプションが付加されたの英数字またはワイド文字。左詰め。2バイトのバイナリの長さバイト (長さバイトを含む) が先行します。	W、A
B	126	バイナリ。右詰め。符号なし。	A、F、P、U
F	4	固定小数点。右詰め。符号付。2または4バイト。	A、B、P、U
G	8	浮動小数点。4または8バイト。	なし
P	15	パック 10 進数。正=A、C、E または F 負=B または D の符号付。	A、B、F、U
U	29	アンパック 10 進数正=A、C、E または F 負=B または D の符号付。	A、B、F、P

 Notes:

- ワイド文字フィールドは、英数字フィールドと同様に、FDT で定義された標準長 (バイト単位) または可変長になります。ワイド文字フィールドを可変長フォーマット以外で書き換える場合は、ユーザーエンコードとの互換性を確保する必要があります。例えば、Unicode は偶数バイトである必要があります (最大 252 バイト)。
- レコードバッファ内のフィールドの先頭に長さバイト (可変フィールド長表記) がある場合は ロング英数字の最大長。それ以外の場合、最大長は 16381 バイトです。
- LA および LB フィールドにのみ、フォーマット要素に長さではなくアスタリスク (*) を指定できます。詳細については、「[アスタリスク \(*\) を使った長さ表記](#)」を参照してください。

フィールドの連続表記

"field-name - field-name" の形式を使用すると、一連の連続するフィールドを（FDT内での並び順に従って）参照することを指定できます。最初と最後のフィールド名をハイフンで結合します。

```
field-name - field-name series notation
```

この範囲内には、マルチプルバリューフィールドおよびピリオディックグループが含まれないようにする必要があります。

グループ名は、最初または最後の名前として指定できませんが、指定範囲内にグループが含まれていてもかまいません。

この範囲内の全フィールドに、標準の長さフォーマットが適用されます。長さやフォーマットの指定もできません。

例	選択される内容
AA-AC フィールド	AA、AB、および AC を参照します。
AA-GC	なし。範囲内にマルチプルバリューフィールドやピリオディックグループを含めることはできません。
GA-AC	なし。範囲の先頭/末尾にはグループフィールドを指定できません。
AA,5,U,-AD	なし。範囲指定では長さやフォーマットは指定できません。

SQL 有意性インジケータおよび連続フィールドの指定

NCフィールドを含むグループまたはフィールド範囲を指定する場合は、読み込み (Lx) コマンドの S 演算子は任意指定です。また、更新 (A1) コマンドでは、S 演算子を指定しないでください。Adabas が、レコードバッファ内のフィールドの直前にフォーマットバッファ内の NC フィールドに対応する空値インジケータが置かれているものと見なすからです。

例えば、FDT に次のフィールド定義があるものとします。

```
01,GR 02,AA,8,A 02,BB,8,A,NC 02,CC,8,A
```

更新タイプのコマンドのフォーマットバッファに次のように指定した場合：GR. または AA-CC. このとき、レコードバッファの構造は次のようになります。

```
AA-value null-indicator-BB BB-value CC-value
```

つまり、フォーマットバッファに S インジケータを指定していない場合にも（指定できません）、レコードバッファシーケンスに空値インジケータが含まれている必要があります。

READ (Lx) コマンドのフォーマットバッファが次のように指定されていた場合：GR,BBS. または AA-CC,BBS. このとき、レコードバッファの構造は次のようになります。

```
AA-value null-indicator-BB BB-value CC-value null-indicator-BB
```

フォーマットバッファ

つまり、最初の空値インジケータはレコードバッファ内に暗黙に出現し、2番目の空値インジケータはフォーマットバッファによって明示的に呼び出されます。

スペース表記 (nX)

nX 構文は、読み込みコマンドと更新コマンドでは使用法が異なります。

nX

読み込みコマンドの場合、nX は次のフィールド値の直前に Adabas が N 個の空白をレコードバッファに挿入することを意味します。

フォーマットバッファ	AA , 5X , BB	読み込むフィールドの名前
レコードバッファ	value-AA 5-blanks value-BB	Adabas から返されるフィールド値

更新コマンドの場合、nX を指定すると、Adabas がレコードバッファ内の n 桁を無視します。

フォーマットバッファ	XX , 5X , YY	更新するフィールドの名前
レコードバッファ	value-XX ignore-5-bytes value-YY	ユーザーが指定したフィールド値

テキスト挿入表記

テキスト構文は、読み込みコマンドと更新コマンドでは使用法が異なります。

'text'

読み込みコマンドの場合、引用符内の文字列をレコードバッファ内で次のフィールド値の直前に挿入します。指定可能な文字列の長さは 1~255 バイトであり、引用符を除く任意の英数字を含めることができます。

例

フォーマットバッファ	AA, 'here is some text', BB	読み込むフィールドの名前
レコードバッファ	value-AA here is some text value-BB	Adabas から返されるフィールド値

更新コマンドの場合、引用符で囲まれた桁数分だけ、レコードバッファ内でその位置に相当する桁が無視されます。

フォーマットバッファでのLA（ロング英数字）フィールドのフィールド長の指定

通常、LA オプションは可変長データとともに使用します。LA オプションを使用する英数字フィールドの長さをフォーマットバッファに指定することもできます。ただし、長さは 16381 バイトに制限されます。

フォーマットバッファ	AA,5,...	読み込むフィールドの名前
レコードバッファ	ABC__	Adabas から返されるフィールド値

LA フィールドは、NU オプション（空値省略）、NC/NNSQL 空値有意性（シグニフィカンス）オプション（ページ）、または MU（マルチプルバリュースフィールド）オプションが指定可能であり、PE（ピリオディック）グループのフィールドにすることもできます。

LA フィールドはディスクリプタではありません。また、サブフィールド、スーパーフィールド、サブディスクリプタ、スーパーディスクリプタ、ハイパーディスクリプタ、フォネティックディスクリプタの親であってもなりません。

Adabas は通常の英字フィールドと同様に LA フィールドを圧縮します。つまり、トレーリング空白は削除されます。バイナリデータを LA フィールドに格納する場合には、このことを頭に入れておく必要があります。

フォーマットバッファのパフォーマンスに関する考慮事項

フォーマットバッファの作成時、次の事項を考慮すると、効率を上げることができます。

- 基本フィールドを個別に参照するのではなく、できる限りグループ名を使用します。グループ名を使用すると、Adabas がフォーマットバッファを解釈するのにかかる時間が短くなります。
- フィールド範囲指定は効率アップにはなりません。フィールド範囲指定は、Adabas により一連の基本フィールドに変換されるためです。
- 長さフォーマットは必要な場合にのみ指定します。これらを指定すると、フォーマットバッファ解釈時およびフィールド処理時に余分な処理時間が必要となります。
- レコード内の同一フィールドを読み込んで更新する場合は、読み込みおよび更新コマンドで同一フォーマットバッファを使用します。詳細については、ページから始まるコマンド ID とフォーマット ID の説明を参照してください。
- ピリオディックグループ（PE）およびマルチプルバリュー（MU）フィールドのオカレンスは、アプリケーションの要件に応じて要求します。つまり、不必要にすべてのオカレンスを

要求しないでください。フォーマットバッファの変換に余分な時間がかかり、多数の空オカレンスの圧縮解除処理が発生します。通常、ピリオディックグループのオカレンスの選択には、AA1-N フィールド名引数が最も効率的です。

12 レコードバッファ

- レコードバッファでの SQL 空値インジケータの指定と読み込み 127
- レコードバッファでの LA (ロング英数字) フィールドのフィールド長の指定 129

レコードバッファ

レコードバッファは、Adabas がデータを返したり、ユーザーが処理対象のデータを指定したりするためのストレージ内のエリアです。レコードバッファが必要になる場合、対応するフォーマットバッファも必要になると見なされます。フォーマットバッファを指定しないと、レコードバッファのペアとなるダミーのフォーマットバッファ（長さゼロ）が作成されます。それぞれのタイプの ABD または バッファ間の関係については、「[バッファタイプ間の関係](#)」を参照してください。

ACBX [ダイレクトコールインタフェース](#)を使用する場合、Adabas [ダイレクトコール](#)に複数のレコードバッファを指定できます。


レコードバッファは、主に読み込みコマンド、検索コマンド、更新コマンドで使用します。

- 読み込みコマンドの場合、[フォーマットバッファ](#)に指定したフィールドの値がレコードバッファに返されます。フィールド値は[フォーマットバッファ](#)に指定した順に返されます。フォーマットバッファとその対応するレコードバッファの一例を次に示します。

フォーマットバッファ	AA, BB	更新するフィールドの名前
レコードバッファ	value-AA value-BB	ユーザーが指定したフィールド値

各値は、[フォーマットバッファ](#)にユーザーが長さやフォーマットを指定しない限り、フィールドの標準の長さやフォーマットで返されます。値が空値である場合は、フィールドのフォーマットに応じて次のように返されます。

フィールドタイプ	空値の表現方法
英数字 (A)	空白 (16 進 '40') またはユーザーが指定したエンコード形式の空白
2 進数 (B)	バイナリの 0 (16 進数 X'00')
固定小数点 (F)	バイナリの 0 (16 進数 X'00')
浮動小数点 (G)	バイナリの 0 (16 進数 X'00')
パック形式 (P)	符号付きパック 10 進数のゼロ (右端の下位バイトに、16 進 '00' に続けて、'0A'、'0B'、'0C'、'0D' または '0F')
アンパック形式 (U)	符号付アンパック 10 進数のゼロ (右端の下位バイトに、X'F0' に続けて X'C0' または X'D0' を伴う)
ワイド文字 (W)	Unicode の空白 (16 進 '20') またはユーザーが指定したエンコード形式の空白

-  **Note:** NC/NN オプション付きフィールドの SQL コンパチブルな空値には、追加の空値と有意性 (シグニフィカンス) インジケータが必要です。詳細については、「[レコードバッファでの SQL 空値インジケータの指定と読み込み](#)」および「[SQL 有意性 \(シグニフィカンス\) インジケータ \(S\)](#)」を参照してください。

Adabas は、要求されたすべてのフィールドの合計長 (標準長または指定された長さ) に等しいバイト数を返します。

- 追加または更新コマンドの場合、フォーマットバッファに指定したフィールドの新しい値を、ユーザーがレコードバッファに指定します。

フォーマットバッファ	XX,YY	更新するフィールドの名前
レコードバッファ	-XX -YY	ユーザーが指定したフィールド値

レコードを更新するときは、レコードバッファに新しい値を指定する必要があります。空値を指定する場合は、上記のフィールドフォーマットに応じて指定する必要があります。

- レコードバッファは、次のコマンドでユーザープログラムと Adabas 間で情報を転送する際にも使用されます。

コマンド	指定するデータ	返されるデータ
OP	更新するファイルおよび操作タイプ (ET、排他制御)	ユーザーデータ (オプション)
LF	-	ファイルのフィールド定義
RE	-	システムファイルに格納されているユーザーデータ
C5	プロテクションログユーザーデータ	-
ET/CL	ユーザーデータ (オプション)	-

OP コマンドの場合、レコードバッファはユーザーのタイプおよび使用されるファイルを示します。

レコードバッファはユーザーデータ (OP、RE、CL、ET コマンド) に使用します。

このchapterでは、次のトピックについて説明します。

レコードバッファでの SQL 空値インジケータの指定と読み込み

Adabas SQL Gateway (ACE) などの SQL をサポートするため、NC/NN (NOT COUNTED/NOT NULL は不可) オプションで定義したフィールドは、レコードバッファに 2 バイトのバイナリ空値インジケータがある SQL の有効空値となります。

値を入力または変更する場合や、値を読み込んでレコードバッファに返す場合、フィールドの "ゼロ" 値が有効なのか、それとも無意味な空値 (未定義) なのかは、レコードバッファに指定された空値インジケータによって決まります。

値を指定したり読み込んだりするときは、次のことを同時に行う必要があります。

- 更新処理の場合、フォーマットバッファのフィールド指定に対応したレコードバッファ位置に空値インジケータを設定します。

レコードバッファ

- 読み込み処理の場合は、フォーマットバッファのフィールドの位置に対応したレコードバッファ位置に空値インジケータが返されたかどうか、プログラムで確認する必要があります。

空値インジケータは、データフォーマットにかかわらず、常に2バイトであり、固定小数点フォーマットです。

読み込み (Lx) または読み込みを伴う検索 (フォーマットバッファエントリがある Sx) コマンドを実行すると、選択したフィールドに含まれている実際の値に応じて、次の空値インジケータ値のいずれかが返されます。

値	説明
X'FFFF'	このフィールドの空値には意味がありません。
0000	このフィールドの空値は有効な値、つまり、ゼロまたは空白です。
xxxxxx	フィールドは桁落としされました。空値インジケータには、データベースレコードに格納されたおりの値の全長 (xxxxxx) が含まれています。

更新 (Ax) または追加 (Nx) コマンドの場合、レコードバッファの空値インジケータ値に次の値を設定する必要があります。

値	説明
X'FFFF'	フィールド値を"未定義"、つまり無意味な空値に設定します。フィールドに設定されたバイナリの0または空白文字は、レコードバッファのフィールドの内容とは無関係です。
0000	レコードバッファにまったく値が指定されていない場合や、バイナリの0や空白が指定されている場合でも、フィールドには有効な空値が含まれています。

追加コマンドの場合、NCオプション付きで定義されたフィールドに対し、レコードバッファにまったく値が提供されていなければ、そのフィールドは空値フィールドとして扱われます。次の例は、NCオプション付きで定義された2バイトのAdabasバイナリフィールドAAの空値の表現方法を示しています。

フィールド定義：01,AA,2,B,NC

	ゼロ以外の値を挿入する場合	空白を挿入する場合	空値を挿入する場合
レコードバッファの空値インジケータ	0 (バイナリの値に有意性あり)	0 (バイナリの空値に有意性あり)	FFFF (バイナリの空値に有意性なし)
データ	0005	0000 (ゼロ)	無関係
Adabasデータベース内の内部表現	0205	0200	C1

更新 (A1/N1) コマンドの場合、NCオプション付きで定義されたフィールドの値は常に有効です。つまり、このフィールドは、空値インジケータ値 "0000" が (指定されていなくても) 指定されているときと同様に扱われます。

読み込みコマンドの場合、NCオプション付きで定義されたフィールドに空値インジケータが指定されていないならば、レコードに有効な値が存在した場合にのみレコードバッファにフィールド値が返されます。無意味なインジケータ値FFFFが指定されているフィールドがデータストレージレコードに存在する場合、レコードが読み込まれるときにレスポンスコード55が返されます。

レコードバッファでのLA（ロング英数字）フィールドのフィールド長の指定

通常、LA オプションは可変長データとともに使用します。LA オプションを使用する英数字フィールドの長さをレコードバッファに指定することもできます。フィールド値の前に、2バイトの長さフィールドが付加されます。長さフィールドには、値の長さに2を加算した値（つまり長さフィールドを含む長さ）が格納されます。

フォーマットバッファ	AA, ...	読み込むフィールドの名前
レコードバッファ	0005ABC ... または 2712 ... (10,000 characters)...	Adabas から返されるフィールド値

13 フォーマットバッファとレコードバッファの例

■ 例 1：基本フィールド使用（標準の長さおよびフォーマット）	133
■ 例 2：基本フィールド使用（長さおよびフォーマットを変更）	134
■ 例 3：ピリオディックグループを参照	135
■ 例 4：ピリオディックグループ GB の第 1、第 2 オカレンスを参照	136
■ 例 5：マルチプルバリュースフィールド MF の 6 番目の値を参照	137
■ 例 6：マルチプルバリュースフィールド MF の 1、2 番目の値を参照	138
■ 例 7：ピリオディックグループ GC の最高オカレンス番号と、マルチプルバリュースフィールド MF に存在する値の数	139

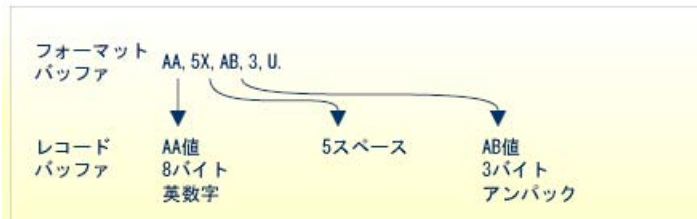
ここでは、フォーマットバッファおよびレコードバッファの例を示します。ここで記述するすべての例で使用されている Adabas ファイルの定義については、「[例で使用されているファイル定義](#)」を参照してください。

このchapterでは、次のトピックについて説明します。

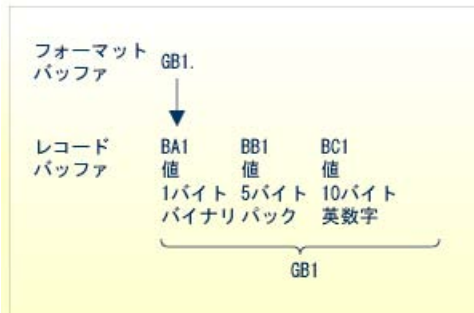
例 1：基本フィールド使用（標準の長さおよびフォーマット）



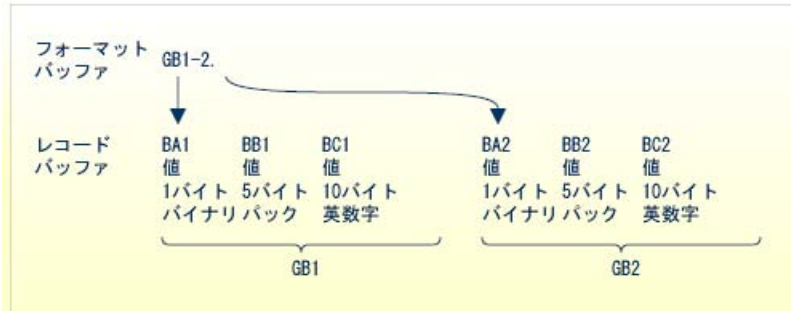
例2：基本フィールド使用（長さとフォーマットを変更）



例 3：ピリオディックグループを参照



例4：ピリオディックグループ GB の第1、第2 オカレンスを参照



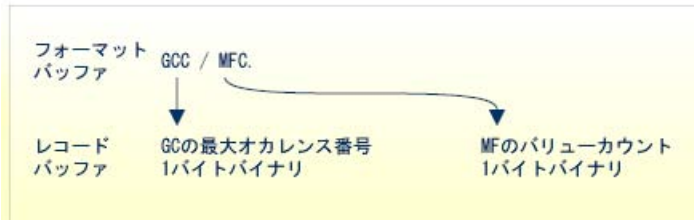
例 5：マルチプルバリューフィールド MF の 6 番目の値を参照



例 6：マルチプルバリューフィールド MF の 1、2 番目の値を参照



例7：ピリオディックグループ GC の最高オカレンス番号と、マルチプルバリューフィールド MF に存在する値の数



14 プリフェッチバッファ

ACB コールでプリフェッチバッファセグメントを使用した場合、Adabas 8 の動作は以前の Adabas リリースと変わりません。ただし、プリフェッチオプション *P* は ACBX ベースのダイレクトコールではサポートされていません。

15 マルチフェッチバッファ

マルチフェッチバッファは、[ACBX](#) [ダイレクトコールインタフェース](#)を使用して実行される一部の Adabas コマンドでのみ必要になります。[ACB](#) [インターフェイスダイレクトコール](#)には必要ありません。

マルチフェッチバッファは、Adabas がマルチフェッチしたレコードのレコードディスクリプタエレメント (RDE) を返すことができるストレージ内のエリアです。このバッファは、(コマンドオプション 1 を "M" に設定して) マルチフェッチオプションをアクティブにした Adabas コマンドでのみ必要になります。RDE の長さはどれも 16 バイトです。

ACBX コマンドのコマンドオプション 1 フィールドにマルチフェッチオプション M を設定すると、対応するフォーマットバッファセグメントのフォーマット指定に基づいて、読み込み中のレコードがすべて指定のレコードバッファセグメントに返されます。レコードバッファセグメントごとに、対応するマルチフェッチバッファセグメントにはレコードバッファセグメントのレコードについて説明したマルチフェッチヘッダーが含まれています。

BT コマンドまたは ET コマンドの場合、コマンドオプション 1 を "M" に設定したときにはマルチフェッチバッファは必要ありません。この場合、ISN バッファはホールドキューから削除する必要のある ISN を格納するのに使用されます。

マルチフェッチバッファが必要になる場合、対応するフォーマットバッファとレコードバッファも必要になると見なされます。対応するバッファを指定しないと、マルチフェッチバッファのペアとなるダミーのフォーマットバッファおよびレコードバッファ (長さゼロ) が作成されます。それぞれのタイプの ABD またはバッファ間の関係については、「[バッファタイプ間の関係](#)」を参照してください。

Adabas [ダイレクトコール](#)に複数のマルチフェッチバッファを指定できます。マルチフェッチ処理の詳細については、「[マルチフェッチ処理](#)」を参照してください。

16 サーチバッファ

■ サーチバッファの形式	147
■ 物理カップリングファイル	153
■ ソフトカップリング	155

サーチバッファとバリュースタックは、次の定義を行うときに、対で使用します。

- FIND コマンド (S1、S2、S4) でレコードの集合を選択するための検索条件
- 読み込みコマンド (L3/6、L9) が論理的な順次読み込みを行う範囲

サーチバッファを指定した場合、バリュースタックも指定されているものと見なされます。バリュースタックが指定されていないと、サーチバッファとペアになるダミーのバリュースタックが作成されます。それぞれのタイプの ABD または バッファ間の関係については、「[バッファタイプ間の関係](#)」を参照してください。

単一の Adabas ダイレクトコールに指定可能なサーチバッファとバリュースタックのペアの数は 1 つだけです。

ユーザーは、サーチバッファに検索式を、バリュースタックには検索式に対応する値を指定します。

サーチバッファの構文は、使用する検索条件のタイプによって異なります。

1. 単一ファイル検索。この検索条件は、単一ファイル内の 1 つ以上のフィールドで構成されます。
2. 物理的にカップリングされたファイルを使用した複数ファイル検索。この検索条件は、ADAINV ユーティリティで物理的にカップリングされている 2 つ以上のファイルに含まれているフィールドで構成されます。
3. ソフトカップリング機能を使用した検索。この機能は、検索、読み込み、内部リストのマッチングの組み合わせができます。

検索条件には、1 つ以上の非ディסקリプタを含めることができます。非ディスクリプタを使用した場合、Adabas は実際に読み込みを行い、どのレコードをユーザーに返すかを判断します。



Notes:

1. ファイルに多数のレコードが含まれる場合、非ディスクリプタを使用して検索を実行すると、レスポンスタイムが長くなることがあります。
2. Adabas データベースにおける非ディスクリプタ値の検索は、メインフレームの場合とオープンシステムの場合とで、フィールド内の空値省略に関して違いがあります。オープンシステムではフィールドが空値省略 (NU) の場合、非ディスクリプタ値を検索したときに空値を含むレコードを返しません。メインフレームシステムでは、非ディスクリプタ値を検索したときにフィールドの空値省略 (NU) は無視されます。このときフィールドを非ディスクリプタ値の検索で使用する必要がある場合は、オープンシステムフィールドの空値省略 (NU) オプションを解除してこの問題を解決することをお勧めします。

サーチバッファの形式

このsectionでは、サーチバッファの形式オプションの概要を示します。各サーチバッファエントリはデリミタ（コンマ、スラッシュ、カッコ、セミコロン）で区切らなければなりません。各エントリ間に1つ以上の空白があってもかまいません。サーチバッファは、ピリオドで終了する必要があります。

このsectionでは、次のトピックについて説明します。

- 検索式
- 検索式の結合
- 1ファイルの検索
- 複数の物理カップリングファイルの検索
- ソフトカップリングを使用した1つ以上のファイルの検索

検索式

検索式はすべての検索タイプに共通です。

$$\left\{ \begin{array}{l} \text{field-name} [i > S] [, \text{length}] [, \text{format}] [, \{ \text{value-operator} | \underline{EQ} \}] \\ (\text{command-id}) \end{array} \right\}$$

検索式は、フィールド名 (*field-name*) とオプションのエントリまたはコマンド ID (*command-id*) で構成されます。

ここでは、この構文の各エレメントについて説明します。

field-name

検索式には、フィールド（ディスクリプタ、非ディスクリプタ）、サブディスクリプタ、スーパーディスクリプタ、ハイパーディスクリプタ、照合ディスクリプタまたはフォネティックディスクリプタを指定できます。非ディスクリプタを使用すると、マルチプルバリューフィールドは指定できますが、サブ/スーパーフィールドは指定できません。

非ディスクリプタを使用する場合、Adabas では検索条件を満たすレコードを決定するのにファイル全体を読み込む必要があります。ただし、ディスクリプタのみを使用した場合、インバーテッドリストが使用され、レコードを読み込む必要はありません。非ディスクリプタを含む検索条件とディスクリプタを含む検索条件は、組み合わせることができます。

ディスクリプタフィールドを初期化せず、論理的に物理レコードの最後に存在している場合、そのレコードのインバーテッドリストエントリはパフォーマンスの理由から生成されま

せん。したがって、そのレコードは検索できません。このとき、インバーテッドリストエントリを生成するには、ファイルのアンロード、圧縮解除、再ロードを行います。あるいは、アプリケーションプログラムを使用してファイルのレコードごとにフィールドを初期化してください。

ディスクリプタが NU オプション（空値省略）と定義されている場合は、空値はインバーテッドリストに格納されません。したがって、空値をもつレコードに対し検索を実行すると、（データストレージの中に空値のディスクリプタをもつレコードがある場合でも）結果は検出レコードなしとなります。この規則は、サブディスクリプタについても適用されます。スーパーディスクリプタの場合は、スーパーディスクリプタを構成するフィールドのうち少なくとも1つのフィールドが NU オプションと定義され、かつそのフィールドが空値であると、スーパーディスクリプタ値は格納されません。



Note: ラージオブジェクト（LB）フィールドをサーチバッファに指定することはできません。同じく、フォーマット選択条件に指定することもできません。

command-id

コマンド ID 値はカッコで囲みます。これは、前回 Sx コマンドで SAVE ISN LIST オプションを指定した結果生成された ISN リストを識別する値です。

i（オカレンスインデックス）

オカレンスインデックス（*i*）は、ピリオディックグループ内のディスクリプタまたは非ディスクリプタの特定のオカレンスを識別し、特定のオカレンスにある値のみを検索できるように制限するのに使用します。インデックスを指定しないと、全オカレンスの値が検索の対象となります。

- インデックスは、1~3 桁から成ります。先行のゼロは可能です。
- インデックスは、次の対象には許可されていません。
 - ピリオディックグループ内のフィールドから生成されたスーパーディスクリプタ
 - マルチプルバリューフィールドであるディスクリプタ、またはマルチプルバリューフィールドから生成されたサブディスクリプタまたはスーパーディスクリプタ。

この場合、ピリオディックグループの全オカレンスの値が検索の対象となります。

S（シグニフィカンス）および空値インジケータ

SQL 空値圧縮オプション NC のあるフィールドの場合、S 空値インジケータを使用して "空値" か "空値でない" かを選択できます。空値インジケータの説明については、前述の「[SQL 有意性（シグニフィカンス）インジケータおよび連続フィールドの指定](#)」を参照してください。



Note: NC オプションは、NU（空値省略）、FI（固定ストレージ）、MU（マルチプルバリュー）、または PE（ピリオディックグループ）を指定したフィールド、またはグループフィールドには適用できません。

SQL 有意性（シグニフィカンス）インジケータ（S）は、フィールド名の後に指定する必要があります（フィールド名 S）。また、対応する SQL 空値インジケータはバリュースタックに指定する必要があります。

検索指数として次の 16 進空値インジケータを使用できます。

X'FFFF' 空値を選択します。

0000 空値以外を選択します。

空値インジケータ値がこれ以外である場合は、Adabas レスポンスコード 52 が返されます。

空値インジケータ値 (16 進の FFFF または 0000) は、長さが 2 バイトの標準長で、フォーマットが固定小数点となります。この長さとはフォーマットは変更しないでください。

S インジケータとともに使用できる値演算子は等号 (=) のみです。他の値演算子を使用すると、Adabas レスポンスコード 61 が返されます。


例

S 有意性演算子が、フィールド名 AA の検索指数に含まれています。

AAS.

FN フィールド値がパック +1 で、かつ AA フィールドが空値 (未定義) のレコードを選択します。

サーチバッファ	FN , 2 , P , D , AAS.	検索指数
バリューバッファ	001F FFFF	フィールド値の指定

 **Note:** インデックスには、無意味な空値は格納されません。このため、空値を検索すると、アプリケーションプログラムのパフォーマンスが大幅に低下する場合があります。

FN フィールド値がパック +1 で、かつ AA フィールドが空値以外のレコードを選択します。

サーチバッファ	FN , 2 , P , D , AAS.	検索指数
バリューバッファ	001F 0000	フィールド値の指定

length

バリューバッファに指定するフィールド/ディスクリプタ値の長さ。この長さを省略した場合、バリューバッファの値は、「許可されているデータ長とフォーマット」表に示すように、フィールド/ディスクリプタの標準長に準拠する必要があります。

format

バリューバッファに指定するフィールド/ディスクリプタ値のフォーマット。このフォーマットを省略した場合、バリューバッファの値は、「許可されているデータ長とフォーマット」表に示すように、フィールド/ディスクリプタの標準長に準拠する必要があります。

ト」表に示すように、フィールド／ディスクリプタの標準のフォーマットに準拠する必要があります。

value-operator

値演算子は、先行するディスクリプタとバリュースタック内の対応する値との間で行う論理演算を示します。

次の演算子を指定できます。

演算子	説明
EQ または =	等しい
GE	より大きいまたは等しい
GT または >	より大きい
LE	より小さいまたは等しい
LT または <	より小さい
NE	等しくない

値演算子を指定しないと、等しい (EQ) が指定されたものと見なされます。

例

次のサーチバッファの例は、値演算子の使用方法を示しています。

例	説明
AA.	AA がバリュースタック内の指定値と等しい (デフォルト)。
AA,LT.	AA がバリュースタック内の指定値より小さい。
AA,GE.	AA がバリュースタック内の指定値より大きいまたは等しい。

NE 演算子 (NOT EQUAL TO) を使用した次のサーチバッファは、FN フィールドの値が "MIKE" でないレコードをすべて選択します。

サーチバッファ	FN,4,A,NE.	検索指数
バリュースタック	MIKE	フィールド値の指定

この例の NE 演算子を EQ (等しい) に置き換えると、FN フィールドの値が "MIKE" であるレコードのみが選択されます。

検索式の結合

結合演算子は、検索式を結合するのに使用します。次の結合演算子が用意されています。

演算子	説明
D	<p>2つの検索式の結果を論理積（AND）演算で結合します。例としては、次のようなものがあります。</p> <p>AA, D, AB.</p>
O	<p>2つの検索式の結果を論理和（OR）演算で結合します。OR演算は、同じディスクリプタを使用する検索式を結合する場合にのみ使用できます。有効な例と無効な例は次のとおりです。</p> <p>有効な例</p> <p>AA, O, AA.</p> <p>無効な例（指定した2つのディスクリプタが異なる）</p> <p>AA, O, AB.</p>
R	<p>さまざまなディスクリプタから生成された ISN リストを指すフィールドまたはコマンド ID を論理和（OR）演算で結合します。例としては、次のようなものがあります。</p> <p>AA, 5, A, R, AB, LT.</p>
S	<p>2つの検索式の FROM-TO 範囲（開始と終了も含めて）を示します。2つの式に同じディスクリプタを指定する必要があります。有効な例と無効な例は次のとおりです。</p> <p>有効な例</p> <p>AA, S, AA.</p> <p>無効な例（指定した2つのディスクリプタが異なる）</p> <p>AA, S, AB.</p>
N	<p>直前の FROM-TO 範囲から単一の値またはある範囲の値を除外します。この演算子は必ず S 演算子とともに指定します。FROM-TO 範囲に指定するフィールドと同じフィールドに適用する必要があります。フィネティックディスクリプタは指定できません。以下に、有効な例と無効な例をいくつか示します。</p> <p>■ 有効な例</p> <p>AA, S, AA, N, AA.</p> <p>無効な例（指定した2つのディスクリプタが異なる）</p> <p>AA, S, AA, N, AB.</p> <p>■ 有効な例</p>

演算子	説明
	<p>AA,S,AA,N,AA,S,AA.</p> <p>無効な例（指定した2つのディスクリプタが異なる）</p> <p>AA,S,AA,N,AA,S,AB.</p> <p>AA,S,AA,N,AA,N,AB.</p>
Y	<p>D、O、R、S、Nの検索式の結果を任意の数だけ論理積（AND）演算で結合します。例としては、次のようなものがあります。</p> <p>AA,D,AB,Y,AA,O,AA,Y,AA,S,AA,N,AA,S,AA.</p> <p>Y結合演算子はカッコのような働きをします。カッコのネストはサポートされておらず、単一レベルにのみ使用できます。Y結合演算子で結合される検索式はすべて、同一のファイルに適用する必要があります。</p>

詳細な例については、「[サーチ/バリュースタックの例](#)」を参照してください。

■ **サーチバッファ内の結合演算子の処理順**

サーチバッファ内の結合演算子の処理順

1つのサーチバッファ内の引数に複数の結合演算子を指定した場合、結合演算子は次の順で処理されます。

1. すべての S/N/O 演算子を評価します。
2. すべての D 演算子を評価します。
3. すべての R 演算子を評価します。
4. すべての Y 演算子を評価します。

例

サーチバッファの指定内容

AA,S,AA,O,AA,D,AB,R,AC,D,AD.

処理される順番

(((AA,S,AA),O,AA),D,AB),R,(AC,D,AD)

サーチバッファの指定内容

AA,D,AB,Y,AA,O,AA,Y,AA,S,AA,N,AA,S,AA.

処理される順番

(AA,D,AB),Y,(AA,O,AA),Y,((AA,S,AA),N,(AA,S,AA))

1 ファイルの検索

単一ファイル検索に関するサーチバッファは次の構文にしたがいます。

```
search-expression [{,connecting-operator,search-expression}...] .
```

*search-expression*の構文については、「[検索式](#)」を参照してください。*connecting-operator*の詳細については、「[検索式の結合](#)」を参照してください。

複数の物理カップリングファイルの検索

2つ以上の物理カップリングファイルのフィールドを使用して、複数ファイル検索を行うサーチバッファの構文は、次のとおりです。

```
/file-x/search-expression[{,connecting-operator,search-expression}...]
{,D,/file-y search-expression/[,{,connecting-operator,search-expression}...]}
```

ここで、*file-x*と*file-y*には、物理カップリングファイルのファイル番号が入ります。物理カップリングファイルを使用したサーチバッファ構文については、「[物理カップリングファイル](#)」を参照してください。*search-expression*の構文については、「[検索式](#)」を参照してください。*connecting-operator*の詳細については、「[検索式の結合](#)」を参照してください。

ソフトカップリングを使用した1つ以上のファイルの検索

ソフトカップリングを使用して検索を行うサーチバッファは、次の構文にしたがいます。

```
(m-file,m-field,s-file,s-field[;m-file,m-field,s-file,s-field
]...)/s-file-x/search-expression[{,connecting-operator,search-expression}...]
[,{,D,/s-file-y/search-expression[{,connecting-operator,search-expression}...]}
```

ここで、*m-field* *s-file-x*と*s-file-y*には、ソフトカップリングファイルのファイル番号が入ります。

ソフトカップリングファイルを使用したサーチバッファ構文については、「[ソフトカップリング](#)」を参照してください。*search-expression*の構文については、「[検索式](#)」を参照してください。*connecting-operator*の詳細については、「[検索式の結合](#)」を参照してください。

物理カップリングファイル

2つ以上の物理カップリングファイルのフィールドを使用して、複数ファイル検索を行うサーチバッファの構文は、次のとおりです。

```
/file-x/search-expression[{,connecting-operator,search-expression}...]
{,D,/file-y search-expression/[,{,connecting-operator,search-expression}...]}
```

物理カップリングファイルの検索条件は任意の順序で指定できます。実際に返される ISN の値は、Adabas コントロールブロックのファイル番号フィールドに指定したカップリングファイルのものであり、このファイルを "プライマリ" ファイルといいます。

この構文の各要素は次のとおりです。

file-x および **file-y**

物理カップリングファイルのファイル番号。指定するファイルはすべて、ADAINVユーティリティのCOUPLE機能を使用して、あらかじめカップリングしておく必要があります。ファイル番号は各ファイルに対し1回だけ指定すればよく、検索条件（1つ以上の検索式と適切な結合演算子で構成）の直前に指定する必要があります。物理カップリングファイルの単一のサーチバッファに最大5（5）ファイルを指定できます。

D

物理カップリングファイルの検索条件間で許可されている結合演算子は、論理積（AND）の記号Dのみです。

search-expression

関連する物理カップリングファイルの検索式。search-expressionの構文については、「[検索式](#)」を参照してください。

connecting-operator

個々の物理カップリングファイルに対する検索条件の検索式をつなぐ結合演算子。物理的にカップリングされたファイルの検索条件を結合する場合、結合演算子には"D"（AND）を使用する必要があります。個別ファイルを対象とする検索条件から構成される検索式を結合する場合、結合演算子には、「[検索式の結合](#)」で説明している演算子をすべて使用することができます。

例

ファイル1のレコードのうち、フィールドRBの値がABCDEであるファイル2のレコードとカップリングされており、かつフィールドABの値が3バイト（長さ指定）アンパック10進（フォーマット指定）の+20であるレコードのISNを検索します。なお、フィールドABは10バイト標準長の英数字フォーマットです。

サーチバッファ	/1/AB,3,U,D,/2/RB.	検索指数
バリューバッファ	character-notation	フィールド値の指定
	020ABCDEbbbbbb	
	hex-notation	

	F0F2F0C1C2C3C4C54040404040	
--	----------------------------	--

ソフトカップリング

ソフトカップリングを使用して検索を行うサーチバッファの構文は次のとおりです。

```
(m-file,m-field,s-file,s-field[{;m-file,m-field,s-file,s-field
}...])/s-file-x/search-expression[{,connecting-operator,search-expression}... ]
[ {,D,/s-file-y/search-expression[{,connecting-operator,search-expression}...]}... ] .
```

ソフトカップリング検索条件の例については、「[サーチ／バリューバッファの例](#)」の例 **15～18** を参照してください。

この構文の各要素は次のとおりです。

m-file*,*m-field

m-file には、主ファイルの番号を指定します。このファイルは、Adabas コントロールブックのファイル番号フィールドに指定されていなければなりません。最終的に返される ISN 結果リストには、主ファイルにのみ存在する ISN が記載されます。

m-field には、ソフトカップリングのリンクフィールドとして使用される主ファイルのフィールドを指定します。このフィールドは、ディスクリプタ、サブディスクリプタ、スーパーディスクリプタまたはハイパーディスクリプタでなければなりません。ピリオディックグループ内のフィールドまたはロング英数字フィールドは指定できません。

m-file、*m-field*、*s-file*、*s-field* の指定を組み合わせたものが、単一のソフトカップリングとなります。最大で 42 のソフトカップリング条件が指定できます。ソフトカップリング全体を 1 組のかっこの中に指定する必要があります。

s-file*, *s-field

s-file には検索ファイルのファイル番号を指定し、*s-field* には検索ファイル内のフィールドを指定します。（検索条件に従って）この検索ファイルから選択された ISN ごとに、*s-field* として指定したフィールドが読み込まれます。このフィールド値は、主ファイルのどの ISN に一致する値があるかを判断するのに使用されます。

このフィールドは、ディスクリプタでも非ディスクリプタでもよく、サブディスクリプタ、スーパーディスクリプタ、ハイパーディスクリプタ、ロング英数字フィールドであってもかまいません。フォーマットは、対応する *m-field* フィールドのものと同じでなければなりません。標準長は異なってもかまいません。このフィールドはピリオディックグループに含まれてはなりません。

m-file、*m-field*、*s-file*、*s-field* の指定を組み合わせたものが、単一のソフトカップリングとなります。最大で 42 のソフトカップリング条件が指定できます。ソフトカップリング全体を 1 組のかっこの中に指定する必要があります。

s-file-x および ***s-file-y***

検索条件を指定するカップリングファイルのファイル番号。ファイル番号は各ファイルに対し1回だけ指定すればよく、検索条件（1つ以上の検索式と適切な結合演算子で構成）の直前に指定する必要があります。単一のサーチバッファに最大5（5）ファイルを指定できません。

D

カップリングファイルの検索条件間で許可されている結合演算子は、論理積（AND）の記号Dのみです。

search-expression

関連するカップリングファイルの検索式。 *search-expression* の構文については、「[検索式](#)」を参照してください。

connecting-operator

個々のカップリングファイルに対する検索条件の検索式をつなぐ結合演算子。物理的にカップリングされたファイルの検索条件を結合する場合、結合演算子には"D"（AND）を使用する必要があります。個別ファイルを対象とする検索条件から構成される検索式を結合する場合、結合演算子には、「[検索式の結合](#)」で説明している演算子をすべて使用することができます。

17 バリュースタック

■ バリュースタックの構造	158
■ SQL 空値とインジケータ	158
■ 符号の取り扱い	159

サーチスタックとバリュースタックは、次の定義を行うときに、対で使用します。

- FIND コマンド (S1、S2、S4) でレコードの集合を選択するための検索条件
- 読み込みコマンド (L3/6、L9) が論理的な順次読み込みを行う範囲

バリュースタックを指定した場合、サーチスタックも指定されているものと見なされます。指定しないと、バリュースタックのペアとなるダミーのサーチスタックが作成されます。それぞれのタイプの ABD またはスタック間の関係については、「[スタックタイプ間の関係](#)」を参照してください。

単一の Adabas ディレクトコールに指定可能なサーチスタックとバリュースタックのペアの数は 1 つだけです。

ユーザーは、サーチスタックに検索式を、バリュースタックには検索式に対応する値を指定します。

サーチスタックに指定した各ディスクリプタの値をバリュースタックに指定します。

検索式がコマンド ID であれば、バリュースタックに対応するエントリを作成しません。

バリュースタックの構文

指定する各値は、サーチスタックに指定した各検索式と同じ順序でなければなりません。指定する全値は、サーチスタックに指定した検索式に対応して同じ順序でなければならず、(サーチスタックに長さフォーマットを指定しないとき) サーチスタック内の関連するディスクリプタの標準長と標準フォーマットに一致していなければなりません。

間に空白を入れたり、他のコンマなどの文字をバリュースタック内の値の間に挿入することはできません。バリュースタックの末尾にピリオドを指定する必要はありません。

SQL 空値とインジケータ

NC (SQL 空値 "NOT COUNTED") オプション付きで定義されたフィールドを検索するときは、サーチスタックのフィールド定義に空値シグニフィカンス (S) インジケータを付け、対応するバリュースタック引数値に 2 バイトのバイナリ空値インジケータを示さなければなりません。バリュースタックの空値インジケータの説明や例については、「[S \(シグニフィカンス\) と空値インジケータ](#)」を参照してください。

符号の取り扱い

バイナリ値は符号なしの数として扱います。固定小数点、アンパックおよびパックの値は符号付数値として扱います。指定できる有効な符号については、このsectionで説明します。

- 固定値の符号
- アンパック値の符号
- パック値の符号

固定値の符号

固定値の場合、符号はビット 0（高位のビット）にあります。

- 0=正
- 1=負（2の補数）

以下に固定値の符号の例を2つ示します。16進数と10進数で表記しています。

```
00000005 = +5
FFFFFFFFB = -5
```

アンパック値の符号

アンパック値の場合、符号は下位バイトの高位4ビットにあります。

- CまたはAまたはFまたはE=正（CAFE）
- BまたはD=負（BD）

以下にアンパック値の符号の例を2つ示します。16進数と10進数で表記しています。

```
F1F2F3 = +123
F1F2D3 = -123
```

パック値の符号

パック値の場合、符号は下位バイトの下位4ビットにあります。

- AまたはCまたはEまたはF=正
- BまたはD=負

パックフィールドから生成したスーパーディスクリプタに対して検索値を指定する場合、正の符号にはFを、負の符号にはDを指定する必要があります。

以下にパック値の符号の例を2つ示します。16進数と10進数で表記しています。

バリュースタック

X'123F' = +123

X'123C' = +123

X'123D' = -123

18 サーチバッファとバリュースタックの例

■ 例 1：単一の検索式の使用	162
■ 例 2：AND で結合した検索式の使用	162
■ 例 3：OR で結合した検索式の使用	163
■ 例 4：FROM-TO で結合した検索式の使用	164
■ 例 5：BUT-NOT で結合した検索式の使用	164
■ 例 6：マルチプルバリュースタックの使用	165
■ 例 7：ピリオディックグループ内のスタックの使用	165
■ 例 8：サブスタックの使用	166
■ 例 9：英数字フォーマットのスーパースタックの使用	167
■ 例 10：バイナリ形式のスーパースタックの使用	167
■ 例 11：以前に生成した ISN リストの使用	167
■ 例 12：値演算子の使用	168
■ 例 13：値演算子および結合演算子の使用	169
■ 例 14：物理カップリングファイルの使用	169
■ 例 15：単一ソフトカップリング条件と単一検索条件の使用	170
■ 例 16：単一のソフトカップリング条件と複数の検索条件の使用	170
■ 例 17：複数のソフトカップリング条件と複数の検索条件の使用	171
■ 例 18：複数のソフトカップリング条件と、物理カップリングによる複数の検索条件の使用	171

ここでは、サーチバッファとバリュースタックの構成例を示します。ここで記述するすべての例で使用されている Adabas ファイルの定義については、「[例で使用されているファイル定義](#)」を参照してください。バリュースタックの値は文字および/または 16 進数で表示します。

例 1：単一の検索式の使用

単一の検索式を使用する検索。

ファイル 1 のレコードのうち、フィールド AA の値が 12345 であるレコードの ISN を選択します。なお、フィールド AA は、8 バイト標準長の数値フォーマットです。

サーチバッファ	AA.	検索引数
バリュースタック	character-notation	フィールド値の指定
	00012345	
	hex-notation	
	F0F0F0F1F2F3F4F5	

サーチバッファに AA,5. と指定し、バリュースタックに値 12345（末尾の空白なし）としても同じ検索が実行されます。

例 2：AND で結合した検索式の使用

2 つの検索式を AND（論理積）演算で結合して検索します。

ファイル 1 のレコードのうち、フィールド AA の値が 12345678 で、かつフィールド AB の値が +2 であるレコードの ISN を選択します。なお、フィールド AA は 8 バイト標準長の数値フォーマットで、フィールド AB は 2 バイト標準長のパック 10 進フォーマットです。

サーチバッファ	AA,D,AB.	検索引数
---------	----------	------

バリュースタック	hex-notation	フィールド値の指定
	F1F2F3F4F5F6F7F8002C	

ファイル1のレコードのうち、フィールド AA の値が 12345678 で、かつフィールド AB の値が +2 であるレコードの ISN を選択します。なお、フィールド AA は 8 バイト標準長の数値フォーマットで、フィールド AB は 3 バイト長（上書き）のアンパック 10 進フォーマット（上書き）です。

サーチバッファ	AA,D,AB,3,U.	検索回数
バリュースタック	character-notation	フィールド値の指定
	12345678002	
	hex-notation	
	F1F2F3F4F5F6F7F8F0F0F2	

この2つ目の検索は、サーチバッファに長さフォーマットを指定する例を示しています。結果は、最初の検索と同じになります。

例 3：OR で結合した検索式の使用

3つの検索式を OR（論理和）演算で結合して検索します。

ファイル2のレコードのうち、フィールド XB の値が 284、285、290 のいずれかであるレコードの ISN を選択します。また長さフォーマットを変更しています。

サーチバッファ	XB,3,U,O,XB,3,U,O,XB,3,U.	検索回数
バリュースタック	character-notation	フィールド値の指定
	284285290	
	hex-notation	

	F2F8F4F2F8F5F2F9F0	
--	--------------------	--

例 4：FROM-TO で結合した検索式の使用

2つの検索式を FROM-TO（範囲）演算で結合して検索します。

ファイル 2 のレコードのうち、フィールド XB の値が +20 から +30 の範囲にあるレコードの ISN を選択します。

サーチバッファ	XB,S,XB.	検索引数
バリュースタック	hex-notation	フィールド値の指定
	020C030C	

例 5：BUT-NOT で結合した検索式の使用

3つの検索式を FROM-TO（範囲）と BUT-NOT（除外）演算で結合して検索します。

ファイル 2 のレコードのうち、フィールド XB の値が、+27 を除く +20 から +30 の範囲にあるレコードの ISN を選択します。

サーチバッファ	XB,S,XB,N,XB.	検索引数
バリュースタック	hex-notation	フィールド値の指定
	020C030C027C	

ファイル 2 のレコードのうち、フィールド XB の値が、+1 から +200 または +500 から +600 の範囲にあるレコードの ISN を選択します。

サーチバッファ	XB,S, XB,O, XB,S, XB.	検索引数
バリュースタック	hex-notation 001C200C500C600C	フィールド値の指定

例 6：マルチプルバリュースタックの使用

マルチプルバリュースタックを使用した検索。

ファイル 1 のレコードのうち、マルチプルバリュースタック MF のいずれかの値が ABC である ISN を選択します。

サーチバッファ	MF.	検索引数
バリュースタック	character-notation ABC hex-notation C1C2C3	フィールド値の指定

検索をマルチプルバリュースタックの特定のオカレンスに制限することはできません。次のサーチバッファエントリは無効です。

サーチバッファ	MF2.	検索引数
---------	------	------

例 7：ピリオディックグループ内のディスクリプタの使用

ピリオディックグループ内のディスクリプタを使用して検索します。

サーチバッファとバリュースタックの例

ファイル1のレコードのうち、ディスクリプタ BA（ピリオディックグループに属している）のいずれかのオカレンスの値が4であるレコードの ISN を選択します。

サーチバッファ	BA.	検索回数
バリュースタック	hex-notation	フィールド値の指定
	04	

ファイル1のレコードのうち、ディスクリプタ BA（ピリオディックグループに属している）の第3オカレンスの値が4であるレコードの ISN を選択します。

サーチバッファ	BA3.	検索回数
バリュースタック	hex-notation	フィールド値の指定
	04	

例8：サブディスクリプタの使用

サブディスクリプタを使用する検索。SAはフィールド RA の先頭4バイトから生成したサブディスクリプタです。

ファイル2のレコードのうち、サブディスクリプタ SA の値が ABCD であるレコードの ISN を選択します。

サーチバッファ	SA.	検索回数
バリュースタック	hex-notation	フィールド値の指定
	C1C2C3C4	

例 9：英数字フォーマットのスーパーディスクリプタの使用

英数字フォーマットのスーパーディスクリプタを使用する検索。SB は、フィールド RA の先頭 8 バイトとフィールド RB の先頭 4 バイトから生成したスーパーディスクリプタです。

ファイル 2 のレコードのうち、スーパーディスクリプタ SB の値が ABCDEFGH1234 であるレコードの ISN を選択します。

サーチバッファ	SB.	検索指数
バリュースキーマ	hex-notation	フィールド値の指定
	C1C2C3C4C5C6C7C8F1F2F3F4	

例 10：バイナリ形式のスーパーディスクリプタの使用

バイナリ形式のスーパーディスクリプタを使用する検索。SC は、フィールド XB と XC から生成したスーパーディスクリプタです。

ファイル 2 のレコードのうち、フィールド XB の値が +20 で、かつフィールド XC の値が 123456 であるレコードの ISN を返します。

サーチバッファ	SC.	検索指数
バリュースキーマ	hex-notation	フィールド値の指定
	020FF1F2F3F4F5F6	

例 11：以前に生成した ISN リストの使用

以前に生成した ISN リスト（コマンド ID で識別される）を使用する検索。

サーチバッファとバリュースタックの例

CID1で識別されるISNリストとCID2で識別されるISNリストのどちらにも存在するISNを選択します。

サーチバッファ	(CID1),D,(CID2).	検索引数
バリュースタック	使用しません。	フィールド値の指定

ファイル1のレコードのうち、CID1で識別されるISNリストにISNが存在し、かつフィールドABの値が+123であるレコードのISNを選択します。

サーチバッファ	(CID1),D,AB,3,U.	検索引数
バリュースタック	character-notation 123 hex-notation F1F2F3	フィールド値の指定

例 12：値演算子の使用

値演算子を使用した検索。

ファイル1のレコードのうち、フィールドABの値が+100より大きいレコードのISNを選択します。

サーチバッファ	AB,3,U,GT.	検索引数
バリュースタック	character-notation 100 hex-notation F1F0F0	フィールド値の指定

例 13：値演算子および結合演算子の使用

値演算子および結合演算子を使った検索。

ファイル 1 のレコードのうち、フィールド AB の値が +100 より大きく、かつフィールド AA の値が A より大きいレコードの ISN を選択します。

サーチバッファ	AB, 3, U, GT, D, AA, 1, GT.	検索指数
バリュースタック	character-notation 100A hex-notation F1F0F0C1	フィールド値の指定

例 14：物理カップリングファイルの使用

カップリングされているファイルを参照する検索式を使用した検索。

ファイル 1 のレコードのうち、フィールド RB の値が ABCDE であるファイル 2 のレコードとカップリングしており、かつフィールド AB の値が +20 であるレコードの ISN を選択します。また、フィールド AB には標準と異なる長さフォーマットで値を指定しています。

サーチバッファ	/1/AB, 3, U, D, /2/RB.	検索指数
バリュースタック	character-notation 020ABCDEbbbbbb hex-notation F0F2F0C1C2C3C4C54040404040	フィールド値の指定

例 15：単一ソフトカップリング条件と単一検索条件の使用

ISN が返されるファイルは、ファイル番号フィールドによって決められます。

ファイル番号	4	
サーチバッファ	(4,AB,1,AC)/1/AB,S,AB.	検索指数
バリュースタック	-----	フィールド値の指定

1. バリュースタックに指定した AB の値についてファイル 1 を検索します。
2. ファイル 1 で得られた ISN ごとに、フィールド AC を読み込み、その値をファイル 4 の対応する値リストと内部的に照合します。ファイル 4 の ISN 結果リストが、ISN バッファに返されます。

例 16：単一のソフトカップリング条件と複数の検索条件の使用

ISN が返されるファイルは、ファイル番号フィールドによって決められます。指定する条件の順序は任意です。

ファイル番号	1	
サーチバッファ	(1,AA,2,AB)/1/AC,D,AE,D,/2/AF,S,AF.	検索指数
バリュースタック	-----	フィールド値の指定

1. AF=... THRU ... に関するファイル 2 の検索（バリュースタックの値を使用）
2. ファイル 2 で得られた各 ISN について、フィールド AB を読み込み、さらにファイル 1 の値リストと内部的に照合します。
3. AC=... AND AE=... に関するファイル 1 の検索（バリュースタックの値を使用）。
4. ステップ 2 と 3 の ISN 結果リストを照合します。この結果得られた ISN が ISN バッファに返されます。

例 17：複数のソフトカップリング条件と複数の検索条件の使用

ファイル番号	1	
サーチバッファ	(1,AA,2,AB; 1,AA,5,BA) /1/AC,D,AE,D,/2/AF,S,AF,D,/5/BC,S,BC.	検索指数
バリュースタック	-----	フィールド値の指定

1. AF=... THRU ... に関するファイル 2 の検索（バリュースタックの値を使用）。
2. ファイル 2 で得られた各 ISN について、フィールド AB を読み込み、さらにファイル 1 の値リストと内部的に照合します。
3. BC=... THRU ... に関するファイル 5 の検索（バリュースタックの値を使用）。
4. ファイル 5 で得られた ISN ごとに、フィールド BA を読み込み、その値をファイル 1 の対応する値リストと内部的に照合します。
5. AC=... AND AE=... に関するファイル 1 の検索（バリュースタックの値を使用）。
6. ステップ 2、4、5 から得られる ISN 結果リストを照合します。この結果得られた ISN が ISN バッファに返されます。

例 18：複数のソフトカップリング条件と、物理カップリングによる複数の検索条件の使用

サーチバッファ	(1,AA,2,AB) /1/AC,D,AE,D,/2/AF,S,AF,D,/5/BC,S,BC.	検索指数
バリュースタック	-----	フィールド値の指定

1. AF=... THRU ... に関するファイル 2 の検索（バリュースタックの値を使用）
2. ファイル 2 で得られた各 ISN について、フィールド AB を読み込み、さらにファイル 1 の値リストと内部的に照合します。
3. BC=... THRU ... に関するファイル 5 の検索（バリュースタックの値を使用）。
4. ファイル 5 で得られた ISN ごとに、ADAINV ユーティリティで作成した物理カップリングリストを使用して ISN を照合します。これは、ファイル 5 と主ファイル（ファイル 1）との間でソフトカップリング条件が使用されていない場合に行われます。物理カップリングリストが存在しない場合、ファイル 5 の ISN は考慮されません。

5. AC=... AND AE=... に関するファイル 1 の検索（バリュースタックの値を使用）。
6. ステップ 2、4、5 から得られる ISN 結果リストを照合します。この結果得られた ISN が ISN バッファに返されます。

19 ISN バッファ

ISN バッファは、コマンドに指定した検索条件を満たすレコードの ISN が返されるストレージ内のエリアです。また、コマンドの `Command Option 1` を "M" (または "P") に設定し、それがそのコマンドの有効な設定であり、かつ **ACB** [ダイレクトコールインターフェイス](#) を使用してそのコマンドを発行した場合、ISN バッファにはマルチフェッチまたはプリフェッチした処理待機中のレコードのレコードディスクリプタエレメント (RDE) が保持されます。コマンドを **ACBX** [ダイレクトコールインターフェイス](#) を使用して発行した場合は、ISN バッファではなくマルチフェッチバッファがこの目的に使用されます。

ダイレクトコールで ISN バッファが必要な場合は、1 つのみ指定してください。

ISN バッファには通常、4 バイトの 2 進数の ISN が昇順で返されます。S2 または S9 コマンドの場合、ISN はユーザー指定のソート順で返されます。ISN バッファが実行結果の ISN 結果リストが入るほど大きくない場合、(要求があれば) あふれた ISN は Adabas WORK データセットに格納されます。このようなあふれた ISN は後で読み込むことができます。

実行結果の ISN を L1 コマンドまたは L4 コマンドの GET NEXT オプションを使用して読み込む場合、ISN バッファは必要ありません。

また、ET コマンドまたは BT コマンドで `Command Option 1` を "M" (または "P") に指定した場合、入力として使用される ISN リストが ISN バッファから提供されます。

20 ユーザーバッファ

ユーザーバッファを使用すると、Adabas ニュークリアスユーザー出口 11 と 4、および Adalink ユーザー出口 1 と 2 (Adabas 7 インストールの場合は A と B) への入力を提供できます。この入力の構文は、ユーザー出口プログラムの要件によって異なります。

単一のダイレクトコールに指定できるユーザーバッファは 1 つだけです。ユーザーバッファの内容は単に Adabas によってアプリケーションに渡されるだけであり、処理されることはありません。

Adabas ユーザー出口 11 と 4 の詳細については、「ユーザー出口 11 (一般処理)」および「ユーザー出口 4 (ユーザー生成のログデータ)」を参照してください。Adalink ユーザー出口 1 と 2 (または A と B) の詳細については、該当する Adabas インストールマニュアルを参照してください。

21 パフォーマンスバッファ

パフォーマンスバッファは、Adabas Review を使用している場合にのみ使用されます。詳細については、Adabas Review のドキュメントを参照してください。

22 コマンド

このchapterでは、各 Adabas コマンドをコマンドコード別かつアルファベット順に詳しく説明します。

情報は次の項目で構成されています。

- **A1 コマンド**：レコードの更新 レコードを更新します（ホールドオプション）。
- **BT コマンド**：トランザクションのバックアウト ET ロジックユーザーのデータベース更新を元に戻します。
- **C1 コマンド**：チェックポイントの書き込み コマンド ID、PLOG、RABN、チェックポイント、バッファフラッシュオプションを書き込みます。
- **C3 コマンド**：チェックポイントの書き込み 排他制御更新ユーザーの SYNX-03 チェックポイントを書き込みます。ユーザーデータを格納するためのオプションです。
- **C5 コマンド**：プロテクションログへのユーザーデータの書き込み SIBA/PLOG 上のユーザーデータを書き込みます。
- **CL コマンド**：ユーザーセッションのクローズ ET セッションを終了し、データベースを更新します。
- **E1 コマンド**：レコードの削除／ファイルのリフレッシュ レコードを削除するか（ホールドオプション）、またはファイルをリフレッシュします。
- **ET コマンド**：トランザクションの終了 現在のトランザクションを終了し、保存します。
- **HI コマンド**：レコードのホールド 他のユーザーがレコードを更新できないようにします。
- **L1/L4 コマンド**：読み込み／読み込みおよびレコードのホールド 指定された ISN のレコードを読み込みます。ホールドされているレコードの待機／リターンコードの発行オプションを有効にして、読み込んでホールドします。
- **L2/L5 コマンド**：物理順に読み込み レコードを物理順に読み込みます。待機／リターンコードの発行オプションを有効にし、物理順に読み込んでホールドします。

- **L3/L6 コマンド：論理順に読み込み** レコードをディスクリプタ値順に読み込みます。待機/リターンコードの発行オプションを有効にして、ディスクリプタ値順に読み込みます。
- **L9 コマンド：ディスクリプタ値の読み込み** 指定されたディスクリプタの値を読み込みます。
- **LF コマンド：フィールド定義の読み込み** ファイルの全フィールドの定義を読み込みます。
- **N1/N2 コマンド：レコードの追加** Adabas が割り当てた ISN で新しいデータベースレコードを追加します。ユーザーが割り当てた ISN で新しいデータベースレコードを追加します。
- **OP コマンド：ユーザーセッションのオープン** ユーザーセッションの開始
- **RC コマンド：コマンド ID/グローバルフォーマット ID の解放** コマンドを発行したユーザーの 1 つ以上のコマンド ID またはグローバルフォーマット ID を解放します。
- **RE コマンド：ET ユーザーデータの読み込み** このユーザー、他のユーザー、または全ユーザーの ET データを読み込みます。
- **RI コマンド：レコードの解放** ホールド中のレコードおよび ISN を解放します。
- **S1、S2、S4 コマンド：レコードの検索** 検索条件を満たしているレコードの個数および ISN を返します。ユーザー指定の順序でレコードおよび ISN の個数を返します。
- **S5 コマンド：カップリングリストの検索** 指定ファイルのカップリングされた ISN のリストを返すか、または保存します。
- **S8 コマンド：ISN リストの処理** 集合演算 (AND、OR、NOT) を使用して、同じファイルからの 2 つの ISN リストを結合します。
- **S9 コマンド：ISN リストのソート** ISN リストを ISN の昇順またはディスクリプタ指定の順番でソートします。

23 A1 コマンド：レコードの更新

■ 機能および使用	182
■ ACB インターフェイスダイレクトコール：A1 コマンド	182
■ ACBX インターフェイスダイレクトコール：A1 コマンド	188
■ バッファ	193
■ その他の考慮事項	193

A1 コマンドは、レコードをホールドオプション付きで更新します。

このchapterでは、次のトピックについて説明します。

機能および使用

A1 コマンドは、レコード内の1つ以上のフィールドの値を変更するのに使用します。更新対象のフィールドがあるレコードは、ファイル番号と ISN 番号で識別されます。更新対象のフィールドをフォーマットバッファに指定し、更新後の値をレコードバッファに指定します。指定したフィールドのみが変更されます。レコード内のそれ以外のフィールドは変更されません。

アソシエータおよびデータストレージへの必要な更新は、Adabas によって行われます。

更新の前にレコードをホールド状態にするためのホールドオプションが用意されています。

マルチユーザーモードで操作している場合、A1 コマンドは更新レコードがホールド状態にある場合にのみ実行できます。



Note: 前回の Adabas リリースの A4 コマンドは A1 コマンドとして実行されます。

ACB インターフェイスダイレクトコール：A1 コマンド

このsectionでは、A1 コマンドの ACB インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの概要](#)
- [コントロールブロックフィールドの説明](#)
- [ACB の例](#)

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	--	--	--
コマンドコード	3~4	英数字	F	U
コマンド ID	5~8	英数字/バイナリ	F	U
ファイル番号	9~10	バイナリ	F	U
レスポンスコード	11~12	バイナリ	--	A
ISN	13~16	バイナリ	F	U
ISN 下限	17~20	バイナリ	--	A ¹
ISN 数	21~24	バイナリ	--	A ¹
フォーマットバッファ長	25~26	バイナリ	F	U
レコードバッファ長	27~28	バイナリ	F	U
	29~34	--	--	--
コマンドオプション 1/2	35~36	英数字	F	U
	37~44	--	--	--
アディション 2	45~48	英数字/バイナリ		A
アディション 3	49~56	英数字	F	A
アディション 4	57~64	英数字	F	A
アディション 5	65~72	英数字	F	U
コマンドタイム	73~76	バイナリ	--	A
ユーザーエリア	77~80	--	--	U

注

- これらのフィールドは使用されますが、カップリングファイルが使用されている場合は Adabas によりリセットされません。

バッファエリア

バッファ	Adabas コール前	Adabas コール後
フォーマット	F	U
レコード	F	U

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

-- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

コマンドコード (ACBCMD)

A1

コマンド ID (ACBCID)

一連の A1 コマンドを使用して複数レコードを更新し、それぞれのコールでフォーマットバッファに同一フィールドを指定する場合（検索コマンドで選択された複数レコードを更新する場合など）には、このフィールドに、空白以外の値またはゼロ以外の値をセットする必要があります。A1 コマンドを L1/L4、L2/L5、または L3/L6 コマンドと組み合わせて使用し、各レコードの同一フィールドを読み込んで更新する場合には、読み込みコマンドで使ったのと同じコマンド ID を A1 コールでも使用する必要があります。どちらの場合も、こうすることで、各 A1 コールの処理に必要な時間を短縮できます。

1 回の A1 コールで、1 レコードだけを更新する、あるいは A1 のコール間でフォーマットバッファを変更する場合は、このフィールドに空白をセットする必要があります。

このフィールドの第 1 バイトには 16 進数 'FF' にセットしないでください。

ファイル番号 (ACBFNR)

読み込むファイルの番号を 2 進数で指定します。物理ダイレクトコールの場合は、次のようファイル番号を指定します。

- 1 バイトファイル番号の場合は、ファイル番号を第 2 バイト (10) に入力します。第 1 バイト (9) は、バイナリの 0 (B'0000 0000') をセットします。
- 2 バイトファイル番号の場合は、2 バイト (9 と 10) を使います。



Note: 2 バイトファイル番号およびデータベース ID を使用する場合は、コントロールブロックの先頭バイトに X'30' を入力しなければなりません。

レスポンスコード (ACBRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、アディクション 2 フィールドの下位 2 バイトにサブコードを伴う場合があります。詳細は『Adabas メッセージおよびコード』を参照してください。

ISN (ACBISN)

更新するレコードの ISN。

ISN 数/下限 (ACBISQ および ACBISL)

これらのフィールドは、カップリングファイルが使用されていない限り、A1 コマンド処理の終了後、空値に設定されます。カップリングファイルが使用されている場合、これらのフィールドは A1 処理で使用され、リセットされません。

フォーマットバッファ長 (ACBFBL)

フォーマットバッファ長 (バイト単位)。ユーザープログラムに定義するフォーマットバッファエリアは、この指定長と同じか、それ以上でなければなりません。

レコードバッファ長 (ACBRBL)

レコードバッファ長 (バイト単位)。ユーザープログラムに定義するレコードバッファエリアは、この指定長と同じか、それ以上でなければなりません。

コマンドオプション 1 およびコマンドオプション 2：ホールドレコードオプション (ACBCOP1 および ACBCOP2)

オプション	説明
H	コマンドオプション 1 フィールドまたはコマンドオプション 2 フィールドを "H" にすると、更新前にレコードがホールド状態になります。現在レコードが別のユーザーによってホールド状態になっている場合、オプション "R" も指定しない限り、レコードが使用可能になるかトランザクションがタイムアウトになるまでコマンドは待機状態になります。
R	このオプションを指定する場合は、このオプションをコマンドオプション 1 フィールドに指定し、オプション "H" をコマンドオプション 2 フィールドに指定する必要があります。オプション "R" を指定すると、レコードがホールドされているため使用できない場合に、レスポンスコード 145 が返されます。コマンドは待機状態にはなりません。

アディション 2 - 圧縮レコードの長さ - (ACBADD2)

コマンドが正常に処理されると、次の情報がこのフィールドに返されます。

- 少なくとも 1 つの有効なフィールド値がレコードバッファにある場合、先頭 2 バイトには、アクセスしたレコードの圧縮レコード長がバイナリ形式で格納されます。
- A1 コマンドによってゼロ以外のレスポンスコードが返された場合、下位 2 バイトには、正確なレスポンスコードの意味を定義するサブコードが含まれていることがあります。レスポンスコードとサブコードについては、『Adabas メッセージおよびコード』を参照してください。

アディション 3 - パスワード - (ACBADD3)

このフィールドは、Adabas Security または ADAESI パスワードを指定するため使用します。データベース、ファイル、またはフィールドがセキュリティ保護されている場合は、有効なセキュリティパスワードまたは ADAESI パスワードをユーザーは指定する必要があります。パスワードを保護するために、コマンド処理時にアディション 3 フィールドに空白がセットされます。

アクセスしたファイルをパスワード保護する場合は、パスワードを保護するために、コマンド処理時、このフィールドに空白がセットされます。

アディション 4 - サイファコード - (ACBADD4)

このフィールドは、サイファコードを指定するために使用します。ファイルを暗号化する場合は、有効なサイファコードを指定する必要があります。ファイルを暗号化しない場合は、このフィールドに空白をセットする必要があります。

Adabasは、コマンド処理時に、サイファコードに空白をセットし、バージョンコードとデータベースIDをフィールドの下位3バイトに返します。詳細については「[コントロールブロックフィールド](#)」を参照してください。

アディション 5 - フォーマット ID、グローバルフォーマット ID - (ACBADD5)

このフィールドを使って別のフォーマット ID を指定して、このコマンドに使用される内部フォーマットバッファを識別したり、またはグローバルフォーマット ID を指定したりできます。

アディション 5 フィールドの 1 バイト目の最上位ビットに 1 が指定されていない場合は、コマンド ID フィールドに指定した値がフォーマット ID として使用されます。

このビットに 1 がセットされていると、アディション 5 フィールドの 5 バイト目から 8 バイト目までがフォーマット ID として使用されます。

さらに、アディション 5 フィールドの第 1 バイトの最上位 (左端) 2 ビットに、1 (B'11') がセットされると、アディション 5 フィールドの全 8 バイトが、グローバルフォーマット ID として使用されます (つまり、複数ユーザーがこのフォーマット ID を同時に使用できます)。

詳細と例については、「[コマンド ID、フォーマット ID、グローバルフォーマット ID](#)」を参照してください。

ACB の例

ここで記述するすべての例で使用されている Adabas ファイルの定義については、「[例で使用されているファイル定義](#)」を参照してください。

■ 例 1

■ 例 2

例 1

ファイル 1 の ISN4 が次の値で更新されます。

フィールド AA 1234

フィールド AB 20

コントロールブロック

コマンドコード	A1	
コマンド ID	bbbb (空白)	1 レコードのみが更新されます。
ファイル番号	1	
ISN	4	
フォーマットバッファ長	10	またはそれ以上
レコードバッファ長	10	またはそれ以上
アデクション 3	bbbbbbbb (空白)	ファイルはセキュリティ保護されません。
アデクション 4	bbbbbbbb (空白)	ファイルは暗号化されない

バッファエリア

フォーマットバッファ	AA,AB,2,U.
レコードバッファ	X'F1F2F3F440404040F2F0'

例 2

ファイル 2 内の一組のレコード（すでに FIND コマンドで識別済み）が次の値で更新されます。

フィールド RA ABCD

フィールド XB 80

フィールド XC 0

コントロールブロック

A1 コマンド：レコードの更新

コマンドコード	A1	
コマンド ID	ABCD	連続するレコードの同じフィールドが更新されるため、空白やゼロでないコマンド ID を指定することをお勧めします。
ファイル番号	2	
ISN	n	各コールの前に、前の FIND コマンドによって取得された各 ISN をこのフィールドに挿入します。
フォーマットバッファ長	9	またはそれ以上
レコードバッファ長	16	またはそれ以上
アデクション 3	パスワード	ファイル 2 はセキュリティ保護されています。
アデクション 4	bbbbbbbb (空白)	ファイルは暗号化されない

バッファエリア

フォーマットバッファ	RA,XB,XC.
レコードバッファ	X'C1C2C3C440404040080CF0F0F0F0F0F0'

すでに FIND コマンドで得た ISN ごとに A1 コールを繰り返します。

ACBX インターフェイスダイレクトコール：A1 コマンド

この section では、A1 コマンドの ACBX インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの概要](#)
- [コントロールブロックフィールドの説明](#)

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	バイナリ	---	---
バージョンインジケータ	3~4	バイナリ	F	U
	5~6	バイナリ	---	---
コマンドコード	7~8	英数字	F	U
	9~10	バイナリ	---	---

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
レスポンスコード	11~12	バイナリ	---	U
コマンド ID	13~16	英数字/バイナリ	F	U
データベース ID	17~20	数値	F	U
ファイル番号	21~24	数値	F	U
	25~28	---	---	---
ISN	29~32	バイナリ	F	U
	33~36	---	---	---
ISN 下限	37~40	バイナリ	---	A ¹
	41~44	---	---	---
ISN 数	45~48	バイナリ	---	A ¹
コマンドオプション 1	49	英数字	F	U
コマンドオプション 2	50	英数字	F	U
	51~64	---	---	---
アディクション 2	65~68	バイナリ		A
アディクション 3	69~76	英数字/バイナリ	F	A
アディクション 4	77~84	英数字	F	A
アディクション 5	85~92	英数字/バイナリ	F	U
	93~114	---	---	---
エラーサブコード	115~116	バイナリ	---	A
	117~144	---	---	---
コマンドタイム	145~152	バイナリ	---	A
ユーザーエリア	153~168	該当なし	---	U
	169~193	---	---	---

