



IBM Systems - iSeries

ファイルおよびファイル・システム
データベース・ファイル管理

バージョン 5 リリース 4





IBM Systems - iSeries

**ファイルおよびファイル・システム
データベース・ファイル管理**

バージョン 5 リリース 4

ご注意！

本書および本書で紹介する製品をご使用になる前に、241ページの『特記事項』に記載されている情報をお読みください。

本書は、IBM i5/OS (製品番号 5722-SS1) のバージョン 5、リリース 4、モディフィケーション 0 に適用されます。また、改訂版で断りがない限り、それ以降のすべてのリリースおよびモディフィケーションに適用されます。このバージョンは、すべての RISC モデルで稼働するとは限りません。また CISC モデルでは稼働しません。

本マニュアルに関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。

<http://www.ibm.com/jp/manuals/main/mail.html>

なお、日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは

<http://www.ibm.com/jp/manuals/> の「ご注文について」をご覧ください。

(URL は、変更になる場合があります)

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原 典： IBM Systems - iSeries
Files and file systems
Database file management
Version 5 Release 4

発 行： 日本アイ・ピー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

第1刷 2006.2

この文書では、平成明朝体™W3、平成明朝体™W7、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、および平成角ゴシック体™W7を使用しています。この(書体*)は、(財)日本規格協会と使用契約を締結し使用しているものです。フォントとして無断複製することは禁止されています。

注* 平成明朝体™W3、平成明朝体™W7、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、平成角ゴシック体™W7

© Copyright International Business Machines Corporation 1998, 2006. All rights reserved.

© Copyright IBM Japan 2006

目次

データベース・ファイル管理	1
V5R4 の新機能	1
印刷可能な PDF	2
データベース・ファイル管理の概要	2
ファイル・タイプ	3
データベース・ファイルの処理	4
ファイル・リソースの割り振り	4
ファイルのコピー	6
ファイルのオープン	86
ファイルの共用	94
オーバーライドの使用	97
パフォーマンス	133
キー順アクセス・パスの回避	134
指定するパラメーターを少なくする	134
レコード・フォーマット・レベル ID の検査	134
ファイルのコピー時におけるエラーの防止	135
セキュリティ	142
オブジェクト権限	142
データ権限	144
ファイル操作に必要な権限	144
ファイル作成時のファイルおよびデータへのアクセス制限	145
データベース・ファイル管理のトラブルシューティング	146
ファイル・エラー検出およびサーバーによる処理	146
サーバーによるファイル内のメッセージおよびメッセージ・モニター	147
サーバーによるファイル内のメジャーおよびマイナー戻りコード	149
ファイル・サーバー・エラーからの回復	150
リファレンス	153
2 バイト文字セットのサポート	153
フィードバック域のレイアウト	194
データベース・ファイル管理の関連情報	237
付録. 特記事項	241
プログラミング・インターフェース情報	242
商標	242
使用条件	243

データベース・ファイル管理

従来型ファイル管理は、オペレーティング・システムの一部であり、IBM® iSeries™ サーバー 上にある従来のファイル・オブジェクト (QSYS.LIB ライブラリー内の *FILE オブジェクト) の格納とアクセスを制御します。一部の旧リリースでは、この機能のことをデータ管理と呼んでいました。データは、内部記憶域 (データベース・オブジェクトなど)、外部記憶媒体 (ディスケット、テープ、プリンターの各オブジェクト)、または別のシステムに格納されます。

注: コード例を使用することにより、お客様は 239 ページの『コードに関する特記事項』の条件に同意することになります。

V5R4 の新機能

V5R4 では、インポート・ファイルからのコピー (CPRFRMIMPF) コマンドとインポート・ファイルへのコピー (CPYTOIMPF) コマンドの機能が強化されました。

CPYTOIMPF コマンドの機能強化

- | ブランク文字の削除 (RMVBLANK) パラメーターが CPYTOIMPF コマンドに追加されました。以前のリリースでは、CPYFRMIMPF コマンドには RMVBLANK パラメーターがありましたが、CPYTOIMPF コマンドにはこのパラメーターがありませんでした。
- | RMVBLANK パラメーターを使用すれば、CPYTOIMPF コマンドの使用時に文字フィールドおよび数字フィールドから先行ブランク文字と末尾ブランク文字を削除することができます。
- | CPYTOIMPF コマンドのフィールド区切り文字 (FLDDL) パラメーターで、水平タブ文字用の特殊値 *TAB が直接サポートされるようになりました。



CPYTOIMPF コマンドと CPYFRMIMPF コマンドの機能強化

- | コピー元ファイルの文字フィールドにブランク文字ストリングが含まれる場合、RMVBLANK パラメーターに *NONE が指定されない限り、CPYTOIMPF コマンドと CPYFRMIMPF コマンドでは対応するコピー先のフィールドに単一ブランクを保持します。
- | ストリング区切り文字を通常の文字として扱うよう指示するために、ストリング・エスケープ文字 (STRESCCHR) パラメーターが追加されました。この新しいパラメーターを使用すれば、ストリング区切り文字自体がストリング・フィールド内の文字である場合に、インポート・ユーティリティは区切り文字を正しく解析することができます。
- | ストリング区切り文字として二重引用符文字が使用されることを示すために、ストリング区切り文字 (STRDLM) パラメーターのデフォルト値として特殊値 *DBLQUOTE が提供されています。
- | コピー元 CCSID (FROMCCSID) パラメーターが CPYTOIMPF コマンドに追加されました。このパラメーターを使用すれば、コピー元ファイル・フィールドが 65535 の場合に使用すべき CCSID を指定することができます。

1 コピー先 CCSID (TOCCSID) パラメーターが CPYFRMIMPF コマンドに追加されました。このパラメータ
1 ーを使用すれば、コピー先ファイル・フィールドの CCSID が 65535 の場合に使用すべき CCSID を指定
1 できます。

新機能および変更された機能を見分ける方法

技術上の変更点を見やすくするために、この情報では、以下のアイコンが使用されています。

-  アイコン。このアイコンの場所から、新規情報や改訂情報が始まっています。
-  アイコン。このアイコンの場所で、新規情報や改訂情報が終わっています。

このリリースの新規情報や改訂情報に関するその他の情報については、「iSeries プログラム資料説明書」を参照してください。

印刷可能な PDF

これは、この情報の PDF を表示したり印刷したりする場合に使用します。


本書の PDF 版を表示またはダウンロードするには、データベース・ファイル管理を選択します。

PDF ファイルの保存

表示または印刷のために PDF をワークステーションに保存するには、以下のようになります。

1. ブラウザーで PDF を右マウス・ボタン・クリックします (リンク上で右マウス・ボタン・クリック)。
2. PDF をローカルに保存するオプションをクリックします。
3. PDF を保存したいディレクトリーに進みます。
4. 「保存」をクリックします。

Adobe Reader のダウンロード

これらの PDF を表示または印刷するには、システムに Adobe Reader がインストールされている必要があります。このアプリケーションは、Adobe Web サイト (www.adobe.com/products/acrobat/readstep.html)  から無料でダウンロードできます。

データベース・ファイル管理の概要

ファイル管理は、アプリケーションがサーバー上のデータを作成およびアクセスするときに使用する機能を提供し、アプリケーションの定義に従ってデータの保全性を確保します。

これまでデータ管理と呼ばれていた従来型ファイル管理は、オペレーティング・システムの一部であり、アプリケーション・プログラムによるデータの格納とアクセスを制御します。データは、内部記憶域 (データベースなど)、外部記憶媒体 (ディスケット、テープ、プリンター)、または別のシステムに格納されます。

ファイル管理では、ユーザーが CL コマンドを使用してファイルを管理 (作成、変更、オーバーライド、または削除) し、一連の操作 (たとえば、読み取り、書き出し、オープン、またはクローズ) を介してデータを作成およびアクセスすることができるようにする機能を提供しています。さらにファイル管理には、外部装置にアクセスし、データの作成とデータへのアクセスにおける、外部装置の属性の使用を制御する機能があります。

プリンターおよびディスク装置をより効率的に使用したい場合は、ファイル管理に、入力または出力に備えてデータをスプールする機能があります。たとえば、プリンターに書き出されるデータは、プリンターが使用できるようになるまで出力待ち行列に保留しておくことができます。

IBM iSeries サーバー・システムでは、各ファイル (ファイル・オブジェクトとも呼ばれる) に、ファイルの特性や、ファイルに関連するデータをレコードに編成する方法を説明した記述があり、また多くの場合には、レコード中のフィールドについて説明した記述が含まれています。オペレーティング・システム (Operating System/400[®] または OS/400[®] プログラム) は、常にこの記述を使用してファイル进行处理します。

これらのファイル・オブジェクトを使用すると、システム上におけるデータの作成およびアクセスを行うことができます。ファイル管理では、複数の異なるタイプのファイルを定義および制御します。各ファイル・タイプには、ファイルを作成および変更するための CL コマンドが関連付けられており、ファイル管理が提供する操作でも、データの作成およびアクセスを行うことができます。

ファイル・タイプ

ファイル管理では、下記のタイプのファイルをサポートします。

- **データベース・ファイル**は、データベース関連のデータをシステムに永久的に格納するためのファイルです (分散ファイルも含まれます)。
- **装置ファイル**は、ディスプレイ、プリンター、テープ、ディスク、および通信回線によって接続されている他のシステムなどの、外部接続装置に対するアクセスを提供するファイルです。次のような装置ファイルがサポートされています。
 - **ディスプレイ・ファイル**。これはディスプレイ装置にアクセスするためのファイルです。
 - **プリンター・ファイル**。これは印刷出力の様式を説明するファイルです。
 - **テープ・ファイル**。これはテープ装置上のデータ・ファイルにアクセスするためのファイルです。
 - **ディスク・ファイル**。これはディスク装置上のデータ・ファイルにアクセスするためのファイルです。
 - **システム間通信機能 (OS/400-ICF) ファイル** (以後 ICF ファイルと呼びます)。これはあるサーバー上のプログラムが、同一サーバー上または別のサーバー上のプログラムと通信するためのファイルです。
- **保管ファイル**は、保管データを (ディスクまたはテープを必要としないで) ディスク上に保管するためのファイルです。
- **分散データ管理 (DDM) ファイル**は、リモート・サーバー上に保管されているデータ・ファイルにアクセスするためのファイルです。

各ファイル・タイプには、それぞれのタイプに固有の一連の特性があり、これによってファイルの使用方法およびファイルが提供できる機能が決まります。ただし、ファイルの概念は、ファイル・タイプに関係なくすべて同じです。ファイルをプログラムで使用するときは、名前をファイルで参照し、ファイル記述と、ファイル・タイプによってはデータ自体の両方を識別します。本書は、ユーザーがすべてのファイル・タイプに共通する特性を理解し、その結果、ファイルの機能を最大限に活用できるようになることを意図しています。

関連概念

6 ページの『ファイルのコピー』

iSeries のコピー機能では、物理ファイルと論理ファイルのコピー、メンバーとレコードのコピー、複合オブジェクトのコピー、およびサーバー間のファイルのコピーが可能です。

86 ページの『ファイルのオープン』

アプリケーション・プログラムでファイルを使用したい場合は、名前でそのファイルを参照します。そこで、そのファイルに関するファイル記述が、プログラムとサーバーとの対話の方法を制御します。

145 ページの『ファイル作成時のファイルおよびデータへのアクセス制限』

権限を指定すると、ファイルへのアクセスを制御できるようになります。ファイル作成時に共通権限を指定するには、作成コマンドで AUT パラメーターを使用してください。

94 ページの『ファイルの共用』

iSeries のファイル管理では、いくつかのレベルに分かれた共用ファイルのサポートが提供されています。多数のユーザー、多数のジョブ、または同じジョブ内の多数のプログラムの中で、ファイルを共有することができます。

102 ページの『オーバーライドの適用』

実行できるオーバーライドには、ファイル・オーバーライドとプログラム装置項目オーバーライドという 2 つの一般的なタイプがあります。

データベース・ファイルの処理

このトピックでは、ファイルを効率的に使用するために実行できるさまざまなタスクについて説明します。

- サーバーが各ファイル・タイプに割り振るリソース。
- 装置ファイルやデータベース・ファイルなどのさまざまなファイル間のデータ移動の方法。
- アプリケーションを使用してファイルにアクセスする場合に利用できるオプション。
- 共用ファイルに利用できるサポート・レベル。
- ファイル名、装置名、リモート・ロケーション名などのファイル属性を変更するプロセス。

ファイル・リソースの割り振り

リソースとは、ジョブやタスクに必要なサーバー部分で、主記憶域、装置、処理装置、プログラム、ファイル、ライブラリー、およびフォルダーなどが含まれます。高水準言語プログラムを作成するときは、各ファイル・タイプに対してサーバーが割り振ったリソースを認識していなければなりません。

通常、サーバーは、リソースの割り振りが必要な操作が要求されると、そのつど割り振りを実行します。たとえば、プログラムで使用される各ファイルに関するリソースは、ファイルがオープンされるときに割り振られます。

プログラムに必要なすべてのリソースを使用できるようにしてからプログラムを実行する場合は、プログラムを実行する前に、ジョブでオブジェクト割り振り(ALCOBJ) の CL コマンドを使用することができます。特に ALCOBJ コマンドでは、データベース・ファイルおよびほとんどの装置を割り振ることができます。

リソースの割り振りが必要な操作には、次のようなものがあります。

- オープン
- 獲得
- リモート・サーバーでのプログラムの開始

ファイル・リソース割り振りの概要

高水準言語プログラムでファイルを使用する時は、操作の実行に必要なリソースのサーバーでの割り振りが必要な操作がいくつかあります。一般的に、サーバーがこのリソースの割り振りを行うのは、複数のユーザーが競合することなくファイルを使用するためです。

たとえば、あるファイルを使用しているアプリケーション・プログラムがある間は、サーバーはそのファイルを削除することを許可しません。サーバーは、ファイルがオープンされるときにロックを取得することによってこれを行います。ファイル削除操作もそのファイルに対してロックしようとして試みますが、そのファイルを使用しているプログラムが、ファイルのオープン以後まだファイルをロックしていて、ロックの競合が生じるためその試みは成功しません。

割り振りが必要なファイル・リソース

サーバーが割り振らなければならないファイル・リソースは、ファイルのタイプおよび実行される操作によって異なります。

ファイル・リソースには、次の種類があります。

• オープン

- スプールされる (SPOOL(*YES)) プリンター・ファイルおよびディスク・ファイルの場合、ファイル・リソースにはファイル記述、指定出力待ち行列、およびサーバーの中のスプール・データ用記憶域が含まれます。データはスプールされるので、装置が使用可能である必要はありません。
- データベース・ファイルの場合、ファイル・リソースはファイル全体から成り、ファイル、メンバー、データ、および関連付けられたアクセス・パスが含まれます。
- スプールされない (SPOOL(*NO)) プリンター・ファイルおよびディスク・ファイルは、テープ・ファイル、ディスプレイ・ファイル、および一部の ICF ファイルと同様、ファイル・リソースにファイル記述および装置を含みます。APPC、APPN、またはシステム間通信を使用する ICF ファイルの場合、ファイル・リソースには、ファイル記述および装置に関連付けられたセッション・リソースが含まれます。
- 保管ファイルの場合は、ファイル・リソースはファイル全体からなり、ファイルおよびデータを含みます。
- DDM ファイルの場合、ファイル・リソースには、ファイル記述および装置に関連付けられたセッション・リソースが含まれます。

• 獲得操作

ディスプレイ・ファイル、および APPC/APPN またはシステム間通信を使用していない ICF ファイルの場合、サーバーは装置をリソースとして割り振ります。APPC/APPN またはシステム間通信を使用する ICF ファイルの場合、リソースには、装置に関連するセッション・リソースが含まれます。

• リモート・サーバーでのプログラムの開始

ファイル・リソースには、APPC および APPN に必要なセッション・リソースが含まれます。

サーバーがリソースを割り振る方法

リソースの割り振りにあたって、リソースが即時使用可能でない場合、サーバーは、事前に定義されている時間だけ待機します。定義された時間内にリソースが使用可能にならなければ、サーバーはエラーを生成します。

ALCOBJ コマンドを使用している場合、コマンドは失敗します。プログラムでファイル操作を実行している場合、操作は正常に実行されず、サーバーはエラー・メッセージをプログラム・メッセージ待ち行列に送ります。このような場合には、ご使用の高水準言語のエラー処理機能を使用して、操作を再試行することができます。たとえば、別のジョブでファイルに関連する装置を使用中であるためにオープン操作が正常に行われない場合は、別のジョブでの装置の使用が完了して、ユーザーのプログラムでその装置が使用できるようになるまで、オープン操作を指定回数だけ再試行することができます。

リソースの割り振り時にサーバーが待機する時間の長さは、ALCOBJ コマンドおよび、ファイルの作成に使用する CL コマンドの WAITFILE パラメーターで指定します。プログラムの実行前に ALCOBJ コマンドを使用する場合は、リソースは使用可能になっているため WAITFILE パラメーターの値は関係ありません。

アプリケーション・プログラムに、装置ファイルで生じている装置エラーを処理するエラー処理プロシージャがある場合は、*IMMED 以外の値を指定して、サーバーがエラーから回復できるようにしなければなりません。エラー回復のために、ユーザー・プログラムにより要求されるオープン操作または獲得操作でのリソースの割り振りは、その装置にするサーバー回復手順が完了しない限り、正常には行われません。

以下のリストは、WAITFILE パラメーターに使用できる値を示しています。

***IMMED**

この値は、待ち時間を許可しないことを示します。ファイル・リソースの即時割り振りが必要です。

***CLS** ジョブのデフォルト待機時間が、ファイル・リソースの割り振り待機時間として使用されます。

number-of-seconds

ファイル・リソースの割り振りのためにプログラムを待機させる最大待機時間を秒数で指定します。有効な値は 1 から 32767 (32 767 秒) の範囲です。

ファイルのコピー

iSeries のコピー機能では、物理ファイルと論理ファイルのコピー、メンバーとレコードのコピー、複合オブジェクトのコピー、およびサーバー間のファイルのコピーが可能です。

コピー機能を使用すると、装置ファイル相互間、データベース・ファイル相互間、装置ファイルとデータベース・ファイル相互間で、iSeries フィールド・レベルでデータの移動ができます。このコピー機能を使用して、任意のフィールドを配列し直し、拡張し、または排除することができます。データベース・ファイルを定義することもできます。

関連概念

3 ページの『ファイル・タイプ』

ファイルのコピー: 概要

コピー機能を使用すると、装置ファイル相互間、データベース・ファイル相互間 (分散 DB2[®] マルチシステム・ファイルを含む)、または装置ファイルとデータベース・ファイル相互間で、データを移動することができます。

従来のコピー・ユーティリティとは異なり、iSeries のコピー機能は、フィールド・レベルでの処理機能を備えています。したがって、このコピー機能を使用すると、フィールドを配列し直し、拡張し、または排除することができます。サーバーは、データベース・ファイルを定義する方法も提供します。特定のコピー・コマンドにより、テープ装置、ディスク装置、データベース・ソース・ファイル、およびオープン照会ファイルの処理を簡素化することができます。

各コピー・コマンドでサポートされている特定のパラメーターについては、CL に関するトピックを参照してください。

関連概念

20 ページの『物理ファイルまたは論理ファイルのコピー』

iSeries 上の物理ファイルまたは論理ファイル (コピー元ファイル) を、まだ存在していない別の物理ファイル (コピー先ファイル) にコピーするには、CPYF コマンドを使用することができます。

ファイルのコピー: コマンド:

このトピックで説明するコマンドを使用して、ファイルにレコードをコピーしたり、ファイルからレコードをコピーすることができます。

ファイルにレコードをコピーしたり、ファイルからレコードをコピーしたりするには、以下のコマンドを使用します。

CPYF ファイル・コピー: データベースまたは外部装置からデータベースまたは外部装置に、ファイルのすべてまたはファイルの一部をコピーします。

CPYFRMTAP

テープからのコピー: テープ・ファイルからデータベース・ファイルまたは装置ファイルへコピーします。コピー元ファイルはテープ・ファイルでなければなりません。コピー先ファイルは物理ファイル、ディスク・ファイル、テープ・ファイル、またはプログラム記述のプリンター・ファイルにすることができます。 **QSYSPRT** を使用すれば、フォーマットされたレコードのリストを入手することができます。

CPYTOTAP

テープへのコピー: データベース・ファイルまたは装置ファイルからテープ・ファイルへコピーします。コピー先ファイルはテープ・ファイルでなければなりません。コピー元ファイルは物理ファイル、論理ファイル、ディスク・ファイル、テープ・ファイル、またはインライン・データ・ファイルにすることができます。

CPYSRCF

ソース・ファイル・コピー: データベース・ソース・ファイルをソース物理ファイルへコピーし、コピー元ファイルからコピー先ファイルの **CCSID** にデータを変換します。 **QSYSPRT** を使用して定様式リストを作成できます (このファイルはソース・レコードに対応して変更されるので、他のコピー・コマンドのファイル様式とは異なります)。コピー元ファイルからコピー先ファイルへレコード・データがコピーされ、レコード様式の違いは無視されます (**CCSID** 以外は、 **CPYF** コマンドの **FMTOPT(*NOCHK)** パラメーター・オプションの場合と同様です)。

CPYFRMQRYF

Query ファイルからのコピー: オープン **Query** ファイルをデータベース・ファイルまたは装置ファイルにコピーします。

コピー・コマンドがエラーを生成する場合、サーバーは、ジョブに対する **DDM** 会話を再利用しません。

CPYF または **CRYSRCF** コマンドで **DDM** ファイルおよびローカル・ファイルを指定する場合、サーバーはリモートおよびローカル・ファイルがソース・サーバーのファイルと同じかどうかを検証しません。なお、ユーザーが **DDM** ファイルを 1 つ指定すると、その同一ファイルにコピーしたり同一ファイルからコピーすることができます。

DBCS 混用フィールドをグラフィック・フィールドにコピーする方法 (**DBCS** 混用フィールド用の後続の 1 バイトのブランクをまず除去するオプションを含む) に関する詳細な情報については、58 ページの『**FMTOPT(*MAP)** または **FMTOPT(*NOCHK)** を使用する **DBCS** グラフィック・フィールド』を参照してください。

この章では、特定のコマンドについて述べる場合以外は、上記のすべてのコマンドをコピー・コマンドとして表記しています。

コピー操作を実行することができる装置ファイルおよびデータベース・ファイルを 8 ページの表 1 に示します。

表 1. コピー操作

コピー元ファイル	コピー先ファイル
DDM	DDM
ディスクett ¹	ディスクett ¹
論理	物理 ²
オープン Query ³	プリンター
物理	*PRINT ⁴
インライン・データ ⁵	テープ
テープ	
注:	
¹	コピー元ファイルとコピー先ファイルが両方ともディスクett・ファイルの場合は、コピー先ファイルはスプールされなければなりません。
²	コピー先ファイルがコピー操作前に存在していない場合、次のように指定すれば、コピー操作でコピー先ファイルとして物理ファイルが作成されます。 <ul style="list-style-type: none"> • CPYF コマンドで CRTFILE(*YES) (コピー元ファイルが物理ファイルまたは論理ファイルの場合) • CPYFRMQRYP コマンドで CRTFILE(*YES)
³	オープン Query ファイルは、CPYFRMQRYP コマンドでのみコピーすることができます。DDM ファイルを使用するオープン Query ファイルに CPYFRMQRYP を使用することはできません。
⁴	TOFILE(*PRINT) が指定された場合、コピー元ファイルのレコードは IBM 提供のプリンター・ファイル QSYSPRT にコピーされ、OUTFMT パラメーターにしたがって様式設定されます。
⁵	インライン・データ・ファイル (装置ファイルのように処理される) は、ジョブが読み取りプログラムによって読み取られるときは、バッチ・ジョブの一環として組み込まれます。

