

**Adabas**

**Glossary**

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

Glossary .....	1
英数字 .....	1
あ行 .....	5
か行 .....	6
さ行 .....	8
た行 .....	10
な行 .....	13
は行 .....	14
ま行 .....	19
や行 .....	20
ら行 .....	20
わ行 .....	21
目次 .....	23



## Glossary

---

用語集には、次の文字で始まる用語が登録されています。

### 英数字

---

- 2 フェーズコミット処理      2 フェーズコミット処理では、個別に管理されているリソース間のトランザクションを一括して反映または廃棄して、業務トランザクションの整合性を維持します。
- ABD リスト      ABD リストは、Adabas ダイレクトコールに使用する [Adabas バッファ記述](#) へのポインタを格納したファイルです。ABD リストは、[ACBX ダイレクトコールインターフェイス](#) を使ってダイレクトコールを実行した場合にのみ使用できます。
- 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。
- ACB インターフェイスダイレクトコール      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 リリースに導入されている従来からのダイレクトコールインターフェイスです。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。
- Adabas ダイレクトコールについては、[Adabas コマンドリファレンス Adabas の呼び出し](#)」を参照してください。
- ACBX インターフェイスダイレクトコール      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 ダイレクトコールについては、*Adabas コマンドリファレンスAdabas の呼び出し*を参照してください。
- Adabas セッション *Adabas* セッションは、ニュークリアスが起動したときに開始し、ニュークリアスが終了したときに終了します。*Adabas* ニュークリアスは、*Adabas* 起動 (ADARUN)、パラメータなどの、そのオペレーティングシステムに応じたジョブ制御により起動されます。
- Adabas ダイレクトアクセスメソッド (ADAM) *Adabas* ダイレクトアクセスメソッド (ADAM) 機能を使用すると、*インバーテッドリスト*にアクセスせずに、データストレージから直接レコードを取り出すことができます。レコードが格納されているデータストレージのブロック番号は、レコードの ADAM キーに基づきランダムマイジングアルゴリズムを使用して計算されます。ADAM は、アプリケーションプログラム、クエリおよびレポートライタの機能に対して完全に透過的に使用されます。詳細については、「*Adabas* ダイレクトアクセスメソッド (ADAM) を使用したランダムアクセス」を参照してください。
- Adabas トリガドライバ *Adabas* ニュークリアスの中で、*Adabas* トリガとストアードプロシージャ機能を全般的に制御する部分。トリガやストアードプロシージャの要求を検出すると、Natural トリガドライバを初期化して要求を実行します。
- Adabas バッファ記述 (ABD) *Adabas* バッファ記述 (ABD) は、ダイレクトコールで使用されるバッファセグメントを定義します。具体的な定義内容は、バッファタイプ (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、パフォーマンス、またはユーザー)、位置、サイズなどのバッファ固有の情報です。バッファ自体またはバッファへのポインタを ABD で定義することもできます。ABD の使用は、*Adabas* 8 以降の **ACBX** *ダイレクトコールインターフェイス* を使った *Adabas* ダイレクトコールにバッファが必要な場合に限定されます。**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は、*Adabas* ダイレクトコールにバッファが必要でも、ABD は使用できません。
- ABD の構造に関する詳細は、「*Adabas* バッファ記述」を参照してください。*Adabas* ダイレクトコールについては、「*Adabas* の呼び出し」を参照してください。
- RABN (相対 *Adabas* ブロック番号) *Adabas* ストレージに対する基礎的なアドレス処理。*Adabas* は、データ、アソシエータ、およびワークディスクスペースをデバイス依存の論理ブロックに分割します。この3つの各エリアのブロックには、RABN 1 から昇順に連続する番号が割り当てられます。データブロック自体とそのアドレスは、Software AG の資料では "RABN" という名称で統一されています。例えば、"*Adabas* が RABN 1~10 をアソシエータに割り当てる" という記述がある場

合、"1"から"10"までの番号が付いた Adabas ストレージブロックがアソシエータに割り当てられたこととなります。また、"Adabas が 50 個の RABN をアソシエータに割り当てる"という記述がある場合には、割り当てられるストレージのブロック数は 50 ですが、RABN の実際の番号はわかりません。

Adabas	Adabasは適応性に優れたデータベースで、データベースのパフォーマンスが重要な要素であるメインフレームプラットフォーム向けの高性能、マルチスレッド型のデータベース管理システムです。メインフレーム、ミッドレンジ、PCを含む複数の異なるプラットフォーム間で、優れた相互運用性、拡張性、移植性を発揮するデータベースです。
ADALINK	個々の TP モニタに固有な Adabas API (アプリケーションプログラムインターフェイス) の一部を示す一般的な名称。Adabas API は、アプリケーションプログラムを Adabas にリンクするときを使用します。実際のモジュール名は、使用する TP モニタに応じて異なります。例えば、バッチまたは TSO プログラムにリンクする場合のモジュール名は ADALNK ですが、CICS の場合は ADALNC です。Adalink という用語は、指定された環境に該当するモジュールを示します。ADALINK と ADALNK はどちらも同じ意味です。
ADARUN	<p>ADARUN コントロールステートメントを使って、Adabas 操作環境の定義を行い、Adabas を起動します。ADARUN コントロールステートメントでも、Adabas ユーティリティを起動できます。</p> <p>ADARUN の主な機能は次のとおりです。</p> <ul style="list-style-type: none"><li>■ すべてのデータベース I/O および他のオペレーティングシステム依存の機能を実行する ADAIOR モジュールをロードします。</li><li>■ ADARUN のパラメータステートメントを解釈した後、ADARUN パラメータの設定に従って必要な Adabas ニュークリアス、またはユーティリティモジュールのロードと変更を行います。</li><li>■ Adabas に制御を渡します。</li></ul> <p>ADARUN ステートメントは、通常、一連のエントリから構成され、それぞれのエントリは 1 つ以上の ADARUN パラメータ設定を示します。このステートメントは、DDCARD (z/OS、z/VM、BS2000) データセットまたは VSE CARD データセットに指定します。</p>
BUB	読み取り不可能なブロックのことです。プライマリの ASSO、DATA、および WORK の場合には、プライマリ ASSO RABN 2 とミラー ASSO RABN 9 に BUB が格納され、ミラーの ASSO、DATA、および WORK の場合には、プライマリ ASSO RABN 9 と