注

- これらのフィールドは使用されますが、カップリングファイルが使用されている場合は Adabas によりリセットされません。

ABD とバッファ

ABD とバッファ	Adabas コール前	Adabas コール後
フォーマット	F	U
レコード	F	U

上記の意味は次に示すとおりです。

- F Adabas コール前にユーザーが入力するフィールド
- A Adabas により入力されるフィールド
- U Adabas コール後も変化なし
- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

バージョンインジケータ (ACBXVER)

F2

コマンドコード (ACBXCMD)

A1

レスポンスコード (ACBXRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、エラーサブコード (ACBXERRC) フィールドにサブコードを伴う場合があります。詳細は、*Adabas* メッセージおよびコードのドキュメントを参照してください。

コマンド ID (ACBXCID)

一連の A1 コマンドを使用して複数レコードを更新し、それぞれのコールでフォーマットバッファに同一フィールドを指定する場合（検索コマンドで選択された複数レコードを更新する場合など）には、このフィールドに、空白以外の値またはゼロ以外の値をセットする必要があります。A1 コマンドを L1/L4、L2/L5、または L3/L6 コマンドと組み合わせて使用し、各レコードの同一フィールドを読み込んで更新する場合には、読み込みコマンドで使ったのと同じコマンド ID を A1 コールでも使用する必要があります。どちらの場合も、こうすることで、各 A1 コールの処理に必要な時間を短縮できます。

1 回の A1 コールで、1 レコードだけを更新する、あるいは A1 のコール間でフォーマットバッファを変更する場合は、このフィールドに空白をセットする必要があります。

このフィールドの第 1 バイトには 16 進数 'FF' にセットしないでください。

データベース ID (ACBXDBID)

このフィールドを使用して、データベース ID を指定します。Adabas コールはこのデータベースに送られます。

このフィールドは 4 バイトのバイナリフィールドですが、現時点では 2 バイトのデータベース ID のみがサポートされています。したがって、データベース ID はフィールドの下位部分（右端のバイト）に指定し、その前にバイナリの 0 を付加する必要があります。

このフィールドがバイナリの 0 に設定されている場合は、Adabas API は DDCARD 入力データで指定された ADARUN カードのデータベース ID か、リンクされているカリクルーチ

ンによりロードされた LNKGBLS モジュールで指定されているデフォルトのデータベース ID 値のいずれかを使用します。

ファイル番号 (ACBXFNFR)

このフィールドを使用して、Adabas コールで送る必要のあるファイル数を指定します。

このフィールドは4バイトのバイナリフィールドですが、ファイル番号はフィールドの下位部分（右端のバイト）に指定し、その前にバイナリの0を付加する必要があります。

ISN (ACBXISN)

このフィールドは、更新対象のレコードの ISN を指定するのに使用します。

ACBXISN フィールドは、8バイトの ACBXISNG フィールドに埋め込まれた4バイトのバイナリフィールドで、まだ使用されていません。ACBXISNG フィールドの上位部分には、バイナリの0を設定します。

ISN 下限 (ACBXISL)

このフィールドは、ハードカップリングファイルが使用されない限り、A1 コマンド処理終了後、空値にセットされます。カップリングファイルが使用されている場合は、このフィールドは A1 コマンド処理中に使用され、リセットされません。

ISN 数 (ACBXISQ)

このフィールドは、ハードカップリングファイルが使用されない限り、A1 コマンド処理終了後、空値にセットされます。カップリングファイルが使用されている場合は、このフィールドは A1 コマンド処理中に使用され、リセットされません。

コマンドオプション1およびコマンドオプション2：ホールドレコードオプション (ACBXCOP1 および ACBXCOP2)

オプション	説明
H	コマンドオプション1フィールドまたはコマンドオプション2を"H"にすると、更新前にレコードがホールド状態になります。現在レコードが別のユーザーによってホールド状態になっている場合、オプション"R"も指定しない限り、レコードが使用可能になるかトランザクションがタイムアウトになるまでコマンドは待機状態になります。
R	このオプションを指定する場合は、このオプションをコマンドオプション1フィールドに指定し、オプション"H"をコマンドオプション2フィールドに指定する必要があります。オプション"R"を指定すると、レコードがホールドされているため使用できない場合に、レスポンスコード145が返されます。コマンドは待機状態にはなりません。

アディション2 - 圧縮レコードの長さ - (ACBXADD2)

コマンドが正常に処理され、かつ少なくとも1つの有効なフィールド値がレコードバッファにある場合、先頭2バイトには、アクセスしたレコードの圧縮レコード長がバイナリ形式で格納されます。

アディション3 - パスワード - (ACBXADD3)

このフィールドは、Adabas Security または ADAESI パスワードを指定するため使用します。データベース、ファイル、またはフィールドがセキュリティ保護されている場合は、有効なセキュリティパスワードまたは ADAESI パスワードをユーザーは指定する必要があります。

パスワードを保護するために、コマンド処理時にアディション3フィールドに空白がセットされます。

アクセスしたファイルをパスワード保護する場合は、パスワードを保護するために、コマンド処理時、このフィールドに空白がセットされます。

アディション4 - サイファコード - (ACBXADD4)

このフィールドは、サイファコードを指定するために使用します。ファイルを暗号化する場合は、有効なサイファコードを指定する必要があります。ファイルを暗号化しない場合は、このフィールドに空白をセットする必要があります。

Adabasは、コマンド処理時に、サイファコードに空白をセットし、バージョンコードとデータベースIDをフィールドの下位3バイトに返します。詳細については「[コントロールブロックフィールド](#)」を参照してください。

アディション5 - フォーマット ID、グローバルフォーマット ID - (ACBXADD5)

このフィールドを使って別のフォーマット ID を指定して、このコマンドに使用される内部フォーマットバッファを識別したり、またはグローバルフォーマット ID を指定したりできます。

アディション5フィールドの1バイト目の最上位ビットに1が指定されていない場合は、コマンド ID フィールドに指定した値がフォーマット ID として使用されます。

このビットに1がセットされていると、アディション5フィールドの5バイト目から8バイト目までがフォーマット ID として使用されます。

さらに、アディション5フィールドの第1バイトの最上位（左端）2ビットに、1 (B'11') がセットされると、アディション5フィールドの全8バイトが、グローバルフォーマット ID として使用されます（つまり、複数ユーザーがこのフォーマット ID を同時に使用できます）。

詳細と例については、「[コマンドID、フォーマットID、グローバルフォーマットID](#)」を参照してください。

エラーサブコード (ACBXERRC)

コマンドがゼロ以外のレスポンスコードを返したときは、このフィールドにレスポンスコードの正確な意味を定義したサブコードが含まれます。レスポンスコードとサブコードについては、『[Adabas メッセージおよびコード](#)』を参照してください。

バッファ

A1 コマンドでは次のバッファを指定する必要があります。

- フォーマットバッファ
- レコードバッファ

フォーマットバッファ

このバッファには、更新するフィールドを指定する必要があります。A1 コマンドを実行する場合、フォーマットバッファに次のものを含めることはできません。

- フィールドの選択条件
- 編集マスクエレメント
- サブディスクリプタフィールドまたはスーパーディスクリプタフィールドへの参照
- 同じフィールドの複数回指定（マルチプルバリューフィールドを除く）
- マルチバリューフィールドやピリオディックグループ内のフィールドに対する "-N" 形式の指定（例えば、ABN や AB1-N）

フォーマットバッファに上記のいずれかが含まれていると、A1 コマンドに対してニュークリアレスレスポンス 44 が返されます。

フォーマットバッファの構文および例については、「[フォーマットバッファ](#)」を参照してください。

レコードバッファ

このバッファには、フォーマットバッファに指定した長さフォーマットに従って、更新する値が入ります。詳細については、「[レコードバッファ](#)」を参照してください。

その他の考慮事項

A1 コマンドを使用する場合、次のことについて考慮します。

1. サブディスクリプタ、スーパーディスクリプタ、およびフォネティックディスクリプタは、直接更新できません。これらのディスクリプタを更新するには、そのディスクリプタの生成に使用されたフィールドを更新する必要があります。対応するサブディスクリプタ、スーパーディスクリプタ、またはフォネティックディスクリプタ値は Adabas により更新されます。
2. 圧縮前の最大レコード長は、理論上は 32,767 バイトですが、実際にはブロックサイズによって制限されます。また、この値は、Adabas セッションの LU パラメータの大きさによって

は、それより小さくなります。この場合の最大レコード長は、LU 値 - フォーマットバッファ長 - 108 になります。圧縮後の最大レコード長は、データストレージブロックサイズ - 4 バイトまたは WORK ブロックサイズ - 110 バイトのどちらか小さい方です。

3. ディスクリプタ値は、253 バイト以下にする必要があります。
4. 標準長を超える長さを使用（フィールドが固定ストレージオプションで定義されている場合は許可されない）してフィールドを更新した場合、その後、そのフィールドを参照するときは、更新に使用された長さを指定する必要があります。以後の参照で標準長を使用すると、英数字フィールドでは桁落ちが発生し、数値フィールドではレスポンスコード 55 が返されます。
5. フォーマットバッファには、マルチプルバリュースフィールドまたはピリオディックグループ内のフィールドを複数回指定できます。マルチプルバリュースフィールドは、インデックスなしで数回指定したり、インデックスをさまざまに変えて指定したりできます。ピリオディックグループ内のフィールドは、常にインデックスが異なる値である必要があります。
6. フォーマットバッファにマルチプルバリュースまたはピリオディックグループのカウントフィールドを指定しても、Adabas では無視されます。レコードバッファの対応する値も無視されます。Adabas ではフォーマットバッファのリテラルは無視され、レコードバッファの対応する位置も無視されます。
7. マルチプルバリュースフィールドを更新し、そのフィールドカウントを変更する必要がある場合、Adabas では、次のルールに従って、マルチプルバリュースカウントフィールドを更新します。
 - 空値省略 (NU) オプションで定義したマルチプルバリュースフィールドの場合、カウントフィールドは現在空値でない値の数に調整されます。空値はすべて削除（前づめ）されます。

フィールド定義	01,MF,5,A,MU,NU
更新前の MF 値	XXXXX,YYYYY
フォーマットバッファ	MF4
レコードバッファ	ZZZZZ
更新後の結果	XXXXX,YYYYY,ZZZZZ MF カウント = 3
更新前の MF 値	XXXXX,YYYYY,ZZZZZ
フォーマットバッファ	MF2
レコードバッファ	bbbbbb (空白)
更新後の結果	XXXXX,ZZZZZ MF カウント = 2
更新前の MF 値	XXXXX,ZZZZZ
フォーマットバッファ	MF1-2
レコードバッファ	bbbbbbbbbb (空白)
更新後の結果	値は省略されます。 MF カウント = 0

- NU オプションなしで定義されたマルチプルバリューフィールドの場合、カウントは現在存在する値の数（空値を含む）を反映するように調整されます。

フィールド定義	01,MF,5,A,MU
更新前の MF 値	XXXXX,YYYYY
フォーマットバッファ	MF4
レコードバッファ	DDDDD
更新後の結果	XXXXX,YYYYY,b (空白) ,DDDDD MF カウント = 4
更新前の MF 値	XXXXX,YYYYY,ZZZZZ
フォーマットバッファ	MF3
レコードバッファ	bbbbbb (空白)
更新後の結果	XXXXX,YYYYY,b (空白) MF カウント = 3

マルチプルバリューフィールドでは、最大 191 個の値が許可されています。

ただし、フォーマットバッファにインデックス指定のないマルチプルバリューフィールドが含まれている場合には、以上のルールは適用されません。この場合、指定した新しい値のみが使用され、それ以外の値は値の内容にかかわらず削除されます。

フィールド定義	01,MF,5,A,MU,NU
更新前の MF 値	XXXXX,YYYYY
フォーマットバッファ	MF
レコードバッファ	AAAAA
更新後の結果	AAAAA MF カウント = 1
更新前の MF 値	XXXXX,YYYYY
フォーマットバッファ	MF1
レコードバッファ	AAAAA
更新後の結果	AAAAA,YYYYY MF カウント = 2
更新前の MF 値	XXXXX,YYYYY,ZZZZZ
フォーマットバッファ	MF
レコードバッファ	bbbbbb (空白)
更新後の結果	値は省略されます。 MF カウント = 0

8. ピリオディックグループに属する1つ以上のフィールドを更新する場合、Adabas では必要に応じて次のルールに従ってピリオディックグループカウントを更新します。

カウントは、フォーマットバッファに指定した最高オカレンス番号に合わせて調整されます（指定したオカレンス番号が現在の最高オカレンス番号よりも大きい場合）。

フィールド定義	01,GB,PE 02,BA,1,B,DE,NU 02,BB,5,P,NU
更新前の GB 値	GB (第 1 オカレンス) BA = 5 BB = 20 GB (第 2 オカレンス) BA = 6 BB = 25 GB カウント = 2
フォーマットバッファ	GB4
レコードバッファ	X'08000000500F'
更新後の結果	GB (第 1 オカレンス) BA = 5 BB = 20 GB (第 2 オカレンス) BA = 6 BB = 25 GB (第 3 オカレンス) BA = 0 BB = 0 GB (第 4 オカレンス) BA = 8 BB = 500 GB カウント = 4
更新前の GB 値	GB (第 1 オカレンス) BA = 5 BB = 20 GB (第 2 オカレンス) BA = 6 BB = 25 GB カウント = 2
フォーマットバッファ	GB1
レコードバッファ	X'00000000000F'
更新後の結果	GB (第 1 オカレンス) BA = 0 BB = 0 GB (第 2 オカレンス) BA = 6 BB = 25 GB カウント = 2

ピリオディックグループでは、最大 191 個のオカレンスが許可されています。

9. 可変長（標準長なし）に定義されたフィールドをフォーマットバッファに指定する場合、関連するレコードバッファ内の値の前に、1バイトのバイナリ長の値（長さバイトを含む）を指定する必要があります。

フィールド定義	01,AA,3,A 01,AB,A
フォーマットバッファ	AA,AB.
レコードバッファ	X'F1F2F306F1F2F3F4F5'

フィールド AA と AB が更新されます。AA の新しい値は "123" です。AB（可変長フィールド）の新しい値は "12345" です。

24 BT コマンド：トランザクションのバックアウト

■ 機能および使用	200
■ ACB インターフェイスダイレクトコール：BT コマンド	200
■ ACBX インターフェイスダイレクトコール：BT コマンド	204
■ ISN バッファ	208

BT コマンドでは、ET ロジックユーザーのデータベース更新を取り消します。

このchapterでは、次のトピックについて説明します。

機能および使用

BT コマンドは、ユーザーの現在の論理トランザクション内で行われた全データベース更新処理（追加、削除、変更）を取り消すときに使用します。この機能は、プログラムエラーまたはトランザクションを正常に処理できない場合に必要となります。BT コマンドは、ET ロジックユーザーのみが使用できます。

Adabas は、BT コマンド処理の最後に内部 ET コマンドを発行します。これにより、現在のデータプロテクションブロックが Adabas ワークおよびデータプロテクションログに物理的に書き込まれ、トランザクション処理中にホールドされた全レコードが解放されます。

コマンドオプション 1 フィールドのオプション ("P") により、**ISN バッファ**内の全レコードをホールド状態に保ちます。マルチフェッチ ("M") オプションは、現在のトランザクションによってホールドされているレコードのサブセットを解放します。ホールド状態から解放されたレコードを **ISN バッファ**に指定します。バッファ内の最初のフルワードは次の 8 バイトのエレメントの数を指定します。

コマンドオプション 2 フィールドのオプション ("F") により、特定のファイルをバックアウト処理から除外できます。指定したファイルに対する更新は取り消されません。ただし、このユーザーがホールドしていたファイル内のレコードはすべてホールド状態から解放されます。

ACB インターフェイスダイレクトコール：BT コマンド

このsectionでは、BT コマンドの ACB インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- **コントロールブロックとバッファの概要**
- **コントロールブロックフィールドの説明**

■ ACB の例

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	--	--	--
コマンドコード	3~4	英数字	F	U
コマンド ID	5~8	バイナリ	--	A
ファイル番号 *	9~10	バイナリ	F *	U
レスポンスコード	11~12	バイナリ	--	A
	13~16	--	--	--
ISN 下限	17~20	バイナリ	F	U
	21-32	--	--	--
ISN バッファ長 **	33~34	バイナリ	F **	U
コマンドオプション 1	35	英数字	F	U
コマンドオプション 2	36	英数字	F	U
	37~72	--	--	--
コマンドタイム	73~76	バイナリ	--	A
ユーザーエリア	77~80	--	--	U

バッファエリア

バッファ	Adabas コール前	Adabas コール後
ISN **	F	U

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

* コマンドオプション 2 が指定された場合にのみ必要

** コマンドオプション 1 が指定された場合にのみ必要

-- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

コマンドコード (ACBCMD)

BT

コマンド ID (ACBCID)

Adabas は、バックアウトされたトランザクションのトランザクションシーケンス番号をこのフィールドに返します。この番号はバイナリ形式です。

ファイル番号 (ACBFNR)

あるファイルをバックアウト処理から除外する場合は、除外するファイルの番号をこのフィールドに指定し、オプション F をコマンドオプション 2 フィールドに指定します。

除外するファイルがない（オプション F が指定されていない）場合、このフィールドは無視されます。



Note: 2 バイトファイル番号およびデータベース ID を使用する場合は、コントロールブロックの先頭バイトに 'X'30' を入力しなければなりません。

レスポンスコード (ACBRSP)

Adabas は、コマンドのレスポンスコードをこのフィールドに返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。BT コマンドによってゼロ以外のレスポンスコードが返された場合、Adabas コントロールブロックの下位 2 バイト、バイト 45~48（アディション 2 フィールド）にレスポンスコードの意味を正確に定義したサブコードが含まれていることがあります。レスポンスコードとサブコードについては、『Adabas メッセージおよびコード』を参照してください。

ISN 下限 (ACBISL)

ISN ホールドオプションを指定した場合、6 バイトの ISN バッファエントリのカウントをこのフィールドに入れる必要があります。

ISN バッファ長 (ACBIBL)

ISN バッファ長（バイト単位）。この長さは、ISN ホールドオプションまたはマルチフェッチオプションを使用するときのみ必要です（コマンドオプション 1 フィールドの説明を参照）。マルチフェッチ機能を指定する場合は、この値を 32KB より小さくする必要があります。

コマンドオプション 1：ISN ホールドオプション (ACBCOP1)



Note: ADARUN PREFETCH=YES でマルチフェッチを設定すると、ET/BT コマンドに対して "P" オプションが自動的に使われます（他のすべてのコマンドに対しては "M" オプションが自動的に使われます）。このフィールドを使用して、このオプション設定を変更することはできません。

デフォルトでは、BT コマンドを実行すると、Adabas はユーザーが現在ホールドしている全 ISN を解放します。

オプション	説明
P	これらの ISN のすべてまたは一部をホールド状態に戻します。ホールド状態に戻る ISN は ISN バッファ に指定する必要があります。また、ISN カウントは ISN 下限フィールドに指定する必要があります。
M (コマンドレベルのマルチフェッチ)	現在のトランザクションによってホールドされているすべての ISN の代わりにサブセットのみを開放します。ホールド状態から解放するレコードは ISN バッファ に指定します。バッファ内の最初のフルワードには、後続の 8 バイトの要素の数を指定します。

コマンドオプション 2：ファイル除外オプション (ACBCOP2)

オプション	説明
F (除外ファイル)	ファイル番号フィールドに指定された単一ファイルをバックアウト処理から除外します。指定されたファイルの更新はバックアウトされません。ただし、ホールド状態のファイルのレコードは解放されます。
空白	全ファイルをバックアウト処理の対象とします。

ACB の例

- 例 1
- 例 2
- 例 3

例 1

現在のユーザートランザクションをバックアウトします。全ファイルにバックアウト処理を行います。

コントロールブロック

コマンドコード	BT	
コマンドオプション 1	空白	ホールドする ISN はありません。
コマンドオプション 2	b	ファイル除外オプションを使用しない

例 2

現在のユーザートランザクションをバックアウトします。ファイル 4 に行った更新は、バックアウト処理に含まれません。

コントロールブロック

コマンドコード	BT	
ファイル番号	4	ファイル 4 はバックアウト処理から除外します。
コマンドオプション 2	F	ファイル除外オプションを使用します。

例 3

現在のユーザートランザクションをバックアウトします。ファイル 6 の ISN 1、2、3 をホールド状態にします。

コントロールブロック

コマンドコード	BT	
コマンドオプション 1	P	ISN（複数も可）をホールド状態にします。
コマンドオプション 2	b	ファイル除外オプションを使用しない

バッファエリア

ISN バッファ	
000600000001	ISN 1
000600000002	ISN 2
000600000003	ISN 2

ACBX インターフェイスダイレクトコール：BT コマンド

このsectionでは、BT コマンドの ACBX インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの概要](#)

■ コントロールブロックフィールドの説明

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	---	---	---
バージョンインジケータ	3~4	バイナリ	F	U
	5~6	バイナリ	---	---
コマンドコード	7~8	英数字	F	U
	9~10	---	---	---
レスポンスコード	11~12	バイナリ	---	A
コマンド ID	13~16	英数字/バイナリ	---	A
データベース ID*	17~20	数値	F*	U
ファイル番号*	21~24	数値	F*	U
	25~36	---	---	---
ISN 下限	37~40	バイナリ	F	U
	45~48	---	---	---
コマンドオプション 1	49	英数字	F	U
コマンドオプション 2	50	英数字	F	U
	51~114	---	---	---
エラーサブコード	115~116	バイナリ	---	A
	117~144	---	---	---
コマンドタイム	145~152	バイナリ	---	A
ユーザーエリア	153~168	該当なし	---	U
	169~193	---	---	---

ABD とバッファ

ABD とバッファ	Adabas コール前	Adabas コール後
ISN **	F	U

上記の意味は次に示すとおりです。

- F Adabas コール前にユーザーが入力するフィールド
- A Adabas により入力されるフィールド
- U Adabas コール後も変化なし
- * コマンドオプション 2 が指定された場合にのみ必要
- ** コマンドオプション 1 が指定された場合にのみ必要
- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

バージョンインジケータ (ACBXVER)

F2

コマンドコード (ACBXCMD)

BT

レスポンスコード (ACBXRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、エラーサブコード (ACBXERRC) フィールドにサブコードを伴う場合があります。詳細は、*Adabas* メッセージおよびコードのドキュメントを参照してください。

コマンド ID (ACBXCID)

Adabas は、バックアウトされたトランザクションのトランザクションシーケンス番号をこのフィールドに返します。この番号はバイナリ形式です。

データベース ID (ACBXDBID)

このフィールドは、バックアウト処理から除外するファイルのデータベース ID を指定するのに使用します。Adabas コールはこのデータベースに送られます。

このフィールドは 4 バイトのバイナリフィールドですが、現時点では 2 バイトのデータベース ID のみがサポートされています。したがって、データベース ID はフィールドの下位部分 (右端のバイト) に指定し、その前にバイナリの 0 を付加する必要があります。

このフィールドがバイナリの 0 に設定されている場合は、Adabas API は DDCARD 入力データで指定された ADARUN カードのデータベース ID か、リンクされているカリクルーチンによりロードされた LNKGBLS モジュールで指定されているデフォルトのデータベース ID 値のいずれかを使用します。

ファイル番号 (ACBXFNR)

このフィールドは、バックアウト処理から除外するファイルの番号を指定するのに使用します (オプション F をコマンドオプション 2 フィールドに指定する必要があります)。

このフィールドは4バイトのバイナリフィールドですが、ファイル番号はフィールドの下位部分（右端のバイト）に指定し、その前にバイナリの0を付加する必要があります。

ファイルが除外されない（オプションFをコマンドオプション2に指定しない）場合、ファイル番号フィールドに指定した値は無視されます。

ISN 下限（ACBXISL）

ISN ホールドオプションを指定した場合、6バイトのISNバッファエントリのカウントをこのフィールドに入れる必要があります。

ACBXISL フィールドは、8バイトのACBXISLGフィールドに埋め込まれた4バイトのバイナリフィールドで、まだ使用されていません。ACBXISLGフィールドの上位部分には、バイナリの0を設定します。

コマンドオプション1：ISN ホールドオプション（ACBXCOP1）

デフォルトでは、BT コマンドを実行すると、Adabas はユーザーが現在ホールドしている全ISNを解放します。

オプション	説明
P	これらのISNのすべてまたは一部をホールド状態に戻します。ホールド状態に戻るISNはISNバッファに指定する必要があります。また、ISNカウントはISN下限フィールドに指定する必要があります。
M（コマンドレベルのマルチフェッチ）	現在のトランザクションによってホールドされているすべてのISNの代わりにサブセットのみを開放します。ホールド状態から解放されたレコードをISNバッファに指定します。バッファ内の最初のフルワードは次の8バイトのエLEMENTの数を指定します。

コマンドオプション2：ファイル除外オプション（ACBXCOP2）

オプション	説明
F（除外ファイル）	ファイル番号フィールドに指定された単一ファイルをバックアウト処理から除外します。指定されたファイルの更新はバックアウトされません。ただし、ホールド状態のファイルのレコードは解放されます。
空白	全ファイルをバックアウト処理の対象とします。

エラーサブコード（ACBXERRC）

コマンドがゼロ以外のレスポンスコードを返したときは、このフィールドにレスポンスコードの正確な意味を定義したサブコードが含まれます。レスポンスコードとサブコードについては、『Adabas メッセージおよびコード』を参照してください。

ISN バッファ

コマンドオプション1フィールドに "P" を設定した場合、レコードをホールド状態に戻す各 ISN を6バイトバイナリのエントリとして指定する必要があります。この6バイトの内容は、次のとおりです。

- 先頭2バイトには、レコードが存在するファイルの番号を指定します。
- 残り4バイトには、ホールド状態にするレコードの ISN を指定します。

コマンドオプション1フィールドに "M" をセットした場合、現在のトランザクションによってホールドされているレコードのサブセットだけが解放されます。ISN バッファ内の最初のフルワードは、8バイトのエLEMENTの数になります。後続の各8バイトグループは、ホールド状態から解放されるレコードのファイル番号/ISN ID であると解釈されます（「[BT/ET のマルチフェッチ処理](#)」参照）。

25 C1 コマンド：チェックポイントの書き込み

- 機能および使用 210
- ACB インターフェイスダイレクトコール：C1 コマンド 210
- ACBX インターフェイスダイレクトコール：C1 コマンド 213

C1 コマンドは、コマンド ID、PLOG、RABN チェックポイントおよびバッファフラッシュオプションを書き込みます。

このchapterでは、次のトピックについて説明します。

機能および使用

C1 コマンドを使用して、チェックポイントを取ることを要求できます。

通常、C1 コマンドは、排他制御更新ユーザー（ET ロジックを使用していないユーザー）またはシングルユーザーモードで動作しているユーザーのみが発行します。

排他ファイル制御更新を要求するユーザープログラムの冒頭で、Adabas は内部的に C1 コマンドを自動的に発行します。

C1 コマンドの結果、Adabas チェックポイントテーブルにチェックポイントエントリが書き込まれます。このチェックポイントエントリには次の特性があります。

- チェックポイント ID（コマンド ID フィールドにユーザーが指定した値）と、現在のデータプロテクションログおよびブロック番号が含まれています。
- このチェックポイントエントリを使って、データベース（あるいは特定ファイル）をチェックポイントが取得されたときの状態に実際に戻すことができます。排他制御更新処理するプログラムを再実行または再スタートするためには、事前にチェックポイントまで戻さなければなりません。

コマンドオプション 1 フィールドまたはコマンドオプション 2 フィールドに "F"（バッファフラッシュ）オプションを指定すると、コマンド処理の最後に Adabas バッファプールの内容が外部ストレージにフラッシュされます。

ACB インターフェイスダイレクトコール：C1 コマンド

このsectionでは、C1 コマンドの ACB インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの概要](#)
- [コントロールブロックフィールドの説明](#)

■ ACB の例

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	--	--	--
コマンドコード	3~4	英数字	F	U
コマンド ID	5~8	英数字	F	A
ファイル番号 *	9~10	バイナリ	F	U
レスポンスコード	11~12	バイナリ	--	A
	13~34	--	--	--
コマンドオプション 1	35	英数字	F	U
コマンドオプション 2	36	英数字	F	U
	37~72	--	--	--
コマンドタイム	73~76	バイナリ	--	A
ユーザーエリア	77~80	--	--	U

バッファエリア

何も使用されていません。

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

* データベース ID は、複数のデータベースにアクセスして、特定のデータベースに対してコマンド処理を制限する必要がある場合にのみ必要になります。

-- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

コマンドコード (ACBCMD)

C1

コマンド ID (ACBCID)

このフィールドには、空白とゼロ以外の値を指定しなければなりません。この値によりチェックポイントが識別されます。各チェックポイントに指定する値は、ユニークにする必要はありません。

"0000" または "SYNC" の値は指定できません。

このフィールドの第 1 バイトには 16 進数 'FF' にセットしないでください。

ファイル番号 (ACBFNR)

データベース ID は、複数のデータベースにアクセスして、特定のデータベースに対してコマンド処理を制限する必要がある場合にのみ必要になります。

レスポンスコード (ACBRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。C1 コマンドによってゼロ以外のレスポンスコードが返された場合、Adabas コントロールブロックのバイト 45~48 (アディクション 2 フィールドの下位 2 バイト) に、レスポンスコードの意味を正確に定義したサブコードが含まれていることがあります。レスポンスコードとサブコードについては、『Adabas メッセージおよびコード』を参照してください。

コマンドオプション 1 およびコマンドオプション 2：フラッシュバッファオプション (ACBCOP1 および ACBCOP2)

オプション	説明
F (フラッシュバッファ)	コマンドオプション 1 フィールドまたはコマンドオプション 2 フィールドにこのオプションを指定すると、コマンド処理の最後に Adabas バッファプールの内容が外部ストレージにフラッシュされます。

ACB の例

チェックポイントエントリを書き込みます。チェックポイントは値 "UCP4" で識別します。

コントロールブロック

コマンドコード	C1	
コマンド ID	UCP4	チェックポイント ID

ACBX インターフェイスダイレクトコール：C1 コマンド

このsectionでは、C1 コマンドの ACBX インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの概要](#)
- [コントロールブロックフィールドの説明](#)

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	---	---	---
バージョンインジケータ	3~4	バイナリ	F	U
	5~6	---	---	---
コマンドコード	7~8	英数字	F	U
	9~10	---	---	---
レスポンスコード	11~12	バイナリ	---	A
コマンド ID	13~16	英数字/バイナリ	F	A
データベース ID*	17~20	数値	F	U
	21~48	---	---	---
コマンドオプション 1	49	英数字	F	U
コマンドオプション 2	50	英数字	F	U
	51~114	---	---	---
エラーサブコード	115~116	バイナリ	---	A
	117~144	---	---	---
コマンドタイム	145~152	バイナリ	---	A
ユーザーエリア	153~168	該当なし	---	U
	169~193	---	---	---

ABD とバッファ

何も使用されていません。

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

* データベース ID は、複数のデータベースにアクセスして、特定のデータベースに対してコマンド処理を制限する必要がある場合にのみ必要になります。

--- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

バージョンインジケータ (ACBXVER)

F2

コマンドコード (ACBXCMD)

C1

レスポンスコード (ACBXRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、エラーサブコード (ACBXERRC) フィールドにサブコードを伴う場合があります。詳細は、Adabas メッセージおよびコードのドキュメントを参照してください。

コマンド ID (ACBXCID)

このフィールドには、空白とゼロ以外の値を指定しなければなりません。この値によりチェックポイントが識別されます。各チェックポイントに指定する値は、ユニークにする必要はありません。

"0000" または "SYNC" の値は指定できません。

このフィールドの第 1 バイトには 16 進数 'FF' にセットしないでください。

データベース ID (ACBXDBID)

このフィールドは、複数のデータベースにアクセスして、特定のデータベースに対するコマンド処理を制限する必要がある場合にのみデータベース ID を指定します。Adabas コールはこのデータベースに送られます。

このフィールドは 4 バイトのバイナリフィールドですが、現時点では 2 バイトのデータベース ID のみがサポートされています。したがって、データベース ID はフィールドの下位部分 (右端のバイト) に指定し、その前にバイナリの 0 を付加する必要があります。

このフィールドがバイナリの0に設定されている場合は、Adabas APIはDDCARD入力データで指定された ADARUN カードのデータベース ID か、リンクされているかリンクルーチンによりロードされた LNKGBLS モジュールで指定されているデフォルトのデータベース ID 値のいずれかを使用します。

コマンドオプション1およびコマンドオプション2：フラッシュバッファオプション（ACBXCOP1 および ACBXCOP2）

オプション	説明
F（フラッシュバッファ）	コマンドオプション1フィールドまたはコマンドオプション2フィールドにこのオプションを指定すると、コマンド処理の最後に、Adabasバッファプールの内容が外部ストレージにフラッシュされます。

エラーサブコード（ACBXERRC）

コマンドがゼロ以外のレスポンスコードを返したときは、このフィールドにレスポンスコードの正確な意味を定義したサブコードが含まれます。レスポンスコードとサブコードについては、『Adabas メッセージおよびコード』を参照してください。

26 C3 コマンド：チェックポイントの書き込み

■ 機能および使用	218
■ ACB インターフェイスダイレクトコール：C3 コマンド	218
■ ACBX インターフェイスダイレクトコール：C3 コマンド	221
■ バッファ	223

C3 コマンドは、Adabas チェックポイントファイルに SYNX-03 チェックポイントを書き込みます。

このchapterでは、次のトピックについて説明します。

機能および使用

C3 コマンドは、排他制御更新ユーザー（ET ロジックを使用していないユーザー）のみが発行できます。

C3 コマンドの主な機能は、Adabas チェックポイントファイルに SYNX-03 チェックポイントを書き込むことです。このチェックポイントエントリには次の特性があります。

- 現在のデータプロテクションログおよびブロック番号が含まれています。
- このチェックポイントエントリを使って、データベース（あるいは特定ファイル）をチェックポイントが取得されたときの状態に実際に戻すことができます。排他制御更新処理するプログラムを再実行または再スタートするためには、事前にチェックポイントまで戻さなければなりません。

コマンドオプション2を指定すると、再スタート用にユーザーデータも Adabas チェックポイントファイルに格納されます。格納されたデータは、後続の OP コマンドまたは RE コマンドで読み取れます。

ACB インターフェイスダイレクトコール：C3 コマンド

このsectionでは、C3 コマンドの ACB インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの概要](#)
- [コントロールブロックフィールドの説明](#)
- [ACB の例](#)

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	--	--	--
コマンドコード	3~4	英数字	F	U
	5~8	--	--	--
ファイル番号**	9~10	バイナリ	F	U
レスポンスコード	11~12	バイナリ	--	A
	13~26	--	--	--
レコードバッファ長	27~28	バイナリ	F	U
	29~35	--	--	--
コマンドオプション 2	36	英数字	F	U
	37~72	--	--	--
コマンドタイム	73~76	バイナリ	--	A
ユーザーエリア	77~80	--	--	U

バッファエリア

バッファ	Adabas コール前	Adabas コール後
フォーマット	*	
レコード	F	U

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

* 使用しませんが、コールステートメントのパラメータリストに含める必要があります。

** データベースIDは、複数のデータベースにアクセスして、特定のデータベースに対してコマンド処理を制限する必要がある場合にのみ必要になります。

-- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

コマンドコード (ACBCMD)

C3

ファイル番号 (ACBFNR)

データベース ID は、複数のデータベースにアクセスして、特定のデータベースに対してコマンド処理を制限する必要がある場合にのみ必要になります。

レスポンスコード (ACBRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。

レスポンスコードとサブコードについては、『Adabas メッセージおよびコード』を参照してください。

レコードバッファ長 (ACBRBL)

このフィールドは、コマンドオプション 2 を指定する場合にのみ使用します。

格納するユーザーデータのバイト数をこのフィールドで指定する必要があります。

指定可能な最大長は 2000 バイトです。

コマンドオプション 2：ユーザーデータ格納 (ACBCOP2)

オプション	説明
E	ユーザーデータがセッション終了時に Adabas チェックポイントファイルに書き込まれます。空白以外のユニークなユーザー ID を指定してユーザーセッションで OP コマンドを発行している場合に限り、このオプションを使用できます。

ACB の例

ユーザープログラムが C3 コマンドを発行し、Adabas チェックポイントファイルに格納されるユーザーデータを提供します。

コントロールブロック

コマンドコード	C3	
レコードバッファ長	17	17 バイトのユーザーデータを格納
コマンドオプション 2	E	ユーザーデータを格納

バッファ

レコードバッファ	EXU-USER ET-DATA
----------	------------------

ACBX インターフェイスダイレクトコール：C3 コマンド

このsectionでは、C3 コマンドの ACBX インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの概要](#)
- [コントロールブロックフィールドの説明](#)

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	---	---	---
バージョンインジケータ	3~4	バイナリ	F	U
	5~6	---	---	---
コマンドコード	7~8	英数字	F	U
	9~10	---	---	---
レスポンスコード	11~12	バイナリ	---	A
---	13~16	---	---	---
データベース ID**	17~20	数値	F	U
	21~49	---	---	---
コマンドオプション 2	50	英数字	F	U
	51~114	---	---	---
エラーサブコード	115~116	バイナリ	---	A
	117~144	---	---	---
コマンドタイム	145~152	バイナリ	---	A
ユーザーエリア	153~168	該当なし	---	U
	169~193	---	---	---

ABD とバッファ

ABD とバッファ	Adabas コール前	Adabas コール後
フォーマット	*	
レコード	F	U

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

* 使用しませんが、Adabas コールに含める必要があります。含めない場合は、Adabas コールが自動的に生成されます。

** データベースIDは、複数のデータベースにアクセスして、特定のデータベースに対してコマンド処理を制限する必要がある場合にのみ必要になります。

-- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

バージョンインジケータ (ACBXVER)

F2

コマンドコード (ACBXCMD)

C3

レスポンスコード (ACBXRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、エラーサブコード (ACBXERRC) フィールドにサブコードを伴う場合があります。詳細は、*Adabas メッセージおよびコードのドキュメント*を参照してください。

データベース ID (ACBXDBID)

このフィールドは、複数のデータベースにアクセスして、特定のデータベースに対するコマンド処理を制限する必要がある場合にのみデータベース ID を指定します。Adabas コールはこのデータベースに送られます。

このフィールドは 4 バイトのバイナリフィールドですが、現時点では 2 バイトのデータベース ID のみがサポートされています。したがって、データベース ID はフィールドの下位部分 (右端のバイト) に指定し、その前にバイナリの 0 を付加する必要があります。

このフィールドがバイナリの 0 に設定されている場合は、Adabas API は DDCARD 入力データで指定された ADARUN カードのデータベース ID か、リンクされているかリンクルーチンによりロードされた LNKGBLS モジュールで指定されているデフォルトのデータベース ID 値のいずれかを使用します。

コマンドオプション 2：ユーザーデータ格納（ACBXCOP2）

オプション	説明
E	ユーザーデータがセッション終了時に Adabas チェックポイントファイルに書き込まれます。空白以外のユニークなユーザー ID を指定してユーザーセッションで OP コマンドを発行している場合に限り、このオプションを使用できます。

エラーサブコード（ACBXERRC）

コマンドがゼロ以外のレスポンスコードを返したときは、このフィールドにレスポンスコードの正確な意味を定義したサブコードが含まれます。レスポンスコードとサブコードについては、『Adabas メッセージおよびコード』を参照してください。

バッファ

次のバッファは C3 コマンドに適用されます。

- フォーマットバッファ
- レコードバッファ

フォーマットバッファ

フォーマットバッファは C3 コマンドでは使用されませんが、Adabas コールに含める必要があります。これが **ACB インターフェイスダイレクトコール** であり、フォーマットバッファが指定されていない場合は、処理エラーが発生します。ACB インターフェイスダイレクトコールはセットシーケンスにバッファが指定されていることを要求します。これが **ACBX インターフェイスダイレクトコール** であり、フォーマットバッファが指定されていない場合は、バッファが自動的に生成されます。

レコードバッファ

レコードバッファには、Adabas チェックポイントファイルに書き込むユーザーデータが含まれています。実際に格納されるバイト数は、レコードバッファ長フィールドで指定した値によって決まります。

ユーザーデータは、空白以外のユニークなユーザー ID が指定されたユーザーセッションの OP コマンドを発行している場合にのみ保存されます。このデータは、ユーザーデータが指定された次の C3 コマンドまたは CL コマンドが発行されるまで保持されます。空白以外のユニークなユーザー ID を指定しないと、後続のセッションでこのデータを読み込むことができません。

27 C5 コマンド：プロテクションログへのユーザーデータの書き込み

- 機能および使用 226
- ACB インターフェイスダイレクトコール：C5 コマンド 226
- ACBX インターフェイスダイレクトコール：C5 コマンド 229
- バッファ 231

C5 コマンドは、SIBA/PLOG にユーザーデータを書き込みます。ただし、Event Replicator for Adabas をインストールした場合は、C5 コマンドを使用して、メッセージを元のアプリケーションから 1 つ以上の Event Replicator サーバーに宛てて送信することもできます。この Event Replicator for Adabas 機能の詳細については、Event Replicator for Adabas のドキュメントを参照してください。

このchapterでは、次のトピックについて説明します。

機能および使用

C5 コマンドは、Adabas データプロテクションログにユーザーデータを書き込むのに使用します。このデータは、ADASEL ユーティリティを使用して読み込んだり、表示したりできます。書き込まれたデータは、Adabas リカバリ処理には影響を及ぼしません。ADASAV ユーティリティおよび ADARES ユーティリティは、C5 コマンドでデータプロテクションログに書き込んだデータをすべて無視します。

ACB インターフェイスダイレクトコール：C5 コマンド

このsectionでは、C5 コマンドの ACB インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの概要](#)
- [コントロールブロックフィールドの説明](#)
- [ACB の例](#)

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	--	--	--
コマンドコード	3~4	英数字	F	U
	5~8	--	--	--
ファイル番号 **	9~10	バイナリ	F	U
レスポンスコード	11~12	バイナリ	--	A
	13~26	--	--	--
レコードバッファ長	27~28	バイナリ	F	U

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	29～34	--	--	--
コマンドオプション 1***	35	英数字	F	U
	36	--	--	--
アディション 1***	37～44	英数字	F	U
	45～72	--	--	--
コマンドタイム	73～76	バイナリ	--	A
ユーザーエリア	77～80	--	--	U

バッファエリア

バッファ	Adabas コール前	Adabas コール後
フォーマット	*	
レコード	F	U

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

* 使用しませんが、コールステートメントのパラメータリストに含める必要があります。

** データベース ID は、複数のデータベースにアクセスして、特定のデータベースに対してコマンド処理を制限する必要がある場合にのみ必要になります。

ただし、Event Replicator for Adabas を使用しておりコマンドオプション 1 を "R" にセットしている場合は、ファイル番号を指定して C5 コマンドを適用するファイルを特定する必要があります。詳細については、Event Replicator for Adabas のドキュメントを参照してください。

*** Event Replicator for Adabas を使用している場合にのみ使用されます。詳細については、Event Replicator for Adabas のドキュメントを参照してください。

-- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

コマンドコード (ACBCMD)

C5

ファイル番号 (ACBFNR)

データベース ID は、複数のデータベースにアクセスして、特定のデータベースに対してコマンド処理を制限する必要がある場合にのみ必要になります。

ただし、Event Replicator for Adabas を使用しておりコマンドオプション 1 を "R" にセットしている場合は、ファイル番号を指定して C5 コマンドを適用するファイルを特定する必要があります。詳細については、Event Replicator for Adabas のドキュメントを参照してください。

レスポンスコード (ACBRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。C5 コマンドによってゼロ以外のレスポンスコードが返された場合、Adabas コントロールブロックのバイト 45~48 (アディクション 2 フィールドの下位 2 バイト) に、レスポンスコードの意味を正確に定義したサブコードが含まれていることがあります。レスポンスコードとサブコードについては、『Adabas メッセージおよびコード』を参照してください。

レコードバッファ長 (ACBRBL)

このフィールドに指定したバイト数が、Adabas データプロテクションログに書き込まれます。

指定可能な最大長は 2048 バイトです。

コマンドオプション 1 (ACBCOP1)

Event Replicator for Adabas を使用している場合にのみ使用されます。それ以外の場合、このフィールドは空白にしなければなりません。詳細については、Event Replicator for Adabas のドキュメントを参照してください。

アディクション 1 (ACBADD1)

Event Replicator for Adabas を使用している場合にのみ使用されます。詳細については、Event Replicator for Adabas のドキュメントを参照してください。

ACB の例

Adabas データプロテクションログに、情報 ULRR0422 UPDATES FOR JANUARY を書き込みます。

コントロールブロック

コマンドコード	C5
レコードバッファ長	28

バッファエリア

レコードバッファ	ULRR0422 UPDATES FOR JANUARY
----------	------------------------------

ACBX インターフェイスダイレクトコール：C5 コマンド

このsectionでは、C5 コマンドの ACBX インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの概要](#)
- [コントロールブロックフィールドの説明](#)

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	---	---	---
バージョンインジケータ	3~4	バイナリ	F	U
	5~6	---	---	---
コマンドコード	7~8	英数字	F	U
	9~10	---	---	---
レスポンスコード	11~12	バイナリ	---	A
	13~16	---	---	---
データベース ID**	17~20	数値	F	U
ファイル番号 ***	21~24	数値	F	U
	25~48	---	---	---
コマンドオプション 1***	49	英数字	F	U
	50~56	---	---	---
アディション 1***	57~64	英数字/バイナリ	F	U
	65~114	---	---	---
エラーサブコード	115~116	バイナリ	---	A
	117~144	---	---	---
コマンドタイム	145~152	バイナリ	---	A
ユーザーエリア	153~168	該当なし	---	U
---	169~193	操作不可	---	---

ABD とバッファ

ABD とバッファ	Adabas コール前	Adabas コール後
フォーマット	*	
レコード	F	U

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

* 使用しませんが、Adabas コールに含める必要があります。含めない場合は、Adabas コールが自動的に生成されます。

** データベース ID は、複数のデータベースにアクセスして、特定のデータベースに対してコマンド処理を制限する必要がある場合にのみ必要になります。

ただし、Event Replicator for Adabas を使用しておりコマンドオプション 1 を "R" にセットしている場合は、ファイル番号を指定して C5 コマンドを適用するファイルを特定する必要があります。詳細については、Event Replicator for Adabas のドキュメントを参照してください。

*** Event Replicator for Adabas を使用している場合にのみ使用されます。詳細については、Event Replicator for Adabas のドキュメントを参照してください。

--- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

バージョンインジケータ (ACBXVER)

F2

コマンドコード (ACBXCMD)

C5

レスポンスコード (ACBXRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、エラーサブコード (ACBXERRC) フィールドにサブコードを伴う場合があります。詳細は、Adabas メッセージおよびコードのドキュメントを参照してください。

データベース ID (ACBXDBID)

このフィールドは、複数のデータベースにアクセスして、特定のデータベースに対するコマンド処理を制限する必要がある場合にのみデータベース ID を指定します。Adabas コールはこのデータベースに送られます。

このフィールドは 4 バイトのバイナリフィールドですが、現時点では 2 バイトのデータベース ID のみがサポートされています。したがって、データベース ID はフィールドの下位部分 (右端のバイト) に指定し、その前にバイナリの 0 を付加する必要があります。

このフィールドがバイナリの0に設定されている場合は、Adabas APIはDDCARD入力データで指定されたADARUNカードのデータベースIDか、リンクされているかリンクルーチンによりロードされたLNKGBLSモジュールで指定されているデフォルトのデータベースID値のいずれかを使用します。

ファイル番号 (ACBXFNR)

Event Replicator for Adabas を使用中であり、かつコマンドオプション2フィールドを "R" に設定した場合、このフィールドはC5コマンドの処理対象となるファイルの番号を指定するのに使用します。詳細については、Event Replicator for Adabas のドキュメントを参照してください。

このフィールドは4バイトのバイナリフィールドですが、ファイル番号はフィールドの下位部分（右端のバイト）に指定し、その前にバイナリの0を付加する必要があります。

コマンドオプション1 (ACBXCOP1)

Event Replicator for Adabas を使用している場合にのみ使用されます。それ以外の場合、このフィールドは空白にしなければなりません。詳細については、Event Replicator for Adabas のドキュメントを参照してください。

アディション1 (ACBXADD1)

Event Replicator for Adabas を使用している場合にのみ使用されます。詳細については、Event Replicator for Adabas のドキュメントを参照してください。

エラーサブコード (ACBXERRC)

コマンドがゼロ以外のレスポンスコードを返したときは、このフィールドにレスポンスコードの正確な意味を定義したサブコードが含まれます。レスポンスコードとサブコードについては、『Adabas メッセージおよびコード』を参照してください。

バッファ

C5 コマンドには、次のバッファが適用されます。

フォーマットバッファ

フォーマットバッファはC5コマンドでは使用されませんが、Adabas コールに含める必要があります。これが **ACB インターフェイスダイレクトコール** であり、フォーマットバッファが指定されていない場合は、処理エラーが発生します。ACBインターフェイスダイレクトコールはセットシーケンスにバッファが指定されていることを要求します。これが **ACBX インターフェイスダイレクトコール** であり、フォーマットバッファが指定されていない場合は、バッファが自動的に生成されます。

レコードバッファ

このバッファには、データプロテクションログに書き込む情報が入ります。

書き込んだ情報を後で ADASEL ユーティリティで選択するには、元々レコードバッファに含まれていた文字列の冒頭部分（1～30文字）を指定します。このため、レコードバッファの冒頭部分にはユーザーデータをユニークに識別できる文字を指定することをお勧めします。これにより、ユーザーデータを正しく識別し、選択できます。

28 CL コマンド：ユーザーセッションのクローズ

■ 機能および使用	234
■ ACB インターフェイスダイレクトコール：CL コマンド	234
■ ACBX インターフェイスダイレクトコール：CL コマンド	238
■ バッファ	241

CL コマンドは、ET セッションを終了して、データベースを更新します。

このchapterでは、次のトピックについて説明します。

機能および使用

CL コマンドは、ユーザーセッションを終了するのに使用します。すべてのユーザープログラムで、データベース処理を完了するとき、CL コマンドを発行することをお勧めします。シングルユーザーモードでデータベース更新を実行するユーザープログラムの場合、更新内容がすべてデータベースに書き込まれるようにするには、処理の最後にCL コマンドを発行する必要があります。

CL コマンドでは、次の処理を行います。

- 内部 ET コマンドを発行します (ET ロジックユーザーのみ)。
- ユーザーデータを Adabas システムファイルに格納します (オプション)。
- ユーザーに関するレコードのうち現在ホールド状態のレコードをすべて解放し、また、ユーザーに割り当てられているコマンド ID エントリ (および対応する ISN リスト) もすべて解放します。
- Adabas ワークから Adabas システムファイルにユーザーの ET データを転送します。この処理は、OP コマンドでユーザー ID を指定した場合にのみ行われます。それ以外の場合、セッション中に格納したどの ET データも保存されません。

シングルユーザーモードのユーザーが CL コマンドを発行すると、データベース (アソシエータ、データストレージ、ワーク、およびデータプロテクションログ) が物理的にクローズされます。このため、シングルユーザーモードのユーザーが CL コマンドの後に他のコマンド (例えば、OP コマンド) を発行することはできません。

ACB インターフェイスダイレクトコール：CL コマンド

このsectionでは、CL コマンドの ACB インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの概要](#)
- [コントロールブロックフィールドの説明](#)

■ ACB の例

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	--	--	--
コマンドコード	3~4	英数字	F	U
コマンド ID	5~8	バイナリ	--	A
ファイル番号 ***	9~10	バイナリ	F	U
レスポンスコード	11~12	バイナリ	--	A
ISN	13~16	バイナリ	--	A
ISN 下限	17~20	バイナリ	--	A
ISN 数	21~24	バイナリ	--	A
	25~26	--	--	--
レコードバッファ長	27~28	バイナリ	F *	U
	29~35	--	--	--
コマンドオプション 2	36	英数字	F	U
	37~72	--	--	--
コマンドタイム	73~76	バイナリ	--	A
ユーザーエリア	77~80	--	--	U

バッファエリア

バッファ	Adabas コール前	Adabas コール後
フォーマット *	**	--
レコード *	F	U

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

* ユーザーデータを格納する場合にのみ必須です。

** 使用されませんが、ユーザーデータを格納する場合にはコールステートメントのパラメータリストに含める必要があります。

*** データベース ID は、複数のデータベースにアクセスして、特定のデータベースに対してコマンド処理を制限する必要がある場合にのみ必要になります。

-- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

コマンドコード (ACBCMD)

CL

コマンド ID (ACBCID)

ET ロジックユーザーの場合、Adabas はユーザーがこのフィールドで最後に正常に実行したトランザクションのトランザクションシーケンス番号を返します。この番号はバイナリ形式で返されます。

CL コマンドには ET コマンドが含まれているため（「[機能および使用](#)」を参照）、トランザクションシーケンス番号も 1 ずつ増加します。

ファイル番号 (ACBFNR)

データベース ID は、複数のデータベースにアクセスして、特定のデータベースに対してコマンド処理を制限する必要がある場合にのみ必要になります。

レスポンスコード (ACBRSP)

Adabas は、コマンドのレスポンスコードをこのフィールドに返します。レスポンスコード 0 は、コマンドが正しく実行されたことを示します。CL コマンドによってゼロ以外のレスポンスコードが返された場合、Adabas コントロールブロックのバイト 47~48（アディション 2 フィールドの下位 2 バイト）に、レスポンスコードの意味を正確に定義したサブコードが含まれていることがあります。レスポンスコードとサブコードについては、『[Adabas メッセージおよびコード](#)』を参照してください。

ISN：I/O の数 (ACBISN)

Adabas は、このフィールドに、このセッションの Adabas コールを実行した結果、発生する入出力処理の回数を返します。

ISN 下限：コマンドの数 (ACBISL)

このフィールドには、Adabas はユーザーセッション中にユーザーが発行したコマンドの数を返します。

ISN 数：CPU 時間 (ACBISQ)

このフィールドには、Adabas は Adabas のコマンド処理でこのユーザーが使用したプロセッサ時間を返します。

時間は 1.048576 秒の単位で返されます。

レコードバッファ長 (ACBRBL)

ユーザーデータを Adabas システムファイルに格納する場合は、レコードバッファ長をこのフィールドに指定する必要があります。指定した長さによって、格納されるユーザーデータのバイト数が決まります。

指定可能な最大長は 2000 バイトです。

ユーザーデータを格納しない場合、このフィールドは使用しません。

コマンドオプション 2：ユーザーデータ格納 (ACBCOP2)

オプション	説明
E	ユーザーデータが Adabas システムファイルに格納されることを示します。

ACB の例

- 例 1
- 例 2

例 1

ユーザープログラムは全データベース処理を完了し、CL コマンドを発行します。格納するユーザーデータはありません。

コントロールブロック

コマンドコード	CL	
コマンドオプション 2	b	格納するユーザーデータはなし

例 2

ユーザープログラムは CL コマンドを発行し、ユーザーデータを Adabas システムファイルに格納します。

コントロールブロック

コマンドコード	CL	
レコードバッファ長	17	17 バイトのユーザーデータを格納
コマンドオプション 2	E	ユーザーデータを格納

バッファ

レコードバッファ	USER 7 NORMAL END
----------	-------------------

ACBX インターフェイスダイレクトコール：CL コマンド

このsectionでは、CL コマンドの ACBX インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの概要](#)
- [コントロールブロックフィールドの説明](#)

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	---	---	---
バージョンインジケータ	3~4	バイナリ	F	U
	5~6	---	---	---
コマンドコード	7~8	英数字	F	U
	9~10	---	---	---
レスポンスコード	11~12	バイナリ	---	A
コマンド ID	13~16	英数字/バイナリ	---	A
データベース ID***	17~20	数値	F	U
	21~28	---	---	---
ISN	29~32	バイナリ	---	A
	33~36	---	---	---
ISN 下限	37~40	バイナリ	---	A
	41~44	---	---	---
ISN 数	45~48	バイナリ	---	A
	49	---	---	---

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
コマンドオプション 2	50	英数字	F	U
	51~114	---	---	---
エラーサブコード	115~116	バイナリ	---	A
	117~144	---	---	---
コマンドタイム	145~152	バイナリ	---	A
ユーザーエリア	153~168	該当なし	---	U
---	169~193	操作不可	---	---

ABD とバッファ

ABD とバッファ	Adabas コール前	Adabas コール後
フォーマット *	**	--
レコード *	F	U

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

* ユーザーデータを格納する場合にのみ必須です。

** 使用しませんが、Adabas コールに含める必要があります。含めない場合は、Adabas コールが自動的に生成されます。

*** データベース ID は、複数のデータベースにアクセスして、特定のデータベースに対してコマンド処理を制限する必要がある場合にのみ必要になります。

-- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

バージョンインジケータ (ACBXVER)

F2

コマンドコード (ACBXCMD)

CL

レスポンスコード (ACBXRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、エラーサブコード (ACBXERRC) フィールドにサブコードを伴う場合があります。詳細は、Adabas メッセージおよびコードのドキュメントを参照してください。

コマンド ID (ACBXCID)

ET ロジックユーザーの場合、Adabas はユーザーがこのフィールドで最後に正常に実行したトランザクションのトランザクションシーケンス番号を返します。この番号はバイナリ形式で返されます。

CL コマンドには ET コマンドが含まれているため（「[機能および使用](#)」を参照）、トランザクションシーケンス番号も 1 ずつ増加します。

データベース ID (ACBXDBID)

このフィールドは、複数のデータベースにアクセスして、特定のデータベースに対するコマンド処理を制限する必要がある場合にのみデータベース ID を指定します。Adabas コールはこのデータベースに送られます。

このフィールドは 4 バイトのバイナリフィールドですが、現時点では 2 バイトのデータベース ID のみがサポートされています。したがって、データベース ID はフィールドの下位部分（右端のバイト）に指定し、その前にバイナリの 0 を付加する必要があります。

このフィールドがバイナリの 0 に設定されている場合は、Adabas API は DDCARD 入力データで指定された ADARUN カードのデータベース ID か、リンクされているカリンクルーチンによりロードされた LNKGBLS モジュールで指定されているデフォルトのデータベース ID 値のいずれかを使用します。

ISN：I/O の数 (ACBXISN)

Adabas は、このフィールドに、このセッションの Adabas コールを実行した結果、発生する入出力処理の回数を返します。

ISN 下限：コマンドの数 (ACBXISL)

このフィールドには、Adabas はユーザーセッション中にユーザーが発行したコマンドの数を返します。

ISN 数：CPU 時間 (ACBXISQ)

このフィールドには、Adabas は Adabas のコマンド処理でこのユーザーが使用したプロセッサ時間を返します。

時間は 1.048576 秒の単位で返されます。

コマンドオプション 2：ユーザーデータ格納 (ACBXCOP2)

オプション	説明
E	ユーザーデータが Adabas システムファイルに格納されることを示します。

エラーサブコード (ACBXERRC)

コマンドがゼロ以外のレスポンスコードを返したときは、このフィールドにレスポンスコードの正確な意味を定義したサブコードが含まれます。レスポンスコードとサブコードについては、『[Adabas メッセージおよびコード](#)』を参照してください。

バッファ

CL コマンドには、次のバッファが適用されます。

- フォーマットバッファ
- レコードバッファ

フォーマットバッファ

フォーマットバッファは CL コマンドでは使用されませんが、Adabas コールに含める必要があります。これが **ACB インターフェイスダイレクトコール** であり、フォーマットバッファが指定されていない場合は、処理エラーが発生します。ACB インターフェイスダイレクトコールはセットシーケンスにバッファが指定されていることを要求します。これが **ACBX インターフェイスダイレクトコール** であり、フォーマットバッファが指定されていない場合は、バッファが自動的に生成されます。

レコードバッファ

コマンドが出力したユーザーデータは、このバッファに格納されます。実際に格納されるバイト数は、レコードバッファ長フィールドで指定した値によって決まります。空白以外のユニークなユーザー ID を指定して OP コマンドを発行している場合に限り、ユーザーデータは保持されます。その場合、このデータはユーザーデータが指定された次の ET または CL コマンドが発行されるまで保持されます。空白以外のユーザー ID を指定しない場合は、後続のセッションでこのデータを読み込むことができません。

29 E1 コマンド：レコードの削除／ファイルのリフ

レッシュ

- 機能および使用 244
- ACB インターフェイスダイレクトコール：E1 コマンド 244
- ACBX インターフェイスダイレクトコール：E1 コマンド 248

E1 コマンドは、ホールドオプション付きでレコードを削除したり、ファイルのリフレッシュしたりします。

このchapterでは、次のトピックについて説明します。

機能および使用

E1 コマンドは、レコードの ISN を指定して 1 レコードを削除したり、ISN の値を 0 にしてファイルのリフレッシュします (0 件状態)。

削除するレコードのファイル番号と ISN を指定する必要があります。Adabas は、データストレージからレコードを削除します。ISN を指定せず、コマンド ID フィールドに空白を設定し、かつ指定したファイルを前回 ADALOD パラメータ PGMREFRESH=YES オプション付きでロードした場合、E1 コマンドはそのファイルの全レコードを削除し、アソシエータとデータストレージを単一エクステントに削減して、そのファイルのリフレッシュします。

レコードの削除またはファイル全体をリフレッシュしたとき、E1 コマンドはアソシエータにも必要な変更を行います。同じ E1 コマンド処理で、レコードの削除とファイルのリフレッシュを同時に行うことはできません。

マルチユーザーモードで実行中で、かつ削除対象のレコードがホールド状態でない場合、Adabas はユーザーに対してそのレコードをホールド状態にします。レコードが別のユーザーに対してホールド状態になっている場合、E1 コマンドはレコードが使用可能になるまで待機状態になります。コマンドオプション 1 フィールドに R オプションを指定した場合に、要求したレコードが使用できないと、レスポンスコード 145 が返されます。



Note: 以前の Adabas リリースでサポートしていた E4 コマンドは、E1 コマンドとして実行されます。

ACB インターフェイスダイレクトコール：E1 コマンド

このsectionでは、E1 コマンドの ACB インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの概要](#)
- [コントロールブロックフィールドの説明](#)

■ ACB の例

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	--	--	--
コマンドコード	3~4	英数字	F	U
コマンド ID	5~8	英数字／バイナリ	F	U
ファイル番号	9~10	バイナリ	F	U
レスポンスコード	11~12	バイナリ	--	A
ISN	13~16	バイナリ	F	U
ISN 下限	17~20	バイナリ	--	A ¹
ISN 数	21~24	バイナリ	--	A ¹
	25~34	--	--	--
コマンドオプション 1	35	英数字	F	U
	36~48	--	--	--
アディション 3	49~56	英数字	F	A
アディション 4	57~64	英数字	F	A
	65~72	--	--	--
コマンドタイム	73~76	バイナリ	--	A
ユーザーエリア	77~80	--	--	U

注

- これらのフィールドは使用されますが、カップリングファイルが使用されている場合は Adabas によりリセットされません。

バッファエリア

何も使用されていません。

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

-- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

コマンドコード (ACBCMD)

E1

コマンド ID (ACBCID)

ファイルをリフレッシュするときは、このフィールドを空白にし、ISN フィールドをゼロにセットします。ISN フィールドをゼロにして、このフィールドに空白以外の値をセットすると、E1 コマンドは、指定ファイルから ISN 0 のレコードを削除しようとするため、レスポンスコード 114 が返されます。

ファイル番号 (ACBFNR)

読み込むファイルの番号を 2 進数で指定します。物理ダイレクトコールの場合は、次のようファイル番号を指定します。

- 1 バイトファイル番号の場合は、ファイル番号を第 2 バイト (10) に入力します。第 1 バイト (9) は、バイナリの 0 (B'0000 0000') をセットします。
- 2 バイトファイル番号の場合は、2 バイト (9 と 10) を使います。



Note: 2 バイトファイル番号およびデータベース ID を使用する場合は、コントロールブロックの先頭バイトに X'30' を入力しなければなりません。

レスポンスコード (ACBRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。E1 コマンドによってゼロ以外のレスポンスコードが返された場合、Adabas コントロールブロックのバイト 47~48 (アディション 2 フィールドの下位 2 バイト) に、レスポンスコードの意味を正確に定義したサブコードが含まれていることがあります。レスポンスコードとサブコードについては、『*Adabas メッセージおよびコード*』を参照してください。

ISN (ACBISN)

削除するレコードの ISN。このフィールドがゼロであり、かつコマンド ID フィールドに空白がセットされていると、ファイル番号フィールドに指定したファイルがリフレッシュされます。このフィールドがゼロでも、コマンド ID フィールドが空白以外の場合、レスポンスコード 114 が返されます。

ISN 数／下限 (ACBISQ および ACBISL)

これらのフィールドは、ハードカップリングファイルが使用されない場合にのみ、E1 コマンド処理の終了後、空値に設定されます。カップリングファイルを使用した場合、これらのフィールドは E1 コマンド処理に使用され、リセットされません。

コマンドオプション 1：レコードが使用不可の場合、レスポンスコード 145 (ACBCOP1)

ユーザーが ET ロジックユーザーであり、削除するレコードがそのユーザーによってホールド状態にされていない場合は、そのレコードはホールド状態になります。削除するレコードが、他のユーザーにホールドされている場合、コマンドオプション 1 フィールドの設定によって次の処理が行われます。

オプション	説明
R (リターン)	削除するレコードが使用不可の場合に、レスポンスコード 145 が返されます。コマンドは待機状態にはなりません。

それ以外の場合、Adabas はレコードが使用できるようになるかトランザクションがタイムアウトになるまで、E1 コマンドを待機状態にします。

アディション 3：パスワード (ACBADD3)

このフィールドは、Adabas Security または ADAESI パスワードを指定するため使用します。データベース、ファイル、またはフィールドがセキュリティ保護されている場合は、有効なセキュリティパスワードまたは ADAESI パスワードをユーザーは指定する必要があります。パスワードを保護するために、コマンド処理時にアディション 3 フィールドに空白がセットされます。

アディション 4：サイファコード (ACBADD4)

このフィールドは、サイファコードを指定するために使用します。ファイルを暗号化する場合は、有効なサイファコードを指定する必要があります。ファイルを暗号化しない場合は、このフィールドに空白をセットする必要があります。

Adabas は、コマンド処理時に、サイファコードに空白をセットし、バージョンコードとデータベース ID をフィールドの下位 3 バイトに返します。詳細については「[コントロールブロックフィールド](#)」を参照してください。

ACB の例

- 例 1

■ 例 2

例 1

ファイル 2 の ISN 4 を削除します。

コントロールブロック

コマンドコード	E1	
ファイル番号	2	削除するレコードはファイル 2 にあります。
ISN	4	ISN4 のレコードを削除します。
アディション 3	パスワード	ファイル 2 はセキュリティ保護されています。

例 2

ファイル番号フィールドに指定したファイル（ファイル番号 4）をリフレッシュします。

コントロールブロック

コマンドコード	E1	
ファイル番号	4	ファイル 4 をリフレッシュします。
ISN		バイナリの 0 に設定します。
コマンド ID	bbbb	空白に設定します。

E1 コマンドはファイル（ファイル番号 4）をリフレッシュします。

ACBX インターフェイスダイレクトコール：E1 コマンド

このsectionでは、E1 コマンドの ACBX インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

■ コントロールブロックとバッファの概要

■ コントロールブロックフィールドの説明

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1～2	---	---	---
バージョンインジケータ	3～4	バイナリ	F	U
	5～6	---	---	---
コマンドコード	7～8	英数字	F	U
	9～10	---	---	---
レスポンスコード	11～12	バイナリ	---	A
コマンド ID	13～16	英数字／バイナリ	F	U
データベース ID	17～20	数値	F	U
ファイル番号	21～24	数値	F	U
	25～28		---	---
ISN	29～32	バイナリ	F	U
	33～36		---	---
ISN 下限	37～40	バイナリ	---	A ¹
	41～44		---	---
ISN 数	45～48	バイナリ	---	A ¹
コマンドオプション 1	49	英数字	F	U
	50～68	---	---	---
アディション 3	69～76	英数字／バイナリ	F	A
アディション 4	77～84	英数字	F	A
	85～114	---	---	---
エラーサブコード	115～116	バイナリ	---	A
	117～144	---	---	---
コマンドタイム	145～152	バイナリ	---	A
ユーザーエリア	153～168	該当なし	---	U
---	169～193	操作不可	---	---

注

- これらのフィールドは使用されますが、カップリングファイルが使用されている場合は Adabas によりリセットされません。

ABD とバッファ

何も使用されていません。

上記の意味は次に示すとおりです。

- F Adabas コール前にユーザーが入力するフィールド
- A Adabas により入力されるフィールド
- U Adabas コール後も変化なし
- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

バージョンインジケータ (ACBXVER)

F2

コマンドコード (ACBXCMD)

E1

レスポンスコード (ACBXRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、エラーサブコード (ACBXERRC) フィールドにサブコードを伴う場合があります。詳細は、*Adabas* メッセージおよびコードのドキュメントを参照してください。

コマンド ID (ACBXCID)

ファイルをリフレッシュするときは、このフィールドを空白にし、ISN フィールドをゼロにセットします。ISN フィールドをゼロにして、このフィールドに空白以外の値をセットすると、E1 コマンドは、指定ファイルから ISN 0 のレコードを削除しようとするため、レスポンスコード 114 が返されます。

データベース ID (ACBXDBID)

このフィールドを使用して、データベース ID を指定します。Adabas コールはこのデータベースに送られます。

このフィールドは 4 バイトのバイナリフィールドですが、現時点では 2 バイトのデータベース ID のみがサポートされています。したがって、データベース ID はフィールドの下位部分 (右端のバイト) に指定し、その前にバイナリの 0 を付加する必要があります。

このフィールドがバイナリの 0 に設定されている場合は、Adabas API は DDCARD 入力データで指定された ADARUN カードのデータベース ID か、リンクされているかリンクルーチンによりロードされた LNKGBLS モジュールで指定されているデフォルトのデータベース ID 値のいずれかを使用します。

ファイル番号 (ACBXFNR)

このフィールドを使用して、Adabas コールで送る必要のあるファイル数を指定します。

このフィールドは4バイトのバイナリフィールドですが、ファイル番号はフィールドの下位部分（右端のバイト）に指定し、その前にバイナリの0を付加する必要があります。

ISN (ACBXISL)

削除するレコードのISN。このフィールドがゼロであり、かつコマンド ID フィールドに空白がセットされていると、ファイル番号フィールドに指定したファイルがリフレッシュされます。このフィールドがゼロでも、コマンド ID フィールドが空白以外の場合、レスポンスコード 114 が返されます。

ACBXISL フィールドは、8 バイトの ACBXISLG フィールドに埋め込まれた4バイトのバイナリフィールドで、まだ使用されていません。ACBXISLG フィールドの上位部分には、バイナリの0を設定します。

このフィールドは、ハードカップリングファイルが使用されない限り、E1 コマンド処理終了後、空値にセットされます。カップリングファイルが使用されている場合は、このフィールドは E1 コマンドの処理に使用され、リセットされません。

ISN 数 (ACBXISQ)

このフィールドは、ハードカップリングファイルが使用されない限り、E1 コマンド処理終了後、空値にセットされます。カップリングファイルが使用されている場合は、このフィールドは E1 コマンドの処理に使用され、リセットされません。

コマンドオプション 1：レコードが使用不可の場合、レスポンスコード 145 (ACBXCOP1)

ユーザーが ET ロジックユーザーであり、削除するレコードがそのユーザーによってホールド状態にされていない場合は、そのレコードはホールド状態になります。削除するレコードが、他のユーザーにホールドされている場合、コマンドオプション 1 フィールドの設定によって次の処理が行われます。

オプション	説明
R (リターン)	削除するレコードが使用不可の場合に、レスポンスコード 145 が返されます。コマンドは待機状態にはなりません。

それ以外の場合、Adabas はレコードが使用できるようになるかトランザクションがタイムアウトになるまで、E1 コマンドを待機状態にします。

アディクション 3：パスワード (ACBXADD3)

このフィールドは、Adabas Security または ADAESI パスワードを指定するため使用します。データベース、ファイル、またはフィールドがセキュリティ保護されている場合は、有効なセキュリティパスワードまたは ADAESI パスワードをユーザーは指定する必要があります。パスワードを保護するために、コマンド処理時にアディクション 3 フィールドに空白がセットされます。

アディション4：サイファコード (ACBXADD4)

このフィールドは、サイファコードを指定するために使用します。ファイルを暗号化する場合は、有効なサイファコードを指定する必要があります。ファイルを暗号化しない場合は、このフィールドに空白をセットする必要があります。

Adabasは、コマンド処理時に、サイファコードに空白をセットし、バージョンコードとデータベースIDをフィールドの下位3バイトに返します。詳細については「[コントロールブロックフィールド](#)」を参照してください。

エラーサブコード (ACBXERRC)

コマンドがゼロ以外のレスポンスコードを返したときは、このフィールドにレスポンスコードの正確な意味を定義したサブコードが含まれます。レスポンスコードとサブコードについては、『*Adabas メッセージおよびコード*』を参照してください。

30 ET コマンド：トランザクションの終了

■ 機能および使用	254
■ ACB インターフェイスダイレクトコール：ET コマンド	255
■ ACBX インターフェイスダイレクトコール：ET コマンド	259
■ バッファ	262

ET コマンドは、現在のトランザクションを終了して保存します。

このchapterでは、次のトピックについて説明します。

機能および使用

ET コマンドは、論理トランザクションの終了を示すのに使用します。各論理トランザクションで、ET コマンドを発行しなければなりません。後続のトランザクションでリストアまたはバックアウトすると、最後の ET コマンドで設定したデータベースステータスに戻ります。