レコードをコピーしている間、一部のコピー・コマンドは、以下の機能を実行することができます。

- 最初のファイル・メンバー、特定のファイル・メンバー、メンバーの総称セット、またはすべてのファイル・メンバーをコピーします (FROMMBR および TOMBR パラメーター)。
- メンバーが存在しない場合、物理コピー先ファイルにメンバーを追加します。
- 既存のファイル・メンバーにレコードを追加する、既存のメンバーの内容を置き換える (MBROPT パラメーター)、または宛先ファイル・メンバーの重複キー・レコードを更新します。
- 以下のメソッドの 1 つによってコピーするレコードを選択します。
 - 複数様式論理ファイルをコピーする場合の、レコード様式名によるレコードの選択 (RCDFMT パラメーター)。
 - コピーを開始する相対レコード番号と、コピーを終了する相対レコード番号の指定 (FROMRCD および TORCD パラメーター)。
 - コピーを開始する特定のレコード・キーと、コピーを終了する特定のレコード・キー値の指定 (FROMKEY および TOKEY パラメーター)。
 - コピーするレコード数の指定 (NBRRCDs パラメーター)。
 - レコード中のフィールドまたはレコードの、1 つまたは複数の文字位置の内容によるレコードの選択 (INCCHAR パラメーター)。
 - レコード内の 1 つまたは複数のフィールドに入っている値によるレコードの選択 (INCREL パラメーター)。
 - コピー元ファイルが到着順処理の場合に、コピー過程でのコピー元ファイルの中の削除済みレコードの除外または組み込み (COMPRESS パラメーター)。

- コピーしたレコード、除外したレコード、またはエラー・レコード (PRINT パラメーター) を指定様式 (OUTFMT パラメーター) で印刷します。
- コピー元ファイル・レコード様式とコピー先ファイル・レコード様式が異なるレコードをコピーします (FMTOPT パラメーター)。様式が異なる場合、以下のいずれかの処置を実行することができます。
 - コピー元ファイル・レコード様式とコピー先ファイル・レコード様式の中で名前が同じで、しかもフィールド属性に互換性のあるフィールドをマップ (*MAP 値)
 - コピー先ファイル・レコード様式の中には存在しないコピー元ファイル・レコード様式中のフィールドの削除 (*DROP 値)
 - 差異はすべて無視して (左から右へ) データの直接コピー (*NOCHK 値)
- ソース・ファイルからデータ・ファイルへ、またはデータ・ファイルからソース・ファイルへコピーします。コピー元ファイルまたはコピー先ファイルが装置ファイルの場合は、この機能は自動的に行われます。両方のファイルが共にデータベース・ファイルの場合は、FMTOPT (CVTSRC) を指定しなければなりません。
- ソース物理ファイルにコピーするときは、順序番号および日付ソース・フィールドの中の順序番号およびゼロ日付を変更します (SRCOPT パラメーター)。番号付け直しを行なう場合、開始順序番号および分値が指定できます (SRCSEQ パラメーター)。
- 指定数の回復可能エラーが発生した場合、コピーを打ち切ります (ERRLVL パラメーター)。
- コピー操作の一環としてコピー先ファイルを作成します (CRTFILE パラメーター)。

ファイルのコピー: サポートされる機能:

iSeries には、データベース・ファイル用および装置ファイル用のさまざまなコピー機能があります。各機能には、関連するパラメーターとファイル・タイプがあります。

以下の表 (表 2 および 11 ページの表 3) は、レコードをコピーするのに使用できる特定のコピー機能 (コピー・コマンドを使用) をコピー元およびコピー先のファイル・タイプ別に要約しています。機能および対応するパラメーターは左側に上から下へ縦にリストし、ファイル・タイプ (および各ファイル・タイプがコピー元ファイルおよびコピー先ファイルになり得るかどうかが) は、上段横に示してあります。X は、関連したパラメーターが、それぞれのファイルのタイプおよび発生する使用方法に有効であることを示しています。

表 2. データベース・ファイルのコピー機能の要約

コピー機能	パラメーター	データベース・ファイル ¹			
		物理		論理	
		コピー元	コピー先	コピー元	コピー先
ファイル選択	FROMFILE ²	X		X	
	TOFILE		X		
メンバー選択	FROMMBR	X		X	
	TOMBR		X		
既存レコードに対する追加、置換、または更新	MBROPT		X		
コピー先ファイルの作成	CRTFILE ³	X	X	X	
コピーされたレコード、除外されたレコード、およびエラー・レコードの印刷	PRINT ⁴	X	X	X	
レコード様式による選択	RCDFMT			X	

表 2. データベース・ファイルのコピー機能の要約 (続き)

コピー機能	パラメーター	データベース・ファイル ¹			
		物理		論理	
		コピー元	コピー先	コピー元	コピー先
相対レコード番号による選択	FROMRCD	X		X ⁵	
	TORCD	X		X ⁵	
キー・フィールド値による選択	FROMKEY	X		X	
	TOKEY	X		X	
コピーするレコード数の指定	NBRRCDS	X		X	
文字内容による選択	INCCHAR	X		X	
フィールド値による選択	INCREL	X		X	
さまざまなデータベース・レコード様式の処理	FMTOPT	X	X	X	
順序番号および日付の更新	SRCOPT	X	X	X	
開始値および増分値の指定	SRCSEQ	X	X	X	
文字様式および 16 進様式の印刷	OUTFMT ⁴	X	X	X	
許可されている最大回復可能エラー	ERRLVL	X	X	X	
削除済みレコードの無視または組み込み	COMPRESS ⁶	X	X		

表 2. データベース・ファイルのコピー機能の要約 (続き)

コピー機能	パラメーター	データベース・ファイル ¹			
		物理		論理	
		コピー元	コピー先	コピー元	コピー先

注:

- ¹ DDM ファイルは、データベース・ファイルのように処理されるように見えますが、その例外が分散データ管理の資料に記載されています。
 - ² CPYFRMQRYP コマンドでは、FROMOPNID パラメーターを使用して、コピー元のオープン Query ファイルのオープン識別名を識別します。FROMFILE パラメーターは、その他のすべてのコピー・コマンドで使用されます。
 - ³ コピー先ファイルがコピー操作前に存在せず、コピー元ファイルが物理ファイルまたは論理ファイルである場合は、コピー・コマンドで CRTFILE(*YES) を指定すれば、コピー操作でコピー先ファイルとして物理ファイルを作成します。
 - ⁴ コピーで特殊な様式設定もページ見出しもないリストを生成するようにプログラム記述プリンター・ファイルを指定することもできれば、TOFILE(*PRINT) を指定して定様式リストを生成することもできます。PRINT(*COPIED) を指定すればコピー・レコードの定様式リストを生成することができ、PRINT(*EXCLD) を指定すれば INCCHAR または INCREL パラメーターによって除外されたレコードの定様式リストを生成することができ、そして PRINT(*ERROR) を指定すれば ERRLVL エラーが生じるレコードの定様式リストを生成することができます。TOFILE(*PRINT) パラメーターを指定することによってリストを要求すると、データを文字で印刷するか、文字と 16 進数様式の両方で印刷するかを、OUTFMT パラメーターで指定します。
 - ⁵ 論理ファイルの場合は、到着順アクセス・パスがあれば、FROMRCD および TORCD パラメーター値を指定することができます。
 - ⁶ 以下の場合、COMPRESS(*NO) を指定することはできません。
 - コピー先ファイル・メンバーまたはコピー先ファイル・メンバーに基づく論理ファイル・メンバーが、次の属性のいずれかを備えたキーによるアクセス・パスを持っている場合:
 - 固有キー (DDS で UNIQUE キーワードを指定)
 - 固定小数点キー・フィールドまたは論理数字キー・フィールドおよび MAINT(*REBLD) 以外
 - DDS の選択/除外仕様 (DYNSLT キーワードの指定なし) および MAINT(*REBLD) 以外
 - フィールド・レベル・マッピングまたはソース/データ変換が必要な場合 (FMTOPT パラメーター)
 - データベース・ファイル・オーバーライド (OVRDBF) コマンドでコピー元ファイルに対して EOFDLY 待機時間が指定されている場合
- 注: 削除済みレコードをコピーするには、コピー元ファイルを到着順に処理しなければなりません。

表 3. 装置ファイルのコピー機能の要約

コピー機能	パラメーター	装置ファイル							
		インライン・データ		ディスク		テープ		プリンター	
		コピー元	コピー先	コピー元	コピー先	コピー元	コピー先	コピー元	コピー先
ファイル選択	FROMFILE	X		X		X			
	TOFILE				X		X		X
メンバー選択	FROMMBR			X		X			
	TOMBR				X		X		

表 3. 装置ファイルのコピー機能の要約 (続き)

コピー機能	パラメーター	装置ファイル							
		インライン・データ		ディスク		テープ		プリンター	
		コピー元	コピー先	コピー元	コピー先	コピー元	コピー先	コピー元	コピー先
既存レコードへの追加、または既存レコードの置換	MBROPT								
コピー先ファイルの作成	CRTFILE								
コピーまたは除外されたレコードの印刷	PRINT ²	X		X	X	X	X		X
レコード様式による選択	RCDFMT								
相対レコード番号による選択	FROMRCD	X		X		X			
	TORCD	X		X		X			
キー・フィールド値による選択	FROMKEY								
	TOKEY								
コピーするレコード数の指定	NBRRCD	X		X		X			
文字内容による選択	INCCHAR	X		X		X			
フィールド値による選択	INCREL								
さまざまなデータベース・レコード様式の処理	FMTOPT								
順序番号または日付の更新	SRCOPT								
開始値および増分値の指定	SRCSEQ								
文字様式または 16 進様式の印刷	OUTFMT ²	X		X	X	X	X		X
許可されている最大回復可能エラー	ERRLVL					X			
削除済みレコードの無視または組み込み	COMPRESS								
注: ¹ コピーで特殊な様式設定もページ見出しもないリストを生成するようにプログラム記述プリンター・ファイル指定することもできれば、 TOFILE(*PRINT) を指定して定様式リストを生成することもできます。 PRINT(*COPIED) を指定すればコピー・レコードの定様式リストを生成することができ、 PRINT(*EXCLD) を指定すれば INCCHAR または INCREL パラメーターによって除外されたレコードの定様式リストを生成することができ、そして PRINT(*ERROR) を指定すれば ERRLVL エラーが生じるレコードの定様式リストを生成することができます。 TOFILE(*PRINT) パラメーターを指定することによってリストを要求すると、データを文字で印刷するか、文字と 16 進数様式の両方で印刷するかを、 OUTFMT パラメーターで指定します。									

ファイルのコピー: 基本機能:

物理または論理データベース・ファイル、オープン Query ファイル、ディスク・ファイル、テープ・ファイル、またはインライン・データ・ファイルをコピー元ファイルとして、それらのファイルからコピーすることができます。コピー先ファイルとしては、物理データベース・ファイル、ディスク・ファイル、テープ・ファイル、プログラム記述のプリンター・ファイル、または *PRINT が使用できます。

TOFILE(*PRINT) が指定されているときは、CPYSRCF コマンドは他のコピー・コマンドとは異なる様式を使用します。この様式はソース情報をより読み取りやすい様式で編成し、複数メンバーのコピーの場合はメンバーがアルファベット順にコピーおよびリストされます。

データベース・ファイルからコピーする場合、コピー先ファイルが存在しないときは CRTFILE(*YES) を指定し、TOFILE パラメーターでコピー先ファイルを作成するためのファイル名およびライブラリー名を識別しなければなりません。ディスクからディスクへコピーすることができるのは、コピー先ファイルがスプール・ファイルで、しかもディスク・スプーリング書き出しプログラムが活動している場合だけです。

コピー元ファイル (CPYFRMQRYP コマンドの場合はコピー元ファイルはオープンされない)、このコマンドは除く)、コピー先ファイル、および QSYSPRT プリンター・ファイル (TOFILE(*PRINT)、PRINT(*COPIED)、PRINT(*EXCLD)、または PRINT(*ERROR) が指定されている場合) は、SHARE(*NO) 属性を指定してオープンされます。共用ファイルの場合はコピーが正しく機能しない可能性があるため、コピー元ファイル、コピー先ファイル、または QSYSPRT プリンター・ファイルが SHARE(*YES) にオーバーライドされ、ファイルがすでにジョブの中でオープンされている場合は、エラー・メッセージが出されてコピーは打ち切られることとなります。

TOFILE(*PRINT) を指定した場合は、レコードは IBM 提供のプリンター・ファイル QSYSPRT にコピーされ、リストは OUTFMT パラメーターによって様式設定されます。

定様式リストを望まない場合、または先頭文字用紙制御を使用 (プリンター・ファイル作成 (CRTPRTF) コマンドまたはプリンター・ファイル・オーバーライド (OVRPRTF) コマンドで CTLCHAR(*FCFC) を指定) したい場合は、TOFILE パラメーターで *PRINT を指定しないで、プログラム記述プリンター・ファイル名 (QSYSPRT など) を指定してください。

ファイル・タイプおよびコピー:

あるタイプのファイルを別のタイプのファイルにコピーするときには、いくつかの考慮事項があります。

コピー元ファイルとコピー先ファイルのタイプが異なる (ソースとデータ) ときは、次のことが当てはまります。CPYFRMQRYP コマンドの場合は、コピー元ファイルは常にデータ・ファイルとして扱われます。

- コピー元ファイルまたはコピー先ファイルが装置ファイル (またはインライン・データ・ファイル) である場合は、各コピー対象レコードごとに、コピー機能が自動的にソース順序番号フィールドおよび日付フィールドを追加または削除します。
- コピー元ファイルおよびコピー先ファイルがデータベース・ファイルの場合は、操作を実行するために、FMTOPT(*CVTSRC) を指定しなければなりません。順序番号フィールドおよび日付フィールドは、装置ファイルの場合と同じように追加または削除され、各レコードのデータ部は、ファイル・レコード様式の中のフィールド定義に関係なくコピーされます。コピー先がソース物理ファイルの場合は、SRCSEQ パラメーターを使用して、SRCOPT(*SEQNBR) も指定されている場合に順序番号を作成する方法を制御することができます。

レコードの順序およびコピー:

データベース・ファイルの中で編成されるレコードの順序をアクセス・パスと呼びます。アクセス・パスには、キー・シーケンスと到着順の 2 種類があります。コピー機能では、データベース・ファイルの中のレコードを、到着順かキー・シーケンスのいずれかで処理することができます。

到着順コピーでは、コピー元ファイルの中にレコードが物理的に存在する順序でレコードを転送します。相対レコード番号が、この順序を表します。相対レコード番号とは、レコードが記憶域内で物理的に存在する位置です。レコードは常にファイルの終わりに追加されるので、相対レコード番号は、レコードがファイルの中に到着した順序を表すことになります。

キー・シーケンス・コピーでは、キー付き物理ファイルのキー値によってレコードを選択し、転送します。これにより、コピー先ファイルの中では異なる物理的順序が生じる可能性があります。コピー先ファイルは、コピー元ファイルの再編成されたバージョンになります。キー値によってファイルがコピーされるとき、次のように、ある特定のレコードの相対レコード番号が変更される可能性があります。

相対レコード番号	到着順	キー・シーケンス
1	1011	0016
2	0762	0762
3	0810	0810
4	3729	1011
5	0016	3729

COPY コマンドで FROMRCD または TORCD パラメーターを指定することによって、キー付き物理ファイルを到着順でコピーすることができます。こうすると、キー・シーケンス・アクセス・パスを使用してレコードをキー・シーケンスで検索することはありません。レコードは、到着順に検索されます。したがって、この方法は、ファイルの中の物理相対レコード位置に意味があり、元のファイルと同じになることが必要となるときに有効です。すべてのレコードを到着順にコピーするには、FROMRCD(1) を指定するのが良い方法です。キー・シーケンスを使用するよりも到着順に物理ファイルをコピーした方がより高速です。

実行するコピーの種類は、コピー元ファイルのタイプおよびコピー対象レコードの選択方式によって決まります。ファイルのコピーは一般に、ファイルにキーがある場合はキー・シーケンスを使用して行われ、それ以外の場合は到着順で行われます。

通常、キー付きファイルからキー付きファイルへのコピーでは、コピー元ファイルの物理順序に関係なく、コピー元ファイル・キーによって、レコードをキー・フィールド順にコピー先ファイルの終わりに入れます。しかし、相対レコード番号 (FROMRCD または TORCD パラメーターの使用) によってコピー元ファイルの中のレコードを選択した場合は、レコードはコピー元ファイルの中でのキー・シーケンスには関係なく、相対レコード番号順にコピー先ファイルの終わりに物理的に入れられることになります。以下の例では、レコード 3 から 5 を指定した場合の COPY コマンドの結果を示します。

コピー元ファイル			コピー先ファイル	
相対レコード番号	キー		相対レコード番号	キー
1	1011		.	—
2	0762		.	—
3	0810	< 到着 < 順 < コピー	1401	0810
4	3729		1402	3729
5	0016		1403	0016

コピー先ファイルにキー・シーケンスがあるときは、キー・シーケンス・アクセス・パスを使用すると、コピー先ファイルでレコードは正しい順序で表示されます。相対レコード番号によるコピーは、常に到着順でコピーされます。

関連概念

30 ページの『コピー対象レコードの選択』

コピー・コマンドのパラメーターを使用して、コピーしたい特定のレコードだけを選択することができます。

ファイル・コピーの完了メッセージの再送:

COPY コマンドを CL プログラムから実行する場合は、コピーされたレコード数を示す完了メッセージが、システム・オペレーターに対して直接送られることはありません。このメッセージを再送して、システム・オペレーター宛てに送ることができます。

以下の例は、SNDPGMMSG コマンドを使用して COPY コマンドを再送するサンプル CL プログラムです。

注: コード例を使用することにより、お客様は 239 ページの『コードに関する特記事項』の条件に同意することになります。

```
PGM
DCL &MSGID TYPE(*CHAR) LEN(7)
DCL &MSGDTA TYPE(*CHAR) LEN(82)
CPYF FROMFILE(LIB1/XXX) TOFILE(LIB2/XXX) +
  MBROPT(*ADD)
RCVMSG MSGID(&MSGID) MSGDTA(&MSGDTA) +
  MSGTYPE(*COMP) RMV(*NO)
SNDPGMMSG MSGID(&MSGID) MSGF(QCPFMSG) +
  MSGTYPE(*INFO) TOMSGQ(QSYSOPR) +
  MSGDTA(&MSGDTA)
ENDPGM
```

コピー機能は、コピー先ファイルに正常にコピーされた各コピー元ファイル・メンバー/ラベルごとに、次の完了メッセージの 1 つを送ります。

- CPC2955 は、通常のコピー完了メッセージです。
- CPC2956 は、COMPRESS(*NO) が指定されているときに使用されます。
- CPC2957 はレコードがコピーされなかったことを示します。
- MBROPT(*UPDADD) が指定された場合、CPC2954 が、CPC2955、CPC2956、または CPC2957 に続く 2 番目の完了メッセージとして送信されます。更新されたレコード数を示します。

コピー・エラーのモニター:

エラー発生時には、さまざまなエラー条件を示すためにエスケープ・メッセージ CPF2817 が送信されます。

このメッセージが送られたときは、コピー元ファイル・メンバーが空の場合 (これについては後述します) を除き、次の事態が生じています。

- (CRTFILE(*YES) がコピー・コマンドで指定されていたとしても) 物理ファイルが作成されませんでした。
- メンバーが物理ファイルであるコピー先ファイルに追加されませんでした。
- (MBROPT(*REPLACE) が指定されていたとしても) コピー先ファイル・メンバーが消去されませんでした。

- コピー先ファイルがオープンされていないので、ディスク・ボリュームまたはテープ・ボリューム上にファイルが作成されませんでした。コピー先ファイルがスプール・ファイルの場合は、スプール・ファイルが作成されませんでした。
- レコードがコピーされませんでした。

CPF2817 エスケープ・メッセージには、特定のエラー状態を示す診断メッセージが、少なくとも 1 つは必ず先行します。 CPF2817 エスケープ・メッセージの直前にある診断メッセージのメッセージ識別コードは、CPF2817 エスケープ・メッセージのメッセージ置き換えデータ (SNDPGMMSG コマンドの MSGDTA パラメーター) として使用されます。したがって、MONMSG コマンドの CMPDTA パラメーターを使用することによって、 CPF2817 エスケープ・メッセージから特定のエラーをモニターすることができます。

たとえば、メッセージ CPF2802 はコピー元ファイルが見つからないことを示す診断メッセージです。次のようにして、コピー元ファイルが見つからない状態のみをモニターします。

```
PGM
      /* The replacement text of escape
      CPF2817 contains the msg ID
      CPF2802 for the 'from-file not
      found' condition */
CPYF FROMFILE(NOLIB/NOFILE) TOFILE(D504/KEY) +
      FROMMBR(NOMBR) TOMBR(MBR1) MBROPT(*ADD)
MONMSG MSGID(CPF2817) CMPDTA(CPF2802) +
      EXEC(SNDPGMMSG TOPGMQ(*EXT) +
      MSG('File NOFILE in NOLIB not found'))
ENDPGM
```

CPF2817 エスケープ・メッセージによって報告されるその他のエラーも含め、from-file not found (コピー元ファイルが見つからない) 以外のすべてのエラーは、このプログラムの中で機能チェックを引き起こします。その理由は、メッセージ CPF2802 からの比較データを持っているときの CPF2817 エスケープ・メッセージだけに MONMSG コマンドが適用されるためです。

CPYFRMQRYF コマンドを実行している場合、通常それは、コピー完了後にオープン Query ファイルをクローズしません。ただし、CPYFRMQRYF コマンドをコマンド入力行から実行した場合は、OPNQRYF コマンドに TYPE(*PERM) の指定がある場合を除き、OPNQRYF コマンドが正しく実行されたあとでエラー・メッセージが生じると、このファイルはクローズされます。エラー・メッセージが生じると、サーバーは自動的に資源の再利用 (RCLRSC) コマンドを実行します。OPNQRYF コマンドに TYPE(*PERM) の指定がある場合、ファイルはサーバーによって自動的にクローズされません。

次のメッセージは、その直後に CPF2817 エスケープ・メッセージを付けて、診断メッセージとして送信される場合があります。これらのメッセージには、(通知メッセージまたはエスケープ・メッセージなどのような) 他のメッセージ・タイプとして送られるものもあります。これらのメッセージが診断メッセージ・タイプとして送られるときは、メッセージ識別コードが CPF2817 エスケープ・メッセージの置換テキストの中に現れます。MONMSG コマンドの CMPDTA パラメーターを使用することによって、状態をモニターすることができます。

CPD2807 CPD2808	CPF2806 CPF2807 CPF2808	CPF2840 CPF2841 CPF2842	CPF2872 CPF2873 CPF2874
CPD2809 CPD2810	CPF2810 CPF2811 CPF2812	CPF2843 CPF2844 CPF2847	CPF2877 CPF2878 CPF2879
CPD2811 CPD2812	CPF2813 CPF2814 CPF2816	CPF2848 CPF2849 CPF2851	CPF2881 CPF2883 CPF2884
CPD2825 CPD2968	CPF2819 CPF2820 CPF2821	CPF2853 CPF2854 CPF2855	CPF2890 CPF2891 CPF2893
CPD2969 CPD2970	CPF2822 CPF2823 CPF2825	CPF2856 CPF2857 CPF2860	CPF2960 CPF2962 CPF2963
CPD2971 CPD2972	CPF2826 CPF2827 CPF2831	CPF2861 CPF2862 CPF2863	CPF2965 CPF2969 CPF9807
CPD2973 CPD2974	CPF2832 CPF2833 CPF2834	CPF2864 CPF2865 CPF2868	CPF9808 CPF9820 CPF9830
CPD2975 CPD2976	CPF2836 CPF2837 CPF2839	CPF2869 CPF2870 CPF2871	
CPD2979 CPD2980			
CPD2981 CPF2801 CPF2802			
CPF2803 CPF2804 CPF2805			