	ミラー ASSO RABN 2 に格納されます。また、PLOG <sub>n</sub> の場合には、プライマリとミラーの PLOG <sub>n</sub> RABN 1 に格納されます。
ID	ターゲット ID の省略形、Adabas コールをそのターゲットに導くために使用される固有の識別子です。
ID テーブル	1つのオペレーティングシステムの境界内にあるすべてのアクティブなターゲットに対する参照データリスト。ID テーブルは共通してアドレス可能なストレージに格納されます。
IIBS	アイソレート ID ビット文字列を表し、ID テーブルヘッダーに含まれる 256 ビット (32 バイト) の文字列です。各ビットは昇順にソートされた論理 ID に対応しています。ビットの値が "1" の場合は、該当する ID はアイソレートです。
LOB グループ	<a href="#">基本ファイル</a> と基本ファイルに関連付けられている <a href="#">LOB ファイル</a> の組み合わせを LOB グループと呼びます。
LOB ファイル	Adabas は、ラージオブジェクトフィールド値を LOB ファイルと呼ばれる個別のファイルに格納します。このファイルは、 <a href="#">基本ファイル</a> と呼ばれる LB フィールドを含むファイルと密接に関連付けられます。バックグラウンドでは Adabas が LB フィールド値 (非常に短いものを除く) を LOB ファイルに格納しますが、この処理はアプリケーションに対して透過的に行われます。アプリケーションプログラムのコマンドは常に基本ファイルに対して発行され、LOB ファイルの存在を意識する必要はありません。  詳細については、「ラージオブジェクト (LB) フィールドの基本」を参照してください。
MIRTAB	プライマリ RABN のステータスを示すミラーテーブル。ASSO、DATA、および WORK の場合は、プライマリとミラーの ASSO RABN 7 に MIRTAB が格納され、PLOG <sub>n</sub> の場合は、プライマリとミラーの PLOG <sub>n</sub> RABN 1 に格納されます。
Natural トリガドライバ	バッチ Natural 環境として動作する Natural ニュークリアスの構成要素。Natural トリガドライバは、トリガとストアードプロシージャを実行します。Adabas トリガドライバとともに動作して、Adabas ニュークリアスと、プロシージャを実行する Natural サブシステムとの処理間コミュニケーションを管理します。
STPLNK	ストアードプロシージャ要求の呼び出しに使用されるストアードプロシージャのリンクルーチン。
STPRBE	プロシージャからコールされるレコードバッファ抽出ルーチンの中で、コール元のルーチンから渡されるパラメータの取得に使用されます。

STPUTRAK プロシージャの起動要求とプロシージャの起動をすべて記録するルーチン。このルーチンを使用して、トレースメッセージの書き込みと、各サブシステムのトリガ処理の監査を行うことができます。STPUTRAK ルーチンは、Adabas トリガとストアードプロシージャ機能にデフォルトで用意されています。

## あ行

アイソレート ID 論理 ID としてユーザーが指定できるアイソレートターゲットの ID。アイソレート ID は、0 より大きく 256 未満である必要があります。アイソレート ID は、ターゲットをアドレスするための物理 ID として解釈されます。

アイソレートターゲット ユーザーに直接コールされるターゲット。

圧縮レコード Adabas では、数種類の圧縮オプションを使用して、データレコードを圧縮形式で保持します。データ圧縮により、データベース内のデータに必要なストレージの量が大幅に削減されます。また、1 回の物理的な転送で送信できる情報量が増加するため、I/O 効率が向上します。

Adabas のデータ圧縮の詳細については、「[圧縮](#)」を参照してください。

アップパーインデックス (UI) 必要に応じて、Adabas によって自動的に作成されるインバーテッドリストのインデックス。下位レベルのインデックスを管理します。アップパーインデックスは、検索効率を向上させるために、Adabas によって使用されます。管理対象の NI と同様に、最初のレベルの UI には、各インデックスブロックの 1 つのディスクリプタに対するエントリのみが含まれます。その他のすべての UI レベルには、各インデックスブロックのすべてのディスクリプタのエントリが含まれます。UI には最小サイズのスペースが必要です。最小サイズとは 2 ブロックです。

アドレスコンバータ Adabas は各データベースレコードを相対 Adabas ブロック番号 (RABN) で識別されたデータストレージブロックに格納します。各レコードの RABN は、アドレスコンバータと呼ばれるテーブルに保持されます。アドレスコンバータは、各データベースファイルに 1 つ存在し、アソシエータに格納されています。アドレスコンバータのエントリは、ISN 順に並べられます (つまり、最初のエントリは ISN 1 のデータの RABN 位置を表し、15 番目のエントリは ISN 15 のデータの RABN 位置を表します)。

スパンドレコードでは、セカンダリアドレスコンバータを使用して、セカンダリレコードが格納されているデータストレージブロックの RABN にセカンダリ ISN をマッピングします。

アドレススペース	プログラムのタスク／処理単位で割り当てられるストレージエリア。アドレススペースは、z/OS ではリージョン、VSE ではパーティション、BS2000 ではタスクです。このドキュメントでは、リージョンという用語を、パーティションやタスクの同義語として使用しています。
アンカーファイル	拡張ファイルでは、最も低い ISN 範囲がアンカーファイルに指定されます。アンカーファイルのファイル番号は、拡張ファイル全体に割り当てられる番号になります。  詳細については、「1つの論理ファイルへの物理ファイルのリンク」および「拡張ファイル」を参照してください。
インバーテッドリスト	ISN ではなくディスクリプタ値で構成されるリスト。このリストは、Adabas 検索コマンドを解決する場合およびレコードを論理順に読み込む場合に使用します。インバーテッドリストは、キーフィールドまたは ディスクリプタに指定した Adabas ファイルのフィールドごとに作成され、継続的に更新されます。このリストは、ノーマルインデックス (NI) および、14 個までのアップパーインデックス (UI) で構成されます。
エレメンタリフィールド	エレメンタリフィールドの値は、レコードごとに1つだけです。
オペレータコマンド	オペレータコマンドは、Adabas セッション中または Adabas ユーティリティのオペレーション中に入力されるものです。オペレータコマンドによって実行される操作の主な内容は次のとおりです。 <ul style="list-style-type: none"><li>■ Adabas セッションまたはユーザーセッションを終了させます。</li><li>■ ニュークリアスまたはユーティリティの情報を表示させます。</li><li>■ コマンド群を CLOG にロギングします。</li><li>■ Adabas のオペレーティングパラメータまたは状態を変更します。</li></ul>