ET コマンドでは、次の処理を行います。

- トランザクションで正しく実行された全更新コマンドに関して、現在のデータプロテクション情報をすべて Adabas データプロテクションログと Adabas ワークに書き込みます。現在のセッションが異常終了した場合、Adabas はこのプロテクション情報を使用してアソシエータおよびデータストレージに対して、このトランザクションの更新を次セッションで再度適用します。
- 現在のトランザクション内でユーザーがホールドした全レコードを解放します。FIND コマンドの結果としてユーザーが保存した ISN リストは解放されません（このような ISN をホールド状態に戻すオプションもあります）。
- 任意で、Adabas システムファイルにユーザーデータを格納できます。このユーザーデータは、プログラム再起動手順の一環として、後で OP コマンドまたは RE コマンドで読み込むことができます。
- このコマンドおよび同じユーザーが後で発行した OP コマンドまたは CL コマンドのコマンド ID フィールドに、トランザクションのユニークなシーケンス番号を返します。このシーケンス番号は、このユーザーに関して最後に正しく処理が完了したトランザクションを識別するのに使用できます。

ET コマンドを正しく実行すると、後続のユーザーまたは Adabas セッションが中断しても、トランザクションで実行した更新処理がすべてデータベースに適用されます。

ET ロジックユーザーが、システムエラーの発生後またはトランザクションの最中に OP コマンドを発行すると、Adabas は BT コマンドを自動的に発行します。

ACB インターフェイスダイレクトコール：ET コマンド

このsectionでは、ET コマンドのACBインターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの概要](#)
- [コントロールブロックフィールドの説明](#)
- [ACB の例](#)

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	--	--	--
コマンドコード	3~4	英数字	F	U
コマンド ID	5~8	バイナリ	--	A
ファイル番号****	9~10	バイナリ	F	U
レスポンスコード	11~12	バイナリ	--	A
	13~16	--	--	--
ISN 下限	17~20	バイナリ	F	U
ISN 数	21~24	バイナリ	--	A
	25~26	--	--	
レコードバッファ長	27~28	バイナリ	F*	U
	29~32	--	--	--
ISN バッファ長**	33~34	バイナリ	F**	U
コマンドオプション 1	35	英数字	F	U
コマンドオプション 2	36	英数字	F	U
	37~72	--	--	--
コマンドタイム	73~76	バイナリ	--	A
ユーザーエリア	77~80	--	--	U

バッファエリア

バッファ	Adabas コール前	Adabas コール後
フォーマット *	***	--
レコード *	F	U
ISN **	F	U

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

* ET データを格納する場合にのみ必須です。

** ISN ホールドオプションには必須、マルチフェッチオプションには任意

*** 使用されませんが、ET データを格納する場合にはコールステートメントのパラメータリストに含める必要があります。

**** データベース ID は、複数のデータベースにアクセスして、特定のデータベースに対してコマンド処理を制限する必要がある場合にのみ必要になります。

-- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

コマンドコード (ACBCMD)

ET

コマンド ID (ACBCID)

Adabas はこのフィールドにバイナリの 0 またはトランザクションシーケンス番号を返します。

更新コマンド (A1、E1、N1、N2) を発行しないで、トランザクションが ET コマンドによって終了すると、バイナリの 0 が返されます。

それ以外の場合、トランザクションシーケンス番号はバイナリ形式で返されます。この番号は、ユーザーセッションごとに 1 から順番に昇順に割り当てられます。

ファイル番号 (ACBFNR)

データベース ID は、複数のデータベースにアクセスして、特定のデータベースに対してコマンド処理を制限する必要がある場合にのみ必要になります。

レスポンスコード (ACBRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ET コマンドによってゼロ以外のレスポンスコードが返された場合、Adabas コントロールブロックのバイト 45~48 (アディクション 2 フィールドの下位 2 バイト) に、レスポンスコードの意味を正確に定義したサブ

コードが含まれていることがあります。レスポンスコードとサブコードについては、『Adabas メッセージおよびコード』を参照してください。

ISN 下限 (ACBISL)

ISN ホールドオプションを指定した場合、6 バイトの ISN バッファエントリのカウントをこのフィールドに入れる必要があります。

ISN 数 (ACBISQ)

このフィールドに Adabas は、タイマの単位として指定された値に応じた、トランザクション経過時間を返します（基本単位は 1.05 秒）。

レコードバッファ長 (ACBRBL)

ユーザーデータを Adabas システムファイルに格納する場合は、格納するデータのバイト数をこのフィールドに指定する必要があります。


Adabas は、このフィールドに指定されたバイト数だけ格納します。指定可能な最大バイト数は、2000 バイトです。

ユーザーデータを格納しない場合、このフィールドは使用しません。

ISN バッファ長 (ACBIBL)

ISN バッファ長（バイト単位）。この長さは、ISN ホールドオプションまたはマルチフェッチオプションを使用するときのみ必要です（コマンドオプション 1 フィールドの説明を参照）。

コマンドオプション 1：ISN ホールドオプション (ACBCOP1)

 **Note:** ADARUN PREFETCH=YES でマルチフェッチを設定すると、ET/BT コマンドに対して P オプションが自動的に使用されます（他のすべてのコマンドに対しては M オプションが自動的に使用されます）。このフィールドを使用して、このオプション設定を変更することはできません。

デフォルトでは、ET コマンドを実行すると、Adabas はユーザーが現在ホールドしている全 ISN を解放します。

オプション	説明
P	これらの ISN の全部または一部をホールド状態に戻します。ホールド状態に戻る ISN は ISN バッファ に指定する必要があります。また、ISN カウントは ISN 下限フィールドに指定する必要があります。
M (コマンドレベルのマルチフェッチ)	現在のトランザクションによってホールドされているレコードのサブセットのみを解放します。ホールド状態から解放されたレコードを ISN バッファ に指定します。ISN バッファ内の最初のフルワードは 8 バイトの要素の数を指定し、その後続く各 8 バイトグループは、ホールド状態から解放される 1 つのファイル番号/ISN ID として解釈されます。

コマンドオプション 2：ユーザーデータ格納 (ACBCOP2)

オプション	説明
E	ユーザーデータが Adabas システムファイルに格納されることを示します。

ACB の例

- 例 1：ユーザーデータなしの ET
- 例 2：ユーザーデータ付きの ET
- 例 3：ISN ホールドオプション付きの ET

例 1：ユーザーデータなしの ET

コントロールブロック

コマンドコード	ET	
コマンドオプション 2	b	格納するユーザーデータはなし

例 2：ユーザーデータ付きの ET

コントロールブロック

コマンドコード	ET	
レコードバッファ長	25	25 バイトのユーザーデータを格納します。
コマンドオプション 2	E	ユーザーデータを格納します。

バッファエリア

レコードバッファ	USER DATA FOR TRANSACTION
----------	---------------------------

例 3：ISN ホールドオプション付きの ET

コントロールブロック

コマンドコード	ET	
ISN 下限	3	
ISN バッファ長	18	
コマンドオプション 1	P	ISN をホールド状態にします。

バッファエリア

ISN バッファ	
000600000001	ISN 1
000600000002	ISN 2
000600000003	ISN 3

ACBX インターフェイスダイレクトコール：ET コマンド

このsectionでは、ET コマンドの ACBX インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの概要](#)
- [コントロールブロックフィールドの説明](#)

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	---	---	---
バージョンインジケータ	3~4	バイナリ	F	U
	5~6	---	---	---
コマンドコード	7~8	英数字	F	U
	9~10	---	---	---
レスポンスコード	11~12	バイナリ	---	A
コマンド ID	13~16	英数字/バイナリ	---	A
データベース ID****	17~20	数値	F	U
	21~36	---	---	---
ISN 下限	37~40	バイナリ	F	U
	41~44	---	---	---
ISN 数	45~48	バイナリ	---	A
コマンドオプション 1	49	英数字	F	U
コマンドオプション 2	50	英数字	F	U
	51~114	---	---	---
エラーサブコード	115~116	バイナリ	---	A
	117~144	---	---	---

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
コマンドタイム	145~152	バイナリ	---	A
ユーザーエリア	153~168	該当なし	---	U
---	169~193	操作不可	---	---

ABD とバッファ

ABD とバッファ	Adabas コール前	Adabas コール後
フォーマット *	***	--
レコード *	F	U
ISN **	F	U

上記の意味は次に示すとおりです。

- F Adabas コール前にユーザーが入力するフィールド
- A Adabas により入力されるフィールド
- U Adabas コール後も変化なし
- * ET データを格納する場合にのみ必須です。
- ** ISN ホールドオプションには必須、マルチフェッチオプションには任意
- *** 使用しませんが、ET データを格納するには Adabas コールに含める必要があります。指定しない場合は、自動的に生成されます。
- **** データベース ID は、複数のデータベースにアクセスして、特定のデータベースに対してコマンド処理を制限する必要がある場合にのみ必要になります。
- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

バージョンインジケータ (ACBXVER)

F2

コマンドコード (ACBXCMD)

ET

レスポンスコード (ACBXRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、エラーサブコード (ACBXERRC) フィールドにサブコードを伴う場合があります。詳細は、*Adabas* メッセージおよびコードのドキュメントを参照してください。

コマンド ID (ACBXCID)

Adabas はこのフィールドにバイナリの 0 またはトランザクションシーケンス番号を返します。

更新コマンド (A1、E1、N1、N2) を発行しないで、トランザクションが ET コマンドによって終了すると、バイナリの 0 が返されます。

それ以外の場合、トランザクションシーケンス番号はバイナリ形式で返されます。この番号は、ユーザーセッションごとに 1 から順番に昇順に割り当てられます。

データベース ID (ACBXDBID)

このフィールドは、複数のデータベースにアクセスして、特定のデータベースに対するコマンド処理を制限する必要がある場合にのみデータベース ID を指定します。Adabas コールはこのデータベースに送られます。

このフィールドは 4 バイトのバイナリフィールドですが、現時点では 2 バイトのデータベース ID のみがサポートされています。したがって、データベース ID はフィールドの下位部分 (右端のバイト) に指定し、その前にバイナリの 0 を付加する必要があります。

このフィールドがバイナリの 0 に設定されている場合は、Adabas API は DDCARD 入力データで指定された ADARUN カードのデータベース ID か、リンクされているカリンクルーチンによりロードされた LNKGBLS モジュールで指定されているデフォルトのデータベース ID 値のいずれかを使用します。

ISN 下限 (ACBXISL)

ISN ホールドオプションを指定した場合、6 バイトの ISN バッファエントリのカウントをこのフィールドに入れる必要があります。

ACBXISL フィールドは、8 バイトの ACBXISLG フィールドに埋め込まれた 4 バイトのバイナリフィールドで、まだ使用されていません。ACBXISLG フィールドの上位部分には、バイナリの 0 を設定します。

ISN 数 (ACBXISQ)

このフィールドに Adabas は、タイマの単位として指定された値に応じた、トランザクション経過時間を返します (基本単位は 1.05 秒)。

コマンドオプション 1：ISN ホールドオプション (ACBXCOP1)

デフォルトでは、ET コマンドを実行すると、Adabas はユーザーが現在ホールドしている全 ISN を解放します。

オプション	説明
P	これらの ISN の全部または一部をホールド状態に戻します。ホールド状態に戻る ISN は ISN バッファ に指定する必要があります。また、ISN カウントは ISN 下限フィールドに指定する必要があります。
M (コマンドレベルのマルチフェッチ)	現在のトランザクションによってホールドされているレコードのサブセットのみを解放します。ホールド状態から解放されたレコードを ISN バッファ に指定します。ISN バッファ内の最初のフルワードは 8 バイトの要素の数を指

オプション	説明
	定し、その後続く各 8 バイトグループは、ホールド状態から解放される 1 つのファイル番号/ISN ID として解釈されます。

コマンドオプション 2：ユーザーデータ格納（ACBXCOP2）

オプション	説明
E	ユーザーデータが Adabas システムファイルに格納されることを示します。

エラーサブコード（ACBXERRC）

コマンドがゼロ以外のレスポンスコードを返したときは、このフィールドにレスポンスコードの正確な意味を定義したサブコードが含まれます。レスポンスコードとサブコードについては、『*Adabas メッセージおよびコード*』を参照してください。

バッファ

ET コマンドには、次のバッファが適用されます。

- フォーマットバッファ
- レコードバッファ
- ISN バッファ

フォーマットバッファ

フォーマットバッファは ET コマンドでは使用されませんが、ET データを格納する場合には Adabas コールに含める必要があります。これが **ACB インターフェイスダイレクトコール** であり、フォーマットバッファが指定されていない場合は、処理エラーが発生します。ACB インターフェイスダイレクトコールはセットシーケンスにバッファが指定されていることを要求します。これが **ACBX インターフェイスダイレクトコール** であり、フォーマットバッファが指定されていない場合は、バッファが自動的に生成されます。

レコードバッファ

ET データを格納する場合、このバッファには Adabas システムファイルに格納されるユーザーデータが入ります。

このデータは、ET データ付きの次の ET コマンドまたは CL コマンドを発行するまで保持されます。ユーザーデータは、空白以外のユニークなユーザー ID を指定した OP コマンドを発行した場合にのみ、ユーザーセッション終了時に保存されます。

ISN バッファ

コマンドオプション1フィールドが"P"の場合は、ホールド状態にする各ISNに6バイトのバイナリエントリを指定する必要があります。先頭2バイトにはレコードが存在するファイル番号を指定し、残り4バイトにはホールド状態にするレコードのISNを指定します。ファイル番号およびISNの整合チェックは行われません。

コマンドオプション1フィールドに"M"をセットした場合、現在のトランザクションによってホールドされているレコードのサブセットだけが解放されます。**ISN バッファ**内の最初のフルワードは、8バイトのエLEMENTの数になります。後続の各8バイトグループは、ホールド状態から解放されるレコードのファイル番号/ISN IDであると解釈されます。詳細については、「[BT/ET のマルチフェッチ処理](#)」を参照してください。

31 HI コマンド：レコードのホールド

- 機能および使用 266
- ACB インターフェイスダイレクトコール：H1 コマンド 266
- ACBX インターフェイスダイレクトコール：H1 コマンド 269

HI コマンドは、他のユーザーによるレコード更新を防ぎます。

このchapterでは、次のトピックについて説明します。

機能および使用

HI コマンドは、レコードをホールド状態にするのに使用します。このコマンドを使用すると、後続の更新処理のためにレコードをホールドし、レコードが解放されるまで他のユーザーがそのレコードを更新できないようにすることができます。

ホールドするレコードのファイル番号と ISN を指定します。

ホールドするレコードが、現在他のユーザーによってホールドされている場合は、Adabas コントロールブロックのコマンドオプション1フィールドの設定によって、実行される処理が制御されます。コマンドオプション1フィールドの設定に応じて、次の処理が行われます。

- "R" の場合、ホールド対象のレコードが使用できないときは、レスポンスコード 145 が返されます。
- "R" 以外の場合、レコードが使用可能になるまでユーザーは待機状態となり、使用可能になるとユーザーは自動的に再度アクティブになります。

ACB インターフェイスダイレクトコール：H1 コマンド

このsectionでは、H1 コマンドの ACB インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの概要](#)
- [コントロールブロックフィールドの説明](#)
- [ACB の例](#)

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	--	--	--
コマンドコード	3~4	英数字	F	U
	5~8	--	--	--
ファイル番号	9~10	バイナリ	F	U
レスポンスコード	11~12	バイナリ	--	A
ISN	13~16	バイナリ	F	U
	17~34	--	--	--
コマンドオプション 1	35	英数字	F	U
	36~72	--	--	--
コマンドタイム	73~76	バイナリ	--	A
ユーザーエリア	77~80	--	--	U

バッファエリア

何も使用されていません。

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

-- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

コマンドコード (ACBCMD)

HI

ファイル番号 (ACBFNR)

ホールドするレコードがあるファイルの番号。



Note: 2 バイトファイル番号およびデータベース ID を使用する場合は、コントロールブロックの先頭バイトに 'X'30' を入力しなければなりません。

レスポンスコード (ACBRSP)

Adabas は、コマンドのレスポンスコードをこのフィールドに返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。HI コマンドによってゼロ以外のレスポンスコードが返された場合、Adabas コントロールブロックのバイト 45~48 (アディクション 2 フィールドの下位 2 バイト) に、レスポンスコードの意味を正確に定義したサブコードが含まれていることがあります。レスポンスコードとサブコードについては、『Adabas メッセージおよびコード』を参照してください。

ISN (ACBISN)

ホールド状態にするレコードの ISN。

コマンドオプション 1：レコードが使用不可の場合、レスポンスコード 145 (ACBCOP1)

ホールド対象のレコードが、現在他のユーザーにホールドされている場合、コマンドオプション 1 フィールドの指定によって次の処理が行われます。

オプション	説明
R (リターン)	ホールド対象のレコードが使用不可の場合にレスポンスコード 145 が返されます。コマンドは待機状態にはなりません。

それ以外の場合は、レコードが使用可能になり、コマンドとユーザーが自動的にアクティブになるまで、またはトランザクションがタイムアウトになるまで、コマンドは待機状態になります。

ACB の例

ファイル番号 2 の ISN 番号 3 のレコードをホールド状態にします。レコードが使用可能になるまで待機状態となります。

コントロールブロック

コマンドコード	HI	
ファイル番号	2	ホールドするレコードはファイル番号 2 にあります。
ISN	3	ISN 番号 3 のレコードをホールドします。
コマンドオプション 1	b	レスポンスコード 145 オプション使用しません

ACBX インターフェイスダイレクトコール：H1 コマンド

このsectionでは、H1 コマンドの ACBX インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの概要](#)
- [コントロールブロックフィールドの説明](#)

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	---	---	---
バージョンインジケータ	3~4	バイナリ	F	U
	5~6	---	---	---
コマンドコード	7~8	英数字	F	U
	9~10	---	---	---
レスポンスコード	11~12	バイナリ	---	A
---	13~16	---	---	---
データベース ID	17~20	数値	F	U
ファイル番号	21~24	数値	F	U
	25~28	---	---	---
ISN	29~32	バイナリ	F	U
	33~48	---	---	---
コマンドオプション 1	49	英数字	F	U
	50~114	---	---	---
エラーサブコード	115~116	バイナリ	---	A
	117~144	---	---	---
コマンドタイム	145~152	バイナリ	---	A
ユーザーエリア	153~168	該当なし	---	U
---	169~193	操作不可	---	---

バッファエリア

何も使用されていません。

上記の意味は次に示すとおりです。

- F Adabas コール前にユーザーが入力するフィールド
- A Adabas により入力されるフィールド
- U Adabas コール後も変化なし
- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

バージョンインジケータ (ACBXVER)

F2

コマンドコード (ACBXCMD)

HI

レスポンスコード (ACBXRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、エラーサブコード (ACBXERRC) フィールドにサブコードを伴う場合があります。詳細は、*Adabas* メッセージおよびコードのドキュメントを参照してください。

データベース ID (ACBXDBID)

このフィールドを使用して、データベース ID を指定します。Adabas コールはこのデータベースに送られます。

このフィールドは 4 バイトのバイナリフィールドですが、現時点では 2 バイトのデータベース ID のみがサポートされています。したがって、データベース ID はフィールドの下位部分 (右端のバイト) に指定し、その前にバイナリの 0 を付加する必要があります。

このフィールドがバイナリの 0 に設定されている場合は、Adabas API は DDCARD 入力データで指定された ADARUN カードのデータベース ID か、リンクされているカリンクルーチンによりロードされた LNKGBLS モジュールで指定されているデフォルトのデータベース ID 値のいずれかを使用します。

ファイル番号 (ACBXFNR)

このフィールドを使用して、Adabas コールで送る必要のあるファイル数を指定します。

このフィールドは 4 バイトのバイナリフィールドですが、ファイル番号はフィールドの下位部分 (右端のバイト) に指定し、その前にバイナリの 0 を付加する必要があります。

ISN (ACBXISN)

このフィールドは、ホールド状態にするレコードの ISN を指定するのに使用します。

ACBXISN フィールドは、8 バイトの ACBXISNG フィールドに埋め込まれた 4 バイトのバイナリフィールドで、まだ使用されていません。ACBXISNG フィールドの上位部分には、バイナリの 0 を設定します。

コマンドオプション 1：レコードが使用不可の場合、レスポンスコード 145 (ACBXCOP1)

ホールド対象のレコードが、現在他のユーザーにホールドされている場合、コマンドオプション 1 フィールドの指定によって次の処理が行われます。

オプション	説明
R (リターン)	ホールド対象のレコードが使用不可の場合にレスポンスコード 145 が返されます。コマンドは待機状態にはなりません。

それ以外の場合は、レコードが使用可能になり、コマンドとユーザーが自動的にアクティブになるまで、またはトランザクションがタイムアウトになるまで、コマンドは待機状態になります。

エラーサブコード (ACBXERRC)

コマンドがゼロ以外のレスポンスコードを返したときは、このフィールドにレスポンスコードの正確な意味を定義したサブコードが含まれます。レスポンスコードとサブコードについては、『Adabas メッセージおよびコード』を参照してください。

32 L1/L4コマンド：読み込み／読み込みおよびレコードのホールド

- 機能および使用 274
- ACB インターフェイスダイレクトコール：L1 コマンドおよび L4 コマンド 275
- ACBX インターフェイスダイレクトコール：L1 コマンドおよび L4 コマンド 285
- バッファ 290

L1 コマンドと L4 コマンドでは、データストレージからレコードを1つ読み込みます。

このchapterでは、次のトピックについて説明します。

機能および使用

読み込むレコードのファイル番号と ISN を指定します。また、**フォーマットバッファ**でどのフィールドを読み込むかを示します。**レコードバッファ**には、要求されたフィールド値が返されます。

L4 コマンドは L1 コマンドと同じ機能を持ちます。ただし、そのレコードをホールド状態にします。ホールドするレコードが現在他のユーザーによってホールド状態になっている場合は、レコードが使用可能になるか、またはトランザクションがタイムアウトになるまで待機状態になります。コマンドオプション1フィールドを "R" にして L4 コマンドを発行した場合、読み込んでホールドするレコードが使用できないときは、レスポンスコード 145 が返されます。

FIRST UNUSED ISN (F) オプションをコマンドオプション2フィールドに指定すると、その ISN フィールドの指定されたファイルについて、次に高い未使用 ISN が返されます。F オプションは、拡張ファイルには使用できません。

マルチフェッチ／プリフェッチオプションを使用すると単一レコードのフェッチに必要な時間が省かれ、パフォーマンスを向上できます。マルチフェッチ／プリフェッチを使用可能にするには、コマンドオプション1フィールドに "M"、"O" (マルチフェッチ)、または "P" (プリフェッチ) を指定します。詳細は、「[マルチフェッチ／プリフェッチ機能の使用](#)」を参照してください。

GET NEXT (N) オプションをコマンドオプション2フィールドに指定すると、ISN リスト (Sx コマンドですでに作成済み) 内の ISN で識別されるレコードが読み込まれます。このとき、各 L1/L4 コールで読み込むレコードの ISN を指定する必要はありません。Adabas によってリストから ISN が選択され、その ISN で識別されるレコードが読み込まれます。

READ BY ISN SEQUENCE (I) オプションをコマンドオプション2フィールドに指定すると、ISN フィールドに指定した ISN で識別されるレコードが読み込まれます。指定した ISN がファイルに存在しない場合は、次に大きな ISN のレコードが読み込まれ、その ISN が ISN フィールドに返されます。

COMPRESSED オプション (フォーマットバッファに "C." を指定することで設定) を使用すると、読み込んだレコードを Adabas 内部に格納するときと同じ圧縮形式で返すように要求できます。

ACB インターフェイスダイレクトコール：L1 コマンドおよびL4 コマンド

このsectionでは、L1 コマンドおよびL4 コマンドの ACB インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの概要](#)
- [コントロールブロックフィールドの説明](#)
- [ACB の例](#)

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	--	--	--
コマンドコード	3~4	英数字	F	U
コマンド ID	5~8	英数字	F	U
ファイル番号	9~10	バイナリ	F	U
レスポンスコード	11~12	バイナリ	--	A
ISN	13~16	バイナリ	F	U*
ISN 下限	17~20	バイナリ	F	U
	21~24	--	--	--
フォーマットバッファ長	25~26	バイナリ	F	U
レコードバッファ長	27~28	バイナリ	F	U
	29~32	--	--	--
ISN バッファ長**	33~34	バイナリ	F	U
コマンドオプション 1	35	英数字	F	U
コマンドオプション 2	36	英数字	F	U
	37~44	--	--	--
アディション 2	45~48	バイナリ／バイナリ	--	A
アディション 3	49~56	英数字	F	A
アディション 4	57~64	英数字	F	A
アディション 5	65~72	英数字	F	U
コマンドタイム	73~76	バイナリ	--	A

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
ユーザーエリア	77~80	--	--	U

バッファエリア

バッファ	Adabas コール前	Adabas コール後
フォーマット	F	U
レコード	--	A
ISN **	--	A

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

* 特殊なオプションの除外

** ISN バッファおよび長さは、マルチフェッチオプションまたはプリフェッチオプションを指定した場合にのみ必須です。

-- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

コマンドコード (ACBCMD)

L1 または L4

コマンド ID (ACBCID)

一連の L1 または L4 コールを使用して複数のレコードを読み、各コールでフォーマットバッファに同一フィールドを指定する場合、このフィールドは空白およびゼロを除いた値にします。これにより、L1 または L4 の各コールに要する処理時間を短縮できます。

GET NEXT オプションを使用する場合は、使用する ISN リストのコマンド ID をこのフィールドに指定する必要があります。また、GET NEXT オプション使用時、L1 または L4 の各コール間でフォーマットバッファを変更しないでください。

1 レコードだけを読み込む場合やフォーマットバッファを L1 または L4 コール間で変更する場合は、このフィールドを空白にしておきます。


コマンド ID 値が X'FFFFFFFF' の場合、自動的にコマンド ID が生成されます。この場合、Adabas ニュークリアスは X'00000001' から始まるコマンド ID 値を生成し、L1 または L4 コールごとに値を 1 ずつ増加します。ユーザー定義コマンド ID を指定する場合、ユーザーは各コマンド ID を固有のものにする必要があります。

別のフォーマット ID およびグローバルフォーマット ID の使用については、アディション 5 フィールドを参照してください。

ファイル番号 (ACBFNR)

読み込むファイルの番号を2進数で指定します。物理ダイレクトコールの場合は、次のようファイル番号を指定します。

- 1 バイトファイル番号の場合は、ファイル番号を第 2 バイト (10) に入力します。第 1 バイト (9) は、バイナリの 0 (B'0000 0000') をセットします。
- 2 バイトファイル番号の場合は、2 バイト (9 と 10) を使います。

 **Note:** 2 バイトファイル番号およびデータベース ID を使用する場合は、コントロールブロックの先頭バイトに X'30' を入力しなければなりません。

レスポンスコード (ACBRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードには、アディション 2 フィールドの右半分にサブコードが返される場合があります。レスポンスコード、サブコードについては『Adabas メッセージおよびコードマニュアル』を参照してください。

レスポンスコード 3 は、エンドオブファイル条件を示します (GET NEXT オプションを使用の場合のみ)。

ISN (ACBISN)

読み込むレコードの ISN を指定します。

GET NEXT (N) オプションを指定すると、Adabas はコマンド ID で識別される ISN リストから ISN を選択し、この ISN のレコードを読み込んで、次のレコードの ISN をこのフィールドに返します。このため、指定した ISN 値は無視されます。

ISN SEQUENCE (I) オプションを指定するときは、このフィールドに ISN 値も指定する必要があります。L1 または L4 コマンドは、指定した ISN のデータレコードをレコードバッファに返します。

F オプションを指定すると、L1 コマンドまたは L4 コマンドはファイルコントロールブロック (FCB) に記録されている最大の ISN をこのフィールドに返します。

レコードが読み込まれると、指定されたオプションに関係なく、そのレコードの ISN がこのフィールドに返されます。オプション F を指定すると、次の番号の未使用の ISN が返されます。

ISN 下限：マルチフェッチレコード数 (ACBISL)

コマンドオプション 1 フィールドに "M" または "O" (マルチフェッチオプション) を指定した場合、このフィールドのゼロ以外の値によって、マルチフェッチ対象の最大レコード数が特定されます。このフィールドの値がゼロの場合は、マルチフェッチ対象のレコード数はレコードバッファと ISN バッファの長さによって制限されます。詳細は、「[マルチフェッチ／プリフェッチ機能の使用](#)」を参照してください。

フォーマットバッファ長 (ACBFBL)

フォーマットバッファ長 (バイト単位)。ユーザープログラムに定義した実際のフォーマットバッファエリアは、この指定長と同じか、それ以上でなければなりません。

レコードバッファ長 (ACBRBL)

レコードバッファ長 (バイト単位)。ユーザープログラムに定義した実際のレコードバッファエリアは、この指定長と同じか、それ以上でなければなりません。

ISN バッファ長：コマンドレベルのマルチフェッチ／プリフェッチオプションのみ (ACBIBL)

ISN バッファ長 (バイト単位)。ユーザープログラムに定義する ISN バッファは、少なくともこの長さと同じ大きさである必要があります。

コマンドオプション 1 (ACBCOP1)

オプション	説明
F (最初の未使用 ISN)	指定したファイルについて、次の番号の未使用の ISN を ISN フィールドに返します。"次の未使用 ISN" は、ファイルコントロールブロック (FCB) を参照することによって特定されます。Adabas 拡張ファイルを読み込むときは F オプションを使用しないでください。
M (マルチフェッチ)	このコマンドのマルチフェッチ処理が実行されます。
O (R オプションを指定してマルチフェッチ処理)	このコマンドのマルチフェッチ処理と R オプションの処理 (下記参照) が実行されます。
P (プリフェッチ)	このコマンドのプリフェッチ処理が実行されます。
R (リターン)	L4 コマンドによる読み込みおよびホールド対象のレコードが使用可能でない場合に、レスポンスコード 145 を返します。

M、O、P のいずれかのオプションを指定すると、コマンドにプリフェッチオプションまたはマルチフェッチオプションが使用されることとなります。ISN 順 (I) オプションまたは GETNEXT (N) オプションのいずれかをコマンドオプション 2 フィールドに指定した場合、そのオプションがアクティブになります。マルチフェッチ／プリフェッチオプションを使用すると単一レコードのフェッチに必要な時間が省かれ、パフォーマンスを向上できます。詳細は、「[マルチフェッチ／プリフェッチ機能の使用](#)」を参照してください。

コマンドオプション 2 (ACBCOP2)


オプション	説明
F (最初の未使用 ISN)	指定したファイルについて、次の番号の未使用の ISN を ISN フィールドに返します。"次の未使用 ISN" は、ファイルコントロールブロック (FCB) を参照することによって特定されます。Adabas 拡張ファイルを読み込むときは F オプションを使用しないでください。
I (ISN 順に読み込み)	ファイルに ISN がある場合、ISN フィールドに指定した ISN で識別されるレコードを読み込みます。ファイルに ISN がない場合は、次の番号の ISN のレコードが読み込まれ、そのレコードの ISN が ISN フィールドに返されます。ファイルに指定の ISN も、次の番号の ISN もない場合は、レコードは読み込まれず、レスポンスコード 3 が返されます。

オプション	説明
N (GET NEXT)	ISN リストの ISN で識別されるレコードが読み込まれます。ユーザーは、L1 または L4 コールに読み込み対象レコードの ISN を指定する必要はありません。Adabas によってリストから ISN が選択され、その ISN で識別されるレコードが読み込まれます。使用する ISN リストは、コマンド ID フィールドで識別される必要があります。また、Sx コマンドによってあらかじめ作成されたものである必要もあります。リスト中の全 ISN を選択し終わると、レスポンスコード 3 が返されます。詳細は「 ISN リスト処理 」を参照してください。

アディクション 2 (圧縮および非圧縮レコード長 (ACBADD2))

コマンドが正常に処理されると、次の情報がこのフィールドに返されます。

- 少なくとも 1 つの有効なフィールド値がレコードバッファにある場合、先頭 2 バイトには、アクセスしたレコードの圧縮レコード長がバイナリ形式で格納されます。
- 後ろ 2 バイトには、フォーマットバッファで選択し、アクセスしたフィールドの非圧縮長がバイナリ形式で格納されます。

 **Note:** プリフェッチ機能の使用時、この長さ情報は返されません。

L1 コマンドまたは L4 コマンドによってゼロ以外のレスポンスコードが返された場合、下位 2 バイトにレスポンスコードの正確な意味を定義したサブコードが含まれていることがあります。レスポンスコードとサブコードについては、『*Adabas メッセージおよびコード*』を参照してください。

アディクション 3：パスワード (ACBADD3)

このフィールドは、Adabas Security または ADAESI パスワードを指定するため使用します。データベース、ファイル、またはフィールドがセキュリティ保護されている場合は、有効なセキュリティパスワードまたは ADAESI パスワードをユーザーは指定する必要があります。パスワードを保護するために、コマンド処理時にアディクション 3 フィールドに空白がセットされます。

アディクション 4：サイファコード (ACBADD4)

このフィールドは、サイファコードを指定するために使用します。ファイルを暗号化する場合は、有効なサイファコードを指定する必要があります。ファイルを暗号化しない場合は、このフィールドに空白をセットする必要があります。

Adabas は、コマンド処理時に、サイファコードに空白をセットし、バージョンコードとデータベース ID をフィールドの下位 3 バイトに返します。詳細については「[コントロールブロックフィールド](#)」を参照してください。

アディクション 5：フォーマット ID、グローバルフォーマット ID (ACBADD5)

このフィールドには、このコマンド用の内部フォーマットバッファを識別する別のフォーマット ID を指定するか、グローバルフォーマット ID を指定します。

アディクション 5 フィールドで 1 バイト目の上位 (左端) ビットに 1 が設定されていない場合、コマンド ID フィールドに指定された値がフォーマット ID として使用されます。

このビットに1がセットされていると、アディション5フィールドの5バイト目から8バイト目までがフォーマットIDとして使用されます。

さらに、アディション5フィールドの第1バイトの最上位（左端）2ビットに、1（B'11'）がセットされると、アディション5フィールドの全8バイトが、グローバルフォーマットIDとして使用されます（つまり、複数ユーザーがこのフォーマットIDを同時に使用できます）。

詳細と例については、「[コマンドID](#)、[フォーマットID](#)、[グローバルフォーマットID](#)」を参照してください。

ACBの例

ここに示した各例で使用している Adabas ファイルの定義については、「[例で使用されているファイル定義](#)」セクションを参照してください。

- [例 1：単一レコードを読み込む](#)
- [例 2：一組のレコードを読み込む](#)
- [例 3：GET NEXT オプションを使用して一組のレコードを読み込む](#)
- [例 4：読み込んでホールドする](#)
- [例 5：READ ISN SEQUENCE オプションを使用して読み込む](#)
- [例 6：マルチプルバリューフィールドおよびピリオディックグループを読み込む](#)

例 1：単一レコードを読み込む

ファイル1のISN4を読み込みます。フィールドAAとABの値が返されます。

コントロールブロック

コマンドコード	L1	
コマンドID	bbbb (空白)	1レコードのみを読み込みます
ファイル番号	1	
ISN	4	
フォーマットバッファ長	6	またはそれ以上
レコードバッファ長	10	またはそれ以上
コマンドオプション2	b	GET NEXT または READ ISN SEQUENCE オプションは使用しません
アディション3	bbbbbbbb (空白)	ファイルはセキュリティ保護されていません
アディション4	bbbbbbbb (空白)	ファイルは暗号化されない

バッファエリア

フォーマットバッファ	AA,AB.
------------	--------

例 2：一組のレコードを読み込む

FIND コマンドですでに ISN がわかっている一組のレコードをファイル 2 から読み込みます。フィールド RA および XB の値を返します。ただし、フィールド XB の値は、長さ 3、フォーマット U で返すものとします。

コントロールブロック

コマンドコード	L1	
コマンド ID	ABCD	各レコードに対し同じフィールドを連続して読み込む場合は、CID を空白以外にすることをお勧めします。
ファイル番号	2	
ISN	n	ISN は、FIND コマンドで作成した ISN リストから取得します。
フォーマットバッファ長	10	またはそれ以上
レコードバッファ長	11	またはそれ以上
コマンドオプション 2	b	GET NEXT または READ ISN SEQUENCE オプションは使用しません
アディション 3	パスワード	ファイルはセキュリティ保護されています
アディション 4	bbbbbbbb (空白)	ファイルは暗号化されない

バッファエリア

フォーマットバッファ	RA,XB,3,U
------------	-----------

*n は、FIND コマンドで作成した ISN リストにある ISN を示します。ISN リストの各 ISN に対して L1 コールを繰り返します。

例 3：GET NEXT オプションを使用して一組のレコードを読み込む

例 2 と同じ要件を GET NEXT オプションでも満たすことができます。

コントロールブロック

コマンドコード	L1	
コマンド ID	ABCD	使用する ISN リストの CID
ファイル番号	2	
ISN	0	ISN リストの第 1 ISN から始まって全 ISN リストが選択されます。
フォーマットバッファ長	10	またはそれ以上

L1/L4 コマンド：読み込み／読み込みおよびレコードのホールド

レコードバッファ長	11	またはそれ以上
コマンドオプション2	N	使用する GET NEXT オプション
アディション3	パスワード	ファイルはセキュリティ保護されています
アディション4	bbbbbbbb (空白)	ファイルは暗号化されない

バッファエリア

フォーマットバッファ	RA,XB,3,U
------------	-----------

ISN リストの各 ISN に対して L1 コールを繰り返します。L1 コール間でコントロールブロックの内容を変更する必要はありません。ISN リスト内の ISN をすべて選択すると、レスポンスコード 3 が返されます。

例 4：読み込んでホールドする

ファイル 2 の ISN 5 を読み込み、更新のためホールドします。フィールド XC および XD の値を返します。

コントロールブロック

コマンドコード	L4	読み込んでホールドします。
コマンド ID	bbbb (空白)	1 レコードのみを読み込みます。
ファイル番号	2	
ISN	5	
フォーマットバッファ長	6	またはそれ以上
レコードバッファ長	14	またはそれ以上
コマンドオプション1	b	レスポンスコード 145 オプション使用しません
コマンドオプション2	b	GET NEXT または READ ISN SEQUENCE オプションは使用しません
アディション3	パスワード	ファイルはセキュリティ保護されています
アディション4	bbbbbbbb (空白)	ファイルは暗号化されない

バッファエリア

フォーマットバッファ	XC,XD
------------	-------

例 5：READ ISN SEQUENCE オプションを使用して読み込む

ファイル 1 を READ ISN SEQUENCE オプションを使用して読み込みます。フィールド AA、AB、および AC の値を返します。

コントロールブロック

コマンドコード	L1	
コマンド ID	BCDE	同じフィールドを返す一連のレコードを読み込む場合は、CID を空白以外にすることをお勧めします。
ファイル番号	1	
ISN	1	ISN 1 が存在しない場合、ファイル内の次に高い ISN のレコードを読み込み、その ISN をこのフィールドに返します。
フォーマットバッファ長	6	またはそれ以上
レコードバッファ長	30	またはそれ以上
コマンドオプション 2	I	呼び出された READ ISN SEQUENCE オプションを読み込みます。
アディション 3	bbbbbbbb (空白)	ファイルはセキュリティ保護されていません
アディション 4	bbbbbbbb (空白)	ファイルは暗号化されない

バッファエリア

フォーマットバッファ	GA,AC.
------------	--------

Adabas は、コントロールブロックの ISN フィールドに、読み込んだレコードの ISN を返します。ISN フィールドに 1 を加算し、L1 コマンドを繰り返すことによって、次に大きい ISN のレコードを読み込むことができます。

例 6：マルチプルバリュースフィールドおよびピリオディックグループを読み込む

ファイル 1 の ISN 2 のレコードを読み込みます。フィールド AA の値、マルチプルバリュースフィールド MF の全値、およびピリオディックグループ GB の全オカレンスを返します。

方法 1：オカレンスが 6 個以上の場合

1. L1 コールを発行して、MF の値の個数および GB の最大オカレンスカウントを取得します。

コントロールブロックフィールド

コマンドコード	L1	
コマンド ID	bbbb (空白)	1 レコードのみを読み込みます
ファイル番号	1	
ISN	2	
フォーマットバッファ長	8	またはそれ以上
レコードバッファ長	2	またはそれ以上
コマンドオプション 2	b	GET NEXT も READ ISN SEQUENCE オプションも使用しません。

バッファエリア

フォーマットバッファ	MFC,GBC
------------	---------

- 上記の L1 コールの結果、MF の値が 4 つあり、GB のオカレンスが 6 つあるレコードが返されたものとして、次のフォーマットバッファを使用して L1 コールを繰り返します。

```
AA,MF01-04,GB01-06
```

どのコールでも、フォーマットバッファおよびレコードバッファの長さはエントリと値をすべて保持できるだけの十分な長さである必要があります。

この手順で一連のレコードを読み込むとき、ステップ 1 では空白以外のコマンド ID を使用します。また、ステップ 2 ではフォーマットバッファの内容がステップ 2 を呼び出すたびに変わる可能性があるため、空白のコマンド ID を使用します。

方法 2：オカレンスが 6 つ未満の場合

値／オカレンスの数が少ない（6 未満）レコードが大部分を占める場合には、例 5 の方法よりも次の方が効率的です。

- MF および GB の個数以外に、MF と GB の値とオカレンスの予想数、およびフィールド AA を要求する L1 コールを発行します。

MF の値の予想個数を 2、GB のオカレンスの予想個数を 3 とすると、ステップ 1 のフォーマットバッファは次のようになります。

```
AA,MFC,GBC,MF01-02,GB01-03.*
```

*90% のレコードで値／オカレンスをすべて取得できるような値が指定された場合、パフォーマンスが最大になります。

- 返された MF の個数が 2 を超えている場合、または GB の個数が 3 を超えている場合、前述のステップ 2 のようなフォーマットバッファを使用すると、その超過分の値やオカレンスを取得できます。それ以外の場合、2 回目のコールは必要ありません。

ACBX インターフェイスダイレクトコール：L1 コマンドおよび L4 コマンド

このsectionでは、L1 コマンドおよび L4 コマンドの ACBX インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの概要](#)
- [コントロールブロックフィールドの説明](#)

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1～2	---	---	---
バージョンインジケータ	3～4	バイナリ	F	U
	5～6	---	---	---
コマンドコード	7～8	英数字	F	U
	9～10	---	---	---
レスポンスコード	11～12	バイナリ	---	A
コマンド ID	13～16	英数字／バイナリ	F	U
データベース ID	17～20	数値	F	U
ファイル番号	21～24	数値	F	U
	25～28	---	---	---
ISN	29～32	バイナリ	F	U*
	33～36	---	---	---
ISN 下限	37～40	バイナリ	F	U
	41～48	---	---	---
コマンドオプション 1	49	英数字	F	U
コマンドオプション 2	50	英数字	F	U
	51～64	---	---	---
アディション 2	65～68	バイナリ	---	A
アディション 3	69～76	英数字／バイナリ	F	A
アディション 4	77～84	英数字	F	A
アディション 5	85～92	英数字／バイナリ	F	U

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	93～114	---	---	---
エラーサブコード	115～116	バイナリ	---	A
	117～144	---	---	---
コマンドタイム	145～152	バイナリ	---	A
ユーザーエリア	153～168	該当なし	---	U
---	169～193	操作不可	---	---

ABD とバッファ

ABD とバッファ	Adabas コール前	Adabas コール後
フォーマット	F	U
レコード	--	A
マルチフェッチ**	--	A

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

* 特殊なオプションの除外

** マルチフェッチオプションを指定した場合のみ、マルチフェッチバッファが必要です。

--- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

バージョンインジケータ (ACBXVER)

F2

コマンドコード (ACBXCMD)

L1 または L4

レスポンスコード (ACBXRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、エラーサブコード (ACBXERRC) フィールドにサブコードを伴う場合があります。詳細は、Adabas メッセージおよびコードのドキュメントを参照してください。

レスポンスコード 3 は、エンドオブファイル条件を示します (GET NEXT オプションを使用の場合のみ)。

コマンド ID (ACBXCID)

一連の L1 または L4 コールを使用して複数のレコードを読み、各コールでフォーマットバッファに同一フィールドを指定する場合、このフィールドは空白およびゼロを除いた値にします。これにより、L1 または L4 の各コールに要する処理時間を短縮できます。

GET NEXT オプションを使用する場合は、使用する ISN リストのコマンド ID をこのフィールドに指定する必要があります。また、GET NEXT オプション使用時、L1 または L4 の各コール間でフォーマットバッファを変更しないでください。

1 レコードだけを読み込む場合やフォーマットバッファを L1 または L4 コール間で変更する場合は、このフィールドを空白にしておきます。

コマンド ID 値が X'FFFFFFFF' の場合、自動的にコマンド ID が生成されます。この場合、Adabas ニュークリアスは X'00000001' から始まるコマンド ID 値を生成し、L1 または L4 コールごとに値を 1 ずつ増加します。ユーザー定義コマンド ID を指定する場合、ユーザーは各コマンド ID を固有のものにする必要があります。

別のフォーマット ID およびグローバルフォーマット ID の使用については、アディション 5 フィールドを参照してください。

データベース ID (ACBXDBID)

このフィールドを使用して、データベース ID を指定します。Adabas コールはこのデータベースに送られます。

このフィールドは 4 バイトのバイナリフィールドですが、現時点では 2 バイトのデータベース ID のみがサポートされています。したがって、データベース ID はフィールドの下位部分（右端のバイト）に指定し、その前にバイナリの 0 を付加する必要があります。

このフィールドがバイナリの 0 に設定されている場合は、Adabas API は DDCARD 入力データで指定された ADARUN カードのデータベース ID か、リンクされているカリンクルーチンによりロードされた LNKGBLS モジュールで指定されているデフォルトのデータベース ID 値のいずれかを使用します。

ファイル番号 (ACBXFNR)

このフィールドを使用して、Adabas コールで送る必要のあるファイル数を指定します。

このフィールドは 4 バイトのバイナリフィールドですが、ファイル番号はフィールドの下位部分（右端のバイト）に指定し、その前にバイナリの 0 を付加する必要があります。

ISN (ACBXISN)

このフィールドは、読み込むレコードの ISN を指定するのに使用します。

GET NEXT (N) オプションを指定すると、Adabas はコマンド ID で識別される ISN リストから ISN を選択し、この ISN のレコードを読み込んで、次のレコードの ISN をこのフィールドに返します。このため、指定した ISN 値は無視されます。

ISN SEQUENCE (I) オプションを指定するときは、このフィールドに ISN 値も指定する必要があります。L1 または L4 コマンドは、指定した ISN のデータレコードをレコードバッファに返します。

F オプションを指定すると、L1 コマンドまたは L4 コマンドはファイルコントロールブロック (FCB) に記録されている最大の ISN をこのフィールドに返します。

レコードが読み込まれると、指定されたオプションに関係なく、そのレコードの ISN がこのフィールドに返されます。オプション F を指定すると、次の番号の未使用の ISN が返されます。

ACBXISN フィールドは、8 バイトの ACBXISNG フィールドに埋め込まれた 4 バイトのバイナリフィールドで、まだ使用されていません。ACBXISNG フィールドの上位部分には、バイナリの 0 を設定します。

ISN 下限：マルチフェッチレコード数 (ACBXISL)

コマンドオプション 1 フィールドに "M" または "O" (マルチフェッチオプション) を指定した場合、このフィールドのゼロ以外の値によって、マルチフェッチ対象の最大レコード数が特定されます。このフィールドの値がゼロの場合は、マルチフェッチ対象のレコード数はレコードバッファと ISN バッファの長さによって制限されます。詳細は、「[マルチフェッチ／プリフェッチ機能の使用](#)」を参照してください。

ACBXISL フィールドは、8 バイトの ACBXISLG フィールドに埋め込まれた 4 バイトのバイナリフィールドで、まだ使用されていません。ACBXISLG フィールドの上位部分には、バイナリの 0 を設定します。

コマンドオプション 1 (ACBXCOP1)

オプション	説明
F (最初の未使用 ISN)	指定したファイルについて、次の番号の未使用の ISN を ISN フィールドに返します。"次の未使用 ISN" は、ファイルコントロールブロック (FCB) を参照することによって特定されます。Adabas 拡張ファイルを読み込むときは F オプションを使用しないでください。
M (マルチフェッチ)	このコマンドのマルチフェッチ処理が実行されます。
O (R オプションを指定してマルチフェッチ処理)	このコマンドのマルチフェッチ処理と R オプションの処理 (下記参照) が実行されます。
R (リターン)	L4 コマンドによる読み込みおよびホールド対象のレコードが使用可能でない場合に、レスポンスコード 145 を返します。

"M" または "O" オプションを指定すると、コマンドにマルチフェッチオプションが使用されることとなります。ISN 順 (I) オプションまたは GET NEXT (N) オプションのいずれかをコマンドオプション 2 フィールドに指定した場合、そのオプションがアクティブになります。マルチフェッチオプションを使用すると、単一レコードをフェッチする時間が必要なくなるため、パフォーマンスを向上させることができます。詳細は、「[マルチフェッチ／プリフェッチ機能の使用](#)」を参照してください。


コマンドオプション 2 (ACBXCOP2)

オプション	説明
F (最初の未使用 ISN)	指定したファイルについて、次の番号の未使用の ISN を ISN フィールドに返します。"次の未使用 ISN" は、ファイルコントロールブロック (FCB) を参照することによって特定されます。Adabas 拡張ファイルを読み込むときは F オプションを使用しないでください。
I (ISN 順に読み込み)	ファイルに ISN がある場合、ISN フィールドに指定した ISN で識別されるレコードを読み込みます。ファイルに ISN がない場合は、次の番号の ISN のレコードが読み込まれ、そのレコードの ISN が ISN フィールドに返されます。ファイルに指定の ISN も、次の番号の ISN もない場合は、レコードは読み込まれず、レスポンスコード 3 が返されます。
N (GET NEXT)	ISN リストの ISN で識別されるレコードが読み込まれます。ユーザーは、L1 または L4 コールに読み込み対象レコードの ISN を指定する必要はありません。Adabas によってリストから ISN が選択され、その ISN で識別されるレコードが読み込まれます。使用する ISN リストは、コマンド ID フィールドで識別され、かつ Sx コマンドで以前に作成したものである必要があります。リスト中の全 ISN を選択し終わると、レスポンスコード 3 が返されます。詳細は「 ISN リスト処理 」を参照してください。

アディショナル 2：圧縮および非圧縮レコード長 (ACBXADD2)

コマンドが正常に処理されると、次の情報がこのフィールドに返されます。

- 少なくとも 1 つの有効なフィールド値がレコードバッファにある場合、先頭 2 バイトには、アクセスしたレコードの圧縮レコード長がバイナリ形式で格納されます。
- 後ろ 2 バイトには、フォーマットバッファで選択し、アクセスしたフィールドの非圧縮長がバイナリ形式で格納されます。

 **Note:** プリフェッチ機能の使用時、この長さ情報は返されません。

アディショナル 3：パスワード (ACBXADD3)

このフィールドは、Adabas Security または ADAESI パスワードを指定するため使用します。データベース、ファイル、またはフィールドがセキュリティ保護されている場合は、有効なセキュリティパスワードまたは ADAESI パスワードをユーザーは指定する必要があります。パスワードを保護するために、コマンド処理時にアディショナル 3 フィールドに空白がセットされます。

アディショナル 4：サイファコード (ACBXADD4)

このフィールドは、サイファコードを指定するために使用します。ファイルを暗号化する場合は、有効なサイファコードを指定する必要があります。ファイルを暗号化しない場合は、このフィールドに空白をセットする必要があります。

Adabas は、コマンド処理時に、サイファコードに空白をセットし、バージョンコードとデータベース ID をフィールドの下位 3 バイトに返します。詳細については「[コントロールブロックフィールド](#)」を参照してください。

アディション5：フォーマット ID、グローバルフォーマット ID (ACBXADD5)

このフィールドには、このコマンド用の内部フォーマットバッファを識別する別のフォーマット ID を指定するか、グローバルフォーマット ID を指定します。

アディション5 フィールドで1バイト目の上位（左端）ビットに1が設定されていない場合、コマンド ID フィールドに指定された値がフォーマット ID として使用されます。

このビットに1がセットされていると、アディション5 フィールドの5バイト目から8バイト目までがフォーマット ID として使用されます。

さらに、アディション5 フィールドの第1バイトの最上位（左端）2ビットに、1 (B'11') がセットされると、アディション5 フィールドの全8バイトが、グローバルフォーマット ID として使用されます（つまり、複数ユーザーがこのフォーマット ID を同時に使用できます）。

詳細と例については、「[コマンドID、フォーマットID、グローバルフォーマットID](#)」セクションを参照してください。

エラーサブコード (ACBXERRC)

コマンドがゼロ以外のレスポンスコードを返したときは、このフィールドにレスポンスコードの正確な意味を定義したサブコードが含まれます。レスポンスコードとサブコードについては、『[Adabas メッセージおよびコード](#)』を参照してください。

バッファ

L1 コマンドおよび L4 コマンドには、次のバッファが適用されます。

- [フォーマットバッファ](#)
- [レコードバッファ](#)
- [ISN バッファ](#)
- [マルチフェッチバッファ](#)

フォーマットバッファ

このバッファに読み込むフィールドを限定します。フォーマットバッファの構文および例については、「[バッファの定義](#)」を参照してください。

"C."がフォーマットバッファの先頭2桁にあると、レコードは非圧縮ではなく圧縮された形式で返されます。

レコードバッファ

Adabasは、バッファに要求されたフィールドの値をこのバッファに返します。ユーザーがフォーマットバッファに長さやフォーマットを標準以外の値に指定していない限り、どの値もフィールドの標準のフォーマットと長さで返されます。

ISN バッファ

ISN バッファが使用されるのは、コマンドレベルのマルチフェッチオプションを要求した場合、および **ACB** インターフェイスを使用してダイレクトコールを行う場合に限られます。詳細は、「[マルチフェッチ／プリフェッチ機能の使用](#)」を参照してください。マルチフェッチを使用した場合、L1 コマンドおよび L4 コマンドはディスクリプタの要素を最大 16 バイトの長さまで ISN バッファに返します（「[BT/ET のマルチフェッチ処理](#)」セクション参照）。

マルチフェッチバッファ

マルチフェッチバッファが使用されるのは、コマンドレベルのマルチフェッチオプションを要求した場合、および **ACBX** インターフェイスを使用してダイレクトコールを行う場合に限られます。詳細は、「[マルチフェッチ／プリフェッチ機能の使用](#)」を参照してください。マルチフェッチを使用した場合、L1 コマンドおよび L4 コマンドはディスクリプタの要素を最大 16 バイトの長さまでマルチフェッチバッファに返します（「[BT/ET のマルチフェッチ処理](#)」セクション参照）。

33

L2/L5 コマンド：物理順に読み込み

■ 機能および使用	294
■ ACB インターフェイスダイレクトコール：L2 コマンドおよび L5 コマンド	294
■ ACBX インターフェイスダイレクトコール：L2 コマンドおよび L5 コマンド	300
■ バッファ	305
■ その他の考慮事項	306

L2コマンドとL5コマンドは、データストレージ内で物理的に並んでいる順にレコードを読み込みます。

このchapterでは、次のトピックについて説明します。

機能および使用

L5コマンドはL2コマンドと同じ機能を持ちます。ただし、さらに各レコードをホールド状態にします。読み込んでホールドするレコードが現在他のユーザーにホールドされている場合は、レコードが使用可能になるまで待機状態になります。コマンドオプション1フィールドに "R" を指定して L5 コマンドを発行した場合、レコードが使用可能でないと、レスポンスコード 145 が返されます。

L2コマンドとL5コマンドでは、論理的な順序でレコードを読み込みません。ただし、レコードが最初から特別な論理的順序でロードされ、この順序を変更するようなファイル更新が行われていない場合は、その限りではありません。

L2 コマンドと L5 コマンドは、アソシエータにアクセスする必要がないため（ただし、L3 コマンドでは必要）、最適な速度でファイル全体を読み込むことができます。全物理ブロックが順番に読み込まれます。

読み込むファイルと、値を取得する各レコード内のフィールドを指定します。そのフィールドをフォーマットバッファに指定します。レコードバッファには、要求されたフィールド値が返されます。

マルチフェッチ/プリフェッチ オプションを指定すると、複数のシーケンシャルレコードへの事前アクセスが可能になるため、処理全体の時間が短くなり、単一レコードをフェッチする時間が必要なくなります。マルチフェッチまたはプリフェッチを使用可能にするには、コマンドオプション1フィールドに "M"、"O"（マルチフェッチ）、または "P"（プリフェッチ）を指定します。詳細は、「[マルチフェッチ/プリフェッチ機能の使用](#)」を参照してください。

COMPRESSED オプション（フォーマットバッファに "C." を指定することで設定）を使用すると、読み込んだレコードが Adabas 内部に格納する圧縮形式で返されます。

ACB インターフェイスダイレクトコール：L2コマンドおよびL5コマンド

このsectionでは、L2 コマンドおよび L5 コマンドの ACB インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの概要](#)
- [コントロールブロックフィールドの説明](#)

■ ACB の例

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	--	--	--
コマンドコード	3~4	英数字	F	U
コマンド ID	5~8	英数字	F	U
ファイル番号	9~10	バイナリ	F	U
レスポンスコード	11~12	バイナリ	--	A
ISN	13~16	バイナリ	F	A
ISN 下限	17~20	バイナリ	F	U
	21~24	--	--	--
フォーマットバッファ長	25~26	バイナリ	F	U
レコードバッファ長	27~28	バイナリ	F	U
	29~32	--	--	--
ISN バッファ長*	33~34	バイナリ	F	U
コマンドオプション 1	35	英数字	F	U
	36~44	--	--	--
アデプション 2	45~48	バイナリ/バイナリ	--	A
アデプション 3	49~56	英数字	F	A
アデプション 4	57~64	英数字	F	A
アデプション 5	65~72	英数字	F	U
コマンドタイム	73~76	バイナリ	--	A
ユーザーエリア	77~80	--	--	U

バッファエリア

バッファ	Adabas コール前	Adabas コール後
フォーマット	F**	U
レコード	--	A
ISN *	F	U

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

* ISN バッファおよび長さは、マルチフェッチオプションまたはプリフェッチオプションを指定した場合にのみ必須です。

** 圧縮オプション制御文字 C を含めることができます。

-- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

コマンドコード (ACBCMD)

L2 または L5

コマンド ID (ACBCID)

このフィールドには、空白およびゼロを除く値を指定する必要があります。Adabas が正しい物理順でレコードを返し、フォーマットバッファを繰り返し解釈するのを避けるために使用されます。この値は、ファイルの順次読み込み処理中は変更しないでください。

このフィールドの第 1 バイトには 16 進数 'FF' にセットしないでください。

コマンド ID 値が X'FFFFFFFF' の場合、自動的にコマンド ID が生成されます。この場合 Adabas ニュークリアスは X'00000001' から始まるコマンド ID 値を生成し、L1 または L5 コールごとに値を 1 ずつ増加します。ユーザー定義コマンド ID を指定する場合、ユーザーは各コマンド ID を固有のものにする必要があります。

別のフォーマット ID およびグローバルフォーマット ID の使用については、アディション 5 フィールドを参照してください。

ファイル番号 (ACBFNR)

読み込むファイルの番号を 2 進数で指定します。物理ダイレクトコールの場合は、次のようファイル番号を指定します。

- 1 バイトファイル番号の場合は、ファイル番号を第 2 バイト (10) に入力します。第 1 バイト (9) は、バイナリの 0 (B'0000 0000') をセットします。
- 2 バイトファイル番号の場合は、2 バイト (9 と 10) を使います。



Note: 2 バイトファイル番号およびデータベース ID を使用する場合は、コントロールブロックの先頭バイトに X'30' を入力しなければなりません。

レスポンスコード (ACBRSP)

Adabasはこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード0は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、アディション2フィールドの下位2バイトにサブコードを伴う場合があります。詳細は『Adabas メッセージおよびコード』を参照してください。

レスポンスコード3は、エンドオブファイル (EOF) 条件が検出されたことを示します。

ISN (ACBISN)

このフィールドを最初のL2またはL5コールの前にゼロに設定すると、ファイルの第1物理ブロック内の第1レコードから順次読み込みが開始されます。

このフィールドを最初のL2またはL5コールの前にISN値に設定すると、指定されたISNのレコードに対して物理的に直後にあるレコードから順次読み込みが開始されます。指定ISNはファイル中に存在している必要ありません。

ユーザーは、最初のL2またはL5コールの後にこのフィールドを修正する必要はありません。読み込んだレコードのISNがこのフィールドに返されます。

ISN 下限：マルチフェッチレコード数 (ACBISL)

コマンドオプション1フィールドに"M"または"O" (マルチフェッチオプション) を指定した場合、このフィールドのゼロ以外の値によって、マルチフェッチする最大レコード数が特定されます。このフィールドの値がゼロの場合は、マルチフェッチ対象のレコード数はレコードバッファとISNバッファの長さによって制限されます。詳細は、「[マルチフェッチ／プリフェッチオプション使用](#)」を参照してください。

フォーマットバッファ長 (ACBFBL)

フォーマットバッファ長 (バイト単位)。ユーザープログラムに定義するフォーマットバッファエリアは、この指定長と同じか、それ以上でなければなりません。コマンドオプション1フィールドにマルチフェッチオプションMまたはOを指定した場合、この値は32KB未満である必要があります。

レコードバッファ長 (ACBRBL)

レコードバッファ長 (バイト単位)。ユーザープログラムに定義するレコードバッファエリアは、この指定長と同じか、それ以上でなければなりません。コマンドオプション1フィールドにマルチフェッチオプションMまたはOを指定した場合、この値は32KB未満である必要があります。

ISN バッファ長：コマンドレベルのマルチフェッチ／プリフェッチオプションのみ (ACBIBL)

ISN バッファ長 (バイト単位)。ユーザープログラムに定義するISNバッファエリアは、この指定長と同じ大きさか、それ以上である必要があります。

コマンドオプション 1 (ACBCOP1)

単一レコードのフェッチに必要な時間を少なくしてパフォーマンスを改善するには、コマンドのマルチフェッチ処理またはプリフェッチ処理を可能にするオプション M、O、または P を設定してください。詳細は、「[マルチフェッチ／プリフェッチ機能の使用](#)」を参照してください。

オプション	説明
M	マルチフェッチを可能にします。
O	以下に説明するように "R" オプション付きでマルチフェッチを使用します。
P	プリフェッチを使用します。
R (リターン)	L5 コマンドによる読み込みおよびホールド対象のレコードが使用可能でない場合に、レスポンスコード 145 を返します。

アディション 2 (圧縮および非圧縮レコード長 (ACBADD2))

コマンドが正常に処理されると、次の情報がこのフィールドに返されます。

- 少なくとも 1 つの有効なフィールド値がレコードバッファにある場合、先頭 2 バイトには、アクセスしたレコードの圧縮レコード長がバイナリ形式で格納されます。
- 後ろ 2 バイトには、フォーマットバッファで選択し、アクセスしたフィールドの非圧縮長がバイナリ形式で格納されます。



Note: プリフェッチ機能の使用時、この長さ情報は返されません。

L2 コマンドまたは L5 コマンドによってゼロ以外のレスポンスコードが返された場合、下位 2 バイトには、正確なレスポンスコードの意味を定義するサブコードが含まれていることがあります。レスポンスコードとサブコードについては、『*Adabas メッセージおよびコード*』を参照してください。

アディション 3：パスワード (ACBADD3)

このフィールドには、Adabas Security または Adabas SAF Security パスワードを指定します。データベース、ファイル、またはフィールドがセキュリティ保護されている場合は、ユーザーは該当するセキュリティパスワードを指定する必要があります。パスワードを保護するために、コマンド処理時にアディション 3 フィールドに空白がセットされます。

アディション 4：サイファコード (ACBADD4)

このフィールドは、サイファコードを指定するために使用します。ファイルを暗号化する場合は、有効なサイファコードを指定する必要があります。ファイルを暗号化しない場合は、このフィールドに空白をセットする必要があります。

Adabas は、コマンド処理時に、サイファコードに空白をセットし、バージョンコードとデータベース ID をフィールドの下位 3 バイトに返します。詳細については「[コントロールブロックフィールド](#)」を参照してください。

アディクション5：フォーマット ID、グローバルフォーマット ID (ACBADD5)

このフィールドには、このコマンド用の内部フォーマットバッファを識別する別のフォーマット ID を指定するか、グローバルフォーマット ID を指定します。

アディクション5 フィールドで1バイト目の上位（左端）ビットに1が設定されていない場合、コマンド ID フィールドに指定された値がフォーマット ID として使用されます。

このビットに1がセットされていると、アディクション5 フィールドの5バイト目から8バイト目までがフォーマット ID として使用されます。

さらに、アディクション5 フィールドの第1バイトの最上位（左端）2ビットに、1 (B'11') がセットされると、アディクション5 フィールドの全8バイトが、グローバルフォーマット ID として使用されます（つまり、複数ユーザーがこのフォーマット ID を同時に使用できます）。

詳細と例については、「[コマンドID、フォーマットID、グローバルフォーマットID](#)」を参照してください。

ACB の例

- [例 1](#)
- [例 2](#)

例 1

ファイル 2 を物理順に読み込みます。各レコードの全フィールド値が返されます。

コントロールブロック

コマンドコード	L2	
コマンド ID	EXL2	空白以外の CID が要求されます。
ファイル番号	2	
ISN	0	全レコードを読み込みます
フォーマットバッファ長	3	またはそれ以上
レコードバッファ長	49	またはそれ以上
コマンドオプション1	b	RETURN オプションを使用しません。
アディクション3	パスワード	ファイル2はセキュリティ保護されています。
アディクション4	bbbbbbbb (空白)	ファイルは暗号化されない

バッファエリア

フォーマットバッファ RG

L2 コールを繰り返して各レコードを取得します。ISN フィールドをコール間で修正する必要はありません。

例 2

ファイル 2 を物理順に読み込みます。フィールド RA、XA、および XB（3 バイトのアンパック）の値を返します。読み込んだ各レコードは、更新のためホールド状態になります。

コントロールブロック

コマンドコード	L5	
コマンド ID	EXL5	空白以外の CID が要求されます。
ファイル番号	2	
ISN	0	全レコードを読み込みます
フォーマットバッファ長	13	またはそれ以上
レコードバッファ長	21	またはそれ以上
コマンドオプション 1	b	レスポンスコード 145 オプション使用しません
アディション 3	パスワード	ファイル 2 はセキュリティ保護されています。
アディション 4	bbbbbbbbb (空白)	ファイル 2 は暗号化されていません

バッファエリア

フォーマットバッファ RA,XA,XB,3,U

L5 コールを繰り返して各レコードを取得します。ISN フィールドを L5 コール間で修正する必要はありません。

論理トランザクションを終了し、ホールドしたレコードを解放するため、ET ロジックユーザーは ET コマンドを発行します。非 ET ユーザーは、A4、E4、または RI コマンドを発行して、ホールドしたレコードを解放します。

ACBX インターフェイスダイレクトコール：L2 コマンドおよび L5 コマンド

この section では、L2 コマンドおよび L5 コマンドの ACBX インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- **コントロールブロックとバッファの概要**

■ コントロールブロックフィールドの説明

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	---	---	---
バージョンインジケータ	3~4	バイナリ	F	U
	5~6	---	---	---
コマンドコード	7~8	英数字	F	U
	9~10	---	---	---
レスポンスコード	11~12	バイナリ	---	A
コマンド ID	13~16	英数字/バイナリ	F	U
データベース ID	17~20	数値	F	U
ファイル番号	21~24	数値	F	U
	25~28	---	---	---
ISN	29~32	バイナリ	F	A
	33~36	---	---	---
ISN 下限	37~40	バイナリ	F	U
	41~48	---	---	---
コマンドオプション 1	49	英数字	F	U
	50~64	---	---	---
アディクション 2	65~68	バイナリ	---	A
アディクション 3	69~76	英数字/バイナリ	F	A
アディクション 4	77~84	英数字	F	A
アディクション 5	85~92	英数字/バイナリ	F	U
	93~114	---	---	---
エラーサブコード	115~116	バイナリ	---	A
	117~144	---	---	---
コマンドタイム	145~152	バイナリ	---	A
ユーザーエリア	153~168	該当なし	---	U
---	169~193	操作不可	---	---

ABD とバッファ

ABD とバッファ	Adabas コール前	Adabas コール後
フォーマット	F**	U
レコード	--	A
マルチフェッチ*	F	U

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

* マルチフェッチバッファは、マルチフェッチオプションを指定した場合にのみ必須です。

** 圧縮オプション制御文字 "C" を含んでもかまいません。

-- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

バージョンインジケータ (ACBXVER)

F2

コマンドコード (ACBXCMD)

L2 または L5

レスポンスコード (ACBXRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、エラーサブコード (ACBXERRC) フィールドにサブコードを伴う場合があります。詳細は、Adabas メッセージおよびコードのドキュメントを参照してください。

レスポンスコード 3 は、エンドオブファイル (EOF) 条件が検出されたことを示します。

コマンド ID (ACBXCID)

このフィールドには、空白およびゼロを除く値を指定する必要があります。Adabas が正しい物理順でレコードを返し、フォーマットバッファを繰り返し解釈するのを避けるために使用されます。この値は、ファイルの順次読み込み処理中は変更しないでください。

このフィールドの第 1 バイトには 16 進数 'FF' にセットしないでください。

コマンド ID 値が X'FFFFFFFF' の場合、自動的にコマンド ID が生成されます。この場合 Adabas ニュークリアスは X'00000001' から始まるコマンド ID 値を生成し、L1 または L5 コールごとに値を 1 ずつ増加します。ユーザー定義コマンド ID を指定する場合、ユーザーは各コマンド ID を固有のものにする必要があります。

別のフォーマット ID およびグローバルフォーマット ID の使用については、アディクション 5 フィールドを参照してください。

データベース ID (ACBXDBID)

このフィールドを使用して、データベース ID を指定します。Adabas コールはこのデータベースに送られます。

このフィールドは 4 バイトのバイナリフィールドですが、現時点では 2 バイトのデータベース ID のみがサポートされています。したがって、データベース ID はフィールドの下位部分 (右端のバイト) に指定し、その前にバイナリの 0 を付加する必要があります。

このフィールドがバイナリの 0 に設定されている場合は、Adabas API は DDCARD 入力データで指定された ADARUN カードのデータベース ID か、リンクされているカリンクルーチンによりロードされた LNKGBLS モジュールで指定されているデフォルトのデータベース ID 値のいずれかを使用します。

ファイル番号 (ACBXFNR)

このフィールドを使用して、Adabas コールで送る必要のあるファイル数を指定します。

このフィールドは 4 バイトのバイナリフィールドですが、ファイル番号はフィールドの下位部分 (右端のバイト) に指定し、その前にバイナリの 0 を付加する必要があります。

ISN (ACBXISN)

このフィールドを最初の L2 または L5 コールの前にゼロに設定すると、ファイルの第 1 物理ブロック内の第 1 レコードから順次読み込みが開始されます。

このフィールドを最初の L2 または L5 コールの前に ISN 値に設定すると、指定された ISN のレコードに対して物理的に直後にあるレコードから順次読み込みが開始されます。指定 ISN はファイル中に存在している必要ありません。

ユーザーは、最初の L2 または L5 コールの後にこのフィールドを修正する必要はありません。読み込んだレコードの ISN がこのフィールドに返されます。

ACBXISN フィールドは、8 バイトの ACBXISNG フィールドに埋め込まれた 4 バイトのバイナリフィールドで、まだ使用されていません。ACBXISNG フィールドの上位部分には、バイナリの 0 を設定します。

ISN 下限：マルチフェッチレコード数 (ACBXISL)

コマンドオプション 1 フィールドに "M" または "O" (マルチフェッチオプション) を指定した場合、このフィールドのゼロ以外の値によって、マルチフェッチする最大レコード数が特定されます。この値がゼロであれば、マルチフェッチするレコード数は [レコードバッファとマルチフェッチバッファの長さによって制限](#)されます。詳細は、「[マルチフェッチ／プリフェッチ機能の使用](#)」を参照してください。

ACBXISL フィールドは、8 バイトの ACBXISLG フィールドに埋め込まれた 4 バイトのバイナリフィールドで、まだ使用されていません。ACBXISLG フィールドの上位部分には、バイナリの 0 を設定します。

コマンドオプション 1 (ACBXCOP1)


単一レコードのフェッチに必要な時間を少なくしてパフォーマンスを改善するには、コマンドのマルチフェッチ処理またはプリフェッチ処理を可能にするオプション M、O、または P を設定してください。詳細は、「[マルチフェッチ／プリフェッチ機能の使用](#)」を参照してください。

オプション	説明
M	マルチフェッチを可能にします。
O	以下に説明するように "R" オプション付きでマルチフェッチを使用可能にします。
R (リターン)	L5 コマンドによる読み込みおよびホールド対象のレコードが使用可能でない場合に、レスポンスコード 145 を返します。

アディション 2：圧縮および非圧縮レコード長 (ACBXADD2)

コマンドが正常に処理されると、次の情報がこのフィールドに返されます。

- 少なくとも 1 つの有効なフィールド値がレコードバッファにある場合、先頭 2 バイトには、アクセスしたレコードの圧縮レコード長がバイナリ形式で格納されます。
- 後ろ 2 バイトには、フォーマットバッファで選択し、アクセスしたフィールドの非圧縮長がバイナリ形式で格納されます。

 **Note:** プリフェッチ機能の使用時、この長さ情報は返されません。

アディション 3：パスワード (ACBXADD3)

このフィールドには、Adabas Security または Adabas SAF Security パスワードを指定します。データベース、ファイル、またはフィールドがセキュリティ保護されている場合は、ユーザーは該当するセキュリティパスワードを指定する必要があります。パスワードを保護するために、コマンド処理時にアディション 3 フィールドに空白がセットされます。

アディション 4：サイファコード (ACBXADD4)

このフィールドは、サイファコードを指定するために使用します。ファイルを暗号化する場合は、有効なサイファコードを指定する必要があります。ファイルを暗号化しない場合は、このフィールドに空白をセットする必要があります。