コピー元ファイルの中のゼロ・レコードのモニター:

コピー元ファイルが物理ファイルまたは論理ファイルで、コピーされることになる 1 つまたは複数のメンバーが空のときは、コピーに関する特殊な考慮事項があります。

なお、メンバーは次の場合に空と見なされます。

- COMPRESS(*NO) が CPYF コマンドで指定され、コピー元ファイルにレコードが入っていない場合。
- COPY コマンドで COMPRESS(*YES) が指定され、コピー元ファイル・メンバーに未削除レコードが入っていない場合。

コピー元のメンバーにレコード選択の指定 (CPYFRMQRYP コマンド、または CPYF コマンドの INCCHAR および INCREL パラメーター) があり、その結果レコードが生成されない場合には、空とは見なされません。

コピー先ファイルが (*PRINT を含む) プリンター・ファイルのとき、またはコピー先ファイルが物理ファイルで、MBROPT(*ADD) または MBROPT(*UPDADD) が指定されているときは、既存のデータが破壊されないため、空のコピー元ファイル・メンバーがコピーされます。コピーされた各メンバーは、通常のコピー完了メッセージによって識別されます。コピー先ファイルがスプール・ファイルの場合は、それぞれの空のコピー元ファイル・メンバーごとに空のスプール出力ファイルが生成されます。CPYF コマンドの PRINT パラメーターで *COPIED、*EXCLD、または *ERROR を指定した場合は、空メンバーがリストに示され、レコードは何も印刷されません。

CPYFRMQRYP コマンドの場合を除けば、空のコピー元ファイル・メンバーは、ディスク・ファイルまたはテープ・ファイルにはコピーされず、MBROPT(*REPLACE) が指定されているときは物理ファイルにもコピーされることはありません。これらのタイプのコピー先ファイルの場合は、空のコピー元ファイル・メンバーはスキップされ、CPF2869 メッセージが (通知メッセージか診断メッセージかのどちらかで) 送られ、空の各メンバーを識別します。空のメンバーは、既存のデータの破壊を回避するためにスキップされます。空コピー元ファイル・メンバーがスキップされるときは、次の考慮事項が適用されます。

- テープ・ファイルまたはディスク・ファイルは、出力ボリューム上に生成されません。ディスク・ファイルがスプール・ファイルのときは、スプール出力ファイルは作成されません。
- 既存のコピー先ファイルの物理ファイル・メンバーは、消去されません。
- コピー先ファイルが存在せず、CRTFILE(*YES) がコピー・コマンドで指定された場合は、物理ファイルが作成されます。
- コピー先ファイルが物理ファイルで、コピー先ファイル・メンバーが存在しない場合は、メンバーがファイルに追加されます。

- CPYF コマンドの PRINT パラメーターで *COPIED、 *EXCLD、または *ERROR を指定した場合は、空メンバーはリストに示されません。

コピー・コマンドで総称名、または FROMMBR パラメーターに対して *ALL を指定した場合は、スキップされた空コピー元ファイル・メンバーは、通知メッセージとして送られるメッセージ CPF2869 によって識別されます。すべてのコピー元ファイル・メンバーがスキップされる場合は、すべてに関する CPF2869 通知メッセージおよびそれに続く CPF2817 エスケープ・メッセージの後で、CPF2870 診断メッセージが送られます。

コピー・コマンドで単一メンバー名または FROMMBR (*FIRST) を指定した場合、またはコピー元ファイルに対して、単一メンバーの処理を強制するオーバーライドがあるときは、スキップされた空メンバーは診断メッセージ CPF2869 によって識別されます。 CPF2869 診断メッセージの後に CPF2817 エスケープ・メッセージが続きます。

次の例では、コピー元ファイルおよびコピー先ファイルが両方ともデータベース・ファイルで、 EMPTY1 および EMPTY2 はコピー元ファイルの中の空メンバーです。

注: コード例を使用することにより、お客様は 239 ページの『コードに関する特記事項』の条件に同意することになります。

```
PGM
      /* No need to monitor for zero records
      when MBROPT(*ADD) specified */
CPYF  FROMFILE(D504/GEORGE) TOFILE(D504/KEN) +
      FROMMBR(EMPTY1) TOMBR(MBR1) MBROPT(*ADD)
CPYF  FROMFILE(D504/GEORGE) TOFILE(D504/KEN) +
      FROMMBR(EMPTY2) TOMBR(MBR2) MBROPT(*REPLACE)
MONMSG MSGID(CPF2817) CMPDTA(CPF2869) +
      EXEC(CLRPFM FILE(D504/KEN) MBR(MBR2))
      /* Monitor for zero records and
      send a message when all members
      to copy are empty */
CPYF  FROMFILE(D504/GEORGE) +
      TOFILE(D504/NEWFILE) FROMMBR(EMPTY*) +
      TOMBR(NEWMBR) MBROPT(*REPLACE)
MONMSG MSGID(CPF2817) CMPDTA(CPF2870) +
      EXEC(SNDPGMMMSG TOPGMQ(*EXT) +
      MSG('All members to copy are empty'))
ENDPGM
```

最初の CPYF コマンドでは MBROPT(*ADD) が指定されているので、コピー元ファイル・メンバーが空の場合に、エスケープ・メッセージがプログラムに送られることはありません。コピーの前に MBR1 が存在しない場合は、(コピー元ファイル・メンバーが空であるかデータが入っているかに関係なく) MBR1 はコピー先ファイルに追加されるということに注意してください。

2 番目の CPYF コマンドでは、コピーの際、コピー元ファイル・メンバーが空の場合に、コピー先ファイル・メンバーを消去することがないので、2 番目の CPYF コマンドの後の MONMSG コマンドで CLRPFM コマンドを開始して、コピー元ファイル・メンバーが空のとき、コピー先ファイル・メンバーを消去します。

3 番目の CPYF コマンドでは、総称名 EMPTY* が使用され、複数のメンバーのコピーが要求されるので、コピー元のメンバーがすべて空の場合は、 CPF2817 エスケープ・メッセージは CPF2870 の比較データを持ちます。

重複コピー先メンバーの作成:

ファイルのコピー操作を再び実行しなくても、重複するコピー先メンバーを作成することができます。

アプリケーションで (コピー元ファイルが空でも、データが入っていても) コピー先ファイル・メンバーの中にレコードの正確なコピーを必要とするときは、代替策として物理ファイル・メンバー消去 (CLRPFM) コマンドを使用します。

```
CLRPFM FILE(X) MBR(XYZ)
CPYF FROMFILE(Y) TOFILE(X) TOMBR(XYZ) +
    MBROPT(*ADD)
```

MBROPT(*ADD) が指定されているので、ファイル Y にデータがない場合でも CPYF コマンドは正常終了します。ファイル X にある MBR(XYZ) には、ファイル Y のメンバーにあるレコードの正確なコピーが入ります。

CCSID の CPYFRMQRYP コマンド・サポート:

Query ファイルからのコピー (CPYFRMQRYP) コマンドは、文字フィールドおよび 2 バイト文字セット (DBCS) フィールドに関するコード化文字セット ID (CCSID) 変換を行います。

オープン Query ファイル (OPNQRYP) コマンドは、すべての文字フィールドと DBCS フィールドを現行ジョブの CCSID に変換します。ただし、65535 の CCSID を持つフィールド、または MAPFLD パラメーターに *HEX が指定された場所は除きます。現行ジョブの CCSID が 65535 の場合、OPNQRYP では変換は行われません。Query ファイルからのコピー (CPYFRMQRYP) コマンドもまた、コピー先ファイル・フィールドの CCSID への変換を行うことができます。したがって、二重変換されてデータが消失する可能性があります。CPYFRMQRYP コマンドを使用する場合、OPNQRYP を実行する前にジョブの CCSID を 65535 に変更して二重変換を回避してください。

CPYFRMQRYP は、異なる照会様式を使用します。これは、文字および DBCS フィールドの CCSID を除いてオープン Query ファイル様式と同じです。この照会様式の CCSID は、以下の条件に従って決定されます。

- OPNQRYP ジョブの CCSID が 65535 の場合、照会様式の文字および DBCS フィールドにはすべてオープン Query ファイル様式と同じ CCSID になります。
- OPNQRYP ジョブの CCSID が 65535 でない場合、照会様式の文字および DBCS フィールドにはすべて CCSID があり、OPNQRYP ジョブのそのフィールド・タイプを基にして CCSID に関連する 1 バイト、混合、または 2 バイトの CCSID にリセットされます。このとき、65535 の CCSID を持つフィールドは変更されません。また、OPNQRYP ジョブの CCSID に関連する混合または 2 バイトの CCSID がない場合は、65535 が使用されます。

関連情報

CCSID の処理

CCSID の CPYSRCF コマンド・サポート:

ソース・ファイル・コピー (CPYSRCF) コマンドを使用すると、データはコピー元ファイルからコピー先ファイルの CCSID に自動的に変換されます。文字データを変換したくない場合は、FMTOPT(*NOCHK) を指定して CPYF コマンドを使用してください。

関連情報

ソース・ファイル・コピー (CPYSRCF) コマンド

CPYF コマンド

コピー・コマンドでのヌル値のサポート:

CPYF および CPYFRMQRYP コマンドを使用して、ヌル可能フィールドを含むファイルをコピーできます。FMTOPT パラメーターを使用して、ヌル可能フィールドをマップできます。

INCREL パラメーターを使用すれば、フィールドがヌルであるかどうかに基づいてレコードを選択できます。

コピー先ファイルにレコードをコピーしている間、以下のコマンドはコピー元ファイルのヌル値を無視しません。

CPYTOTAP

CPYTODKT

CPYFRMTAP

CPYFRMDKT

コピー先ファイルにレコードをコピーしている間、CPYF または CPYFRMQRYF コマンドの以下の条件または値で使用するとコピー元ファイルのヌル値は無視されます。

FMTOPT(*NOCHK)

FMTOPT(*CVTSRC)

コピー先装置ファイル

ヌル値を含むレコードの選択は依然として可能です。ただし、(ヌル値ではなく) バッファ内のユーザー指定値またはデフォルト値のみがコピー先ファイルにコピーされます。この場合、ヌル値は再予約されません。コピー・コマンドが実行されているときに作成されるあらゆる印刷リスト (TOFILE(*PRINT)、PRINT(*COPIED)、PRINT(*EXCLUDE)、および PRINT(*ERROR) を含む) もヌル値を無視します。

関連情報

CPYF コマンド

CPYFRMQRYF コマンド

物理ファイルまたは論理ファイルのコピー

iSeries 上の物理ファイルまたは論理ファイル (コピー元ファイル) を、まだ存在していない別の物理ファイル (コピー先ファイル) にコピーするには、CPYF コマンドを使用することができます。

たとえば、次のようにできます。

```
CPYF FROMFILE(PERSONNEL/PAYROLL)
      TOFILE(TESTPAY/PAYROLL) MBROPT(*ADD)
      CRTFILE(*YES) ERLVL(10)
```

フル・サービス・コピー・サポート

多数のパラメーターによって変更される種々のコピー・コマンドにより、非常に柔軟にデータをコピーすることができます。たとえば、通常では、データを既存のファイル (つまりコピー先ファイル) にコピーすることができます。上記の例に示されているように、CPYF または CPYFRMQRYF コマンドで CRTFILE パラメーターを使用して、コピー操作中にコピー先ファイルを作成することができます。詳細については、22 ページの『コピー先ファイルの作成 (CRTFILE パラメーター)』を参照してください。

iSeries コピー・コマンドに関する基本機能については、6 ページの『ファイルのコピー: 概要』を参照してください。

必要な情報のみをコピーする

コピー機能では、ファイルの中の選択したレコードおよびメンバーのみを指定することができます。

- 24 ページの『レコードの追加、置換、および更新 (MBROPT パラメーター)』

- 44 ページの『コピー対象メンバーの選択』
- 30 ページの『コピー対象レコードの選択』

異なる様式およびサーバーにまたがってコピーする

- 48 ページの『異なるデータベース・レコード様式間でのコピー (FMTOPT パラメーター)』 ソース・ファイルからデータ・ファイルへ、またはデータ・ファイルからソース・ファイルへコピーすることができます。コピー元ファイルまたはコピー先ファイルが装置ファイルの場合は、この機能は自動的に行われます。両方のファイルが共にデータベース・ファイルの場合は、FMTOPT (CVTSRC) を指定しなければなりません。どちらかのファイルが装置ファイルまたはインライン・データ・ファイルである場合は、FMTOPT パラメーターは適用されません。
- 73 ページの『異なるサーバー間でのコピー』 これは、特に、データウェアハウジングを使用している場合や、他のプラットフォームから iSeries にデータを移動するために既存のエクスポート製品を使用したい場合に重要です。

特定の必要性を満たすようにコピー機能を機能させる

コピー機能を介してユーザーが使用可能なオプションを注意深く使用して、広範囲にわたるタスクを実現することができます。

- 43 ページの『レコードの印刷 (PRINT、OUTFMT、および TOFILE(*PRINT) パラメーター)』
- 64 ページの『ソース・ファイルの順序番号フィールドおよび日付フィールドの追加または変更 (SRCOPT および SRCSEQ パラメーター)』
- 135 ページの『ファイルのコピー時におけるエラーの防止』
- 133 ページの『パフォーマンス』

関連概念

6 ページの『ファイルのコピー: 概要』

コピー機能を使用すると、装置ファイル相互間、データベース・ファイル相互間 (分散 DB2 マルチシステム・ファイルを含む)、または装置ファイルとデータベース・ファイル相互間で、データを移動することができます。

22 ページの『コピー先ファイルの作成 (CRTFILE パラメーター)』

データの受け入れ先になる既存のコピー先ファイルがないときに、物理ファイルまたは論理ファイルをコピーするには、CRTFILE(*YES) を指定することによってコピー先ファイルを作成することができます。

44 ページの『コピー対象メンバーの選択』

iSeries には、ファイル・メンバーをコピーするためのオプションがいくつかあります。

24 ページの『レコードの追加、置換、および更新 (MBROPT パラメーター)』

いくつかの iSeries コピー・コマンドでは、MBROPT パラメーターにさまざまな属性を指定してコピー先ファイル内にデータを追加、または既存のデータを置換することができます。

30 ページの『コピー対象レコードの選択』

コピー・コマンドのパラメーターを使用して、コピーしたい特定のレコードだけを選択することができます。

関連情報

CPYF コマンド

CPYFRMQRYF コマンド

コピー先ファイルの作成 (CRTFILE パラメーター)

データの受け入れ先になる既存のコピー先ファイルがないときに、物理ファイルまたは論理ファイルをコピーするには、 CRTFILE(*YES) を指定することによってコピー先ファイルを作成することができます。

そのような場合、新しいコピー先ファイルの名前を TOFILE パラメーターで指定するとともに、ユーザーが所要の権限を持っている既存のライブラリーの名前を使用してそのコピー先ファイル名を修飾してください。(CRTPF コマンドに対する権限も持っていないければなりません)。異なるファイルまたはライブラリーに指定し、作成されたコピー先ファイルをオーバーライドすることはできません。

CRTFILE(*YES) は、メンバーおよびレコードを新しいファイルに追加します。

新しく作成されたファイルは、特定の権限、機能、および関連したユーザー・プロファイルを持っています。ご使用のサーバーは、 CPYF または CPYFRMQRYP コマンドを使用するかどうかに基づいて、新しいファイルに異なる識別コードおよび属性を指定します。

関連概念

20 ページの『物理ファイルまたは論理ファイルのコピー』

iSeries 上の物理ファイルまたは論理ファイル (コピー元ファイル) を、まだ存在していない別の物理ファイル (コピー先ファイル) にコピーするには、CPYF コマンドを使用することができます。

23 ページの『ファイル・コピー (CPYF) によって作成されるコピー先ファイルの権限、ユーザー・プロファイル、およびファイル機能』

ファイル・コピー (CPYF) コマンドによってローカル物理ファイルが作成された場合は、その作成されたコピー先ファイルには、コピー元ファイルのすべての権限が付与されます。これらの権限には、共通リスト、私用リスト、および権限リストが含まれています。

『CPYF または CPYFRMQRYP コマンドのいずれかでの CRTFILE(*YES) の指定』

CPYF コマンドと CPYFRMQRYP コマンドでは、CRTFILE(*YES) を指定した場合の結果が異なります。

CPYF または CPYFRMQRYP コマンドのいずれかでの CRTFILE(*YES) の指定:

CPYF コマンドと CPYFRMQRYP コマンドでは、CRTFILE(*YES) を指定した場合の結果が異なります。

CPYF コマンドで CRTFILE(*YES) を指定した場合、作成されるコピー先ファイルのレコード様式とアクセス・パスのタイプは、コピー元ファイルと同じになります。新しいコピー先ファイルのファイル・レベルおよび様式レベル識別コードは、コピー元ファイルのファイル・レベルおよび様式レベル識別コードと同じになります。コピーされるコピー元ファイル・メンバーのテキストが、作成されたコピー先ファイル・メンバーのテキストとして使用されます。

コピー元ファイルが論理ファイルの場合は、物理ファイル属性、SIZE(*NOMAX)、ALLOCATE(*NO)、および CONTIG(*NO) がサーバーによって割り当てられます。コピー元ファイルが複数のレコード様式を持つ論理ファイルである場合は、コピー先ファイルは CPYF コマンドの RCDFMT パラメーターで指定されたレコード様式で作成されます。

CPYFRMQRYP コマンドで CRTFILE(*YES) を指定する場合、新しいコピー先ファイルのファイル・レベルおよび様式レベルの識別コードは、その新しいコピー先ファイルが作成されたときに生成されます。また、物理ファイルの属性は、対応するオープン Query ファイル (OPNQRYP) コマンドの FILE パラメーターに指定された最初のファイルと一致しています。ただし、いくつかの属性は、サーバーによって割り当てられます。ファイルは、CONTIG(*NO)、SIZE(*NOMAX)、ALLOCATE(*NO)、AUT(*NORMAL)、および FILETYPE(*DATA) として作成されます。

指定された様式で各フィールドの名前、タイプ、長さ、ヌル可能、日付または時刻様式、区切り記号、および小数点以下の桁数の属性が使用されます。このファイルは、キー・フィールドを持たず、到着順物理ファイルとして作成されます。

いくつかの場合、OPNQRYF コマンドは、新しいコピー先ファイルに指定される様式を変更します。OPNQRYF コマンドが次のグループ化関数のいずれかを使用している場合は、新しいコピー先ファイルの様式はヌル可能になります。

- %STRDEV
- %VAR
- %SUM
- %AVG
- %MIN
- %MAX

注: 様式が変更された新しいコピー先ファイルは、OPNQRYF コマンドに指定された様式レベル識別コードとは異なる様式レベル識別コードを持ちます。

関連概念

22 ページの『コピー先ファイルの作成 (CRTFILE パラメーター)』

データの受け入れ先になる既存のコピー先ファイルがないときに、物理ファイルまたは論理ファイルをコピーするには、CRTFILE(*YES) を指定することによってコピー先ファイルを作成することができます。

30 ページの『指定レコード様式名の使用によるレコードの選択 (RCDFMT パラメーター)』

RCDFMT パラメーターを使用して、特定のレコード様式のレコードをコピー対象として選択することができます。このパラメーターは CPYF コマンドでのみ使用できることに注意してください。

ファイル・コピー (CPYF) によって作成されるコピー先ファイルの権限、ユーザー・プロファイル、およびファイル機能:

ファイル・コピー (CPYF) コマンドによってローカル物理ファイルが作成された場合は、その作成されたコピー先ファイルには、コピー元ファイルのすべての権限が付与されます。これらの権限には、共通リスト、私用リスト、および権限リストが含まれています。

CPYFRMQRYF によってローカル物理ファイルが作成された場合は、対応するオープン Query ファイル (OPNQRYF) コマンドの FILE パラメーターに指定された最初のファイルの権限が付与されます。これらの権限には、共通リスト、私用リスト、および権限リストが含まれています。

CPYF および CPYFRMQRYF のいずれの場合でも、作成されたコピー先ファイルの所有者は、コピー・コマンドを実行中のユーザー・プロファイルになります。コピー・コマンドを実行中のユーザーは、オブジェクトに対する *ALL 権限を継承します。このことは、ユーザーがグループ・プロファイルのメンバーであり、さらに、ユーザーがプロファイルに関して指定された OWNER(*GRPPRF) を持っていない場合以外は適用されます。

OWNER(*GRPPRF) を指定した場合は、グループ・プロファイルがコピー先ファイルの所有者になります。この場合、コピー・コマンドを実行中のユーザー・プロファイルが、新しいファイルに対してデータを書き込んだり、メンバーを加えたりするための十分な権限を持っていない場合、コピー・コマンドは失敗します。

作成されたコピー先ファイルは、コピー元ファイルのファイル機能を保持しません。コピー先ファイルでは、コピー元ファイルで許されていたかどうかにかかわらず、更新、削除、読み取り、および書き込み操作ができます。以下は、新しいコピー先ファイルに関する特殊な考慮事項です。

- メンバーにコピーされるレコードの数が、作成されたコピー先ファイルの最大サイズを超える場合は、システム・オペレーターの介入なしでコピー先ファイルは拡張されます。
- コピー元ファイルが SQL 表、ビュー、または索引の場合、作成されたコピー先ファイルは SQL 表ではない物理ファイルになります。ただし、コピー元ファイルに LOB、データ・リンク、またはユーザー定義のタイプが含まれている場合には、作成されるコピー先ファイルは SQL 表になります。
- コピー元ファイルが SQL 表の場合は、デフォルト値は保存されません。デフォルト・データ・タイプ値が、作成される物理ファイルに適用されます。
- コピー元ファイルにそれに関連したトリガー・プログラムがある場合は、CPYF コマンドおよび CPYFRMQRYP コマンドは、CRTFILE パラメーターが使用されるときは、トリガー情報をコピー先ファイルにコピーしません。
- 制約付きファイルから新規ファイル (CRTFILE(*YES)) を作成する場合、制約定義は新規ファイルにコピーされません。
- ユーザー定義機能付きファイルから新規ファイル (CRTFILE(*YES)) を作成する場合、ユーザー定義機能は新規ファイルにコピーされません。

関連概念

22 ページの『コピー先ファイルの作成 (CRTFILE パラメーター)』

データの受け入れ先になる既存のコピー先ファイルがないときに、物理ファイルまたは論理ファイルをコピーするには、CRTFILE(*YES) を指定することによってコピー先ファイルを作成することができます。

レコードの追加、置換、および更新 (MBROPT パラメーター)

いくつかの iSeries コピー・コマンドでは、MBROPT パラメーターにさまざまな属性を指定してコピー先ファイル内にデータを追加、または既存のデータを置換することができます。

該当するコマンドは、CPYF、CPYFRMQRYP、CPYFRMTAP、および CPYSRCF です。さらに、CPYF コマンドは、コピー先ファイル・メンバーの重複キー・レコードの更新および非重複キー・レコードの追加が可能になります。

これらのタスクを行うには、MBROPT パラメーターで *REPLACE、*ADD、または *UPDADD を指定できます。

関連概念

20 ページの『物理ファイルまたは論理ファイルのコピー』

iSeries 上の物理ファイルまたは論理ファイル (コピー元ファイル) を、まだ存在していない別の物理ファイル (コピー先ファイル) にコピーするには、CPYF コマンドを使用することができます。

29 ページの『トリガー・プログラムを使用するファイルへのレコードのコピー』

トリガーを使用すれば、特定の変更操作が発生したときに自動的にコピー操作を実行することができます。

ファイルをコピーする際の *REPLACE の指定:

*REPLACE を指定すると、本質的にはメンバーを消去することになります。コピーされたレコードが、操作が完了するときにメンバー内にある唯一のレコードになります。MBROPT(*REPLACE) を指定するには、メンバーを消去する権限を持っていないければなりません。