## か行

---

拡張ファイル	アクセスするファイル数を減らすために、同一フォーマットのレコードをもった複数の物理ファイルをリンクして、1つの論理ファイルに結合することができます。このようなファイル構造のことを拡張ファイルと呼び、この拡張ファイルを構成する物理ファイルのことをコンポーネントファイルと呼びます。拡張ファイルは、それぞれに異なった論理 ISN の範囲を持った 128 までのコンポーネントファイルによって構成されます。1つの拡張ファイルのレコード数は、4,294,967,294 までです。
--------	--

---

	<p>詳細については、「<a href="#">1つの論理ファイルへの物理ファイルのリンク</a>」および「<a href="#">拡張ファイル</a>」を参照してください。</p>
仮想マシン (VM)	<p>z/VM エリア内のユーザージョブに割り当てられるストレージスペースのことです。</p>
関与トリガ	<p>関与トリガは同期型でなければなりません。このタイプのトリガが起動すると、関与トリガを起動した Adabas コマンドと同じユーザー ID が割り当てられているプロシージャが実行されます。したがって、このトリガはトランザクションのロジックに全面的に関与することになります。</p> <p>このタイプとは異なるトリガとして、<a href="#">非関与トリガ</a>があります。</p>
疑似シリンダ	<p>実際の DASD シリンダを持たない固定ブロックアドレス (FBA) デバイス上の論理シリンダ。</p>
基本ファイル	<p>ラージオブジェクト (LB) フィールドを扱う際に、Adabas は LB フィールド値を <a href="#">LOB ファイル</a> と呼ばれる個別のファイルに格納します。このファイルは、基本ファイルと呼ばれる LB フィールドを含むファイルと密接に関連付けられます。バックグラウンドでは Adabas が LB フィールド値 (非常に短いものを除く) を LOB ファイルに格納しますが、この処理はアプリケーションに対して透過的に行われます。アプリケーションプログラムのコマンドは常に基本ファイルに対して発行され、LOB ファイルの存在を意識する必要はありません。</p> <p>詳細については、「<a href="#">ラージオブジェクト (LB) フィールドの基本</a>」を参照してください。</p>
グローバルトランザクション	<p>グローバルトランザクションは、1つ以上のオペレーティングシステムイメージで複数のデータベース処理を制御している場合に、リソースの変更に伴って発生する処理の単位です。</p>
グローバルパラメータ	<p>いくつかの ADARUN パラメータはグローバルパラメータ、つまり、クラスタの全ニュークリアスに対して同じ値を持たなければなりません。また、グローバルパラメータの一部には、セッションの初期化時に設定され、変更することができないものがあります。</p>
グローバルフォーマット ID	<p>グローバルフォーマット ID は、デコード結果のフォーマットバッファに割り当てられる ID です。フォーマットバッファは、グローバルに1回保存されます。保存後には、そのフォーマットバッファをグローバルフォーマット ID で参照したり、再使用したりすることができます。グローバルフォーマット ID は、グローバルに1回保存されます。同じ Adabas ニュークリアスで実行中のすべてのユーザーは、このグローバルフォーマット ID を使用することがで</p>

きます。フォーマットバッファは、固有のフォーマットIDによって識別されます。

これとは違うタイプとして、**フォーマットID**があります。フォーマットIDは、デコード結果のフォーマットバッファをユーザーごとに1回保存し、そのユーザーのみが使用できます。

グローバルフォーマットIDは、Adabas コントロールブロックフィールド（ACBADD5 または ACBXADD5）に指定します。フォーマットIDの詳細は、「**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。

## コマンド ID

コマンドIDは、自動的に生成されたかユーザーが指定した、空白以外でゼロ以外の値です。コマンドIDは次の用途で使用されません。

- デコードされた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。
- 後からアクセスできるように、Sx コマンドによって生成された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。

コマンドIDは、Adabas コントロールブロックフィールド（ACBCID または ACBXCID）に指定します。コマンドIDの詳細は、「**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。

## コミュニケーター

オペレーティングシステム間で、リモートターゲットをアクセス可能にして通信するためのルーチン。EntireNet-Workはコミュニケーターの1つです。

# さ行

## サービス

Adabas コールを処理し、応答を発行するもの。Adabas ニュークリアスは、サービスの1つです。ターゲットも参照してください。

## 再開位置

**再開位置**とは、中間データセットから復元可能な実行処理の最終位置を意味します。

## サブタスク

親タスクから派生したタスク。

## ジェネラルコントロールブロック (GCB)

Adabas によって各データベースに割り当てられる2つのコントロールブロック。GCBは、データベースとその場所を示します。GCBはアソシエータの一部です。

準リエントラントプログラム	TP モニタ機能へのコール間にコードを変更できるプログラム。モニタ機能呼び出す場合、すべてのユーザーデータは TP モニタシステムから取得された特別なワークエリアに保存しなければなりません。
ストアドプロシージャ	ストアドプロシージャは、Adabas によって実行されるプロシージャですが、アプリケーションから直接起動することも、手動で起動することもできます。
スパンドレコード	<p>Adabas バージョン 8 (以降) では、スパンドレコードの概念が導入されています。データベース内では、論理レコードは多くの物理レコードに分割されます。それぞれのレコードは単一のデータストレージ (DS) ブロックに格納されます。分割された各物理レコードには、それぞれ ISN が割り当てられます。スパンドレコードは、1つのプライマリレコードと1つ以上のセカンダリレコードで構成されます。</p> <p>スパンドレコードの詳細については、「スパンドレコードのサポート」を参照してください。</p>
スレッド	Adabas はマルチスレッド処理を提供し、スループットを最大化します。I/O アクティビティにより、アクティブなスレッドでコマンドの処理が中断された場合は、自動的に別のスレッドに切り替えられます。ユーザーは、最大 250 個までの Adabas セッションで使用できるように、8 KB スレッドの数を設定できます。
スレッドタイム	<p>ニュークリアスがコマンドの処理に要した時間。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。時間は、1/4096 ミリ秒を計測単位として、バイナリ形式で示されます。</p> <p>Adabas スレッドタイムは、Adabas コールの ACBXCMDT (コマンドタイム) フィールドに出力されます。</p>
セカンダリ ISN	スパンドレコードのセカンダリレコードに割り当てられた ISN。
セカンダリレコード	<p>Adabas バージョン 8 (以降) では、スパンドレコードの概念が導入されています。スパンドレコード内のプライマリレコードより後の物理レコードは、セカンダリレコードと呼ばれ、論理レコードの残りのデータを含みます。つまり、スパンドレコードは、1つのプライマリレコードと1つ以上のセカンダリレコードから構成されます。</p> <p>スパンドレコードの詳細については、「スパンドレコードのサポート」を参照してください。</p>