Adabas は、コマンド処理時に、サイファコードに空白をセットし、バージョンコードとデータベース ID をフィールドの下位 3 バイトに返します。詳細については「[コントロールブロックフィールド](#)」を参照してください。

アディション 5：フォーマット ID、グローバルフォーマット ID (ACBXADD5)

このフィールドには、このコマンド用の内部フォーマットバッファを識別する別のフォーマット ID を指定するか、グローバルフォーマット ID を指定します。

アディション 5 フィールドで 1 バイト目の上位 (左端) ビットに 1 が設定されていない場合、コマンド ID フィールドに指定された値がフォーマット ID として使用されます。

このビットに1がセットされていると、アディション5フィールドの5バイト目から8バイト目までがフォーマットIDとして使用されます。

さらに、アディション5フィールドの第1バイトの最上位（左端）2ビットに、1（B'11'）がセットされると、アディション5フィールドの全8バイトが、グローバルフォーマットIDとして使用されます（つまり、複数ユーザーがこのフォーマットIDを同時に使用できます）。

詳細と例については、「[コマンドID](#)、[フォーマットID](#)、[グローバルフォーマットID](#)」を参照してください。

エラーサブコード（ACBXERRC）

コマンドがゼロ以外のレスポンスコードを返したときは、このフィールドにレスポンスコードの正確な意味を定義したサブコードが含まれます。レスポンスコードとサブコードについては、『[Adabas メッセージおよびコード](#)』を参照してください。

バッファ

L2 コマンドおよび L5 コマンドには、次のバッファが適用されます。

- [フォーマットバッファ](#)
- [レコードバッファ](#)
- [ISN バッファ](#)
- [マルチフェッチバッファ](#)

フォーマットバッファ

値を取得するフィールドをこのバッファに指定します。フォーマットバッファの構文および例については、「[バッファの定義](#)」を参照してください。

"C." がフォーマットバッファの先頭2桁にあると、COMPRESSED オプションが有効になります。これにより、レコードは、圧縮解除された形式ではなく圧縮された形式で返されます。

一度 L2 コマンドまたは L5 コマンドを発行した後で、フォーマットバッファを変更することはできません。

レコードバッファ

Adabasは、バッファに要求されたフィールドの値をこのバッファに返します。ユーザーがフォーマットバッファに長さやフォーマットを標準以外の値に指定していない限り、すべての値は各フィールドの標準長およびフォーマットに従って返されます。

ISN バッファ

ISN バッファが使用されるのは、コマンドレベルのマルチフェッチオプションを使用した場合、および **ACB インターフェイス**ダイレクトコールを使用してコマンドを発行した場合に限られます。詳細は、「[マルチフェッチ／プリフェッチ機能の使用](#)」を参照してください。マルチフェッチを使用した場合、L2コマンドまたはL5コマンドはレコードディスクリプタエレメントを最大16バイトの長さまでISN バッファに返します（「[BT/ET のマルチフェッチ処理](#)」セクション参照）。

マルチフェッチバッファ

マルチフェッチバッファが使用されるのは、コマンドレベルのマルチフェッチオプションを使用した場合、および **ACBX インターフェイス**ダイレクトコールを使用してコマンドを発行した場合に限られます。詳細は、「[マルチフェッチ／プリフェッチ機能の使用](#)」を参照してください。マルチフェッチを使用した場合、L2コマンドまたはL5コマンドはレコードディスクリプタエレメントを最大16バイトの長さまでマルチフェッチバッファに返します（「[BT/ET のマルチフェッチ処理](#)」セクション参照）。

その他の考慮事項

L2 コマンドと L5 コマンドを使用する場合、次の事項も考慮しなければなりません。

1. L2 コマンドと L5 コマンドで使用したコマンド ID は、Adabas により内部的に保存および使用されます。この ID は、エンドオブファイル条件が検出されるか、RC コマンドか CL コマンドが発行されるか、Adabas セッションが終了したときに解放されます。同じコマンド ID は、解放されるまで他の読み込みシーケンシャルコマンドでは使用できません。
2. L2 または L5 コマンドを使用して読み込んでいるファイルに対して、レコードの更新と削除を行うことができます。Adabas では、ユーザーに返した最後のレコードと次に返すレコードに関する情報を保持しているため、ユーザーがその間にレコードを更新または削除しても、正しく次のレコードを返します。
3. L2 コマンドまたは L5 コマンドで読み込もうとするファイルを別のユーザーが更新している場合は、L2 コマンドまたは L5 コマンドでそのファイルを読み込んでも、レコードが返されないことがあります。また、ファイルの順次読み込み中に同じレコードが複数回返されることもあります。

34 L3/L6 コマンド：論理順に読み込み

■ 機能および使用	308
■ ACB インターフェイスダイレクトコール：L3 コマンドおよび L6 コマンド	309
■ ACBX インターフェイスダイレクトコール：L3 コマンドおよび L6 コマンド	319
■ バッファ	326
■ その他の考慮事項	331

L3コマンドとL6コマンドは、指定のディスクリプタの値の順番に従ってファイルを論理順に読み込むのに使用します。

このchapterでは、次のトピックについて説明します。

機能および使用

次の情報を指定します。

- 読み込むファイル
- 読み込む順序を制御する検索ディスクリプタ（フォネティックディスクリプタおよびピリオディックグループ内のディスクリプタ、またはピリオディックグループ内のフィールドから作られたディスクリプタは使用できません）
- 値を取得する各レコード内のフィールド

各 L3 コマンドまたは L6 コマンドは、あるレコードの 1 つ以上のフィールドに対して要求された値を **レコードバッファ** に返します。コマンドを繰り返すと、各フィールドの値が検索ディスクリプタ値の順に返されます。

昇順および降順オプションを使用すると、レコードを昇順または降順に読み込むことを指定できます。昇順、降順、および開始値オプションを使用すると、順次読み込みを開始する前または順次読み込みの間に、指定された値への位置付けを行うことができます。これにより、レコードを 1 つずつ読み込むことなく、ファイル内の特定のレコードへの位置付けを行うことができます。

マルチフェッチ/プリフェッチオプションを指定すると、1 つ以上のシーケンシャルレコードへの事前アクセスが可能になるため、処理全体の時間が短縮され、単一レコードをフェッチする時間が必要なくなります。マルチフェッチおよびプリフェッチを使用可能にするには、コマンドオプション 1 フィールドに "M" または "O"（マルチフェッチ）を指定します。詳細は、「[マルチフェッチ/プリフェッチ機能の使用](#)」を参照してください。

フォーマットバッファに "C." があると、読み込んだ各レコードのフィールド値が圧縮形式で返されます。それ以外の場合は非圧縮形式で返されます。

L3 コマンドの処理では、内部テーブル（シーケンシャルコマンドのテーブル）を使用して、最後に返された値および ISN の情報が保持されます。標準ファイルと拡張ファイルでは操作方法が異なり、パフォーマンスの理由から、それぞれの値および ISN はテーブル内で処理が異なります。特に、標準ファイルのエントリは、最後のコールで返された値および ISN を保持します。拡張ファイルのエントリは、次のコールでアプリケーションに返す必要のある値および ISN を保持します。つまり、標準ファイルに対して更新が同時に発生した場合は、最後に挿入された値/ISN が最後に返された値/ISN と同じかまたはそれ以上である限り、最後に挿入された値と ISN の組み合わせが L3 コマンドによって返されます。拡張ファイルに対して更新が同時に発生した場合は、最後に挿入された値/ISN が、次に返される値/ISN としてシーケンシャルコマンド

のテーブルのエントリに保存されている値と同じかまたはそれ以上である限り、最後に挿入された値と ISN の組み合わせが L3 コマンドによって返されます。

L6 コマンドは L3 コマンドと同じ機能を持ちます。ただし、各レコードをホールド状態にします。読み込んでホールドしようとしているレコードが現在他のユーザーによってホールドされている場合、レコードが使用可能になるか、または操作タイムアウトが発生するまで、読み込み操作は停止し、ユーザーは待機状態になります。コマンドオプション1 フィールドが "O" または "R" で、要求したレコードがすでにホールドされている場合、L6 コマンドによってレスポンスコード 145 が返されます。

- 特定の値への再位置決め
- 読み込み方向の変更

特定の値への再位置決め

順番制御ディスクリプタの開始値を指定する方法については、「[ACB インターフェイスダイレクトコール：L3 コマンドおよびL6 コマンド](#)」または「[ACBX インターフェイスダイレクトコール：L3 コマンドおよびL6 コマンド](#)」の「ISN：開始ISN／レコード読み込み ISN オプション」で ISN コントロールブロックフィールドの説明を参照してください。順次読み込み中に特定の値に再位置決めを行うには、次の手順に従います。

1. アディション1 フィールド（順次制御ディスクリプタ）の最後6桁を空白にします。
2. 読み込み順の再位置決め先を示す値をバリュースタックに入れます。
3. ISN フィールドを0にします。
4. コマンドオプション2 フィールドに "A"、"D"、または "V" を設定します。

読み込み方向の変更

再位置決めを行わなくても、コマンドオプション2で "A" または "D" を指定することによって、いつでも論理読み込み方向を昇順から降順に変更できます。

プリフェッチまたはマルチフェッチ処理も要求した場合、このオプションはサポートされません。

ACB インターフェイスダイレクトコール：L3 コマンドおよびL6 コマンド

このsectionでは、L3 コマンドおよびL6 コマンドの ACB インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの概要](#)
- [コントロールブロックフィールドの説明](#)

■ ACB の例

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	バイナリ	--	--
コマンドコード	3~4	英数字	F	U
コマンド ID	5~8	英数字	F	U
ファイル番号	9~10	バイナリ	F	U
レスポンスコード	11~12	バイナリ	--	A
ISN	13~16	バイナリ	F	A
ISN 下限	17~20	バイナリ	F	U
	21~24	--	--	--
フォーマットバッファ長	25~26	バイナリ	F	U
レコードバッファ長	27~28	バイナリ	F	U
サーチバッファ長	29-30	バイナリ	F *	U
バリュースearchバッファ長	31-32	バイナリ	F *	U
ISN バッファ長 **	33~34	バイナリ	F	U
コマンドオプション 1	35	英数字	F	U
コマンドオプション 2	36	英数字	F	U
アディクション 1	37~44	英数字	F	A
アディクション 2	45~48	バイナリ/バイナリ	--	A
アディクション 3	49~56	英数字	F	A
アディクション 4	57~64	英数字	F	A
アディクション 5	65~72	英数字	F	U
コマンドタイム	73~76	バイナリ	--	A
ユーザーエリア	77~80	--	--	U

バッファエリア

バッファ	Adabas コール前	Adabas コール後
フォーマット	F ***	U
レコード	--	A
サーチ *	F	U
バリュウ *	F	U
ISN **	F	U

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

* VALUE START オプション使用時のみ必要

** ISN バッファおよび長さは、マルチフェッチオプションまたはプリフェッチオプションを指定した場合にのみ必須です。

*** 圧縮オプション制御文字 "C" を含んでもかまいません。

-- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

コマンドコード (ACBCMD)

L3 または L6

コマンド ID (ACBCID)

このフィールドには、空白およびゼロを除く値を指定する必要があります。この値は、Adabas が正しい順序でレコードを返し、フォーマットバッファを繰り返し解釈するのを避けるために使用されます。

このフィールドの第 1 バイトに 16 進の FF を指定することはできません。また、ファイルの順次読み込み中は、このフィールドを変更しないでください。

コマンド ID 値が X'FFFFFFFF' の場合、コマンド ID は自動的に生成されます。Adabas ニュークリアスは、X'00000001' から始まるコマンド ID 値を生成し、L3 または L6 コールごとに値を 1 ずつ増やします。

ユーザー定義コマンド ID を指定する場合、ユーザーは各コマンド ID を固有のものにする必要があります。

別のフォーマット ID およびグローバルフォーマット ID の使用については、アディション 5 フィールドを参照してください。

ファイル番号 (ACBFNR)

読み込むファイルの番号を2進数で指定します。物理ダイレクトコールの場合は、次のようファイル番号を指定します。

- 1 バイトファイル番号の場合は、ファイル番号を第2バイト (10) に入力します。第1バイト (9) は、バイナリの0 (B'0000 0000') をセットします。
- 2 バイトファイル番号の場合は、2バイト (9 と 10) を使います。



Note: 2 バイトファイル番号およびデータベース ID を使用する場合は、コントロールブロックの先頭バイトに X'30' を入力しなければなりません。

レスポンスコード (ACBRSP)

Adabasはこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード0は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、アディクション2フィールドの下位2バイトにサブコードを伴う場合があります。詳細は『Adabas メッセージおよびコード』を参照してください。

レスポンスコード3は、エンドオブファイル条件が検出されたことを示します。

ISN：開始 ISN／レコード読み込み ISN オプション (ACBISN)

アディクション1フィールドの後ろ6バイトに空白を設定すると、論理読み込みの開始位置や後続位置を特定できます。これには、ISN フィールドに加えて、バリュースタックに指定した開始ディスクリプタ値を使用し、開始位置または再開位置を指定します。

アディクション1フィールドの後ろ6バイトが、最後に指定された READ コマンドと同じであり、現在の読み込み順の位置を変更しない場合は、ISN フィールドは無視されます。

Adabas は、読み込んだレコードの ISN を ISN フィールドに返します。

論理読み込み順の位置決めまたは再位置決めが行われるとき、開始ディスクリプタ値との比較演算子を次のように指定した場合は、ISN フィールドは使用されません。

- 昇順読み込みの場合に GT (より大きい)
- 降順読み込みの場合に LT (より小さい)

これ以外を指定した場合には、ISN フィールドが使用され、次のルールが適用されます。

昇順読み込みのルール (ASCENDING オプション)

バリュースタックで指定した開始ディスクリプタ値が論理読み込みに存在する場合は、次のように位置付けされます。

- ISN にゼロを指定した場合、開始ディスクリプタ値と一致する最初 (最小) の ISN。
- ISN にゼロ以外を指定した場合、開始ディスクリプタ値で指定した ISN よりも大きく、かつ一番近い ISN。存在しない場合は、指定した値の次に大きいディスクリプタ値の最初の ISN。

バリュースタックで指定した開始ディスクリプタ値が論理読み込みに存在しない場合は、指定したISNに関係なく、つねに次に大きいディスクリプタ値の最初のISNに位置付けられます。

降順読み込みのルール（DECENDING オプション）

バリュースタックで指定した開始ディスクリプタ値が論理読み込みに存在する場合は、次のように位置付けされます。

- ISN にゼロを指定した場合、開始ディスクリプタ値と一致する最後（最大）の ISN。
- ISN にゼロ以外を指定した場合、開始ディスクリプタ値で指定した ISN よりも小さく、かつ一番近い ISN。存在しない場合は、指定した値の次に小さいディスクリプタ値の最後の ISN。

バリュースタックで指定した開始ディスクリプタ値が論理読み込みに存在しない場合は、指定したISNに関係なく、つねに次に小さいディスクリプタ値の最後のISNに位置付けられます。

ISN 下限：マルチフェッチレコード数（ACBISL）

コマンドオプション1フィールドに "M" または "O"（マルチフェッチオプション）を指定した場合、このフィールドのゼロ以外の値によって、マルチフェッチする最大レコード数が特定されます。このフィールドの値がゼロの場合は、マルチフェッチ対象のレコード数はレコードバッファと ISN バッファの長さによって制限されます。詳細は、「[マルチフェッチ／プリフェッチ機能の使用](#)」を参照してください。

フォーマットバッファ長（ACBFBL）

フォーマットバッファ長（バイト単位）。ユーザープログラムで定義するフォーマットバッファエリアは、この指定長と同じか、それ以上でなければなりません。

レコードバッファ長（ACBRBL）

レコードバッファ長（バイト単位）。ユーザープログラムで定義するレコードバッファエリアは、この指定長と同じか、それ以上でなければなりません。

サーチバッファ長（ACBSBL）

サーチバッファ長（バイト単位）。ユーザープログラムで定義するサーチバッファエリアは、この指定長と同じか、それ以上でなければなりません。

バリュースタック長（ACBVBL）

バリュースタック長（バイト単位）。ユーザープログラムで定義するバリュースタックエリアは、この指定長と同じか、それ以上でなければなりません。

ISN バッファ長（ACBIBL）

ISN バッファ長（バイト単位）。

コマンドオプション1フィールドに "P" を指定した場合、L3 コマンドおよび L6 コマンドでは、ISN バッファを使用して、プリフェッチしたディスクリプタ値を格納します。ISN バッファは、最大ディスクリプタ値に、各値の先頭 16 バイトのヘッダーを加えた大きさ以上にする必要があります。ユーザープログラムで定義する実際の ISN バッファエリアは、この指定長と同じか、それ以上でなければなりません。

コマンドオプション 1 (ACBCOP1)

オプション	説明
M (マルチフェッチ)	マルチフェッチ処理がアクティブになります。
O (R オプションを指定してマルチフェッチ処理)	マルチフェッチ処理とオプション R (下記参照) 処理がアクティブになります。
P (プリフェッチ)	プリフェッチ処理がアクティブになります。
R (リターン)	L6 コマンドによる読み込みおよびホールド対象のレコードが使用可能でない場合に、レスポンスコード 145 を返します。

"M"、"O"、"P" のいずれかを指定すると、(コマンドレベルの) プリフェッチオプションまたはマルチフェッチオプションが使用されることになります。マルチフェッチ/プリフェッチオプションを使用すると、1つ以上のシーケンシャルレコードへの事前アクセスが可能になり、処理全体の時間が短縮され、単一レコードをフェッチする時間がなくなるため、パフォーマンスを向上させることができます。詳細は「[マルチフェッチ/プリフェッチ機能の使用](#)」を参照してください。

コマンドオプション 2 (ACBCOP2)

ディスクリプタエントリの処理順を指定します。

オプション	説明
A (昇順)	ディスクリプタエントリを昇順に処理します。開始値の指定 (オプション) を伴うこともあります。開始ディスクリプタ値はバリューストックに指定します。また、ISN フィールドに空値以外の値を設定すると、ディスクリプタ値が一致する複数のエントリ内での位置付けを行えます。サーチバッファまたはバリューストックを省略 (SBL または VBL をゼロ (0) に設定) すると、すべてのエントリを処理します。
D (降順)	ディスクリプタエントリを降順に処理します。開始値の指定 (オプション) を伴うこともあります。開始ディスクリプタ値はバリューストックに指定します。また、ISN フィールドに空値以外の値を設定すると、ディスクリプタ値が一致する複数のエントリ内での位置付けを行えます。サーチバッファまたはバリューストックを省略 (SBL または VBL をゼロ (0) に設定) すると、すべてのエントリを処理します。
V (開始値)	ディスクリプタの最小値 (またその次の値) からではなく、ユーザー指定値から昇順に順次読み込みを開始 (続行) します。このユーザー指定値はバリューストックに示し、また値の長さフォーマットはサーチバッファに指定する必要があります。ISN フィールドに空値以外の値を指定すると、読み込み操作をさらに限定できます。このオプションはバージョン 6 以前の Adabas で使用され、既存プログラムをサポートするために維持されています。
空白	開始値を指定せずに、昇順に順次読み込みを開始します。存在するすべての値が処理されます。サーチバッファとバリューストックは無視されます。このオプションはバージョン 6 以前の Adabas で使用され、既存プログラムをサポートするために維持されています。

アディクション1：順次制御ディスクリプタ（ACBADD1）

このフィールドには読み込み順序の制御に使用するディスクリプタを指定する必要があります。ディスクリプタ、サブディスクリプタ、またはスーパーディスクリプタを指定できます。


- フォネティックディスクリプタ、ピリオディックグループ内のディスクリプタ、およびピリオディックグループ内のフィールドから生成されたディスクリプタは指定できません。
- マルチプルバリュースフィールドのディスクリプタは指定できます。その場合、ファイルの順次読み込み中に同一レコードが何回も（同一レコード内で異なる値ごとに1回）読み込まれる可能性があります。
- 指定のディスクリプタが空値省略（NU）オプションで定義されている場合は、ディスクリプタが空値のレコードは読み込まれません。
- ディスクリプタ名は、このフィールドの先頭2桁に指定します。初回コール時は、それ以外の部分を空白にする必要があります。

アディクション1フィールドは、L3またはL6の各コール間で変更しないでください。ただし、順次読み込み中に、特定の値に位置を変更することはできます。これを行うには、このフィールドの後ろ6桁を空白にする方法、位置を示す値をバリュースバッファに追加する方法、ISNフィールドをゼロにする方法、およびコマンドオプション2フィールドにA、D、またはVを設定する方法があります。

アディクション2（圧縮および非圧縮レコード長（ACBADD2））

コマンドが正常に処理されると、次の情報がこのフィールドに返されます。

- 少なくとも1つの有効なフィールド値がレコードバッファにある場合、先頭2バイトには、アクセスしたレコードの圧縮レコード長がバイナリ形式で格納されます。
- 後ろ2バイトには、フォーマットバッファで選択し、アクセスしたフィールドの非圧縮長がバイナリ形式で格納されます。

 **Note:** プリフェッチ機能の使用時、この長さ情報は返されません。

L3コマンドまたはL6コマンドによってゼロ以外のレスポンスコードが返された場合、下位2バイトには、正確なレスポンスコードの意味を定義するサブコードが含まれていることがあります。レスポンスコードとサブコードについては、『*Adabas* メッセージおよびコード』を参照してください。

アディクション3：パスワード（ACBADD3）

このフィールドは、Adabas Security または ADAESI パスワードを指定するため使用します。データベース、ファイル、またはフィールドがセキュリティ保護されている場合は、有効なセキュリティパスワードまたは ADAESI パスワードをユーザーは指定する必要があります。パスワードを保護するために、コマンド処理時にアディクション3フィールドに空白がセットされます。

アディクション4：サイファコード（ACBADD4）

このフィールドは、サイファコードを指定するために使用します。ファイルを暗号化する場合は、有効なサイファコードを指定する必要があります。ファイルを暗号化しない場合は、このフィールドに空白をセットする必要があります。

Adabasは、コマンド処理時に、サイファコードに空白をセットし、バージョンコードとデータベースIDをフィールドの下位3バイトに返します。詳細については「[コントロールブロックフィールド](#)」を参照してください。

アディクション5：フォーマットID、グローバルフォーマットID（ACBADD5）

このコマンド用の内部フォーマットバッファを識別する別のフォーマットIDを表示したり、全ユーザーが内部フォーマットバッファを使用できるようにグローバルフォーマットIDを表示したりします。

アディクション5フィールドの左端のビットに0が指定されていると、コマンドIDフィールドに指定された値がフォーマットIDとして使用されます。

このビットに1がセットされていると、アディクション5フィールドの5バイト目から8バイト目までがフォーマットIDとして使用されます。

さらに、アディクション5フィールドの第1バイトの最上位（左端）2ビットに、1（B'11'）がセットされると、アディクション5フィールドの全8バイトが、グローバルフォーマットIDとして使用されます（つまり、複数ユーザーがこのフォーマットIDを同時に使用できます）。

詳細と例については、「[コマンドID、フォーマットID、グローバルフォーマットID](#)」を参照してください。

ACBの例

- [例 1](#)
- [例 2：論理順に読み込んでレコードをホールドする](#)
- [例 3：ASCENDING オプション](#)
- [例 4：再位置決めを伴う DESCENDING オプション](#)

例 1

ファイル2を論理順に読み込みます。ディスクリプタRBを順序制御に使用します。フィールドRAおよびRBの値を返します。ファイル全体を読み込みます。

コントロールブロック

コマンドコード	L3	
コマンドID	EX01	空白でないコマンドIDを要求
ファイル番号	2	
ISN	0	全レコードを読み込みます
フォーマットバッファ長	6	またはそれ以上
レコードバッファ長	18	またはそれ以上
サーチバッファ長	0	
バリューストックバッファ長	0	

コマンドオプション1	b	レスポンスコード 145 またはプリフェッチオプションは使用しません
コマンドオプション2	A	昇順（サーチバッファとバリュースタックは指定しません）
アディショナル1	RBbbbbbb	順序制御に使用するディスクリプタ RB。bbbbbb には空白が入ります。
アディショナル3	パスワード	ファイルはセキュリティ保護されています
アディショナル4	bbbbbb (空白)	ファイルは暗号化されない

バッファエリア

フォーマットバッファ	RA,RB.
------------	--------

L3 コールを繰り返して各レコードを取得します。CID とアディショナル1 フィールドは L3 コールの間に変更しないでください。全レコードの読み込みが終わると、レスポンスコード 3 が返されます。

例 2：論理順に読み込んでレコードをホールドする

ファイル1を論理順に読み込みます。ディスクリプタ AA を順序制御として使用します。フィールド AA と AB の値が返されます。読み込んだ各レコードはホールド状態になります。

コントロールブロック

コマンドコード	L6	
コマンド ID	EX02	空白でないコマンド ID を要求
ファイル番号	1	
ISN	0	全レコードを読み込みます。
フォーマットバッファ長	3	またはそれ以上
レコードバッファ長	10	またはそれ以上
サーチバッファ長	0	
バリュースタック長	0	
コマンドオプション1	b	レスポンスコード 145 またはプリフェッチオプションは使用しません
コマンドオプション2	b	昇順（サーチバッファとバリュースタックは指定しません）
アディショナル1	AAbbbbbb	順序制御に使用するディスクリプタ AA。bbbbbb には空白が入ります。
アディショナル3	bbbbbb (空白)	ファイルはセキュリティ保護されていません
アディショナル4	bbbbbb (空白)	ファイルは暗号化されない

バッファエリア

フォーマットバッファ	GA.
------------	-----

論理トランザクションを終了し、ホールドしたレコードを解放するため、ET ロジックユーザーは ET コマンドを発行します。非 ET ユーザーは、A4、E4、または RI コマンドを発行して、ホールドしたレコードを解放します。

例 3：ASCENDING オプション

値 N から読み込みを始めること以外は、例 1 と同じです。

コントロールブロック

コマンドコード	L3	
コマンド ID	EX03	空白でないコマンド ID を要求
ファイル番号	2	
ISN	0	全レコードを読み込みます
フォーマットバッファ長	6	またはそれ以上
レコードバッファ長	18	またはそれ以上
サーチバッファ長	7	またはそれ以上
バリューバッファ長	1	またはそれ以上
コマンドオプション 1	b	レスポンスコード 145 オプション使用しません
コマンドオプション 2	A	開始値が指定された ASCENDING オプション
アディション 1	RBbbbbbb	RB を順序制御に使用します。bbbbbb には空白が入ります。
アディション 3	パスワード	ファイルはセキュリティ保護されています
アディション 4	bbbbbbb (空白)	ファイルは暗号化されない

バッファエリア

フォーマットバッファ	RA,RB.	
サーチバッファ	RB,1,A.	
バリューバッファ	N	値 N で読み込みを開始します。

最初の L3 コマンドでディスクリプタ RB に値 N を持つ最初のレコードが返されます。この値を持つレコードが存在しない場合は、N より大きな値 (NA など) を持つ最初のレコードが返されます。

例 4：再位置決めを伴う DESCENDING オプション

再位置決めすること以外は、例 3 の要求と同じです。順次読み込み処理は値 Q から続行されま
す。

コントロールブロック

コマンドコード	L3	
コマンド ID	EX03	再位置決めを行うときは、CID フィールドを変更しないでください。
ファイル番号	2	
ISN	0	バリューによる再位置決めを行うときは、このフィールドをゼロに設定する必要があります。
フォーマットバッファ長	6	またはそれ以上
レコードバッファ長	18	またはそれ以上
サーチバッファ長	7	またはそれ以上
バリューバッファ長	1	またはそれ以上
コマンドオプション 1	b	レスポンスコード 145 オプション使用しません
コマンドオプション 2	D	再位置決めを伴う DESCENDING オプション
アディクション 1	RBbbbbbb	このフィールドの最後の 6 桁には、再位置決め用に空白を設定しておく必要があります。
アディクション 3	パスワード	ファイル 2 はセキュリティ保護されています。
アディクション 4	bbbbbbbb (空白)	ファイル 2 は暗号化されていません

バッファエリア

フォーマットバッファ	RA,RB.	
サーチバッファ	RB,1,A,LE.	
バリューバッファ	Q	値 Q に再位置決めします。

ACBX インターフェイスダイレクトコール：L3 コマンドおよび L6 コマンド

この section では、L3 コマンドおよび L6 コマンドの ACBX インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- コントロールブロックとバッファの概要

■ コントロールブロックフィールドの説明

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1～2	---	---	---
バージョンインジケータ	3～4	バイナリ	F	U
	5～6	---	---	---
コマンドコード	7～8	英数字	F	U
	9～10	---	---	---
レスポンスコード	11～12	バイナリ	---	A
コマンド ID	13～16	英数字／バイナリ	F	U
データベース ID	17～20	数値	F	U
ファイル番号	21～24	数値	F	U
	25～28	---	---	---
ISN	29～32	バイナリ	F	A
	33～36	---	---	---
ISN 下限	37～40	バイナリ	F	U
	41～48	---	---	---
コマンドオプション 1	49	英数字	F	U
コマンドオプション 2	50	英数字	F	U
	51～56	---	---	---
アディション 1	57～64	英数字／バイナリ	F	A
アディション 2	65～68	バイナリ	---	A
アディション 3	69～76	英数字／バイナリ	F	A
アディション 4	77～84	英数字	F	A
アディション 5	85～92	英数字／バイナリ	F	U
	93～114	---	---	---
エラーサブコード	115～116	バイナリ	---	A
	117～144	---	---	---
コマンドタイム	145～152	バイナリ	---	A
ユーザーエリア	153～168	該当なし	---	U
---	169～193	操作不可	---	---

ABD とバッファ

ABD とバッファ	Adabas コール前	Adabas コール後
フォーマット	F ***	U
レコード	--	A
サーチ *	F	U
バリュー *	F	U
マルチフェッチ **	F	U

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

* VALUE START オプション使用時のみ必要

** マルチフェッチオプションを指定した場合のみ、マルチフェッチバッファが必要です。

*** 圧縮オプション制御文字 "C" を含んでもかまいません。

--- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

バージョンインジケータ (ACBXVER)

F2

コマンドコード (ACBXCMD)

L3 または L6

レスポンスコード (ACBXRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、エラーサブコード (ACBXERRC) フィールドにサブコードを伴う場合があります。詳細は、*Adabas* メッセージおよびコードのドキュメントを参照してください。

レスポンスコード 3 は、エンドオブファイル条件が検出されたことを示します。

コマンド ID (ACBXCID)

このフィールドには、空白およびゼロを除く値を指定する必要があります。この値は、Adabas が正しい順序でレコードを返し、フォーマットバッファを繰り返し解釈するのを避けるために使用されます。

このフィールドの第1バイトに16進のFFを指定することはできません。また、ファイルの順次読み込み中は、このフィールドを変更しないでください。

コマンドID値がX'FFFFFFFF'の場合、コマンドIDは自動的に生成されます。Adabasニュークリアスは、X'00000001'から始まるコマンドID値を生成し、L3またはL6コールごとに値を1ずつ増やします。

ユーザー定義コマンドIDを指定する場合、ユーザーは各コマンドIDを固有のものにする必要があります。

別のフォーマットIDおよびグローバルフォーマットIDの使用については、アディクション5フィールドを参照してください。

データベースID (ACBXDBID)

このフィールドを使用して、データベースIDを指定します。Adabasコールはこのデータベースに送られます。

このフィールドは4バイトのバイナリフィールドですが、現時点では2バイトのデータベースIDのみがサポートされています。したがって、データベースIDはフィールドの下位部分(右端のバイト)に指定し、その前にバイナリの0を付加する必要があります。

このフィールドがバイナリの0に設定されている場合は、Adabas APIはDDCARD入力データで指定されたADARUNカードのデータベースIDか、リンクされているカリンクルーチンによりロードされたLNKGBLSモジュールで指定されているデフォルトのデータベースID値のいずれかを使用します。

ファイル番号 (ACBXFNR)

このフィールドを使用して、Adabasコールで送る必要のあるファイル数を指定します。

このフィールドは4バイトのバイナリフィールドですが、ファイル番号はフィールドの下位部分(右端のバイト)に指定し、その前にバイナリの0を付加する必要があります。

ISN：開始ISN/レコード読み込みISNオプション (ACBXISN)

アディクション1フィールドの後ろ6バイトに空白を設定すると、論理読み込みの開始位置や後続位置を特定できます。これには、ISNフィールドに加えて、バリュースタックに指定した開始ディスクリプタ値を使用し、開始位置または再開位置を指定します。

アディクション1フィールドの後ろ6バイトが、最後に指定されたREADコマンドと同じであり、現在の読み込み順の位置を変更しない場合は、ISNフィールドは無視されます。

ACBXISNフィールドは、8バイトのACBXISNGフィールドに埋め込まれた4バイトのバイナリフィールドで、まだ使用されていません。ACBXISNGフィールドの上位部分には、バイナリの0を設定します。

Adabasは、読み込んだレコードのISNをISNフィールドに返します。

論理読み込み順の位置決めまたは再位置決めが行われるとき、開始ディスクリプタ値との比較演算子を次のように指定した場合は、ISNフィールドは使用されません。

- 昇順読み込みの場合にGT(より大きい)

■ 降順読み込みの場合に LT（より小さい）

これ以外を指定した場合には、ISN フィールドが使用され、次のルールが適用されます。

昇順読み込みのルール（ASCENDING オプション）

バリュースタックで指定した開始ディスクリプタ値が論理読み込みに存在する場合は、次のように位置付けされます。

- ISN にゼロを指定した場合、開始ディスクリプタ値と一致する最初（最小）の ISN。
- ISN にゼロ以外を指定した場合、開始ディスクリプタ値で指定した ISN よりも大きく、かつ一番近い ISN。存在しない場合は、指定した値の次に大きいディスクリプタ値の最初の ISN。

バリュースタックで指定した開始ディスクリプタ値が論理読み込みに存在しない場合は、指定した ISN に関係なく、つねに次に大きいディスクリプタ値の最初の ISN に位置付けられます。

降順読み込みのルール（DECENDING オプション）

バリュースタックで指定した開始ディスクリプタ値が論理読み込みに存在する場合は、次のように位置付けされます。

- ISN にゼロを指定した場合、開始ディスクリプタ値と一致する最後（最大）の ISN。
- ISN にゼロ以外を指定した場合、開始ディスクリプタ値で指定した ISN よりも小さく、かつ一番近い ISN。存在しない場合は、指定した値の次に小さいディスクリプタ値の最後の ISN。

バリュースタックで指定した開始ディスクリプタ値が論理読み込みに存在しない場合は、指定した ISN に関係なく、つねに次に小さいディスクリプタ値の最後の ISN に位置付けられます。

ISN 下限：マルチフェッチレコード数（ACBXISL）

コマンドオプション 1 フィールドに "M" または "O"（マルチフェッチオプション）を指定した場合、このフィールドのゼロ以外の値によって、マルチフェッチする最大レコード数が特定されます。この値がゼロであれば、マルチフェッチするレコード数はレコードバッファとマルチフェッチバッファの長さによって制限されます。詳細は、「[マルチフェッチ／プリフェッチ機能の使用](#)」を参照してください。

ACBXISL フィールドは、8 バイトの ACBXISLG フィールドに埋め込まれた 4 バイトのバイナリフィールドで、まだ使用されていません。ACBXISLG フィールドの上位部分には、バイナリの 0 を設定します。

コマンドオプション 1 (ACBXCOP1)

オプション	説明
M (マルチフェッチ)	マルチフェッチ処理がアクティブになります。
O (R オプションを指定してマルチフェッチ処理)	マルチフェッチ処理とオプション R (下記参照) 処理がアクティブになります。
R (リターン)	L6 コマンドによる読み込みおよびホールド対象のレコードが使用可能でない場合に、レスポンスコード 145 を返します。

M または O オプションを指定すると、(コマンドレベルの) マルチフェッチオプションが使用されます。マルチフェッチオプションを使用すると、1 つ以上のシーケンシャルレコードへの事前アクセスが可能になり、処理全体の時間が短縮され、単一レコードをフェッチする時間が必要なくなるため、パフォーマンスを向上させることができます。詳細は「[マルチフェッチ/プリフェッチ機能の使用](#)」を参照してください。

コマンドオプション 2 (ACBXCOP2)

ディスクリプタエントリの処理順を指定します。

オプション	説明
A (昇順)	ディスクリプタエントリを昇順に処理します。開始値の指定 (オプション) を伴うこともあります。開始ディスクリプタ値はバリュースタックに指定します。また、ISN フィールドに空値以外の値を設定すると、ディスクリプタ値が一致する複数のエントリ内での位置付けを行えます。サーチバッファまたはバリュースタックを省略 (SBL または VBL をゼロ (0) に設定) すると、すべてのエントリを処理します。
D (降順)	ディスクリプタエントリを降順に処理します。開始値の指定 (オプション) を伴うこともあります。開始ディスクリプタ値はバリュースタックに指定します。また、ISN フィールドに空値以外の値を設定すると、ディスクリプタ値が一致する複数のエントリ内での位置付けを行えます。サーチバッファまたはバリュースタックを省略 (SBL または VBL をゼロ (0) に設定) すると、すべてのエントリを処理します。
V (開始値)	ディスクリプタの最小値 (またその次の値) からではなく、ユーザー指定値から昇順に順次読み込みを開始 (続行) します。このユーザー指定値はバリュースタックに示し、また値の長さフォーマットはサーチバッファに指定する必要があります。ISN フィールドに空値以外の値を指定すると、読み込み操作をさらに限定できます。このオプションはバージョン 6 以前の Adabas で使用され、既存プログラムをサポートするために維持されています。
空白	開始値を指定せずに、昇順に順次読み込みを開始します。存在するすべての値が処理されます。サーチバッファとバリュースタックは無視されます。このオプションはバージョン 6 以前の Adabas で使用され、既存プログラムをサポートするために維持されています。

アディショナル 1: 順次制御ディスクリプタ (ACBXADD1)

このフィールドには読み込み順序の制御に使用するディスクリプタを指定する必要があります。ディスクリプタ、サブディスクリプタ、またはスーパーディスクリプタを指定できません。


- フォネティックディスクリプタ、ピリオディックグループ内のディスクリプタ、およびピリオディックグループ内のフィールドから生成されたディスクリプタは指定できません。
- マルチプルバリューフィールドのディスクリプタは指定できます。その場合、ファイルの順次読み込み中に同一レコードが何回も（同一レコード内で異なる値ごとに1回）読み込まれる可能性があります。
- 指定のディスクリプタが空値省略（NU）オプションで定義されている場合は、ディスクリプタが空値のレコードは読み込まれません。
- ディスクリプタ名は、このフィールドの先頭2桁に指定します。初回コール時は、それ以外の部分を空白にする必要があります。

アディクション1フィールドは、L3またはL6の各コール間で変更しないでください。ただし、順次読み込み中に、特定の値に位置を変更することはできます。これを行うには、このフィールドの後ろ6桁を空白にする方法、位置を示す値をバリュースタックに追加する方法、ISNフィールドをゼロにする方法、およびコマンドオプション2フィールドにA、D、またはVを設定する方法があります。

アディクション2：圧縮および非圧縮レコード長（ACBXADD2）

コマンドが正常に処理されると、次の情報がこのフィールドに返されます。

- 少なくとも1つの有効なフィールド値がレコードバッファにある場合、先頭2バイトには、アクセスしたレコードの圧縮レコード長がバイナリ形式で格納されます。
- 後ろ2バイトには、フォーマットバッファで選択し、アクセスしたフィールドの非圧縮長がバイナリ形式で格納されます。

 **Note:** プリフェッチ機能の使用時、この長さ情報は返されません。

アディクション3：パスワード（ACBXADD3）

このフィールドは、Adabas SecurityまたはADAESIパスワードを指定するため使用します。データベース、ファイル、またはフィールドがセキュリティ保護されている場合は、有効なセキュリティパスワードまたはADAESIパスワードをユーザーは指定する必要があります。パスワードを保護するために、コマンド処理時にアディクション3フィールドに空白がセットされます。

アディクション4：サイファコード（ACBXADD4）

このフィールドは、サイファコードを指定するために使用します。ファイルを暗号化する場合は、有効なサイファコードを指定する必要があります。ファイルを暗号化しない場合は、このフィールドに空白をセットする必要があります。

Adabasは、コマンド処理時に、サイファコードに空白をセットし、バージョンコードとデータベースIDをフィールドの下位3バイトに返します。詳細については「[コントロールブロックフィールド](#)」を参照してください。

アディクション5：フォーマットID、グローバルフォーマットID（ACBXADD5）

このコマンド用の内部フォーマットバッファを識別する別のフォーマットIDを表示したり、全ユーザーが内部フォーマットバッファを使用できるようにグローバルフォーマットIDを表示したりします。

アディション5フィールドの左端のビットに0が指定されていると、コマンドIDフィールドに指定された値がフォーマットIDとして使用されます。

このビットに1がセットされていると、アディション5フィールドの5バイト目から8バイト目までがフォーマットIDとして使用されます。

さらに、アディション5フィールドの第1バイトの最上位（左端）2ビットに、1（B'11'）がセットされると、アディション5フィールドの全8バイトが、グローバルフォーマットIDとして使用されます（つまり、複数ユーザーがこのフォーマットIDを同時に使用できます）。

詳細と例については、「[コマンドID](#)、[フォーマットID](#)、[グローバルフォーマットID](#)」セクションを参照してください。

エラーサブコード（ACBXERRC）

コマンドがゼロ以外のレスポンスコードを返したときは、このフィールドにレスポンスコードの正確な意味を定義したサブコードが含まれます。レスポンスコードとサブコードについては、『[Adabas メッセージおよびコード](#)』を参照してください。

バッファ

L3 コマンドおよび L6 コマンドには、次のバッファが適用されます。

- [フォーマットバッファ](#)
- [レコードバッファ](#)
- [サーチバッファ](#)
- [バリューストックバッファ](#)
- [サーチバッファとバリューストックバッファの例](#)
- [ISN バッファ](#)
- [マルチフェッチバッファ](#)

フォーマットバッファ


このバッファには、値を取得するフィールドを指定する必要があります。フォーマットバッファの構文および例については、「[バッファの定義](#)」を参照してください。

先頭2桁にC.を指定すると、COMPRESSED オプションが有効になります。これにより、レコードは、圧縮解除された形式ではなく圧縮された形式で返されます。

レコードバッファ

Adabas は、バッファに要求されたフィールドの値をこのバッファに返します。フィールド値は、フォーマットバッファに別のフォーマットや長さをユーザーが指定していない限り、標準フォーマットと標準長で返されます。

サーチバッファ

 **Note:** サーチバッファでは要素をコンマで区切り、ピリオドで終了します。サーチバッファの構文については別途説明します。

サーチバッファを使用するときは、コントロールブロックにサーチバッファ長 (SBL) を指定する必要があります。サーチバッファを使用しない場合は、SBL を 0 に設定する必要があります。

VALUE START (V) オプションをコマンドオプション 2 に使用する場合、順序制御に使用するディスクリプタの開始値をバリュースearchバッファに指定し、値の長さ形式をサーチバッファに指定する必要があります。

ASCENDING オプションまたは DESCENDING オプションをコマンドオプション 2 に使用する場合、サーチバッファを使用できますが、必ずしも使用する必要はありません。サーチバッファかバリュースearchバッファのどちらかを省略すると、所定のディスクリプタのすべての値が処理されます。バリュースearchバッファに開始値、終了値またはその両方を指定する場合は、検索されるディスクリプタエントリ数を制限するためにサーチバッファを使用する必要があります。

バリュースearchバッファに指定したディスクリプタ値の長さおよびフォーマットが、指定ディスクリプタの標準長およびフォーマットと異なる場合には、これらをサーチバッファに指定しなければなりません。

バリュースearchバッファに 1 つの値 (つまり、オプション A 付きの開始値またはオプション D 付きの終了値) を指定するとき、サーチバッファの形式は次のとおりです。

```
name [ ,length ] [ ,format ] [ ,comparator ]
```

バリュースearchバッファに 2 つの値 (つまり、開始値と終了値) を指定するとき、サーチバッファの形式は次のとおりです。

```
name [ ,length ] [ ,format ] ,S ,name [ ,length ] [ ,format ]
```

サーチバッファの構文ステートメントで使用されるエレメントを以下に示します。

name	順序制御に使用するディスクリプタの名前。指定する名前は、アディション1フィールドに指定した名前と同じである必要があります。	
length	バリュースタックに指定した値の長さ。長さを指定しない場合、ディスクリプタの標準長に応じて値が指定されます。指定可能な長さについては「 長さおよびデータフォーマット 」セクションを参照してください。	
format	バリュースタックに指定した値のフォーマット。フォーマットを指定しない場合、ディスクリプタの標準フォーマットに応じて値が指定されます。指定可能なフォーマット設定については「 長さおよびデータフォーマット 」セクションを参照してください。	
comparator	読み込み範囲の指定。	
	GE	値範囲の中の最大値よりも大きいか等しい（デフォルト）。
	GT	値範囲の中の最大値よりも大きい。
	LE	値範囲の中の最小値よりも小さいか等しい。
S	LT	値範囲の中の最小値よりも小さい。
	2つの検索式を使って開始-終了の範囲を示します。2つの式に同じディスクリプタを指定する必要があります。	
	AA,S,AA.	有効
AA,S,AB.	無効	

コマンドオプション2にAまたはDを選択した場合やサーチバッファとバリュースタックのオプションを選択した場合の結果については、「[例3 制御オプションの組み合わせ](#)」を参照してください。

バリュースタック

バリュースタック長（VBL）は、コントロールブロックに設定する必要があります。バリュースタックを使用しない場合は、VBLを0に設定する必要があります。

VALUE START オプションを使用する場合、またはバリューによる再位置決めを行う場合、読み込みを開始する（または継続する）値をこのバッファに指定する必要があります。

コマンドオプション2でASCENDING オプションまたはDESCENDING オプションを使用する場合、バリュースタックを使用できますが、必ずしも使用する必要はありません。サーチバッファかバリュースタックのどちらかを省略すると、所定のディスクリプタのすべての値が処理されます。開始値、終了値、あるいはその両方をバリュースタックに指定する場合、検索するディスクリプタエントリ数を制限するためにサーチバッファが必要です。

コマンドオプション2でASCENDING（A）オプションを指定すると、指定値を持つ最初のレコードから読み込みが開始（または継続）されます。この値を持つレコードがない場合は、次に大きい値を持つ最初のレコードから処理が行われます。

バリュースタックに2つの値が指定されている場合、1つ目の値は範囲の下限值、2つ目の値は上限値を示します。各値が開始値であるか終了値であるかは、コマンドオプション2での指定がAかDかによって決まります。

また、ISNフィールドがゼロでない場合には、このISNより大きいISNを持つレコードについて、指定値に該当する最初のレコードから読み込みが開始（または継続）されるか、またはそのレコードで終了します。該当するISNがない場合は、次に大きい（または、小さい）値を持つ最初のレコードが読み込まれます。

サーチバッファとバリュースタックの例

- 例1：サーチバッファおよびバリュースタックを使用した ASCENDING オプション
- 例2：サーチバッファおよびバリュースタックを使用した DESCENDING オプション
- 例3：制御オプションの組み合わせ

例1：サーチバッファおよびバリュースタックを使用した ASCENDING オプション

順序制御に使用するディスクリプタのインバーテッドリスト

値	ISN リスト
A	1, 4
B	2
D	3, 5

最初の L3 コマンドまたは L6 コマンドはコマンドオプション2に A（ASCENDING オプション）を指定、後続の L3 コマンドまたは L6 コマンドはコマンドオプション2に A を指定し、アディショナル1の最後6バイトを空白にリセットします。

ユーザー指定値	ユーザー指定 ISN	読み込み開始（または継続）の ISN
A	0	1
A	1	4
A	2	4
A	4	2
A	5	2
B	0	2
B	1	2
B	2	3
B	3	3
BABC	1	3
C	0	3
D	0	3

ユーザー指定値	ユーザー指定 ISN	読み込み開始（または継続）の ISN
D	3	5
D	4	5
D	5	レスポンスコード 3
E-Z	-	レスポンスコード 3

例 2：サーチバッファおよびバリュースタックを使用した DESCENDING オプション

順序制御に使用するディスクリプタのインバーテッドリスト

値	ISN リスト
A	1, 9, 25
B	3, 18, 21
C	7, 8, 11

最初の L3 コマンドまたは L6 コマンドはコマンドオプション 2 に D（DESCENDING オプション）、開始値に ISN=0、サーチバッファの比較演算子に LT、バリュースタックに C を指定します。開始位置は、次に低いディスクリプタ値の最大 ISN になります。この例では、ディスクリプタ B の ISN 21 です。

例 3：制御オプションの組み合わせ

下表は、ASCENDING/DESCENDING オプションと、サーチバッファおよびバリュースタックの各設定の組み合わせ例を示しています。これらの例は、ファイルが読み込まれており、かつ ISN=0（一致する開始値に位置付けていない）の場合に適用されます。

2. L3 または L6 コマンドを使用して読み込んでいるファイルに対して、レコードの更新と削除を行うことができます。挿入したレコード（複数可）を読み込む場合は、レコード挿入後に再位置付けを行う必要があります。
3. L3 コマンドまたは L6 コマンドを使用して読み込んでいるファイルを他のユーザーが更新している場合は、L3 コマンドまたは L6 コマンドでそのファイルを読み込んでも、レコードが返されないことがあります。また、ファイルの順次読み込み中に同じレコードが複数回返されることもあります。



Note: 現在の値よりも大きい値で読み込み順序を制御するディスクリプタフィールドを更新しないでください。

35 L9 コマンド：ディスクリプタ値の読み込み

■ 機能および使用	334
■ ACB インターフェイスダイレクトコール：L9 コマンド	334
■ ACBX インターフェイスダイレクトコール：L9 コマンド	342
■ バッファ	348
■ その他の考慮事項	352

L9 コマンドは、指定のディスクリプタの値を読み込みます。

このchapterでは、次のトピックについて説明します。

機能および使用

L9 コマンドは、ある範囲にあるディスクリプタの値とその値を持つレコード数を求めるために使用します。

ディスクリプタを持つファイル、値を取得するディスクリプタ、および処理の開始値を指定します。各L9コールは、ディスクリプタの次の値を**レコードバッファ**に返し、その値を持つレコード数をISN 数フィールドに返します。値は正である場合と負である場合があります。空値省略 (NU) オプションで定義されたディスクリプタの場合、空値は返されません。

マルチフェッチ/プリフェッチ オプションを使用すると、一度に複数のディスクリプタ値が読み込まれるため、パフォーマンスが向上します。マルチフェッチを使用可能にするには、コマンドオプション1フィールドに "M" を指定します。詳細は、「[マルチフェッチ/プリフェッチ機能の使用](#)」を参照してください。

コマンドオプション2フィールドに "T" を指定すると、各値のISNがレコードバッファに返されます。L9 コマンドは、アソシエータ内のインバーテッドリストだけを読み込みます。データストレージにはアクセスしません。

論理読み込み順を変更しなくても、コマンドオプション2で "A" または "D" を指定することによって、いつでも論理読み込み方向を昇順から降順に変更できます。コマンドオプション2に A または D を選択した場合やサーチバッファとバリューストレージのオプションを選択した場合の結果については、「[例3 制御オプションの組み合わせ](#)」を参照してください。昇順および降順オプションは、マルチフェッチ処理ではサポートされていません。

ACB インターフェイスダイレクトコール：L9 コマンド

このsectionでは、L9コマンドのACBインターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの概要](#)
- [コントロールブロックフィールドの説明](#)

■ ACB の例

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	--	--	--
コマンドコード	3~4	英数字	F	U
コマンド ID	5~8	英数字	F	U
ファイル番号	9~10	バイナリ	F	U
レスポンスコード	11~12	バイナリ	--	A
ISN	13~16	バイナリ	--	A
ISN 下限	17~20	バイナリ	F	A
ISN 数	21~24	バイナリ	--	A
フォーマットバッファ長	25~26	バイナリ	F	U
レコードバッファ長	27~28	バイナリ	F	U
サーチバッファ長	29-30	バイナリ	F	U
バリュースタック長	31-32	バイナリ	F	U
ISN バッファ長*	33~34	バイナリ	F	U
コマンドオプション 1	35	英数字	F	U
コマンドオプション 2	36	英数字	F	U
アディクション 1	37~44	英数字	F	U
アディクション 2	45~48	バイナリ	--	A
アディクション 3	49~56	英数字	F	A
	57~64	--	--	--
アディクション 5	65~72	英数字	F	U
コマンドタイム	73~76	バイナリ	--	A
ユーザーエリア	77~80	--	--	U

バッファエリア

バッファ	Adabas コール前	Adabas コール後
フォーマット	F	U
レコード	--	A
サーチ	F	U
バリュー	F	U
ISN *	--	A

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

* ISN バッファおよび長さは、マルチフェッチオプションまたはプリフェッチオプションを指定した場合にのみ必須です。

-- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

コマンドコード (ACBCMD)

L9

コマンド ID (ACBCID)

このフィールドには、空白およびゼロを除く値を指定する必要があります。この値は、Adabas が正しい順序で値を返し、フォーマットバッファを繰り返し解釈するのを避けるために使用されます。

このフィールドは、ファイルの順次読み込み中に変更しないでください。

このフィールドの第 1 バイトには 16 進数 'FF' にセットしないでください。

ファイル番号 (ACBFNR)

読み込むファイルの番号を 2 進数で指定します。物理ダイレクトコールの場合は、次のようファイル番号を指定します。

- 1 バイトファイル番号の場合は、ファイル番号を第 2 バイト (10) に入力します。第 1 バイト (9) は、バイナリの 0 (B'0000 0000') をセットします。
- 2 バイトファイル番号の場合は、2 バイト (9 と 10) を使います。



Note: 2 バイトファイル番号およびデータベース ID を使用する場合は、コントロールブロックの先頭バイトに X'30' を入力しなければなりません。

レスポンスコード (ACBRSP)

Adabasはこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード0は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、アディション2フィールドの下位2バイトにサブコードを伴う場合があります。詳細は『Adabas メッセージおよびコード』を参照してください。

レスポンスコード3は、エンドオブファイル条件が検出されたことを示します。

ISN：ピリオディックグループのオカレンス (ACBISN)

値を求めたいディスクリプタがピリオディックグループに属する場合、このフィールドには、求められた値が見つかったオカレンス番号が返されます。オカレンス番号は下位2バイトにバイナリ形式で返されます。

プリフェッチオプションを指定した場合、プリフェッチした値のオカレンスはISNバッファ値の先頭のヘッダーに返されます。

ISN 下限：レコードバッファで最小の ISN (ACBISL)

このフィールドに返される値は次のとおりです。

コマンドオプション1	コマンドオプション2	L9 コマンドの処理内容
空白	I	各値の ISN がレコードバッファに返されます。ISN 下限フィールドには、値は返されません。
	Iが設定されていない	返された ISN リストの先頭の ISN 番号が ISN 下限フィールドに入ります。
P (プリフェッチ)	I	先頭の ISN 番号がレコードバッファに返されます。プリフェッチされたすべてのディスクリプタ値は、先頭に16バイトのヘッダーが付いて ISN バッファに返されます。ISN 下限フィールドには値は返されません。
	Iが設定されていない	プリフェッチされた最後の値の先頭の ISN 番号が ISN 下限フィールドに入ります。
M (マルチフェッチ)	I	マルチフェッチされたレコードのグループがレコードバッファに返されます。このレコードの記述は呼び出し元の ISN バッファに返されます。ISN 下限フィールドには値は返されません。
	Iが設定されていない	マルチフェッチされた最後の値の先頭の ISN 番号が ISN 下限フィールドに入ります。

コマンドオプション1が"M" (マルチフェッチオプション) に設定されている場合、次の設定を行うことができます。

- ISN 下限フィールドにゼロ以外の値を設定すると、マルチフェッチする値の数を制限できます。
- すべての値をマルチフェッチする場合は、ISN 下限フィールドにゼロを指定します。

詳細は、「[マルチフェッチ／プリフェッチ機能の使用](#)」を参照してください。

ISN 数：レコード件数 (ACBISQ)

コマンドオプション 2 に RETURN ISN オプション (I) を指定した場合を除き、このフィールドにはレコードバッファに返された値を持つレコード数が返されます。

プリフェッチオプションを指定した場合、プリフェッチした値の数は ISN バッファ値の先頭のヘッダーに返されます。

フォーマットバッファ長 (ACBFBL)

フォーマットバッファ長 (バイト単位)。ユーザープログラムで定義するフォーマットバッファエリアは、この指定長と同じか、それ以上でなければなりません。

レコードバッファ長 (ACBRBL)

レコードバッファ長 (バイト単位)。ユーザープログラムで定義するレコードバッファエリアは、この指定長と同じか、それ以上でなければなりません。

サーチバッファ長 (ACBSBL)

サーチバッファ長 (バイト単位)。ユーザープログラムで定義するサーチバッファエリアは、この指定長と同じか、それ以上でなければなりません。

バリューバッファ長 (ACBVBL)

バリューバッファ長 (バイト単位)。ユーザープログラムで定義するバリューバッファエリアは、この指定長と同じか、それ以上でなければなりません。

ISN バッファ長：コマンドレベルのマルチフェッチ/プリフェッチオプションのみ (ACBIBL)
ISN バッファ長 (バイト単位)。

コマンドオプション 1 フィールドに "P" が設定されている場合は、L9 コマンドでは ISN バッファを使用して、プリフェッチしたディスクリプタ値を保持します。ISN バッファは、最大ディスクリプタ値に、各値の先頭 16 バイトのヘッダーを加えた大きさ以上にする必要があります。ユーザープログラムで定義する実際の ISN バッファエリアは、この指定長と同じか、それ以上でなければなりません。

コマンドオプション 1：コマンドレベルのマルチフェッチ/プリフェッチ オプション (ACBCOP1)

このうちのいずれかのオプションを指定すると、(コマンドレベルの) プリフェッチオプションまたはマルチフェッチオプションが使用されます。マルチフェッチ/プリフェッチオプションを使用すると、同時に複数のディスクリプタ値を読み込み、単レコードをフェッチする時間が必要なくなるため、パフォーマンスを向上させることができます。詳細は「[マルチフェッチ/プリフェッチ機能の使用](#)」を参照してください。

オプション	説明
M (マルチフェッチ)	<p>マルチフェッチ処理が有効です。</p> <p>このオプションと組み合わせて、次の設定を行うことができます。</p> <p>■ ISN 下限フィールドにゼロ以外の値を設定すると、マルチフェッチする値の数を制限できます。</p>

オプション	説明
	■ すべての値をマルチフェッチする場合は、ISN 下限フィールドにゼロを指定します。M オプションが指定された ISN バッファのデータフォーマットには、レコードディスクリプタの標準フォーマットが反映されます。
P (プリフェッチ)	プリフェッチ処理が有効です。

M または P オプションが指定されている場合、L9 コマンドでは、プリフェッチまたはマルチフェッチしたすべてのディスクリプタ値の先頭に 16 バイトのヘッダーを付け、ISN バッファに格納します。ISN バッファは、一番大きなディスクリプタ値に 16 バイトを加えた大きさを格納できる必要があります。ユーザープログラムの ISN バッファエリアは、指定された ISN バッファ長と等しいか、それ以上である必要があります。

コマンドオプション 2：RETURN ISN オプション (ACBCOP2)

オプション	説明
I	Adabas は、値自体の他、各値の ISN もレコードバッファに格納します。L9 コマンドは、アソシエータ内のインバーテッドリストだけを読み込みます。データストレージにはアクセスしません。
A	ディスクリプタのエントリを昇順に処理します。
D	ディスクリプタのエントリを降順に処理します。

アディクション 1：ディスクリプタ名 (ACBADD1)

サーチバッファ長およびバリューバッファ長の両方にゼロが指定されている場合、アディクション 1 フィールドの値は、値を求めるディスクリプタの名前になります。この名前は、フォーマットバッファで指定したディスクリプタ名と同じでなければなりません。

この場合、L9 は指定されたディスクリプタのすべての値をファイルの始めから処理します。

ディスクリプタ名は、このフィールドの先頭 2 桁に指定します。残りの桁は空白にしなければなりません。

アディクション 2：レスポンスサブコード (ACBADD2)

L9 コマンドによってゼロ以外のレスポンスコードが返された場合、このフィールドの下位 2 バイトにレスポンスコードの正確な意味を定義したサブコードが含まれていることがあります。レスポンスコードとサブコードについては、『Adabas メッセージおよびコード』を参照してください。

アディクション 3：パスワード (ACBADD3)

このフィールドは、Adabas Security または ADAESI パスワードを指定するため使用します。データベース、ファイル、またはフィールドがセキュリティ保護されている場合は、有効なセキュリティパスワードまたは ADAESI パスワードをユーザーは指定する必要があります。パスワードを保護するために、コマンド処理時にアディクション 3 フィールドに空白がセットされます。

アディクション5：フォーマット ID、グローバルフォーマット ID (ACBADD5)

このコマンド用の内部フォーマットバッファを識別する別のフォーマット ID を表示したり、全ユーザーが内部フォーマットバッファを使用できるようにグローバルフォーマット ID を表示したりします。

アディクション5 フィールドの左端のビットに 0 が指定されていると、コマンド ID フィールドに指定された値がフォーマット ID として使用されます。

このビットに 1 がセットされていると、アディクション5 フィールドの5 バイト目から 8 バイト目までがフォーマット ID として使用されます。

さらに、アディクション5 フィールドの第1バイトの最上位（左端）2 ビットに、1 (B'11') がセットされると、アディクション5 フィールドの全 8 バイトが、グローバルフォーマット ID として使用されます（つまり、複数ユーザーがこのフォーマット ID を同時に使用できます）。

詳細と例については、「[コマンド ID](#)、[フォーマット ID](#)、[グローバルフォーマット ID](#)」を参照してください。

ACB の例

- [例 1](#)
- [例 2](#)
- [例 3：制御オプションの組み合わせ](#)

例 1

ファイル 2 内のディスクリプタ RB の全値を返します。

コントロールブロック

コマンドコード	L9	
コマンド ID	L901	空白でないコマンド ID を要求
ファイル番号	2	
フォーマットバッファ長	3	またはそれ以上
レコードバッファ長	10	またはそれ以上
サーチバッファ長	5	またはそれ以上
バリューストックバッファ長	1	またはそれ以上
アディクション 3	パスワード	ファイル 2 はセキュリティ保護されています。

バッファエリア

フォーマットバッファ	RB.	値は標準の長さおよびフォーマットで返されます。
サーチバッファ	RB,1.	ディスクリプタ RB の値が返され、開始値は標準フォーマットと長さ 1 で指定されます。
バリューバッファ	b	処理は RB の最初の値が b またはそれ以上の場合に始まります。

各 L9 コールで次の値を受け取ります（値は昇順で与えられます）。この値を持つレコードの数は ISN 数フィールドに返されます。

例 2

ファイル 1 内のディスクリプタ AB の値を返します。20 以上の値だけを返すこととします。

コントロールブロック

コマンドコード	L9	
コマンド ID	L902	空白でないコマンド ID を要求
ファイル番号	1	
フォーマットバッファ長	7	またはそれ以上
レコードバッファ長	3	またはそれ以上
サーチバッファ長	7	またはそれ以上
バリューバッファ長	2	またはそれ以上
アデクション 3	bbbbbbbb (空白)	ファイル 1 はセキュリティ保護されていません

バッファエリア

フォーマットバッファ	AB,3,U.	値は長さ = 3、フォーマット = アンパックで返されます。
サーチバッファ	AB,2,U.	ディスクリプタ AB の値が返され、開始値は 2 バイトのアンパック数字で指定されます。
バリューバッファ	X'F2F0'	処理は AB の最初の値が 20 またはそれ以上の場合に始まります。

例 3：制御オプションの組み合わせ

下表は、ASCENDING/DESCENDING オプションと、サーチバッファおよびバリューバッファの各設定の組み合わせ例を示しています。これらの例は、ファイルが読み込まれている場合に適用されます。

コマンド コード	コマンド オプション2	サーチ バッファ	バリュー バッファ	最小 値	最大 値
L9	A	DE[, GE].	V1		V1
L9	D	DE[, GE].	V1	V1	
L9	A	DE, GT.	V1		V2
L9	D	DE, GT.	V1	V1	
L9	A	DE, LE.	V2		V2
L9	D	DE, LE.	V2	V2	
L9	A	DE, LT.	V2		V2
L9	D	DE, LT.	V2	V2	
L9	A	DE, S, DE.	V1V2		V2
L9	D	DE, S, DE.	V2V2	V2	
L9	A	なし	なし		V2
L9	D	なし	なし	V2	

順序オプションの概要

ACBX インターフェイスダイレクトコール：L9 コマンド

このsectionでは、L9 コマンドの ACBX インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- **コントロールブロックとバッファの概要**

■ コントロールブロックフィールドの説明

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	---	---	---
バージョンインジケータ	3~4	バイナリ	F	U
	5~6	---	---	---
コマンドコード	7~8	英数字	F	U
	9~10	---	---	---
レスポンスコード	11~12	バイナリ	---	A
コマンド ID	13~16	英数字/バイナリ	F	U
データベース ID	17~20	数値	F	U
ファイル番号	21~24	数値	F	U
	25~28	---	---	---
ISN	29~32	バイナリ	---	A
	33~36	---	---	---
ISN 下限	37~40	バイナリ	F	A
	41~44	---	---	---
ISN 数	45~48	バイナリ	---	A
コマンドオプション 1	49	英数字	F	U
コマンドオプション 2	50	英数字	F	U
	51~56	---	---	---
アディション 1	57~64	英数字/バイナリ	F	U
---	65~68	---	---	---
アディション 3	69~76	英数字/バイナリ	F	A
	77~84	---	---	---
アディション 5	85~92	英数字/バイナリ	F	U
	93~114	---	---	---
エラーサブコード	115~116	バイナリ	---	A
	117~144	---	---	---
コマンドタイム	145~152	バイナリ	---	A
ユーザーエリア	153~168	該当なし	---	U

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
---	169~193	操作不可	---	---

ABD とバッファ

ABD とバッファ	Adabas コール前	Adabas コール後
フォーマット	F	U
レコード	---	A
サーチ	F	U
バリュー	F	U
マルチフェッチ*	---	A

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

* マルチフェッチバッファは、マルチフェッチオプションを指定した場合にのみ必須です。

--- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

バージョンインジケータ (ACBXVER)

F2

コマンドコード (ACBXCMD)

L9

レスポンスコード (ACBXRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、エラーサブコード (ACBXERRC) フィールドにサブコードを伴う場合があります。詳細は、Adabas メッセージおよびコードのドキュメントを参照してください。

レスポンスコード 3 は、エンドオブファイル条件が検出されたことを示します。

コマンド ID (ACBXCID)

このフィールドには、空白およびゼロを除く値を指定する必要があります。この値は、Adabas が正しい順序で値を返し、フォーマットバッファを繰り返し解釈するのを避けるために使用されます。

このフィールドは、ファイルの順次読み込み中に変更しないでください。

このフィールドの第1バイトには16進数'FF'にセットしないでください。

データベース ID (ACBXDBID)

このフィールドを使用して、データベース ID を指定します。Adabas コールはこのデータベースに送られます。

このフィールドは4バイトのバイナリフィールドですが、現時点では2バイトのデータベース ID のみがサポートされています。したがって、データベース ID はフィールドの下位部分（右端のバイト）に指定し、その前にバイナリの0を付加する必要があります。

このフィールドがバイナリの0に設定されている場合は、Adabas APIはDDCARD入力データで指定されたADARUNカードのデータベースIDか、リンクされているリンクルーチンによりロードされたLNKGBLSモジュールで指定されているデフォルトのデータベースID値のいずれかを使用します。

ファイル番号 (ACBXFNR)

このフィールドを使用して、Adabas コールで送る必要のあるファイル数を指定します。

このフィールドは4バイトのバイナリフィールドですが、ファイル番号はフィールドの下位部分（右端のバイト）に指定し、その前にバイナリの0を付加する必要があります。

ISN：ピリオディックグループのオカレンス (ACBXISN)

値を求めたいディスクリプタがピリオディックグループに属する場合、このフィールドには、求められた値が見つかったオカレンス番号が返されます。オカレンス番号は下位2バイトにバイナリ形式で返されます。

プリフェッチオプションを指定した場合、プリフェッチした値のオカレンスはISNバッファ値の先頭のヘッダーに返されます。

ISN 下限：レコードバッファで最小の ISN (ACBXISL)

このフィールドに返される値は次のとおりです。

コマンドオプション1	コマンドオプション2	L9 コマンドの処理内容
空白	I	各値の ISN がレコードバッファに返されます。ISN 下限フィールドには、値は返されません。
	Iが設定されていない	返された ISN リストの先頭の ISN 番号が ISN 下限フィールドに入ります。
P (プリフェッチ)	I	先頭の ISN 番号がレコードバッファに返されます。プリフェッチされたすべてのディスクリプタ値は、先頭に16バイトのヘッダーが付いて ISN バッファに返されます。ISN 下限フィールドには値は返されません。
	Iが設定されていない	プリフェッチされた最後の値の先頭の ISN 番号が ISN 下限フィールドに入ります。

コマンドオプション1	コマンドオプション2	L9 コマンドの処理内容
M (マルチフェッチ)	I	マルチフェッチされたレコードのグループがレコードバッファに返されます。このレコードの記述は呼び出し元のISNバッファに返されます。ISN 下限フィールドには値は返されません。
	Iが設定されていない	マルチフェッチされた最後の値の先頭の ISN 番号が ISN 下限フィールドに入ります。

コマンドオプション1が"M" (マルチフェッチオプション) に設定されている場合、次の設定を行うことができます。

- ISN 下限フィールドにゼロ以外の値を設定すると、マルチフェッチする値の数を制限できません。
- すべての値をマルチフェッチする場合は、ISN 下限フィールドにゼロを指定します。

詳細は、「[マルチフェッチ／プリフェッチ機能の使用](#)」を参照してください。

ACBXISL フィールドは、8 バイトの ACBXISLG フィールドに埋め込まれた4バイトのバイナリフィールドで、まだ使用されていません。ACBXISLG フィールドの上位部分には、バイナリの0を設定します。

ISN 数：レコード件数 (ACBXISQ)

"RETURN ISN" (コマンドオプションI) を指定した場合を除いて、このフィールドにはレコードバッファに返された値を持つレコード数が返されます。

プリフェッチオプションを指定した場合、プリフェッチした値の数はISNバッファ値の先頭のヘッダーに返されます。

コマンドオプション1：コマンドレベルのマルチフェッチ／プリフェッチ オプション (ACBXCOP1)

このフィールドにMオプションを指定すると、(コマンドレベルの) マルチフェッチオプションが使用されることとなります。マルチフェッチオプションを使用すると、同時に複数のディスクリプタ値を読み込み、単一レコードをフェッチする時間がなくなるため、パフォーマンスを高めることができます。詳細は「[マルチフェッチ／プリフェッチ機能の使用](#)」を参照してください。

オプション	説明
M (マルチフェッチ)	<p>マルチフェッチ処理が有効です。</p> <p>このオプションと組み合わせて、次の設定を行うことができます。</p> <ul style="list-style-type: none"> ■ ISN 下限フィールドにゼロ以外の値を設定すると、マルチフェッチする値の数を制限できます。 ■ すべての値をマルチフェッチする場合は、ISN 下限フィールドにゼロを指定します。Mを指定すると、ISN バッファのデータフォーマットはレコードディスクリプタの標準フォーマットに合わされます。

M オプションが指定されている場合、L9 コマンドでは、マルチフェッチしたすべてのディスクリプタ値の先頭に16バイトのヘッダーを付け、マルチフェッチバッファに格納します。マルチフェッチバッファは、一番大きなディスクリプタ値に16バイトを加えた大きさを格納できる必要があります。ユーザープログラムに定義する実際のマルチフェッチバッファエリアは、対応するマルチフェッチバッファ ABD に指定されているマルチフェッチバッファの長さ以上の大きさである必要があります。

コマンドオプション 2：RETURN ISN オプション (ACBXCOP2)

オプション	説明
I	各値の ISN リストをレコードバッファに返します。L9 コマンドは、アソシエータ内のインバーテッドリストだけを読み込みます。データストレージにはアクセスしません。
A	ディスクリプタのエントリを昇順に処理します。
D	ディスクリプタのエントリを降順に処理します。

アディション 1：ディスクリプタ名 (ACBXADD1)

サーチバッファ長およびバリューバッファ長の両方にゼロが指定されている場合、アディション 1 フィールドの値は、値を求めるディスクリプタの名前になります。この名前は、フォーマットバッファで指定したディスクリプタ名と同じでなければなりません。

この場合、L9 は指定されたディスクリプタのすべての値をファイルの始めから処理します。

ディスクリプタ名は、このフィールドの先頭 2 桁に指定します。残りの桁は空白にしなければなりません。

アディション 3：パスワード (ACBXADD3)

このフィールドは、Adabas Security または ADAESI パスワードを指定するため使用します。データベース、ファイル、またはフィールドがセキュリティ保護されている場合は、有効なセキュリティパスワードまたは ADAESI パスワードをユーザーは指定する必要があります。パスワードを保護するために、コマンド処理時にアディション 3 フィールドに空白がセットされます。

アディション 5：フォーマット ID、グローバルフォーマット ID (ACBXADD5)

このコマンド用の内部フォーマットバッファを識別する別のフォーマット ID を表示したり、全ユーザーが内部フォーマットバッファを使用できるようにグローバルフォーマット ID を表示したりします。

アディション 5 フィールドの左端のビットに 0 が指定されていると、コマンド ID フィールドに指定された値がフォーマット ID として使用されます。

このビットに 1 がセットされていると、アディション 5 フィールドの 5 バイト目から 8 バイト目までがフォーマット ID として使用されます。

さらに、アディション 5 フィールドの第 1 バイトの最上位 (左端) 2 ビットに、1 (B'11') がセットされると、アディション 5 フィールドの全 8 バイトが、グローバルフォーマット ID として使用されます (つまり、複数ユーザーがこのフォーマット ID を同時に使用できます)。

詳細と例については、「[コマンドID](#)、[フォーマットID](#)、[グローバルフォーマットID](#)」を参照してください。

エラーサブコード (ACBXERRC)