CPYFRMQRYP コマンド以外のコピー・コマンドで *REPLACE を指定した場合、コピー元ファイルにレコードが入っていないと、コピー・コマンド処理は正常に行われません。CPYFRMQRYP コマンドで *REPLACE を指定した場合は、オープン Query ファイルにレコードが入っていてもコピー先ファイルのメンバーが消去されます。

*REPLACE は、CPYSRCF コマンドのデフォルト値です。CPYSRCF 以外のコピー・コマンドでは、デフォルト値 *NONE を持っていますが、*NONE が有効なのは、装置ファイルへのコピーの場合だけです。

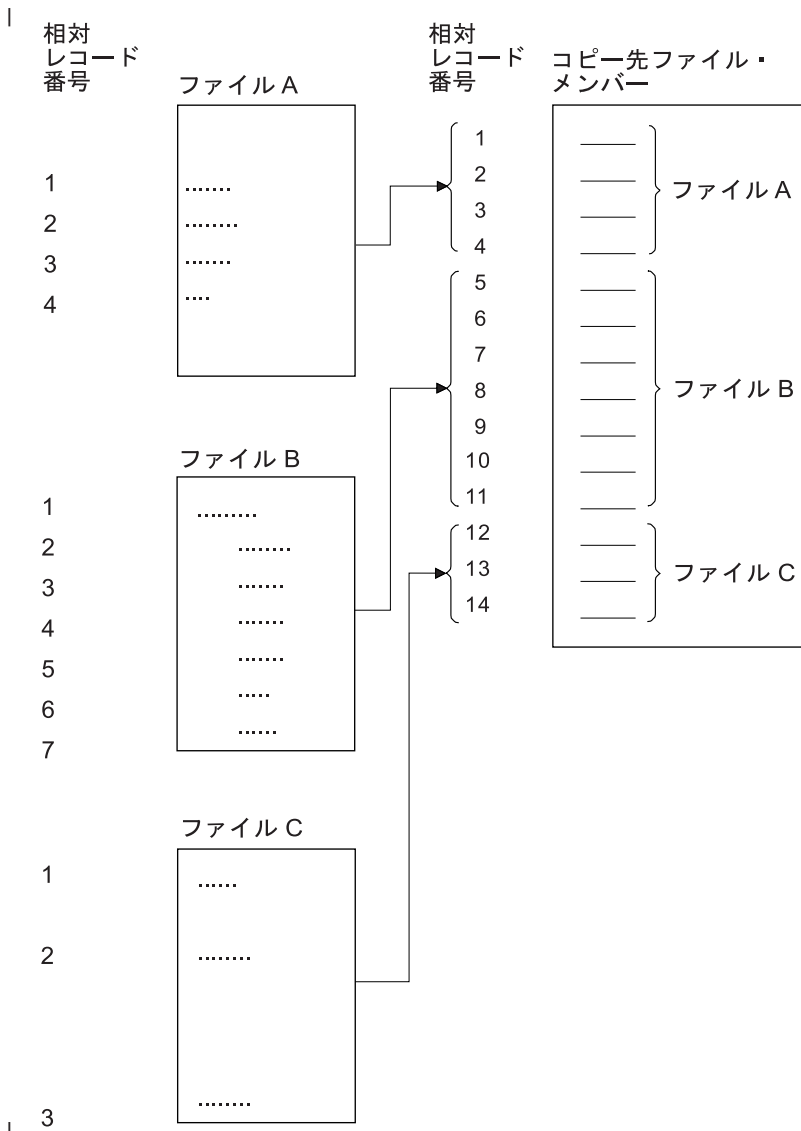
ファイルをコピーする際の *ADD の指定:

*ADD を指定すると、各コピー対象レコードはメンバーの中の既存のレコードの終わりに追加されます。

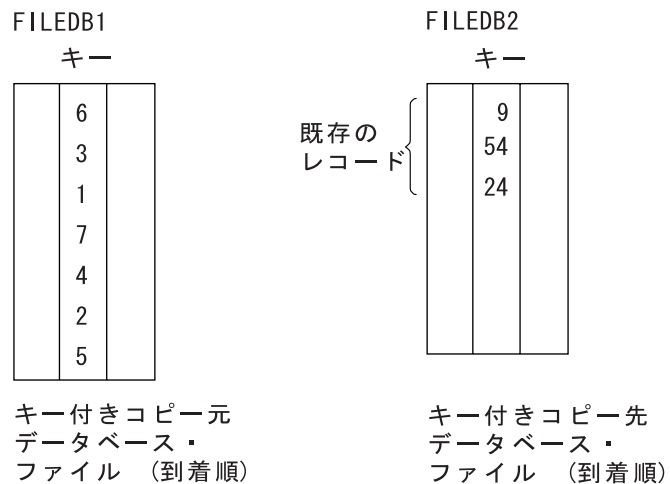
キー付きファイルの場合でもやはりこのことが当てはまるということに注意してください。ただし、キー付きファイルの場合、追加されたレコードはキー順アクセス・パスによってアクセスすると、キー順にマージされたように見えます。Query ファイルからのコピーの場合、結果のファイルの相対レコード番号はコピー元のファイルの相対レコード番号に対応していないことがあります。

*ADD を指定したときは、コピー元ファイルにレコードが入っていない場合でも、コピーは正常に完了します。

- | MBROPT(*ADD) を指定して、キーのないデータベース・ファイルに 3 つのファイルのコピーを行った場合、結果のコピー先ファイルは以下の図のようになります。

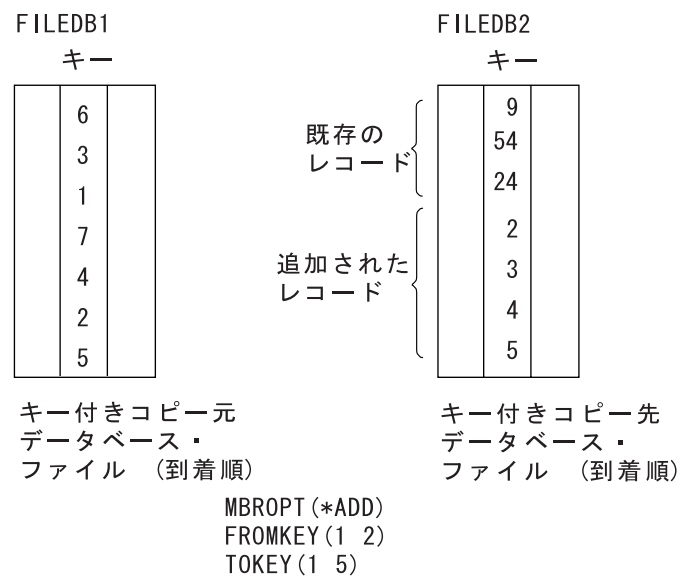


MBROPT(*ADD) を指定すると、ファイルがたとえキー・シーケンス・ファイルであっても、レコードは常にファイルの終わりに物理的に追加されます。次の図では、FILEDB1 はキー付きのコピー元物理ファイルで、FILEDB2 はキー付きのコピー先物理ファイルです。これらのファイルは、記憶域内に物理的に存在しているとおりに表示されています。FILEDB2 にはすでに 3 つのレコードが入っています。



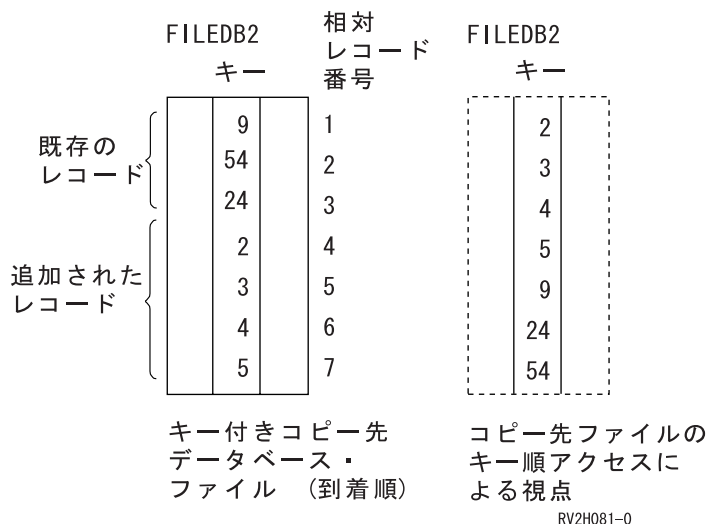
RV2H079-0

MBROPT(*ADD)、FROMKEY(1 2)、および TOKEY(1 5) を指定した場合は、4つのレコードがキー・フィールド順に、FILEDB2の終わりに追加されます。



RV2H080-0

ただし、追加されたレコードは、キー・シーケンス・アクセス・パスを介して表示すると新しいファイル内でマージされているように見えます。



コピーのためにレコードを選択するには、複数の方法があります。そうした方法の 1 つに、相対レコード番号による選択方式があります。前の例を使用すると、FILEDB2 から 3 番目のファイルにコピーするレコードを相対レコード番号の 3 番から 5 番まで選択した場合は、4、5、および 9 ではなく、24、2、および 3 のキー値のレコードがコピーされることとなります。

関連概念

41 ページの『削除済みレコードのコピー (COMPRESS パラメーター)』

コピー元ファイルからコピーされるレコードの相対レコード番号を保存するために、削除済みレコードをコピーしたい場合があります。コピー・コマンドで COMPRESS(*NO) パラメーターを指定することによって、削除済みレコードと未削除レコードの両方を物理ファイル・メンバー間でコピーすることができます。

64 ページの『ソース・ファイルの順序番号フィールドおよび日付フィールドの追加または変更 (SRCOPT および SRCSEQ パラメーター)』

ファイルのコピー時には、順序番号フィールド、日付フィールド順序番号、および日付フィールドを追加または変更することができます。

31 ページの『相対レコード番号によるレコードの選択 (FROMRCD および TORCD パラメーター)』

FROMRCD および TORCD パラメーターを使用すれば、レコード番号を指定してファイルをコピーすることができます。これらのパラメーターは CPYF コマンドでのみ使用できることに注意してください。

ファイルをコピーする際の *UPDADD の指定:

CPYF コマンドに *UPDADD を指定すると、コピー元レコードをコピー先ファイルに移動する前にコピー元キー値が作成されます。このキー値は、コピー先ファイルのキー仕様を使用して、コピー元ファイルのデータから作成されます。

必要なフィールドまたはデータのマッピング、データ変換、またはレコード選択は、キー値が作成される前に行われます。コピー先ファイルは、このキー値がそこに存在するかどうかを検査されます (コピー元ファイル・データの重複キー)。キー値がコピー先ファイルに存在する場合、そのコピー先ファイル・レコードは同じキー値を持つコピー元ファイル・レコードによって更新されます。

MBROPT(*UPDADD) を CPYF コマンドで指定する場合、次の規則が適用されます。

- コピー先ファイルは、1 次キーまたは固有キーを持つローカル・データベースの物理ファイルであることが必要です。

- CRTFILE(*YES) を指定することはできません。コピー先ファイルは CPYF を実行する前に存在している必要があります。
- CPYF は複数様式からのコピーは行えません。
- 検出された重複キーはスキップされずに、新しいコピー元ファイルのレコード値によって更新されます。重複キー・エラー (CPF5026) は、ERRLVL エラーに含まれません。
- CPF5027 は、ERRLVL エラーに含まれます。このエラーは、レコードを他のプロセスがロックしている場合に生じることがあります。このエラーを回避するために、CPYF を実行する前にジョブ内でコピー先ファイルを事前割り振りすることもできます。CRTPF および CHGPF コマンドの WAITRCD パラメーターを使用して、コピー先ファイルでレコードのロック解除を待機する時間を制限することもできます。
- 既存のすべての FMTOPT 値は、許可されます。ただし、MBROPT(*UPDADD) を使用している場合は、更新したくないレコードの更新を回避するようにしてください。必要でなければ、同じレコードを複数回更新することも回避してください。
- FMTOPT(*NOCHK) が指定されている場合、またはコピー元ファイルが装置ファイルである場合、重複キーの判別に NULL は使用されません。
- コピー先ファイルに対して最低でも次の権限を持っている必要があります。
 - オブジェクト操作権 (*OBJOPR)
 - 追加権 (*ADD)
 - 更新権 (*UPD)

トリガー・プログラムを使用するファイルへのレコードのコピー:

トリガーを使用すれば、特定の変更操作が発生したときに自動的にコピー操作を実行することができます。

トリガー・プログラムは、一連のトリガー・アクションを含むプログラムです。トリガー・アクションとは、特定の変更操作 (トリガー・イベント) が特定の表またはファイルで発生したときに自動的に実行される、一連のアクション (高水準言語ステートメント、SQL ステートメント、または iSeries ユーティリティ) です。

レコードのコピー先となる物理ファイルが、そのファイルに関連した *INSERT トリガー・プログラムを持つ場合、レコードがファイルにコピーされるたびにトリガー・プログラムが呼び出されます。削除済みレコードがコピーされた場合は、トリガー・プログラムは呼び出されません。トリガー・プログラム実行中にエラーが発生した場合、コピー操作は失敗します。しかし、エラーが発生する前にコピーされたファイルは、コピー先ファイルに残ります。

コピー元ファイルに、関連したトリガー・プログラムがある場合、CPYF コマンドおよび CPYFRMQRYP コマンドで CRTFILE パラメーターが使用されても、トリガー情報をコピー先ファイルにコピーしません。

関連概念

24 ページの『レコードの追加、置換、および更新 (MBROPT パラメーター)』

いくつかの iSeries コピー・コマンドでは、MBROPT パラメーターにさまざまな属性を指定してコピー先ファイル内にデータを追加、または既存のデータを置換することができます。

関連情報

トリガー

データベース・プログラミング

コピー対象レコードの選択

コピー・コマンドのパラメーターを使用して、コピーしたい特定のレコードだけを選択することができます。

関連概念

13 ページの『レコードの順序およびコピー』

データベース・ファイルの中で編成されるレコードの順序をアクセス・パスと呼びます。アクセス・パスには、キー・シーケンスと到着順の 2 種類があります。コピー機能では、データベース・ファイルの中のレコードを、到着順かキー・シーケンスのいずれかで処理することができます。

20 ページの『物理ファイルまたは論理ファイルのコピー』

iSeries 上の物理ファイルまたは論理ファイル (コピー元ファイル) を、まだ存在していない別の物理ファイル (コピー先ファイル) にコピーするには、CPYF コマンドを使用することができます。

関連情報

データベース・プログラミング

制御言語 (CL)

指定レコード様式名の使用によるレコードの選択 (RCDFMT パラメーター):

RCDFMT パラメーターを使用して、特定のレコード様式のレコードをコピー対象として選択することができます。このパラメーターは CPYF コマンドでのみ使用できることに注意してください。

論理ファイルから物理ファイルへコピーするとき、論理ファイルに複数のレコード様式がある場合は、FMTOPT(*NOCHK) を指定しない限り、レコード様式名を指定しなければなりません。

FMTOPT(*NOCHK) を使用する場合は、RCDFMT(*ALL) を指定して、すべてのコピー元ファイル・レコード様式をコピー先ファイルにコピーすることができます。このレコード様式名を使用して、コピーするレコードが選択されます。

以下の例は、コピー・コマンドでレコード様式 ORDHDR を使用して、論理ファイル ORDFILL から物理ファイル INVOICE にレコードをコピーする方法を示しています。

```
CPYF FROMFILE(DSTPRODLB/ORDFILL) +  
      TOFILE(DSTPRODLB/INVOICE) RCDFMT(ORDHDR) +  
      MBROPT(*ADD)
```

複数のレコード様式を持つ論理ファイルから装置ファイルへコピーするときは、使用する単一のレコード様式を指定することも、RCDFMT(*ALL) を指定してすべてのレコード様式を使用してコピーすることもできます。レコード様式の長さが異なっている場合は、短い方のレコードの後ろに空白が埋め込まれます。

関連概念

22 ページの『CPYF または CPYFRMQRYP コマンドのいずれかでの CRTFILE(*YES) の指定』

CPYF コマンドと CPYFRMQRYP コマンドでは、CRTFILE(*YES) を指定した場合の結果が異なります。

36 ページの『指定レコード数の選択 (NBRRCD S パラメーター)』

FROMKEY パラメーターまたは FROMRCD パラメーターを指定するときは、TOKEY パラメーターまたは TORCD パラメーターの代わりに、コピーするレコード数 (NBRRCD S パラメーター) を指定することができます。

41 ページの『削除済みレコードのコピー (COMPRESS パラメーター)』

コピー元ファイルからコピーされるレコードの相対レコード番号を保存するために、削除済みレコード

をコピーしたい場合があります。コピー・コマンドで COMPRESS(*NO) パラメーターを指定することによって、削除済みレコードと未削除レコードの両方を物理ファイル・メンバー間でコピーすることができます。

関連情報

DB2 マルチシステム (iSeries 用)

相対レコード番号によるレコードの選択 (FROMRCD および TORCD パラメーター):

FROMRCD および TORCD パラメーターを使用すれば、レコード番号を指定してファイルをコピーすることができます。これらのパラメーターは CPYF コマンドでのみ使用できることに注意してください。

相対レコード番号は、キー付き論理ファイル以外のファイル・タイプをコピー元ファイルとするコピーの場合に指定することができます。キー付き物理ファイルは、FROMRCD パラメーターまたは TORCD パラメーターに対して相対レコード番号を指定すれば、到着順にコピーすることができます。レコードのコピーは、

- 指定したレコード番号 (FROMRCD パラメーター) から指定したレコード番号 (TORCD パラメーター) まで、または
- 指定したレコード数 (NBRRCDS パラメーター) がコピーされてしまうまで行われます。

指定した終了レコード番号またはレコード数に達する前に、ファイルの終わりに達した場合でも、コピーは正常に完了します。

相対レコード番号を指定した場合、レコードのコピーは指定した相対レコード番号のレコードから始まり、コピー元データベース・ファイルでのレコードの物理的な順序で行われます。これは、物理ファイルにキー順アクセス・パスがある場合であっても同様です。COMPRESS パラメーターを FROMRCD および TORCD パラメーターと共に使用して、どのレコードをコピー対象として選択するかをさらに定義することができます。

コピー元ファイルが、到着順アクセス・パスを持つ物理ファイルまたは論理ファイルの場合は、TORCD の値はその前にある削除レコードと未削除レコードの両方をカウントする相対レコード番号です。コピー元ファイルが装置ファイルまたはインライン・データ・ファイルの場合は、TORCD の値は未削除レコードのみをカウントするレコード番号です (I 様式ディスク・ファイルの場合でも同様)。

削除されたレコードは、削除されていないレコードの間に位置を保持しています。ただし、指定されたサブセットの中であって、COMPRESS(*NO) が指定された場合、その相対レコード番号は必ずしも保持しません。COMPRESS(*YES) を指定した場合、削除されたレコードはスキップされ、コピーされません。この場合、指定されたレコード番号 (FROMRCD パラメーター) が削除レコードであるときは、コピーはその後の最初の削除されていないレコードから始まります。

以下の例は、このコマンドを使用して、ファイル EMP1 の相対レコード番号 500 から相対レコード番号 1000 までのレコードをファイル EMP1T にコピーする方法を示しています。

```
CPYF FROMFILE(PERSONNEL/EMP1) +
      TOFILE(TESTLIB1/EMP1T) MBROPT(*REPLACE) +
      FROMRCD(500) TORCD(1000)
```

注: レコード番号を使用してレコードを選択する場合は、同じ CPYF コマンドでレコード・キーを使用 (FROMKEY/TOKEY パラメーター) してレコードを選択することはできません。

分散ファイルに対して FROMRCD および TORCD パラメーターを使用する方法については、DB2 Multisystem for iSeries を参照してください。

関連概念

25 ページの『ファイルをコピーする際の *ADD の指定』

*ADD を指定すると、各コピー対象レコードはメンバーの中の既存のレコードの終わりに追加されません。

レコード・キーによるレコードの選択 (FROMKEY および TOKEY パラメーター):

レコード・キーは、キー付きデータベース・ファイルからコピーする場合にのみ指定できます。このパラメーターは CPYF コマンドでのみ使用できることに注意してください。

レコードのコピーができるのは、

- 指定したキー値 (FROMKEY パラメーター) から指定したキー値 (TOKEY パラメーター) まで、または
- 指定したレコード数 (NBRRCDS パラメーター) に達するまでです。

指定した終了キー値またはレコード数に達する前に、ファイルの終わりに達した場合でも、コピーは正常に完了します。

FROMKEY の値に一致するキーを持つレコードがコピー元ファイル・メンバーに存在せず、指定した値よりも大きいキーを持つレコードが少なくとも 1 つはある場合、最初にコピーされるレコードは、FROMKEY の値より大きいキーを持つ最初のレコードです。指定したキー値がメンバー中のどのレコードよりも大きい場合は、エラー・メッセージが出され、メンバーはコピーされません。

FROMKEY パラメーターまたは TOKEY パラメーターで *BLDKEY を指定すれば、キーの中のフィールドに対する文字値および数値の通常の表示様式によるリストを使用することができます。各要素は、対応するキー・フィールド・データ・タイプに変換されます。次いで、データベースに対する複合キー値 (複数のフィールドで構成されるキー) が提供されます。

完全なデータベース・キーに含まれるより少ない数の値を指定した場合は、部分キーが作成され、データベースに渡されます。データベース・キーに含まれるよりも多い数の値を指定した場合は、終了エラーが起きます。コマンドは値は常にキーの中で最左端から連続するフィールドに対して値を適用するので、フィールドをスキップすることはできません。

文字フィールドの右側には空白が埋め込まれます。数字フィールドは正しいゼロ埋め込みによって、暗黙指定の小数点に調整されます。

外部文字様式で数字フィールドを指定する場合の通常の規則がすべて適用されます。 *NAN (Not a Number: 数ではない) という浮動小数点値は使用できません。

さまざまな警告メッセージを解釈できるように、33 ページの『コピー操作で行われるキー・ストリングの比較』を理解しておく必要もあります。

注: レコード・キーを使用してレコードを選択する場合は、同じ CPYF コマンドでレコードを選択するのに相対レコード番号 (FROMRCD/TORCD パラメーター) を使用することはできません。

キー付き物理ファイルからレコード・キーによってレコードを選択するときは、COMPRESS(*NO) を指定してはなりません。削除済みレコードは、ファイルのキー順アクセス・パスには含まれていないため決してコピーされることはなく、したがって圧縮は自動的に行なわれません。

この方式では、削除済みレコードはコピーの際に取り消されるので、たとえ MBROPT(*REPLACE) を指定した場合でも、相対レコード番号が新しいファイルの中で変更されていることもあり得ます。

ユーザーは、固定長 2 進文字キー・フィールドについて、FROMKEY パラメーターと TOKEY パラメーターに *BLDKEY を指定するときは、固定長のフィールドの長さに等しいオペランドを入力する必要があります。

関連概念

36 ページの『指定レコード数の選択 (NBRRCDs パラメーター)』

FROMKEY パラメーターまたは FROMRCD パラメーターを指定するときは、TOKEY パラメーターまたは TORCD パラメーターの代わりに、コピーするレコード数 (NBRRCDs パラメーター) を指定することができます。

コピー操作で行われるキー・ストリングの比較:

TOKEY の値を指定した場合、コピー操作で行われる検査は、検索される各レコードのキー・ストリングの論理文字比較です。最初の TOKEY パラメーター様式を使用してキー・ストリングを明示的に指定できます。または、所定の値のリストを使用するコピー操作によってキー・ストリングが暗黙的に作成されます。

データベースがレコードをキー順アクセス・パスで識別する際の順序とは異なる順序が比較結果として検出されると、警告メッセージが出されます (ただし、コピー操作は続行されます)。次のような場合は、順序が異なることがあります。