- 世代
- 世代とはリカバリに使用される論理ユニットであり、この世代ごとの情報がリカバリログ (RLOG) に保存されます。世代には、次の一連のオペレーション間のすべてのアクティビティが含まれます。
- ADASAV SAVE/RESTORE (データベース)
  - RESTORE GCB
  - SAVE DELTA/RESTORE DELTA (データベース)
- 詳細については、「[世代：リカバリの単位](#)」を参照してください。
- セッション
- ユーザーセッションは、一連の Adabas コールから成り、OP コマンドで開始し、CL コマンドで終了させることができます。ユーザーとは、バッチモードプログラム、あるいは端末を使用する人を意味します。各ユーザーの一意性は、ユーザー ID、マシン、アドレススペース、および端末 ID により確保されます。
- Adabas セッションは、Adabas がアクティブになったときに開始され、Adabas が終了するまで続きます。この間に、Adabas ニュークリアスは、データベースに対して行われたすべての変更内容の正確な履歴として、一連のプロテクションエントリを作成します。この一連のプロテクションエントリは、WORK データセット (パート 1) に書き込まれ、さらにプロテクションログにもブロックごとに書き出されます。各ブロックにはニュークリアスのセッション番号、固有のブロック番号、およびタイムスタンプが含まれています。
- セット
- フラグビットに "1" が設定されている状態をセットと呼びます。

## た行

- ターゲット
- Adabas コールの受信側。ターゲットはコマンドキューを管理し、ADAMPM を使用してルーターと通信します。ターゲットは、[サービス](#)としても分類されます。Adabas ニュークリアスはターゲットの 1 つです。
- ディスクリプタ
- ディスクリプタは検索キーです。ユニークディスクリプタの値は、ファイルのレコードごとに異なります。ディスクリプタフィールドに応じて、アソシエータのインバーテッドリストにエントリが作成され、ディスクスペースが追加されたり、必要なオーバーヘッドが処理されたりします。
- 選択条件の中では、あらゆるフィールドを使用できます。広範囲に使用されるフィールドが検索条件としてディスクリプタ (キー) に定義されると、選択処理が大幅に速くなります。これは、Adabas

が、データストレージからレコードを読み込まずに、インバーテッドリストから直接、ディスクリプタの値にアクセスできるためです。

ディスクリプタフィールドは、検索コマンドでソートキーとして使用したり、論理的な順次読み込み処理（昇順または降順の値）を制御する方法として使用したり、ファイルカップリングの基準として使用したりすることができます。

フィールドの一部をサブディスクリプタとして定義することができます。また、フィールドやその一部を組み合わせてスーパーディスクリプタとして定義することもできます。ユーザー指定のアルゴリズムに基づいて、ハイパーディスクリプタや照合ディスクリプタを定義したり、発音によるエンコードアルゴリズムに基づいて、フォネティックディスクリプタを定義したりすることもできます。フォネティックディスクリプタは、言語固有の要件に合わせてカスタマイズすることが可能です。詳細については、「[概念および機能](#)」の「[Adabas 設計](#)」を参照してください。

#### データ圧縮

データ圧縮により、必要なストレージの量が大幅に削減されます。また、1回の物理的な転送で送信できる情報量が増加するため、I/O 効率が向上します。

Adabas では、データレコードを圧縮形式で保持します。圧縮には、数種類のオプションがサポートされています。詳細については、「[圧縮](#)」を参照してください。

#### データベース

Adabas では、データベースは、関連するファイルのグループを指します。

物理データベースは、Adabas ユーティリティを使って定義します。それぞれのデータベースは、データベース ID 番号 (DBID) で識別されます。物理データベースは、1セットのアソシエータとデータストレージのデータセットから成り、単一の DBID で識別されます。物理データベースにある物理ファイルには、アドレススペース内で実行している Adabas ニュークリアスを使用してアクセスできます。

#### データベース管理者 (DBA)

データベースのリソースを制御および管理します。担当する作業としては、データベース分散の定義、構造およびリソースの割り当て、プログラミングおよびオペレーション標準の作成および保守、高いパフォーマンスの確保、ユーザーに発生した問題の解決、ユーザー教育の定義および実施、データベースアクセスおよびセキュリティの制御、新しいデータベースリソースアプリケーションの拡張および統合計画、システムのアップグレードなどがあります。データベースアナリストと呼ばれることもあります。