コマンドがゼロ以外のレスポンスコードを返したときは、このフィールドにレスポンスコードの正確な意味を定義したサブコードが含まれます。レスポンスコードとサブコードについては、『[Adabas メッセージおよびコード](#)』を参照してください。

バッファ

L9 コマンドには、次のバッファが適用されます。

- [フォーマットバッファ](#)
- [レコードバッファ](#)
- [サーチバッファ](#)
- [バリューバッファ](#)
- [ISN バッファ](#)
- [マルチフェッチバッファ](#)

フォーマットバッファ



Note: フォーマットバッファではエレメントをコンマで区切り、ピリオドで終了します。フォーマットバッファの形式の詳細については、「[フォーマットバッファ](#)」を参照してください。

取得する値を返すときの形式をこのバッファに指定しなければなりません。

L9 操作のフォーマットバッファの形式は次のとおりです。

```
name [ ,length ] [ ,format ] .
```

上記の意味は次に示すとおりです。

- name** 値を求めるディスクリプタの名前。フォネティックディスクリプタとハイパーディスクリプタは指定できません。ユーザー出口で DECODE オプションを指定している場合に使用できるのは、照合ディスクリプタのみです。この場合に返される値は、インデックスではなく元のフィールド値です。マルチプルバリューフィールドとして定義したサブディスクリプタ、スーパーディスクリプタ、およびディスクリプタを指定できます。
- length** 取得する値の長さを指定します。長さを指定しないと、値はディスクリプタの標準長で返されます。
- format** 取得する値のフォーマットを指定します。指定したフォーマットは、ディスクリプタの標準フォーマットとの互換性が必要です。フォーマットを指定しない場合、値はディスクリプタの標準フォーマットで返されます。

レコードバッファ

サーチバッファおよびバリュースタックに指定したディスクリプタ値がこのフィールドに返されます。L9 コールごとに別の値が返されます。ディスクリプタに空値省略オプションが定義されていると、ディスクリプタ値が空値の場合は返されません。コマンドオプション 2 フィールドに "I" を設定した場合、要求した値自体の他、その値がある ISN のリストもレコードバッファに返されます。また、ISN は昇順で返されます。

ディスクリプタ値は次の形式で返されます。

```
length value count ISN-list
```

上記の意味は次に示すとおりです。

length 1 バイトのバイナリ値（返される値の長さ）。ディスクリプタのタイプに応じた標準長である場合、このフィールドは 0 になります。


value ディスクリプタ値

count 指定したディスクリプタのファイルに存在する値の数。この数は 1 または 2 バイトで返されます。1 バイトの場合、フォーマットは X'cc' で、cc には個数が入ります。2 バイトの場合、フォーマットは X'ccc' で、ccc には個数が入ります。

ISN-list コマンドオプション 2 フィールドに I を設定した場合、この値があるレコードの ISN がレコードバッファの残部に返されます。1 つのレコードバッファには、1 つの NI（ノーマルインデックス）ブロックの ISN リストが返されます。このため、次のようになります。

- レコードバッファは、NI ブロック全体を格納するのに十分な大きさである必要があります。
- 同一値が昇順 ISN リストに複数現れる場合があります。

サーチバッファ

 **Note:** サーチバッファでは要素をコンマで区切り、ピリオドで終了します。サーチバッファの形式の詳細については、「[サーチバッファ](#)」を参照してください。

サーチバッファ長およびバリュースタック長の両方にゼロが指定されている場合、アディショナル 1 フィールドの値は、値を求めるディスクリプタの名前になります。この場合、L9 は指定されたディスクリプタのすべての値をファイルの始めから処理します。サーチバッファとバリュースタックは使用されません。

開始値、終了値、あるいはその両方をバリュースタックに指定する場合、検索するディスクリプタエントリ数を制限するためにサーチバッファが必要です。

バリュースタックに指定したディスクリプタ値の長さおよびフォーマットが、指定ディスクリプタの標準長およびフォーマットと異なる場合には、これらをサーチバッファに指定しなければなりません。

L9 コマンド：ディスクリプタ値の読み込み

バリューストックに1つの値（つまり、コマンドオプション2にAを指定した場合には開始値、Dを指定した場合には終了値）を与えるとき、サーチバッファの構文は次のとおりです。

```
name [ i ] [ ,length ] [ ,format ] [ ,comparator ]
```

バリューストックに2つの値（つまり、開始値と終了値）を与えるとき、サーチバッファの構文は次のとおりです。

```
name [ i ] [ ,length ] [ ,format ] ,S ,name [ ,length ] [ ,format ]
```

ここで、サーチバッファの要素は次のとおりです。

name	値を求めるディスクリプタの名前。フォーマットバッファで指定した名前と同じ名前である必要があります。								
i	ディスクリプタがピリオディックグループに属し、特定のオカレンス値だけを求める場合は、オカレンス番号の1桁から3桁のインデックスをディスクリプタ名に指定します。								
length	バリューストックに指定した値の長さ。長さを指定しない場合、ディスクリプタの標準長に応じて値が指定されます。指定可能な長さに関するページを参照してください。								
format	バリューストックに指定した値のフォーマット。フォーマットを指定しない場合、ディスクリプタの標準フォーマットに応じて値が指定されます。指定可能なフォーマットに関するページを参照してください。								
comparator	読み込み範囲。 <table border="1"> <tr> <td>GE</td> <td>値範囲の中の最大値よりも大きいか等しい（デフォルト）。</td> </tr> <tr> <td>GT</td> <td>値範囲の中の最大値よりも大きい。</td> </tr> <tr> <td>LE</td> <td>値範囲の中の最小値よりも小さいか等しい。</td> </tr> <tr> <td>LT</td> <td>値範囲の中の最小値よりも小さい。</td> </tr> </table>	GE	値範囲の中の最大値よりも大きいか等しい（デフォルト）。	GT	値範囲の中の最大値よりも大きい。	LE	値範囲の中の最小値よりも小さいか等しい。	LT	値範囲の中の最小値よりも小さい。
GE	値範囲の中の最大値よりも大きいか等しい（デフォルト）。								
GT	値範囲の中の最大値よりも大きい。								
LE	値範囲の中の最小値よりも小さいか等しい。								
LT	値範囲の中の最小値よりも小さい。								
S	2つの検索式を使って開始-終了の範囲を示します。2つの式に同じディスクリプタを指定する必要があります。 <table border="1"> <tr> <td>AA,S,AA.</td> <td>有効</td> </tr> <tr> <td>AA,S,AB.</td> <td>無効</td> </tr> </table>	AA,S,AA.	有効	AA,S,AB.	無効				
AA,S,AA.	有効								
AA,S,AB.	無効								

バリューストック

サーチバッファ長とバリューストック長のどちらもバイナリの0に設定した場合、アクション1フィールドの値は、値を取得するディスクリプタの名前になります。この場合、L9は指定されたディスクリプタのすべての値を、コマンドオプション2フィールドで指定した順序で処理します。順序は、Aオプションを指定した場合にはファイルの始めから、Dオプションを指定した場合には終わりからになります。サーチバッファとバリューストックは使用されません。

開始値、終了値、あるいはその両方をバリューストックに指定する場合、検索するディスクリプタエントリ数を制限するためにサーチバッファが必要です。

バリューストックに与えた値が1つで、サーチバッファの比較演算子をGEかGTにすると、その値はコマンドオプション2がAの場合は開始値、Dの場合は終了値になります。

バリュースタックに2つの値が指定されている場合、1つ目の値は範囲の下限値、2つ目の値は上限値を示します。各値が開始値であるか終了値であるかは、コマンドオプション2での指定がAかDかによって決まります。

指定値がファイル内に存在しないと、コマンドオプション2の設定がAかDかによって、Adabasは次に高い値か低い値を探します。この場合、サーチバッファ比較演算子LEとGEはそれぞれLTとGTと同じになります。指定値も存在せず、より高い（または低い）値も存在しない場合は、レスポンスコード3が返されます。

ISN バッファ

コマンドレベルのマルチフェッチオプションを使用する場合と、このコマンドを **ACB** **ダイレクトコールインターフェイス** から発行する場合にのみ、ISN バッファが使用されます。L9 コマンドの実行後に ISN バッファに保持される情報は、コマンドオプション1 フィールドに M（マルチフェッチ）または P（プリフェッチ）オプションを指定した結果となります。ISN バッファのデータフォーマットは、指定したオプションに応じて異なります。

- **マルチフェッチオプション (M) のデータフォーマット**
- **プリフェッチオプション (P) のデータフォーマット**

マルチフェッチオプション (M) のデータフォーマット

レコードディスクリプタのデータフォーマットについては、「**READ (Lx) コマンドのマルチフェッチ処理**」セクションを参照してください。

プリフェッチオプション (P) のデータフォーマット

 **Note:** プリフェッチオプションは、**ACB** **インターフェイスダイレクトコール**でのみ有効です。

ISN バッファには、プリフェッチオプションでプリフェッチしたディスクリプタ値（16バイトのヘッダー付き）が保持されます。各値の先頭にある16バイトのヘッダーのフォーマットは次のとおりです。

バイト	使用方法
1~2	ディスクリプタの長さ（このヘッダーを含む）
3~4	ニュークリアスレスポンスコード
5~8	ニュークリアス内部 ID
9~12	ピリオディックグループオカレンス（ISN フィールドの説明参照）
13~16	レコード数（ISN 数フィールドの説明参照）

マルチフェッチバッファ

コマンドレベルのマルチフェッチオプションを使用する場合と、このコマンドを **ACBX** **ダイレクトコールインターフェイス** から発行する場合にのみ、マルチフェッチバッファが使用されます。レコードディスクリプタのデータフォーマットについては、「**READ (Lx) コマンドのマルチフェッチ処理**」セクションを参照してください。詳細は、「**マルチフェッチ/プリフェッチ機能の使用**」を参照してください。

その他の考慮事項

L9 コマンドを使用するときには、次の事項を考慮する必要があります。

1. L9 コマンドで使用したコマンド ID は、Adabas により内部的に保存および使用されます。この ID は、エンドオブファイル条件が検出されるか、RC コマンドか CL コマンドが発行されるか、Adabas セッションが終了したときに解放されます。コマンド ID は、解放されるまで、別のコマンドには使用できません。
2. L9 コマンドを使用して読み込んでいるファイルを他のユーザーが更新している場合は、L9 コマンドでそのファイルを読み込んでも、レコードが返されないことがあります。
3. L9 コマンドを使用して読み込んでいるファイルに対して、レコードを更新および削除できます。Adabas は L9 コマンドで返した最終値と次に返す値を保存し、コール間で更新または削除が行われても正しく次の値を返します。ただし、L9 コマンドでアクセスされるレコードはいくつかの理由（例えば、他のユーザーが更新または削除した場合）により変更された場合、L9 コマンドでは変更がなかったかのように処理を続行します。つまり、インバーテッドリストのエントリにアクセスする直前に他のコマンドでレコードが削除されても、L9 コマンドではそのレコードがまだ有効なエントリであるとみなされます。
4. L9 コマンドで使用する内部フォーマットバッファは、先の L9 コマンドが作成したものである必要があります。L9 以外のコマンドは、L9 コマンドで作成された内部フォーマットバッファを使用できません。

36 LF コマンド：フィールド定義の読み込み

- 機能および使用 354
- ACB インターフェイスダイレクトコール：LF コマンド 354
- ACBX インターフェイスダイレクトコール：LF コマンド 357
- バッファ 360

LF コマンドは、ファイル内の全フィールドの定義情報を読み込みます。

このchapterでは、次のトピックについて説明します。

機能および使用

LF コマンドは、ファイルのフィールド定義情報を読み込むのに使用します。このコマンドは本来Adabasサブシステムで使用されるものであり、一般にアプリケーションプログラムでは使用されません。

ユーザーは取得するフィールド定義があるファイルの番号を指定します。

Adabasはコマンドオプション2フィールドの指定に従って、レコードバッファに3種類のフォーマットでフィールド情報を返します。

ACB インターフェイスダイレクトコール：LF コマンド

このsectionでは、LFコマンドのACBインターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの概要](#)
- [コントロールブロックフィールドの説明](#)
- [ACB の例](#)

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用のACBフィールドにバイナリの0を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	--	--	--
コマンドコード	3~4	英数字	F	U
	5~8	--	--	--
ファイル番号	9~10	バイナリ	F	U
レスポンスコード	11~12	バイナリ	--	A
	13~26	--	--	--
レコードバッファ長	27~28	バイナリ	F	U
	29~35	--	--	--

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
コマンドオプション 2	36	英数字	F	U
	37~48	--	--	--
アディション 3	49~56	英数字	F	A
	57~72	--	--	--
コマンドタイム	73~76	バイナリ	--	A
ユーザーエリア	77~80	--	--	U

バッファエリア

バッファ	Adabas コール前	Adabas コール後
フォーマット	*	--
レコード	--	A

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

* 使用しませんが、コールステートメントのパラメータリストに含める必要があります。

-- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

コマンドコード (ACBCMD)

LF

ファイル番号 (ACBFNR)

フィールド定義情報を取得するファイルの番号。



Note: 2 バイトファイル番号およびデータベース ID を使用する場合は、コントロールブロックの先頭バイトに X'30' を入力しなければなりません。

レスポンスコード (ACBRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。LF コマンドによってゼロ以外のレスポンスコードが返された場合、Adabas コントロールブロックのバイト 45~48 (アディション 2 フィールドの下位 2 バイト) に、レスポンスコードの意味を正確に定義したサブコードが含まれていることがあります。レスポンスコードとサブコードについては、『Adabas メッセージおよびコード』を参照してください。

レコードバッファ長 (ACBRBL)

レコードバッファ長 (バイト単位)。指定長は、そのファイルのフィールド定義情報が十分入るだけの長さが必要です。ただし、ユーザープログラムで定義したレコードバッファエリアのサイズを超えない値を指定する必要があります。内部フォーマット (I) コマンドオプションを指定した場合、最大で4アソシエータブロックの出力が可能です。

コマンドオプション2：表示する情報のタイプ (ACBCOP2)

コマンドオプション2フィールドの設定により、レコードバッファに返されるフィールド情報のフォーマットと種類が決定されます。

オプション	説明
S	<p>照合ディスクリプタ、サブフィールド、スーパーフィールド、サブディスクリプタ、スーパーディスクリプタ、ハイパーディスクリプタ、フォネティックディスクリプタの情報を含む全フィールド情報が返されます。</p> <p>Adabas 8 を使用している場合、コマンドオプション2に "S" を設定して LF コマンドを実行したときに、ラージオブジェクト (LB) フィールドが検出されると、LB フィールドの状態が F タイプのフィールドエレメントに返されます。第2フォーマットバイト (エレメントのオフセット7またはバイト8) のビット6が、このフィールドに LB (ラージオブジェクト) オプションが設定されたことを示すために設定されました。第2フォーマットバイトのビット1は、LB フィールドの定義に NB (空白圧縮なし) オプションが指定されているかどうかを表します。</p> <p>注意: ACB インターフェイスダイレクトコールでは、あまり大きな LB フィールドデータを格納したり、読み込んだりすることができません。すべてのデータは、最大 32 KB (-1 バイト) 長のレコードバッファセグメントに収まる大きさである必要があります。</p>
I	Adabas の内部形式で全フィールド情報が返されます。

空白またはバイナリの0を指定すると、サブディスクリプタ、スーパーディスクリプタ、ハイパーディスクリプタ、フォネティックディスクリプタ、照合ディスクリプタの情報を除くフィールド情報が返されます。これは、Adabas バージョン4と同じ形式です。

アディション3：パスワード (ACBADD3)

セキュリティ保護のためのパスワードを指定します。使用するファイルがセキュリティ保護されていない場合、このフィールドは空白にしておきます。ファイルがセキュリティ保護されている場合は、該当するパスワードを指定しなければなりません。

アクセスしたファイルがパスワード保護されている場合は、パスワード値を保護するために、コマンド処理後空白が設定されます。

ACB の例

ファイル 1 のフィールド定義情報を読み込みます。

コントロールブロック

コマンドコード	LF	
ファイル番号	1	ファイル 1 のフィールド定義を要求します。
レコードバッファ長	100	
コマンドオプション 2	S	全ディスクリプタおよびサブ/スーパーフィールドの情報が返されます。
アデプション 3	bbbbbbbb (空白)	ファイルはセキュリティ保護されていません

ACBX インターフェイスダイレクトコール：LF コマンド

この section では、LF コマンドの ACBX インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの概要](#)
- [コントロールブロックフィールドの説明](#)

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	---	---	---
バージョンインジケータ	3~4	バイナリ	F	U
	5~6	---	---	---
コマンドコード	7~8	英数字	F	U
	9~10	---	---	---
レスポンスコード	11~12	バイナリ	---	A
	13~16	---	---	---
データベース ID	17~20	数値	F	U
ファイル番号	21~24	数値	F	U
	25~49	---	---	---
コマンドオプション 2	50	英数字	F	U

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	51～68	---	---	---
アディション3	69～76	英数字／バイナリ	F	A
	77～114	---	---	---
エラーサブコード	115～116	バイナリ	---	A
	117～144	---	---	---
コマンドタイム	145～152	バイナリ	---	A
ユーザーエリア	153～168	該当なし	---	U
---	169～193	操作不可	---	---

ABD とバッファ

ABD とバッファ	Adabas コール前	Adabas コール後
フォーマット	*	--
レコード	--	A

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

* 使用しませんが、Adabas コールに含める必要があります。含めない場合は、Adabas コールが自動的に生成されます。

-- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

バージョンインジケータ (ACBXVER)

F2

コマンドコード (ACBXCMD)

LF

レスポンスコード (ACBXRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、エラーサブコード (ACBXERRC) フィールドにサブコードを伴う場合があります。詳細は、*Adabas メッセージおよびコードのドキュメント*を参照してください。

データベース ID (ACBXDBID)

このフィールドを使用して、データベース ID を指定します。Adabas コールはこのデータベースに送られます。

このフィールドは4バイトのバイナリフィールドですが、現時点では2バイトのデータベース ID のみがサポートされています。したがって、データベース ID はフィールドの下位部分（右端のバイト）に指定し、その前にバイナリの0を付加する必要があります。

このフィールドがバイナリの0に設定されている場合は、Adabas APIはDDCARD入力データで指定された ADARUN カードのデータベース ID か、リンクされているカリンクルーチンによりロードされた LNKGBLS モジュールで指定されているデフォルトのデータベース ID 値のいずれかを使用します。

ファイル番号 (ACBXFNR)

このフィールドは、フィールド定義情報を取得するファイルの番号を指定するのに使用します。

このフィールドは4バイトのバイナリフィールドですが、ファイル番号はフィールドの下位部分（右端のバイト）に指定し、その前にバイナリの0を付加する必要があります。

コマンドオプション 2：表示する情報のタイプ (ACBXCOP2)

コマンドオプション 2 フィールドの設定により、レコードバッファに返されるフィールド情報のフォーマットと種類が決定されます。

オプション	説明
S	照合ディスクリプタ、サブフィールド、スーパーフィールド、サブディスクリプタ、スーパーディスクリプタ、ハイパーディスクリプタ、フォネティックディスクリプタの情報を含む全フィールド情報が返されます。 Adabas 8 を使用している場合、コマンドオプション 2 に "S" を設定して LF コマンドを実行したときに、ラージオブジェクト (LB) フィールドが検出されると、LB フィールドの状態が F タイプのフィールドエレメントに返されます。第2フォーマットバイト（エレメントのオフセット7またはバイト8）のビット6が、このフィールドに LB（ラージオブジェクト）オプションが設定されたことを示すために設定されました。第2フォーマットバイトのビット1は、LB フィールドの定義に NB（空白圧縮なし）オプションが指定されているかどうかを表します。
I	Adabas の内部形式で全フィールド情報が返されます。

空白またはバイナリの0を指定すると、サブディスクリプタ、スーパーディスクリプタ、ハイパーディスクリプタ、フォネティックディスクリプタ、照合ディスクリプタの情報を除くフィールド情報が返されます。これは、Adabas バージョン 4 と同じ形式です。

アディション 3：パスワード (ACBXADD3)

セキュリティ保護のためのパスワードを指定します。使用するファイルがセキュリティ保護されていない場合、このフィールドは空白にしておきます。ファイルがセキュリティ保護されている場合は、該当するパスワードを指定しなければなりません。

アクセスしたファイルがパスワード保護されている場合は、パスワード値を保護するために、コマンド処理後空白が設定されます。

エラーサブコード (ACBXERRC)

コマンドがゼロ以外のレスポンスコードを返したときは、このフィールドにレスポンスコードの正確な意味を定義したサブコードが含まれます。レスポンスコードとサブコードについては、『*Adabas* メッセージおよびコード』を参照してください。

バッファ

LF コマンドには、次のバッファが適用されます。

- フォーマットバッファ
- レコードバッファ

フォーマットバッファ

フォーマットバッファは LF コマンドでは使用されませんが、Adabas コールに含める必要があります。これが **ACB インターフェイスダイレクトコール** であり、フォーマットバッファが指定されていない場合は、処理エラーが発生します。ACB インターフェイスダイレクトコールはセットシーケンスにバッファが指定されていることを要求します。これが **ACBX インターフェイスダイレクトコール** であり、フォーマットバッファが指定されていない場合は、バッファが自動的に生成されます。

レコードバッファ

全フィールド定義は、レコードバッファに返されます。

- コマンドオプション 2 が "S" の場合
- コマンドオプション 2 が "I" の場合
- コマンドオプション 2 が "S" および "I" 以外の場合

コマンドオプション 2 が "S" の場合

コマンドオプション 2 フィールドに "S" を指定すると、照合ディスクリプタ、サブディスクリプタ、スーパーディスクリプタ、ハイパーディスクリプタ、フォネティックディスクリプタ、サブフィールド、スーパーフィールドなど、あらゆるフィールド情報が次のフォーマットで返されます。



Note: Adabas 8 を使用している場合、コマンドオプション 2 に "S" を設定して LF コマンドを実行したときに、ラージオブジェクト (LB) フィールドが検出されると、LB フィールドの状態が F タイプの FDT フィールド定義に返されます。第 2 フォーマットバイト (エレメントのオフセット 7 またはバイト 8) のビット 6 が、このフィールドに LB (ラージオブジェクト) オプションが設定されたことを示すために設定されました。第 2 フォー

マットバイトのビット 1 は、LB フィールドの定義に NB（空白圧縮なし）オプションが指定されているかどうかを表します。

バイト	使用方法
1~2	情報の合計長
3~4	FDT にあるフィールド数 (SDT を含む)
5~n	フィールド定義。各エントリは 8 バイトで、最大エントリ数は 926 です。
(n+1) ~m	次の情報からなる特殊ディスクリプタテーブル (SDT) <ul style="list-style-type: none"> ■ サブ/スーパーディスクリプタ (またはサブ/スーパーフィールド) ■ フォネティックディスクリプタ ■ ハイパーディスクリプタ ■ 照合ディスクリプタ <p>サブ/フォネティック/照合ディスクリプタ要素の長さは 8 バイトです。スーパー/ハイパーディスクリプタ要素の長さは、8 バイトのエントリが 2 つ以上入る大きさとなります。</p>

FDT フィールド定義


FDT フィールド定義の構文は次のとおりです。

```
'F' field-name option level length format
```

次の表でこの構文について説明します。

記号	バイト	使用方法																
'F'	1	'F'は FDT フィールド定義を示します。																
<i>field-name</i>	2~3	フィールド名																
<i>option</i>	4	定義オプション <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">ビット 1=1</td> <td>ディスクリプタ</td> </tr> <tr> <td>ビット 2=1</td> <td>固定長</td> </tr> <tr> <td>ビット 3=1</td> <td>マルチプルバリューフィールド</td> </tr> <tr> <td>ビット 4=1</td> <td>空値省略</td> </tr> <tr> <td>ビット 5=1</td> <td>ピリオディックグループフィールド</td> </tr> <tr> <td>ビット 6=1</td> <td>フォネティックディスクリプタの親</td> </tr> <tr> <td>ビット 7=1</td> <td>サブ/スーパーディスクリプタの親</td> </tr> <tr> <td>ビット 8=1</td> <td>ユニークディスクリプタ</td> </tr> </table>	ビット 1=1	ディスクリプタ	ビット 2=1	固定長	ビット 3=1	マルチプルバリューフィールド	ビット 4=1	空値省略	ビット 5=1	ピリオディックグループフィールド	ビット 6=1	フォネティックディスクリプタの親	ビット 7=1	サブ/スーパーディスクリプタの親	ビット 8=1	ユニークディスクリプタ
ビット 1=1	ディスクリプタ																	
ビット 2=1	固定長																	
ビット 3=1	マルチプルバリューフィールド																	
ビット 4=1	空値省略																	
ビット 5=1	ピリオディックグループフィールド																	
ビット 6=1	フォネティックディスクリプタの親																	
ビット 7=1	サブ/スーパーディスクリプタの親																	
ビット 8=1	ユニークディスクリプタ																	
<i>level</i>	5	レベル番号 (バイナリ)																
<i>length</i>	6	長さ																
<i>format</i>	7	データのタイプ																

記号	バイト	使用方法
		A 英数字
		B バイナリ
		F 固定小数点
		G 浮動小数点
		P パック 10 進数
		U アンパック 10 進数
		W ワイド文字
	8	オプション (続き)
	ビット 1	未使用。ただし、Adabas 8 を使用しており、コマンドオプション 2 に "S" を設定して LF コマンドを実行したときに、ラージオブジェクト (LB) フィールドが検出された場合を除く。この場合、ビット 1 は LB フィールドが NB (空白圧縮なし) オプションで定義されているかどうかを示します。
	ビット 2=1	NV (変換なし) オプション
	ビット 3	(未使用)
	ビット 4=1	XI (UQ から PE オカレンス番号を除外する) オプション
	ビット 5=1	LA (ロング英数字フィールド) オプション
	ビット 6	未使用。ただし、Adabas 8 を使用しており、コマンドオプション 2 に "S" を設定して LF コマンドを実行したときに、ラージオブジェクト (LB) フィールドが検出された場合を除く。この場合、ビット 6 はフィールドに LB (ラージオブジェクト) オプションが設定されていることを示します。
	ビット 7=1	NN オプション
	ビット 8=1	NC オプション

 **Note:** ピリオディックグループ内のフィールドは、次の属性を持ちます。オプションフィールド (バイト 4) のビット 5=1。レベルフィールド (バイト 5) は 1 より大きいレベル番号になります。ピリオディックグループフィールド自体は、常にオプションのビット 5=1 で、レベル番号 1 です。

SDT フィールド定義

X SDT-definition

Xには次のいずれかが入ります。

- C 照合ディスクリプタ（「[照合ディスクリプタ定義](#)」セクション参照）
- H ハイパーディスクリプタ（「[ハイパーディスクリプタ定義](#)」セクション参照）
- P フォネティックディスクリプタ（「[フォネティック定義](#)」セクション参照）
- S サブフィールド／ディスクリプタ（「[サブフィールド／サブディスクリプタ定義](#)」セクション参照）
- T スーパーフィールド／ディスクリプタ（「[スーパーフィールド／スーパーディスクリプタ定義](#)」セクション参照）

X'00' エレメントの続き

照合ディスクリプタ定義

```
'C' name option exit length p-field-name
```

記号	バイト	使用方法	
C	1	照合ディスクリプタを示します。	
name	2~3	照合ディスクリプタ名	
option	4	定義オプション	
		ビット 1=1	ディスクリプタ
		ビット 2=1	UQ から PE オカレンス番号を除外
		ビット 3=1	マルチプルバリュー形式
		ビット 4=1	空値省略
		ビット 5=1	ピリオディックグループフィールド
		ビット 6~7	(未使用)
		ビット 8=1	ユニークディスクリプタ
exit	5	照合ディスクリプタ出口番号（バイナリ）（指定可能値 1~8）	
length	6	長さ	
p-field-name	7~8	親フィールド名	

ハイパーディスクリプタ定義

```
'H' name option exit length format X'00'  
X'00' X'00' p-fieldname-list ...
```

記号	バイト	使用方法	
H	1	ハイパーディスクリプタ定義を示します。	
name	2~3	ハイパーディスクリプタ名	
option	4	定義オプション	
		ビット 1	(未使用)
		ビット 2=1	固定長
		ビット 3=1	マルチプルバリュー
		ビット 4=1	空値省略
		ビット 5=1	ピリオディックグループ
		ビット 6~7	(未使用)
		ビット 8=1	ユニークディスクリプタ
level	5	ハイパー出口番号 (バイナリ) (指定可能値 1~31)	
length	6	長さ	
format	7	フォーマット：	
		A	英数字
		B	バイナリ
		F	固定小数点
		G	浮動小数点
		P	パック 10 進数
		U	アンパック 10 進数
X'00'	8	オプション (続き)	
		ビット 1~3	(未使用)
		ビット 4=1	XI (UQ から PE オカレンス番号を除外する) オプション
		ビット 5~8	(未使用)

ハイパーディスクリプタ親フィールド名リストは、ハイパーディスクリプタ定義を拡張したものです。そのフォーマットは、8 バイトのグループごとに次のようになります。

記号	バイト	説明
X'00'	1	X'00' は定義の延長を示します。
X'00'	2	(未使用)
p-fieldname-list...	3~8	親フィールド名一覧。各 2 バイトで合計 6 バイトです (つまり、フィールドは 3 つ指定できます。3 つより少ない名前を指定した場合、残りの部分には X'00' が入ります)。

フォネティックディスクリプタ定義

```
'P' desc-name option p-field-name X'0000'
```

記号	バイト	説明
P	1	フォネティックディスクリプタを示します。
<i>desc-name</i>	2~3	フォネティックディスクリプタ名
<i>option</i>	4	(未使用)
<i>p-field-name</i>	5~6	親フィールド名
X'0000'	7~8	使用しません/空値に設定されます。

サブフィールド/サブディスクリプタ定義

```
'S' s-name option p-field-name from to
```

記号	バイト	使用方法	
S	1	サブディスクリプタ/サブフィールドを示します。	
<i>s-name</i>	2~3	サブディスクリプタ名またはサブフィールド名	
<i>option</i>	4	定義オプション	
		ビット 1=1	ディスクリプタ
		ビット 2=1	UQ から PE オカレンス番号を除外
		ビット 3=1	マルチプルバリュ形式
		ビット 4=1	空値省略
		ビット 5=1	ピリオディックグループフィールド
		ビット 6~7	(未使用)
		ビット 8=1	ユニークディスクリプタ
<i>p-field-name</i>	5~6	親フィールド名	
<i>from</i>	7	開始バイト	
<i>to</i>	8	終了バイト	

スーパーディスクリプタ/スーパーフィールド定義

```
'T' sup-name option p-field-name from to
X'00' X'000000'p-field-name from to
```

記号	バイト	使用方法
T	1	スーパーディスクリプタ/スーパーフィールドを示します。
<i>sup-name</i>	2~3	スーパーディスクリプタ/スーパーフィールド名
<i>option</i>	4	定義オプション

記号	バイト	使用方法
		ビット 1=1 ディスクリプタ
		ビット 2=1 UQ から PE オカレンス番号を除外
		ビット 3=1 マルチプルバリュー形式
		ビット 4=1 空値省略
		ビット 5=1 ピリオディックグループフィールド
		ビット 6~7 (未使用)
		ビット 8=1 ユニークディスクリプタ
<i>p-field-name</i>	5~6	親フィールド名
<i>from</i>	7	開始バイト
<i>to</i>	8	終了バイト

スーパーディスクリプタ/スーパーフィールド定義の延長は、8 バイトグループごとに、次の形式にしたがいます。

記号	バイト	説明
X'00'	1	定義の延長を示します。
X'000000'	2~4	(未使用)
<i>p-field-name</i>	5~6	親フィールド名
<i>from</i>	7	開始バイト
<i>to</i>	8	終了バイト

コマンドオプション 2 が "I" の場合

コマンドオプション 2 フィールドに "I" が設定されている場合、フィールド情報は次の Adabas 内部形式で返されます。

バイト	内容						
1	X'80'						
2	B'00000xyz' xyz には 暗号化ビット (1=yes、0=no) が入ります。 <table border="1" style="margin-left: 20px;"> <tr> <td>x</td> <td>ユーザー</td> </tr> <tr> <td>y</td> <td>新しい</td> </tr> <tr> <td>z</td> <td>古い</td> </tr> </table>	x	ユーザー	y	新しい	z	古い
x	ユーザー						
y	新しい						
z	古い						
3~4	バイナリの 0						
5~8	フィールド定義テーブル (FDT 固有)、FDT インデックス、特殊ディスクリプタテーブル (SDT) を含めた FDT の合計長 [=p 変数 4]						
9~12	フィールド定義テーブル (FDT 固有) の全長 [=n 変数 8]						
13~n	FDT フィールドディスクリプタエレメント (1 エレメントごとに 20 バイト。次の説明参照)						

バイト	内容
n+1~n+4	FDT インデックスの全長 [= m 変数 n]
n+5~m	FDT インデックス
m+1~m+4	特殊ディスクリプタテーブル (SDT) の全長 [= p 変数 m]
m+5~p	特殊ディスクリプタテーブル (SDT)


各FDTフィールドディスクリプタエレメントのフォーマットについては、次の表で説明します。

オフセット	内容
0	フィールドレベル
1 - 2	フィールド名
3	特殊なフィールドオプション
4 - 6	予約
7	デフォルトフィールド長
8	フィールドフォーマット
9	ディスクリプタ定義オプション
10	特殊なディスクリプタ親オプション
11	ピリオディックグループカウントフィールド
12~13	FDT エレメントチェーンポインタ
14	フィールドセキュリティレベル
15 - 19	予約

FDT エレメントの意味については、Adabas アーキテクチャ訓練情報で説明します。

FDT は、4 以下のアソシエータブロックに格納されます。従って、レコードバッファの最大長は 4 アソシエータブロック長です。

コマンドオプション 2 が "S" および "I" 以外の場合

 **Note:** 古いプログラムをサポートするため、コマンドオプション 2 フィールドに "I" または "S" 以外の値も指定できますが、この場合、新機能はサポートされません。新しいプログラムについては、その値を "I" または "S" に設定することをお勧めします。

コマンドオプション 2 フィールドに "I" または "S" 以外の値を指定した場合、照合/サブ/スーパー/ハイパー/フォネテックのディスクリプタ情報を除くフィールド情報が返されます。これは、Adabas バージョン 4 と同じフォーマットです。

n field-def

LF コマンド：フィールド定義の読み込み

上記の意味は次に示すとおりです。

n ファイル中のフィールド数。この数は、レコードバッファの最初の4バイトに4バイトの2進数として表示されます。

field-def ファイル中の各フィールドに関するフィールド定義情報。各フィールド情報は、次表のフォーマットに基づき6バイトで表示されます。

バイト	使用方法
1	レベル番号 (バイナリ)
2-3	名前 (英数字)
4	標準長 (バイナリ)
5	標準フォーマット (英数字)
	A 英数字
	B バイナリ
	F 固定小数点
	G 浮動小数点
	P パック 10 進数
	U アンパック 10 進数
	W ワイド文字
6	定義オプション
	ビット 1=1 ディスクリプタ
	ビット 2=1 固定ストレージ
	ビット 3=1 マルチプルバリューフィールド
	ビット 4=1 空値省略
	ビット 5=1 ピリオディックグループフィールド
	ビット 6=1 フォネティックソースフィールド
	ビット 7=1 サブ/スーパーディスクリプタソースフィールド
	ビット 8=1 ユニークディスクリプタ

フィールド定義情報は間に空白を置かず、直前のフィールド定義情報に続きます。

37 N1 コマンドと N2 コマンド：レコードの追加

- 機能および使用 370
- ACB インターフェイスダイレクトコール：N1 コマンドおよび N2 コマンド 370
- ACBX インターフェイスダイレクトコール：N1 コマンドおよび N2 コマンド 375
- バッファ 380

N1 コマンドと N2 コマンドは、ファイルに新しいレコードを追加します。

N1 コマンドは、Adabas が割り当てた ISN を持つ新規データベースレコードを追加します。N2 コマンドは、ユーザーが割り当てた ISN を持つ新規データベースレコードを追加します。

このchapterでは、次のトピックについて説明します。

機能および使用

ユーザーは、レコードを追加するファイルと値を与えるフィールドを指定します。追加レコード内で指定のないフィールドには空値が入ります。

Adabas はレコードに ISN を割り当て、レコードをデータストレージに追加し、必要に応じてアソシエータの更新を行います。

N2 コマンドは、レコードに割り当てる ISN をユーザーが指定する場合に使用します。レコードに割り当てた ISN をそのまま維持するには、アンロードしたファイルを USERISN=YES オプション付きで再ロードする必要があります。

ユーザーが ET ロジックユーザーでありマルチユーザーモードで実行している場合、追加したレコードはホールド状態になります。

ACB インターフェイスダイレクトコール：N1 コマンドおよび N2 コマンド

このsectionでは、N1 コマンドおよび N2 コマンドの ACB インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの概要](#)
- [コントロールブロックフィールドの説明](#)
- [ACB の例](#)

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	--	--	--
コマンドコード	3~4	英数字	F	U
コマンド ID	5~8	英数字	F	U
ファイル番号	9~10	バイナリ	F	U
レスポンスコード	11~12	バイナリ	--	A
ISN	13~16	バイナリ	F	A/U ¹
ISN 下限	17~20	バイナリ	--	A ²
ISN 数	21~24	バイナリ	--	A ²
フォーマットバッファ長	25~26	バイナリ	F	U
レコードバッファ長	27~28	バイナリ	F	U
	29~44	--	--	--
アディション 2	45~48	英数字	--	A
アディション 3	49~56	英数字	F	A
アディション 4	57~64	英数字	F	A
アディション 5	65~72	英数字	F	U
コマンドタイム	73~76	バイナリ	--	A
ユーザーエリア	77~80	--	--	U

注

1. N1 コマンドの場合には Adabas によりデータが設定され、N2 コマンドの場合には変更はありません。
2. これらのフィールドは使用されますが、カップリングファイルが使用されている場合は Adabas によりリセットされません。

バッファエリア

バッファ	Adabas コール前	Adabas コール後
フォーマット	F	U
レコード	F	U

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

-- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

コマンドコード (ACBCMD)

N1 または N2

コマンド ID (ACBCID)

複数のレコードを一連の N1 または N2 コールで追加し、各コールで同一フィールドをフォーマットバッファに指定する場合は、空白またはゼロ以外の値を設定します。これにより各 N1 または N2 コール処理に必要な時間を削減できます。

1 レコードだけを追加する場合やフォーマットバッファを N1 または N2 コール間で変更する場合は、このフィールドを空白にしておきます。

このフィールドの第 1 バイトには 16 進数 'FF' にセットしないでください。

ファイル番号 (ACBFNR)

読み込むファイルの番号を 2 進数で指定します。物理ダイレクトコールの場合は、次のようファイル番号を指定します。

- 1 バイトファイル番号の場合は、ファイル番号を第 2 バイト (10) に入力します。第 1 バイト (9) は、バイナリの 0 (B'0000 0000') をセットします。
- 2 バイトファイル番号の場合は、2 バイト (9 と 10) を使います。



Note: 2 バイトファイル番号およびデータベース ID を使用する場合は、コントロールブロックの先頭バイトに X'30' を入力しなければなりません。

レスポンスコード (ACBRSP)


Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、アディクション 2 フィールドの下位 2 バイトにサブコードを伴う場合があります。詳細は『Adabas メッセージおよびコード』を参照してください。

ISN (ACBISN)

N1 コマンドを実行する場合、レコードに割り当てた ISN が Adabas から返されます。

N2 コマンドを実行する場合は、レコードに割り当てる ISN を指定する必要があります。ファイル中のレコードにすでに割り当てられている ISN は指定できません。また、このファ

イルに対する制限 (MAXISN) の範囲内で指定する必要があります。MAXISN は、ファイルのロード時に DBA が指定します。

 **Note:** 該当ファイルに MAXISN パラメータで指定された値を超える ISN は、割り当てられません。

ISN 数/下限 (ACBISQ および ACBISL)

これらのフィールドは、ハードカップリングファイルが使用されない限り、N1 または N2 コマンド処理の終了後、空値に設定されます。カップリングファイルを使用した場合、これらのフィールドは N1 または N2 コマンド処理で使用され、リセットされません。

フォーマットバッファ長 (ACBFBL)

フォーマットバッファ長 (バイト単位)。ユーザープログラムに定義するフォーマットバッファエリアは、この指定長と同じか、それ以上でなければなりません。

レコードバッファ長 (ACBRBL)

レコードバッファ長 (バイト単位)。ユーザープログラムに定義するレコードバッファエリアは、この指定長と同じか、それ以上でなければなりません。

アディクション 2：圧縮レコード長 (ACBADD2)

コマンドが正常に処理されると、次の情報がこのフィールドに返されます。

- 少なくとも 1 つの有効なフィールド値がレコードバッファにある場合、先頭 2 バイトには、新たに追加した圧縮レコード長がバイナリ形式で入ります。
- N1 コマンドまたは N2 コマンドによってゼロ以外のレスポンスコードが返された場合、下位 2 バイトにレスポンスコードの正確な意味を定義したサブコードが含まれていることがあります。レスポンスコードとサブコードについては、『*Adabas メッセージおよびコード*』を参照してください。

アディクション 3：パスワード (ACBADD3)

このフィールドは、Adabas Security または ADAESI パスワードを指定するため使用します。データベース、ファイル、またはフィールドがセキュリティ保護されている場合は、有効なセキュリティパスワードまたは ADAESI パスワードをユーザーは指定する必要があります。パスワードを保護するために、コマンド処理時にアディクション 3 フィールドに空白がセットされます。

アディクション 4：サイファコード (ACBADD4)

このフィールドは、サイファコードを指定するために使用します。ファイルを暗号化する場合は、有効なサイファコードを指定する必要があります。ファイルを暗号化しない場合は、このフィールドに空白をセットする必要があります。

Adabas は、コマンド処理時に、サイファコードに空白をセットし、バージョンコードとデータベース ID をフィールドの下位 3 バイトに返します。詳細については「[コントロールブロックフィールド](#)」を参照してください。

アディクション 5：フォーマット ID、グローバルフォーマット ID (ACBADD5)

このコマンド用の内部フォーマットバッファを識別する別のフォーマット ID を表示したり、全ユーザーが内部フォーマットバッファを使用できるようにグローバルフォーマット ID を表示したりします。

アディクション5 フィールドの左端のビットに0が指定されていると、コマンド ID フィールドに指定された値がフォーマット ID として使用されます。

このビットに1がセットされていると、アディクション5 フィールドの5バイト目から8バイト目までがフォーマット ID として使用されます。

さらに、アディクション5フィールドの第1バイトの最上位（左端）2ビットに、1（B'11'）がセットされると、アディクション5フィールドの全8バイトが、グローバルフォーマット ID として使用されます（つまり、複数ユーザーがこのフォーマット ID を同時に使用できます）。

詳細と例については、「[コマンドID](#)、[フォーマットID](#)、[グローバルフォーマットID](#)」を参照してください。

ACB の例

- 例 1
- 例 2

例 1

1つのレコードをファイル1に追加します。レコードのISNはAdabasが割り当てます。指定するフィールド値は、次のとおりです。

フィールド	値
AA	ABCD
MF (値1)	AAA
MF (値2)	BBB
BA (第1オカレンス)	5
BA (第2オカレンス)	6

コントロールブロック

コマンドコード	N1	
コマンド ID	bbbb (空白)	1レコードのみを追加します。
ファイル番号	1	
フォーマットバッファ長	15	またはそれ以上
レコードバッファ長	16	またはそれ以上
アディクション3	bbbbbbbb (空白)	ファイル1はセキュリティ保護されていません
アディクション4	bbbbbbbb (空白)	ファイルは暗号化されない

バッファエリア

フォーマットバッファ	AA,MF1-2,BA1-2.
レコードバッファ	X'C1C2C3C440404040C1C1C1C2C2C20506'

例 2

1つのレコードをファイル 2 に追加します。レコードの ISN はユーザーが指定します。指定するフィールド値は次のとおりです。

フィールド	値
RA	12345678
RB	ABCD

コントロールブロック

コマンドコード	N2	
コマンド ID	bbbb (空白)	1 レコードのみを追加します。
ファイル番号	2	
ISN	20	ISN 20 をレコードに割り当てます。
フォーマットバッファ長	6	またはそれ以上
レコードバッファ長	18	またはそれ以上
アディクション 3	パスワード	ファイル 2 はセキュリティ保護されています。
アディクション 4	bbbbbbbb (空白)	ファイルは暗号化されない

バッファエリア

フォーマットバッファ	RA,RB.
レコードバッファ	X'F1F2F3F4F5F6F7F8C1C2C3C44040404040'

ACBX インターフェイスダイレクトコール：N1 コマンドおよび N2 コマンド

このsectionでは、N1 コマンドおよび N2 コマンドの ACBX インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの概要](#)

■ コントロールブロックフィールドの説明

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	---	---	---
バージョンインジケータ	3~4	バイナリ	F	U
	5~6	---	---	---
コマンドコード	7~8	英数字	F	U
	9~10	---	---	---
レスポンスコード	11~12	バイナリ	---	A
コマンド ID	13~16	英数字/バイナリ	F	U
データベース ID	17~20	数値	F	U
ファイル番号	21~24	数値	F	U
	25~28	---	---	---
ISN	29~32	バイナリ	F	A/U ¹
	33~36	---	---	---
ISN 下限	37~40	バイナリ	---	A ²
	41~44	---	---	---
ISN 数	45~48	バイナリ	---	A ²
	49~64	---	---	---
アディクション 2	65~68	バイナリ	---	A
アディクション 3	69~76	英数字/バイナリ	F	A
アディクション 4	77~84	英数字	F	A
アディクション 5	85~92	英数字/バイナリ	F	U
	93~114	---	---	---
エラーサブコード	115~116	バイナリ	---	A
	117~144	---	---	---
コマンドタイム	145~152	バイナリ	---	A
ユーザーエリア	153~168	該当なし	---	U
---	169~193	操作不可	---	---

注

1. N1 コマンドの場合には Adabas によりデータが設定され、N2 コマンドの場合には変更はありません。
2. これらのフィールドは使用されますが、カップリングファイルが使用されている場合は Adabas によりリセットされません。

ABD とバッファ

ABD とバッファ	Adabas コール前	Adabas コール後
フォーマット	F	U
レコード	F	U

上記の意味は次に示すとおりです。

- F Adabas コール前にユーザーが入力するフィールド
- A Adabas により入力されるフィールド
- U Adabas コール後も変化なし
- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

バージョンインジケータ (ACBXVER)

F2

コマンドコード (ACBXCMD)

N1 または N2

レスポンスコード (ACBXRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、エラーサブコード (ACBXERRC) フィールドにサブコードを伴う場合があります。詳細は、Adabas メッセージおよびコードのドキュメントを参照してください。

コマンド ID (ACBXCID)

複数のレコードを一連の N1 または N2 コールで追加し、各コールで同一フィールドをフォーマットバッファに指定する場合は、空白またはゼロ以外の値を設定します。これにより各 N1 または N2 コール処理に必要な時間を削減できます。

1 レコードだけを追加する場合やフォーマットバッファを N1 または N2 コール間で変更する場合は、このフィールドを空白にしておきます。

このフィールドの第 1 バイトには 16 進数 'FF' にセットしないでください。

データベース ID (ACBXDBID)

このフィールドを使用して、データベース ID を指定します。Adabas コールはこのデータベースに送られます。

このフィールドは 4 バイトのバイナリフィールドですが、現時点では 2 バイトのデータベース ID のみがサポートされています。したがって、データベース ID はフィールドの下位部分（右端のバイト）に指定し、その前にバイナリの 0 を付加する必要があります。

このフィールドがバイナリの 0 に設定されている場合は、Adabas API は DDCARD 入力データで指定された ADARUN カードのデータベース ID か、リンクされているかリンクルーチンによりロードされた LNKGBLS モジュールで指定されているデフォルトのデータベース ID 値のいずれかを使用します。

ファイル番号 (ACBXFNR)

このフィールドを使用して、Adabas コールで送る必要のあるファイル数を指定します。

このフィールドは 4 バイトのバイナリフィールドですが、ファイル番号はフィールドの下位部分（右端のバイト）に指定し、その前にバイナリの 0 を付加する必要があります。

ISN (ACBXISN)

N1 コマンドを実行する場合、レコードに割り当てた ISN が Adabas から返されます。

N2 コマンドを実行する場合は、レコードに割り当てる ISN を指定する必要があります。ファイル中のレコードにすでに割り当てられている ISN は指定できません。また、このファイルに対する制限 (MAXISN) の範囲内で指定する必要があります。MAXISN は、ファイルのロード時に DBA が指定します。



Note: 該当ファイルに MAXISN パラメータで指定された値を超える ISN は、割り当てられません。

ACBXISN フィールドは、8 バイトの ACBXISNG フィールドに埋め込まれた 4 バイトのバイナリフィールドで、まだ使用されていません。ACBXISNG フィールドの上位部分には、バイナリの 0 を設定します。

ISN 下限 (ACBXISL)

このフィールドは、ハードカップリングファイルが使用されない限り、N1 または N2 コマンド処理が終了後、空値にセットされます。カップリングファイルが使用されている場合は、このフィールドは N1 または N2 コマンドの処理に使用され、リセットされません。

ACBXISL フィールドは、8 バイトの ACBXISLG フィールドに埋め込まれた 4 バイトのバイナリフィールドで、まだ使用されていません。ACBXISLG フィールドの上位部分には、バイナリの 0 を設定します。

ISN 数 (ACBXISQ)

このフィールドは、ハードカップリングファイルが使用されない限り、N1 または N2 コマンド処理が終了後、空値にセットされます。カップリングファイルが使用されている場合は、このフィールドは N1 または N2 コマンドの処理に使用され、リセットされません。

アディクション 2：圧縮レコード長 (ACBXADD2)

コマンドが正常に処理され、かつ少なくとも 1つの有効なフィールド値がレコードバッファにある場合、先頭 2 バイトには、新たに追加した圧縮レコードの長さがバイナリ形式で格納されます。

アディクション 3：パスワード (ACBXADD3)

このフィールドは、Adabas Security または ADAESI パスワードを指定するため使用します。データベース、ファイル、またはフィールドがセキュリティ保護されている場合は、有効なセキュリティパスワードまたは ADAESI パスワードをユーザーは指定する必要があります。パスワードを保護するために、コマンド処理時にアディクション 3 フィールドに空白がセットされます。

アディクション 4：サイファコード (ACBXADD4)

このフィールドは、サイファコードを指定するために使用します。ファイルを暗号化する場合は、有効なサイファコードを指定する必要があります。ファイルを暗号化しない場合は、このフィールドに空白をセットする必要があります。

Adabas は、コマンド処理時に、サイファコードに空白をセットし、バージョンコードとデータベース ID をフィールドの下位 3 バイトに返します。詳細については「[コントロールブロックフィールド](#)」を参照してください。

アディクション 5：フォーマット ID、グローバルフォーマット ID (ACBXADD5)

このコマンド用の内部フォーマットバッファを識別する別のフォーマット ID を表示したり、全ユーザーが内部フォーマットバッファを使用できるようにグローバルフォーマット ID を表示したりします。

アディクション 5 フィールドの左端のビットに 0 が指定されていると、コマンド ID フィールドに指定された値がフォーマット ID として使用されます。

このビットに 1 がセットされていると、アディクション 5 フィールドの 5 バイト目から 8 バイト目までがフォーマット ID として使用されます。

さらに、アディクション 5 フィールドの第 1 バイトの最上位 (左端) 2 ビットに、1 (B'11') がセットされると、アディクション 5 フィールドの全 8 バイトが、グローバルフォーマット ID として使用されます (つまり、複数ユーザーがこのフォーマット ID を同時に使用できます)。

詳細と例については、「[コマンド ID、フォーマット ID、グローバルフォーマット ID](#)」を参照してください。

エラーサブコード (ACBXERRC)

コマンドがゼロ以外のレスポンスコードを返したときは、このフィールドにレスポンスコードの正確な意味を定義したサブコードが含まれます。レスポンスコードとサブコードについては、『[Adabas メッセージおよびコード](#)』を参照してください。

バッファ

- フォーマットバッファ
- レコードバッファ

フォーマットバッファ

レコードバッファに指定する値と対応するフィールドを、このバッファに指定しなければなりません。N1 コマンドを実行する場合、フォーマットバッファ内に下記のものがあるとはいけません。

- 選択条件付きフォーマット ("field-name operator value...")
- 編集マスク
- サブ/スーパーディスクリプタフィールドの参照。
- 同じフィールドの複数回指定 (マルチプルバリューフィールドを除く)
- レコードフォーマットの-N 形式の指定 (例えば、ABN、AB1-N)

フォーマットバッファ内に上記のいずれかが含まれていると、N1 コマンドに対してニュークリアスレスポンス 44 が返されます。フォーマットバッファに指定されていないフィールドには、追加レコード内では空値がセットされます。フォーマットバッファ構成の構文および例は別途示します。

N1 コマンドまたは N2 コマンドの場合、フォーマットバッファに指定されない非 NU ディスクリプタには、次のルールが適用されます。

- フォーマットバッファに指定しなかった非 NU ディスクリプタの中で、フォーマットバッファに指定したフィールドよりフィールド定義テーブル (FDT) 上で手前に定義されているフィールドについては、ディスクリプタのインバーテッドリストに空値がエントリされます。
- フォーマットバッファに指定しなかった非 NU ディスクリプタフィールド内で、フォーマットバッファに指定したフィールドよりフィールド定義テーブル (FDT) 上で後ろに定義されているフィールドについては、ディスクリプタのインバーテッドリストに空値はエントリされません。

そのため、フォーマットバッファでは非 NU ディスクリプタフィールドをすべて参照するか、または (FDT 順の) すべての非 NU ディスクリプタより後ろにあるフィールドを 1 つ以上指定してください。これにより、非 NU ディスクリプタのインバーテッドリストに、空値が正しくエントリされます。

ピリオディックグループ内の非 NU ディスクリプタについては、フォーマットバッファに指定された最高オカレンス番号の前に位置する空値オカレンスについてのみインバーテッドリストに空値がエントリされます。

N1 コマンドまたは N2 コマンドを使用するとき、次のフォーマットバッファに関する事項を考慮しなければなりません。

1. サブディスクリプタ、スーパーディスクリプタ、ハイパーディスクリプタ、およびフォネティックディスクリプタはフォーマットバッファに指定できません。このようなディスクリプタの親フィールドをフォーマットバッファに指定した場合、Adabas は自動的にこれらのディスクリプタ値を正しく作成します。
2. 圧縮前の最大レコード長は、理論上は 32,767 バイトですが、実際にはブロックサイズによって制限されます。また、この値は、Adabas セッションの LU パラメータの大きさによっては、それより小さくなります。この場合の最大レコード長は、LU 値 - フォーマットバッファ長 - 108 になります。圧縮後の最大レコード長は、データストレージブロックサイズ - 4 バイトまたは WORK ブロックサイズ - 110 バイトのどちらか小さい方です。
3. フィールドに標準長を超える長さを使用している場合（フィールドを固定ストレージオプションで定義している場合には許可されません）、このフィールドを後で参照するときは、ここで使用した長さを指定します。後でそれを標準長で参照すると、英数字フィールドでは値の桁落ちが発生し、数値フィールドではゼロ以外のレスポンスコードが返されます。
4. マルチプルバリュースフィールドだけはフォーマットバッファに複数指定できます。
5. フォーマットバッファに指定したマルチプルバリュースカウントフィールド、ピリオディックグループカウントフィールド、またはリテラル値は無視されます。レコードバッファ内の対応値も無視されます。
6. マルチプルバリュースフィールドをフォーマットバッファに指定する場合、Adabas は次のルールに従って、マルチプルバリュースカウントフィールドを設定します。
 - NU オプション付きで定義したマルチプルバリュースフィールドに対するカウントフィールドは、現在空白でない値の個数を反映するように調整されます。空白値は完全に削除されます。

フィールド定義	01,MF,5,A,MU,NU
フォーマットバッファ	MF1-3
レコードバッファ	XXXXXYYYYYZZZZZ
追加後の結果	XXXXX,YYYYY,ZZZZZ MF カウント = 3
フォーマットバッファ	MF1-3
レコードバッファ	XXXXXbbbbbbZZZZZ
追加後の結果	XXXXX,ZZZZZ MF カウント = 2
フォーマットバッファ	MF1-3
レコードバッファ	bbbbbbbbbbbbbbbb (空白)
追加後の結果	値は省略されます。 MF カウント = 0

- NU オプションなしで定義したマルチプルバリュースフィールドに対するカウントフィールドは、存在する値（空値も含む）の個数を反映するように調整されます。

フィールド定義	01,MF,5,A,MU
フォーマットバッファ	MF1-3
レコードバッファ	XXXXXXYYYYYbbbb
追加後の結果	XXXXX,YYYYY,b (空白) MF カウント = 3
フォーマットバッファ	MF1
レコードバッファ	bbbbbb (空白)
追加後の結果	b (空白) MF カウント = 1

マルチプルバリュースフィールドでは最大 191 個の値が許可されています。

7. ピリオディックグループまたはピリオディックグループ内のフィールドをフォーマットバッファに指定すると、ピリオディックグループカウントはフォーマットバッファに指定された最高オカレンス番号と同じです。最高オカレンス値で空値が省略される場合は、カウントはそれに応じて調整されます。

フィールド定義	01,GB,PE 02,BA,1,B,DE,NU 02,BB,5,P,NU
フォーマットバッファ	GB1-2.
レコードバッファ	X'08000000500F09000000600F'
追加後の結果	GB (第 1 オカレンス) BA = 8 BB = 500 GB (第 2 オカレンス) BA = 9 BB = 600 GB カウント = 2
フォーマットバッファ	GB1-2.
レコードバッファ	X'00000000000F0000000000F'
追加後の結果	GB (第 1 オカレンス) 値は省略 GB (第 2 オカレンス) 値は省略 GB カウント = 0

ピリオディックグループでは、最大 191 個のオカレンスが許可されています。

8. 8.可変長（標準長がない）に定義されたフィールドをフォーマットバッファに指定する場合、対応するレコードバッファ内の値の前に、値の長さを示す1バイトの2進数（長さバイトを含む）を指定する必要があります。

フィールド定義	01,AA,3,A 01,AB,A
フォーマットバッファ	AA,AB.
レコードバッファ	X'F1F2F306F1F2F3F4F5'

フィールド AA と AB が更新されます。AA の値は 123 です。AB（可変長フィールド）の値は 12345 です。

レコードバッファ

フォーマットバッファに指定した各フィールドの値をこのレコードバッファに指定する必要があります。

各値は、フォーマットバッファに標準長以外の長さやフォーマットを指定する場合を除いて、フィールドの標準長および標準フォーマットに従って指定する必要があります。

フィールドが可変長フィールド（標準長なし）として定義されている場合、フィールドの長さ（長さバイトも含む）を持つ1バイトのバイナリフィールドを値の直前に指定する必要があります。

値の入るフィールドがユニークディスクリプタとして定義されている場合、指定した値がすでにディスクリプタに存在しない必要があります。存在すると、コマンドは拒否されます。

38 OP コマンド：ユーザーセッションのオープン

■ 機能および使用	386
■ ユーザータイプ	387
■ ACB インターフェイスダイレクトコール：OP コマンド	388
■ ACBX インターフェイスダイレクトコール：OP コマンド	395
■ バッファ	401
■ ユーザーキューエレメント	403
■ タイムリミットの超過	404
■ コントロールブロックフィールドに返される値	404

OP コマンドは、ユーザーセッションの開始を示します。

このchapterでは、次のトピックについて説明します。

機能および使用

どのユーザーも Adabas セッションの開始には OP コマンドを使用することをお勧めします。

OP コマンドは、次のいずれかがユーザーに該当する場合は必須となります。

- ADARUN パラメータ OPENRQ=YES を指定してニュークリアスを実行します。
- 排他ファイル制御 (EXF) での更新を行います。
- 先の ET コマンドで Adabas システムファイルに格納したユーザーデータを読み込みます。
- C3、CL、ETのいずれかのコマンドを使用して、ユーザーデータを Adabas システムファイルに格納します。
- ユーザーに特別な処理優先権を割り当てます。
- ユーザーはアクセスオンリーユーザーです (更新コマンドは許可されません)。
- ADARUN パラメータ TT または TNAX のそれぞれの指定とは異なるトランザクションタイムリミット、または非アクティビティタイムリミットをユーザーに対して設定します。このユーザー設定は、ADARUN パラメータ MXTT および MXTNA でそれぞれ設定した最大値 (2 バイト) 以下となります。
- ユーザーセッションに特殊なデータエンコードまたは構成を指定します。

上記以外では、OP コマンドはオプションです。現在 Adabas に認識されていないユーザーが最初に Adabas コマンドを発行すると、Adabas は内部的に OP コマンドを発行します。

Adabas Security で保護されたファイルにアクセスするユーザーは、必ずしも OP コマンドを発行する必要はありませんが、そのセキュリティ保護されたファイルを呼び出すコマンドにはすべてパスワードを指定する必要があります。システムが Adabas 外部セキュリティインターフェイス (ADAESI)、および付属のセキュリティパッケージで保護されている場合には、OP コマンドが必要になることがあります。詳細については、DBA またはシステムセキュリティ担当者にお問い合わせください。

OP コマンドがアクティブな ET ロジックユーザーから発行され、かつそのユーザーが ET 状態でない (1つ以上のレコードがホールド状態で OP、ET、BT コマンドが前に発行されなかった) 場合、Adabas はユーザーに対し内部的に BT コマンドを発行し、OP コマンドに対しレスポンスコード 9 を返します。OP コマンドが他のアクティブなユーザーから発行される場合、Adabas は OP コマンド処理の前に内部的に CL コマンドを発行します。

シングルユーザーモードで操作するユーザーは、1回のセッションで OP コマンドを一度のみ発行できます。

ユーザータイプ

各ユーザーセッションは、ユーザーの行うアクセスおよび更新のタイプで Adabas により次のように識別されます。

- アクセスオンリーユーザー
- 排他制御ユーザー
- ET ロジックユーザー

アクセスオンリーユーザー

アクセスオンリー（ACC パラメータ）を指定して OP コマンドを発行すると、ユーザーはアクセスオンリーユーザーと定義されます。このようなユーザーは、レコードをホールド、更新、削除、追加するコマンド、ET コマンド、および BT コマンドを発行できません。



Note: アクセス／更新レベルのセキュリティも、ファイルおよびフィールドに対しては Adabas Security で制御でき、データベースおよびファイルに対しては Adabas SAF Security で制御できます。セキュリティ制御は、ユーザータイプの制御を補完するものです。つまり、Adabas Security によってアクセスオンリーユーザーのアクセスレベルをファイル、フィールド、値にも定義できるようになります。ただし、アクセスレベルを更新レベルに変更することはできません。

排他制御ユーザー

排他ファイル制御（EXU または EXF パラメータ）を指定して OP コマンドを発行すると、ユーザーは排他制御ユーザーと定義されます。このユーザーは非 ET ロジックユーザーであるとみなされます（ただし、UPD パラメータも指定すると、このユーザーは ET ロジックユーザーと定義されます）。このユーザーが ET コマンドを発行すると、最初の ET コマンドが発行された時点で ET ロジックユーザーに変わります。

ET ロジックユーザー

その他の全ユーザー（OP コマンドを発行しないユーザーも含みます）は、ET ロジックユーザーと定義されます。このユーザーの発行したトランザクションには、トランザクションタイムリミットが適用されます。

ACB インターフェイスダイレクトコール：OP コマンド

このsectionでは、OPコマンドのACBインターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの情報](#)
- [コントロールブロックフィールドの説明](#)
- [ACB の例](#)

コントロールブロックとバッファの情報

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	--	--	--
コマンドコード	3~4	英数字	F	U
コマンド ID	5~8	英数字	--	A
ファイル番号 **	9~10	バイナリ	F	U
レスポンスコード	11~12	バイナリ	--	A
ISN	13~16	バイナリ	--	A
ISN 下限	17~20	バイナリ	F	A
ISN 数	21~24	バイナリ	F	A
	25~26	--	--	--
レコードバッファ長	27~28	バイナリ	F	U
	29~34	--	--	--
コマンドオプション 1	35	英数字	F	U
コマンドオプション 2	36	英数字	F	U
アディション 1	37~44	英数字	F	U
アディション 2	45~48	英数字	--	A
	49~56	--	--	--
アディション 4	57~64	バイナリ	F	A
アディション 5	65~72	バイナリ	--	A
コマンドタイム	73~76	バイナリ	--	A
ユーザーエリア	77~80	--	--	U

バッファエリア

バッファ	Adabas コール前	Adabas コール後
フォーマット	*	
レコード	F	A

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

* 使用しませんが、コールステートメントのパラメータリストに含める必要があります。

** データベースIDは、複数のデータベースにアクセスして、特定のデータベースに対してコマンド処理を制限する必要がある場合にのみ必要になります。

-- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

コマンドコード (ACBCMD)

OP

コマンド ID (ACBCID)

Adabas は、このユーザーの先行セッションが CL コマンドで正常終了したか、先行セッションがない場合にバイナリのゼロを返します。

このユーザーの先行セッションが CL コマンドで正常終了しなかった場合は、最後に正常終了したユーザートランザクションのシーケンス番号を返します。

この情報が返されるのは、ET ロジックユーザーに限ります。ユーザーが ET ロジックユーザーでない場合、Adabas による変更は行われません。

ファイル番号 (ACBFNR)

データベース ID は、複数のデータベースにアクセスして、特定のデータベースに対してコマンド処理を制限する必要がある場合にのみ必要になります。

レスポンスコード (ACBRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、アディクション 2 フィールドの下位 2 バイトにサブコードを伴う場合があります。詳細は『Adabas メッセージおよびコード』を参照してください。

ISN (ACBISN)

このフィールドにはバイナリのゼロが返されます。

ISN 下限：非アクティビティタイムリミット (ACBISL)

このフィールドは、ユーザーが非アクティビティタイムリミットを指定するために使用します。このタイムリミットは、ADARUN パラメータ MXTNA で指定した最大値に従う必要があります。このフィールドにバイナリのゼロが指定されている場合、Adabas セッションの ADARUN パラメータ (TNAA、TNAE、TNAX) に設定されている、非アクティビティタイムリミットが有効になります。

OP コマンドが正常終了した後、Adabas はこのフィールドに Adabas システム情報およびコールタイプ情報を返します。このフィールドに表示されていたタイムアウト情報は、アディクション5フィールドのバイト4~5に返されます。詳細は、「[コントロールブロックフィールドに返される値](#)」を参照してください。

ISN 数：トランザクションタイムリミット (ACBISQ)

ユーザーがトランザクションタイムリミットを指定するために使用します。このタイムリミットは、ADARUN パラメータ MXTT で指定した最大値に従う必要があります。このフィールドにバイナリのゼロが指定されている場合、Adabas セッションの ADARUN TT パラメータに設定されているトランザクションタイムリミットが有効になります。

OP コマンドが正常終了した後、Adabas はこのフィールドに Adabas システムリリース情報を返します。このフィールドに表示されていたタイムアウト情報は、アディクション5フィールドのバイト6~7に返されます。詳細は、「[コントロールブロックフィールドに返される値](#)」を参照してください。

レコードバッファ長 (ACBRBL)

レコードバッファ長をこのフィールドに指定しなければなりません。指定する長さは、要求された全レコードバッファエントリを格納するのに十分な長さである必要があります。レコードバッファが空の場合は、レコードバッファ長を0に設定する必要があります。

Adabas システムファイルに格納されたユーザーデータを取り出す場合、指定する長さはレコードバッファにユーザーデータを挿入できるだけの十分な長さである必要があります。そうでない場合は、ユーザーデータの桁落ちが発生します。

コマンドオプション 1：ファイル制限オプション (ACBCOP1)

オプション	説明
R (ファイル制限)	ユーザーによるアクセス/更新を、レコードバッファに指定したファイルリストのファイルだけに制限します。指定したファイルが使用できない場合、レスポンスコード 48 が返されます。ユーザーがファイルリストにないファイルへのアクセスや更新を試行すると、レスポンスコード 17 が返されます。R オプションを指定せずに、該当のファイルリストにないファイルにアクセスしようとする、Adabas は Adabas ユーティリティによってそのファイルが使用されているかどうかを調べます。使用されていない場合は、ファイルがユーザーのリストに追加されます。詳細は「 ユーザーキューエレメント 」を参照してください。

コマンドオプション 2：ユーザーデータ読み込み（ACBCOP2）

オプション	説明
E	ユーザーデータのあるユーザーによって発行されたコマンドのうち、最後に正常終了したC3、CL、ETコマンドによってAdabasシステムファイルに保存されたユーザー（ET）データが、レコードバッファに返されます。このオプションは、このユーザーセッションとユーザーデータを保存したセッションで同一のユーザーIDを指定した場合に限り使用できます。

アディション 1：ユーザー ID（ACBADD1）

このフィールドは、ユーザーセッションにユーザー ID を指定する場合に使用します。後で制限を受けることがないように、常にユーザー ID を指定するようにしてください。

ユーザー ID の値は、このユーザーに対してユニークでなければなりません（他のユーザーによって同時に使用されないもの）。また、A から 9 の値で始まる必要があります。この値がユニークでない場合は、レスポンスコード 48 が返されます。非アクティブな状態が 60 秒以上継続している場合は、該当ユーザー ID を持つユーザーに対して内部的な自動バックアウトがスケジュールされ、OP コマンドにはレスポンスコード 9 が返されます。OP コマンドを再度発行すると、既存ユーザーのユーザー ID（ETID）を引き継ぐことができます。このとき、既存ユーザーは、内部的なバックアウトトランザクション（BT）コマンドによってユーザー ID を失い、次のコマンド発行時にレスポンスコード 9 を受け取ります。

次のいずれかの場合は、ユーザー ID を指定しなければなりません。

- 初めてユーザーデータの読み込み／保存を行う場合。後続のユーザーセッションやAdabasセッションでデータを使用可能にする場合。

ユーザー ID（ETID）を指定するユーザーが ET データを保存し、後続セッションでも使用できるようにする場合。ユーザー ID を指定しない場合、そのユーザーが保存した ET データは、現行セッションでのみ有効となります。保存対象のデータは、ET コマンドを発行して指定します。また、クローズ（CL）コマンドを発行する場合には、セッションの終了時に指定できます。前回のセッションでユーザー ID を指定した状態で保存した ET データは、OP コマンドまたは RE コマンドのいずれかで読み込むことができます。

- ユーザーに特別な優先権を割り当てる場合（優先権は Adabas Online System、または ADADBS ユーティリティの PRIORITY 機能で割り当てます）。
- マルチクライアントファイルに対する処理を開始する場合。マルチクライアントファイルの詳細は、『Adabas DBA リファレンスマニュアル』を参照してください。ユーザーのオーナー ID が空白の場合や指定されていない場合、マルチクライアントファイルにアクセスすると、レスポンスコード 3 または 113 が返されます。

上記以外の場合は、このフィールドを空白にできます。

アディション 2：トランザクションシーケンス番号（ACBADD2）

このフィールドには、最後に正常終了した ET コマンド（ET データ付き）のトランザクションシーケンス番号を返します。

OP コマンドによってゼロ以外のレスポンスコードが返された場合、下位 2 バイトには、正確なレスポンスコードの意味を定義するサブコードが含まれていることがあります。レスポンスコードとサブコードについては、『Adabas メッセージおよびコード』を参照してください。

アディクション 4：最大値の設定 (ACBADD4)

このフィールドには、次のユーザー固有の最大バイナリ値を設定することができます。

バイト	使用方法
57～58	Sx コマンドの実行後、内部 ISN エlement テーブルに格納可能な ISN の最大数。デフォルト値を増やすと Adabas ワークに対するアクセス回数が少なくなります。最大値は 1000 です。
59～60	ユーザーが同時にホールドできる最大レコード数。デフォルトは、ADARUN NISNHQ パラメータによって設定される値です。最大値は、ADARUN NH パラメータによって設定される値から 1 を引いた値の 1/4、または 65535 のいずれか小さい方です。
61～62	ユーザーに対して同時にアクティブにできるコマンド ID の最大数。この値は LQ の 1/240 以下でなければなりません (LQ は、ADARUN シーケンシャルコマンドテーブル長パラメータ値です。デフォルトは 10000)。
63～64	Sx コマンドの実行に使用可能な最長時間。

上記のいずれの値も指定しない場合には、システム全体のニュークリアスの値はデフォルトに設定されます。ユーザーは、このフィールドにユーザー固有の値を指定する前に、これらの値の実際のシステムデフォルト値について DBA に確認しておく必要があります。

値はバイナリで指定しなければなりません。空白またはバイナリのゼロを指定した場合、無指定とみなされます。

Adabas はコマンド処理中にアディクション 4 フィールドを空白に設定し、このフィールドの右端 (下位) 3 バイトにバージョンコードおよびデータベース ID を返します。詳細については「[コントロールブロックフィールド](#)」を参照してください。

アディクション 5：返されるタイムアウト値 (ACBADD5)

ユーザーが設定したタイムアウトの値は、アディクション 5 フィールドの右端のフルワードの左半分 (上位) および右半分 (下位) にそれぞれ返されます。詳細は、「[コントロールブロックフィールドに返される値](#)」を参照してください。

ACB の例

- 例 1：アクセスオンリーユーザー
- 例 2：ET ロジックユーザー
- 例 3：非 ET ロジックの排他制御ユーザー
- 例 4：ET ロジックでの排他制御ユーザー
- 例 5：ワイド文字フィールドのエンコード

例 1：アクセスオンリーユーザー

アクセスオンリーユーザーセッションをオープンします。

コントロールブロック

コマンドコード	OP	
レコードバッファ長	4	またはそれ以上

バッファエリア

レコードバッファ	ACC. (または ACC=. または ACC=file-list)
----------	------------------------------------

選択したすべてのファイルにアクセスできます。

例 2：ET ロジックユーザー

ユーザーセッションをオープンし、そのセッションでファイル 8 と 9 にアクセスし、ファイル 8 と 16 を更新します。また、セッション中にユーザーデータを Adabas システムファイルに格納します。前のセッションで格納したユーザーデータも読み込みます。このユーザーのユーザー ID は USER0001 です。

コントロールブロック

コマンドコード	OP	
レコードバッファ長	15	またはそれ以上
コマンドオプション 2	E	ユーザーデータを読み込みます。
アディション 1	USER0001	ユーザーデータを格納したり読み込んだりする場合には、ユーザー ID が必要です。

バッファエリア

レコードバッファ ACC=9,UPD=8,16.