- キーに昇順フィールドと降順フィールドが混在している
- *HEX 以外のソート・シーケンスが有効になっているフィールドがキーに含まれている
- 下記のいずれかの DDS キーワードがキーに含まれている

ABSVAL

絶対値

ALTSEQ

代替照合順序

ALWNULL

ヌル許可

DATFMT

日付形式 (*MDY、*DMY、*YMD、*JUL、SAA *EUR、または SAA *USA)

DIGIT 数字強制

SIGNED

符号付き数字

TIMFMT

時刻様式 (*USA)

ZONE ゾーン強制

昇順フィールドと降順フィールドの両方がファイル・キーに含まれている場合、コピー操作が最後のコピー対象レコードを探索するために昇順キー・テストと降順キー・テストのどちらを使用するかは、最初の (最も左側の) キー・フィールドによって決められます。

*BLDKEY の使用は、パック 10 進数、2 進数、および浮動小数点フィールドに対して値を指定する (および正しく埋め込ませる) 最も容易な方法です。

例: キー作成機能:

このトピックの例は、キー作成機能のさまざまなパラメーターの使用法を示しています。

キー・フィールド番号	タイプ	長さ	小数点以下の精度	値
1	CHAR	6		KEN
2	ZONED	6	2	54.25
3	BINARY	4	1	10.1

FROMKEY (または TOKEY) パラメーターを次のように指定できます。

```
FROMKEY( 2 x'D2C5D5404040F0F0F5F4F2F50065')
```

または、*BLDKEY 値を使用して、FROMKEY を次のように指定できます。

```
FROMKEY(*BLDKEY (KEN 54.25 10.1))
```

以下に、キー・フィールド 1 および 2 を使用する別の例を挙げます。

```
FROMKEY(2 'KEN 005425')
```

または、*BLDKEY 値を次のように指定できます。

```
FROMKEY(*BLDKEY (KEN 54.25))
```

例: FROMKEY と TOKEY の使用:

この例では、ファイル EMP1 中のレコードがファイル EMP1T にコピーされます。EMP1T はテスト・ライブラリー内のファイルです。レコードのサブセットが必要なだけなので、コピーを開始するキー値 (FROMKEY) とコピーを終了するキー値 (TOKEY) を指定します。どちらも完全キー値です。

FROMKEY パラメーターおよび TOKEY パラメーターで指定されている 1 は、最初のキー・フィールドから開始して、レコード・キーを探索するのに使用されるキー・フィールドの数を示すことに注意してください。

```
CPYF FROMFILE(PERSONNEL/EMP1) +
      TOFILE(TESTLIB1/EMP1T) MBROPT(*REPLACE) +
      FROMKEY(1 438872) TOKEY(1 810199)
```

キー値のすべての桁を指定する必要があります。その理由は、値がキー・フィールド長より短い場合は、右側にゼロが埋め込まれることになるからです。したがって、FROMKEY(1 8) として指定された 5 桁のキー・フィールドでは、16 進 F800000000 に等しいキーの探索が行われます。キー値がブランクまたは特殊文字を含んでいる場合は、アポストロフィで囲む必要があります。

レコード・キー (FROMKEY および TOKEY パラメーター) で使用される可変長フィールド:

キー・フィールドの数と 1 つの値を使用して FROMKEY または TOKEY パラメーターを指定するとき、そのストリングにはそれぞれ可変長キー・フィールド用に 2 バイトの長さのフィールドがなければなりません。

可変長キー・フィールドにはブランクを埋め込まなければなりません。これで、可変長キー・フィールドに続くキーの位置は正確になります。データは 16 進様式で指定できます。

可変長キー・フィールド用に FROMKEY または TOKEY パラメーターで *BLDKEY を指定したとき、2 バイトの長さのフィールドを含まない文字ストリングを指定してください。キー値用に入力されたデータの総計のみが、キーの比較用に使用されます。可変長キー・フィールドにはゼロ長のストリングを指定できません。

レコード・キー (FROMKEY および TOKEY) で使用される日付、時刻、およびタイム・スタンプ・フィールド:

キー・フィールドの数と 1 つの値を使用して FROMKEY または TOKEY パラメーターを指定するとき、コピー元ファイルの対応するキー・フィールドが日付、時刻、またはタイム・スタンプ・フィールドの場合、データは変換されません。

指定されたユーザー入カストリング (区切り記号を含む) は、日付、時刻、またはタイム・スタンプ・フィールドと同一様式でなければなりません。そうでない場合、ファイル・オープン・エラーが生じたり、レコードのコピー結果が望ましくないものになる場合があります。

*BLDKEY が FROMKEY または TOKEY パラメーターで指定され、コピー元ファイルの対応するキー・フィールドが日付、時刻、またはタイム・スタンプ・フィールドの場合、システムはユーザー入力キー・フィールド値をコピー元ファイル・フィールドの様式 (および区切り記号) に変換しようとします。以下の規則がその変換に適用されます。

- **コピー元ファイルのフィールドが日付キー・フィールドである場合**、システムは最初に、ユーザー入力キー値がコピー・コマンドを実行する現行ジョブに指定されたのと同じ区切り記号を持つかどうかを判別します。様式として *MDY、*DMY、*YMD、または *JUL、区切り記号としてスラッシュ (/)、ハイフン (-)、ピリオド (.), コンマ (,), またはブランク () があります。ユーザー入力キー値が現行ジョブ指定の様式および区切り記号様式でない場合、Systems Application Architecture[®] (SAA[®]) (システム・アプリケーション体系) 様式 (*ISO、*USA、*EUR、または *JIS) のいずれかであるかどうかを判別します。また、YYYYDDD 様式 (区切り記号なし) であるかどうかも判別します。ユーザー入力キー値がそれらの様式のいずれかであるとシステムが判別した場合、入カストリングはコピー元ファイル日付フィールドの現行様式 (および区切り記号) に変換され、キーの比較に使用されます。ユーザー入カストリング様式が判別されないか、または長さかデータ値が有効でない場合、診断メッセージが出されます。ユーザー入力キー値の日付の部分は左詰めにしなければなりません。後ろにはブランクを埋め込むことができます。
- **コピー元ファイルのフィールドが時刻キー・フィールドである場合**、システムは最初に、ユーザー入力キー値がコピー・コマンドを実行する現行ジョブに指定されたのと同じ区切り記号を持つかどうかを判別します。様式として HHMMSS、区切り記号としてコロン (:), コンマ (,), ピリオド (.), またはブランク () があります。ユーザー入力キー値が現行ジョブ指定の様式および区切り記号様式でない場合、システムは SAA 様式 (*ISO、*USA、*EUR、または *JIS) のいずれかであるかどうかを判別します。ユーザー入力キー値がそれらの様式のいずれかであるとシステムが判別した場合、入カストリングはコピー元ファイル時刻フィールドの現行様式 (および区切り記号) に変換され、キーの比較に使用されます。ユーザー入カストリング様式が判別されないか、または長さかデータ値が有効でない場合、診断メッセージが出されます。ユーザー入力キー値の時刻の部分は左詰めにしなければなりません。後ろにはブランクを埋め込むことができます。
- **コピー元ファイルのフィールドがタイム・スタンプ・キー・フィールドである場合**、システムは最初に、ユーザー入力キー値が SAA 様式または YYYYMMDDHHMMSS 様式かどうかを判別します。ユーザー入力キー値がそれらの様式のいずれかであるとシステムが判別した場合、入カストリングは現行 SAA タイム・スタンプ様式に変換され、キーの比較に使用されます。ユーザー入カストリング様式が判別されないか、または長さかデータ値が有効でない場合、診断メッセージが出されます。ユーザー入力キー値のタイム・スタンプの部分は左詰めにしなければなりません。後ろにはブランクを埋め込むことができます。

レコード・キー (FROMKEY および TOKEY パラメーター) で使用されるヌル可能フィールド:

キー・フィールドの数および 1 つの値を使用して FROMKEY または TOKEY パラメーターを指定するとき、ヌル値は無視されます。実際にヌルである値のためのバッファー・デフォルト値のみが、比較用に使用されます。

*BLDKEY が FROMKEY または TOKEY パラメーターで指定されたとき、いずれの *BLDKEY 値もヌル可能フィールドを参照できません。参照した場合、エラー・メッセージが送られます。

レコード・キー (FROMKEY および TOKEY) で使用される各種 CCSID:

キー・フィールドの数および 1 つの値を使用して FROMKEY または TOKEY パラメーターを指定するとき、入力ストリングの CCSID 変換は行われません。

文字フィールド、DBCS 混用フィールド、DBCS 択一フィールド、または DBCS 専用フィールドに対する FROMKEY または TOKEY パラメーターに *BLDKEY を指定したとき、指定された値はコピー・コマンドを実行しているプロセスの CCSID に含まれると見なされます。各キー値は、ジョブの CCSID からコピー元ファイル・キー・フィールドの CCSID に変換されます。変換表が定義されていないかまたは入力キー値の変換中にエラーが生じた場合、メッセージが送信されてコピー操作が終了します。値が正しく変換されると、変換値を使用してコピーされる最初と最後のレコードを判別するキー値を作成します。

レコード・キー (FROMKEY および TOKEY) で使用される DBCS グラフィック・フィールド:

キー・フィールドの数および 1 つの値を使用して FROMKEY または TOKEY パラメーターを指定するとき、入力ストリングの変換は行われません。入力ストリングはそのまま使用されます。

DBCS グラフィック・フィールドに対する FROMKEY または TOKEY パラメーターに *BLDKEY を指定する場合、DBCS データはシフトアウト文字およびシフトイン文字で囲む必要があります。DBCS データは、ジョブの CCSID の関連した DBCS CCSID であると見なされます。シフトアウト文字およびシフトイン文字は、キーを作成する前に削除されます。次のような場合、メッセージが送信されてコピー操作は終了します。

- 入力ストリングがシフトアウトおよびシフトイン (SO-SI) 文字で囲まれていないか、または
- データがコピー元ファイル・キー・フィールドの DBCS CCSID に変換できない

指定レコード数の選択 (NBRRCDS パラメーター):

FROMKEY パラメーターまたは FROMRCD パラメーターを指定するときは、TOKEY パラメーターまたは TORCD パラメーターの代わりに、コピーするレコード数 (NBRRCDS パラメーター) を指定することができます。

NBRRCDS パラメーターを TORCD パラメーターあるいは TOKEY パラメーターとともに指定することはできません。指定した開始キー値または開始レコード番号から始めて、指定したレコード数がコピーされます。

注: このパラメーターは、コマンド CPYF、CPYFRMDKT、CPYFRMQRYF、CPYFRMTAP、CPYTODKT、および CPYTOTAP で使用できます。

FROMKEY パラメーターまたは FROMRCD パラメーターを指定せずに、NBRRCDS パラメーターを指定することができます。この場合、ファイル内の最初のレコードからコピーされます。指定レコード数は、実際にコピー先ファイルにコピーされるレコード数であり、これには、

- 1 • COMPRESS(*NO) が指定されている場合はコピー元ファイルの中の削除済みレコードが含まれます。
- 1 • INCCHAR パラメーターおよび INCREL パラメーターによって除外されたレコードは含まれません。

以下の例は、コピー・コマンドを使用して、ファイル EMP1 にある 1000 個のレコードをファイル EMP1T にコピーする方法を示しています。レコードは EMP1 の最初のメンバーからコピーされて、EMP1T の最初のメンバーの中のレコードを置き換えます。

```
CPYF FROMFILE(PERSONNEL/EMP1) +  
  TOFILE(TESTLIB1/EMP1T) MBROPT(*REPLACE) +  
  NBRRCDS(1000)
```

また、NBRRCDS パラメーターを使用すれば、リスト上のレコードのサブセットを調べることもできます。

```
CPYF FROMFILE(PERSONNEL/EMP1) TOFILE(*PRINT) +  
  FROMRCD(250) NBRRCDS(10) OUTFMT(*HEX)
```

オープン Query ファイルのコピーが正しく行われた場合、そのファイル位置は予測不能です。同一ファイルを使用して、異なるプログラムまたは別の CPYFRMQRYF を実行したい場合は、ファイルを位置決めするかファイルをクローズして、同じ OPNQRYF コマンドを使用してファイルをオープンし直さなければなりません。データベース・ファイル位置決め (POSDBF) コマンドを使用すれば、ファイルを位置決めすることができます。場合によっては、高水準言語プログラムのステートメントを使用することができます。

関連概念

30 ページの『指定レコード様式名の使用によるレコードの選択 (RCDFMT パラメーター)』
RCDFMT パラメーターを使用して、特定のレコード様式のレコードをコピー対象として選択することができます。このパラメーターは CPYF コマンドでのみ使用できることに注意してください。

32 ページの『レコード・キーによるレコードの選択 (FROMKEY および TOKEY パラメーター)』
レコード・キーは、キー付きデータベース・ファイルからコピーする場合にのみ指定できます。このパラメーターは CPYF コマンドでのみ使用できることに注意してください。

文字内容に基づくレコードの選択 (INCCHAR パラメーター):

レコードやフィールドの特定の位置から始まる文字の内容に基づいて、レコードを選択できます。

注: このパラメーターは CPYF コマンドでのみ使用できます。

INCCHAR パラメーターを FROMKEY または FROMRCD パラメーターと共に使用できます。レコードの選択は、最初に位置または相対レコード番号によって、次にレコードまたはフィールドの特定位置にある文字によって行うことができます。

1 から 256 バイトからなる文字ストリングをテストすることができます。文字ストリングに特殊文字またはブランクが含まれている場合は、そのストリング全体をアポストロフィで囲む必要があります。

また、INCCHAR パラメーターの演算子として *CT (含む) を指定することもできます。これによって、コピー元ファイルの中の各レコードを走査して、選択文字ストリングを探索することを指定します。走査の開始に当たっては、フィールドまたはレコードの中の任意の有効な開始位置を指定することができます。データの走査は、その位置からフィールドまたはレコードの最右端のバイトまで行われることとなります。

INCCHAR パラメーターと INCREL パラメーターの両方を指定した場合は、レコードは INCCHAR および INCREL の両条件を満たす場合に限りコピーされます。

以下の例は、ファイル DBIN の中であって、80 桁目を開始位置とする XXX を持つすべてのレコードをテストする方法を示しています。次いで、それらのレコードをファイル DKTOUT にコピーします。この例には、レコード全体の長さに対する相対位置のテストを行なうので、*RCD を INCCHAR パラメーターで指定しなければなりません。

```
CPYF FROMFILE(DBIN) TOFILE(DKTOUT) +  
  INCCHAR(*RCD 80 *EQ XXX)
```

レコードの中のある特定のフィールドの中のある位置の XXX をテストする場合は、指定するのは *RCD ではなく、フィールド名とフィールドの始めを基準とした文字の開始位置となります。

```
CPYF FROMFILE(DBIN) TOFILE(DKTOUT) +
  INCCCHAR(FLDA 6 *EQ XXX)
```

複数様式論理ファイルからのコピー時に RCD FMT(*ALL) が指定されている場合、またはコピー元ファイルが装置ファイルまたはインライン・データ・ファイルである場合は、フィールド名を指定することはできません。

2 進文字フィールドの場合、INCCCHAR パラメーターは 2 進文字の比較規則を使用します。比較は CPYF によって実行され、埋め込みや切り捨ては実行されません。

INCCCHAR パラメーターで使用される可変長フィールド:

INCCCHAR パラメーターで *RCD が指定された場合、開始位置はバッファーでの位置を表します。可変長フィールドの 2 バイトの長さのフィールドは、位置を決めるときに考慮しなければなりません。

INCCCHAR 値が複数フィールドに及ぶ場合は、可変長フィールドに 1 バイト・ブランク (X'40') を埋め込んでください。

フィールド名を指定するときには、INCCCHAR スtringの可変長フィールドを指定できます。開始位置は、可変長コピー元フィールド値のデータ部分での位置を表します。比較されるバイト数は、INCCCHAR スtringに指定された値のバイト数です。可変長コピー元フィールドの現行データが INCCCHAR パラメーターで指定された値より短い場合、比較のために、コピー元フィールド・データに 1 バイト・ブランク (X'40') が埋め込まれます。

INCCCHAR 値にはゼロ長の Stringを指定できません。

INCCCHAR パラメーターで使用されるヌル可能フィールド:

INCCCHAR パラメーターに、ヌル可能文字フィールド名およびヌル可能 DBCS フィールド名を指定することができます。しかし、空フィールド値との論理比較は偽として認識され、レコードはコピーされません。

*RCD 特殊値をフィールド名として指定した場合、特殊な処理は一切行われません。実際にヌルである値のためのバッファー・デフォルト値のみが、比較されます。

INCCCHAR パラメーターで使用される各種 CCSID:

INCCCHAR パラメーターで *RCD が指定された場合、入力 Stringの変換は行われません。入力されたバイト・Stringは、コピー元ファイルのレコード・バッファーの指定位置で比較されます。

フィールド名が指定されたとき、入力 Stringはコピー・コマンドを実行しているジョブの CCSID であると見なされます。入力 Stringは、コピー元フィールドの CCSID に変換されます。変換表が定義されていないかまたは入力 Stringの変換中にエラーが生じた場合、メッセージが送信されてコピー操作が終了します。値を正しく変換できた場合は、変換された値がレコード選択に使用されます。

INCCCHAR パラメーターで使用される DBCS グラフィック・フィールド:

INCCCHAR パラメーターでグラフィック・フィールドを指定した場合、DBCS データはシフトアウト文字およびシフトイン文字で囲む必要があります。DBCS データは、ジョブの CCSID の関連した DBCS CCSID であると見なされます。

- | なお、そのフィールドの CCSID に有効な変換をしないとエラーが生じます。シフトアウト文字およびシフトイン文字は、比較を行なう前に除去されます。位置は、比較を開始する DBCS 文字の位置を示します。

フィールド値に基づくレコードの選択 (INCREL パラメーター):

INCREL パラメーターを使用して、フィールド全体の値をテストすることによってコピーするレコードを選択します。INCCHAR パラメーターの場合とは異なり、INCREL パラメーターを使用できるのはデータベース・ファイルからコピーするときに限られ、1つのコピー・コマンドで異なるフィールドの中の異なる値をテストすることができます。

注: このパラメーターは CPYF コマンドでのみ使用できます。

1つの INCREL パラメーターで 50 個までの AND および OR 関係を使用することができます。OR 関係は AND 関係をグループ化します。たとえば、次の INCREL パラメーターは次のようなことを意味しています。フィールド FLDA が 5 より大きく、フィールド FLDB が 6 より小さい場合はそのレコードを選択します。あるいは FLDB が 9 (FLDA は任意の値) の場合もそのレコードを選択します。

```
INCREL((*IF FLDA *GT 5) (*AND FLDB *LT 6) +
(*OR FLDB *EQ 9))
```

指定する値はフィールド・タイプと互換性がなければなりません。各 INCREL 関係セットは括弧で囲む必要があります。

比較値セットが 1 つだけまたは複数ある場合、最初の比較値セットの最初の値として値 *IF を指定しなければなりません。複数の比較値セットを指定する場合は、2 番目以降のそれぞれの比較値セットの最初の値として *AND または *OR のどちらかを指定しなければなりません。

以下の説明では、IF セットとオプションでその後続く 1 つまたは複数の AND セットを IF グループと呼び、OR セットとオプションでその後続く 1 つまたは複数の AND セットを OR グループと呼びます。完全グループ (後続の AND セットをもたない単一の IF セット、または OR セット) がすべて真の結果になるまで、各グループに指定されたすべての比較が行われます。少なくとも 1 つのグループが真の結果になれば、そのレコードはコピー先ファイルにコピーされます。

最初の比較値セット (*IF フィールド名演算子値) およびその IF セットに論理的に結合されている AND セット (ある場合) が、最初に評価されます。IF グループのすべてのセットの結果が真であれば、テストは終了し、レコードがコピーされます。IF グループのいずれかの結果が偽であり、しかも OR グループが続いていれば、別の比較が始まります。OR セットとその後の AND セット (ある場合) が評価されます (次の OR セットまで)。OR グループのすべての結果が真の場合、レコードがコピーされます。いずれかの結果が偽であり、しかも別の OR グループが続いていれば、いずれかの OR グループがすべて真になるか、OR グループがなくなるまで処理が続行されます。結果がすべて真である IF または OR グループがなければ、レコードは除外されます (コピー先ファイルにコピーされません)。

INCCHAR パラメーターと INCREL パラメーターの両方を指定した場合は、レコードは INCCHAR および INCREL の両条件を満たす場合に限りコピーされます。

複数様式論理ファイルからのコピー時に RCD_FMT(*ALL) が指定されている場合は、INCREL パラメーターを指定することはできません。

2 進文字フィールドの場合、INCREL パラメーターは *EQ と *NE のチェックだけを行なうことができます。

INCREL パラメーターで使用される可変長フィールド:

INCREL パラメーターには可変長文字フィールドを使用することができます。2 バイトの長さのフィールドなしで文字を入力してください。入力されたデータの長さによって比較に使用されるバイト数が決定されます。

可変長コピー元フィールドの実際のデータが INCREL パラメーターで指定された値より短い場合、コピー元フィールド・データには比較のために 1 バイト・ブランク (X'40') が埋め込まれます。

INCREL パラメーターで使用される日付、時刻、およびタイム・スタンプ・フィールド:

INCREL パラメーターで、日付、時刻、およびタイム・スタンプ・フィールドを使用することができます。レコードを選択するかどうかを決定するために、入力フィールド値は時間として、日付、時刻、およびタイム・スタンプ・フィールドの値と比較されます。

システムは、入力ストリングと実際フィールド値を時間として比較するために内部様式に変換しようとしています。以下の規則がその変換に適用されます。

- **コピー元ファイルのフィールドが日付フィールドである場合**、システムはユーザー入力フィールド値が、コピー・コマンドを実行する現行ジョブに指定されたのと同様式であり、同じ区切り記号を持つかどうかを判別します。様式として *MDY、*DMY、*YMD、または *JUL、区切り記号としてスラッシュ (/)、ハイフン (-)、ピリオド (.)、コンマ (,)、またはブランク () があります。ユーザー入力フィールド値が現行ジョブの様式および区切り記号を使用しない場合、SAA 様式 (*ISO、*USA、*EUR、または *JIS) のいずれかであるかどうか、あるいは区切り記号なしの YYYYDDDD 様式であるかどうかを判別します。ユーザー入力フィールド値がそれらの様式のいずれかであるとシステムが判別した場合、入力ストリングは内部様式に変換されます。コピー元フィールドがその内部様式に変換され、比較作業が行われます。ユーザー入力ストリング様式が判別されないか、または長さかデータ値が有効でない場合、診断メッセージが出されてコピー操作は終了します。ユーザー入力フィールド値の日付の部分は左詰めにしなければなりません。後ろにはブランクを埋め込むことができます。
- **コピー元ファイルのフィールドが時刻フィールドである場合**、システムはユーザー入力フィールド値がコピー・コマンドを実行する現行ジョブに指定されたのと同様式であり、同じ区切り記号を持つかどうかを判別します。様式として HHMMSS、区切り記号としてコロン (:)、コンマ (,)、ピリオド (.)、またはブランク () があります。ユーザー入力フィールド値が現行ジョブ指定の様式および区切り記号様式でない場合、システムは SAA 様式 (*ISO、*USA、*EUR、または *JIS) のいずれかであるかどうかを判別します。ユーザー入力キー値がそれらの様式のいずれかであるとシステムが判別した場合、入力ストリングは内部様式に変換されます。コピー元フィールドがその内部様式に変換され、比較作業が行われます。ユーザー入力ストリング様式が判別されないか、または長さかデータ値が有効でない場合、診断メッセージが出されてコピー操作は終了します。ユーザー入力フィールド値の時刻の部分は左詰めにしなければなりません。後ろにはブランクを埋め込むことができます。
- **コピー元ファイルのフィールドがタイム・スタンプ・フィールドである場合**、システムは最初に、ユーザー入力フィールド値が SAA 様式または YYYYMMDDHHMMSS 様式 (区切り記号なし) かどうかを判別します。ユーザー入力フィールド値がそれらの様式のいずれかであるとシステムが判別した場合、入力ストリングは内部様式に変換されます。コピー元フィールドがその内部様式に変換され、比較作業が行われます。ユーザー入力ストリング様式が判別されないか、または長さかデータ値が有効でない場合、診断メッセージが出されてコピー操作は終了します。ユーザー入力フィールド値のタイム・スタンプの部分は左詰めにしなければなりません。後ろにはブランクを埋め込むことができます。