- 同期トリガ
- 同期トリガでは、該当ユーザーの Adabas コマンド処理が中断されます。開始 Adabas コマンドは、トリガされたプロシージャの実行が完了するまで中断されます。プロシージャの結果が Adabas コマンドに影響する可能性があります。
- このタイプとは異なるトリガとして、[非同期トリガ](#)があります。
- トランザクション
- Adabas のデータ保護、リカバリ、およびユーザー再スタートは、論理トランザクションの概念に基づいています。論理トランザクションとは、データベース内の情報に論理的な不整合が生じないように、ユーザーの定義に従って、全体的に実行する必要がある処理の最小単位のことです。
- 論理トランザクションは1つ以上の Adabas コマンドで構成されます。これらのコマンドが実行されることによって、業務の論理ユニットを完了するために必要なデータベースの読み込みおよび更新が行われます。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最初のコマンドで始まり、ET (トランザクションの終了)、BT (トランザクションのバックアウト)、CL (クローズ)、または OP (オープン) コマンドが同じユーザーに発行されたときに終了します。
- 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 コマンドの実行が正常に完了すると、後続のユーザーまたは Adabas セッションの中断に関係なく、トランザクション中に行われた更新処理のすべては物理的にデータベースに適用されます。
- ET コマンドを正常に実行できない場合、そのトランザクションの実行中に発生した更新は、BT コマンドを手動で発行するか、または自動バックアウトルーチンの自動処理のいずれかによってバックアウトされます。詳細については、「[概念および機能](#)」の「[Adabas の使用](#)」を参照してください。
- トランスレータ
- ユーザーによる Adabas コールの論理 ID を、ターゲットの対応する物理 ID に変換するプロセス。
- トリガ
- 特定の条件に合致した場合に、Adabas が自動的に実行するプロシージャ。この条件は、DBMS に送信される Adabas コマンドごとに特定されます。条件の基準は、ターゲットファイル番号と、任意指定のコマンドタイプまたはフィールド(または両方)です。コマンドタイプとは、FIND、READ、STORE、UPDATE、および DELETE コマンドのことです。フィールドは、コマンドの対応するフォーマットバッファ内にある必要があります。

トリガの各タイプの詳細については、[プレコマンドトリガ](#)、[ポストコマンドトリガ](#)、[非同期トリガ](#)、[同期トリガ](#)、[関与トリガ](#)、および[非関与トリガ](#)を参照してください。

トリガ定義	実行対象のプロシージャの名前と属性、およびイベントのトリガを起動する選択条件（Adabas コマンドタイプ、ファイル名またはファイル番号、およびフィールド名）。
トリガテーブル	トリガ定義を格納する、Adabas ニュークリアス内のバッファ。
トリガメンテナンス機能	Adabas Online System (AOS) からアクセス可能な Natural アプリケーション。体系化されたメニューを使用して、トリガ定義の作成と管理、Adabas トリガのプロファイルの定義、ニュークリアスにおけるトリガのアクティビティの監視と一部の制御を行うことができます。

## な行

内部シーケンス番号 (ISN)	<p>Adabas レコードには、レコードが識別できるように、必ず内部シーケンス番号 (ISN) が割り当てられます。各レコードに割り当てられる ISN は、格納場所とは関係のない独自の ISN です。</p> <p>物理データベースファイルにあるレコードの ISN は、長さが 4 バイトで、範囲が MINISN から MAXISN までの値です。レプリケートされたファイルの場合は、どのファイルもレコードの ISN は同じになります。パーティション分割されたファイルの場合は、物理ファイルごとに ISN 範囲が異なります。</p>
ニュークリアス	<p>ニュークリアスはプログラムの集まりを指し、Adabas を操作したり、すべての処理を調整したり、ユーザープログラムのステートメントを Adabas コマンドに変換したりします。すべてのプログラムは、ニュークリアスを介して Adabas ファイルにアクセスします。データアクセスや更新などのデータベースのアクティビティはすべて、Adabas ニュークリアスにより管理されます。ほとんどの場合、1つのニュークリアスが1つの物理データベースを管理します。</p> <p>Adabas Parallel Services を使用する場合、単一オペレーティングシステム環境で1つのデータベースを対象に最大 31 個の Adabas ニュークリアスのクラスタを実行することができます。Adabas Cluster Services では、IBM 並列シスプレックス環境がサポートされているため、シスプレックスとして設定した複数オペレーティングシステム環境で最大 32 個の Adabas ニュークリアスのクラスタを実行することができます。この場合も、クラスタの実行対象のデータベースは 1 つです。</p>

 **Note:** 単一オペレーティングシステムイメージ (ADASMP) または複数 z/OS イメージ (Adaplex+) 環境で、単一の物理データベースを対象に複数ニュークリアスを実行する場合は、「[概念および機能](#)」の「[オプションの拡張](#)」を参照してください。

## 入力／出力バッファ

Adabas 入力／出力バッファエリアには、よく使用されるデータとデータ間の関係が含まれており、このエリアのサイズは Adabas セッションごとに変更することができます。このエリアでは、物理的入力／出力 (I/O) アクティビティが最小化されるため、コンピュータの処理時間が節約されます。起動時に入力／出力バッファと Adabas ニュークリアスがメインメモリにロードされます。

バッファには、データベースから読み込まれたブロックとデータベースに書き込まれるブロックが格納されます。

データベースから読み込まれるブロックのうち、最も頻繁にアクセスされるブロックは、バッファアルゴリズムによってメモリに保持されます。データベース内のブロックが必要になると、バッファの内容がチェックされ、そのブロックがすでにメモリにあるかどうか調べられます。これにより、不必要な読み込みが避けられます。

複数の更新が行われた場合、更新内容はブロックに蓄積された後、外部ストレージに書き込まれます (フラッシュ)。

## ノーマルインデックス (NI)

[インバーテッドリスト](#)の個別のディスクリプタに対するインデックス。ノーマルインデックスには、各ディスクリプタ値に対するエントリが含まれます。エントリには、値、値が含まれるレコードの数、およびそれらのレコードの ISN が含まれます。

# は行

## バイナリオカレンスカウンタ (BOC)

バイナリオカレンスカウンタ (BOC) には、オカレンスの数、またはファイルのマルチプルバリュー (MU) フィールドとピリオディックグループ (PE) フィールドの数が保存されます。

## バッファフラッシュ

バッファプール内のアソシエータおよびデータストレージのブロックは、最後のバッファフラッシュの実行以降に更新されると、書き込みフラグがオンに設定されます。バッファがフラッシュされると、これらのブロックはそれぞれ、アソシエータまたはデータストレージのデータセットに書き込まれます。ブロックはフラッシュ後にもそのままバッファプールに残りますが、書き込みフラグはオフになります。

バッファフラッシュには、同期型と非同期型があります。更新コマンドは、同期フラッシュの実行中には選択できませんが、非同期フラッシュの実行中には選択できます。

#### 非 DB ターゲット

Adabas ニュークリアスではないターゲット。Access および X-COM は非 DB ターゲットです。

#### 非関与トリガ

非関与トリガには、同期型と非同期型があります。非関与トリガのユーザー ID は、開始コマンドを識別する Adabas ユーザーキューエレメント (UQE) のユーザー ID とは異なります。したがって、このトリガは、開始コマンドのトランザクションロジックには関与しません。