例 3：非 ET ロジックの排他制御ユーザー

ファイル 10、11 および 12 を排他制御で扱うユーザーセッションをオープンします。ユーザーは ET コマンドを発行しません。Adabas システムファイルに対して、ユーザーデータを格納することも、読み込むこともありません。

コントロールブロック

コマンドコード	OP	
レコードバッファ長	13	またはそれ以上
コマンドオプション 2	b (空白)	ユーザーデータの保存および読み込みは実行されません
アティション 1	bbbbbbbb (空白)	ユーザー ID は必要ありません

バッファエリア

レコードバッファ EXU=10,11,12.

例 4：ET ロジックでの排他制御ユーザー

ファイル 10、11 および 12 を排他制御で扱うユーザーセッションをオープンします。ユーザーは ET コマンドを発行します。

コントロールブロック

コマンドコード	OP	
レコードバッファ長	26	またはそれ以上
コマンドオプション 2	b (空白)	ユーザーデータの保存および読み込みは実行されません
アティション 1	bbbbbbbb (空白)	ユーザー ID は必要ありません

バッファエリア

レコードバッファ EXU=10,11,12,UPD=10,11,12.

例 5：ワイド文字フィールドのエンコード

ワイド文字フィールドをシフト JIS にエンコードするユーザーセッションをオープンします。ユーザーはファイル番号 1 を更新します。

コントロールブロック

コマンドコード	OP	
レコードバッファ長	16	またはそれ以上

バッファエリア

レコードバッファ	UPD=1,WCODE=932.
----------	------------------

ACBX インターフェイスダイレクトコール：OP コマンド

このsectionでは、OP コマンドの ACBX インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの情報](#)
- [コントロールブロックフィールドの説明](#)

コントロールブロックとバッファの情報

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	---	---	---
バージョンインジケータ	3~4	バイナリ	F	U
	5~6	---	---	---
コマンドコード	7~8	英数字	F	U
	9~10	---	---	---
レスポンスコード	11~12	バイナリ	---	A
コマンド ID	13~16	英数字/バイナリ	---	A
データベース ID**	17~20	数値	F	U
	21~28	---	---	---
ISN	29~32	バイナリ	---	A

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	33～36	---	---	---
ISN 下限	37～40	バイナリ	F	A
	41～44	---	---	---
ISN 数	45～48	バイナリ	F	A
コマンドオプション 1	49	英数字	F	U
コマンドオプション 2	50	英数字	F	U
	51～56	---	---	---
アディクション 1	57～64	英数字／バイナリ	F	U
アディクション 2	65～68	バイナリ	---	A
	69～76	---	---	---
アディクション 4	77～84	英数字	F	A
アディクション 5	85～92	英数字／バイナリ	---	A
	93～114	---	---	---
エラーサブコード	115～116	バイナリ	---	A
	117～144	---	---	---
コマンドタイム	145～152	バイナリ	---	A
ユーザーエリア	153～168	該当なし	---	U
---	169～193	操作不可	---	---

ABD とバッファ

ABD とバッファ	Adabas コール前	Adabas コール後
フォーマット	*	
レコード	F	A

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

* 使用しませんが、Adabas コールに含める必要があります。含めない場合は、Adabas コールが自動的に生成されます。

** データベースIDは、複数のデータベースにアクセスして、特定のデータベースに対してコマンド処理を制限する必要がある場合にのみ必要になります。

-- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

バージョンインジケータ (ACBXVER)

F2

コマンドコード (ACBXCMD)

OP

レスポンスコード (ACBXRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、エラーサブコード (ACBXERRC) フィールドにサブコードを伴う場合があります。詳細は、Adabas メッセージおよびコードのドキュメントを参照してください。

コマンド ID (ACBXCID)

Adabas は、このユーザーの先行セッションが CL コマンドで正常終了したか、先行セッションがない場合にバイナリのゼロを返します。

このユーザーの先行セッションが CL コマンドで正常終了しなかった場合は、最後に正常終了したユーザートランザクションのシーケンス番号を返します。

この情報が返されるのは、ET ロジックユーザーに限ります。ユーザーが ET ロジックユーザーでない場合、Adabas による変更は行われません。

データベース ID (ACBXDBID)

このフィールドを使用して、データベース ID を指定します。Adabas コールはこのデータベースに送られます。

このフィールドは 4 バイトのバイナリフィールドですが、現時点では 2 バイトのデータベース ID のみがサポートされています。したがって、データベース ID はフィールドの下位部分 (右端のバイト) に指定し、その前にバイナリの 0 を付加する必要があります。

このフィールドがバイナリの 0 に設定されている場合は、Adabas API は DDCARD 入力データで指定された ADARUN カードのデータベース ID か、リンクされているカリンクルーチンによりロードされた LNKGBLS モジュールで指定されているデフォルトのデータベース ID 値のいずれかを使用します。

ISN (ACBXISN)

このフィールドにはバイナリのゼロが返されます。

ISN 下限：非アクティビティタイムリミット (ACBXISL)

このフィールドは、ユーザーが非アクティビティタイムリミットを指定するために使用します。このタイムリミットは、ADARUN パラメータ MXTNA で指定した最大値に従う必要があります。このフィールドにバイナリのゼロが指定されている場合、Adabas セッションの ADARUN パラメータ (TNAA、TNAE、TNAX) に設定されている、非アクティビティタイムリミットが有効になります。

ACBXISL フィールドは、8 バイトの ACBXISLG フィールドに埋め込まれた 4 バイトのバイナリフィールドで、まだ使用されていません。ACBXISLG フィールドの上位部分には、バイナリの 0 を設定します。

OP コマンドが正常終了した後、Adabas はこのフィールドに Adabas システム情報およびコールタイプ情報を返します。このフィールドに表示されていたタイムアウト情報は、アディクション 5 フィールドのバイト 4~5 に返されます。詳細は、「[コントロールブロックフィールドに返される値](#)」を参照してください。

ISN 数：トランザクションタイムリミット (ACBXISQ)

ユーザーがトランザクションタイムリミットを指定するために使用します。このタイムリミットは、ADARUN パラメータ MXTT で指定した最大値に従う必要があります。このフィールドにバイナリのゼロが指定されている場合、Adabas セッションの ADARUN TT パラメータに設定されているトランザクションタイムリミットが有効になります。

ACBXISQ フィールドは、8 バイトの ACBXISQG フィールドに埋め込まれた 4 バイトのバイナリフィールドで、まだ使用されていません。ACBXISQG フィールドの上位部分には、バイナリの 0 を設定します。

OP コマンドが正常終了した後、Adabas はこのフィールドに Adabas システムリリース情報を返します。このフィールドに表示されていたタイムアウト情報は、アディクション 5 フィールドのバイト 6~7 に返されます。詳細は、「[コントロールブロックフィールドに返される値](#)」を参照してください。

コマンドオプション 1：ファイル制限オプション (ACBXCOP1)

オプション	説明
R (ファイル制限)	ユーザーによるアクセス/更新を、レコードバッファに指定したファイルリストのファイルだけに制限します。指定したファイルが使用できない場合、レスポンスコード 48 が返されます。ユーザーがファイルリストにないファイルへのアクセスや更新を試行すると、レスポンスコード 17 が返されます。R オプションを指定せずに、該当のファイルリストにないファイルにアクセスしようとする、Adabas は Adabas ユーティリティによってそのファイルが使用されているかどうかを調べます。使用されていない場合は、ファイルがユーザーのリストに追加されます。詳細は「 ユーザーキューエレメント 」を参照してください。

コマンドオプション 2：ユーザーデータ読み込み (ACBXCOP2)

オプション	説明
E	ユーザーデータのあるユーザーによって発行されたコマンドのうち、最後に正常終了した C3、CL、ET コマンドによって Adabas システムファイルに保存されたユーザー (ET) データが、レコードバッファに返されます。このオプションは、このユーザーセッションとユーザーデータを保存したセッションで同一のユーザー ID を指定した場合に限り使用できます。

アディクション1：ユーザー ID (ACBXADD1)

このフィールドは、ユーザーセッションにユーザー ID を指定する場合に使用します。後で制限を受けることがないように、常にユーザー ID を指定するようにしてください。

ユーザー ID の値は、このユーザーに対してユニークでなければなりません（他のユーザーによって同時に使用されないもの）。また、A から 9 の値で始まる必要があります。この値がユニークでない場合は、レスポンスコード 48 が返されます。非アクティブな状態が 60 秒以上継続している場合は、該当ユーザー ID を持つユーザーに対して内部的な自動バックアウトがスケジュールされ、OP コマンドにはレスポンスコード 9 が返されます。OP コマンドを再度発行すると、既存ユーザーのユーザー ID (ETID) を引き継ぐことができます。このとき、既存ユーザーは、内部的なバックアウトトランザクション (BT) コマンドによってユーザー ID を失い、次のコマンド発行時にレスポンスコード 9 を受け取ります。

次のいずれかの場合は、ユーザー ID を指定しなければなりません。

- 初めてユーザーデータの読み込み／保存を行う場合。後続のユーザーセッションや Adabas セッションでデータを使用可能にする場合。

ユーザー ID (ETID) を指定するユーザーが ET データを保存し、後続セッションでも使用できるようにする場合。ユーザー ID を指定しない場合、そのユーザーが保存した ET データは、現行セッションでのみ有効となります。保存対象のデータは、ET コマンドを発行して指定します。また、クローズ (CL) コマンドを発行する場合には、セッションの終了時に指定できます。前回のセッションでユーザー ID を指定した状態で保存した ET データは、OP コマンドまたは RE コマンドのいずれかで読み込むことができます。

- ユーザーに特別な優先権を割り当てる場合（優先権は Adabas Online System、または ADADBS ユーティリティの PRIORITY 機能で割り当てます）。
- マルチクライアントファイルに対する処理を開始する場合。マルチクライアントファイルの詳細は、『Adabas DBA リファレンスマニュアル』を参照してください。ユーザーのオーナー ID が空白の場合や指定されていない場合、マルチクライアントファイルにアクセスすると、レスポンスコード 3 または 113 が返されます。

上記以外の場合は、このフィールドを空白にできます。

アディクション2：トランザクションシーケンス番号 (ACBXADD2)

このフィールドには、最後に正常終了した ET コマンド (ET データ付き) のトランザクションシーケンス番号を返します。

アディクション 4：最大値の設定（ACBXADD4）

このフィールドには、次のユーザー固有の最大バイナリ値を設定することができます。

バイト	使用方法
77～78	Sx コマンドの実行後、内部 ISN エlement テーブルに格納可能な ISN の最大数。デフォルト値を増やすと Adabas ワークに対するアクセス回数が少なくなります。最大値は 1000 です。
79～80	ユーザーが同時にホールドできる最大レコード数。デフォルトは、ADARUN NISNHQ パラメータによって設定される値です。最大値は、ADARUN NH パラメータによって設定される値から 1 を引いた値の 1/4、または 65535 のいずれか小さい方です。
81～82	ユーザーに対して同時にアクティブにできるコマンド ID の最大数。この値は LQ の 1/240 以下でなければなりません（LQ は、ADARUN シーケンシャルコマンドテーブル長パラメータ値です。デフォルトは 10000）。
83～84	Sx コマンドの実行に使用可能な最長時間。

上記のいずれの値も指定しない場合には、システム全体のニュークリアスの値はデフォルトに設定されます。ユーザーは、このフィールドにユーザー固有の値を指定する前に、これらの値の実際のシステムデフォルト値について DBA に確認しておく必要があります。

値はバイナリで指定しなければなりません。空白またはバイナリのゼロを指定した場合、無指定とみなされます。

Adabas はコマンド処理中にアディクション 4 フィールドを空白に設定し、このフィールドの右端（下位）3 バイトにバージョンコードおよびデータベース ID を返します。詳細については「[コントロールブロックフィールド](#)」を参照してください。

アディクション 5：返されるタイムアウト値（ACBXADD5）

ユーザーが設定したタイムアウトの値は、アディクション 5 フィールドの右端のフルワードの左半分（上位）および右半分（下位）にそれぞれ返されます。詳細は、「[コントロールブロックフィールドに返される値](#)」を参照してください。

エラーサブコード（ACBXERRC）

コマンドがゼロ以外のレスポンスコードを返したときは、このフィールドにレスポンスコードの正確な意味を定義したサブコードが含まれます。レスポンスコードとサブコードについては、『*Adabas メッセージおよびコード*』を参照してください。

バッファ

OP コマンドには、次のバッファが適用されます。

- フォーマットバッファ
- レコードバッファ

フォーマットバッファ

フォーマットバッファは OP コマンドでは使用されませんが、Adabas コールに含める必要があります。これが **ACB インターフェイスダイレクトコール** であり、フォーマットバッファが指定されていない場合は、処理エラーが発生します。ACB インターフェイスダイレクトコールはセットシーケンスにバッファが指定されていることを要求します。これが **ACBX インターフェイスダイレクトコール** であり、フォーマットバッファが指定されていない場合は、バッファが自動的に生成されます。

レコードバッファ

レコードバッファで、以下を指定します。

- アクセスまたは更新するファイルと更新の種類
- セッションでの英数字フィールドまたはワイド文字フィールドのエンコード方法
- レコードバッファとバリュースタックのフィールドの構成。この構成は、Entire Net-Work で設定したりリモートコールに対する構成を書き変えます。

このレコードバッファの構文は次のとおりです。

```
[ { keyword [ = file-list ] } , _ _ ]  
[ACODE = alpha-key ]  
[WCOde = wchar-key ]  
[ARC = architecture-key ].
```

上記の意味は次に示すとおりです。


- *keyword* には次のいずれかが入ります。

ACC ESS ファイルにアクセスするのみです。

UPD ATE ファイルを更新します (ET ロジックを意味します)。

EXF 排他ファイル制御。他のユーザーはファイルにアクセスすることも更新することもできません。

EXU ファイルをユーザーの排他制御の下で更新します。他のユーザーは、このユーザーセッションがアクティブである間はファイルを更新できません。排他制御が与えられるのは、他のアクティブなユーザーがそのファイルに EXF/EXU または UPD パラメータを指定して OP コマンドを発行していない場合に限られます。

- "file-list" には、先行するファイルリストキーワードが適用可能な Adabas ファイルを示す 1～5 桁のファイル番号 (先頭のゼロ可) が入ります。
- ACODE は、ユーザーセッション中、A フィールド用の特別なエンコードを割り当てます。
- *alpha-key* には、指定したエンコードディスクリプタオブジェクトのキーが入ります。
- WCODE は、ユーザーセッション中、W フィールド用の特別なエンコードを割り当てます。
- *wchar-key* には、指定したエンコードディスクリプタオブジェクトのキーが入ります。
- ARC は、レコードバッファおよびバリュースタックのフィールド用に特別なデータアーキテクチャを定義します。この定義は、Entire Net-Work でリモートコール用に定義されているアーキテクチャキーよりも優先されます。
-  **Note:** ARC 設定は、Adabas コントロールブロック (ACB) およびサーチバッファやフォーマットバッファなどレイアウトが固定されているバッファに対して ADALNK/LNKUES が実行するデータ変換に影響を与えません。
- *architecture-key* は、次の値を合計した数値になります。

バイト順	b=0	上位バイト 1 番目
	b=1	下位バイト 1 番目
エンコードファミリ	e=0	ASCII エンコードファミリ
	e=2	EBCDIC エンコードファミリ (ローカルコールのデフォルト)。
浮動小数点フォーマット	f=0	IBM370 浮動小数点フォーマット
	f=4	VAX 浮動小数点フォーマット
	f=8	IEEE 浮動小数点フォーマット

デフォルトは $ARC = b + e + f = 2$ です。つまり、上位バイト 1 番目、EBCDIC エンコードファミリ、および、IBM370 浮動小数点フォーマット ($b=0$ 、 $e=2$ 、 $f=0$) です。


Intel386 PC のユーザーデータの場合は、例えば $b=1$ 、 $e=0$ 、 $f=8$ 、または $ARC=9$ のようになります。

レコードバッファは、ピリオド (".") で終わる必要があります。

レコードバッファにパラメータを指定しないと、ピリオド (".") のみが含まれることとなります。この場合、Adabas コントロールブロックでレコードバッファ長をゼロに設定できるため、レコードバッファを用意する必要はありません。

ユーザータイプのキーワードを一連のファイルに適用する場合、そのキーワードを適用する各ファイルはコンマで区切ってファイル番号を指定します。1つのキーワード内に重複するファイル番号があってもかまいません。別のキーワードにファイル番号が重複してもかまいません。各キーワードは1回だけ指定できます。

UPD および EXU はファイルのアクセスも含んでいます。ACC が唯一指定したキーワードである場合、ファイルリストは必要ありません。

 **Note:** Natural Lightstorm (NLS) version 3.1 のユーザーの場合、OP コマンドでは、NLS がオプション R (ファイル制限) を指定してファイルリスト内のファイルを Adabas に通知するため、レコードバッファにファイルリストを入力する必要があります。

キーワードを指定しないと、ユーザーは自動的に ET ロジックユーザーとなります。

ユーザーキューエレメント

ユーザーがアクティブの間、Adabasはそのユーザーに対するユーザーキューエレメント (UQE) を保持します。

- ユーザーの種類とファイルリスト
- エンコード情報

ユーザーの種類とファイルリスト

UQE はユーザーが現在使用中のファイルを示す最大 5000 までのファイル番号リストを持ちます。ファイルリストは、非ETユーザーがOPコマンドを発行したときに生成され、ETユーザーの場合には作成されません。ユーザーセッション中に、ファイルリストが変更されることもあります。OP コマンドを発行しないと、ファイルリストには当初何のファイルも記載されません。ファイルリスト内の各ファイルは、次のいずれかに区分されます。

- ACC、アクセスのみ
- EXF/EXU、アクセスおよび更新 (排他的制御下)
- UPD、アクセスおよび更新
- UTI、アクセスおよび更新 (Adabas ユーティリティにより使用)

後で現在ユーザーファイルリストにないファイルにアクセスしようとする、そのファイルが現在 Adabas ユーティリティで使用中かどうかチェックされます。使用中でないと、そのファイルがユーザーの UQE に追加され、ACC とマークされます。ただし、OP コマンドで、レコードバッファ内のファイルリストでコマンドオプション1フィールドにファイル制限オプションを

指定した場合には、ファイルリストにないファイルにアクセスしようとする、レスポンスコード 17 が返されます。

後で現在ユーザーファイルリストにないファイルを更新しようとする、次のチェックが行われます。

- その要求はユーザータイプと矛盾しないか?例えば、アクセスオンリーユーザーが更新コマンドを発行することはできません。
- 他のユーザーまたは Adabas ユーティリティが排他制御の下で更新しているファイルか?

ファイルがそのユーザーに使用可能であると判断されると、そのファイルはユーザーの UQE に追加され、UPD とマークされます。

エンコード情報

使用するエンコード方法を決める ACODE、WCODE および（または）ARC をユーザーが指定すると、その情報は Adabas ニュークリアスに伝えられて UQE に格納されます。

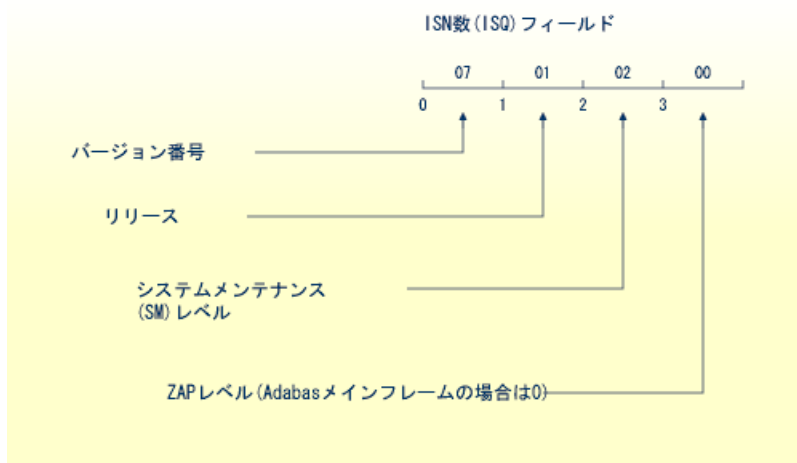
タイムリミットの超過

ユーザーが非アクティビティタイムリミットを超える場合に行う処理については、『Adabas オペレーションマニュアル』のタイムアウト特性に関するセクションを参照してください。

コントロールブロックフィールドに返される値

VMS/UNIX システムとの互換性を保つため、ISN 下限フィールドおよび ISN 数フィールドに値が返されることがあります。その結果、以前の Adabas バージョンでこのフィールドにホールドされたユーザー固有のタイムアウト値がアディション 5 フィールドに返されます。このことは、対応するコマンドログフィールドにも影響します。

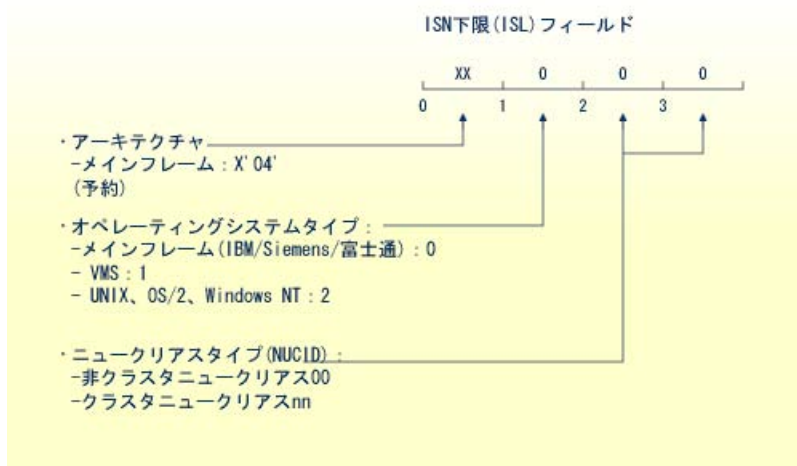
OP の実行後、ISN 数フィールドには次のバイナリの値が入ります。



OP 実行後に返されるバイナリ値

この情報は VMS/UNIX との互換性を確保するためのものです。

ISN 下限 (ISL) フィールドには、次のバイナリの情報が入ります。

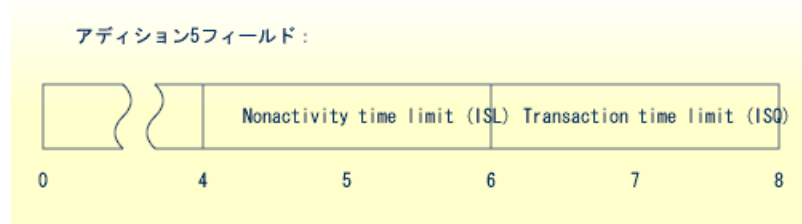


ISLから返されるバイナリ情報

右端2バイトは、ユーザープログラムが非クラスタ (00) またはクラスタ (nn) ニュークリアス環境で実行しているかどうかを示します。特定のクラスタニュークリアスタイプのリターンコードは Software AG 技術サポートが使用します。

OP コマンド：ユーザーセッションのオープン

ユーザーがOPコマンド発行時に設定したISL/ISQフィールドのタイムアウト値がアディクション5フィールドに、次のように入ります。



タイムアウト値

アディクション5コマンドログエントリもこれに対応します。

39 RCコマンド：コマンドID／グローバルフォーマット ID の解放

- 機能および使用 408
- ACB インターフェイスダイレクトコール：RC コマンド 409
- ACBX インターフェイスダイレクトコール：RC コマンド 413

RC コマンドは、このユーザーの1つ以上のコマンド ID または1つのグローバルフォーマット ID を解放します。

このchapterでは、次のトピックについて説明します。

機能および使用

RC コマンドを使用すると、現在ユーザーに割り当てられている1つ以上のコマンド ID を解放したり、1つまたはすべてのグローバルフォーマット ID を削除したりできます。

- 内部フォーマットバッファプールのコマンド ID。関連する内部フォーマットも解放されません。
- ISN リスト (TBI) のコマンド ID。
- シーケンシャルコマンドテーブル (TBLES/TBQ) のコマンド ID。
- 内部フォーマットバッファプールか、TBI、TBLES、TBQのどちらか、または両方に指定したコマンド ID 値に等しいか、それ以上のコマンド ID。
- ユーザーグループの特別なグローバルフォーマット ID。
- 既存のすべてのグローバルフォーマット ID。

選択オプションを指定しないと、入力したコマンド ID が上記のすべての領域から解放されません。コマンド ID を解放すると、関連する TBI および (または) TBLES/TBQ エントリも削除されます。ただし、内部フォーマットバッファプールエントリは必ずしも解放されません。

RC コマンドは、次のいずれかの場合に使用します。

- SAVE ISN LIST オプション付きの S_x コマンドを実行して、Adabas ワークに格納されている ISN リストの処理をユーザーが完了した場合。RC コマンドの発行により、Adabas は現在そのリストが占めていたスペースを再使用できます。
- エンドオブファイル条件に達する前に、ファイルの順次読み込み (L2/L5、L3/L6、L9) コマンドをユーザーが終了する場合。
- 空白以外のコマンド ID を使用した一連の L1/L4、A1、N1/N2 コマンドをユーザーが完了した場合。

ACB インターフェイスダイレクトコール：RC コマンド

このsectionでは、RCコマンドのACBインターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの情報](#)
- [コントロールブロックフィールドの説明](#)
- [ACB の例](#)

コントロールブロックとバッファの情報

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	--	--	--
コマンドコード	3~4	英数字	F	U
コマンド ID	5~8	英数字	F	U
ファイル番号	9~10	バイナリ	F	U
レスポンスコード	11~12	バイナリ	--	A
	13~34	--	--	--
コマンドオプション 1	35	英数字	F	U
コマンドオプション 2	36	英数字	F	U
アディション 1	37~44	英数字	F	U
	45~64	--	--	--
アディション 5	65~72	英数字	F	U
コマンドタイム	73~76	バイナリ	--	A
ユーザーエリア	77~80	--	--	U

バッファエリア

何も使用されていません。

上記の意味は次に示すとおりです。

- F Adabas コール前にユーザーが入力するフィールド
- A Adabas により入力されるフィールド
- U Adabas コール後も変化なし
- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

コマンドコード (ACBCMD)

RC

コマンド ID (ACBCID)

このフィールドには、解放または参照するコマンド ID を指定します。空白またはバイナリのゼロを指定すると、ユーザーに現在割り当てられている全コマンド ID が解放されます。

ファイル番号 (ACBFNR)

コマンドオプション 1 フィールドを D、E、または O に指定する場合は、解放するフォーマットまたはグローバルフォーマット ID と関連するファイルの番号を 2 進数で指定しなければなりません。

物理ダイレクトコールの場合は、次のようファイル番号を指定します。

- 1 バイトファイル番号の場合は、ファイル番号を第 2 バイト (10) に入力します。第 1 バイト (9) は、バイナリの 0 (B'0000 0000') をセットします。
- 2 バイトファイル番号の場合は、2 バイト (9 と 10) を使います。



Note: 2 バイトファイル番号およびデータベース ID を使用する場合は、コントロールブロックの先頭バイトに X'30' を入力しなければなりません。

レスポンスコード (ACBRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、アディション 2 フィールドの下位 2 バイトにサブコードを伴う場合があります。詳細は『Adabas メッセージおよびコード』を参照してください。

コマンドオプション 1/2：解放するコマンド ID のタイプ (ACBCOP1 および ACBCOP2)

これらのフィールドは、内部フォーマットバッファプール、ISN リストテーブル (TBI)、またはシーケンシャルコマンドのテーブル (TBLES/TBQ) から解放されるコマンド ID、フォーマット ID、またはグローバルフォーマット ID を示します。テーブルの詳細については、「[プログラミング全般に関する考慮事項](#)」を参照してください。

コマンドオプション 1 および 2 の両フィールドに空白かバイナリのゼロを設定した場合、コマンド ID フィールドに指定されたコマンド ID は、すべてのテーブルから解放されます。

いずれかのコマンドオプションフィールドに次の値を設定した場合には、コマンド ID、フォーマット ID、またはグローバルフォーマット ID に関連付けられているリソースが解放されます。解放される対象は次の表のとおりです。

オプション	解放される対象
C	すべての既存グローバルフォーマット
D	指定したファイル番号およびディスクリプタ名に該当するすべてのフォーマット
E	指定したファイル番号およびディスクリプタ名に該当するすべてのグローバルフォーマット
F	指定したコマンド ID に関連付けられているフォーマット
G	指定したコマンド ID 以上のコマンド ID に関連付けられているすべての既存フォーマット
I	指定したコマンド ID に関連付けられている ISN リスト (TBI)
L	アディクション 5 フィールドに格納されているグローバルフォーマット ID。
O	指定したファイル番号のアディクション 5 フィールドに格納されているグローバルフォーマット ID。
S	指定したコマンド ID に関連付けられているシーケンシャルコマンド (TBLES/TBQ)
X	指定したコマンド ID 以上のコマンド ID に関連付けられているすべての ISN リスト (TBI) およびシーケンシャルコマンド (TBLES/TBQ)。内部フォーマットは解放されません。

Smith/Jones の問題が発生する可能性がある環境で正しいデータが返されるように L3 または L6 コマンドでフォーマットを作成した場合に、オプション D と E を使用します。この場合に基礎となるフォーマット ID は、サイズが 12 バイトになり、8 バイトのフォーマット ID、2 バイトのファイル番号、2 バイトのディスクリプタ名から構成されます。

アディクション 1：ディスクリプタ名 (ACBADD1)

コマンドオプション D または E が指定される場合、アディクション 1 フィールドの最初の 2 バイトには、解放するフォーマットまたはグローバルフォーマット ID に関連する英数字のディスクリプタフィールド名を含んでいる必要があります。残りのすべての位置には空白を設定しなければなりません。

解放するフォーマットが L3 または L6 コマンドで作成されていない場合、このフィールドは使用されません。

アディクション 5：解放するグローバルフォーマット ID (ACBADD5)

解放するグローバルフォーマット ID を指定します。

ACB の例

- 例 1
- 例 2
- 例 3
- 例 4
- 例 5

例 1

コマンド ID 0003 を解放します。

コントロールブロック

コマンドコード	RC	
コマンド ID	X'0003'	コマンド ID 003 を解放します。
コマンドオプション 1/2	bb (空白)	全コマンド ID タイプを解放します

例 2

現在ユーザーに割り当てられている全コマンド ID を解放します。

コントロールブロック

コマンドコード	RC	
コマンド ID	X'00000000'	バイナリのゼロは全コマンド ID を解放することを示します
コマンドオプション 1/2	bb (空白)	全コマンド ID タイプを解放します

例 3

シーケンシャルコマンドテーブルまたは内部フォーマットバッファプールにあって、ユーザーに割り当てられているコマンド ID をすべて解放します。

コントロールブロック

コマンドコード	RC	
コマンド ID	X'00000000'	バイナリのゼロは全コマンド ID を解放することを示します
コマンドオプション 1	F	F は、内部フォーマットバッファプールにあるコマンド ID を解放することを示します。
コマンドオプション 2	S	S は、シーケンシャルコマンドテーブルにあるコマンド ID を解放することを示します。

例 4

複数のファイルに同じグローバルフォーマット ID を定義している場合、全ファイルのグローバルフォーマット ID を解放します。

コントロールブロック

コマンドコード	RC	
コマンドオプション 1/2	L	アディクション5フィールドで指定したグローバルフォーマット ID のフォーマットを解放します。
アディクション 5	C'TGLOB001'	この番号の第1バイトの上位（左端）2ビットに B'11' が設定されている場合、全8バイトがグローバルフォーマット ID であることを示します。

例 5

複数のファイルに同じグローバルフォーマット ID を定義している場合、ファイル3のグローバルフォーマット ID を解放します。

コントロールブロック

コマンドコード	RC	
ファイル番号	03	グローバルフォーマット ID を解放するファイルの番号を2進数で指定します。
コマンドオプション 1/2	O	ファイル番号フィールドで指定したファイルに関して、アディクション5フィールドで指定したグローバルフォーマット ID のフォーマットを解放します。
アディクション 5	C'TGLOB001'	この番号の第1バイトの上位（左端）2ビットに B'11' が設定されている場合、全8バイトがグローバルフォーマット ID であることを示します。

ACBX インターフェイスダイレクトコール：RC コマンド

このsectionでは、RC コマンドの ACBX インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの情報](#)

■ コントロールブロックフィールドの説明

コントロールブロックとバッファの情報

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1～2	---	---	---
バージョンインジケータ	3～4	バイナリ	F	U
	5～6	---	---	---
コマンドコード	7～8	英数字	F	U
	9～10	---	---	---
レスポンスコード	11～12	バイナリ	---	A
コマンド ID	13～16	英数字／バイナリ	F	U
データベース ID	17～20	数値	F	U
ファイル番号	21～24	数値	F	U
	25～48	---	---	---
コマンドオプション 1	49	英数字	F	U
コマンドオプション 2	50	英数字	F	U
	51～56	---	---	---
アディション 1	57～64	英数字／バイナリ	F	U
	65～84		---	---
アディション 5	85～92	英数字／バイナリ	F	U
	93～114	---	---	---
エラーサブコード	115～116	バイナリ	---	A
	117～144	---	---	---
コマンドタイム	145～152	バイナリ	---	A
ユーザーエリア	153～168	該当なし	---	U
---	169～193	操作不可	---	---

ABD とバッファ

何も使用されていません。

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

-- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

バージョンインジケータ (ACBXVER)

F2

コマンドコード (ACBXCMD)

RC

レスポンスコード (ACBXRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、エラーサブコード (ACBXERRC) フィールドにサブコードを伴う場合があります。詳細は、*Adabas* メッセージおよびコードのドキュメントを参照してください。

コマンド ID (ACBXCID)

このフィールドには、解放または参照するコマンド ID を指定します。空白またはバイナリのゼロを指定すると、ユーザーに現在割り当てられている全コマンド ID が解放されます。

データベース ID (ACBXDBID)

このフィールドを使用して、データベース ID を指定します。Adabas コールはこのデータベースに送られます。

このフィールドは 4 バイトのバイナリフィールドですが、現時点では 2 バイトのデータベース ID のみがサポートされています。したがって、データベース ID はフィールドの下位部分 (右端のバイト) に指定し、その前にバイナリの 0 を付加する必要があります。

このフィールドがバイナリの 0 に設定されている場合は、Adabas API は DDCARD 入力データで指定された ADARUN カードのデータベース ID か、リンクされているかリンクルーチンによりロードされた LNKGBLS モジュールで指定されているデフォルトのデータベース ID 値のいずれかを使用します。

ファイル番号 (ACBXFNR)

コマンドオプション 1 フィールドを D、E、または O に指定する場合は、このフィールドを使用して、解放するフォーマットまたはグローバルフォーマット ID に関連付けられたファイルの番号を指定する必要があります。

このフィールドは 4 バイトのバイナリフィールドですが、ファイル番号はフィールドの下位部分 (右端のバイト) に指定し、その前にバイナリの 0 を付加する必要があります。

コマンドオプション 1 および 2：解放するコマンド ID のタイプ (ACBXCOP1 および ACBXCOP2)

これらのフィールドは、内部フォーマットバッファプール、ISN リストテーブル (TBI)、またはシーケンシャルコマンドのテーブル (TBLES/TBQ) から解放されるコマンド ID、フォーマット ID、またはグローバルフォーマット ID を示します。テーブルの詳細については、「[プログラミング全般に関する考慮事項](#)」を参照してください。

コマンドオプション 1 および 2 の両フィールドに空白かバイナリのゼロを設定した場合、コマンド ID フィールドに指定されたコマンド ID は、すべてのテーブルから解放されます。

いずれかのコマンドオプションフィールドに次の値を設定した場合には、コマンド ID、フォーマット ID、またはグローバルフォーマット ID に関連付けられているリソースが解放されます。解放される対象は次の表のとおりです。

オプション	解放される対象
C	すべての既存グローバルフォーマット
D	指定したファイル番号およびディスクリプタ名に該当するすべてのフォーマット
E	指定したファイル番号およびディスクリプタ名に該当するすべてのグローバルフォーマット
F	指定したコマンド ID に関連付けられているフォーマット
G	指定したコマンド ID 以上のコマンド ID に関連付けられているすべての既存フォーマット
I	指定したコマンド ID に関連付けられている ISN リスト (TBI)
L	アディクション 5 フィールドに格納されているグローバルフォーマット ID。
O	指定したファイル番号のアディクション 5 フィールドに格納されているグローバルフォーマット ID。
S	指定したコマンド ID に関連付けられているシーケンシャルコマンド (TBLES/TBQ)
X	指定したコマンド ID 以上のコマンド ID に関連付けられているすべての ISN リスト (TBI) およびシーケンシャルコマンド (TBLES/TBQ)。内部フォーマットは解放されません。

Smith/Jones の問題が発生する可能性がある環境で正しいデータが返されるように L3 または L6 コマンドでフォーマットを作成した場合に、オプション D と E を使用します。この場合に基礎となるフォーマット ID は、サイズが 12 バイトになり、8 バイトのフォーマット ID、2 バイトのファイル番号、2 バイトのディスクリプタ名から構成されます。

アディクション 1：ディスクリプタ名 (ACBXADD1)

コマンドオプション D または E が指定される場合、アディクション 1 フィールドの最初の 2 バイトには、解放するフォーマットまたはグローバルフォーマット ID に関連する英数字のディスクリプタフィールド名を含んでいる必要があります。残りのすべての位置には空白を設定しなければなりません。

解放するフォーマットが L3 または L6 コマンドで作成されていない場合、このフィールドは使用されません。

アディション 5：解放するグローバルフォーマット ID (ACBXADD5)

解放するグローバルフォーマット ID を指定します。

エラーサブコード (ACBXERRC)

コマンドがゼロ以外のレスポンスコードを返したときは、このフィールドにレスポンスコードの正確な意味を定義したサブコードが含まれます。レスポンスコードとサブコードについては、『*Adabas* メッセージおよびコード』を参照してください。

40 RE コマンド：ET ユーザーデータの読み込み

▪ 機能および使用	420
▪ ACB インターフェイスダイレクトコール：RE コマンド	420
▪ ACBX インターフェイスダイレクトコール：RE コマンド	425
▪ バッファ	428

RE コマンドは、現在のユーザー、他のユーザー、または全ユーザーの ET (ユーザー) データを読み込みます。

このchapterでは、次のトピックについて説明します。

機能および使用

RE コマンドは、前に C3、CL または ET コマンドで Adabas システムチェックポイントファイルに格納されたユーザーデータを読み込むのに使用します。ユーザーデータはレコードバッファに返されます。このユーザーデータは、ユーザーまたは Adabas セッションの異常終了後、ユーザー再スタートをするのに必要です。

コマンドオプション1フィールドに何も指定しないと、このコマンドを発行したユーザーのユーザーデータが読み込まれます。OP コマンドで同じユーザー ID を指定して、前セッションと現在のセッションを開始した場合、RE コマンドは前セッションで格納したユーザーデータを読み込みます。

コマンドオプションの指定に応じて、RE コマンドで他のユーザーのユーザーデータ (各ユーザー ID を指定する)、または全ユーザーのユーザーデータを読み込むことができます。

- コマンドオプション1フィールドに "T" を指定した場合、アディション1フィールドにデータを格納したユーザーのユーザー ID が指定されていると、そのユーザーのユーザーデータが読み込まれます。
- コマンドオプション1フィールドに "A" を指定した場合、現在および後続の RE コマンドは全ユーザーデータをユーザー ID の論理順に読み込みます。また、RE コマンドが完了するごとに、ユーザー ID がアディション1フィールドに返されます。オプション "A" を指定した RE コマンドは、チェックポイントファイルに書き込まれたユーザーデータのうち、トランザクションが ET コマンドで終了しているもののみを読み込みます。ET コマンドで終了していないトランザクションのユーザーデータは読み込みません。

ACB インターフェイスダイレクトコール：RE コマンド

このsectionでは、RE コマンドの ACB インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの情報](#)
- [コントロールブロックフィールドの説明](#)

■ ACB の例

コントロールブロックとバッファの情報

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1～2	--	--	--
コマンドコード	3～4	英数字	F	U
コマンド ID	5～8	バイナリ	--	A
ファイル番号 ****	9～10	バイナリ	F	U
レスポンスコード	11～12	バイナリ	--	A
ISN	13～16	バイナリ	F	A
	17～26	--	--	--
レコードバッファ長	27～28	バイナリ	F	U
	29～34	--	--	--
コマンドオプション 1	35	英数字	F	U
	36	--	--	--
アディション 1	37～44	英数字	F *	U/A **
アディション 2	45～48	英数字	--	A
	49～72	--	--	--
コマンドタイム	73～76	バイナリ	--	A
ユーザーエリア	77～80	--	--	U

バッファエリア

バッファ	Adabas コール前	Adabas コール後
フォーマット	***	--
レコード	--	A

上記の意味は次に示すとおりです。

- F Adabas コール前にユーザーが入力するフィールド
- A Adabas により入力されるフィールド
- U Adabas コール後も変化なし
- * コマンドオプション1が "I" の場合、ET データのユーザー ID
- ** コマンドオプション1が "A" の場合、レコードバッファの ET データのユーザー ID
- *** 使用しませんが、コールステートメントのパラメータリストに含める必要があります。
- **** データベース ID は、複数のデータベースにアクセスして、特定のデータベースに対してコマンド処理を制限する必要がある場合にのみ必要になります。
- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

コマンドコード (ACBCMD)

RE

コマンド ID (ACBCID)

Adabas は、トランザクションのシーケンス番号か、バイナリのゼロを返します。

Adabas は、ユーザーデータを読み込むユーザーアクティブでなく、かつこのユーザーの先行セッションがないか、CL コマンドで正常終了した場合にバイナリのゼロを返します。非 ET ロジックユーザーの場合もバイナリのゼロが返されます。

このユーザーが現在アクティブであるか、またはこのユーザーの先行セッションが CL コマンドで正常終了しなかった場合、Adabas は、最後に正常終了したユーザートランザクションのシーケンス番号を返します。

ファイル番号 (ACBFNR)

データベース ID は、複数のデータベースにアクセスして、特定のデータベースに対してコマンド処理を制限する必要がある場合にのみ必要になります。

レスポンスコード (ACBRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。コマンドオプション1に A を指定した場合、レスポンスコード3はユーザーデータのエンドオブファイルを示します。ゼロ以外のレスポンスコードには、アディション2フィールドの右半分にサブコードが返される場合があります。レスポンスコード、サブコードについては『Adabas メッセージおよびコードマニュアル』を参照してください。

ISN (ACBISN)

コマンドオプション1が A の場合、ISN から開始して論理順にユーザーデータが返されます。このフィールドにゼロを指定すると、全ユーザーデータが返されます。

レコードバッファ長 (ACBRBL)

レコードバッファの長さ。指定した長さによって、返されるユーザーデータのバイト数が決まります。

指定した長さがユーザーデータよりも短いと、指定したバイト数のみがレコードバッファに入り、右側のバイトが桁落ちします。

コマンドオプション1 (ACBCOP1)

コマンドオプション1フィールドに何も指定しない場合は、コマンドを発行したユーザーのユーザーデータが読み込まれます。OPコマンドで同じユーザーIDを指定して、前セッションと現在のセッションを開始した場合、REコマンドは前セッションで格納したユーザーデータを読み込みます。

コマンドオプション1フィールドに値を指定すると、その値に応じて、別の指定ユーザーのユーザーデータか、全ユーザーのユーザーデータを読み込みます。

オプション	説明
I (ユーザーのID)	データを格納したユーザーのIDがアディクション1フィールドに指定されている場合は、別のユーザーが保存したユーザーデータを読み込みます。
A (すべてのユーザー)	現在および後続のREコマンドは、レコードバッファ内の全ユーザーデータをユーザーIDの論理順 (ISN 順) に読み込みます。ISN フィールドに開始 ISN を指定できます。ISN フィールドにゼロを指定すると、全ユーザーデータが返されます。現在および後続のREコマンドが完了するごとに、コマンド終了時にレコードバッファ内にあるユーザーデータのユーザーIDがアディクション1フィールドに返されます。オプションAを指定したREコマンドは、チェックポイントファイルに書き込まれたユーザーデータのうち、トランザクションがETコマンドで終了しているものだけを読み込みます。ETコマンドで終了していないトランザクションのユーザーデータは読み込まれません。このオプションでは、レスポンスコードフィールドに返されるレスポンスコード3は、ユーザーデータのエンドオブファイルを示します。

アディクション1：ユーザーID (ACBADD1)

他のユーザーが格納したユーザーデータを読み込む場合、このフィールドにはデータを格納したユーザーのユーザーIDを指定しなければなりません。コマンドオプション1にAを指定した場合、このフィールドには各REコマンドの終了時に、レコードバッファ内にあるユーザーデータのユーザーIDが入ります。

アディクション2：トランザクションシーケンス番号 (ACBADD2)

データを保存したユーザーがETロジックユーザーであった場合、Adabasはこのフィールドに、ETまたはCLコマンドでユーザーデータを保存した最後の正常終了トランザクションのシーケンス番号を返します。

REコマンドによってゼロ以外のレスポンスコードが返された場合、アディクション2フィールドの下位2バイトにレスポンスコードの正確な意味を定義したサブコードが含まれていることがあります。レスポンスコードとサブコードについては、『Adabas メッセージおよびコード』を参照してください。

ACB の例

- 例 1
- 例 2
- 例 3

例 1

以前に ET コマンドで格納したユーザーデータを読み込みます。

コントロールブロック

コマンドコード	RE	
レコードバッファ長	100	ユーザーデータの 100 バイトを読み込みます。
コマンドオプション 1	空白	このユーザーが格納したユーザーデータを読み込みます。
ISN	0	

例 2

他のユーザー（ユーザー ID = USER0002）が格納したユーザーデータを読み込みます。

コントロールブロック

コマンドコード	RE	
レコードバッファ長	150	ユーザーデータの 150 バイトを読み込みます。
コマンドオプション 1	I	他のユーザーが格納したユーザーデータを読み込みます。
アディション 1	USER0002	ユーザーデータを格納したユーザーのユーザー ID
ISN	0	

例 3

全ユーザーデータおよび対応するユーザー ID を読み込みます。

コントロールブロック

コマンドコード	RE	
レコードバッファ長	250	ユーザーデータの 250 バイトを読み込みます。
コマンドオプション 1	A	
ISN	0	全ユーザーデータを読み込みます。

ACBX インターフェイスダイレクトコール：RE コマンド

このsectionでは、RE コマンドの ACBX インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの情報](#)
- [コントロールブロックフィールドの説明](#)

コントロールブロックとバッファの情報

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	---	---	---
バージョンインジケータ	3~4	バイナリ	F	U
	5~6	---	---	---
コマンドコード	7~8	英数字	F	U
	9~10	---	---	---
レスポンスコード	11~12	バイナリ	---	A
コマンド ID	13~16	英数字/バイナリ	---	A
データベース ID****	17~20	数値	F	U
ファイル番号	21~24	数値	F	U
	25~28	---	---	---
ISN	29~32	バイナリ	F	A
	33~48	---	---	---
コマンドオプション 1	49	英数字	F	U
	50~56	---	---	---
アディション 1	57~64	英数字/バイナリ	F*	U/A **
アディション 2	65~68	バイナリ	---	A
	69~114	---	---	---
エラーサブコード	115~116	バイナリ	---	A
	117~144	---	---	---
コマンドタイム	145~152	バイナリ	---	A
ユーザーエリア	153~168	該当なし	---	U
---	169~193	操作不可	---	---

ABD とバッファ

ABD とバッファ	Adabas コール前	Adabas コール後
フォーマット	***	--
レコード	--	A

上記の意味は次に示すとおりです。

- F Adabas コール前にユーザーが入力するフィールド
- A Adabas により入力されるフィールド
- U Adabas コール後も変化なし
- * コマンドオプション 1 が "I" の場合、ET データのユーザー ID
- ** コマンドオプション 1 が "A" の場合、レコードバッファの ET データのユーザー ID
- *** 使用しませんが、Adabas コールに含める必要があります。含めない場合は、Adabas コールが自動的に生成されます。
- **** データベース ID は、複数のデータベースにアクセスして、特定のデータベースに対してコマンド処理を制限する必要がある場合にのみ必要になります。
- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

バージョンインジケータ (ACBXVER)

F2

コマンドコード (ACBXCMD)

RE

レスポンスコード (ACBXRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、エラーサブコード (ACBXERRC) フィールドにサブコードを伴う場合があります。詳細は、*Adabas* メッセージおよびコードのドキュメントを参照してください。

コマンドオプション 1 に A を指定した場合、レスポンスコード 3 はユーザーデータのエンドオブファイルを示します。

コマンド ID (ACBXCID)

Adabas は、トランザクションのシーケンス番号か、バイナリのゼロを返します。

Adabas は、ユーザーデータを読み込むユーザーアクティブでなく、かつこのユーザーの先行セッションがないか、CL コマンドで正常終了した場合にバイナリのゼロを返します。非 ET ロジックユーザーの場合もバイナリのゼロが返されます。

このユーザーが現在アクティブであるか、またはこのユーザーの先行セッションが CL コマンドで正常終了しなかった場合、Adabas は、最後に正常終了したユーザートランザクションのシーケンス番号を返します。

データベース ID (ACBXDBID)

このフィールドは、複数のデータベースにアクセスして、特定のデータベースに対するコマンド処理を制限する必要がある場合にのみデータベース ID を指定します。Adabas コールはこのデータベースに送られます。

このフィールドは 4 バイトのバイナリフィールドですが、現時点では 2 バイトのデータベース ID のみがサポートされています。したがって、データベース ID はフィールドの下位部分（右端のバイト）に指定し、その前にバイナリの 0 を付加する必要があります。

このフィールドがバイナリの 0 に設定されている場合は、Adabas API は DDCARD 入力データで指定された ADARUN カードのデータベース ID か、リンクされているカリンクルーチンによりロードされた LNKGBLS モジュールで指定されているデフォルトのデータベース ID 値のいずれかを使用します。

ファイル番号 (ACBXFNR)

このフィールドを使用して、Adabas コールで送る必要のあるファイル数を指定します。

このフィールドは 4 バイトのバイナリフィールドですが、ファイル番号はフィールドの下位部分（右端のバイト）に指定し、その前にバイナリの 0 を付加する必要があります。

ISN (ACBXISN)

コマンドオプション 1 が "A" の場合、ISN から開始して論理順にユーザーデータが返されます。このフィールドにゼロを指定すると、全ユーザーデータが返されます。

ACBXISN フィールドは、8 バイトの ACBXISNG フィールドに埋め込まれた 4 バイトのバイナリフィールドで、まだ使用されていません。ACBXISNG フィールドの上位部分には、バイナリの 0 を設定します。

コマンドオプション 1 (ACBXCOP1)

コマンドオプション 1 フィールドに何も指定しない場合は、コマンドを発行したユーザーのユーザーデータが読み込まれます。OP コマンドで同じユーザー ID を指定して、前セッションと現在のセッションを開始した場合、RE コマンドは前セッションで格納したユーザーデータを読み込みます。

コマンドオプション 1 フィールドに値を指定すると、その値に応じて、別の指定ユーザーのユーザーデータか、全ユーザーのユーザーデータを読み込みます。

オプション	説明
I (ユーザーの ID)	データを格納したユーザーの ID がアディション 1 フィールドに指定されている場合は、別のユーザーが保存したユーザーデータを読み込みます。
A (すべてのユーザー)	現在および後続の RE コマンドは、レコードバッファ内の全ユーザーデータをユーザー ID の論理順 (ISN 順) に読み込みます。ISN フィールドに開始 ISN を指定できます。ISN フィールドにゼロを指定すると、全ユーザーデータが返されます。現在および後続の RE コマンドが完了するごとに、コマンド終了時にレコードバッファ内にあるユーザーデータのユーザー ID がアディション 1 フィールドに返されます。オプション A を指定した RE コマンドは、チェックポイントファイルに書き込まれたユーザーデータのうち、トランザクションが ET コマンドで終了しているものだけを読み込みます。ET コマンドで終了していないトランザクションのユーザーデータは読み込まれません。このオプションでは、レスポンスコードフィールドに返されるレスポンスコード 3 は、ユーザーデータのエンドオブファイルを示します。

アディション 1：ユーザー ID (ACBXADD1)

コマンドオプション 1 に "I" を指定した場合、このフィールドにはデータを格納したユーザーのユーザー ID を設定する必要があります。

コマンドオプション 1 に "A" を指定した場合、このフィールドには各 RE コマンドの終了時に、レコードバッファ内にあるユーザーデータのユーザー ID が入ります。

アディション 2：トランザクションシーケンス番号 (ACBXADD2)

データを保存したユーザーが ET ロジックユーザーであった場合、Adabas はこのフィールドに、ET または CL コマンドでユーザーデータを保存した最後の正常終了トランザクションのシーケンス番号を返します。

エラーサブコード (ACBXERRC)

コマンドがゼロ以外のレスポンスコードを返したときは、このフィールドにレスポンスコードの正確な意味を定義したサブコードが含まれます。レスポンスコードとサブコードについては、『Adabas メッセージおよびコード』を参照してください。

バッファ

RE コマンドには、次のバッファが適用されます。

- フォーマットバッファ

■ レコードバッファ

フォーマットバッファ

フォーマットバッファは RE コマンドでは使用されませんが、Adabas コールに含める必要があります。これが **ACB インターフェイスダイレクトコール** であり、フォーマットバッファが指定されていない場合は、処理エラーが発生します。ACB インターフェイスダイレクトコールはセットシーケンスにバッファが指定されていることを要求します。これが **ACBX インターフェイスダイレクトコール** であり、フォーマットバッファが指定されていない場合は、バッファが自動的に生成されます。

レコードバッファ

Adabas は、レコードバッファにユーザーデータを返します。ユーザーデータが存在しない場合、RE 操作の最後でレコードバッファに空白がつけられます。

41 RI コマンド：レコードの解放

- 機能および使用 432
- ACB インターフェイスダイレクトコール：RI コマンド 432
- ACBX インターフェイスダイレクトコール：RI コマンド 434

RI コマンドは、ホールドされているレコードと ISN を解放します。

このchapterでは、次のトピックについて説明します。

機能および使用

RI コマンドは、コマンドを発行したユーザーがホールドしているレコードの ISN を解放します。1つのデータベースファイルに関して選択した ISN、または発行したユーザーがホールドしている全ファイル内の全 ISN を解放できます。

Adabas コントロールブロックフィールドに解放するレコードの ISN とファイル番号をユーザーが指定します。ISN フィールドにゼロを指定すると、全ファイルに関して現在このユーザーにホールドされている全レコードを解放します。



Note: ET ロジックユーザーが現在のトランザクションで更新を行っている場合は、RI コマンドでレコードのホールドを解放しないでください。解放すると、データ保全性が失われることがあります。ET ユーザーは ET または CL コマンドで ISN を解放しなければなりません。

ACB インターフェイスダイレクトコール：RI コマンド

このsectionでは、RI コマンドの ACB インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの情報](#)
- [コントロールブロックフィールドの説明](#)
- [ACB の例](#)

コントロールブロックとバッファの情報

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	--	--	--
コマンドコード	3~4	英数字	F	U
	5~8	--	--	--
ファイル番号	9~10	バイナリ	F	U
レスポンスコード	11~12	バイナリ	--	A
ISN	13~16	バイナリ	F	U
	17~72	--	--	--
コマンドタイム	73~76	バイナリ	--	A
ユーザーエリア	77~80	--	--	U

バッファエリア

何も使用されていません。

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

-- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

コマンドコード (ACBCMD)

RI

ファイル番号 (ACBFNR)

解放するレコードがあるファイルの番号。



Note: 2 バイトファイル番号およびデータベース ID を使用する場合は、コントロールブロックの先頭バイトに X'30' を入力しなければなりません。

レスポンスコード (ACBRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、アディショナル 2 フィールドの下位 2 バイトにサブコードを伴う場合があります。詳細は『Adabas メッセージおよびコード』を参照してください。

ISN (ACBISN)

解放するレコードの ISN。ISN 値を入力する場合はファイル番号も指定する必要があります。ユーザーがホールドした全レコードを解放する場合、このフィールドにバイナリのゼロを設定します。

ACB の例

- 例 1
- 例 2

例 1

ファイル 2 の ISN 3 のレコードをホールド状態から解放します。

コントロールブロック

コマンドコード	RI	
ファイル番号	2	解放するレコードはファイル 2 にあります。
ISN	3	ISN が 3 のレコードを解放します。

例 2

発行したユーザーがホールドしているすべてのレコードをホールド状態から解放します。

コントロールブロック

コマンドコード	RI	
ファイル番号	-	ISN フィールドがゼロの場合、このフィールドの値は無視されます。
ISN	0	すべてのファイルからすべてのホールドレコードの ISN を解放します。

ACBX インターフェイスダイレクトコール：RI コマンド

このsectionでは、RIコマンドのACBインターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの情報](#)

■ コントロールブロックフィールドの説明

コントロールブロックとバッファの情報

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	---	---	---
バージョンインジケータ	3~4	バイナリ	F	U
	5~6	---	---	---
コマンドコード	7~8	英数字	F	U
	9~10	---	---	---
レスポンスコード	11~12	バイナリ	---	A
	13~16	---	---	---
データベース ID	17~20	数値	F	U
ファイル番号	21~24	数値	F	U
	25~28	---	---	---
ISN	29~32	バイナリ	F	U
	33~114	---	---	---
エラーサブコード	115~116	バイナリ	---	A
	117~144	---	---	---
コマンドタイム	145~152	バイナリ	---	A
ユーザーエリア	153~168	該当なし	---	U
---	169~193	操作不可	---	---

ABD とバッファ

何も使用されていません。

上記の意味は次に示すとおりです。

- F Adabas コール前にユーザーが入力するフィールド
- A Adabas により入力されるフィールド
- U Adabas コール後も変化なし
- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

バージョンインジケータ (ACBXVER)

F2

コマンドコード (ACBXCMD)

RI

レスポンスコード (ACBXRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、エラーサブコード (ACBXERRC) フィールドにサブコードを伴う場合があります。詳細は、*Adabas* メッセージおよびコードのドキュメントを参照してください。

データベース ID (ACBXDBID)

このフィールドを使用して、データベース ID を指定します。Adabas コールはこのデータベースに送られます。

このフィールドは 4 バイトのバイナリフィールドですが、現時点では 2 バイトのデータベース ID のみがサポートされています。したがって、データベース ID はフィールドの下位部分 (右端のバイト) に指定し、その前にバイナリの 0 を付加する必要があります。

このフィールドがバイナリの 0 に設定されている場合は、Adabas API は DDCARD 入力データで指定された ADARUN カードのデータベース ID か、リンクされているカリンクルーチンによりロードされた LNKGBLS モジュールで指定されているデフォルトのデータベース ID 値のいずれかを使用します。

ファイル番号 (ACBXFNR)

このフィールドは、解放するレコードが含まれているファイルの番号を指定するのに使用します。

このフィールドは 4 バイトのバイナリフィールドですが、ファイル番号はフィールドの下位部分 (右端のバイト) に指定し、その前にバイナリの 0 を付加する必要があります。

ISN (ACBXISN)

解放するレコードの ISN。ISN 値を入力する場合はファイル番号も指定する必要があります。ユーザーがホールドした全レコードを解放する場合、このフィールドにバイナリのゼロを設定します。

ACBXISN フィールドは、8 バイトの ACBXISNG フィールドに埋め込まれた 4 バイトのバイナリフィールドで、まだ使用されていません。ACBXISNG フィールドの上位部分には、バイナリの 0 を設定します。

エラーサブコード (ACBXERRC)

コマンドがゼロ以外のレスポンスコードを返したときは、このフィールドにレスポンスコードの正確な意味を定義したサブコードが含まれます。レスポンスコードとサブコードについては、『*Adabas* メッセージおよびコード』を参照してください。

42 S1、S2、S4 コマンド：レコードの検索

- 機能および使用 440
- ACB インターフェイスダイレクトコール：S1、S2、S4 の各コマンド 441
- ACBX インターフェイスダイレクトコール：S1、S2、S4 の各コマンド 453
- バッファ 459


S1、S2、S4 の各コマンドは、検索条件を満たすレコードの数と ISN を返します。

このchapterでは、次のトピックについて説明します。

機能および使用

S1、S2、S4 の各コマンドは、ある検索条件を満たすレコードを選択します。これらのコマンドは、Adabas 拡張ファイルにも実行できます。


S1、S2、S4 の各コマンドを実行すると、条件を満たすレコードの個数およびそのレコードの ISN リストが得られます。S1/S4 コマンドを使用すると、昇順にソートされた ISN が返されます。S2 コマンドを使用すると、ユーザーがアディション 1 フィールドに指定したソート順で ISN が返されます。ソート順に関係なく、ISN は [ISN バッファ](#) に返されます。

 **Note:** 昇順でない ISN リストは、後続の S8 コマンドでは処理できません。

次の種類の検索が可能です。

- 単一ファイル検索。検索条件は、単一ファイルにある 1 つ以上のフィールドから構成されます。
- 物理的にカップリングされたファイルを使用した複数ファイル検索。この検索条件は、ADAINV ユーティリティで物理的にカップリングした 2 つ以上のファイルに含まれているフィールドから構成されます。
- ソフトカップリング機能を使用した検索。この機能は、検索、読み込み、内部リストのマッチングの組み合わせができます。

検索条件は、ディスクリプタとして定義されていない 1 つ以上のフィールドで構成してもかまいません。非ディスクリプタが使用されていると、Adabas はユーザーに返すレコードを判断するために、読み込み処理を実行します。検索条件内にディスクリプタだけが使用されている場合、Adabas はアソシエータのインバーテッドリストを使ってこの条件を処理するため、読み込み処理は実行しません。

 **Note:** Adabas データベースにおける非ディスクリプタ値の検索は、メインフレームの場合とオープンシステムの場合とで、フィールド内の空値省略に関して違いがあります。オープンシステムではフィールドが空値省略 (NU) の場合、非ディスクリプタ値を検索したときに空値を含むレコードを返しません。メインフレームシステムでは、非ディスクリプタ値を検索したときにフィールドの空値省略 (NU) は無視されます。このときフィールドを非ディスクリプタ値の検索で使用する必要がある場合は、オープンシステムフィールドの空値省略 (NU) オプションを解除してこの問題を解決することをお勧めします。

有効なコマンド ID を指定した場合、Adabas は最初の S1、S2、S4 の各コマンドで ISN バッファに入りきれなかった ISN を Adabas ワークに格納します。オーバーフローした ISN は、同じコマンド ID を指定した S1、S2、S4 の各コマンドで ISN バッファに戻すことができます。

Adabas は、オーバーフローした ISN リストの最後にある ISN がユーザーに返されたときにそのリストを解放します。ユーザーが後の処理のために全 ISN リストを保持する必要がある場合は、SAVE ISN LIST オプションを使用できます。このオプションを指定すると、全 ISN リストが Adabas ワーク上に格納されます。この ISN リストは、RC、CL またはリリース CID オプション付き Sx コマンドが発行されるか、または Adabas セッションが終了するまで解放されません。

ユーザーが L1/L4 コマンドの GET NEXT オプションを使用して ISN で識別されるレコードを読み込むときは、ISN バッファエントリも SAVE ISN LIST オプションも必要ありません。この場合、L1/L4 コマンドは、Adabas が格納した ISN リストから自動的に ISN を読み込みます。

フォーマットバッファにフィールド名を入れることにより、ユーザーは ISN 結果リストの最初の ISN レコードからフィールドの内容を読み込むことができます。フィールドの内容は、[レコードバッファ](#)に読み込まれます。S4 コマンドを使用すると、ISN 結果リストの最初の ISN がホールド状態となります。

ACB インターフェイスダイレクトコール：S1、S2、S4 の各コマンド

この section では、S1、S2、S4 の各コマンドの ACB インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの情報](#)
- [コントロールブロックフィールドの説明](#)
- [ACB の例](#)

コントロールブロックとバッファの情報

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	--	--	--
コマンドコード	3~4	英数字	F	U
コマンド ID	5~8	英数字	F	U
ファイル番号	9~10	バイナリ	F	U
レスポンスコード	11~12	バイナリ	--	A
ISN	13~16	バイナリ	--	A
ISN 下限	17~20	バイナリ	F	U
ISN 数	21~24	バイナリ	F*	A

S1、S2、S4 コマンド：レコードの検索

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
フォーマットバッファ長	25~26	バイナリ	F	U
レコードバッファ長	27~28	バイナリ	F	U
サーチバッファ長	29-30	バイナリ	F	U
バリュースバッファ長	31-32	バイナリ	F	U
ISN バッファ長	33~34	バイナリ	F	U
コマンドオプション 1	35	英数字	F	U
コマンドオプション 2	36	英数字	F	U
アディション 1	37~44	英数字	F	U
アディション 2	45~48	バイナリ/バイナリ	--	A
アディション 3	49~56	英数字	F	A
アディション 4	57~64	英数字	F	A
アディション 5	65~72	英数字	F	U
コマンドタイム	73~76	バイナリ	--	A
ユーザーエリア	77~80	--	--	U

バッファエリア

バッファ	Adabas コール前	Adabas コール後
フォーマット	F	U
レコード	--	A
サーチ	F	U
バリュー	F	U
ISN	--	A

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

* オプションのタイムアウト値、秒単位

-- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

コマンドコード (ACBCMD)

S1、S2、S4

コマンド ID (ACBCID)

この値は、Adabas ワークに格納された ISN リストが完了したか、オーバーフローしたかを識別します。また、READ FIRST RECORD オプションが有効な場合、後続のコマンドに対してフォーマットバッファを識別するために使用します（「[フォーマットバッファ](#)」参照）。

SAVE ISN LIST オプションを使用する場合、またはオーバーフローした ISN を格納する場合には、このフィールドに空白やゼロ以外の値を指定する必要があります。

このフィールドの第 1 バイトには 16 進数 'FF' にセットしないでください。


詳細は「[ISN リスト処理](#)」を参照してください。

ファイル番号 (ACBFNR)

ISN を選択するファイルの番号。

カップリングされたファイルを使用する問い合わせを行う場合は、このフィールドに指定したファイルがプライマリファイルとみなされます。物理カップリングファイルのファイル番号は 255 以下でなければなりません。

Adabas 拡張ファイルでも検索できます。

 **Note:** 2 バイトファイル番号およびデータベース ID を使用する場合は、コントロールブロックの先頭バイトに X'30' を入力しなければなりません。

レスポンスコード (ACBRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、アディション 2 フィールドの下位 2 バイトにサブコードを伴う場合があります。詳細は『[Adabas メッセージおよびコード](#)』を参照してください。

ISN (ACBISN)

Adabas はこのフィールドに、ISN 結果リストの先頭 ISN を返します。ISN が 1 件もなかった場合、このフィールドにはゼロが設定されます。

ISN 下限 (ACBISL)

このフィールドを最初の S1 または S4 コールで使用すると、ISN 結果リストの最小 ISN 値を指定できます。したがって、リストには、このフィールドに指定した ISN よりも大きな ISN のみが含まれます。このフィールドをゼロに設定すると、Adabas は該当するすべての ISN を返します。



Note: S2 コールの場合は、このフィールドに指定した値は無効です。ソートされている ISN リスト (S2 コマンド) の場合は、このフィールドにゼロを設定するか、正しい ISN を指定する必要があります。

また、Adabas ワークにセーブした ISN リストから ISN グループを取得するときにも使用します。

ISN 数 (ACBISQ)

このフィールドは、Sx コマンドの実行にかかる最大秒数を定義するのに使用します。

Adabas は、最初の Sx コールの結果として、検索条件を満たすレコードの数をこのフィールドに返します。セキュリティバイバリューを使用する場合、値 0 (1 レコードを検出) または 1 (複数のレコードを検出) とともにレスポンスコード 1 をこのフィールドに返します。詳細は、『*Adabas Security マニュアル*』を参照してください。

また、のちに Sx コールで Adabas ワークから ISN を取得すると、返された ISN の数がこのフィールドに格納されます。ISN 自体は ISN バッファに返されます。

フォーマットバッファ長 (ACBFBL)

フォーマットバッファ長 (バイト単位)。ユーザープログラムに定義するフォーマットバッファエリアは、この指定長と同じか、それ以上でなければなりません。

レコードバッファ長 (ACBRBL)

レコードバッファ長 (バイト単位)。ユーザープログラムに定義するレコードバッファエリアは、この指定長と同じか、それ以上でなければなりません。

サーチバッファ長 (ACBSBL)

サーチバッファ長 (バイト単位)。ユーザープログラムで定義するサーチバッファエリアは、指定されたこの長さ以上である必要があります。

バリューバッファ長 (ACBVBL)

バリューバッファ長 (バイト単位)。ユーザープログラムで定義するバリューバッファエリアは、指定されたこの長さ以上である必要があります。

ISN バッファ長 (ACBIBL)

ISN バッファ長 (バイト単位)。この長さによって ISN バッファに配置される ISN の数が決定します。

このフィールドをゼロにすると、ISN は ISN バッファに追加されません。L1 または L4 コマンドの GETNEXT オプションを指定して ISN 結果リストを読み込む場合、または条件に合ったレコード数を求めるためだけにコマンドを発行する場合、このフィールドはゼロにする必要があります。

ゼロ以外の値を指定する場合は、4 の倍数にする必要があります。そうでないと、Adabas は長さを次に小さい 4 の倍数に調整します。

コマンドオプション 1 (ACBCOP1)

オプション	説明
H (ISN リストの格納)	Sx コマンドを実行して得られた全 ISN リストを、指定したコマンド ID で Adabas ワークに格納します。正しいコマンド ID を指定する必要があります。コマンド ID を指定しないと、ISN リストは Adabas ワークに格納されず、ISN バッファに格納されなかった ISN はすべて失われます。
R (リターン)	S4 コマンドを使用している場合は、読み込んでホールドするレコードが使用不可であれば、レスポンスコード 145 を返します。

コマンドオプション 2 (ACBCOP2)

オプション	説明
D (降順)	S2 コマンドを使用している場合は、ディスクリプタ値を降順でソートします。

S2 コマンドでコマンドオプション 2 を指定しないと、ディスクリプタ値は昇順でソートされます。

コマンドオプション 1/2：CID 解放オプション (ACBCOP1 および ACBCOP2)

コマンドオプション 1 かコマンドオプション 2 フィールドのどちらかに、I オプションを指定できます。

オプション	説明
I	コマンド実行の最初に、コマンド ID フィールドで指定されているコマンド ID (CID) 値を解放します。コマンド ID が解放されるのは ISN リストテーブル内だけなので、同じコマンド ID が ISN 結果リストを識別するのに再度使用されます。

アディション 1：S2 コマンド、ソート制御用ディスクリプタ (ACBADD1)

S2 コマンドを使用している場合は、ソート順を制御するディスクリプタ (複数可) を指定する必要があります。ソート引数の指定がなければレスポンスコード 28 が返されます。

ディスクリプタは 3 つまで指定できます。サブディスクリプタおよびスーパーディスクリプタも指定できます。フォネティックディスクリプタ、またはピリオディックグループ内にあるディスクリプタは指定できません。マルチプルバリュースフィールドは使用できます。この場合、ISN レコード内の最低値に基づいてソートされます。

このフィールド内で使用しない桁は、空白にしておかなければなりません。例

XXYYbbbb

上記の意味は次に示すとおりです。

XX 主ソートディスクリプタ

YY 副ソートディスクリプタ

bbbb 空白

ソートできる ISN の数は、DBA が設定した SORT ワークエリアのサイズ（ADARUN LS パラメータ）に依存します。ソートエリアが小さすぎる場合、ソートは行われず、レスポンスコード 1 が返され、ISN は昇順で返されます。

アディクション 2（圧縮および非圧縮レコード長（ACBADD2））

コマンドが正常に処理されると、次の情報がこのフィールドに返されます。

- 少なくとも 1 つの有効なフィールド値がレコードバッファにある場合、先頭 2 バイトには、アクセスしたレコードの圧縮レコード長がバイナリ形式で格納されます。
- 後ろ 2 バイトには、フォーマットバッファで選択し、アクセスしたフィールドの非圧縮長がバイナリ形式で格納されます。

Sx コマンドによってゼロ以外のレスポンスコードが返された場合、下位 2 バイトにレスポンスコードの正確な意味を定義したサブコードが含まれていることがあります。レスポンスコードとサブコードについては、『*Adabas メッセージおよびコード*』を参照してください。

アディクション 3：パスワード（ACBADD3）

このフィールドは、Adabas Security または ADAESI パスワードを指定するため使用します。データベース、ファイル、またはフィールドがセキュリティ保護されている場合は、有効なセキュリティパスワードまたは ADAESI パスワードをユーザーは指定する必要があります。パスワードを保護するために、コマンド処理時にアディクション 3 フィールドに空白がセットされます。

アディクション 4：サイファコードおよびバージョン／ニュークリアス ID（ACBADD4）

このフィールドは、サイファコードを指定するために使用します。ファイルを暗号化する場合は、有効なサイファコードを指定する必要があります。ファイルを暗号化しない場合は、このフィールドに空白をセットする必要があります。

Adabas はコマンド処理時にサイファコードを空白に設定し、このフィールドの右端（下位）3 バイトにバージョンコードおよびデータベース ID を返します。詳細については「[コントロールブロックフィールド](#)」を参照してください。