INCREL パラメーターで使用されるヌル可能フィールド:

INCREL パラメーターで、フィールド値として *NULL の値を入力できます。*EQ および *NE 演算子 *NULL 値とともに使用して、データベース・ファイルのフィールドにヌル値があるかどうかを確認することができます。

*NULL 値を指定するとき、*EQ は値がヌルであることを、*NE は値がヌルでないことを意味します。

*NULL 値は、ヌル可能フィールドに限定されません。

INCREL パラメーターで使用される各種 CCSID:

文字フィールド、DBCS 混用フィールド、DBCS 択一フィールド、または DBCS 専用フィールドに対する入カストリングは、コピー・コマンドを実行しているジョブの CCSID に含まれると見なされます。入カストリングは、コピー元フィールドの CCSID に変換されます。

変換表が定義されていないかまたは入カストリングの変換中にエラーが生じた場合、メッセージが送信されてコピー操作が終了します。値が正しく変換されると、変換された値がレコード選択に使用されます。

INCREL パラメーターで使用される DBCS グラフィック・フィールド:

INCREL パラメーターでグラフィック・フィールドを指定する場合、DBCS データはシフトアウト文字およびシフトイン文字で囲む必要があります。DBCS データは、ジョブの CCSID の関連した DBCS CCSID であると見なされます。

なお、そのフィールドの CCSID に有効な変換をする必要があります。そうでない場合、マッピング・エラーが生じます。シフトアウト文字およびシフトイン文字は、比較を行なう前に除去されます。

削除済みレコードのコピー (COMPRESS パラメーター):

コピー元ファイルからコピーされるレコードの相対レコード番号を保存するために、削除済みレコードをコピーしたい場合があります。コピー・コマンドで COMPRESS(*NO) パラメーターを指定することによって、削除済みレコードと未削除レコードの両方を物理ファイル・メンバー間でコピーすることができます。

COMPRESS(*NO) を使用しないと、まだ削除されていないレコードだけが、コピー元ファイルからコピーされます。

注: このパラメーターは CPYF コマンドでのみ使用できます。

関連概念

25 ページの『ファイルをコピーする際の *ADD の指定』

*ADD を指定すると、各コピー対象レコードはメンバーの中の既存のレコードの終わりに追加されません。

30 ページの『指定レコード様式名の使用によるレコードの選択 (RCDFMT パラメーター)』

RCDFMT パラメーターを使用して、特定のレコード様式のレコードをコピー対象として選択することができます。このパラメーターは CPYF コマンドでのみ使用できることに注意してください。

COMPRESS(*NO) パラメーターと CPYF コマンドの要件:

COMPRESS(*NO) を使用するには、このトピックで説明する条件が満たされていなければなりません。

- コピー元ファイルとコピー先ファイルの両方が物理ファイルである。
- コピー元ファイルとコピー先ファイルが同じタイプ (ソースまたはデータ) である。
- コピーを実行するには、コピー元ファイルとコピー元ファイルのレコード様式が同一であるか、FMTOPT(*NOCHK) を指定する必要があります。
- 次の (デフォルト) パラメーター値をすべてコピー・コマンドで使用する必要があります。
 - PRINT(*NONE)
 - INCCHAR(*NONE)
 - INCREL(*NONE)
 - SRCOPT(*SAME)
 - ERRVLV(0)

COMPRESS(*NO) パラメーターと CPYF コマンドの制限:

このトピックで説明されている場合では、COMPRESS(*NO) を指定することはできません。

COMPRESS(*NO) は、コピー先ファイルに対する以下のタイプのアクセス・パスの場合は、アクセス・パスが論理ファイルに含まれ、コピー先ファイルに基づいているときを含めて、使用することができません。

- 固有キー (DDS で UNIQUE キーワードを指定した場合)。
- DYNSLT キーワード (ファイルに関する DDS の中) の指定を伴わない選択指定または除外指定、および即時メンテナンスまたは遅延メンテナンス (CRTPF コマンドまたは CRTLF コマンドで指定される MAINT(*IMMED) または MAINT(*DLY))。
- 浮動小数点キー・フィールドまたは論理数字キー・フィールド (ファイルに関する DDS の中)、および即時メンテナンスまたは遅延メンテナンス (CRTPF コマンドまたは CRTLF コマンドで指定される MAINT(*IMMED) または MAINT(*DLY))。論理数字キー・フィールドは以下のフィールドのいずれかであることを注意してください。
 - 論理ファイル内の数字キー・フィールド
 - JFLD キーワードで、基礎となる物理ファイルの中とは異なる属性を持つ TO フィールドとして指定されたフィールド
 - JDUPSEQ キーワードで、基礎となる物理ファイルの中とは異なる属性を持つ順序付けフィールドとして指定されたフィールド

以下のいずれかの場合には COMPRESS(*NO) を指定することはできません。

- JRNPF コマンドを使用してコピー先ファイルをジャーナルした場合
- コピー先ファイル・メンバーが使用中またはコピー先ファイル・メンバーにかかるアクセス・パスが使用中の場合
- EOFDLY 待機時間がコピー元ファイルに対して OVRDBF コマンドで指定されている場合

COMPRESS(*NO) パラメーターと CPYF コマンドの詳細:

COMPRESS(*NO) では、レコードがブロック単位で転送されるので、システムがコピーを迅速に実行することが可能になりますが、必ずしも常に そうとは限りません。普通は、COMPRESS(*NO) 機能でパフォーマンスに著しい影響が及ぶことはありません。

- | COMPRESS(*NO) パラメーターを指定する前に、COMPRESS(*NO) パラメーターの使用によってシステム
- | が以下のアクションを実行することに注意してください。
- | • レコードがコピーされる前に、システムはコピー先ファイル・メンバーを使用するすべてのキー順アク
- | セス・パスを無効化します。
- | • コピー操作が完了した後、システムはアクセス・パスを再作成します。

キー順アクセス・パスの再作成に必要な実行時間およびリソースの方が、削除済みレコードをコピーすることによって得られるパフォーマンス上の利点を上回る場合もあります。

COMPRESS(*NO) パラメーターを指定しなくても、システムが内部機能を使用してコピーを実行することもあります。コピーを実行する方法の選択は、コピー前のコピー元ファイル・メンバーおよびコピー先ファイル・メンバーの中のレコード数、およびコピー先ファイル・メンバーに対するキー順アクセス・パスの数に基づきます。

MBROPT(*REPLACE) を指定した場合、コピー先メンバーに対するキー順アクセス・パスは、すべて無効にして再作成しなければなりませんから、COMPRESS(*NO) の指定によってアクセス・パスの再作成に要する追加オーバーヘッドが生じることはありません。

コピー元ファイルがキー付き物理ファイルで、ファイルを到着順に処理させるための、FROMRCD 相対レコード番号値も TORCD 相対レコード番号値もコピー・コマンドで指定されていない場合、キー順アクセス・パスに削除済みレコードが含まれることはありませんから、COMPRESS(*NO) に意味はなくなります。

レコードの印刷 (PRINT、OUTFMT、および TOFILE(*PRINT) パラメーター)

コピーされたすべてのレコード、除外されたすべてのレコード、または ERRLVL 出力エラーが生じたすべてのレコードを印刷することができます。これを行うには、コピー・コマンド上に PRINT 特殊値を指定します。

注: このトピックで説明するパラメーターは、CPYF、CPYFRMQRYF、および CPYFRMTAP コマンドで使用できます。

文字様式または 16 進様式を使用して、単一のコピー・コマンド上で 1 つまたは複数のリストを指定することができます。レコードの不定様式リストを印刷することもできます。

コピーされたすべてのレコードのリストの印刷:

コピーしたすべてのレコードのリストを印刷するには、コピー・コマンドで TOFILE(*PRINT) を指定してください。レコードの印刷は、IBM 提供のプリンター・ファイル QSYSPRT を使用して行われます。

除外されたレコードのリストの印刷:

PRINT パラメーターで *EXCLD を指定して、コピーから除外したレコードのみのリストを印刷してください。PRINT(*EXCLD) を指定すると、レコードは、コピー元ファイルの様式で印刷されます。

コピーされたレコードのリストの印刷:

PRINT パラメーターで *COPIED を指定すると、コピーしたレコードのみのリストが印刷されます。

PRINT(*COPIED) と MBROPT(*UPDADD) との両方が指定されている場合、コピーされたレコードおよび更新されたレコードはすべて同一のリストに印刷されます。更新された各レコードの後には、更新されたレコードであることを示すメッセージが続きます。

エラーを引き起こすレコードのリストの印刷:

PRINT パラメーターで *ERROR を指定すると、ERRLVL 出力エラーを引き起こしたレコードのリストが印刷されます。(この場合も、ERRLVL パラメーターが、起こりうる回復可能エラーの数を制御します。) *ERROR リストに印刷されるレコードの最大数は、指定された ERRLVL 値より 1 だけ大きい数です。そのリストは、PRINT(*COPIED) および PRINT(*EXCLD) のリストと類似しています。

リスト様式の選択:

OUTFMT パラメーターを使用して、リストを文字様式または 16 進様式のいずれかで印刷するかを指定します。デフォルト値は *CHAR で、レコードは文字様式で印刷されます。*HEX を指定した場合、レコードは文字様式および 16 進数様式の両方で印刷されます。

TOFILE(*PRINT) を指定した場合は、OUTFMT パラメーターでレコードの印刷に使用される様式を再度指定します。

PRINT(*EXCLD) を指定する場合、レコードは、コピー元ファイルの様式で印刷されます。すべての文字データは、コピー元ファイル・フィールドで指定された CCSID の状態です。TOFILE(*PRINT) および PRINT(*COPIED) のリスト出力でコピー先ファイルが印刷ファイルであるとき、文字データはコピー先フ

ファイル・フィールドで指定される CCSID の状態です。

例:

この例では、コピーされなかったすべてのレコード (つまり除外されたレコード) が印刷されます。

```
CPYF FROMFILE(DKTIN) TOFILE(LIB1/PF) +
      MBROPT(*ADD) INCCHAR(*RCD 80 *EQ X) +
      PRINT(*EXCLD)
```

レコードは、文字様式で印刷されます。

関連概念

『不定様式印刷リストの作成』

不定様式印刷リストが必要な場合、またはコピー元ファイル・レコードが先頭文字用紙制御 (CTLCHAR(*FCFC)) を使用して様式化される場合には、プログラム記述のプリンター・ファイル名を指定しなければなりません。このファイル名には、QSYSPRT またはユーザー定義値を (*PRINT の代わりに) 指定することができます。

135 ページの『ファイルのコピー時におけるエラーの防止』

コピーの前に特定の条件や状況を計画するとき、多くのコピー・エラーを防止することができます。

不定様式印刷リストの作成:

不定様式印刷リストが必要な場合、またはコピー元ファイル・レコードが先頭文字用紙制御 (CTLCHAR(*FCFC)) を使用して様式化される場合には、プログラム記述のプリンター・ファイル名を指定しなければなりません。このファイル名には、QSYSPRT またはユーザー定義値を (*PRINT の代わりに) 指定することができます。

先頭文字用紙制御を使用してコピー元ファイル・レコードを様式化するには、プリンター・ファイル作成 (CRTPRTF)、プリンター・ファイル変更 (CHGPRTF)、またはプリンター・ファイル・オーバーライド (OVRPRTF) コマンドで CTLCHAR(*FCFC) を指定してください。

コピー・コマンドで TOFILE(*PRINT) を指定し、しかも PRINT パラメーター値が *COPIED、*EXCLD、または *ERROR (あるいはそれらの組み合わせ) である場合は、以下の制約事項が適用されます。

- QSYSPRT ファイルは、スプールされなければなりません [SPOOL(*YES)]。
- 要求されたファイルごとに別々の印刷ファイルがオープンされるため、装置ファイル内または OVRPRTF コマンドに QSYSPRT を指定しなければなりません。

すべてのレコードが単一のスプール・ファイルにコピーされ、コピーされるおのこのメンバーまたはラベル識別コードごとに改ページしてデータが新しい印刷ページから印刷されます。

関連概念

43 ページの『レコードの印刷 (PRINT、OUTFMT、および TOFILE(*PRINT) パラメーター)』

コピーされたすべてのレコード、除外されたすべてのレコード、または ERRLVL 出力エラーが生じたすべてのレコードを印刷することができます。これを行うには、コピー・コマンド上に PRINT 特殊値を指定します。

コピー対象メンバーの選択

iSeries には、ファイル・メンバーをコピーするためのオプションがいくつかあります。

関連概念

20 ページの『物理ファイルまたは論理ファイルのコピー』

iSeries 上の物理ファイルまたは論理ファイル (コピー元ファイル) を、まだ存在していない別の物理ファイル (コピー先ファイル) にコピーするには、CPYF コマンドを使用することができます。

ファイル・メンバーのコピー: 概要:

複数のデータベース・メンバーまたはディスク・ラベルは、対応する同名のコピー先ファイル・メンバーまたはラベルにコピーすることができます。また、1 つずつコピーおよび連結して、単一のコピー先ファイル・メンバーまたはラベルにすることもできます。

コピー先ファイルがスプール・ファイルの場合、コピー・コマンドは、各メンバーやラベルをそれぞれ別々のスプール・ファイルにコピーします。TOFILE(*PRINT) が指定されている場合、すべてのメンバー/ラベルは単一のスプール・ファイルにコピーされ、各メンバー/ラベルごとにレコードが新しいページから始まります。

単一のメンバーやラベルまたは複数のメンバーやラベルは、使用されるコピー・コマンドに応じて TOMBR(*FROMMBR)、TOLABEL(*FROMMBR)、または TOMBR(*FROMLABEL) を指定することによって、対応する同名のコピー先ファイル・メンバーまたはラベルにコピーすることができます。コピー先ファイルがテープである場合は、単一のコピー元ファイル・メンバーまたはラベルからコピーするのでない限り、この指定を行うことはできません。*FROMMBR は CPYSRCF コマンドの TOMBR パラメーターのデフォルト値であり、この場合、コピー元ファイルのメンバーはコピー先ファイルの同じ名前のメンバーにコピーされます。

許可されるコピー操作とパラメーター:

特定のパラメーターを使用して、特定の条件のもとで特定のメンバーを選択してコピーすることができます。

以下の表は、ソース・ファイル・タイプに基づいてメンバーやラベルをコピーできるファイル・タイプを示しています。

表4. ソース・ファイル・タイプ別の、メンバーやラベルのコピー先として可能なファイル・タイプ

ディスクのコピー先:	データベースのコピー先:
データベース (物理ファイル)	データベース (物理ファイル)
ディスク (注 1)	ディスク
テープ (注 2)	テープ (注 2)
プリンター	プリンター
*PRINT	*PRINT
注:	
1. ディスク間コピー操作では、コピー先ファイルは、スプールされていなければなりません。	
2. 複数のコピー元ファイル・メンバーまたはラベルをコピーできる先は、単一のテープ・ファイル・ラベルに限られます。	

以下の表は、コピー・コマンドで有効なメンバーまたはラベル・パラメーターを示しています。

表5. コピー・コマンドで有効なメンバーまたはラベル・パラメーター

	FROMMBR ¹	FROMLABEL	TOMBR	TOLABEL
CPYF	X		X	
CPYFRMDKT		X	X	
CPYFRMQRYF			X	
CPYFRMTAP		X	X	

表 5. コピー・コマンドで有効なメンバーまたはラベル・パラメーター (続き)

	FROMMBR ¹	FROMLABEL	TOMBR	TOLABEL
CPYSRCF	X		X	
CPYTODKT	X			X
CPYTOTAP	X			
CPYFRMIMPF	X		X	
CPYTOIMPF	X		X	
注:				
¹	FROMMBR は CPYFRMQRYF コマンドのパラメーターではありません。照会されるメンバーは OPNQRYF コマンドで指定されているからです。			

1 ファイル内のすべてのメンバーのコピー:

- 1 データベース・ファイルの場合、すべてのメンバーをコピーするには FROMMBR または FROMLABEL
- 1 パラメーターで *ALL を指定します。

1 ファイル内の特定メンバーだけのコピー:

- 1 データベース・ファイルの場合、最初に FROMMBR または FROMLABEL パラメーターで総称名を指定
- 1 します。次に、総称名を変更して、各メンバーまたはラベルに共通している開始文字ストリングに * (アス
- 1 タリスク) が続いたものにします。

- 1 たとえば、FROMMBR(ORD*) と指定した場合、コピー・コマンドは、ORD で始まるデータベース・メン
- 1 バーすべてをコピーします。

- 1 注: ディスケットから総称セットをコピーする場合に、コピー中の ラベルが他のディスク・ボリュー
- 1 ムに続いていると、コピー・コマンドは継続ボリューム上にある関係するラベルをすべてコピーしま
- 1 す。これは、すべてのラベルをコピーする場合も同様です。

コピー操作のメンバー名の指定:

コピー操作のさまざまなメンバー名を指定することができます。

- 1 TOMBR (*FIRST) を指定すると、コピー操作はラベル ID を指定しません。したがって、OVRTAPF コマ
- 1 ンドでラベル ID (LABEL パラメーター) を指定する必要があります。コピー・コマンドで特殊値 *FIRST
- 1 または *TAPF を指定した場合、コピー・コマンドは装置ファイル記述にあるラベルを使用します。

- 1 コピー元ファイルがテープの場合、コピー・コマンドはコピー先テープ・ファイルのラベルとしてコピー元
- 1 ファイル・ラベルを使用します。コピー先ファイルがデータベース・ファイルの場合は、コピー先ファイル
- 1 のメンバー名として、コピー元ファイル・ラベルの右端のブランク以外の文字が使用されます。最大 10 文
- 1 字までまたはコピー元ファイル・ラベルの右端のピリオドまでの文字かのどちらかが使用されます。コピー
- 1 操作では、データベースのコピー先ファイルとして有効なメンバー名だけが使用されます。

コピー元ファイルがラベル付きではないテープ・ファイルの場合は、コピー元テープ・ファイル上のデー

タ・ファイルに対応するコピー先ファイル・メンバーまたはラベル名が CPYnnnnn の様式で作成されま

す。なお、nnnnn はデータ・ファイルのテープ順序番号です。

テープを FROMMBR パラメーターまたは TOMBR パラメーターで指定する場合は、最大 10 文字の長さ

にすることができます。ラベルに特殊文字または 10 文字を超える文字が含まれる場合は、そのラベルを次

のコマンドの 1 つで指定しなければなりません。

- 1 • テープ装置ファイル作成 (CRTTAPF)

- | • テープ装置ファイル変更 (CHGTAPF)
- | • テープ・ファイル・オーバーライド (OVRTAPF)
- | データベース・ファイル・オーバーライド (OVRDBF) およびテープ・ファイル・オーバーライド (OVRTAPF) コマンドでの特殊な考慮事項:
- | OVRDBF および OBRTAPF コマンドで特定のパラメーターを指定した場合の結果に注意する必要があります。
- | コピー元またはコピー先データベース・ファイルの場合は、 MBR パラメーターが OVRDBF (データベース・ファイル・オーバーライド) コマンドで指定されると、オーバーライド・メンバー名がコピー・コマンドで指定した値の代わりに使用されます。 OVRDBF コマンドで MBR パラメーター値を指定せずに TOFILE パラメーターを指定した場合は、データベース・ファイルの中の最初のメンバー (作成順) が、コピー・コマンドで指定したメンバーの代わりに使用されます。コピー元またはコピー先テープ・ファイルの場合は、 LABEL パラメーターがコピー先ファイルの OVRTAPF コマンドで指定されると、コピー・コマンドで指定したラベルの代わりにオーバーライド・ラベル名が使用されます。
- | 複数のメンバーまたはラベルを名前の似た対応するコピー先ファイルまたはラベルにコピーする場合、単一のメンバーまたはラベルに対してコピー元ファイルもオーバーライドするのではないが、単一のコピー先メンバーまたはラベルに対してオーバーライドを使用することはできません。

コピー機能がメンバーをコピー先ファイルに追加する方法:

コピー機能では、メンバーが存在しないときは、そのメンバーをコピー先ファイルに追加します。使用されるメンバー名は、コピー・コマンドの TOMBR パラメーター値またはコピー先ファイルに関するオーバーライドで指定されるメンバー名かどちらかです。

TOMBR(*FROMMBR) または TOMBR(*FROMLABEL) がコピー・コマンドで指定されている (しかも、オーバーライドされていない) 場合は、コピー元ファイル・メンバー名またはラベル識別コードが、ファイルに追加されるメンバーに使用されます。

TOMBR(*FIRST) がコピー・コマンドで指定されている場合、または MBR パラメーターなしで TOFILE パラメーターを指定するオーバーライドがある場合は、メンバー名はわかりません。この場合は、以下の条件が満たされていない限り、コピー機能ではメンバーを追加しません。

- CRTFILE(*YES) をコピー・コマンドで指定した
- コピー機能でコピー先ファイルを作成しなければならない

CPYFRMQRYP コマンド以外は、特定のメンバー名を指定せずにコピー機能でコピー先ファイルを作成する場合、そのコピー元ファイル名が、そのファイルに追加されるメンバーに使用されます。

CPYFRMQRYP コマンドの場合は、コピー操作で作成された物理ファイルに追加されるメンバーには、TOMBR パラメーターで指定した名前が与えられます。 TOMBR(*FIRST) を指定すると、コピー先ファイルのメンバーの名前は、 CPYFRMQRYP コマンドの TOFILE パラメーターで指定したコピー先ファイル名と同じになります。コピー先ファイルを作成したり、レコードを新規ファイル・メンバーに追加したりするとき、コピー・コマンドは、MBROPT パラメーター値を無視します。

コピー元ファイルがデータベース・ファイルの場合、コピー元ファイル・メンバーのメンバー・テキストおよび SEU ソース・タイプが、コピー先ファイルに追加されるメンバーに使用されます。コピー元ファイルが装置ファイルまたはインライン・データ・ファイルの場合は、テキストはメッセージ CPX0411 から取られ、 SEU ソース・タイプは TXT になります。コピー元ファイルとコピー先ファイルが共にデータベース・ソース・ファイルの場合は、追加されるメンバーの SEU ソース・タイプ情報は、コピー元ファイルの

対象メンバーと同じになります。コピー先ファイル・メンバーが追加されると、SHARE(* NO) および EXPDATE(*NONE) 属性が常にコピー先ファイル・メンバーに割り当てられます。また、新規メンバーの作成日付に現在のシステム日付 (コピー元ファイル・メンバーの追加日付ではない) を設定されます。

コピーによって親ファイルであるコピー先ファイルにメンバーが追加された場合、その時点で制約が確立されます。

異なるデータベース・レコード様式間でのコピー (FMTOPT パラメーター)

データベース・ファイルからデータベース・ファイルへコピーする場合、レコード様式が同じではない場合またはファイルがタイプを異にする (ソースまたはデータ) ときには、FMTOPT パラメーターを使用しなければなりません。どちらかのファイルが装置ファイルまたはインライン・データ・ファイルである場合は、FMTOPT パラメーターは適用されません。レコード長が異なる場合は、レコードの切り捨てもしくははブランクまたはゼロによる埋め込みが行われます。レコードが切り捨てられる場合は、メッセージが送信されます。