このタイプとは異なるトリガとして、関与トリガがあります。

#### 非同期トリガ

非同期トリガは、Adabas コマンドの開始とは無関係に実行されます。ユーザーに対する Adabas コマンドの処理は、トリガされたプロシージャが別の処理として実行される間に、中断されずに続行されます。トリガされたプロシージャが実行されるのは、開始するコマンドからの応答を受信した後です。

このタイプとは異なるトリガとして、**同期トリガ**があります。

#### ピリオディック (PE) グループフィールド

ピリオディック (PE) グループフィールドは、FDT 内の連続フィールドを定義します。これらのフィールドはレコード内でまとめて繰り返されます。非ピリオディックグループフィールドのメンバ同様に PE メンバは、PE グループフィールドの直後にあり、PE フィールドより高いレベル番号を持ち、個別にも、グループとしてもアクセス可能です。それぞれの PE には、オカレンスの数を格納するバイナリオカレンスカウンタ (BOC) があります。

ピリオディックグループの繰り返し回数は、1 レコードあたり 191 にするか、または最大で 65,534 程度にすることができます。また、ピリオディックグループは 1 つ以上の **MU フィールド** を含むことができます。使用されないオカレンスまたは値には、ストレージスペースは不要です。実際の制限は、多くの要素により決定されます。MU および PE グループフィールドの実際の制限の詳細は、「**MU および PE オプションおよびフィールドタイプ**」を参照してください。



**Note:** 1 つのファイルに 191 個を超える MU または PE フィールドを使用する場合、ファイルがこれを使用できるように明示的に許可する必要があります (デフォルトでは許可されていません)。この場合には、ADADBS MUPEX 機能または ADACMP COMPRESS MUPEX および MUPECOUNT パラメータを使用します。

## ファイル

Adabas では、ファイルは同じ形式を持つ関連レコードのグループを指します。ただし、例外もあるので、「*概念および機能*」の「*Adabas 設計*」を参照してください。

単一の Adabas データベースに割り当てられたディスクストレージスペースは、*論理* Adabas ファイルにセグメント化されます。データベース内のスペース全体のうち、特定の部分が、各論理ファイルに割り当てられます。ファイルのレコードを格納すると、このスペースに空きがなくなる場合には、共通のフリースペースプールから追加のスペースが自動的に割り当てられます。このダイナミックなスペース割り当てと、解放されたスペースのダイナミックなりカバ리를併用すると、特に操作することなく Adabas データベースを長期間にわたって実行できます。

Adabas *物理* ファイルにはデータベースのレコードが格納されます。それぞれの物理ファイルは、ファイル番号で識別されます。物理データベースごとの物理ファイルの最大数（物理ファイル番号の最大数）は 5000 または ASSOR1 ブロックサイズから 1 を引いた値のうち、小さい方になります。

*拡張* ファイルは、1 つ以上の場所に存在する物理ファイルから構成される論理ファイルです。どの物理ファイルもフィールド定義テーブル (FDT) は同じですが、ISN の範囲はファイルごとに異なります。1 つ以上のフィールドのデータ内容（フィールド値の条件）によって、データレコードがどの物理ファイルに格納されるかが決まります。

マルチクライアントファイルは、オーナー ID を介してレコードにアクセス可能な Adabas ファイルです。関連ユーザーがアクセスまたは更新できるレコードは、ユーザーまたはグループのオーナー ID が一致するレコードに制限されます。オーナー ID を使用することによって、マルチクライアントファイルは 1 つの Adabas ファイルとして扱われますが、これを複数の論理ファイルとして扱うこともできます。この場合、同じオーナー ID に属するレコードのグループがそれぞれ論理ファイルとなります。スーパーオーナー ID の場合は、ファイルのすべてのレコードにアクセスできます。

## ファイルコントロールブロック (FCB)

Adabas によってデータベースの各ファイルに割り当てられるコントロールブロック。ファイルごとに 1 つずつ FCB が存在します。FCB は、ファイルの物理的な場所と、そのファイルに関連付けられている RABN を示します。FCB はアソシエータの一部です。

## フィールド

Adabas では、フィールドが情報の論理的な最小単位（現在の給与など）で、この単位はユーザーにより定義され、参照されます。

Adabas では、次の 4 種類のフィールドがサポートされています。

	レコードごとに単一の値	レコードごとに複数の値
単一のフィールド	エレメンタリ	MU
マルチプルフィールド	グループ	PE

基本的なフィールドタイプは、エレメンタリフィールドとマルチプルバリュースフィールドの2つです。エレメンタリフィールドの値は、レコードごとに1つだけです。マルチプルバリュース (MU) フィールドに格納できる値の最大数は、1レコードあたり 65,534 個 (オカレンス) 程度です。それぞれのマルチプルバリュースフィールドには、オカレンスの数を格納するバイナリオカレンスカウンタ (BOC) があります。

FDT 内の連続したフィールドに頻繁にアクセスする場合、グループフィールドを定義して、これらのフィールドを参照することができます。フィールドのレベルと Adabas ショートネーム以外に、グループフィールドに定義された属性はありません。このフィールドは、FDT 内のメンバフィールドの直前にあります。メンバフィールドをグループフィールドに割り当てる際には、より高いフィールドレベル番号を使用します。Adabas は7個までのフィールドレベルをサポートしています。

ピリオディック (PE) グループフィールドは、FDT の連続したフィールドを定義します。これらのフィールド全体が、レコード内でまとめて繰り返されます。通常のグループフィールドのメンバと同様に PE メンバは、PE グループフィールドの直後に続き、PE フィールドより高いレベル番号が割り当てられ、個別にも、グループとしてもアクセス可能です。各 PE には、オカレンスの数を格納する BOC があります。ピリオディックグループは、1レコードあたり 65,534 回程度まで繰り返すことが可能であり、1つ以上のマルチプルバリュースフィールドで構成されています。

MU フィールドおよび PE グループのオカレンス数の上限は、実際にはデータストレージレコードの最大長 (ADALOD MAXRECL パラメータ) によって決まり、デフォルトではデータストレージブロックのサイズから 4 を引いた数です。