アディクション 5：フォーマット ID、グローバルフォーマット ID（ACBADD5）

このコマンド用の内部フォーマットバッファを識別する別のフォーマット ID を指定するか、グローバルフォーマット ID を指定します。

アディクション 5 フィールドの上位ビットにゼロ（0）が設定されている場合、コマンド ID フィールドに指定された値がフォーマット ID として使用されます。

このビットに 1 がセットされていると、アディクション 5 フィールドの 5 バイト目から 8 バイト目までがフォーマット ID として使用されます。

さらに、アディション5フィールドの第1バイトの最上位（左端）2ビットに、1（B'11'）がセットされると、アディション5フィールドの全8バイトが、グローバルフォーマット ID として使用されます（つまり、複数ユーザーがこのフォーマット ID を同時に使用できます）。

詳細は、「[コマンドID、フォーマットID、グローバルフォーマットID](#)」を参照してください。

ACB の例

ここで記述するすべての例で使用されている Adabas ファイルの定義については、「[例で使用されているファイル定義](#)」を参照してください。

- 例 1
- 例 2
- 例 3
- 例 4
- 例 5
- 例 6
- 例 7
- 例 8

例 1

ディスクリプタ AA で、A から J の値を持つファイル 1 内のレコードを選択します。

コントロールブロック

コマンドコード	S1	
コマンド ID	bbbb (空白)	ISN は Adabas ワークに格納されません
ファイル番号	1	
ISN 下限	0	条件に合ったレコードの ISN をすべて返します
フォーマットバッファ長	1	またはそれ以上
サーチバッファ長	12	またはそれ以上
バリューバッファ長	2	またはそれ以上
ISN バッファ長	200	ISN が 50 個を超えないものと想定します。
コマンドオプション 1	b (空白)	SAVE ISN LIST オプションは使用しません
アディション 3	bbbbbbbb (空白)	ファイルはセキュリティ保護されていません。

バッファエリア

S1、S2、S4 コマンド：レコードの検索

フォーマットバッファ	.	読み込みは実行されません
サーチバッファ	AA,1,S,AA,1.	
バリューバッファ	C'AJ'	

例 2

READ オプション付き FIND。ファイル 1 のレコードのうち、フィールド AA が ABCDEFGH であるレコードの ISN を選択します。また、レコードのフィールド AC の値をデータストレージから読み込みます。

コントロールブロック

コマンドコード	S1	
コマンド ID	bbbb (空白)	ISN は Adabas ワークに格納されません
ファイル番号	1	
ISN 下限	0	条件に合ったレコードの ISN をすべて返します
フォーマットバッファ長	3	またはそれ以上
レコードバッファ長	20	またはそれ以上
サーチバッファ長	3	またはそれ以上
バリューバッファ長	8	またはそれ以上
ISN バッファ長	4	ISN の個数が 1 を超えないものと想定します。
コマンドオプション 1	b (空白)	SAVE ISN LIST オプションは使用しません
アデプション 3	bbbbbbbb (空白)	ファイルはセキュリティ保護されていません
アデプション 4	bbbbbbbb (空白)	ファイルは暗号化されない

バッファエリア

フォーマットバッファ	AC.	フィールド AC の値を返します。
サーチバッファ	AA.	
バリューバッファ	C'ABCDEFGH' X'C1C2C3C4C5C6C7C8'	

例 3

ISN バッファオーバーフローのある FIND。ファイル 1 のレコードのうち、フィールド AA の値が A から D の範囲にあるレコードを選択します。ISN バッファオーバーフローを取り扱いません。

コントロールブロック

コマンドコード	S1	
コマンド ID	ABCD	空白でないコマンド ID が必要です。
ファイル番号	1	
ISN 下限	0	条件に合ったレコードの ISN をすべて返します
フォーマットバッファ長	1	またはそれ以上
レコードバッファ長	0	またはそれ以上
サーチバッファ長	12	またはそれ以上
バリューバッファ長	2	またはそれ以上
ISN バッファ長	100	各コールにつき 25 件までの ISN を返します。
コマンドオプション 1	b (空白)	SAVE ISN LIST オプションは使用しません
アディション 3	bbbbbbbb (空白)	ファイルはセキュリティ保護されていません

バッファエリア

フォーマットバッファ	.	読み込みは実行されません
サーチバッファ	AA,1,S,AA,1.	
バリューバッファ	C'AD' X'C1C4'	

Adabas は、初めの S1 コールの結果として ISN バッファに最大 25 個の ISN を返します。25 個以上になると残りの ISN はコマンド ID ABCD の下で Adabas ワークに格納されます。このオーバーフローした ISN は、同じコマンド ID を使用して S1 コールを繰り返せば読み込めます。

例 4

SAVE ISN LIST オプション付き FIND。ファイル 2 のレコードのうち、フィールド XB の値が +80 であるレコードをすべて選択します。ISN 結果リスト全体を Adabas ワークに格納します。

コントロールブロック

S1、S2、S4 コマンド：レコードの検索

コマンドコード	S1	
コマンド ID	BCDE	SAVE ISN LIST オプションを使用する場合は、空白でないコマンド ID が必要です。
ファイル番号	2	
ISN 下限	0	条件に合った ISN をすべて選択します
フォーマットバッファ長	1	またはそれ以上
レコードバッファ長	0	またはそれ以上
サーチバッファ長	3	またはそれ以上
バリューバッファ長	2	またはそれ以上
ISN バッファ長	200	各コールにつき最大 50 個の ISN が返されます。
コマンドオプション 1	H	SAVE ISN LIST オプションを使用します
アディション 3	パスワード	ファイルはセキュリティ保護されています

バッファエリア

フォーマットバッファ	.	読み込みは実行されません
サーチバッファ	XB.	
バリューバッファ	X'080C'	

コマンド ID BCDE の S1 コマンドを繰り返すと、ユーザーは、このコールの結果格納される ISN リストからどの ISN グループも読み込めます。Adabas は、ISN 下限フィールドに指定した ISN の次に大きい ISN からバッファに収まるだけの ISN を挿入します。

例 5

ソート付き FIND。ファイル 1 のレコードのうち、フィールド AA の値が A から F の範囲にあるレコードをすべて選択します。ISN 結果リストをフィールド AB の値の昇順で返します。

コントロールブロック

コマンドコード	S2	
コマンド ID	CDEF	S2 コマンドを使用する場合は、空白でないコマンド ID が必要です。
ファイル番号	1	
ISN 下限	0	条件に合った ISN をすべて選択します
フォーマットバッファ長	1	またはそれ以上
レコードバッファ長	0	またはそれ以上
サーチバッファ長	12	またはそれ以上
バリューバッファ長	2	またはそれ以上
ISN バッファ長	100	各コールにつき最大 25 個の ISN が返されます。

コマンドオプション1	b	SAVE ISN LIST オプションは使用しません。
コマンドオプション2	b	降順ソートオプションは使用しません。
アディション1	ABbbbbbb	実行結果の ISN は、フィールド AB の値でソートされます。 bbbbbb には空白が入ります。
アディション3	bbbbbbbb (空白)	ファイルはセキュリティ保護されていません

バッファエリア

フォーマットバッファ	.	読み込みは実行されません
サーチバッファ	AA,1,S,AA,1.	
バリューバッファ	C'AF' X'C1C6'	

例 6

ホールド付き FIND。ファイル 1 のレコードのうち、フィールド AA の値が 87654321 のレコードを選択します。また、レコードを読み込んでホールド状態にします。フィールド AB と AC の値を返します。

コントロールブロック

コマンドコード	S4	
コマンド ID	bbbb (空白)	SAVE ISN LIST オプションを使用せず、オーバーフロー ISN も想定していないため、空白のコマンド ID を使用できます。
ファイル番号	1	
ISN 下限	0	条件に合った ISN をすべて選択します
フォーマットバッファ長	6	またはそれ以上
レコードバッファ長	22	またはそれ以上
サーチバッファ長	3	またはそれ以上
バリューバッファ長	8	またはそれ以上
ISN バッファ長	4	ただ 1 つの ISN が必要です。
コマンドオプション1	b (空白)	SAVE ISN LIST オプションは使用しません。
アディション3	bbbbbbbb (空白)	ファイルはセキュリティ保護されていません
アディション4	bbbbbbbb (空白)	ファイルは暗号化されない

バッファエリア

S1、S2、S4 コマンド：レコードの検索

フォーマットバッファ	AB,AC.	ISN リストの第 1 の ISN のレコードを読み込み、フィールド AB と AC の値を返します。
サーチバッファ	AA.	
バリュースタック	C'87654321' X'F8F7F6F5F4F3F2F1'	

例 7

カップリングファイルを使用した FIND。フィールド AB の値が 100 であるファイル 1 のレコードのうち、フィールド RB の値が ABCDE のファイル 2 のレコードとカップリングされたレコードを選択します。

コントロールブロック

コマンドコード	S1	
コマンド ID	EFGH	ISN のオーバーフローが発生する可能性があるため、空白でないコマンド ID を使用します。
ファイル番号	1	ファイル 1 がプライマリファイルです。
ISN 下限	0	条件に合った ISN をすべて選択します
フォーマットバッファ長	1	またはそれ以上
レコードバッファ長	0	またはそれ以上
サーチバッファ長	14	またはそれ以上
バリュースタック長	12	またはそれ以上
ISN バッファ長	100	各コールにつき最大 25 個の ISN を返します。
コマンドオプション 1	b	SAVE ISN LIST オプションは使用しません。
アディショナル 3	パスワード	ファイル 2 はセキュリティ保護されています。

バッファエリア

フォーマットバッファ	.	読み込みは実行されません
サーチバッファ	/1/AB,D,/2/RB.	
バリュースタック	X'100CC1C2C3C4C540404040'	

ファイル 1 がプライマリファイルとして指定されているため、ファイル 1 の ISN が返されます。ファイル 2 の ISN も必要な場合は、ファイル番号フィールドをファイル番号 2 と指定し、FIND を繰り返します。サーチバッファの検索条件の順番を変える必要はありません。

例 8

複合検索条件を使用した FIND。サブディスクリプタ SA の値が "ABCD" で、フィールド XB の値が "80" より小さく、フィールド XE の値が "MMMMM" から "ZZZZZ" (ただし、"Sbbbb" から "TZZZZ" までは除く) であるファイル 2 のレコードを選択します。

コントロールブロック

コマンドコード	S1	
コマンド ID	GGGG	SAVE ISN LIST オプションを使用するため、空白でないコマンド ID が必要です。
ファイル番号	2	
ISN 下限	0	条件に合った ISN をすべて選択します
フォーマットバッファ長	1	またはそれ以上
レコードバッファ長	0	またはそれ以上
サーチバッファ長	35	またはそれ以上
バリューバッファ長	27	またはそれ以上
ISN バッファ長	0	ISN バッファに ISN を返しません。
コマンドオプション 1	H	SAVE ISN LIST オプションを使用します
アディション 3	パスワード	ファイル 2 はセキュリティ保護されています。

バッファエリア

フォーマットバッファ	.	読み込みは実行されません
サーチバッファ	SA,D,XB,3,U,LT,D,XE,S,XE,N,XE,S,XE.	
バリューバッファ	C'ABCD080MMMMMZSSbbbbTZZZZ'	

ACBX インターフェイスダイレクトコール：S1、S2、S4 の各コマンド

このsectionでは、S1、S2、S4 の各コマンドの ACBX インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- コントロールブロックとバッファの情報

■ コントロールブロックフィールドの説明

コントロールブロックとバッファの情報

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	---	---	---
バージョンインジケータ	3~4	バイナリ	F	U
	5~6	---	---	---
コマンドコード	7~8	英数字	F	U
	9~10	---	---	---
レスポンスコード	11~12	バイナリ	---	A
コマンド ID	13~16	英数字/バイナリ	F	U
データベース ID	17~20	数値	F	U
ファイル番号	21~24	数値	F	U
	25~28	---	---	---
ISN	29~32	バイナリ	---	A
	33~36	---	---	---
ISN 下限	37~40	バイナリ	F	U
	41~44	---	---	---
ISN 数	45~48	バイナリ	F*	A
コマンドオプション 1	49	英数字	F	U
コマンドオプション 2	50	英数字	F	U
	51~56	---	---	---
アディション 1	57~64	英数字/バイナリ	F	U
アディション 2	65~68	バイナリ	---	A
アディション 3	69~76	英数字/バイナリ	F	A
アディション 4	77~84	英数字	F	A
アディション 5	85~92	英数字/バイナリ	F	U
	93~114	---	---	---
エラーサブコード	115~116	バイナリ	---	A
	117~144	---	---	---
コマンドタイム	145~152	バイナリ	---	A
ユーザーエリア	153~168	該当なし	---	U

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
---	169~193	操作不可	---	---

ABD とバッファ

ABD とバッファ	Adabas コール前	Adabas コール後
フォーマット	F	U
レコード	---	A
サーチ	F	U
バリュー	F	U
ISN	---	A

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

* オプションのタイムアウト値、秒単位

--- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

バージョンインジケータ (ACBXVER)

F2

コマンドコード (ACBXCMD)

S1、S2、S4

レスポンスコード (ACBXRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、エラーサブコード (ACBXERRC) フィールドにサブコードを伴う場合があります。詳細は、*Adabas* メッセージおよびコードのドキュメントを参照してください。

コマンド ID (ACBXCID)

この値は、Adabas ワークに格納された ISN リストが完了したか、オーバーフローしたかを識別します。また、READ FIRST RECORD オプションが有効な場合、後続のコマンドに対してフォーマットバッファを識別するために使用します（「[フォーマットバッファ](#)」参照）。

SAVE ISN LIST オプションを使用する場合、またはオーバーフローした ISN を格納する場合には、このフィールドに空白やゼロ以外の値を指定する必要があります。

このフィールドの第1バイトには16進数'FF'にセットしないでください。

詳細は「[ISN リスト処理](#)」を参照してください。

データベース ID (ACBXDBID)

このフィールドを使用して、データベース ID を指定します。Adabas コールはこのデータベースに送られます。

このフィールドは4バイトのバイナリフィールドですが、現時点では2バイトのデータベース ID のみがサポートされています。したがって、データベース ID はフィールドの下位部分（右端のバイト）に指定し、その前にバイナリの0を付加する必要があります。

このフィールドがバイナリの0に設定されている場合は、Adabas APIはDDCARD入力データで指定されたADARUNカードのデータベース ID か、リンクされているカリンクルーチンによりロードされたLNKGBLSモジュールで指定されているデフォルトのデータベース ID 値のいずれかを使用します。

ファイル番号 (ACBXFNFR)

このフィールドは、ISN を選択するファイルの番号を指定するのに使用します。

このフィールドは4バイトのバイナリフィールドですが、ファイル番号はフィールドの下位部分（右端のバイト）に指定し、その前にバイナリの0を付加する必要があります。



Note: カップリングされた対のファイルで演算するコマンドの場合、このフィールドは、ISNまたはデータが返されるプライマリファイルを指定します。物理カップリングファイルのファイル番号は255以下でなければなりません。

Adabas 拡張ファイルでも検索できます。

ISN (ACBXISN)

Adabas はこのフィールドに、ISN 結果リストの先頭 ISN を返します。ISN が1件もなかった場合、このフィールドにはゼロが設定されます。

ISN 下限 (ACBXISL)

このフィールドを最初のS1またはS4コールで使用すると、ISN 結果リストの最小 ISN 値を指定できます。したがって、リストには、このフィールドに指定した ISN よりも大きな ISN のみが含まれます。このフィールドをゼロに設定すると、Adabas は該当するすべての ISN を返します。



Note: S2 コールの場合は、このフィールドに指定した値は無効です。ソートされている ISN リスト (S2 コマンド) の場合は、このフィールドにゼロを設定するか、正しい ISN を指定する必要があります。

ACBXISL フィールドは、8バイトのACBXISLGフィールドに埋め込まれた4バイトのバイナリフィールドで、まだ使用されていません。ACBXISLGフィールドの上位部分には、バイナリの0を設定します。

また、Adabas ワークにセーブした ISN リストから ISN グループを取得するときにも使用します。

ISN 数 (ACBXISQ)

このフィールドは、Sx コマンドの実行にかかる最大秒数を定義するのに使用します。

ACBXISQ フィールドは、8 バイトの ACBXISQG フィールドに埋め込まれた 4 バイトのバイナリフィールドで、まだ使用されていません。ACBXISQG フィールドの上位部分には、バイナリの 0 を設定します。

Adabas は、最初の Sx コールの結果として、検索条件を満たすレコードの数をこのフィールドに返します。セキュリティバイバリューを使用する場合、値 0 (1 レコードを検出) または 1 (複数のレコードを検出) とともにレスポンスコード 1 をこのフィールドに返します。詳細は、『Adabas Security マニュアル』を参照してください。

また、のちに Sx コールで Adabas ワークから ISN を取得すると、返された ISN の数がこのフィールドに格納されます。ISN 自体は ISN バッファに返されます。

コマンドオプション 1 (ACBXCOP1)

オプション	説明
H (ISN リストの格納)	Sx コマンドを実行して得られた全 ISN リストを、指定したコマンド ID で Adabas ワークに格納します。正しいコマンド ID を指定する必要があります。コマンド ID を指定しないと、ISN リストは Adabas ワークに格納されず、ISN バッファに格納されなかった ISN はすべて失われます。
R (リターン)	S4 コマンドを使用している場合は、読み込んでホールドするレコードが使用不可であれば、レスポンスコード 145 を返します。

コマンドオプション 2 (ACBXCOP2)

オプション	説明
D (降順)	S2 コマンドを使用している場合は、ディスクリプタ値を降順でソートします。

S2 コマンドでコマンドオプション 2 を指定しないと、ディスクリプタ値は昇順でソートされます。

コマンドオプション 1/2：CID 解放オプション (ACBXCOP1 および ACBXCOP2)

コマンドオプション 1 かコマンドオプション 2 フィールドのどちらかに、"I" オプションを指定できます。

オプション	説明
I	コマンド実行の最初に、コマンド ID フィールドで指定されているコマンド ID (CID) 値を解放します。コマンド ID が解放されるのは ISN リストテーブル内だけなので、同じコマンド ID が ISN 結果リストを識別するのに再度使用されます。

アディション 1：S2 コマンド、ソート制御用ディスクリプタ (ACBXADD1)

S2 コマンドを使用している場合は、ソート順を制御するディスクリプタ (複数可) を指定する必要があります。ソート引数の指定がなければレスポンスコード 28 が返されます。

ディスクリプタは3つまで指定できます。サブディスクリプタおよびスーパーディスクリプタも指定できます。フォネティックディスクリプタ、またはピリオディックグループ内にあるディスクリプタは指定できません。マルチプルバリューフィールドは使用できます。この場合、ISN レコード内の最低値に基づいてソートされます。

このフィールド内で使用しない桁は、空白にしておかなければなりません。例

```
XXYYbbbb
```

上記の意味は次に示すとおりです。

XX 主ソートディスクリプタ

YY 副ソートディスクリプタ

bbbb 空白

ソートできる ISN の数は、DBA が設定した SORT ワークエリアのサイズ (ADARUN LS パラメータ) に依存します。ソートエリアが小さすぎる場合、ソートは行われず、レスポンスコード 1 が返され、ISN は昇順で返されます。

アディクション 2：圧縮および非圧縮レコード長 (ACBXADD2)

コマンドが正常に処理されると、次の情報がこのフィールドに返されます。

- 少なくとも 1 つの有効なフィールド値がレコードバッファにある場合、先頭 2 バイトには、アクセスしたレコードの圧縮レコード長がバイナリ形式で格納されます。
- 後ろ 2 バイトには、フォーマットバッファで選択し、アクセスしたフィールドの非圧縮長がバイナリ形式で格納されます。

アディクション 3：パスワード (ACBXADD3)

このフィールドは、Adabas Security または ADAESI パスワードを指定するため使用します。データベース、ファイル、またはフィールドがセキュリティ保護されている場合は、有効なセキュリティパスワードまたは ADAESI パスワードをユーザーは指定する必要があります。パスワードを保護するために、コマンド処理時にアディクション 3 フィールドに空白がセットされます。

アディクション 4：サイファコードおよびバージョン/ニュークリアス ID (ACBXADD4)

このフィールドは、サイファコードを指定するために使用します。ファイルを暗号化する場合は、有効なサイファコードを指定する必要があります。ファイルを暗号化しない場合は、このフィールドに空白をセットする必要があります。

Adabas はコマンド処理時にサイファコードを空白に設定し、このフィールドの右端 (下位) 3 バイトにバージョンコードおよびデータベース ID を返します。詳細については「[コントロールブロックフィールド](#)」を参照してください。

アディクション 5：フォーマット ID、グローバルフォーマット ID (ACBXADD5)

このコマンド用の内部フォーマットバッファを識別する別のフォーマット ID を指定するか、グローバルフォーマット ID を指定します。

アディション5フィールドの上位ビットにゼロ（0）が設定されている場合、コマンド ID フィールドに指定された値がフォーマット ID として使用されます。

このビットに1がセットされていると、アディション5フィールドの5バイト目から8バイト目までがフォーマット ID として使用されます。

さらに、アディション5フィールドの第1バイトの最上位（左端）2ビットに、1（B'11'）がセットされると、アディション5フィールドの全8バイトが、グローバルフォーマット ID として使用されます（つまり、複数ユーザーがこのフォーマット ID を同時に使用できます）。

詳細は、「[コマンドID](#)、[フォーマットID](#)、[グローバルフォーマットID](#)」を参照してください。

エラーサブコード（ACBXERRC）

コマンドがゼロ以外のレスポンスコードを返したときは、このフィールドにレスポンスコードの正確な意味を定義したサブコードが含まれます。レスポンスコードとサブコードについては、『[Adabas メッセージおよびコード](#)』を参照してください。

バッファ

S1、S2、S4 の各コマンドでは、次のバッファを指定する必要があります。

- [フォーマットバッファ](#)
- [レコードバッファ](#)
- [サーチバッファとバリューバッファ](#)
- [ISN バッファ](#)

フォーマットバッファ

ISN 結果リストの先頭の ISN を持つレコードをデータストレージから読み込む場合、値を取得するレコード内のフィールドをこのバッファに指定する必要があります。フォーマットバッファの構文および例については、「[バッファの定義](#)」を参照してください。S2 コマンドの ISN ソート順を制御するためのユーザー指定のフィールドをアディション1フィールドに指定する必要があります。

読み込みを行わない場合、このバッファの空白以外の第1文字はピリオド（.）である必要があります。

有効なコマンド ID を指定すると、デコードされたフォーマットバッファが保持され、同じコマンド ID を使用した後続のコマンドでもそのバッファを使用できるようになります。

レコードバッファ

フォーマットバッファに READ オプションを使用するフィールド定義がある場合、Adabasはこのバッファに要求されたフィールドの値を返します。

その値は、特にユーザーがフォーマットバッファに標準長と異なるフィールド長および／またはフォーマットを要求しない限り、フィールドの標準長とフォーマットに従って返されます。

サーチバッファとバリュースearchバッファ

サーチバッファとバリュースearchバッファは、検索条件を定義するのに使用します。検索式（複数でも可）をサーチバッファに与え、検索式に相当する値をバリュースearchバッファに指定します。

フォーマットバッファの構文および例については、「[バッファの定義](#)」を参照してください。

ISN バッファ

Adabas は、このバッファに ISN 結果リストを配置します。各 ISN は 4 バイトの 2 進数で返されます。ISN は、S2 コマンドを使用していなければ昇順で返されます。S2 コマンドを使用している場合は、ユーザー指定のソート順で返されます。

問い合わせにファイルカップリング条件が含まれている場合、ISN 結果リストにはプライマリファイルに属する ISN のみが含まれます。プライマリファイルとは、コントロールブロックのファイル番号フィールドで指定するファイルです。

ISN バッファ長フィールドが 4 未満の場合、このバッファには ISN が 1 つも返されません。有効なコマンド ID を使用した場合、ISN バッファ長がゼロでなく、かつ ISN バッファが実行結果の全 ISN を入れるのに十分な長さではないと、Adabas はオーバーフローした ISN を Adabas ワークに格納します。その後、同じコマンド ID の S1、S2、または S4 コールを使用してこの ISN を取り出すことができます。

詳細は「[ISN リスト処理](#)」を参照してください。

43 S5 コマンド：カップリングリストの検索

- 機能および使用 462
- ACB インターフェイスダイレクトコール：S5 コマンド 462
- ACBX インターフェイスダイレクトコール：S5 コマンド 466
- バッファ 470

S5 コマンドは、指定されたファイルとカップリングされた ISN のリストを返したり保存したりします。

このchapterでは、次のトピックについて説明します。

機能および使用

S5 コマンドは、他のファイルのあるレコードとカップリングされているファイル内のレコードを求める場合に使用します。

ユーザーはファイル番号とファイル内の 1 つの ISN を指定し、さらにカップリングされた ISN を取得するファイル番号を指定します。また、オプションとして、取得する ISN 以上の ISN 値も指定できます。

Adabas は、アソシエータ内のカップリングリストを使用して、指定レコードに対してカップリングされているレコードを探します。データストレージにはアクセスしません。

Adabas は該当する ISN を [ISN バッファ](#)に返します。

ACB インターフェイスダイレクトコール：S5 コマンド

このsectionでは、S5コマンドのACBインターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの概要](#)
- [コントロールブロックフィールドの説明](#)
- [ACB の例](#)

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	--	--	--
コマンドコード	3~4	英数字	F	U
コマンド ID	5~8	英数字	F	U
ファイル番号	9~10	バイナリ	F	U
レスポンスコード	11~12	バイナリ	--	A
ISN	13~16	バイナリ	F	A
ISN 下限	17~20	バイナリ	F	U
ISN 数	21~24	バイナリ	--	A
	25~32	--	--	--
ISN バッファ長	33~34	バイナリ	F	U
コマンドオプション 1	35	英数字	F	U
コマンドオプション 2	36	英数字	F	U
アディクション 1	37~44	英数字	F	U
	45~48	--	--	--
アディクション 3	49~56	英数字	F	A
	57~72	--	--	--
コマンドタイム	73~76	バイナリ	--	A
ユーザーエリア	77~80	--	--	U

バッファエリア

バッファ	Adabas コール前	Adabas コール後
フォーマット	*	--
レコード	*	--
サーチ	*	--
バリュー	*	--
ISN	--	A

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

* 使用しませんが、コールステートメントのパラメータリストに含める必要があります。

-- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

コマンドコード (ACBCMD)

S5

コマンド ID (ACBCID)

SAVE ISN LIST オプションを使用する場合、またはオーバーフローした ISN を Adabas ワークに格納して読み込む場合、このフィールドには空白やゼロ以外の値を設定する必要があります。

このフィールドの第 1 バイトには 16 進数 'FF' にセットしないでください。

ファイル番号 (ACBFNR)

カップリングされた ISN を取得するファイルの番号。このファイルは、プライマリファイルと呼ばれ、アディクション 1 フィールドに指定したファイルとカップリングしていなければなりません。Adabas 拡張ファイルは指定できません。物理カップリングファイルのファイル番号は 255 以下でなければなりません。

レスポンスコード (ACBRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードには、アディクション 2 フィールドの右半分にサブコードが返される場合があります。レスポンスコード、サブコードについては『Adabas メッセージおよびコードマニュアル』を参照してください。

ISN (ACBISN)

この ISN のレコードにカップリングされた ISN が返されます。この ISN は、アディクション 1 フィールドに指定したファイル内に存在する必要があります。Adabas は、カップリングされた ISN 結果リストから、先頭の ISN をこのフィールドに返します。

ISN 下限 (ACBISL)

このフィールドを最初の Sx コールで使用すると、ISN 結果リストが、このフィールドに指定した ISN より大きい ISN に制限されます。このフィールドをゼロにすると、Adabas は条件に合った ISN をすべて返します。

また、このフィールドは、Adabas ワークからセーブされた ISN リストの ISN グループを取得するときにも使用します。

ISN 数 (ACBISQ)

最初の S5 コールは、指定したファイル内の検索条件を満たすレコードの数を返します。

のちに Sx コールで Adabas ワークから ISN を取得すると、このフィールドには ISN バッファに配置された ISN の数が入ります。

ISN バッファ長 (ACBIBL)

ISN バッファ長 (バイト単位)。この長さによって ISN バッファに配置される ISN の数が決定します。

このフィールドをゼロにすると、ISN は ISN バッファに追加されません。

後続の処理のために ISN リストをワークに保存するには、コマンドオプション1フィールドに H を指定し、有効なコマンド ID を使用します。L1 コマンドまたは L4 コマンドの GET NEXT オプションで ISN 結果リストを読み込む場合、または条件に合ったレコード数を確認する目的でのみそのコマンドを発行する場合は、この ISN バッファ長フィールドをゼロにする必要があります。

ゼロ以外の値を指定する場合は、4 の倍数にする必要があります。それ以外の場合、長さには 4 の倍数で次に小さい整数が設定されます。

コマンドオプション 1：SAVE ISN LIST オプション (ACBCOP1)

オプション	説明
H (ISN リストの格納)	S5 コマンドを実行して得られた全 ISN リストを、指定したコマンド ID で Adabas ワークに格納します。正しいコマンド ID を指定する必要があります。コマンド ID を指定しないと、ISN リストは Adabas ワークに格納されず、ISN バッファに格納されなかった ISN はすべて失われます。

コマンドオプション 1 または 2：コマンド ID オプションの解放 (ACBCOP1 または ACBCOP2)

コマンドオプション 1 かコマンドオプション 2 フィールドのどちらかに、I オプションを指定できます。

オプション	説明
I	コマンド ID フィールドに指定されているコマンド ID (CID) の値を解放します。この処理は、S5 コマンドの実行中に最初に行われます。コマンド ID が解放されるのは ISN リストテーブル内だけなので、同じコマンド ID が ISN 結果リストを識別するのに再度使用されます。

アディション 1：ファイル番号 (ACBADD1)

ISN フィールドに指定した ISN を持つファイルの番号。ファイル番号は、このフィールドの先頭の 2 バイトに指定する必要があります。番号はバイナリ形式で指定します。このフィールドの残りのバイトは空白にする必要があります。このフィールドは、一連の S5 コール間で変更しないでください。

アディション 3：パスワード (ACBADD3)

このフィールドは、Adabas Security または ADAESI パスワードを指定するため使用します。データベース、ファイル、またはフィールドがセキュリティ保護されている場合は、有効なセキュリティパスワードまたは ADAESI パスワードをユーザーは指定する必要があります。パスワードを保護するために、コマンド処理時にアディション 3 フィールドに空白がセットされます。

ACB の例

ISN 5 を持つファイル 1 内のレコードとカップリングされているファイル 2 のレコードを選択します。SAVE ISN LIST オプションを使用します。

コントロールブロック

コマンドコード	S5	
コマンド ID	S501	SAVE ISN LIST オプションを使用するため、空白でないコマンド ID が必要です。
ファイル番号	2	ファイル 2 のレコードを選択します。
ISN	5	ISN5 のレコードとカップリングしているレコードを選択します。
ISN 下限	0	条件に合った ISN をすべて選択します
ISN バッファ長	0	ISN バッファには ISN は返されません
コマンドオプション 1	H	SAVE ISN LIST オプションを使用します。
アディション 1	X'0001404040404040'	選択されるレコードとカップリングしているレコードはファイル 1 にあります。
アディション 3	パスワード	ファイルはセキュリティ保護されています

ACBX インターフェイスダイレクトコール：S5 コマンド

この section では、S5 コマンドの ACBX インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの概要](#)
- [コントロールブロックフィールドの説明](#)

コントロールブロックとバッファの概要

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1～2	---	---	---
バージョンインジケータ	3～4	バイナリ	F	U
	5～6	---	---	---
コマンドコード	7～8	英数字	F	U
	9～10	---	---	---
レスポンスコード	11～12	バイナリ	---	A
コマンド ID	13～16	英数字/バイナリ	F	U
データベース ID	17～20	数値	F	U
ファイル番号	21～24	数値	F	U
	25～28	---	---	---
ISN	29～32	バイナリ	F	A
	33～36	---	---	---
ISN 下限	37～40	バイナリ	F	U
	41～44	---	---	---
ISN 数	45～48	バイナリ	---	A
コマンドオプション 1	49	英数字	F	U
コマンドオプション 2	50	英数字	F	U
	51～56	---	---	---
アディション 1	57～64	英数字/バイナリ	F	U
	65～68	---	---	---
アディション 3	69～76	英数字/バイナリ	F	A
	77～114	---	---	---
エラーサブコード	115～116	バイナリ	---	A
	117～144	---	---	---
コマンドタイム	145～152	バイナリ	---	A
ユーザーエリア	153～168	該当なし	---	U
---	169～193	操作不可	---	---

ABD とバッファ

ABD とバッファ	Adabas コール前	Adabas コール後
ISN	---	A

上記の意味は次に示すとおりです。

- F Adabas コール前にユーザーが入力するフィールド
- A Adabas により入力されるフィールド
- U Adabas コール後も変化なし
- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

バージョンインジケータ (ACBXVER)

F2

コマンドコード (ACBXCMD)

S5

レスポンスコード (ACBXRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、エラーサブコード (ACBXERRC) フィールドにサブコードを伴う場合があります。詳細は、*Adabas* メッセージおよびコードのドキュメントを参照してください。

コマンド ID (ACBXCID)

SAVE ISN LIST オプションを使用する場合、またはオーバーフローした ISN を Adabas ワークに格納して読み込む場合、このフィールドには空白やゼロ以外の値を設定する必要があります。

このフィールドの第 1 バイトには 16 進数 'FF' にセットしないでください。

データベース ID (ACBXDBID)

このフィールドを使用して、データベース ID を指定します。Adabas コールはこのデータベースに送られます。

このフィールドは 4 バイトのバイナリフィールドですが、現時点では 2 バイトのデータベース ID のみがサポートされています。したがって、データベース ID はフィールドの下位部分 (右端のバイト) に指定し、その前にバイナリの 0 を付加する必要があります。

このフィールドがバイナリの 0 に設定されている場合は、Adabas API は DDCARD 入力データで指定された ADARUN カードのデータベース ID か、リンクされているカリンクルーチンによりロードされた LNKGBLS モジュールで指定されているデフォルトのデータベース ID 値のいずれかを使用します。

ファイル番号 (ACBXFNR)

このフィールドは、カップリングされた ISN を選択するファイルの番号を指定するのに使用します。このファイルは、プライマリファイルと呼ばれ、アディション 1 フィールドに指定

したファイルとカップリングしていなければなりません。Adabas 拡張ファイルは指定できません。物理カップリングファイルのファイル番号は 255 以下でなければなりません。

このフィールドは 4 バイトのバイナリフィールドですが、ファイル番号はフィールドの下位部分（右端のバイト）に指定し、その前にバイナリの 0 を付加する必要があります。

ISN (ACBXISN)

この ISN のレコードにカップリングされた ISN が返されます。この ISN は、アディクション 1 フィールドに指定したファイル内に存在する必要があります。Adabas は、カップリングされた ISN 結果リストから、先頭の ISN をこのフィールドに返します。

ACBXISN フィールドは、8 バイトの ACBXISNG フィールドに埋め込まれた 4 バイトのバイナリフィールドで、まだ使用されていません。ACBXISNG フィールドの上位部分には、バイナリの 0 を設定します。

ISN 下限 (ACBXISL)

このフィールドを最初の S_x コールで使用すると、ISN 結果リストが、このフィールドに指定した ISN より大きい ISN に制限されます。このフィールドをゼロにすると、Adabas は条件に合った ISN をすべて返します。

ACBXISL フィールドは、8 バイトの ACBXISLG フィールドに埋め込まれた 4 バイトのバイナリフィールドで、まだ使用されていません。ACBXISLG フィールドの上位部分には、バイナリの 0 を設定します。

また、このフィールドは、Adabas ワークからセーブされた ISN リストの ISN グループを取得するときにも使用します。

ISN 数 (ACBXISQ)

最初の S5 コールは、指定したファイル内の検索条件を満たすレコードの数を返します。

のちに S_x コールで Adabas ワークから ISN を取得すると、このフィールドには ISN バッファに配置された ISN の数が入ります。

コマンドオプション 1：SAVE ISN LIST オプション (ACBXCOP1)

オプション	説明
H (ISN リストの格納)	S5 コマンドを実行して得られた全 ISN リストを、指定したコマンド ID で Adabas ワークに格納します。正しいコマンド ID を指定する必要があります。コマンド ID を指定しないと、ISN リストは Adabas ワークに格納されず、ISN バッファに格納されなかった ISN はすべて失われます。

コマンドオプション1または2：コマンド ID 解放オプション（ACBXCOP1またはACBXCOP2）

コマンドオプション1かコマンドオプション2フィールドのどちらかに、「I」オプションを指定できます。

オプション	説明
I	コマンド ID フィールドに指定されているコマンド ID (CID) の値を解放します。この処理は、S5 コマンドの実行中に最初に行われます。コマンド ID が解放されるのは ISN リストテーブル内だけなので、同じコマンド ID が ISN 結果リストを識別するのに再度使用されます。

アディション1：ファイル番号（ACBXADD1）

ISN フィールドに指定した ISN を持つファイルの番号。ファイル番号は、このフィールドの先頭の2バイトに指定する必要があります。番号はバイナリ形式で指定します。このフィールドの残りのバイトは空白にする必要があります。このフィールドは、一連の S5 コール間で変更しないでください。

アディション3：パスワード（ACBXADD3）

このフィールドは、Adabas Security または ADAESI パスワードを指定するため使用します。データベース、ファイル、またはフィールドがセキュリティ保護されている場合は、有効なセキュリティパスワードまたは ADAESI パスワードをユーザーは指定する必要があります。パスワードを保護するために、コマンド処理時にアディション3フィールドに空白がセットされます。

エラーサブコード（ACBXERRC）

コマンドがゼロ以外のレスポンスコードを返したときは、このフィールドにレスポンスコードの正確な意味を定義したサブコードが含まれます。レスポンスコードとサブコードについては、『Adabas メッセージおよびコード』を参照してください。

バッファ

S5 コマンドには、次のバッファが適用されます。

- フォーマットバッファ
- レコードバッファ
- サーチバッファ
- バリ्यूバッファ

■ ISN バッファ

フォーマットバッファ

これが **ACB** インターフェイスダイレクトコールであり、フォーマットバッファが指定されていない場合は、処理エラーが発生します。ACB インターフェイスダイレクトコールはセットシーケンスにバッファが指定されていることを要求します。**ACBX** インターフェイスダイレクトコールの場合は、フォーマットバッファを指定する必要はありません。

レコードバッファ

ACB インターフェイスダイレクトコールの場合は、レコードバッファを指定しないと、処理エラーが発生します。ACB インターフェイスダイレクトコールの処理では、セットシーケンスにバッファが指定されていることが前提になっています。**ACBX** インターフェイスダイレクトコールの場合は、レコードバッファを指定する必要はありません。

サーチバッファ

ACB インターフェイスダイレクトコールの場合は、サーチバッファを指定しないと、処理エラーが発生します。ACB インターフェイスダイレクトコールの処理では、セットシーケンスにバッファが指定されていることが前提になっています。**ACBX** インターフェイスダイレクトコールの場合は、サーチバッファを指定する必要はありません。

バリュースタック

ACB インターフェイスダイレクトコールの場合は、バリュースタックを指定しないと、処理エラーが発生します。ACB インターフェイスダイレクトコールの処理では、セットシーケンスにバッファが指定されていることが前提になっています。**ACBX** インターフェイスダイレクトコールの場合は、バリュースタックを指定する必要はありません。

ISN バッファ

Adabas は、ISN バッファに ISN 結果リストを置きます。各 ISN は 4 バイトの 2 進数で返されます。ISN リストの最初の ISN は、コントロールブロックの ISN フィールドにも返されます。

ISN は 昇順で返されます。

ISN バッファ長がゼロでも、実行結果の全 ISN を入れるだけの十分な大きさでもなく、かつ有効なコマンド ID を使用した場合、Adabas はオーバーフローした ISN を Adabas ワークに格納します。これらの ISN は、同じコマンド ID を使用した S5 コールで取り出すことができます。詳細は「**ISN リスト処理**」を参照してください。

44 S8 コマンド：ISN リストの処理

- 機能および使用 474
- ACB インターフェイスダイレクトコール：S8 コマンド 475
- ACBX インターフェイスダイレクトコール：S8 コマンド 479
- バッファ 483

S8 コマンドは、同じファイルから作成された2つの ISN リストを AND、OR または NOT の論理演算子で結合します。詳しくは「[ISN リスト処理](#)」を参照してください。

このchapterでは、次のトピックについて説明します。

機能および使用

S8 コマンドは、先の S_x コマンドで作成された2つの ISN リストに論理演算処理を行います。どちらの ISN リストも、次の条件を満たす必要があります。

- 同じファイルから生成されています。
- ISN 順になっています。
- WORK データセットに格納されています。
- 両リストを作成するときリストに割り当てたコマンド ID で識別できます。

ISN 順に並んでない S2 コマンドまたは S9 コマンドで作成された ISN リストは使用できません。

ISN リストを作成し、S8 コマンドを実行するまでの間に、これらの ISN リストに対する処理（アクセスまたは更新）を行ってはいけません。

S8 コマンドは、次の論理演算に使用できます。

演算子	ISN 結果リストに格納される ISN の内容
AND	両方の ISN リスト
OR	どちらか一方の ISN リスト
NOT	2 番目ではなく最初の ISN リスト

実行結果の ISN は、コマンドオプションの指定やコマンド ID フィールドの設定に従って、ISN バッファに返されたり、ISN 昇順で WORK データセットに格納されたりします。

- コマンド ID に空白やゼロ以外の値を指定して、SAVE ISN LIST オプションも同時に指定した場合、ISN 結果リストは ISN バッファと WORK データセットの両方に格納されます。
- コマンド ID を指定しないで SAVE ISN LIST オプションを使用すると、ISN 結果リストは ISN バッファにのみ格納され、WORK データセットには格納されません。

ACB インターフェイスダイレクトコール：S8 コマンド

このsectionでは、S8コマンドのACBインターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- コントロールブロックとバッファの情報
- コントロールブロックフィールドの説明
- ACB の例

コントロールブロックとバッファの情報

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	--	--	--
コマンドコード	3~4	英数字	F	U
コマンド ID	5~8	英数字	F	U
ファイル番号	9~10	バイナリ	F	U
レスポンスコード	11~12	バイナリ	--	A
ISN	13~16	バイナリ	--	A
ISN 下限	17~20	バイナリ	F	U
ISN 数	21~24	バイナリ	--	A
	25~32	--	--	--
ISN バッファ長	33~34	バイナリ	F	U
コマンドオプション 1	35	英数字	F	U
コマンドオプション 2	36	英数字	F	U
アディション 1	37~44	英数字	F	U
	45~48	--	--	--
アディション 3	49~56	英数字	F	A
	57~72	--	--	--
コマンドタイム	73~76	バイナリ	--	A
ユーザーエリア	77~80	--	--	U

バッファエリア

バッファ	Adabas コール前	Adabas コール後
フォーマット	*	--
レコード	*	--
サーチ	*	--
バリュウ	*	--
ISN	--	A

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

* 使用しませんが、コールステートメントのパラメータリストに含める必要があります。

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

コマンドコード (ACBCMD)

S8

コマンド ID (ACBCID)

コマンドオプション 1 フィールドにコマンドオプションを指定した場合、このフィールドに空白およびゼロ以外のコマンド ID を指定する必要があります。

- i (コマンド ID の解放) オプションは、S8 実行時の最初の処理として、指定されたコマンド ID および関連する ISN リストを解放します。
- H (SAVE ISN LIST) オプションを指定すると、S8 コマンドの実行結果の ISN リストが、指定したコマンド ID で格納されます。コマンド ID を指定しないと、ISN リストは Adabas ワークに格納されず、ISN バッファに格納されなかった ISN はすべて失われます。

詳しくは「[ISN リスト処理](#)」を参照してください。

このフィールドの第 1 バイトには 16 進数 'FF' にセットしないでください。

ファイル番号 (ACBFNR)

処理する 2 つの ISN リストの作成元となったファイルの番号を指定します。

レスポンスコード (ACBRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、アディション 2 フィールドの下位 2 バイトにサブコードを伴う場合があります。詳細は『[Adabas メッセージおよびコード](#)』を参照してください。

ISN (ACBISN)

Adabasはこのフィールドに、ISN 結果リストの先頭 ISN を返します。実行結果の ISN リストがない場合、このフィールドは変更されません。これは最初のコールでも、Adabas WORK データセットから ISN を読み取るのに使用する後続のコールでも同様です。

ISN 下限 (ACBISL)

このフィールドを最初の S_x コールで使用すると、ISN 結果リストが、このフィールドに指定した ISN より大きい ISN に制限されます。このフィールドをゼロにすると、Adabas は条件に合った ISN をすべて返します。

また、このフィールドは、Adabas WORK データセットからセーブされた ISN リストの ISN グループを取得するときにも使用します。

ISN 数 (ACBISQ)

最初の S8 コールの結果として ISN リストに入る ISN の数が返されます。

また、Adabas WORK データセットから ISN を取得する後続の S8 コールの結果として、ISN バッファに返された ISN の数が表示されます。

ISN バッファ長 (ACBIBL)

ISN バッファ長 (バイト単位)。この長さによって ISN バッファに配置される ISN の数が決定します。

このフィールドをゼロにすると、ISN は ISN バッファに追加されません。L1 または L4 コマンドの GETNEXT オプションを指定して ISN 結果リストを読み込む場合、または条件に合ったレコード数を求めるためだけにコマンドを発行する場合、このフィールドはゼロにする必要があります。

ゼロ以外の値を指定する場合は、4 の倍数にする必要があります。それ以外の場合、長さには 4 の倍数で次に小さい整数が設定されます。

コマンドオプション 1：SAVE ISN LIST オプション、コマンド ID 解放オプション (ACBCOP1)

オプション	説明
H (ISN リストの格納)	S8 コマンドを実行して得られた全 ISN リストを、指定したコマンド ID で Adabas ワークに格納します。正しいコマンド ID を指定する必要があります。コマンド ID を指定しないと、ISN リストは Adabas ワークに格納されず、ISN バッファに格納されなかった ISN はすべて失われます。
I	S8 コマンド実行中の最初の処理として、コマンド ID フィールドに指定されているコマンド ID (CID) 値とすべての関連 ISN リストを解放します。コマンド ID が解放されるのは ISN リストテーブル内だけなので、同じコマンド ID が ISN 結果リストを識別するのに再度使用されます。

コマンドオプション 2：論理演算（ACBCOP2）

このフィールドに入力された値は、ISN リストに行う論理演算を示します。

オプション	オペレーション	ISN 結果リストに格納される ISN の内容
D	AND	両方の ISN リスト
O	OR	いずれかの ISN リスト
N	NOT	2 番目ではなく最初の ISN リスト

アディショナル 1：コマンド ID（ACBADD1）

このフィールドには、処理対象の ISN リストを識別するコマンド ID を（コマンド ID ごとに 4 バイト）指定しなければなりません。各 ISN リストは、現在 Adabas WORK データセットに格納されていて、同一ファイルの ISN で構成されている必要があります。

アディショナル 3：パスワード（ACBADD3）

このフィールドは、Adabas Security または ADAESI パスワードを指定するため使用します。データベース、ファイル、またはフィールドがセキュリティ保護されている場合は、有効なセキュリティパスワードまたは ADAESI パスワードをユーザーは指定する必要があります。パスワードを保護するために、コマンド処理時にアディショナル 3 フィールドに空白がセットされます。

ACB の例

2 つの ISN リストのいずれかに属する ISN を持つ第 3 の ISN リストを生成するために、2 つのリスト間に論理 OR 演算を行います。処理する ISN リストは、コマンド ID U020 と U021 の下で Adabas WORK データセットに格納したものです。ISN 結果リストはコマンド ID U999 の下で Adabas ワークに格納します。SAVE ISN LIST オプションを使用します。

コントロールブロック

コマンドコード	S8	
コマンド ID	U999	ISN 結果リストをコマンド ID U999 の下で格納します。
ISN 下限	0	実行結果の全 ISN を選択します。
ISN バッファ長	0	ISN バッファには ISN は返されません
コマンドオプション 1	H	SAVE ISN LIST オプション使用
コマンドオプション 2	O	OR 演算を行います。
アディショナル 1	U020U021	コマンド ID U020 と U021 で識別される ISN リストを処理します。
アディショナル 3	bbbbbbbb (空白)	ファイルはセキュリティ保護されていません

ACBX インターフェイスダイレクトコール：S8 コマンド

このsectionでは、S8 コマンドの ACBX インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの情報](#)
- [コントロールブロックフィールドの説明](#)

コントロールブロックとバッファの情報

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	---	---	---
バージョンインジケータ	3~4	バイナリ	F	U
	5~6	---	---	---
コマンドコード	7~8	英数字	F	U
	9~10	---	---	---
レスポンスコード	11~12	バイナリ	---	A
コマンド ID	13~16	英数字/バイナリ	F	U
データベース ID	17~20	数値	F	U
ファイル番号	21~24	数値	F	U
	25~28	---	---	---
ISN	29~32	バイナリ	---	A
	33~36	---	---	---
ISN 下限	37~40	バイナリ	F	U
	41~44	---	---	---
ISN 数	45~48	バイナリ	---	A
コマンドオプション 1	49	英数字	F	U
コマンドオプション 2	50	英数字	F	U
	51~56	---	---	---
アディション 1	57~64	英数字/バイナリ	F	U
	65~68	---	---	---
アディション 3	69~76	英数字/バイナリ	F	A
	77~114	---	---	---

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
エラーサブコード	115～116	バイナリ	---	A
	117～144	---	---	---
コマンドタイム	145～152	バイナリ	---	A
ユーザーエリア	153～168	該当なし	---	U
---	169～193	操作不可	---	---

ABD とバッファ

ABD とバッファ	Adabas コール前	Adabas コール後
ISN	---	A

上記の意味は次に示すとおりです。

- F Adabas コール前にユーザーが入力するフィールド
- A Adabas により入力されるフィールド
- U Adabas コール後も変化なし
- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

バージョンインジケータ (ACBXVER)

F2

コマンドコード (ACBXCMD)

S8

レスポンスコード (ACBXRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、エラーサブコード (ACBXERRC) フィールドにサブコードを伴う場合があります。詳細は、*Adabas* メッセージおよびコードのドキュメントを参照してください。

コマンド ID (ACBXCID)

コマンドオプション 1 フィールドにコマンドオプションを指定した場合、このフィールドに空白およびゼロ以外のコマンド ID を指定する必要があります。

- "I" (コマンド ID の解放) オプションは、S8 実行時の最初の処理として、指定されたコマンド ID および関連する ISN リストを解放します。

- "H" (SAVE ISN LIST) オプションを指定すると、S8 コマンドの実行結果の ISN リストが、指定したコマンド ID で格納されます。コマンド ID を指定しないと、ISN リストは Adabas ワークに格納されず、ISN バッファに格納されなかった ISN はすべて失われます。

詳しくは「[ISN リスト処理](#)」を参照してください。

このフィールドの第 1 バイトには 16 進数 'FF' にセットしないでください。

データベース ID (ACBXDBID)

このフィールドを使用して、データベース ID を指定します。Adabas コールはこのデータベースに送られます。

このフィールドは 4 バイトのバイナリフィールドですが、現時点では 2 バイトのデータベース ID のみがサポートされています。したがって、データベース ID はフィールドの下位部分 (右端のバイト) に指定し、その前にバイナリの 0 を付加する必要があります。

このフィールドがバイナリの 0 に設定されている場合は、Adabas API は DDCARD 入力データで指定された ADARUN カードのデータベース ID か、リンクされているかリンクルーチンによりロードされた LNKGBLS モジュールで指定されているデフォルトのデータベース ID 値のいずれかを使用します。

ファイル番号 (ACBXFNR)

このフィールドは、処理する 2 つの ISN リストの作成元となったファイルの番号を指定するのに使用します。

このフィールドは 4 バイトのバイナリフィールドですが、ファイル番号はフィールドの下位部分 (右端のバイト) に指定し、その前にバイナリの 0 を付加する必要があります。

ISN (ACBXISN)

Adabas はこのフィールドに、ISN 結果リストの先頭 ISN を返します。実行結果の ISN リストがない場合、このフィールドは変更されません。これは最初のコールでも、Adabas WORK データセットから ISN を読み取るのに使用する後続のコールでも同様です。

ISN 下限 (ACBXISL)

このフィールドを最初の S_x コールで使用すると、ISN 結果リストが、このフィールドに指定した ISN より大きい ISN に制限されます。このフィールドをゼロにすると、Adabas は条件に合った ISN をすべて返します。

ACBXISL フィールドは、8 バイトの ACBXISLG フィールドに埋め込まれた 4 バイトのバイナリフィールドで、まだ使用されていません。ACBXISLG フィールドの上位部分には、バイナリの 0 を設定します。

また、このフィールドは、Adabas WORK データセットからセーブされた ISN リストの ISN グループを取得するときにも使用します。

ISN 数 (ACBXISQ)

最初の S8 コールの結果として ISN リストに入る ISN の数が返されます。

また、Adabas WORK データセットから ISN を取得する後続の S8 コールの結果として、ISN バッファに返された ISN の数が表示されます。

コマンドオプション1：SAVEISNLIST オプション、コマンド ID 解放オプション (ACBXCOP1)

オプション	説明
H (ISN リストの格納)	S8 コマンドを実行して得られた全 ISN リストを、指定したコマンド ID で Adabas ワークに格納します。正しいコマンド ID を指定する必要があります。コマンド ID を指定しないと、ISN リストは Adabas ワークに格納されず、ISN バッファに格納されなかった ISN はすべて失われます。
I	S8 コマンド実行中の最初の処理として、コマンド ID フィールドに指定されているコマンド ID (CID) 値とすべての関連 ISN リストを解放します。コマンド ID が解放されるのは ISN リストテーブル内だけなので、同じコマンド ID が ISN 結果リストを識別するのに再度使用されます。

コマンドオプション2：論理演算 (ACBXCOP2)

このフィールドに入力された値は、ISN リストに行う論理演算を示します。

オプション	オペレーション	ISN 結果リストに格納される ISN の内容
D	AND	両方の ISN リスト
O	OR	いずれかの ISN リスト
N	NOT	2 番目ではなく最初の ISN リスト

アディショナル1：コマンド ID (ACBXADD1)

このフィールドには、処理対象の ISN リストを識別するコマンド ID を (コマンド ID ごとに 4 バイト) 指定しなければなりません。各 ISN リストは、現在 Adabas WORK データセットに格納されていて、同一ファイルの ISN で構成されている必要があります。

アディショナル3：パスワード (ACBXADD3)

このフィールドは、Adabas Security または ADAESI パスワードを指定するため使用します。データベース、ファイル、またはフィールドがセキュリティ保護されている場合は、有効なセキュリティパスワードまたは ADAESI パスワードをユーザーは指定する必要があります。パスワードを保護するために、コマンド処理時にアディショナル3 フィールドに空白がセットされます。

エラーサブコード (ACBXERRC)

コマンドがゼロ以外のレスポンスコードを返したときは、このフィールドにレスポンスコードの正確な意味を定義したサブコードが含まれます。レスポンスコードとサブコードについては、『Adabas メッセージおよびコード』を参照してください。

バッファ

S8 コマンドには、次のバッファが適用されます。

- フォーマットバッファ
- レコードバッファ
- サーチバッファ
- バリ्यूバッファ
- ISN バッファ

フォーマットバッファ

これが **ACB** インターフェイスダイレクトコールであり、フォーマットバッファが指定されていない場合は、処理エラーが発生します。ACB インターフェイスダイレクトコールはセットシーケンスにバッファが指定されていることを要求します。**ACBX** インターフェイスダイレクトコールの場合は、フォーマットバッファを指定する必要はありません。

レコードバッファ

ACB インターフェイスダイレクトコールの場合は、レコードバッファを指定しないと、処理エラーが発生します。ACB インターフェイスダイレクトコールの処理では、セットシーケンスにバッファが指定されていることが前提になっています。**ACBX** インターフェイスダイレクトコールの場合は、レコードバッファを指定する必要はありません。

サーチバッファ

ACB インターフェイスダイレクトコールの場合は、サーチバッファを指定しないと、処理エラーが発生します。ACB インターフェイスダイレクトコールの処理では、セットシーケンスにバッファが指定されていることが前提になっています。**ACBX** インターフェイスダイレクトコールの場合は、サーチバッファを指定する必要はありません。

バリ्यूバッファ

ACB インターフェイスダイレクトコールの場合は、バリ्यूバッファを指定しないと、処理エラーが発生します。ACB インターフェイスダイレクトコールの処理では、セットシーケンスにバッファが指定されていることが前提になっています。**ACBX** インターフェイスダイレクトコールの場合は、バリ्यूバッファを指定する必要はありません。

ISN バッファ

Adabas は、このバッファに ISN 結果リストを配置します。各 ISN は 4 バイトの 2 進数で返されます。ISN は ISN 順に返されます。

ISN バッファが実行結果の全 ISN を入れるのに十分な大きさでなく、しかも空白やゼロ以外のコマンド ID を使用した場合、Adabas はオーバーフローした ISN を WORK データセットに格納します。これらの ISN は、その後同じコマンド ID の S_x コールで読み込むことができます。詳しくは「[ISN リスト処理](#)」を参照してください。

45 S9 コマンド：ISN リストのソート

■ 機能および使用	486
■ ACB インターフェイスダイレクトコール：S9 コマンド	486
■ ACBX インターフェイスダイレクトコール：S9 コマンド	492
■ バッファ	497

S9 コマンドは、ISN リストを昇順またはユーザー指定のディスクリプタ順にソートします。

このchapterでは、次のトピックについて説明します。

機能および使用

S9 コマンドは、前の S_x コマンドを使用して作成した ISN リストや、ユーザーの与えた ISN リストをソートするのに使用します。ソート対象の ISN リストは、**ISN バッファ**にあるものか、または Adabas WORK データセットにあるものになります（ISN リストの作成時にそのリストに割り当てられたコマンド ID がアディクション 4 フィールドに指定されている場合）。

ISN リストは次の順序でソートされます。

- ISN 値の順（ISN の昇順）
- ユーザー指定の 1 個から 3 個のディスクリプタ値の順

ソート順を制御するのに 1~3 個のディスクリプタを指定できます。昇順、降順のどちらかを指定できます。

ソートする ISN リストには、ISN が昇順で含まれている必要があります。つまり、このリストとして、S2 コマンドまたは S9 コマンドのディスクリプタ順オプションで作成した ISN リストを使用できません。

ISN 結果リストは、**ISN バッファ**に返されますが、オプションで Adabas WORK データセットに格納することもできます。

S9 コマンドは Adabas 拡張ファイルに対しても実行できます。

ACB インターフェイスダイレクトコール：S9 コマンド

このsectionでは、S9 コマンドの ACB インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- **コントロールブロックとバッファの情報**
- **コントロールブロック**

■ ACB の例

コントロールブロックとバッファの情報

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1~2	--	--	--
コマンドコード	3~4	英数字	F	U
コマンド ID	5~8	英数字	F	U
ファイル番号	9~10	バイナリ	F	U
レスポンスコード	11~12	バイナリ	--	A
ISN	13~16	バイナリ	--	A
ISN 下限	17~20	バイナリ	F	U
ISN 数	21~24	バイナリ	F	A
	25~32	--	--	--
ISN バッファ長	33~34	バイナリ	F	U
コマンドオプション 1	35	英数字	F	U
コマンドオプション 2	36	英数字	F	U
アディクション 1	37~44	英数字	F	U
	45~48	--	--	--
アディクション 3	49~56	英数字	F	A
アディクション 4	57~64	英数字	F	A
	65~72	--	--	--
コマンドタイム	73~76	バイナリ	--	A
ユーザーエリア	77~80	--	--	U

バッファエリア

バッファ	Adabas コール前	Adabas コール後
フォーマット	*	--
レコード	*	--
サーチ	*	--
バリュー	*	--
ISN	F	A

上記の意味は次に示すとおりです。

F Adabas コール前にユーザーが入力するフィールド

A Adabas により入力されるフィールド

U Adabas コール後も変化なし

* 使用しませんが、コールステートメントのパラメータリストに含める必要があります。

-- 未使用

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACB フィールドにバイナリの 0 を設定することをお勧めします。

コマンドコード (ACBCMD)

S9

コマンド ID (ACBCID)

このフィールドには、空白やゼロ以外の値を設定してもかまいません。このコマンド ID は、I (コマンド ID の解放) オプションおよび H (SAVE ISN LIST) オプションにのみ適用されます。Adabas ワークに格納した ISN リストを取り出すためのコマンド ID は、アディクション 4 フィールドに指定する必要があります。

このフィールドの第 1 バイトは、16 進数の FF であってはなりません。

ファイル番号 (ACBFNR)

ソートする ISN リストの作成元となったファイルの番号。

S9 コマンドは Adabas 拡張ファイルに対しても実行できます。



Note: 2 バイトファイル番号およびデータベース ID を使用する場合は、コントロールブロックの先頭バイトに X'30' を入力しなければなりません。

レスポンスコード (ACBRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、コマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、アディクション 2 フィールドの下位 2 バイトにサブコードを伴う場合があります。詳細は『Adabas メッセージおよびコード』を参照してください。

ISN (ACBISN)

Adabas はこのフィールドに、ISN 結果リストの先頭 ISN を返します。該当する ISN がなかった場合、このフィールドは変更されません。これは最初のコールでも、Adabas WORK データセットから ISN を読み取るのに使用する後続のコールでも同様です。

ISN 下限 (ACBISL)

このフィールドを最初の Sx コールで使用すると、ISN 結果リストが、このフィールドに指定した ISN より大きい ISN に制限されます。このフィールドをゼロにすると、Adabas は条件に合った ISN をすべて返します。

また、このフィールドは、Adabas WORK データセットから、セーブされた ISN リストの ISN グループを取得するときにも使用します。

ISN 数 (ACBISQ)

ソートする ISN リストが ISN バッファにある場合、このフィールドにはソート対象の ISN の数が指定されている必要があります。セキュリティバイバリューを使用する場合、値 0 (1 レコードを検出) または 1 (複数のレコードを検出) とともにレスポンスコード 1 をこのフィールドに返します。詳細は、『Adabas Security マニュアル』を参照してください。

Adabas は、最初の S9 コールの結果として、ISN 結果リストに含まれるレコードの数をこのフィールドに返します。また、WORK データセットから ISN を取得するための後続の S9 コールの結果として、Adabas は、ISN バッファに配置された ISN の数を返します。

ISN バッファ長 (ACBIBL)

ISN バッファ長 (バイト単位) は、ISN バッファに格納する ISN の数を決めるのに使用します。

- このフィールドをゼロにすると、ISN は ISN バッファに追加されません。ISN 結果リストを L1 コマンドまたは L4 コマンドの GETNEXT オプションで読み込む場合は、このフィールドにゼロを設定し、コマンドオプション 1 フィールドに H を指定します。また、コマンド条件に合ったレコードの数を求める目的でのみ S9 コマンドを発行する場合は、このフィールドをゼロにし、コマンド ID を指定しないでください。そうすると、ソートした ISN リストが返されることも、格納されることもありません。
- ゼロ以外の値を指定する場合は、4 の倍数にする必要があります。そうでないと、Adabas は次に小さい 4 の倍数にします。

ソートする ISN が ISN バッファにある場合、このフィールドはソートする ISN の数の 4 倍以上である必要があります。オーバーフローした ISN はコマンド ID フィールドに指定したコマンド ID を使用して Adabas ワークに格納され、後続の Sx コマンドで読み込むことができます。

コマンドオプション 1：SAVE ISN LIST オプション (ACBCOP1)

オプション	説明
H (ISN リストの格納)	S9 コマンドを実行して得られた全 ISN リストを、指定したコマンド ID で Adabas ワークに格納します。正しいコマンド ID を指定する必要があります。コマンド ID を指定しないと、ISN リストは Adabas ワークに格納されず、ISN バッファに格納されなかった ISN はすべて失われます。L1 または L4 コマンドの GETNEXT オプションを指定して ISN 結果リストを読み込む場合、ISN バッファ長フィールドをゼロに設定してこのオプションを使用します。このオプションを指定した場合、コマンドオプション 2 に I (コマンド ID の解放) オプションを指定することはできません。

コマンドオプション 2：降順オプション (ACBCOP2)

オプション	説明
D (降順)	ISN リストを降順にソートします。このオプションは、ISN 値でソートするとき指定できません。

S9 コマンドでコマンドオプション 2 を指定しないと、ISN リストは昇順でソートされます。
 コマンドオプション 1 または 2：コマンド ID オプションの解放 (ACBCOP1 または ACBCOP2)

コマンドオプション 1 かコマンドオプション 2 フィールドのどちらかに、「I」オプションを指定できます。

オプション	説明
I	S9 コマンド実行の最初に、コマンド ID フィールドで指定されているコマンド ID (CID) 値を解放します。コマンド ID が解放されるのは ISN リストテーブル内だけなので、同じコマンド ID が ISN 結果リストを識別するのに再度使用されます。コマンドオプション 1 として H (SAVE ISN LIST) オプションを指定した場合、このオプションをコマンドオプション 2 として指定することはできません。

アディション 1：ソート順 (ACBADD1)

使用するソート順をこのフィールドに指定します。

値を ISN とすると、ソート順に "ISNbbbb" 値が使用されることとなります (bbbb には空白が入ります)。

ソート順を 1 つ以上のディスクリプタ値にする場合、使用ディスクリプタをこのフィールドに指定しなければなりません。ディスクリプタは 3 つまで指定できます。ディスクリプタ、サブディスクリプタ、およびスーパーディスクリプタを指定できます。フォネティックディスクリプタとピリオディックグループ内のディスクリプタは指定できません。マルチプルバリュースフィールドは指定できますが、この場合、指定レコード内の最低値に基づいて ISN がソートされます。ディスクリプタはバイト 1 から開始して (左詰め) 指定し、残りの桁は空白でなければなりません。

ソートできる ISN の数は、DBA が設定した ADARUN パラメータのサイズに依存します。この制限を超えると、ソートは実行されずレスポンスコード 1 が返されます。

アディション 3：パスワード (ACBADD3)

このフィールドは、Adabas Security または ADAESI パスワードを指定するため使用します。データベース、ファイル、またはフィールドがセキュリティ保護されている場合は、有効なセキュリティパスワードまたは ADAESI パスワードをユーザーは指定する必要があります。パスワードを保護するために、コマンド処理時にアディション 3 フィールドに空白がセットされます。

アディション 4：コマンド ID (ACBADD4)

ソートする ISN リストが Adabas WORK データセットにある場合は、リストが格納されているコマンド ID を、このフィールドの最初の 4 バイトに指定しなければなりません。

ソートする ISN リストが ISN バッファにある場合、このフィールドは空白にしなければなりません。

Adabas はコマンド処理中にアディクション 4 フィールドを空白に設定し、このフィールドの右端（下位）3 バイトにバージョンコードおよびデータベース ID を返します。詳細については「[コントロールブロックフィールド](#)」を参照してください。

ACB の例

- [例 1](#)
- [例 2](#)

例 1

ISN バッファ内の ISN リストを、ISN 順でソートします。622 個の ISN をソートします。

コントロールブロック

コマンドコード	S9	
コマンド ID	S901	空白およびゼロ以外のコマンド ID が必要
ファイル番号	1	ソートする ISN リストはファイル 1 から派生
ISN 数	622	622 個の ISN をソートします。
ISN 下限	0	ISN をすべて選択
ISN バッファ長	2488	またはそれ以上。ソートする ISN につき 4 バイトが必要です。
コマンドオプション 1	H	SAVE ISN LIST オプション使用
コマンドオプション 2	b (空白)	昇順を使用します。
アディクション 1	ISNbbbb	ISN 値をソート順に使用します (bbbb には空白が入ります)。
アディクション 3	bbbbbbbb (空白)	ファイルはセキュリティ保護されていません
アディクション 4	bbbbbbbb (空白)	ソートする ISN リストは ISN バッファにあります。

バッファエリア

ISN バッファ	ソートする ISN をこのバッファに格納しておきます。各 ISN は 4 バイトの 2 進数である必要があります。
----------	---

例 2

Adabas ワークに格納されている ISN リストをソートします。ISN リストを格納するためのコマンド ID は U066 です。主および副シーケンスフィールドとしてディスクリプタ AA および AB を使用してリストをソートします。降順ソートとします。

コントロールブロック

コマンドコード	S9	
コマンド ID	S902	空白およびゼロ以外のコマンド ID が必要
ファイル番号	1	ソートする ISN リストはファイル 1 から派生
ISN 下限	0	ISN をすべて選択
ISN バッファ長	0	ISN バッファには ISN は返されません
コマンドオプション 1	H	SAVE ISN LIST オプション使用
コマンドオプション 2	D	降順を使用します。
アデクション 1	AAABbbbb	AA を主シーケンスフィールド、AB を副シーケンスフィールドとして使用します (bbbb には空白が入ります)。
アデクション 3	bbbbbbbb (空白)	ファイルはセキュリティ保護されていません
アデクション 4	U066bbbb	ソートする ISN リストは、コマンド ID "U066" で Adabas ワークに格納されています (bbbb には空白が入ります)。

ACBX インターフェイスダイレクトコール：S9 コマンド

このsectionでは、S9 コマンドの ACBX インターフェイスダイレクトコールについて説明します。本書には、次のトピックが含まれています。

- [コントロールブロックとバッファの情報](#)
- [コントロールブロックフィールドの説明](#)

コントロールブロックとバッファの情報

コントロールブロック

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

フィールド	位置	フォーマット	Adabas コール前	Adabas コール後
	1～2	---	---	---
バージョンインジケータ	3～4	バイナリ	F	U
	5～6	---	---	---
コマンドコード	7～8	英数字	F	U
	9～10	---	---	---
レスポンスコード	11～12	バイナリ	---	A
コマンド ID	13～16	英数字／バイナリ	F	U
データベース ID	17～20	数値	F	U
ファイル番号	21～24	数値	F	U
	25～28	---	---	---
ISN	29～32	バイナリ	---	A
	33～36	---	---	---
ISN 下限	37～40	バイナリ	F	U
	41～44	---	---	---
ISN 数	45～48	バイナリ	F	A
コマンドオプション 1	49	英数字	F	U
コマンドオプション 2	50	英数字	F	U
	51～56	---	---	---
アディション 1	57～64	英数字／バイナリ	F	U
	65～68	---	---	---
アディション 3	69～76	英数字／バイナリ	F	A
アディション 4	77～84	英数字	F	A
	85～114	---	---	---
エラーサブコード	115～116	バイナリ	---	A
	117～144	---	---	---
コマンドタイム	145～152	バイナリ	---	A
ユーザーエリア	153～168	該当なし	---	U
---	169～193	操作不可	---	---

ABD とバッファ

ABD とバッファ	Adabas コール前	Adabas コール後
ISN	F	A

上記の意味は次に示すとおりです。

- F Adabas コール前にユーザーが入力するフィールド
- A Adabas により入力されるフィールド
- U Adabas コール後も変化なし
- 未使用

コントロールブロックフィールドの説明

ダイレクトコールが開始される前に、未使用の ACBX フィールドにバイナリの 0 を設定することをお勧めします。

バージョンインジケータ (ACBXVER)

F2

コマンドコード (ACBXCMD)

S9

レスポンスコード (ACBXRSP)

Adabas はこのフィールドに、コマンドのレスポンスコードを返します。レスポンスコード 0 は、このコマンドが正しく実行されたことを示します。ゼロ以外のレスポンスコードは、エラーサブコード (ACBXERRC) フィールドにサブコードを伴う場合があります。詳細は、Adabas メッセージおよびコードのドキュメントを参照してください。

コマンド ID (ACBXCID)

このフィールドには、空白やゼロ以外の値を設定してもかまいません。このコマンド ID は、"I" (コマンド ID の解放) オプションおよび "H" (SAVE ISN LIST) オプションにのみ適用されます。Adabas ワークに格納した ISN リストを取り出すためのコマンド ID は、アディション 4 フィールドに指定する必要があります。

このフィールドの第 1 バイトは、16 進数の FF であってはなりません。

データベース ID (ACBXDBID)

このフィールドを使用して、データベース ID を指定します。Adabas コールはこのデータベースに送られます。

このフィールドは 4 バイトのバイナリフィールドですが、現時点では 2 バイトのデータベース ID のみがサポートされています。したがって、データベース ID はフィールドの下位部分 (右端のバイト) に指定し、その前にバイナリの 0 を付加する必要があります。

このフィールドがバイナリの 0 に設定されている場合は、Adabas API は DDCARD 入力データで指定された ADARUN カードのデータベース ID か、リンクされているかリンクルーチンによりロードされた LNKGBLS モジュールで指定されているデフォルトのデータベース ID 値のいずれかを使用します。

ファイル番号 (ACBXFNR)

このフィールドは、ソートする ISN リストの作成元となったファイルの番号を指定するのに使用します。

このフィールドは 4 バイトのバイナリフィールドですが、ファイル番号はフィールドの下位部分（右端のバイト）に指定し、その前にバイナリの 0 を付加する必要があります。

S9 コマンドは Adabas 拡張ファイルに対しても実行できます。

ISN (ACBXISN)

Adabas はこのフィールドに、ISN 結果リストの先頭 ISN を返します。該当する ISN がなかった場合、このフィールドは変更されません。これは最初のコールでも、Adabas WORK データセットから ISN を読み取るのに使用する後続のコールでも同様です。

ISN 下限 (ACBXISL)

このフィールドを最初の S_x コールで使用すると、ISN 結果リストが、このフィールドに指定した ISN より大きい ISN に制限されます。このフィールドをゼロにすると、Adabas は条件に合った ISN をすべて返します。

ACBXISL フィールドは、8 バイトの ACBXISLG フィールドに埋め込まれた 4 バイトのバイナリフィールドで、まだ使用されていません。ACBXISLG フィールドの上位部分には、バイナリの 0 を設定します。

また、このフィールドは、Adabas WORK データセットから、セーブされた ISN リストの ISN グループを取得するときにも使用します。

ISN 数 (ACBXISQ)

ソートする ISN リストが ISN バッファにある場合、このフィールドにはソート対象の ISN の数が指定されている必要があります。セキュリティバイバリューを使用する場合、値 0（1 レコードを検出）または 1（複数のレコードを検出）とともにレスポンスコード 1 をこのフィールドに返します。詳細は、『*Adabas Security マニュアル*』を参照してください。