注: このパラメーターは、CPYF および CPYFRMQRYP コマンドで使用できます。

データベース・ファイル相互間のコピーで、FMTOPT(*CVTSRC) または FMTOPT(*NOCHK) を指定した場合、いずれかのコピー元ファイルのレコードからコピーされたレコード・データの長さがコピー先ファイルのレコードの長さに満たないときは、コピー先ファイルのレコードの余分なバイトはデフォルト値に設定されます。*NULL 以外のデフォルト値がフィールドに対する DDS (DFT キーワード) で指定されていれば、そのフィールドは指定のデフォルト値に初期設定されます。これが指定されていない場合は、すべての数値フィールドはゼロに、すべての文字フィールドはヌルに、すべての日付、時刻、タイム・スタンプ・フィールドは現在のサーバーの日付、および時刻に初期設定されます。*NULL が DFT キーワードで指定されている場合、デフォルトのバッファ値のみが使用されます。*NULL デフォルトは無視されます。

コピー元ファイルまたはコピー先ファイルが装置ファイルか、またはインライン・データ・ファイルの場合は、コピーされる各レコードのソース順序番号フィールドおよび日付フィールドは、コピーによって自動的に追加または削除されます。

一方のファイルがデータ・ファイルで、他方のファイルがソース・ファイルの場合は、コピーを実行するには FMTOPT(*CVTSRC) を指定しなければなりません。順序番号フィールドおよび日付フィールドは適宜追加または削除され、各レコードのデータ部はファイル・レコード様式の中の他のフィールド定義に関係なくコピーされます。SRCSEQ パラメーターを使用し、SRCOPT(*SEQNBR) も指定されていれば、順序番号の作成方法を制御することができます。

FMTOPT *MAP を使用すると、2 進文字フィールドはゼロで埋め込まれます。

データベース間コピーの場合は、次の指定を行うことによってレコード様式に違いがあっても一致させることができます。

- *DROP を指定して、コピー元ファイル・レコード様式の中のフィールドのうち、コピー先ファイル・レコード様式の中に同じ名前のフィールドがないものを削除します。
- *MAP を指定して、コピー元ファイルの中のフィールドをコピー先ファイルの中の同名のフィールドの属性に変換し、コピー元ファイルの中にはないコピー先ファイルの中の余分のフィールドにデフォルト値を充てんします。デフォルト値には次のものがあります。
 - フィールドに対して DFT キーワードが指定されている場合には、そのパラメーター値 (*NULL を含む)。
 - ブランク (DFT キーワードのない文字フィールドの場合)。
 - ゼロ (DFT キーワードのない数字フィールドの場合)。

- DFT キーワードを使用しないフィールドの現在の日付、時刻、タイム・スタンプ。

同じ名前を持つフィールドが、ファイル・レコード様式の中で異なる位置にある場合は、たとえこれらのフィールドが同じ属性を備えていても、*MAP は必須になります。

- *DROP および *MAP を指定して、コピー先ファイルの中では名前を付けられていないコピー元ファイルの中のフィールドを排除し、残りのフィールドをマッピング規則によって変換して、異なる属性および桁数を持つコピー先ファイルに適合させます。
- *NOCHK を指定して、違いを無視します。データは左から右へ、一方のファイルからもう一方のファイルにそのままコピーされます。このとき、ヌル値は無視されます。コピーされたレコードは、切り捨てかデフォルトのバッファ値による埋め込みかどちらかを施されます。検査が行われないので、コピー先ファイルの中のフィールドには、そのフィールドの定義に対して有効でないデータが入っていることもあり得ます。

フィールドの排除およびマッピングは、フィールド名の比較に基づいています。コピー元ファイルの中のすべてのフィールドと同じ名前がコピー先ファイルの中にあるのであれば、*DROP を指定しなければなりません。名前は同じでも属性またはレコードの中の位置が異なる場合は、*MAP を指定しなければなりません。排除されたフィールドはコピーされません。マッピングを行うには、両方のレコード様式に少なくとも 1 つの同名フィールドがなければなりません。

*MAP が指定されると、コピー元ファイル・レコード様式の中には存在しないコピー先ファイル・レコード様式の中のフィールドに、この節で前述したようにデフォルト値が充てんされます。同じ名前および属性を持つフィールドの場合は、たとえ様式の中の位置が異なっている場合でも、コピー元ファイル・レコード様式の中のフィールドが、コピー先ファイル・レコード様式の中で同じ名前を持つフィールドにマップされます。

たとえば、フィールド CUSNO はレコード様式 ORDHD の中では最初のフィールドですが、レコード様式 ORDHD1 の中では 2 番目のフィールドです。CUSNO フィールドは *MAP を指定してコピーされると、ORDHD1 の 2 番目のフィールドにマップされます。

注: 大規模レコード様式 (多くのフィールド) を持つファイルは、様式が多少異なっても、同じ様式のレベル識別コードを持つ場合があります。コピー元ファイルとコピー先ファイルのレコード様式名が同一の場合、これらのファイルをコピーしている時に問題が発生する可能性があります。

FMTOPT(*NONE) または FMTOPT(*MAP) を使用してそのようなファイルをコピーしているときは、コピー元ファイルとコピー先ファイルのレコード様式名を異なるものにするをお勧めします。

表 6 は、FMTOPT パラメーターの各値ごとのデータベース間コピー操作の要約です。

表 6. データベース間のコピー操作

FMTOPT パラメー ター値 (注 4 を 参照)	データベース・ファイルのレコード様式				
	コピー元ファイルとコピー先ファイルの全部のフィールド名が同じ (同名)	コピー元ファイルとコピー先ファイルの一部のフィールド名が同じ	コピー元ファイルとコピー先ファイルで同じフィールド名なし		
	属性および相対順序も同じ (注 1 を参照)	属性および相対順序は不同 (注 1 を参照)	同名のフィールドは属性および相対順序も同じ (注 1 を参照)	同名のフィールドにも属性および相対順序が異なるものあり (注 1 を参照)	
*NONE	完全コピー	コマンド終了	コマンド終了	コマンド終了	コマンド終了

表 6. データベース間のコピー操作 (続き)

FMTOPT パラメータ値 (注 4 を参照)	データベース・ファイルのレコード様式				
	コピー元ファイルとコピー先ファイルの全部のフィールド名が同じ (同名)	コピー元ファイルとコピー先ファイルの一部のフィールド名が同じ	コピー元ファイルとコピー先ファイルで同じフィールド名なし		
*DROP	完全コピー (値は無視)	コマンド終了	コピー元ファイルに余分のフィールドがあれば、それだけを排除して、それ以外はコピーされる。コピー先ファイルに余分のフィールドがあれば、コマンドは打ち切られる。コピー元ファイルおよびコピー先ファイルに余分のフィールドがあれば、コマンドは打ち切られる。	コマンド終了	コマンド終了
*MAP (注 2 を参照)	完全コピー (値は無視)	完全コピー (対応するフィールドはマップ)	コピー元ファイルに余分のフィールドがあれば、コマンドは打ち切られる。コピー先ファイルに余分のフィールドがあれば、それは充てんされ、同名のフィールドはマップされる。コピー元ファイルおよびコピー先ファイルに余分のフィールドがあれば、コマンドは打ち切られる。		コマンド終了
*MAP および *DROP (注 2 を参照)	完全コピー (値は無視)	完全コピー (対応するフィールドはマップ)	コピー元ファイルに余分なフィールドがあれば排除され、同名のフィールドはマップされる。コピー先ファイルの余分なフィールドがあれば充てんされる。		コマンド終了
*NOCHK	完全コピー (値は無視)	完全コピー (フィールドに関係なく直接データ転送) (注 3 を参照)			

表 6. データベース間のコピー操作 (続き)

FMTOPT パラメータ値 (注 4 を参照)	データベース・ファイルのレコード様式		
		コピー元ファイルとコピー先ファイルの全部のフィールド名が同じ (同名)	コピー元ファイルとコピー先ファイルの一部のフィールド名が同じ

注:

1. フィールド属性にはデータ・タイプ (文字、ゾーン、パック 10 進数、2 進数または浮動小数点)、フィールド長、小数部 (数字フィールドの場合)、日付または時刻様式 (日付または時刻フィールドの場合)、ヌル機能、CCSID、およびフィールドが可変長か固定かどうかが含まれます。
2. マッピングは、コピー元ファイル・フィールド内のデータを、対応する (類似名の) コピー先ファイル・フィールドの属性に変換することから成り立っています。対応するフィールドの属性にデータを変換できない場合、コピーは打ち切られます。
3. レコードは、必要に応じて、埋め込みまたは切り捨てを実施されます。コピー元ファイル内のデータは、コピー先ファイル・レコード様式と一致しないことがあります。
4. *CVTFLOAT 値または *NULLFLAGS 値が指定された場合、FMTOPT に他のいかなる値を指定してもそれは無視されます (*CVTFLOAT 値および *NULLFLAGS 値は例外)。

異なるフィールド・タイプおよび属性でのデータの指定:

FMTOPT(*MAP) を使用して、固定長フィールドと可変長フィールド間、および最大長が異なる可変長フィールド間のデータをマップすることができます。

長さゼロの可変長フィールドを可変長コピー先フィールドにマップする場合、コピー先の長さはゼロに設定されます。

長さゼロの可変長フィールドを固定長コピー先フィールドにマップする場合、そのコピー先フィールドが DBCS 専用フィールドでなければ、コピー先フィールドには単一バイトのブランク (X'40') が埋め込まれます。DBCS 専用コピー先フィールドの場合は、シフトアウトおよびシフトイン (SO-SI) 文字で囲まれた X'4040' に設定されます。

可変長フィールドを可変長フィールドにマップする場合、コピー元フィールドの長さがゼロではなく、大括弧で囲まれた DBCS フィールドとの間でグラフィック・フィールドがマップされなければ、以下の処置が行われます。

- コピー元フィールドのデータ長がコピー先フィールドの最大長より短い場合は、可変長コピー元フィールドの長さが可変長コピー先フィールドにコピーされます。
- コピー元フィールド・データの長さがコピー先フィールドの最大長より長い場合、コピー元フィールドのデータはコピー先フィールドの最大長で切り捨てられ、コピー先フィールドの長さはその最大長になります。なお、データはその保全性を確保する方法で切り捨てられます。

注: 次の例では、x がブランク、< がシフトアウト文字、> がシフトイン文字を表しています。2 バイトの長さは、この例では、文字で表わしていますが、実際には 2 進数です。

可変長コピー元
文字フィールド
(最大長 8 文字)

可変長コピー先
文字フィールド
(最大長 5 文字)

00XXXXXXXX — マップされる → 00XXXXX
03[ABC]XXXXX — マップされる → 03[ABC]XX
07[ABCDEFGX] — マップされる → 05[ABCDE]

可変長 DBCS 専用
コピー元フィールド
(最大長 8 文字)

可変長 DBCS 混用
コピー先フィールド
(最大長 5 文字)

04<AA>XXXX — マップされる → 04<AA>X
08<AABBC> — マップされる → 05<AA>X

RV2H082-1

可変長フィールドから固定長フィールドへのマッピング: コピー元フィールドのデータの長さがコピー先フィールドのデータ長以下の場合、データは固定長コピー先フィールドにコピーされて保全性を保証するように埋められます。

コピー元フィールド・データの長さがコピー先フィールドのデータ長より長い場合、コピー元フィールド・データはコピー先フィールドにコピーされてその保全性を保証する方法で右側が切り捨てられます。

可変長コピー元
文字フィールド
(最大長 8 文字)

固定長コピー先
文字フィールド
(6 文字)

00XXXXXXXX — マップされる →]XXXXXXXX
04[ABCD]XXXX — マップされる → [ABCD]XX
08[ABCDEFGH] — マップされる → [ABCDEF]

RV2H083-1

固定長フィールドから可変長フィールドへのマッピング: コピー先フィールドがコピー元フィールド長以上の最大長を持つ場合、コピー元フィールド・データはコピー先フィールドのデータ部分にコピーされて右側に単一バイトのブランクが埋め込まれます。コピー先フィールド長はコピー元フィールド長の長さに決定されます。

固定長コピー元
文字フィールド
(6 文字)

可変長コピー先
文字フィールド
(最大長 9 文字)

XXXXXX — マップされる → 06XXXXXXXXXX
[ABC]XXX — マップされる → 06[ABC]XXXXXXXX
[ABCDEF] — マップされる → 06[ABCDEF]XXX

RV2H084-1

コピー元フィールドの長さが可変長コピー先フィールドの最大長より長い場合、可変長コピー先フィールドの長さ部分は可変長コピー先フィールドの最大長に決定されます。固定長コピー元フィールドのデータは可

変長コピー先フィールドのデータ部分にコピーされ、その保全性を保証する方法で右側が切り捨てられます。

固定長コピー元
文字フィールド
(8 文字)

可変長コピー先
文字フィールド
(最大長 4 文字)

AB C D E F G H ——— マップされる —→ 0 4 A B C D

固定長 DBCS 専用
コピー元フィールド
(8 文字)

可変長 DBCS 専用
コピー先フィールド
(最大長 4 文字)

< A A B B C C > ——— マップされる —→ 0 4 < A A >

RV2H085-1

FMTOPT(*MAP) または FMTOPT(*NOCHK) を使用する日付、時刻、およびタイム・スタンプ・フィールド: 以下のいずれかの場合には、FMTOPT(*MAP) または FMTOPT(*NOCHK) を CPYF コマンドで指定する必要があります。

- コピー元ファイルがデータベース・データ・ファイルである。
- コピー先ファイルが物理データ・ファイルである。
- レコード様式が同一でない。

コピー元ファイルとコピー先ファイルに対応する日付、時刻、およびタイム・スタンプ・フィールドは、同一様式の属性とレコード様式用の同一区切り記号を持たなければなりません。CPYFRMQRYP コマンドについては、(コピー元ファイル様式ではなく) オープン Query ファイル・レコード様式が使用されること以外は同じことが当てはまります。

FMTOPT(*NOCHK) を使用すると、レコード・データはフィールド・タイプに関係なくコピー先ファイルへ左から右に直接コピーされます。

FMTOPT(*CVTSRC) を使用すると、レコードのデータ部分はフィールド・タイプに関係なくコピー先ファイルへ左から右に直接コピーされます。

FMTOPT(*DROP) を使用すると、コピー先ファイルになくコピー元ファイルにあるフィールドは除去されます。コピー元ファイルおよびコピー先ファイルの同名フィールドが日付、時刻、またはタイム・スタンプ・フィールドである場合、対応するフィールドは同一タイプでなければならず、同一様式の属性と区切り記号を持ち、同じレコード様式中で同名フィールドと相対位置になければなりません。そうでない場合、FMTOPT(*MAP) も必要となることがあります。

FMTOPT(*MAP) を指定すると、様式や区切り記号に関係なく日付、時刻、およびタイム・スタンプ・フィールドの同一タイプ間でのコピーを行います。また、日付、時刻、およびタイム・スタンプ・フィールド、長さ、様式、および値が変換可能なゾーン 10 進数または文字フィールド・タイプとの間でコピーを行うことができます。なお、FMTOPT(*MAP) はコピー先フィールド・タイプ (適用する場合は様式および区切り記号) に変換する場合に必要です。

54 ページの表 7 は、日付、時刻、およびタイム・スタンプで行われる変換を概説しています。

表 7. 変換表

日付タイプ	様式	許容されるフィールド長	方向	データ・タイプ	様式	許容されるフィールド長
日付	任意日付様式	6、8、または 10	<-->	日付	任意	6、8、または 10
ゾーン 10 進数	(MMDDYY)	6,0	<-->	日付	任意	6、8、または 10
ゾーン 10 進数	(DDMMYY)	6,0	<-->	日付	任意	6、8、または 10
ゾーン 10 進数	(YYMMDD)	6,0	<-->	日付	任意	6、8、または 10
ゾーン 10 進数	(YYDDD)	5,0	<-->	日付	任意	6、8、または 10
文字	(MMdDDdYY)	6 最小	<-->	日付	任意	6、8、または 10
文字	(DDdMMdYY)	6 最小	<-->	日付	任意	6、8、または 10
文字	(YYdMMdDD)	6 最小	<-->	日付	任意	6、8、または 10
文字	(YYdDDD)	6 最小	<-->	日付	任意	6、8、または 10
文字	(*USA)	6 最小	---->	日付	任意	6、8、または 10
文字	(*ISO)	6 最小	---->	日付	任意	6、8、または 10
文字	(*EUR)	6 最小	---->	日付	任意	6、8、または 10
文字	(*JIS)	6 最小	---->	日付	任意	6、8、または 10
文字	(YYYYDDD)	6 最小	---->	日付	任意	6、8、または 10
時刻	任意時刻様式	8	<-->	時刻	任意	8
ゾーン 10 進数	(HHMMSS)	6,0	<-->	時刻	任意	8
文字	(HHtMMtSS)	4 最小	---->	時刻	任意	8
文字	(*USA)	4 最小	---->	時刻	任意	8
文字	(*ISO)	4 最小	---->	時刻	任意	8
文字	(*EUR)	4 最小	---->	時刻	任意	8
文字	(*JIS)	4 最小	---->	時刻	任意	8
文字	(HHtMMtSS)	8 最小	<----	時刻	任意	8
タイム・スタンプ	SAA 様式	26	<-->	タイム・スタンプ	SAA	26
ゾーン 10 進数	(YYYYMMDDHHMMSS)	14、0	<-->	タイム・スタンプ	SAA	26
文字	SAA 様式	14 最小	---->	タイム・スタンプ	SAA	26
文字	(YYYYMMDDHHMMSS)	14 最小	<-->	タイム・スタンプ	SAA	26

表 7. 変換表 (続き)

日付タイプ	様式	許容されるフィールド長	許容されるフィールド方向	データ・タイプ	様式	許容されるフィールド長
注: 様式列の意味は次のとおりです。						
d	= 日付区切り記号					
t	= 時刻区切り記号					
任意	= ジョブ様式または SAA 様式					
許容するフィールド長で最小 というのは、指定する長さが変換に必要な最小値であることを意味します。この長さが望ましい前提様式に対して充分でない場合、変換エラーが生じることがあります。日付、時刻、およびタイム・スタンプのデータ・タイプおよびキーワードの詳細については、DDS concepts を参照してください。						

文字フィールドを日付、時刻、またはタイム・スタンプ・フィールドに変換するとき、FMTOPT(*MAP)が指定されて対応するコピー元フィールド名とコピー先フィールド名が一致し、システムは文字フィールドが同じデータ様式を持つかを判別します。以下の規則が適用されます。

- **文字フィールドを日付フィールドに変換する場合**、文字フィールドに必要な最小長は 6 です。サーバーは最初に文字フィールド・データが、コピー・コマンドを実行する現行ジョブに指定されたのと同様式であり、同じ区切り記号を持つかどうかを判別します。様式として *MDY、*DMY、*YMD、または *JUL、区切り記号としてスラッシュ (/)、ハイフン (-)、ピリオド (.)、コンマ (,)、またはブランク () があります。文字フィールドが現行ジョブ指定の様式および区切り記号様式でない場合、SAA 様式 (*ISO、*USA、*EUR、または *JIS) のいずれかであるかどうかを判別します。また、YYYYDDD 様式 (区切り記号なし) であるかどうかも判別します。文字フィールドがそれらの様式のいずれかであることをサーバーが判別したとき、それを日付コピー先フィールドに変換します。文字フィールドの日付部分は左詰めにしなければなりません。後ろにはブランクを埋め込むことができます。
- **文字フィールドを時刻フィールドに変換する場合**、文字フィールドに必要な最小長は 4 です。サーバーは最初に文字フィールド・データが、コピー・コマンドを実行する現行ジョブに指定されたのと同様式であり、同じ区切り記号を持つかどうかを判別します。様式として *HMS、区切り記号としてコロン (:)、コンマ (,)、ピリオド (.)、またはブランク () があります。文字フィールドが現行ジョブ指定の様式および区切り記号様式でない場合、サーバーは SAA 様式 (*ISO、*USA、*EUR、または *JIS) のいずれかであるかどうかを判別します。文字フィールドがそれらの様式のいずれかであることをサーバーが判別した場合、それを時刻コピー先フィールドに変換します。文字フィールドの時刻部分は左詰めにしなければなりません。そして、後ろにブランクを埋め込むことができます。
- **文字フィールドをタイム・スタンプ・フィールドに変換する場合**、文字フィールドに必要な最小フィールド長は 14 です。サーバーは最初に、文字フィールド・データが次のいずれかのタイプであるかどうかを判別します。

- SAA 様式
- YYYYMMDDHHMMSS 様式

文字フィールドがそれらの様式のいずれかであることをサーバーが判別した場合、それをタイム・スタンプ・コピー先フィールドに変換します。文字フィールドのタイム・スタンプ部分は左詰めにしなければなりません。そして、後ろにブランクを埋め込むことができます。

日付、時刻、またはタイム・スタンプ・フィールドを文字フィールドに変換するとき、FMTOPT(*MAP)が指定され、対応するコピー元フィールド名とコピー先フィールド名が一致し、サーバーが日付、時刻、タイム・スタンプ・フィールドを現在のジョブによって指定されている様式に変換します。以下の規則が適用されます。

- **日付フィールドを文字フィールドに変換する場合**、文字フィールドに必要な最小フィールド長は 6 です。サーバーは最初に、コピー・コマンドを実行する現行ジョブの日付様式および区切り記号の属性を判別します。様式として *MDY、*DMY、*YMD、または *JUL、区切り記号としてスラッシュ (/)、ハイフン (-)、ピリオド (.)、コンマ (,)、またはブランク () があります。日付フィールドは、現行ジョブの指定様式の文字フィールドに変換されます。文字フィールドが変換に必要なフィールド長より長い場合、データは左詰めにされて後ろにブランクが追加されます。
- **時刻フィールドを文字フィールドに変換する場合**、文字フィールドに必要な最小フィールド長は 8 です。サーバーは最初に、コピー・コマンドを実行する現行ジョブの時刻区切り記号の属性を判別します。これにはコロン (:)、コンマ (,)、ピリオド (.)、またはブランク () があります。時刻フィールドは、*HMS 様式 (現行ジョブの指定区切り記号を含む) の文字フィールドに変換されます。文字フィールドが変換に必要なフィールド長より長い場合、データは左詰めにされて後ろにブランクが追加されます。
- **タイム・スタンプ・フィールドを文字フィールドに変換する場合**、文字フィールドに必要な最小フィールド長は 14 です。タイム・スタンプ・フィールドは、YYYYMMDDHHMMSS 様式 (区切り記号なし) の文字フィールドに変換されます。文字フィールドが変換に必要なフィールド長より長い場合、データは左詰めにされて後ろにブランクが追加されます。

ゾーン 10 進数フィールドを日付、時刻、またはタイム・スタンプ・フィールドに変換するとき、FMTOPT(*MAP) が指定されて、対応するコピー元フィールド名およびコピー先フィールド名が一致し、サーバーはゾーン 10 進数フィールドが現行ジョブの指定する様式であると見なします。以下の規則が適用されます。

- **ゾーン 10 進数フィールドを日付フィールドに変換する場合**、サーバーは、ゾーン 10 進数フィールド・データがコピー・コマンドを実行する現行ジョブに指定されたのと同じ日付様式 (区切り記号なし) であると見なします。これは *MDY、*DMY、*YMD、または *JUL となります。ゾーン 10 進数フィールド長は、5、0 (現行ジョブ様式が *JUL の場合) または 6、0 (現行ジョブ様式が *MDY、*DMY、または *YMD の場合) でなければなりません。サーバーは、日付コピー先フィールドにそれを変換またはコピーしようとします。
- **ゾーン 10 進数フィールドを時刻フィールドに変換する場合**、サーバーは、ゾーン 10 進数フィールド・データが *HMS 様式 (区切り記号なし) であることを前提とします。ゾーン 10 進数フィールド長は 6、0 でなければなりません。サーバーは、時刻コピー先フィールドにそれを変換またはコピーしようとします。
- **ゾーン 10 進数フィールドをタイム・スタンプ・フィールドに変換する場合**、サーバーは、ゾーン 10 進数フィールド・データが YYYYMMDDHHMMSS 様式 (区切り記号なし) であると見なします。ゾーン 10 進数フィールド長は 14、0 でなければなりません。サーバーは、タイム・スタンプコピー先フィールドにそれを変換またはコピーしようとします。