#### フィールド定義テーブル (FDT)

ファイルのレコード構造と内容を定義するテーブル。データベースファイルごとに1つの FDT があります。FDT はアソシエータの固定エリアに格納され、3つの部分から構成されます。1つ目の部分は、レコードの物理順に沿ったフィールドのリストで、2つ目の部分は、1つ目の部分のレコードに対する簡易インデックスです。3つ目の部分は、ファイルのサブフィールド、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、およびフォネティックディスクリプタを定義します。

フィールドの一部（サブフィールド）またはフィールドの組み合わせ（スーパーフィールド）をエレメンタリフィールドとして定義することができます。サブフィールドとスーパーフィールドは、読み込みのみ使用できます。これらのフィールドが変更されるのは、元のフィールドが更新された場合のみです。

#### フォーマット ID

フォーマット ID は、デコード結果のフォーマットバッファに割り当てられる ID です。フォーマットバッファは、ユーザーごとに 1 回保存されます。保存後には、そのフォーマットバッファをフォーマット ID で参照したり、再使用したりすることができます。個別のフォーマット ID は、ユーザーごとに 1 回保存され、そのユーザーのみが使用可能です。フォーマットバッファは、固有のフォーマット ID と端末 ID の組み合わせによって識別されます。

これとは違うタイプとして、[グローバルフォーマット ID](#) があります。グローバルフォーマット ID は、デコード結果のフォーマットバッファをグローバルに 1 回保存し、同じ Adabas ニュークリアスで実行中の複数のユーザーとそのフォーマットバッファを共有します。

フォーマット ID は、Adabas コントロールブロックフィールド（ACBADD5 または ACBXADD5）に指定します。フォーマット ID の詳細は、「[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。

#### 物理 ID

ターゲットの ID。この識別子は "0" より大きく、"65,536" 未満である必要があります。データベース ID（DBID）は物理 ID です。

#### プライマリ ISN

[スパンドレコードのプライマリレコード](#)に割り当てられた ISN。

#### プライマリレコード

Adabas バージョン 8（以降）では、[スパンドレコード](#)の概念が導入されています。スパンドレコードの最初の物理レコードはプライマリレコードと呼ばれ、圧縮レコードの先頭部分を含みます。スパンドレコードは、1つのプライマリレコードと1つ以上の[セカンダリレコード](#)で構成されます。

スパンドレコードの詳細については、「[スパンドレコードのサポート](#)」を参照してください。

#### プレコマンドトリガ

プレコマンド（または"プレ"）トリガは、Adabas ニュークリアスによる開始 Adabas コマンドの処理前に実行されます。Adabas コマンドの実行前にチェックが行われ、トリガを起動すべきかどうか確認されます。

このタイプとは異なるトリガとして、[ポストコマンドトリガ](#)があります。

プレトリガキュー	プレコマンドトリガの起動により、実行待ちになっているプロシージャを格納します。詳細については、 <a href="#">プレコマンドトリガ</a> を参照してください。
プロシージャ	プロシージャは、Naturalの標準機能を使用して記述され、テストされた Natural サブプログラムです。トリガであるか、ストアドプロシージャであるかにかかわらず、同じタイプのパラメータがサブプログラムに渡されます。
ポストコマンドトリガ	<p>ポストコマンド（または"ポスト"）トリガは、Adabas ニュークリアスによる開始 Adabas コマンドの処理後に実行されます。トリガされたプロシージャは、Adabas ニュークリアスからのコマンドのリターンコードがゼロの場合にのみ実行されます。リターンコードがゼロ以外の場合は、トリガがあるかどうかはチェックされず、処理は通常どおりに続きます。正常に実行されたコマンドについては、コマンドの処理の完了前に、トリガがあるかどうかチェックされます。</p> <p>このタイプとは異なるトリガとして、<a href="#">プレコマンドトリガ</a>があります。</p>
ポストトリガキュー	ポストコマンドトリガの起動により、実行待ちになっているプロシージャを格納します。詳細については、 <a href="#">ポストコマンドトリガ</a> を参照してください。

## ま行

マルチプルバリュー (MU) フィールド	<p>マルチプルバリュー (MU) フィールドには、複数の値を格納することができます。ファイル内の各 MU フィールドのオカレンスは、1レコードあたり 191 個 (オカレンス) にするか、または最大で 65,534 個 (オカレンス) 程度にすることができます。実際のリミットは、多くの要素により決定されます。MU および PE グループフィールドの実際の制限の詳細は、「<a href="#">MU および PE オプション および フィールドタイプ</a>」を参照してください。それぞれの MU フィールドには、オカレンスの数を格納するバイナリオカレンスカウンタ (BOC) があります。</p> <p> <b>Note:</b> 1つのレコードに 191 個を超える MU または PE グループを使用する場合、ファイルがこれを使用できるように明示的に許可する必要があります (デフォルトでは許可されていません)。この場合には、ADADBS MUPEX 機能または ADACMP COMPRESS MUPEX および MUPECOUNT パラメータを使用します。</p>
----------------------	--

## や行

ユーザー	Adabas コールを生成し、ADALINK を使用してコミュニケーションを行うバッチまたはオンラインのアプリケーションプログラム。
ユーザーセッション	ユーザーセッションとは、バッチモードプログラム、あるいは端末を使用する人を意味します。ユーザーセッションは、 <a href="#">Adabas セッション</a> 中にのみ存在できます。つまり、Adabas ニュークリアスがアクティブなときのみです。これは、一連の Adabas コールから成り、オープンユーザーセッション (OP) コマンド (任意) で開始し、クローズユーザーセッション (CL) コマンドで終了させることもできます。
ユーザープログラム	Adabas コールを生成し、ADALINK を使用してコミュニケーションを行うバッチまたはオンラインのアプリケーションプログラム。
ユーティリティセッション	ユーティリティセッションは、バッチで実行されるか、または Adabas Online System を使用してオンラインで実行されます。ユーティリティには、Adabas ニュークリアスがアクティブであることが必要なものとそうでないものがあります。ユーティリティの実行には、ADARUN 起動パラメータも使用されます。