ACBXISQ フィールドは、8 バイトの ACBXISQG フィールドに埋め込まれた 4 バイトのバイナリフィールドで、まだ使用されていません。ACBXISQG フィールドの上位部分には、バイナリの 0 を設定します。

Adabas は、最初の S9 コールの結果として、ISN 結果リストに含まれるレコードの数をこのフィールドに返します。また、WORK データセットから ISN を取得するための後続の S9 コールの結果として、Adabas は、ISN バッファに配置された ISN の数を返します。

コマンドオプション 1：SAVE ISN LIST オプション (ACBXCOP1)

オプション	説明
H (ISN リストの格納)	S9 コマンドを実行して得られた全 ISN リストを、指定したコマンド ID で Adabas ワークに格納します。正しいコマンド ID を指定する必要があります。コマンド ID を指定しないと、ISN リストは Adabas ワークに格納されず、ISN バッファに格納されなかった ISN はすべて失われます。L1 または L4 コマンドの GET NEXT オプションを指定して ISN 結果リストを読み込む場合、ISN バッファ長フィールドをゼロに設定してこのオプションを使用します。このオプションを指定した場合、コマンドオプション 2 に "I" (コマンド ID の解放) オプションを指定することはできません。

コマンドオプション 2：降順オプション (ACBXCOP2)

オプション	説明
D (降順)	ISN リストを降順にソートします。このオプションは、ISN 値でソートするとき指定できません。

S9 コマンドでコマンドオプション 2 を指定しないと、ISN リストは昇順でソートされます。

コマンドオプション 1 または 2：コマンド ID 解放オプション (ACBXCOP1 および ACBXCOP2)

コマンドオプション 1 かコマンドオプション 2 フィールドのどちらかに、"I" オプションを指定できます。

オプション	説明
I	S9 コマンド実行の最初に、コマンド ID フィールドで指定されているコマンド ID (CID) 値を解放します。コマンド ID が解放されるのは ISN リストテーブル内だけなので、同じコマンド ID が ISN 結果リストを識別するのに再度使用されます。コマンドオプション 1 として "H" (SAVE ISN LIST) オプションを指定した場合、このオプションをコマンドオプション 2 として指定することはできません。

アディショナル 1：ソート順 (ACBXADD1)

使用するソート順をこのフィールドに指定します。

値を ISN とすると、ソート順に "ISNbbbb" 値が使用されることとなります (bbbb には空白が入ります)。

ソート順を 1 つ以上のディスクリプタ値にする場合、使用ディスクリプタをこのフィールドに指定しなければなりません。ディスクリプタは 3 つまで指定できます。ディスクリプタ、サブディスクリプタ、およびスーパーディスクリプタを指定できます。フォネティックディスクリプタとピリオディックグループ内のディスクリプタは指定できません。マルチプルバリュースフィールドは指定できますが、この場合、指定レコード内の最低値に基づいて ISN がソートされます。ディスクリプタはバイト 1 から開始して (左詰め) 指定し、残りの桁は空白でなければなりません。

ソートできる ISN の数は、DBA が設定した ADARUN パラメータのサイズに依存します。この制限を超えると、ソートは実行されずレスポンスコード 1 が返されます。

アディクション 3：パスワード (ACBXADD3)

このフィールドは、Adabas Security または ADAESI パスワードを指定するため使用します。データベース、ファイル、またはフィールドがセキュリティ保護されている場合は、有効なセキュリティパスワードまたは ADAESI パスワードをユーザーは指定する必要があります。パスワードを保護するために、コマンド処理時にアディクション 3 フィールドに空白がセットされます。

アディクション 4：コマンド ID (ACBXADD4)

ソートする ISN リストが Adabas WORK データセットにある場合は、リストが格納されているコマンド ID を、このフィールドの最初の 4 バイトに指定しなければなりません。

ソートする ISN リストが ISN バッファにある場合、このフィールドは空白にしなければなりません。

Adabas はコマンド処理中にアディクション 4 フィールドを空白に設定し、このフィールドの右端（下位）3 バイトにバージョンコードおよびデータベース ID を返します。詳細については「[コントロールブロックフィールド](#)」を参照してください。

エラーサブコード (ACBXERRC)

コマンドがゼロ以外のレスポンスコードを返したときは、このフィールドにレスポンスコードの正確な意味を定義したサブコードが含まれます。レスポンスコードとサブコードについては、『[Adabas メッセージおよびコード](#)』を参照してください。

バッファ

S9 コマンドには、次のバッファが適用されます。

- フォーマットバッファ
- レコードバッファ
- サーチバッファ
- バリューストックバッファ

■ ISN バッファ

フォーマットバッファ

これが **ACB インターフェイスダイレクトコール** であり、フォーマットバッファが指定されていない場合は、処理エラーが発生します。ACB インターフェイスダイレクトコールはセットシーケンスにバッファが指定されていることを要求します。**ACBX インターフェイスダイレクトコール** の場合は、フォーマットバッファを指定する必要はありません。

レコードバッファ

ACB インターフェイスダイレクトコール の場合は、レコードバッファを指定しないと、処理エラーが発生します。ACB インターフェイスダイレクトコールの処理では、セットシーケンスにバッファが指定されていることが前提になっています。**ACBX インターフェイスダイレクトコール** の場合は、レコードバッファを指定する必要はありません。

サーチバッファ

ACB インターフェイスダイレクトコール の場合は、サーチバッファを指定しないと、処理エラーが発生します。ACB インターフェイスダイレクトコールの処理では、セットシーケンスにバッファが指定されていることが前提になっています。**ACBX インターフェイスダイレクトコール** の場合は、サーチバッファを指定する必要はありません。

バリューバッファ

ACB インターフェイスダイレクトコール の場合は、バリューバッファを指定しないと、処理エラーが発生します。ACB インターフェイスダイレクトコールの処理では、セットシーケンスにバッファが指定されていることが前提になっています。**ACBX インターフェイスダイレクトコール** の場合は、バリューバッファを指定する必要はありません。

ISN バッファ

ユーザーは、ソートする ISN リストをこのバッファに指定できます。

Adabas は、このバッファに ISN 結果リストを配置します。各 ISN は 4 バイトの 2 進数で返されます。ISN ソートオプションを有効にした（アディクション 1 フィールドに「ISNbbbb」を入力した）場合、ISN は昇順で返されます。それ以外の場合、ISN はユーザーがディスクリプタに指定した値の順で返されます。

ISN バッファ長がゼロでなく、また、実行結果の全 ISN を入れるだけの十分な大きさでもなく、かつ有効なコマンド ID が指定された場合、Adabas ではオーバーフローした ISN を Adabas WORK データセットに格納します。これらの ISN は、その後同じコマンド ID をアディクション 4 フィールドに指定した S9 コールで読み込むことができます。詳細については、「[ISN リスト処理](#)」セクションを参照してください。

46 プログラミング例

このchapterでは、各種言語の Adabas ダイレクトコールの例を示します。

情報は次の項目で構成されています。

- 例で使用されているファイル定義
- **ACB** の例

47 例で使用されているファイル定義

このドキュメントのすべての例で、次のファイル定義が使用されています。2つのファイル構造（ファイル1とファイル2）を使用しています。

次の表は、ファイル1と2の構造を示しています。

SF 標準フォーマット

SL 標準長

ファイル1

ファイル1は、セキュリティ保護されておらず、また、暗号化もされていません。このドキュメントで示す例では、ファイル1がカップリングベースのディスクリプタ AA により、ファイル2とカップリングされていると想定しています。

Adabas 定義	説明
01,GA	グループ GA。フィールド AA と AB で構成されています。
02,AA,8,A,DE,NU	エレメンタリフィールド AA。SL は 8 バイト、SF は英数字、ディスクリプタ、空値省略です。
02,AB,2,P,DE,NU	エレメンタリフィールド AB。SL は 2、SF はパック、ディスクリプタ、空値省略です。
01,AC,20,A,NU	エレメンタリフィールド AC。SL は 20、SF は英数字、空値省略です。
01,MF,3,A,MU,DE,NU	マルチプルバリュフィールド MF。SL は 3、SF は英数字、ディスクリプタ、空値省略です。
01,GB,PE	ピリオディックグループ GB
02,BA,1,B,DE,NU	エレメンタリフィールド BA（ピリオディックグループ GB 内）。SL は 1、SF はバイナリ、ディスクリプタ、空値省略です。
02,BB,5,P,NU	エレメンタリフィールド BB（ピリオディックグループ GB 内）。SL は 5、SF はパック、空値省略です。

例で使用されているファイル定義

Adabas 定義	説明
02,BC,10,A,NU	エレメンタリフィールド BC (ピリオディックグループ GB 内)。SL は 10、SF は英数字、空値省略です。
01,GC,PE	ピリオディックグループ GC
02,CA,7,A,DE,NU	エレメンタリフィールド CA (ピリオディックグループ GC 内)。SL は 7、SF は英数字、ディスクリプタ、空値省略です。
02,CB,10,A,MU,NU SF	マルチプルバリューフィールド CB (ピリオディックグループ GC 内)。SL は 10、SF は英数字、空値省略です。

ファイル 2

ファイル 2 は、セキュリティ保護されています。暗合化はされていません。このドキュメントで示す例では、ファイル 2 がカップリングベースのフィールド RA により、ファイル 1 とカップリングされているものとしています。

Adabas 定義	説明
01,RG	グループ RG レコード内の全フィールドから成ります。
02,RA,8,A,DE,NU	エレメンタリフィールド RA。SL は 8、SF は英数字、ディスクリプタ、空値省略です。
02,RB,10,A,DE	エレメンタリフィールド RB。SL は 10、SF は英数字、ディスクリプタです。
02,GX	グループ GX。フィールド XA、XB、XC、XD、および XE で構成されています。
03,XA,10,A	エレメンタリフィールド XA。SL は 10、SF は英数字です。
03,XB,2,P,DE	エレメンタリフィールド XB。SL は 2、SF はパック、ディスクリプタです。
03,XC,6,U	エレメンタリフィールド XC。SL は 6、SF はアンパックです。
03,XD,8,A,DE,NU	エレメンタリフィールド XD。SL は 8、SF は英数字、ディスクリプタ、空値省略です。
03,XE,5,A,DE,NU	エレメンタリフィールド XE。SL は 5、SF は英数字、ディスクリプタ、空値省略です。
SA=RA(1,4)	サブディスクリプタ SA。フィールド RA の 1~4 バイトで構成されており、形式は英数字です。
SB=RA(1,8),RB(1,4)	スーパーディスクリプタ SB。フィールド RA の 1~8 バイトとフィールド RB の 1~4 バイトで構成されており、形式は英数字です。
SC=XB(1,2),XC(1,6)	スーパーディスクリプタ SC。フィールド XB の 1~2 バイトとフィールド XC の 1~6 バイトで構成されており、形式はバイナリです。

48 ACB の例

■ ACB アセンブラの例	504
■ ACB COBOL の例	511
■ ACB PL/I の例	518
■ ACB Fortran の例	525

このchapterでは、ACBインターフェイスを使用する次のようなダイレクトコールの例について説明します。

ACB アセンブラの例

このsectionでは、アセンブラでの Adabas ダイレクトコールの使用例を示します。いずれの例でも、[以前に定義した Adabas ファイル](#)を使用しています。

```
*** CONTROL BLOCK
      DS      0F
CB     DS     0CL80      USER CONTROL BLOCK
      DC     CL2' '     RESERVED FOR ADABAS USE
CCODE  DC     CL2' '     COMMAND CODE
CID    DC     CL4' '     COMMAND ID
FNR    DC     H'0'      FILE NUMBER
RC     DC     H'0'      RESPONSE CODE
ISN    DC     F'0'      ISN
ISNLL  DC     F'0'      ISN LOWER LIMIT
ISNQ   DC     F'0'      ISN QUANTITY
FBL    DC     H'100'    FORMAT BUFFER LENGTH
RBL    DC     H'250'    RECORD BUFFER LENGTH
SBL    DC     H'50'     SEARCH BUFFER LENGTH
VBL    DC     H'100'    VALUE BUFFER LENGTH
IBL    DC     H'20'     ISN BUFFER LENGTH
COPT1  DC     CL1' '     COMMAND OPTION 1
COPT2  DC     CL1' '     COMMAND OPTION 2
ADD1   DC     CL8' '     ADDITIONS 1
ADD2   DC     CL4' '     ADDITIONS 2
ADD3   DC     CL8' '     ADDITIONS 3
ADD4   DC     CL8' '     ADDITIONS 4
ADD5   DC     CL8' '     ADDITIONS 5
CTIME  DC     F'0'      COMMAND TIME
UAREA  DC     CL4' '     USER AREA
*
*
*** USER BUFFER AREAS
FB     DC     CL100' '   FORMAT BUFFER
RB     DC     CL250' '   RECORD BUFFER
SB     DC     CL50' '    SEARCH BUFFER
VB     DC     CL100' '   VALUE BUFFER
IB     DC     CL20' '    ISN BUFFER
* * *
```

このsectionでは、次の例を記載しています。

- [例 1](#)
- [例 2](#)

■ 例3：ET ロジックでのユーザーセッション

例1

- XB=99 であるファイル 2 のレコードを検索します。
- GET NEXT オプションを使って選択した各レコードを読みます。

オープンコマンドの発行

```

EXMP1 MVC CCODE,=C'OP'          OP COMMAND
      MVC RB(4),=C'ACC.'        ACCESS ONLY REQUESTED
      CALL ADABAS,(CB,FB,RB)    CALL ADABAS
      CLC RC,=H'0'             CHECK RESPONSE CODE
      BNE EX1ERR               BRANCH IF NOT 0

```

検索コマンドの発行

```

      MVC CCODE,=C'S1'          FIND COMMAND
      MVC CID,=C'S101'         NONBLANK CID REQUIRED FOR
*                                IDENTIFICATION OF THE LIST
      MVC FNR,=H'2'           FILE 2
      MVC ISNLL,=F'0'         ALL QUALIFYING ISNS DESIRED
      MVC IBL,=H'0'           ISN BUFFER NOT REQUIRED
      MVI FB,C'.'             NO READ OF DATA STORAGE
      MVC SB(7),=C'XB,3,U.'    SEARCH CRITERION
      MVC VB(3),=C'099'       SEARCH VALUE
      CALL ADABAS,(CB,FB,RB,SB,VB) CALL ADABAS
      CLC RC,=H'0'           CHECK RESPONSE CODE
      BNE EX1ERR             BRANCH IF NOT 0
      CLC ISNQ,=F'0'         CHECK NUMBER OF ISNS FOUND
      BE EX1EXIT             BRANCH TO EXIT IF NO ISNS FOUND

```

条件を満たす各レコードの読み取り

```

EX1B MVC CCODE,=C'L1'          READ COMMAND
      MVC ISN,=F'0'          BEGIN WITH 1ST ISN IN LIST
      MVI COPT2,C'N'         GET NEXT OPTION TO BE USED
      MVC FB(3),=C'RG.'      ALL FIELDS TO BE RETURNED
EX1C CALL ADABAS,(CB,FB,RB)    CALL ADABAS
      CLC RC,=H'0'           CHECK RESPONSE CODE
      BE EX1D               BRANCH IF RESPONSE CODE 0
      CLC RC,=H'3'           CHECK IF ALL RECORDS READ
      BE EX1EXIT           BRANCH IF YES
      B EX1ERR             BRANCH TO ERROR ROUTINE
EX1D . . .                   PROCESS RECORD . . .
      B EX1C               BRANCH TO READ NEXT RECORD

```

エラールーチン

```
EX1ERR EQU *
*          DISPLAY ERROR MESSAGE
*          .          TERMINATE USER PROGRAM
```

クローズコマンドの発行

```
EX1EXIT MVC  CCODE,=C'CL'          CLOSE COMMAND
          CALL ADABAS,(CB)          CALL ADABAS
          CLC  RC,=H'0'             CHECK RESPONSE CODE
          BNE  EX1ERR               BRANCH IF NOT 0
```

例 2

- ファイル 1 のレコードは、物理順に全部読み込まれます。
- 読み込んだ各レコードを次の値で更新します。
 - フィールド AA = ABCDEFGH
 - フィールド AB = 500
- ユーザーは、ファイル 1 の排他制御権限を持つことが前提です。

オープンコマンドの発行

```
EXMP2 MVC  CCODE,=C'OP'          OPEN COMMAND
          MVC  RB(6),=C'EXU=1.'    EXCLUSIVE CONTROL REQUESTED
          CALL ADABAS,(CB,FB,RB)   CALL ADABAS
          CLC  RC,=H'0'             CHECK RESPONSE CODE
          BE   EX2A                 BRANCH IF RESPONSE CODE 0
          B    EX2ERR               BRANCH IF NOT 0
```

物理読み込みシーケンシャルコマンドの発行

```
EX2A MVC  CID,=C'L201'           NONBLANK CID REQUIRED
          MVC  FNR,=H'1'           FILE 1 TO BE READ
          MVC  ISN,=F'0'           ALL RECORDS TO BE READ
          MVC  FB(3),=C'GA.'       VALUES FOR FIELDS AA AND AB
*          (GROUP GA) TO BE RETURNED
EX2B MVC  CCODE,=C'L2'           READ PHYS. SEQ.
          CALL ADABAS,(CB,FB,RB)   CALL ADABAS
          CLC  RC,=H'0'             CHECK RESPONSE CODE
          BE   EX2C                 BRANCH IF RESPONSE CODE 0
          CLC  RC,=H'3'             CHECK FOR END-OF-FILE
          BE   EX2EXIT              BRANCH TO EXIT IF END-OF-FILE
          B    EX2ERR               BRANCH TO ERROR ROUTINE
```

レコードの更新

- 読み込まれたフィールドと同じフィールドが更新されます。
- 同じ CID とフォーマットバッファを更新コマンドに使用できます。

- 更新対象のレコードの ISN は、L2 コマンドの結果として、ISN フィールドにすでに存在していることが前提です。

```

EX2C MVC  CCODE,=C'A1'          UPDATE COMMAND
      MVC  RB(8),=C'ABCDEFGH'    VALUE FOR FIELD AA
      MVC  RB+8(2),=PL2'500'    VALUE FOR FIELD AB
      CALL ADABAS,(CB,FB,RB)    CALL ADABAS
      CLC  RC,=H'0'            CHECK RESPONSE CODE
      BE   EX2B                BRANCH TO READ NEXT RECORD

```

エラールーチン

```

EX2ERR EQU  *
*          .                    DISPLAY ERROR MESSAGE
*          .                    TERMINATE USER PROGRAM

```

ユーザーセッションのクローズ

```

EX2EXIT MVC  CCODE,=C'CL'      CLOSE COMMAND
      CALL ADABAS,(CB)        CALL ADABAS
      CLC  RC,=H'0'          CHECK RESPONSE CODE
      BNE  EX2ERR            BRANCH IF NOT 0
      . . .

```

例 3：ET ロジックでのユーザーセッション

ユーザーセッションの初期化時に、ユーザープログラムは、前回のユーザーセッションで、最後に正常に完了したトランザクションを示す情報を表示します。

ユーザートランザクションごとのユーザープログラムの処理内容は次のとおりです。

- ファイル 1 と 2 の更新用のキーとして使われる 8 文字の入力を端末から受け入れます。
- ファイル 1 に対して FIND コマンドを発行し、フィールド AA = 入力キーのレコードが存在するかどうか判断します。

レコードが 1 件も見つからなかった場合、ユーザープログラムはメッセージを発行します。レコードが見つかった場合のユーザープログラムの処理内容は次のとおりです。

- ファイル 1 からそのレコードを削除します
- ファイル 2 に新しいレコード（フィールド RA = 入力された入力キー）を追加します。

その他のフィールドは空値にします。

レコードを正常に追加できない場合、ユーザープログラムは、BT コマンドを発行して、エラーメッセージを表示します。

両方とも正常に更新された場合、ユーザープログラムは ET コマンドを発行します。

- セッションの初期化
- トランザクション処理

セッションの初期化

図のプログラムのセクションは、ユーザーセッションの初期化時にのみ実行されます。

オープンコマンドの発行

OP コマンドが発行されるたびに、前回のセッションの ET データが読み込まれます。

```

EXMP3 EQU *
      MVC CCODE,=C'OP'          OPEN COMMAND
      MVI COPT2,C'E'           ET DATA TO BE READ
      MVC ADD1,=C'USER0001'    USER IDENTIFICATION
      MVC ADD3,=C'PASSWORD'    USER PASSWORD
      MVC RB(8),=C'UPD=1,2.'   FILES 1 AND 2 TO BE UPDATED
      CALL ADABAS,(CB,FB,RB)   CALL ADABAS
      CLC RC,=H'0'            CHECK RESPONSE CODE
      BE EX3A                 BRANCH IF RESPONSE CODE 0
      CLC RC,=H'9'            CHECK FOR RESPONSE CODE 9
      BE EXMP3                 BRANCH TO REPEAT OPEN
      B EX3ERR                 BRANCH IF NOT 0 OR 9
EX3A EQU *
      CLC CID,=F'0'           CHECK IF ET DATA FROM
*                               PREVIOUS SESSION EXISTS
      BE EX3B                 BRANCH IF NO ET DATA
*                               . . .

```

ET データの表示

レコードバッファ内にある ET データを端末画面に表示して、前回のユーザーセッションで最後に正常に完了したトランザクションをユーザーに知らせます。

```

      B EX3C                 BRANCH TO BEGIN TRANS. PROCESS.
EX3B EQU *

```

ET データを受信しなかった場合

ET データを受け取らなかった場合は、前回のユーザーセッションで正常に完了したトランザクションがなかったことを示すメッセージが表示されます。

トランザクション処理

このセクションは、ユーザートランザクションごとに実行されます。

```
EX3C EQU *
* . . . ACCEPT INPUT FROM TERMINAL . . .
```

検索コマンドの発行

ファイル1に対してFINDコマンドを発行し、入力された入力キーと等しいレコードがフィールドAAに存在するかどうか判断します。

```
EX3D EQU *
MVC CCODE,=C'S4' FIND WITH HOLD COMMAND
MVC CID,=C' ' ISN LIST NOT TO BE SAVED
MVC FNR,=H'1' FILE 1
MVC ISNLL,=F'0' ALL QUALIFY. ISNS TO BE RETURNED
MVI FB,C'. ' NO READ OF DATA STORAGE
MVC SB(3),=C'AA.' SEARCH CRITERION
MVC VB(8),INPUT SEARCH VALUE
CALL ADABAS,(CB,FB,RB,SB,VB,IB) CALL ADABAS
CLC RC,=H'0' CHECK RESPONSE CODE
BE EX3E BRANCH IF RESPONSE CODE 0
B EX3ERR BRANCH TO ERROR ROUTINE
EX3E EQU *
CLC ISNQ,=F'0' CHECK NUMBER OF RECORDS FOUND
BNE EX3F BRANCH IF RECORD FOUND
```

レコードが見つからなかった場合のメッセージの発行

レコードが1件も見つからなかった場合、ユーザープログラムによって修正を要求するメッセージが発行されます。

```
B EX3C RETURN TO ACCEPT USER INPUT
```

ファイル1からのレコードの削除

削除対象のレコードのISNは、S4コマンドの結果として、ISNフィールドにホールド状態で存在しています。

```

EX3F EQU *
MVC CCODE,=C'E4'          DELETE COMMAND
CALL ADABAS,(CB)          CALL ADABAS
CLC RC,=H'0'              CHECK RESPONSE CODE
BE EX3G                   BRANCH IF RESPONSE CODE 0
CLC RC,=H'9'              CHECK IF CURRENT TRANS. HAS BEEN
*                           BACKED OUT BY ADABAS
BE EX3D                   IF YES, BRANCH TO REPEAT S4
B EX3ERR                  BRANCH TO ERROR ROUTINE

```

ファイル 2 への新しいレコードの追加

```

EX3G EQU *
MVC CCODE,=C'N1'          ADD NEW RECORD
MVC FNR,=H'2'             FILE 2
MVC FB(6),=C'RA.'        VALUE BEING PROVIDED FOR RA
MVC RB(8),INPUT           VALUE FOR FIELD RA
CALL ADABAS,(CB,FB,RB)   CALL ADABAS
CLC RC,=H'0'              CHECK RESPONSE CODE
BE EX3I                   BRANCH IF RESPONSE CODE 0
CLC RC,=H'9'              WAS TRANSACTION BACKED OUT?
BE EX3D                   IF YES, RETURN TO REISSUE TRANS.

```

新しいレコードを追加できない場合

新しいレコードの追加が成功しなかった場合、トランザクションはバックアウトされ、エラー状態が存在することがユーザーに通知されます。

```

MVC CCODE,=C'BT'          BACKOUT TRANSACTION
CALL ADABAS,(CB)          CALL ADABAS
CLC RC,=H'0'              CHECK IF RESPONSE CODE 0
BE EX3H                   BRANCH IF 0

```

バックアウトが成功しなかった場合

バックアウトが成功しなかった場合は、その結果を示すメッセージが発行されます。

```

B EX3ERR                  BRANCH TO ERROR ROUTINE
EX3H EQU *

```

バックアウトに成功した場合

バックアウトに成功した場合は、エラーの検出後、トランザクションがバックアウトされたことを示すメッセージが発行されます。

```
B      EX3ERR                                BRANCH TO ERROR ROUTINE
```

更新の正常実行:ET データでの ET コマンドの発行

更新が正常に実行された場合は、ET データでの ET コマンドが発行されます。

```
EX3I EQU *
      MVC  CCODE,=C'ET'          END OF TRANSACTION COMMAND
      MVI  COPT2,C'E'           ET DATA TO BE WRITTEN
      MVC  RB(8),INPUT          ET DATA CONSISTS OF INPUT KEY OF THIS TRANSACTION
      CALL ADABAS,(CB,FB,RB)    CALL ADABAS
      CLC  RC,=H'0'            CHECK RESPONSE CODE
      BE   EX3C                IF RESPONSE CODE 0, RETURN TO RECEIVE INPUT FOR
*                               THE NEXT TRANSACTION
      CLC  RC,=H'9'            CHECK IF CURRENT TRANSACTION HAS BEEN BACKED OUT
*                               BY ADABAS
      BE   EX3D                IF CURRENT TRANSACTION HAS BEEN BACKED OUT,
*                               RETURN TO REISSUE TRANSACTION
```

エラールーチン

```
EX3ERR EQU *
*      .                        NONZERO RESPONSE CODE RECEIVED
*      .                        DISPLAY ERROR MESSAGE
*      .                        TERMINATE USER PROGRAM
*      . . .
INPUT DS   CL8                KEY ENTERED FROM TERMINAL
```

ACB COBOL の例

このsectionでは、COBOLでの Adabas ダイレクトコールの使用例を示します。各例では、以前に定義した **Adabas ファイル** を使用します。

```
*
*** CONTROL BLOCK
01 CONTROL-BLOCK.
   02 FILLER                PIC X(2) VALUE SPACES.
   02 COMMAND-CODE          PIC X(2) VALUE SPACES.
   02 COMMAND-ID           PIC X(4) VALUE SPACES.
   02 FILE-NUMBER          PIC S9(4) COMP VALUE +0.
   02 RESPONSE-CODE        PIC S9(4) COMP VALUE +0.
   02 ISN                  PIC S9(8) COMP VALUE +0.
   02 ISN-LOWER-LIMIT      PIC S9(8) COMP VALUE +0.
```

```

02 ISN-QUANTITY          PIC S9(8) COMP VALUE +0.
02 FORMAT-BUFFER-LENGTH  PIC S9(4) COMP VALUE +100.
02 RECORD-BUFFER-LENGTH  PIC S9(4) COMP VALUE +250.
02 SEARCH-BUFFER-LENGTH  PIC S9(4) COMP VALUE +50.
02 VALUE-BUFFER-LENGTH   PIC S9(4) COMP VALUE +100.
02 ISN-BUFFER-LENGTH     PIC S9(4) COMP VALUE +20.
02 COMMAND-OPTION-1      PIC X VALUE SPACE.
02 COMMAND-OPTION-2      PIC X VALUE SPACE.
02 ADDITIONS-1           PIC X(8) VALUE SPACES.
02 ADDITIONS-2           PIC X(4) VALUE SPACES.
02 ADDITIONS-3           PIC X(8) VALUE SPACES.
02 ADDITIONS-4           PIC X(8) VALUE SPACES.
02 ADDITIONS-5           PIC X(8) VALUE SPACES.
02 COMMAND-TIME          PIC S9(8) COMP VALUE +0.
02 USER-AREA             PIC X(4) VALUE SPACES.
*
*** USER BUFFER AREAS
01 FORMAT-BUFFER         PIC X(100) VALUE SPACES.
01 RECORD-BUFFER         PIC X(250) VALUE SPACES.
01 SEARCH-BUFFER         PIC X(50) VALUE SPACES.
01 VALUE-BUFFER          PIC X(100) VALUE SPACES.
01 ISN-BUFFER            PIC X(20) VALUE SPACES.
*
```

```

*** ADDITIONAL FIELDS USED IN THE EXAMPLES
01 PROGRAM-WORK-AREA.
05     COMM-ID PIC X(4).
05     COMM-ID-X REDEFINES COMM-ID PIC S9(8) COMP.
05     INPUT-KEY PIC X(8).
05     RECORD-BUFFER-EX2.
10     RECORD-BUFFER-A PIC X(8).
10     RECORD-BUFFER-B PIC S9(3) COMP-3.
05     RECORD-BUFFER-EX3.
10     OPEN-RECORD-BUFFER.
15     OPEN-RECORD-BUFFER-X PIC X(8).
15     FILLER PIC S9(8) COMP.
10     FILLER PIC X(18).
10     UPDATED-XC PIC X(6).
10     LAST-XD PIC X(8).
10     FILLER PIC X(5).
05     USER-DATA.
10     RESTART-XD PIC X(8).
10     RESTART-ISN PIC S9(8) COMP.
05     SYNC-CHECK-SWITCH PIC 9 VALUE 0.
05     AB-VALUE PIC S9(4) COMP-3 VALUE +500.
*
```

このsectionでは、次の例を記載しています。

- 例 1
- 例 2

■ 例3：ET ロジックでのユーザーセッション

例1

- XB=99 であるファイル2のレコードを検索します。
- GET NEXT オプションを使って選択した各レコードを読みます。

オープンコマンドの発行

```
EXMP1.
MOVE      'OP' TO COMMAND-CODE.
MOVE      'ACC.' TO RECORD-BUFFER.
CALL      'ADABAS'
          USING CONTROL-BLOCK, FORMAT-BUFFER, RECORD-BUFFER.
IF RESPONSE-CODE NOT EQUAL TO 0 GO TO EX1ERR.
```

検索コマンドの発行

```
MOVE      'S1' TO COMMAND-CODE.
MOVE      'S101' TO COMMAND-ID.
MOVE      2 TO FILE-NUMBER.
MOVE      0 TO ISN-LOWER-LIMIT.
MOVE      0 TO ISN-BUFFER-LENGTH.
MOVE      '.' TO FORMAT-BUFFER.
MOVE      'XB,3,U.' TO SEARCH-BUFFER.
MOVE      '099' TO VALUE-BUFFER.
CALL      'ADABAS' USING CONTROL-BLOCK, FORMAT-BUFFER,
          RECORD-BUFFER, SEARCH-BUFFER, VALUE-BUFFER.
IF RESPONSE-CODE NOT EQUAL TO 0 GO TO EX1ERR.
EX1A.
IF ISN-QUANTITY = 0 GO TO EX1EXIT.
```

条件を満たす各レコードの読み取り

```
EX1B.
MOVE      'L1' TO COMMAND-CODE.
MOVE      0 TO ISN.
MOVE      'N' TO COMMAND-OPTION-2.
MOVE      'RG.' TO FORMAT-BUFFER.
EX1C.
CALL      'ADABAS'
          USING CONTROL-BLOCK, FORMAT-BUFFER, RECORD-BUFFER.
IF RESPONSE-CODE = 0 GO TO EX1D.
IF RESPONSE-CODE = 3 GO TO EX1EXIT.
EX1D.
. . . PROCESS RECORD . . .
GO TO EX1C.
```

エラールーチン

```
EX1ERR.  
*      .DISPLAY ERROR MESSAGE  
*      .TERMINATE USER PROGRAM
```

クローズコマンドの発行

```
EX1EXIT.  
MOVE      'CL' TO COMMAND-CODE.  
CALL      'ADABAS' USING CONTROL-BLOCK.  
IF RESPONSE-CODE NOT EQUAL TO 0 GO TO EX1ERR.
```

例 2

- ファイル 1 のレコードは、物理順に全部読み込まれます。
- 読み込んだ各レコードを次の値で更新します。
 - フィールド AA = ABCDEFGH
 - フィールド AB = 500
- ユーザーは、ファイル 1 の排他制御権限を持つことが前提です。

オープンコマンドの発行

```
EXMP2.  
MOVE      'OP' TO COMMAND-CODE.  
MOVE      'EXU=1.' TO RECORD-BUFFER.  
CALL      'ADABAS' USING  
          CONTROL-BLOCK, FORMAT-BUFFER, RECORD-BUFFER.  
IF RESPONSE-CODE NOT EQUAL TO 0 GO TO EX2ERR.
```

物理読み込みシーケンシャルコマンドの発行

```
EX2A.  
MOVE      'L201' TO COMMAND-ID.  
MOVE      1 TO FILE-NUMBER.  
MOVE      0 TO ISN.  
MOVE      'GA.' TO FORMAT-BUFFER.  
EX2B.  
MOVE      'L2' TO COMMAND-CODE.  
CALL      'ADABAS' USING  
          CONTROL-BLOCK, FORMAT-BUFFER, RECORD-BUFFER.  
IF RESPONSE-CODE = 0 GO TO EX2C.  
IF RESPONSE-CODE = 3 GO TO EX2EXIT.  
GO TO EX2ERR.
```

レコードの更新

- 読み込まれたフィールドと同じフィールドが更新されます。
- 同じ CID とフォーマットバッファを更新コマンドに使用できます。

- 更新対象のレコードの ISN は、L2 コマンドの結果として、ISN フィールドにすでに存在していることが前提です。

EX2C.

```
MOVE      'A1' TO COMMAND-CODE.
MOVE      'ABCDEFGH' TO RECORD-BUFFER-A.
MOVE      AB-VALUE TO RECORD-BUFFER-B.
CALL      'ADABAS' USING
          CONTROL-BLOCK, FORMAT-BUFFER, RECORD-BUFFER-EX2.
IF RESPONSE-CODE NOT EQUAL TO 0 GO TO EX2ERR.
GO TO EX2B.
```

エラールーチン

EX2ERR.

```
. DISPLAY ERROR MESSAGE
. TERMINATE USER PROGRAM
```

ユーザーセッションのクローズ

EX2EXIT.

```
MOVE      'CL' TO COMMAND-CODE.
CALL      'ADABAS' USING CONTROL-BLOCK.
IF RESPONSE-CODE NOT EQUAL TO 0 GO TO EX2ERR.
```

例 3 : ET ロジックでのユーザーセッション

ユーザーセッションの初期化時に、ユーザープログラムは、前回のユーザーセッションで、最後に正常に完了したトランザクションを示す情報を表示します。

ユーザートランザクションごとのユーザープログラムの処理内容は次のとおりです。

- ファイル 1 と 2 の更新用のキーとして使われる 8 文字の入力を端末から受け入れます。
- ファイル 1 に対して FIND コマンドを発行し、フィールド AA = 入力キーのレコードが存在するかどうか判断します。

レコードが 1 件も見つからなかった場合、ユーザープログラムはメッセージを発行します。レコードが見つかった場合のユーザープログラムの処理内容は次のとおりです。

- ファイル 1 からそのレコードを削除します
- ファイル 2 に新しいレコード（フィールド RA = 入力された入力キー）を追加します。

その他のフィールドは空値にします。

レコードを正常に追加できない場合、ユーザープログラムは、BT コマンドを発行して、エラーメッセージを表示します。

両方とも正常に更新された場合、ユーザープログラムは ET コマンドを発行します。

セッションの初期化

プログラムのこのセクションは、ユーザーセッションの初期化中にのみ実行されます。

- OP コマンドが発行されるときに、前回のセッションの ET データが読み込まれます。
- メッセージが端末画面に表示され、前回のユーザーセッションで正常に完了した最後のトランザクションが通知されます。

```

EX3.
    MOVE      'OP' TO COMMAND-CODE.
    MOVE      'E' TO COMMAND-OPTION-2.
    MOVE      'USER0002' TO ADDITIONS-1.
    MOVE      'PASSWORD' TO ADDITIONS-3.
    MOVE      'UPD=1,2.' TO RECORD-BUFFER.
    CALL      'ADABAS' USING
              CONTROL-BLOCK, FORMAT-BUFFER, RECORD-BUFFER.
    IF RESPONSE-CODE = 9 GO TO EX3.
    IF RESPONSE-CODE NOT EQUAL TO 0
      GO TO EX3ERR.

EX3A.
    MOVE      COMMAND-ID TO COMM-ID.
    IF COMM-ID-X = +0
      GO TO EX3B.
* Display ET data (contained in RECORD BUFFER) on screen to inform user of
* last successfully processed transaction of previous user session.
      . . .DISPLAY ET DATA. . .
    GO TO EX3C.

EX3B.
*** No ET data received.
* Display message that no transactions were successfully processed during
* the previous user session
      . . .DISPLAY MESSAGE . . .
*** Transaction processing.
* This section is executed for each user transaction.
EX3C.
*      . . .ACCEPT INPUT FROM TERMINAL. . .
* Issue Find command for file 1 to determine if record exists with field AA
* equal to input key entered.
EX3D.
    MOVE      'S4' TO COMMAND-CODE.
    MOVE      SPACES TO COMMAND-ID.
    MOVE      1 TO FILE-NUMBER.
    MOVE      0 TO ISN-LOWER-LIMIT.
    MOVE      '.' TO FORMAT-BUFFER.
    MOVE      'AA.' TO SEARCH-BUFFER.
    MOVE      INPUT-KEY TO VALUE-BUFFER.
    CALL      'ADABAS' USING
              CONTROL-BLOCK, FORMAT-BUFFER, RECORD-BUFFER,
              SEARCH-BUFFER, VALUE-BUFFER, ISN-BUFFER.
    IF RESPONSE-CODE = 0
      GO TO EX3E.

```

```
GO TO EX3ERR.
EX3E.
    IF ISN-QUANTITY NOT EQUAL TO ZEROS
        GO TO EX3F.
***No records found, issue message requesting correction.
    . . .ISSUE MESSAGE . . .
    GO TO EX3C.
*** Delete record from file 1.
*   ISN of record to be deleted is already in ISN field and in hold
status
*   as a result of the S4 command.
    EX3F.
        MOVE      E3' TO COMMAND-CODE.
        CALL      'ADABAS' USING CONTROL-BLOCK.
        IF RESPONSE-CODE = 0
            GO TO EX3G.
        IF RESPONSE-CODE = 9
            GO TO EX3D.
        GO TO EX3ERR.
*** Add new record to file 2.
    EX3G.
        MOVE      'N1' TO COMMAND-CODE.
        MOVE      2 TO FILE-NUMBER.
        MOVE      'RA.' TO FORMAT-BUFFER.
        MOVE      INPUT-KEY TO RECORD-BUFFER.
        CALL      'ADABAS' USING
            CONTROL-BLOCK, FORMAT-BUFFER, RECORD-BUFFER.
        IF RESPONSE-CODE = 0
            GO TO EX3I.
        IF RESPONSE-CODE = 9
            GO TO EX3D.
*** Attempt to add new record not successful.
*   Backout transaction.
*   Notify user that error condition exists.
        MOVE      'BT' TO COMMAND-CODE.
        CALL      'ADABAS' USING control-block.
        IF RESPONSE-CODE = 0
            GO TO EX3H.
*** Backout not successful.
*   Issue message indicating that the backout was not successful
    GO TO EX3ERR.
    EX3H.
*** Backout successful.
*   Issue message indicating the error condition detected while while
adding a
*   new record
        GO TO EX3ERR.
*** Updates successfully executed.
*   Issue ET command with ET data.
    EX3I.
        MOVE      'ET' TO COMMAND-CODE.
        MOVE      'E' TO COMMAND-OPTION-2.
```

```

MOVE      INPUT-KEY TO RECORD-BUFFER.
CALL      'ADABAS' USING
          CONTROL-BLOCK, FORMAT-BUFFER, RECORD-BUFFER.
IF RESPONSE-CODE = 0
    GO TO EX3C.
IF RESPONSE-CODE = 9
    GO TO EX3D.
*** Error Routine
EX3ERR.
*         . DISPLAY ERROR MESSAGE
*         . TERMINATE USER PROGRAM
          . . .

```

ACB PL/I の例

このsectionでは、PL/I での Adabas ダイレクトコールの使用例を示します。各例では、[以前に定義した Adabas ファイル](#)を使用します。

```

/**** CONTROL BLOCK ****/
DCL 1    CONTROL_BLOCK,
      02  FILLER1                CHAR (2) INIT (' '),
      02  COMMAND_CODE           CHAR (2) INIT (' '),
      02  COMMAND_ID             CHAR (4) INIT (' '),
      02  FILE_NUMBER            BIN FIXED (15) INIT (0),
      02  RESPONSE_CODE          BIN FIXED (15) INIT (0),
      02  ISN                    BIN FIXED (31) INIT (0),
      02  ISN_LOWER_LIMIT        BIN FIXED (31) INIT (0),
      02  ISN_QUANTITY           BIN FIXED (31) INIT (0),
      02  FORMAT_BUFFER_LENGTH   BIN FIXED (15) INIT (100),
      02  RECORD_BUFFER_LENGTH   BIN FIXED (15) INIT (250),
      02  SEARCH_BUFFER_LENGTH   BIN FIXED (15) INIT (50),
      02  VALUE_BUFFER_LENGTH    BIN FIXED (15) INIT (100),
      02  ISN_BUFFER_LENGTH      BIN FIXED (15) INIT (20),
      02  COMMAND_OPTION_1       CHAR(1) INIT (' '),
      02  COMMAND_OPTION_2       CHAR(1) INIT (' '),
      02  ADDITIONS_1            CHAR(8) INIT (' '),
      02  ADDITIONS_2            CHAR(4) INIT (' '),
      02  ADDITIONS_3            CHAR(8) INIT (' '),
      02  ADDITIONS_4            CHAR(8) INIT (' '),
      02  ADDITIONS_5            CHAR(8) INIT (' '),
      02  COMMAND_TIME           BIN FIXED (31) INIT (0),
      02  USER_AREA              CHAR(4) INIT (' ');

/**** USER BUFFER AREAS ****/
DCL FORMAT_BUFFER    CHAR(100),
      RECORD_BUFFER   CHAR(250),
      SEARCH_BUFFER   CHAR(50),

```

```
VALUE_BUFFER      CHAR(100),
ISN_BUFFER         CHAR(20);
*
*
/****   ADDITIONAL FIELDS USED IN THE EXAMPLES   ****/
DCL
  COMM_ID_X  BIN FIXED(31);
  COMM_ID    CHAR(4) BASED (ADDR(COMM_ID_X));
DCL  INPUT_KEY CHAR(8);
DCL  SYNC_CHECK_SWITCH CHAR(1) INIT('0');
DCL 1 RECORD_BUFFER_EX2,
    2  RECORD_BUFFER_A  CHAR(8),
    2  RECORD_BUFFER_B  DEC FIXED(3,0),
    2  FILLER3 CHAR(240);
DCL 1 RECORD_BUFFER_EX3,
    2  OPEN_RECORD_BUFFER,
    3  OPEN_RECORD_BUFFER_X CHAR(8),
    3  FILLER4 BIN FIXED(31),
    2  FILLER5 CHAR(18),
    2  UPDATED_XC CHAR(6),
    2  LAST_XD CHAR(8),
    2  FILLER6 CHAR(5),
    1  USER_DATA,
    2  RESTART_XD CHAR(8),
    2  RESTART_ISN BIN FIXED(31);
DCL  ADABAS ENTRY OPTIONS(ASM);
```

このsectionでは、次の例を記載しています。

- 例 1
- 例 2
- 例 3

例 1

- XB=99 であるファイル 2 のレコードを検索します。
- GET NEXT オプションを使って選択した各レコードを読みます。

オープンコマンドの発行

```
*** Issue Open Command **/  
EXMP1:  
  COMMAND_CODE = 'OP';  
  RECORD_BUFFER = 'ACC.';  
  CALL ADABAS (CONTROL_BLOCK,FORMAT_BUFFER,RECORD_BUFFER);  
  IF RESPONSE_CODE > 0      THEN GOTO EX1ERR;
```

検索コマンドの発行

```
/** Issue Find Command **/  
  COMMAND_CODE = 'S1';  
  COMMAND_ID = 'S101';  
  FILE_NUMBER = 2;  
  ISN_LOWER_LIMIT = 0;  
  ISN_BUFFER_LENGTH = 0;  
  FORMAT_BUFFER = '.';  
  SEARCH_BUFFER = 'XB,3,U.';  
  VALUE_BUFFER = '099';  
  CALL ADABAS (CONTROL_BLOCK, FORMAT_BUFFER,  
              RECORD_BUFFER, SEARCH_BUFFER, VALUE_BUFFER);  
  IF RESPONSE_CODE > 0 THEN GOTO EX1ERR;  
EX1A:  
  IF ISN_QUANTITY = 0 THEN GOTO EX1EXIT;  
EX1B:  
  COMMAND_CODE = 'L1';  
  ISN = 0;  
  COMMAND_OPTION_1 = 'N';  
  FORMAT_BUFFER = 'RG.';  
EX1C:  
  CALL ADABAS (CONTROL_BLOCK,FORMAT_BUFFER,RECORD_BUFFER);  
  IF RESPONSE_CODE = 0 THEN  
    GOTO EX1D;  
  IF RESPONSE_CODE = 3 THEN  
    GOTO EX1EXIT;  
EX1D:  
  . . .PROCESS RECORD . . .  
  GOTO EX1C;
```

エラールーチン

```
/** Error Routine **/  
EX1ERR:  
/* . DISPLAY ERROR MESSAGE */  
/* . TERMINATE USER PROGRAM */
```

クローズコマンドの発行


```

/** Issue Close Command **/
EX1EXIT:
  COMMAND_CODE = 'CL';
  CALL ADABAS (CONTROL_BLOCK);
  IF RESPONSE_CODE > 0 THEN
    GOTO EX1ERR;

```

例 2

- ファイル 1 のレコードは、物理順に全部読み込まれます。
- 読み込んだ各レコードを次の値で更新します。
 - フィールド AA = ABCDEFGH
 - フィールド AB = 500
- ユーザーは、ファイル 1 の排他制御権限を持つことが前提です。

オープンコマンドの発行

```

/** Issue Open Command ***/
EXMP2:
  COMMAND_CODE = 'OP';
  RECORD_BUFFER = 'EXU=1.';
  CALL ADABAS (CONTROL_BLOCK,FORMAT_BUFFER,RECORD_BUFFER);
  IF RESPONSE_CODE > 0 THEN GOTO EX2ERR;

```

物理読み込みシーケンシャルコマンドの発行

```

/** Issue Read Physical Seq. Command ***/
EX2A:
  COMMAND_ID = 'L201';
  FILE_NUMBER = 1;
  ISN = 0;
  FORMAT_BUFFER = 'GA.';
EX2B:
  COMMAND_CODE = 'L2';
  CALL ADABAS (CONTROL_BLOCK,FORMAT_BUFFER,RECORD_BUFFER);
  IF RESPONSE_CODE = 0 THEN GOTO EX2C;
  IF RESPONSE_CODE = 3 THEN GOTO EX2EXIT;
  GOTO EX2ERR;

```

レコードの更新

```

/**** Update record. ****/
/* Same fields are to be updated as were read. */
/* Same CID and FORMAT BUFFER can be used for update. */
/* ISN of record to be updated is already in ISN field as a result of */
/* the L2 command. */
EX2C:
    COMMAND_CODE = 'A1';
    RECORD_BUFFER_A = 'ABCDEFGH';
    RECORD_BUFFER_B = 500;
    CALL ADABAS (CONTROL_BLOCK,FORMAT_BUFFER,
                RECORD_BUFFER_EX2);
    IF RESPONSE_CODE > 0 THEN    GOTO EX2ERR;
    GOTO EX2B;

```

エラールーチン

```

/**** Error Routine ****/
EX2ERR:
/* . DISPLAY ERROR MESSAGE */
/* . TERMINATE USER PROGRAM */

```

ユーザーセッションのクローズ

```

/* Close User Session */
EX2EXIT:
    COMMAND_CODE = 'CL';
    CALL ADABAS (CONTROL_BLOCK);
    IF RESPONSE_CODE > 0 THEN    GOTO EX2ERR;

```

例 3

この例は、ET ロジックでのユーザーセッションを示します。ユーザープログラムは次の機能を実行します。

1. ユーザーセッションの初期化中に、前回のユーザーセッションで最後に正常に処理したトランザクションに関する情報を表示します。
2. ユーザートランザクションごとに次のことを行います。
 - ファイル 1 と 2 の更新用のキーとして使用される 8 文字の入力を端末から受け入れます。
 - ファイル 1 に対して FIND コマンドを発行し、フィールド AA= 入力キーのレコードが存在するかどうか判断します。
 - レコードが 1 件も見つからなかった場合は、メッセージを発行します。
 - レコードが見つかった場合：
 - ファイル 1 からレコードを削除します。
 - ファイル 2 に新しいレコードを追加します。フィールド RA= 入力キーを入力し、その他のフィールドは空値にします。

- レコードを正常に追加できない場合は、BT コマンドを発行して、エラーメッセージを表示します。
- 更新が正常に終了した場合は、ET コマンドを発行します。

セッションの初期化

プログラムのこのセクションは、ユーザーセッションの初期化中にのみ実行されます。

- OP コマンドが発行されるときに、前回のセッションの ET データが読み込まれます。
- メッセージが端末画面に表示され、前回のユーザーセッションで正常に完了した最後のトランザクションが通知されます。

```

EX3:
  COMMAND_CODE = 'OP';
  COMMAND_OPTION_2 = 'E';
  ADDITIONS_1 = 'USER0003';
  ADDITIONS_3 = 'PASSWORD';
  RECORD_BUFFER = 'UPD=1,2.';
  CALL ADABAS (CONTROL_BLOCK,FORMAT_BUFFER,RECORD_BUFFER);
  IF RESPONSE_CODE = 9 THEN GOTO EX3;
  IF RESPONSE_CODE > 0 THEN
    GOTO EX3ERR;
EX3A:
  COMM_ID = COMMAND_ID;
  IF COMM_ID_X = 0 THEN
    GOTO EX3B;
/* Display ET data (contained in RECORD BUFFER) on screen to inform user of
   last successfully processed transaction of previous user session. */
  . . .DISPLAY ET DATA. . .
  GOTO EX3C;
EX3B:
/*                                     */
/**** No ET data received. */
/* Display message that no transactions were successfully processed during
   the previous user session. */
  . . .DISPLAY MESSAGE . . .
/*                                     */
/**** Transaction processing. ****/
/* This section is executed for each user transaction. */
EX3C:
  . . .ACCEPT INPUT FROM TERMINAL. . .
/*                                     */
/* Issue Find command for file 1 to determine if rec exists with field AA
   equal to input key entered. */
EX3D:
  COMMAND_CODE = 'S4';
  COMMAND_ID = ' ';
  FILE_NUMBER = 1;
  ISN_LOWER_LIMIT = 0;
  FORMAT_BUFFER = '.';

```

```
SEARCH_BUFFER = 'AA.';
VALUE_BUFFER = INPUT_KEY;
CALL ADABAS (CONTROL_BLOCK,FORMAT_BUFFER,RECORD_BUFFER,
SEARCH_BUFFER,VALUE_BUFFER,ISN_BUFFER);
IF RESPONSE_CODE = 0 THEN
    GOTO EX3E;
GOTO EX3ERR;
EX3E:
    IF ISN_QUANTITY > 0 THEN
        GOTO EX3F;
/*
/* No record found, issue message requesting correction. */
    . . .ISSUE MESSAGE . . .
    GOTO EX3C;
/*
/* Delete record from file 1. */
/* ISN of record to be deleted is already in ISN field and in hold
status
as a result of the S4 command. */
EX3F:
    COMMAND_CODE = 'E4';
    CALL ADABAS (CONTROL_BLOCK);
    IF RESPONSE_CODE = 0 THEN
        GOTO EX3G;
    IF RESPONSE_CODE = 9 THEN
        GOTO EX3D;
    GOTO EX3ERR;
/**Add new record to file 2. */
EX3G:
    COMMAND_CODE = 'N1';
    FILE_NUMBER = 2;
    FORMAT_BUFFER = 'RA.';
    RECORD_BUFFER = INPUT_KEY;
    CALL ADABAS (CONTROL_BLOCK,FORMAT_BUFFER,RECORD_BUFFER);
    IF RESPONSE_CODE = 0 THEN
        GOTO EX3I;
    IF RESPONSE_CODE = 9 THEN
        GOTO EX3D;
/*
/* Attempt to add new record not successful. Backout transaction and
notify
user that error condition exists. */
    COMMAND_CODE = 'BT';
    CALL ADABAS (CONTROL_BLOCK);
    IF RESPONSE_CODE = 0 THEN
        GOTO EX3H;
/*
/* Backout not successful. */
    . . .ISSUE MESSAGE INDICATING BACKOUT NOT SUCCESSFUL . .
    GO TO EX3ERR.
/*
EX3H:
```

```

/**** Backout successful.      ****/
/* Issue message indicating error condition detected while adding new
record.*/
    . . .ISSUE MESSAGE. . .
    GOTO EX3ERR;
/*                                     */
/**** Updates successfully executed. ****/
/* Issue ET command with ET data.    */
EX3I:
    COMMAND_CODE = 'ET';
    COMMAND_OPTION_2 = 'E';
    RECORD_BUFFER = INPUT_KEY;
    CALL ADABAS (CONTROL_BLOCK,FORMAT_BUFFER,RECORD_BUFFER);
    IF RESPONSE_CODE = 0 THEN
        GOTO EX3C;
    IF RESPONSE_CODE = 9 THEN
        GOTO EX3D;
/*                                     */
/**** Error Routine      ****/
EX3ERR:
/* . DISPLAY ERROR MESSAGE */
/* . TERMINATE USER PROGRAM */
    . . .

```

ACB Fortran の例

このsectionでは、FORTRAN での Adabas ダイレクトコールの使用例を示します。各例では、以前に定義した **Adabas ファイル** を使用します。

```

C   **** CONTROL BLOCK ****
INTEGER*4      CB(20),CID,ISN,ISNL,ISNQ
INTEGER*4      ADD1(2),ADD2,ADD3(2),ADD4(2),ADD5(2)
INTEGER*4      CTIME,UAREA
INTEGER*2      CBI(40),CCODE,FNR,RC,FBL,RBL,SBL,VBL,IBL
LOGICAL*1      CBL(80),COPT1,COPT2
EQUIVALENCE    (CB(1),CBI(1),CBL(1))
EQUIVALENCE    (CID,CB(2)),(ISN,CB(4))
EQUIVALENCE    (ISNL,CB(5)),(ISNQ,CB(6))
EQUIVALENCE    (ADD1(1),CB(10)),(ADD2,CB(12)),(ADD3(1),CB(13))
EQUIVALENCE    (ADD4(1),CB(15)),(ADD5(1),CB(17))
EQUIVALENCE    (CTIME,CB(19)),(UAREA,CB(20))
EQUIVALENCE    (CCODE,CBI(2)),(FNR,CBI(5)),(RC,CBI(6))
EQUIVALENCE    (FBL,CBI(13)),(RBL,CBI(14)),(SBL,CBI(15))
EQUIVALENCE    (VBL,CBI(16)),(IBL,CBI(17))
EQUIVALENCE    (COPT1,CBL(35)),(COPT2,CBL(36))

C   **** USER BUFFER AREAS ****

```

```

INTEGER*4 FB(25),RB(50),SB(10),VB(10),IB(50)
*
*
C   *** ADDITIONAL FIELDS USED IN THIS EXAMPLE ***
LOGICAL*1 BLANK/1H /,COPH/1HH/,PERIOD/1H./,COPN/1HN/
INTEGER*2 S1/2HS1/,L1/2HL1/,CL/2HCL/
INTEGER*4 CID1/4HS101/,FB1/4H. /,FB2/4HRG. /,SB1/4HXB,3/
INTEGER*4 SB2/4H,U. /,VB1/4H099 /

```

このsectionでは、次の例を記載しています。

■ 例 1

例 1

- XB=99 であるファイル 2 のレコードを検索します。
- GET NEXT オプションを使って選択した各レコードを読みます。

コントロールブロックの初期化

```

c*** Initialize Control Block
      DO 5 I=1,80
        CBL(I)=BLANK
5     CONTINUE
      DO 10 I=3,6
        CB(I)=0
10    CONTINUE
      CBI(13)=100
      CBI(14)=200
      CBI(15)=40
      CBI(16)=40
      CBI(17)=200
      CBI(19)=0

```

検索コマンドの発行

```

c*** Issue FIND Command
      CCODE=S1
      CID=CID1
      FNR=2
      ISNL=0
      COPT1=COPH
      FB(1)=FB1
      SB(1)=SB1
      SB(2)=SB2
      VB(1)=VB1
      CALL ADABAS(CB,FB,RB,SB,VB,IB)
      IF(RC.NE.0) GO TO 50
      IF(ISNQ.EQ.0) GO TO 100

```

選択した各レコードの読み取り

```
c***Read Each Record Selected
15  CONTINUE
    CCODE=L1
    ISN=0
    COPT1=COPN
    FB(1)=FB2
    CALL ADABAS(CB,FB,RB)
    IF(RC.EQ.0) GO TO 30
    IF(RC.EQ.3) GO TO 100
    PRINT 60,RC,CCODE
60  FORMAT(1H0,'ADABAS ERROR CODE',I4,' FROM '.A2,' COMMAND')
    GO TO 50
30  CONTINUE
C   ...PROCESS RECORD...
    GO TO 15
50  CONTINUE
    STOP
100 CONTINUE
    ...
```


索引

A

- A1 コマンド, 181
 - ACB の例, 186
 - ACB フィールドの説明, 184
 - ACB レイアウト, 182
 - ACBX フィールドの説明, 190
 - ACBX レイアウト, 188
 - 機能および使用方法, 182
 - その他の考慮事項, 193
 - バッファ記述, 193
- ABD (参照 Adabas バッファ記述 (ABD))
- ACB (参照 Adabas コントロールブロック (ACB))
- ACBX (参照 拡張 Adabas コントロールブロック)
- Adabas
 - コントロールブロックの構造, 59
- Adabas コントロールブロック (ACB)
 - ACBX との比較, 77
 - DSECT, 67
 - 説明, 59, 61
 - フィールド, 62
 - フォーマット, 61
 - 例, 67
- Adabas バッファ記述 (ABD)
 - DSECT, 91
 - 説明, 85
 - フィールド, 88
 - フォーマット, 87
- Adabas ユーザー ID, 21

B

- BT コマンド, 199
 - ACB の例, 203
 - ACB フィールドの説明, 202
 - ACB レイアウト, 201
 - ACBX フィールドの説明, 206
 - ACBX レイアウト, 205
 - ISN バッファの要求, 208
 - 機能および使用方法, 200
 - 説明, 19
 - マルチフェッチ処理, 46

C

- C インジケータ, 109
- C1 コマンド, 209
 - ACB の例, 212

- ACB フィールドの説明, 212
- ACB レイアウト, 211
- ACBX フィールドの説明, 214
- ACBX レイアウト, 213
- 機能および使用方法, 210

- C3 コマンド, 217
 - ACB の例, 220
 - ACB フィールドの説明, 220
 - ACB レイアウト, 218
 - ACBX フィールドの説明, 222
 - ACBX レイアウト, 221
 - 機能および使用方法, 218
 - レコードバッファの要求, 223
- C5 コマンド, 225
 - ACB の例, 228
 - ACB フィールドの説明, 227
 - ACB レイアウト, 226
 - ACBX フィールドの説明, 230
 - ACBX レイアウト, 229
 - 機能および使用方法, 226
 - レコードバッファの要求, 232
- CL コマンド, 233
 - ACB の例, 237
 - ACB フィールドの説明
 - フィールドの説明, 236
 - ACB レイアウト, 235
 - ACBX フィールドの説明
 - フィールドの説明, 239
 - ACBX レイアウト, 238
 - 機能および使用方法, 234
 - レコードバッファの要求, 241

D

- DSECT
 - ABD, 91
 - ACB, 67
 - ACBX, 77

E

- E1 コマンド, 243
 - ACB の例, 247
 - ACB フィールドの説明, 246
 - ACB レイアウト, 245
 - ACBX フィールドの説明, 250
 - ACBX レイアウト, 249
 - 機能および使用方法, 244
- ET コマンド, 253
 - ACB の例, 258

ACB フィールドの説明, 256
ACB レイアウト, 255
ACBX フィールドの説明, 260
ACBX レイアウト, 259
機能および使用方法, 254
説明, 20
バッファ要件, 262

H

HI コマンド, 265
ACB の例, 268
ACB フィールドの説明, 267
ACB レイアウト
レイアウト, 266
ACBX フィールドの説明, 270
ACBX レイアウト
レイアウト, 269
機能および使用方法, 266

I

ISN バッファ
説明, 173
ISN リスト
処理, 37
処理するコマンド, 473
処理例, 39
ソート, 485
ワークからの取得, 37
ワークへの格納, 37
ISN リストのソート, 485

L

L インジケータ, 110
L1/L4 コマンド, 273
ACB の例, 280
ACB フィールドの説明, 276
ACB レイアウト
レイアウト, 275
ACBX フィールドの説明, 286
ACBX レイアウト
レイアウト, 285
ISN バッファの要件, 291
機能および使用方法, 274
バッファ要件, 290
フォーマットバッファの要件, 290
マルチフェッチバッファの要件, 291
レコードバッファの要件, 291
L2/L5 コマンド, 293
ACB の例, 299
ACB フィールドの説明, 296
ACB レイアウト, 295
ACBX フィールドの説明, 302
ACBX レイアウト, 301
機能および使用方法, 294
その他の考慮事項, 306
バッファ要件, 305
L3/L6 コマンド, 307
ACB の例, 316
ACB フィールドの説明, 311
ACB レイアウト, 310
ACBX フィールドの説明, 321

ACBX レイアウト, 320
機能および使用方法, 308
コントロールブロック
バッファの説明, 326
その他の考慮事項, 331
L9 コマンド, 333
ACB の例, 340
ACB フィールドの説明, 336
ACB レイアウト, 335
ACBX フィールドの説明, 344
ACBX レイアウト, 343
機能および使用方法, 334
コントロールブロック
バッファの説明, 348
その他の考慮事項, 352
LF コマンド, 353
ACB の例, 357
ACB フィールドの説明, 355
ACB レイアウト, 354
ACBX フィールドの説明, 358
ACBX レイアウト, 357
機能および使用方法, 354
レコードバッファの要件, 360

N

N インジケータ, 113
N1/N2 コマンド, 369
ACB の例, 374
ACB フィールドの説明, 372
ACB レイアウト, 370
ACBX フィールドの説明, 377
ACBX レイアウト, 376
機能および使用方法, 370
バッファ要件, 380

O

OP コマンド, 385
ACB の例, 393
ACB フィールドの説明, 389
ACB レイアウト, 388
ACBX フィールドの説明, 397
ACBX レイアウト, 395
機能および使用方法, 386
コントロールブロック
フィールドに返される値, 404
タイムリミットの超過, 404
バッファ, 401
フォーマットバッファの要件, 401
ユーザーキューエレメント, 403
ユーザータイプ, 387
レコードバッファの要件, 401

P

PLOG (参照 プロテクションログ (PLOG))

R

RC コマンド, 407
ACB の例, 412
ACB フィールドの説明, 410
ACB レイアウト, 409

ACBX フィールドの説明, 415
 ACBX レイアウト, 414
 機能および使用方法, 408
 RE コマンド, 419
 ACB の例, 424
 ACB フィールドの説明, 422
 ACB レイアウト, 421
 ACBX フィールドの説明, 426
 ACBX レイアウト, 425
 機能および使用方法, 420
 バッファ, 428
 フォーマットバッファ, 429
 レコードバッファ, 429
 RI コマンド, 431
 ACB の例, 434
 ACB フィールドの説明, 433
 ACB レイアウト, 432
 ACBX フィールドの説明, 436
 ACBX レイアウト, 435
 機能および使用方法, 432

S

S インジケータ, 115
 S1、S2、S4 コマンド, 439
 ACB の例, 447
 ACB フィールドの説明, 443
 ACB レイアウトブロック, 441
 ACBX フィールドの説明, 455
 ACBX レイアウトブロック, 454
 機能および使用方法, 440
 バッファ記述, 459
 S5 コマンド, 461
 ACB の例, 466
 ACB フィールドの説明, 464
 ACB レイアウト, 462
 ACBX フィールドの説明, 468
 ACBX レイアウト, 466
 ISN バッファ, 471
 機能および使用方法, 462
 サーチバッファ, 471
 バッファ, 470
 バリュースearchバッファ, 471
 フォーマットバッファ, 471
 レコードバッファ, 471
 S8 コマンド, 473
 ACB の例, 478
 ACB フィールドの説明, 476
 ACB レイアウト, 475
 ACBX フィールドの説明, 480
 ACBX レイアウト, 479
 ISN バッファ, 484
 機能および使用方法, 474
 サーチバッファ, 483
 バリュースearchバッファ, 483
 フォーマットバッファ, 483
 レコードバッファ, 483
 S9 コマンド, 485
 ACB の例, 491
 ACB フィールドの説明, 488
 ACB レイアウト, 487
 ACBX フィールドの説明, 494
 ACBX レイアウト, 492
 ISN バッファ, 498

機能および使用方法, 486
 サーチバッファ, 498
 バッファ記述, 497
 バリュースearchバッファ, 498
 フォーマットバッファ, 498
 レコードバッファ, 498
 SYNX-03 チェックポイント, 217
 SYNX-03 チェックポイントの書き込み, 217

あ

圧縮形式
 読み込むフィールドの指定, 104

お

大文字, 3
 オープン
 ユーザーセッション, 385

か

解放
 グローバルフォーマット ID, 407
 コマンド ID, 407
 角カッコ ([]), 3
 拡張 Adabas コントロールブロック (ACBX)
 ACB との比較, 77
 DSECT, 77
 説明, 59, 67
 フィールド, 69
 フォーマット, 68

き

競合更新, 22

く

繰り返しの構文要素, 4
 グローバルフォーマット ID
 解放, 407

こ

更新
 A1 コマンドを使用するフィールド値, 181
 競合, 22
 排他制御, 25
 ホールドオプションの使用, 23
 構文形式
 大文字, 3
 角カッコ ([]), 3
 区切り文字と記号, 4
 繰り返しの要素, 4
 小文字, 3
 サブパラメータ, 4
 斜体, 3
 省略記号 (...), 4
 字下げ, 4
 選択型の要素, 3
 相互排他的な選択肢, 3
 縦線 (|), 3

中カッコ ({}), 3
 必須の要素, 3
 標準フォント, 3
 太字, 3
 構文中での選択肢, 3
 構文中の区切り文字と記号, 4
 コマンド ID
 ISN リストとの併用, 34
 解放, 35, 407
 固有のフォーマット ID との併用, 33
 使用例, 35
 自動生成, 35
 定義, 30
 内部識別, 35
 読み込み、更新、および検索コマンド, 32
 読み込みシーケンシャルコマンド, 32
 小文字, 3
 コントロールブロック
 説明, 59
 ロギング, 82

さ

削除
 レコード, 243
 サブパラメータの構文, 4
 サーチバッファ
 検索式, 147
 検索式、シグニフィカンスおよび空値インジケータ, 148
 検索式結合演算子, 151
 検索式とオカレンスインデックス, 148
 検索式とフィールドタイプ, 147
 検索式の値演算子, 150
 検索式フィールドの長さ, 149
 検索式フィールドのフォーマット, 149
 構文, 147
 説明, 145
 ソフトカップリングの使用, 155
 物理カップリングファイルの使用, 153
 例, 161

し

斜体, 3
 終了
 トランザクション, 253
 省略記号 (...), 4
 字下げ, 4
 自動バックアウト
 ET コマンドとの併用, 19

す

スーパーディスクリプタ
 単一ファイルの検索回数, 148

せ

セッション
 ユーザーセッションのオープン, 385
 選択型の構文要素, 3

た

タイムアウトコントロール
 トランザクションリミット, 18
 非アクティビティリミット, 26
 ユーザーが非アクティビティリミットを超えたときに実行される処理, 26
 縦線 (|), 3
 ダイレクトコール
 ABD DSECT, 91
 ABD フィールド, 88
 ABD フォーマット, 87
 ACB DSECT, 67
 ACB の例, 67
 ACB フィールド, 62
 ACB フォーマット, 61
 ACBX DSECT, 77
 ACBX の記述, 67
 ACBX フィールド, 69
 ACBX フォーマット, 68
 Adabas コントロールブロック (ACB), 61
 Adabas バッファ記述 (ABD), 85
 COBOL ACB の例, 511
 FORTRAN ACB の例, 525
 PL/I ACB の例, 518
 アセンブラ ACB の例, 504
 アソシエータ読み込みコマンド, 12
 概要, 7, 93
 コマンド, 179
 コントロールブロックの構造, 59
 コントロールブロックのロギング, 82
 コール手順, 51
 タイプ, 8
 データストレージ読み込みコマンド, 10
 データベース更新コマンド, 13
 データベース問い合わせコマンド, 8
 特殊コマンド, 15
 プログラミングの考慮事項, 29
 論理トランザクション制御コマンド, 14

ち

チェックポイント
 SYNX-03 の書き込み, 217
 書き込み, 209
 チェックポイントの書き込み, 209
 中カッコ ({}), 3

て

ディスクリプタ
 値範囲の決定, 334
 デッドロック
 検出, 25
 リソースの制御, 24

と

トランザクション
 ET ロジックプログラムのタイムリミット, 18
 終了, 253
 終了するコマンド, 20
 シーケンス番号, 18
 自動バックアウトルーチン, 19

バックアウト, 199
 バックアウトするコマンド, 19
 論理の定義, 18
 トランザクションのバックアウト, 199

は

排他ファイル制御
 更新, 25
 バリュースタック
 SQL 空値とインジケータ, 158
 説明, 157
 符号の取り扱い, 159
 例, 161
 パフォーマンス
 フォーマットバッファに関する考慮事項, 123
 パフォーマンスバッファ
 説明, 177

ひ

非アクティビティタイムリミット, 26
 必須の構文要素, 3
 標準フォント, 3

ふ

ファイル
 アクセスの制御, 403
 ディスクリプタによる論理順での読み込み, 307
 リフレッシュ, 243
 ファイルのリフレッシュ, 243
 フィールド
 値の変更, 182
 読み込み, 353
 フィールド値の変更, 181
 フォーマット ID
 グローバル, 30
 グローバルに使用, 33
 固有のコマンド ID との併用, 33
 定義, 30
 読み込み、更新、および検索コマンド, 32
 フォーマットバッファ
 SQL 有意性インジケータ (S) , 115
 アスタリスクを使った長さ表記, 118
 インデックスまたは範囲の表記, 105
 カウントインジケータ (C) , 109
 構文, 100
 最高オカレンス/値インジケータ (N) , 113
 スペース表記 (nX) , 122
 説明, 99
 テキスト挿入表記, 122
 長さインジケータ (C) , 110
 パフォーマンスの考慮事項, 123
 ピリオディックグループ参照, 105
 ピリオディックグループ内のマルチプルバリュースタック参照, 108
 フィールド構文, 103
 フィールド長およびデータフォーマット, 117
 フィールドのタイプ, 104
 フィールドの連続表記, 121
 フィールド名の指定, 104
 フォーマット選択条件の構文, 101
 編集マスク記号, 119

マルチプルバリュースタック参照, 106
 例, 131
 レコードフォーマットの指定, 102
 ロング英数字 (LA) オプションが指定されている英数字
 フィールド, 123

太字, 3

プリフェッチ
 コマンドオプション P を指定した呼び出し, 48
 使用するタイミング, 47
 ファイルおよびコマンドの除外, 47
 プログラミングの考慮事項, 48
 マルチフェッチとの比較, 43
 目的, 42
 呼び出し, 43
 プリフェッチバッファ
 説明, 141
 プロテクションログ (PLOG)
 ユーザーデータの書き込み, 225

ほ

ホールド
 更新用レコード, 265
 レコード, 273
 ホールド機能
 更新, 23
 コマンド, 23
 ホールド状態からの解放, 24
 レコード/ISN, 22

ま

マルチクライアントファイル
 オーナー ID
 ダイレクトコールコマンドによる指定, 391, 399
 マルチフェッチ
 ET/BT コマンド処理, 46
 プリフェッチとの比較, 43
 目的, 42
 呼び出し, 43
 読み込みコマンド処理, 45
 マルチフェッチバッファ
 説明, 143

ゆ

ユーザーセッション
 オープン, 385
 クローズ, 233
 ユーザーセッションのクローズ, 233
 ユーザーデータ
 格納するためのダイレクトコールコマンド, 21
 機能, 21
 プロテクションログへの書き込み, 225
 読み込み
 ダイレクトコールコマンドの使用, 419
 読み込むためのダイレクトコールコマンド, 21
 ユーザーバッファ
 説明, 175

よ

読み込み
 レコード, 273

読み込みコマンド
マルチフェッチ処理, 45

り

リソースのデッドロック
回避, 24

れ

レコード
解放
 ダイレクトコールコマンドの使用, 431
 カップリングの検索, 462
検索
 ダイレクトコールコマンドの使用, 439
更新時のホールド, 265
削除, 243
追加, 369
物理順での読み込み, 293
読み込み, 273
読み込みとホールド, 273
レコードの更新
 更新時のホールド, 265
レコードバッファ
 SQL 空値インジケータ, 127
 説明, 125
 例, 131
 ロング英数字 (LA) オプションが指定されている英数字
 フィールド, 129

ろ

論理トランザクション, 18