## ら行

ラージオブジェクトフィールド (LB フィールド)	ラージオブジェクトフィールド (LB フィールド) は、理論的な最大サイズが 2GB の英数字フィールドフィールドです。このようなフィールドは、ドキュメント (HTML、XML、Microsoft Word または PDF ドキュメントなど)、画像 (JPG や BMP ファイルなど)、またはその他の複合データなどを、データベース内の単一フィールドに保存するために使用されます。  詳細については、「ラージオブジェクトオプション LB」、および「ラージオブジェクト (LB) フィールドの基本」を参照してください。
リージョン	z/OS、VSE、および BS2000 (VM 以外) オペレーティングシステムによってユーザージョブに割り当てられるストレージスペースの総称です。
リセット	フラグビットに "0" が設定されている状態をリセットと呼びます。
ルーター	1つのオペレーティングシステムの境界内で通信を行うための中心的なルーチン。このルーチンは、ユーザーによる ADALINK ルーチン、またはターゲットによる ADAMPM の発行とともに呼び出されます。ルーターの主な目的は、ADALINK と Adabas の間の情報転送です。また、ID テーブルの管理にも使用されます。z/VM

および BS2000 環境では、ADALINK 機能またはその他の Adabas 機能の間でルーター機能が分割されます。z/OS の Adabas SVC と VSE は、ルーターの 1 つです。

レコード	Adabas では、レコードは、完結した情報のまとまりを構成する関連フィールドの集合を指します。例えば、従業員 1 人の全給与データがこれに該当します。
レコードバッファ	呼び出し側プログラムのパラメータエリア（ユーザーバッファ）の一部。Adabas で読み込み、検索、および更新処理を実行するときに転送対象のデータが格納されます。Adabas は、データのフィールド定義を読み込むときに、レコードバッファにフィールド定義情報も返します。
論理 ID	メッセージの送信先ターゲットのユーザーの ID。"0" より大きく、"256" 未満である必要があります（明示的または暗黙的に、ACBFNR の第 1 バイトの内容が論理 ID です）。
論理的重複	他のレコードの内容によって示される情報を継続的に保存する処理のこと。例えば、クレジットコントロールルーチンで顧客に対する請求額をすべて合計する場合、情報へのアクセスの速さは、顧客レコード内で継続的に合計を処理する方が、要求ごとに関連する請求額を読み込んでから合計するよりも速くなります。後者の処理では、大量のレコードにランダムにアクセスする必要があるからです。
論理トランザクション	<p>データベース内の情報に論理的な不整合が生じないように、ユーザーの定義に従って、全体的に実行する必要がある処理の最小単位。</p> <p>論理トランザクションは 1 つ以上の Adabas コマンドで構成されます。これらのコマンドが実行されることによって、業務の論理ユニットを完了するために必要なデータベースの読み込みおよび更新が行われます。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最初のコマンドで始まり、ET（トランザクションの終了）、BT（トランザクションのバックアウト）、CL（クローズ）、または OP（オープン）コマンドが同じユーザーに発行されたときに終了します。</p>

## わ行

ワークロードバランシング	Adabas Parallel Services は、各ニュークリアスが処理するユーザーの数およびコマンド数に応じて、クラスタニュークリアス全体の負荷を均等に分散します。
--------------	---



# 索引

---

## A

- Adabas
  - 定義, 3
- Adabas セッション
  - 定義, 10
- Adabas 相対ブロック番号 (RABN)
  - 定義, 2
- ADALINK
  - 定義, 3
- ADARUN
  - 定義, 3

## D

- DBA (参照 データベース管理者 (DBA) )

## F

- FDT (参照 フィールド定義テーブル (FDT) )

## I

- ISN (参照 内部シーケンス番号 (ISN) )

## R

- RABN (参照 Adabas 相対ブロック番号 (RABN) )

## S

- STPRBE ルーチン
  - 定義, 4
- STPUTRAK ルーチン
  - 定義, 5

## あ

- アドレスコンバータ
  - 定義, 5
- アドレススペース
  - 定義, 6

## お

- オペレータコマンド, 6

## か

- 拡張ファイル
  - 定義, 16

## く

- グループ
  - フィールドタイプ, 17
- グローバルトランザクション
  - 定義, 7

## こ

- コマンド
  - オペレータ, 6

## さ

- サブフィールド, 18
- サービス
  - 定義, 8

## す

- ストアドプロシージャ
  - 定義, 9
- スレッド
  - 定義, 9
- スレッドタイム
  - 定義, 9
- スーパーフィールド, 18

## せ

- セッション
  - Adabas, 10
  - ユーザー, 10

## た

- ターゲット
  - 定義, 10

## て

- ディスクリプタ
  - インバーテッドリストの並び順を決定する値, 6

定義, 10  
データ圧縮  
定義, 11  
データベース  
定義, 11  
定義、物理, 11  
物理データベースの定義, 11  
データベース管理者 (DBA)  
定義, 11

## と

トランザクション  
定義, 12  
トランスレータ  
定義, 12  
トリガ  
関与タイプ, 7  
定義, 12  
同期タイプ, 12  
非関与タイプ, 15  
非同期タイプ, 15  
プレコマンドタイプ, 18  
ポストコマンドタイプ, 19  
トリガテーブル, 13  
トリガメンテナンス機能, 13

## な

内部シーケンス番号 (ISN)  
定義, 13

## に

入力/出力バッファ  
アルゴリズム, 14  
定義, 14  
ニュークリアス  
定義, 13  
ニュークリアスクラスタ, 13  
ニュークリアスのクラスタ, 13

## は

バイナリオカレンスカウンタ  
定義, 17  
バッファフラッシュ  
定義, 14  
入力/出力バッファから, 14

## ひ

ピリオディックグループ  
定義, 18

## ふ

ファイル  
定義, 16  
物理, 16  
フィールド  
定義, 16  
レベル, 17

フィールドタイプ  
エレメンタリ, 17  
グループ, 17  
ピリオディックグループ, 18  
マルチプルバリュウ, 17  
フィールド定義テーブル (FDT)  
定義, 17  
プロシージャ  
定義, 19

## ま

マルチクライアントファイル  
定義, 16  
マルチプルバリュウフィールド  
定義, 17

## ゆ

ユーザーセッション  
定義, 10  
ユーザープログラム  
定義, 20

## り

リージョン  
定義, 20

## る

ルーター  
定義, 20

## れ

レコード  
定義, 21  
レコードバッファ  
定義, 21