日付、時刻、またはタイム・スタンプ・フィールドをゾーン 10 進数フィールドに変換するとき、FMTOPT(*MAP) が指定されて、対応するコピー元フィールド名およびコピー先フィールド名が一致し、サーバーは現行ジョブの指定様式を使用してゾーン 10 進数データがどの様式であるべきかを判別します。このとき、以下のことが適用されます。

- **日付フィールドをゾーン 10 進数フィールドに変換する場合**、サーバーは、ゾーン 10 進数フィールド・データがコピー・コマンドを実行する現行ジョブに指定されたのと同じ日付様式 (区切り記号なし) であると見なします。これは *MDY、*DMY、*YMD、または *JUL となります。ゾーン 10 進数フィールド長は、5、0 (現行ジョブ様式が *JUL の場合) または 6、0 (現行ジョブ様式が *MDY、*DMY、または *YMD の場合) でなければなりません。サーバーは、日付フィールドをそれに変換またはコピーしようとします。

- **時刻フィールドをゾーン 10 進数フィールドに変換する場合**、サーバーは、ゾーン 10 進数フィールド・データが *HMS 様式 (区切り記号なし) であると見なします。ゾーン 10 進数フィールド長は 6、0 でなければなりません。サーバーは、時刻フィールドをそれに変換またはコピーしようとします。
- **タイム・スタンプ・フィールドをゾーン 10 進数フィールドに変換する場合**、サーバーは、ゾーン 10 進数フィールド・データが YYYYMMDDHHMMSS 様式 (区切り記号なし) であると見なします。ゾーン 10 進数フィールド長は 14、0 でなければなりません。サーバーは、タイム・スタンプ・フィールドをそれに変換またはコピーしようとします。

データ値、データ様式、またはデータ長のエラーのために変換に障害が出た場合、情報メッセージが送信されます。なお、コピー先ファイル・フィールドはそのデフォルト値に設定されます。

関連情報

DDS 概念

FMTOPT(*MAP) または FMTOPT(*NOCHK) を使用するヌル可能フィールド: 以下の場合には、FMTOPT(*MAP) または FMTOPT(*NOCHK) を、CPYF コマンドで指定しなければなりません。

- コピー元ファイルがデータベース・データ・ファイルである。
- コピー先ファイルが物理データ・ファイルである。
- レコード様式が同一でない。

レコード様式が同一であるべきなので、コピー元ファイルとコピー先ファイルの対応するフィールドは、両方ともヌル可能フィールドであるかまたは両方がそうでないかのどちらかでなければなりません。

CPYFRMQRYP コマンドについては、(コピー元ファイル様式ではなく) オープン Query ファイル・レコード様式が使用されること以外は同じことが当てはまります。

FMTOPT(*MAP) を使用する場合:

- **ヌル値は、ヌル可能なコピー元ファイル・フィールドから、同様の名前を持つヌル可能なコピー先フィールドにコピーされます。**このコピーは、フィールド属性および長さに互換性のある場合にのみ行うことができます。
- **ヌル可能でないフィールドは、フィールドの属性と長さに互換性がある場合にそのフィールドからコピーされたり、そこにコピーされます。**コピー先フィールドで予期される結果は、以下のとおりです。
 - **ヌル可能フィールドからヌル可能フィールドにコピーした場合**

コピー元ファイル・フィールドのヌル値は、コピー先ファイル・フィールドにコピーされます。コピー元ファイル・フィールドのヌルでない値も、コピー先ファイル・フィールドにコピーされます。コピー元ファイル・フィールドのヌルでない値がコピー中に変換エラーとなった場合、コピー先ファイル・フィールドのデフォルト値がコピー先ファイル・フィールドに入れられます。

- **ヌル可能でないフィールドからヌル可能フィールドにコピーした場合**

コピー元ファイル・フィールドのヌルでない値は、コピー先ファイル・フィールドにコピーされます。コピー元ファイル・フィールドの値がコピー中に変換エラーとなった場合、コピー先ファイル・フィールドのデフォルト値がコピー先ファイル・フィールドに入れられます。

- **ヌル可能フィールドからヌル可能でないフィールドにコピーした場合**

コピー元ファイル・フィールドのヌルでない値は、コピー先ファイル・フィールドにコピーされません。ヌルでない値をコピーしている時に変換エラーが生じた場合、またはコピー元ファイル・フィールドがヌルの場合には、コピー先ファイル・フィールドのデフォルト値がコピー先ファイル・フィールドに入れられます。

FMTOPT(*NONE) を使用した場合、データベース・ファイルを同一レコード様式の物理データ・ファイルにコピーするときに、コピー元ファイル・フィールドのヌル値はコピー先ファイルにコピーされます。

FMTOPT(*DROP) を使用した場合、ヌル値はコピーされます。

FMTOPT(*NOCHK) または FMTOPT(*CVTSRC) を使用した場合、レコード・データはフィールド・タイプに関係なくコピー先ファイルの左から右に直接コピーされます。*NOCHK または *CVTSRC を指定した場合、ヌル値はコピーされません。それは、このレコード様式は同一である必要がないからです。ヌル値ではなく、ユーザー指定の値かデフォルト値のいずれかがコピー先ファイルにコピーされます。

FMTOPT(*MAP) または FMTOPT(*NOCHK) を使用する CCSID: FMTOPT(*NOCHK) が指定されたとき、CCSID 変換は行われません。レコード・データはフィールド・タイプまたは CCSID に関係なくコピー先ファイルの左から右に直接コピーされます。

FMTOPT(*MAP) が指定されて有効な変換がコピー元フィールドの CCSID とコピー先ファイル・フィールドの CCSID 間に定義されると、文字データはコピー先ファイル・フィールドの CCSID に変換されます。しかし、コピー元ファイル・フィールドの CCSID またはコピー先ファイル・フィールドの CCSID が 65535 の場合、変換は行われません。

FMTOPT(*NONE) が指定されたとき、コピー元ファイルおよびコピー先ファイルの属性は、対応するフィールドの CCSID が 1 つも 65535 ではない場合、同一様式でなければなりません。

CPYFRMQRYF コマンドを使用した場合、FMTOPT 規則は、変更照会様式がコピー元ファイル様式の代わりに使用されること以外は同じです。

FMTOPT(*MAP) または FMTOPT(*NOCHK) を使用する DBCS グラフィック・フィールド: 大括弧で囲われた DBCS フィールドにグラフィック・フィールドをマップするとき、シフトアウト文字およびシフトイン文字が DBCS データの前後に追加されます。大括弧で囲われた DBCS フィールドからグラフィック・フィールドにマップするとき、このシフトアウト文字とシフトイン文字は除去されます。可変長フィールドにおいて、グラフィック・フィールド長は DBCS 文字の数で表され、大括弧で囲われた DBCS フィールド長はバイト数 (シフトアウト文字およびシフトイン文字を含む) で表されます。可変の大括弧で囲われた DBCS フィールドにマップする、またはそこから可変長グラフィック・フィールドをマップする場合に、この相違を考慮してください。

DBCS 混用フィールドをグラフィック・フィールドにコピーする場合 CPYF コマンドに FMTOPT(*MAP) を指定すると、DBCS 混用フィールドに SBCS データ (ブランクを含む) があると、変換エラーが発生します。グラフィック・フィールドにコピーする際には、有効な DBCS データ (DBCS 混用フィールドにある) に続く後ろの SBCS ブランクを無視した方がよい場合があります。これにより、変換エラーなしでコピー操作を行うことができます。このタイプのコピーは、OPNQRYF コマンドと CPYFRMQRYF コマンドの組み合わせを使用して行うことができます。OPNQRYF コマンドは、後ろの 1 バイト・ブランクを除去してデータを可変長 DBCS 混用フィールドに入れます。FMTOPT(*MAP) を指定した CPYFRMQRYF コマンドは、可変長 DBCS 混用フィールドをグラフィック・フィールドにコピーします。

たとえば、FILEO という名前のファイル内にある DBCS 混用フィールドを、FILEG という名前のファイル内にあるグラフィック・フィールドにコピーするとします。追加のファイル (FILEV) が作成されなければなりません。

元のコピー元ファイル FILEO の DDS:

```
***** データの始め *****
      A          R FMT01
      A          FLD1          100          CCSID(65535)
```



```

      A          FLD2          70          CCSID(65535)
      A          FLD3          20A
***** データの終わり *****

```

FILEV 用の DDS: このファイル様式は、OPNQRYF コマンドの FORMAT パラメーターで指定されます。FILEO との唯一の相違は、グラフィック・フィールドに変換する DBCS 混用フィールドが可変長として定義されていることです。

```

***** データの始め *****
      A          R FMT01
      A          FLD1          100          VARLEN CCSID(65535)
      A          FLD2          70          VARLEN CCSID(65535)
      A          FLD3          20A
***** データの終わり *****

```

新規ファイル FILEG 用の DDS: グラフィック・フィールドは、固定長として定義されます。しかし、必要に応じて可変長にすることもできます。

```

***** データの始め *****
      A R FMT01 A FLD1 4G CCSID(65535) A FLD2 3G CCSID(65535) A FLD3 20A
***** データの終わり *****

```

データを FILEO にある DBCS 混用フィールドから FILEG にあるグラフィック・フィールドにコピーするのに使用されるコマンドは以下のとおりです。

```

CHGJOB CCSID(65535)
OPNQRYF FILE((MYLIB/FILEO)) FORMAT(MYLIB/FILEV *ONLY) MAPFLD((FLD1 '%STRIP(1/FLD1 *TRAIL)') (FLD2 '%STRIP(1/FLD2 *TRAIL)'))
CPYFRMQRYF FROMOPNQID(FILEO) TOFILE(MYLIB/FILEG) MBROPT(*REPLACE) FMTOPT(*MAP)

```

汎用文字セット (UCS-2) グラフィック・フィールドの変換:

汎用文字セット (UCS-2) グラフィック・フィールドを変換するには、FMTOPT(*MAP) を使用します。

FMTOPT(*MAP) を使用して UCS-2 グラフィック・フィールドに コピーする場合、データは、コピー元フィールドの CCSID から UCS-2 グラフィック・コピー先フィールドの CCSID に変換されます。変換後の長さが UCS-2 グラフィック・コピー先フィールドの長さよりも短い場合、埋め込みが行われます。変換後の長さが UCS-2 グラフィック・コピー先フィールドの長さよりも長い場合、切り捨てが行われます。

FMTOPT(*MAP) を使用して UCS-2 グラフィック・フィールドから コピーする場合、データは、UCS-2 グラフィック・フィールドの CCSID からコピー先フィールドの CCSID に変換されます。変換後の長さがコピー先フィールドの長さよりも短い場合、埋め込みが行われます。変換後の長さがコピー先フィールドの長さよりも長い場合、切り捨てが行われます。DBCS グラフィック・フィールドおよび UCS-2 グラフィック・フィールド以外では、2 バイトデータの前後にシフトアウト文字とシフトイン文字が追加されます。

UCS-2 変換の可能性には、可変長フィールドも含まれます。一般に、可変長コピー先フィールドの結果としての長さは、最小が変換されたコピー元フィールドの長さで、最大がコピー先フィールドの長さです。

UCS-2 グラフィック・フィールドをコピー元またはコピー先とするとき、CCSID 変換が実行されると、データのバイト長は変換後に異なることがあります。コピー先フィールドの最大長は、変換されたデータのすべてがコピー先フィールドに入るかどうかを判別します。

最初のバイトがすでに有効な SBCS 文字 (シフトアウト X'0E' でない) に変換されているデータを UCS-2 グラフィック・フィールドから文字フィールドまたは DBCS 択一フィールドにコピーする場合、SBCS に変換できない残りの文字 (UCS-2 フィールドの実際の 2 バイト・データ) には 1 バイトの置換文字が設定されます。

最初のバイトがすでに DBCS シフトアウト X'0E' に変換されているデータを UCS-2 グラフィック・フィールドから DBCS 専用、DBCS グラフィック・フィールド、または DBCS 択一フィールドにコピーする場合、DBCS に変換できない残りの文字には 2 バイトの置換文字が設定されます。

FMTOPT(*MAP) を使用して UCS-2 データを、または UCS-2 データに変換する場合、正常に行うためには、コピー先フィールドの結果としての変換データがコピー先フィールドのタイプに標準化される必要があります。そうしないと変換エラーが生じて、コピー先フィールドは省略値に設定されます。

UCS-2 グラフィック・フィールドの制約事項:

コピー元またはコピー先が UCS-2 グラフィック・フィールドの場合、いくつかの制約事項があります。

FMTOPT(*MAP) を指定した CPYF および CPYFRMQRYP コマンドは、対応するフィールドが UCS-2 または DBCS グラフィック・フィールド (65535 を含む任意の CCSID) であるか、65535 以外の CCSID を持つ文字、DBCS 混合、DBCS 択一、または DBCS 専用フィールドでなければ、UCS-2 グラフィック・フィールドをコピー元またはコピー先にはできません。CCSID に 65535 が許可される他の唯一の対応するタイプは、DBCS グラフィックです。

CPYF コマンドの次のパラメーターは、UCS-2 グラフィック・フィールド、UTF-16 グラフィック・フィールド、UTF-8 文字フィールドをサポートしません。

- UCS-2 グラフィック・キー・フィールド、UTF-16 グラフィック・キー・フィールド、UTF-8 文字キー・フィールドを参照する FROMKEY(*BLDKEY)
- UCS-2 グラフィック・キー・フィールド、UTF-16 グラフィック・キー・フィールド、UTF-8 文字キー・フィールドを参照する TOKEY(*BLDKEY)
- UCS-2 グラフィック・フィールド名、UTF-16 グラフィック・フィールド名、UTF-8 文字フィールド名を指定する INCCHAR
- UCS-2 グラフィック・フィールド名、UTF-16 グラフィック・フィールド名、UTF-8 文字フィールド名を指定する INCREL

関連資料

165 ページの『DBCS ファイルのコピー』

スプール DBCS ファイルも、非スプール DBCS ファイルも同様にコピーすることができます。

汎用コード化文字セット・トランスフォーメーション形式 (UTF-8 文字および UTF-16 グラフィック) の変換: FMTOPT(*MAP) を使用して UTF-8 文字フィールドまたは UTF-16 グラフィック・フィールドにコピーする場合、コピー元フィールドの CCSID のデータは、コピー先の UTF-8 文字フィールドまたは UTF-16 グラフィック・フィールドの CCSID に変換されます。変換後の長さがコピー先の UTF-8 文字フィールドまたは UTF-16 グラフィック・フィールドの長さよりも短い場合、埋め込みが行われます。変換後の長さがコピー先の UTF-8 文字フィールドまたは UTF-16 グラフィック・フィールドの長さよりも長い場合、切り捨てが行われます。

FMTOPT(*MAP) を使用して UTF-8 文字フィールドまたは UTF-16 グラフィック・フィールドからコピーする場合、コピー元の UTF-8 文字フィールドまたは UTF-16 グラフィック・フィールドの CCSID のデータは、コピー先フィールドの CCSID に変換されます。変換後の長さがコピー先フィールドの長さよりも短い場合、埋め込みが行われます。変換後の長さがコピー先フィールドの長さよりも長い場合、切り捨てが行われます。

UTF-8 または UTF-16 の変換候補には、可変長フィールドも含まれます。一般に、可変長コピー先フィールドの結果としての長さは、最小が変換されたコピー元フィールドの長さで、最大がコピー先フィールドの長さです。

UTF-8 または UTF-16 のフィールドをコピー元またはコピー先とするとき、CCSID 変換が実行されると、変換後のデータのバイト長が異なる可能性があります。コピー先フィールドの最大長は、変換されたデータのすべてがコピー先フィールドに入るかどうかを判別します。

最初のバイトがすでに有効な SBCS 文字 (シフトアウト X'0E' でない) に変換されているデータを UTF-8 または UTF-16 のフィールドから文字フィールドまたは DBCS 択一フィールドにコピーする場合、SBCS に変換できない残りの文字 (UTF-8 または UTF-16 のフィールドの実際の 2 バイト・データ) には 1 バイトの置換文字が設定されます。

最初のバイトがすでに SBCS シフトアウト (X'0E') に変換されているデータを UTF-8 または UTF-16 のフィールドから DBCS 専用フィールド、DBCS グラフィック・フィールド、または DBCS 択一フィールドにコピーする場合、DBCS に変換できない残りの文字には 2 バイトの置換文字が設定されます。

FMTOPT(*MAP) を使用して UTF-8 または UTF-16 のデータとの間で変換する場合、正常に行うためには、コピー先フィールドの結果としての変換データがコピー先フィールドのタイプに準拠している必要があります。そうしないと変換エラーが生じて、コピー先フィールドは省略値に設定されます。

非正規化ファイルから正規化ファイルにコピーする場合は、FMTOPT (*MAP) を指定する必要があります。以下の表のいずれかが該当する場合には、コピー先ファイルは正規化ファイルになります。

コピー元ファイル (非正規化)	コピー先ファイル (正規化)
UTF-8	UTF-8
UTF-8	UTF-16
UTF-16	UTF-16
UTF-16	UTF-8
UCS2	UTF-8
UCS2	UTF-16

System/370 浮動小数点およびヌル・フィールドの変換:

システム/370™ 様式の浮動小数点フィールドおよびヌル・フィールドを iSeries 様式にコピーするには、浮動小数点フィールドに FMTOPT(*CVTFLOAT)、ヌル・フィールドに FMTOPT(*NULLFLAGS) を使用してください。これら 2 つの値は、1 つのコマンドで同時に使用することができます。FMTOPT(*CVTFLOAT *NULLFLAGS) のようになります。

CPYF コマンドの FMTOPT(*CVTFLOAT) パラメーターは、システム/370 16 進様式からの各浮動小数点フィールドを iSeries で使用されている IEEE 様式に変換します。CPYF コマンドは、物理コピー先ファイルの外部記述によって識別されるこれらのフィールドを変換します。

CPYF コマンドの FMTOPT(*NULLFLAGS) パラメーターは、各ヌル可能フィールドの後にバイト (またはフラグ) を置き、対応する入力フィールドがヌルかどうかを示すのに使用します。CPYF コマンドは、物理コピー先ファイルの外部記述によってヌル可能と識別されるフィールドを取ります。そのバイト (またはフラグ) がブランク (X'40') となっているか、または X'00' が含まれている場合、データはヌルでないと見なされます。そのフラグを他の値にすると、対応する入力フィールドは無視され、出力値はヌルに設定されます。

*CVTFLOAT または *NULLFLAGS が使用され、入力ファイルが外部に記述されている場合、入力ファイルの外部記述は、コピーしたデータのマッピングに使用されません。

*CVTFLOAT および *NULLFLAGS (両方またはどちらか一方) を使用する場合、コピー先ファイルが既存のデータベースで外部記述された物理データ・ファイルであることを確認してください。

次の条件のどれかが真である場合、*CVTFLOAT および *NULLFLAGS 値は、指定できません。

- 複数様式の論理コピー元ファイルに対して RCDFMT(*ALL) が指定されている。
- デフォルト以外の値が CRTFILE に対して指定されており、コピー先ファイルが存在しない。
- デフォルト以外の値が FROMKEY、TOKEY、INCCHAR、INCREL、SRCOPT、または SRCSEQ パラメーターに対して指定されている。

*CVTFLOAT または *NULLFLAGS のどちらかが使用されているとき、FMTOPT パラメーター以外の値はすべて無視されます。*CVTFLOAT および *NULLFLAGS の両方が CPYF コマンドで使用されている場合、両方の値が認識されます。

*CVTFLOAT 値が指定されたとき (および *NULLFLAGS が指定されていない)、コピー元ファイルのレコード長はコピー先ファイルのレコード長となります。*NULLFLAGS 値が指定されたとき、コピー元ファイルのレコード長は、コピー先ファイルのレコード長とコピー先ファイルのヌル可能フィールドの数の合計に等しくなります。コピー元ファイルのレコード長が、予期された長さを下回ることはありません。コピー元ファイルのレコード長が予期した長さよりも大きい場合は、照会メッセージが QSYSOPR メッセージ待ち行列に送られ、継続するかどうか尋ねられます。継続する場合は、コピー元ファイルの後続のデータ (フィールド) は、コピー先ファイルで切り捨てられます。

コピー先ファイルには、正確な結果様式記述が含まれていなければなりません。コピー元ファイルのデータは、コピー先ファイルのデータと対応する同じ位置になければなりません。そうでない場合、出力マッピング・エラーまたは予想不能の結果が発生する可能性があります。日付、時刻、およびタイム・スタンプが、コピー先ファイルのフィールド仕様の外部様式になければなりません。コピー先ファイルの数値データ型は、予期するデータが十分に入る大きさでなければなりません。たとえば、数字を損なわないために、コピー先ファイルの 10 進数データ・タイプは、フィールドのバイト長で最大の精度のものを使用して作成されなければなりません。CCSID 変換がないので、コピー先ファイルのフィールドは、CCSID を予期して作成しなければなりません。

浮動小数点フィールドとヌル・フィールドの変換の詳細については、『システム/370 浮動小数点およびヌル・フィールドの変換におけるエラー』を参照してください。

システム/370 浮動小数点およびヌル・フィールドの変換におけるエラー:

すべての変換エラーは、エラーがある最大 10 個のレコードに対する CPF2958 メッセージが出される原因となります。10 を超えるレコードで変換エラーが生じた場合、コピー操作の後にエラーを生じさせたレコードの数を示す CPF2959 メッセージが発行されます。

コピー先ファイルに浮動小数点が存在せず *CVTFLOAT が指定されている場合は、エラー・メッセージは出されません。しかし、浮動小数点変換は行われません。コピー先ファイルにヌル可能フィールドがなく、*NULLFLAGS が指定されている場合は、エラー・メッセージは出されません。しかし、コピー元ファイルのデータには、ヌル・バイト (またはフラグ) が含まれていないと想定されます。*CVTFLOAT および *NULLFLAGS を使用しているとき、CPYF コマンドは、コピー元ファイルのデータがコピー先ファイルの様式で定義されていると想定します。

*CVTFLOAT および *NULLFLAGS 値は、システム/370 16 進数浮動小数点フィールドの変換またはヌル・フラグを含む変換を必要とするファイルにのみ使用してください。これらの値を他のファイルに使用すると、予想できない結果が起こり、データ破壊および出力エラーが発生する可能性があります。システム/370 浮動小数点フィールドは、一度しか変換することができません。このフィールドを 2 回以上変換すると複数の変換が発生し、データが破壊されることとなります。

この CPYF 機能は、DB2 アンロード機能 (DSNTIAUL) によってテープに置かれたデータと互換性があります。CPYF 機能を実行してデータを変換する前に、SQL または DDS を使用して結果表またはファイルを iSeries サーバー上に手操作で作成する必要があります。これを行うには、DSNTIAUL 機能によって生成される様式を使用してください。

ファイルのコピーに関する変換規則:

表 8 は、コピー元ファイルとコピー先ファイルのレコード様式に関して、マップされるフィールド間で可能なフィールド変換を示しています。同じ名前のフィールドがコピー元ファイルとコピー先ファイルの様式の間で互換性のない属性を持つ場合、コピーを実行するのに使用できるのは FMTOPT(*NOCHK) だけです。X は変換が有効であることを示し、ブランクは有効でないフィールド・マッピングを示してあります。

文字フィールドをマップするときに、コピーされるフィールドはコピーが行われる先のフィールドより長い場合は、右側が切り捨てられます。たとえば、長さ 10 の文字フィールドを長さ 6 の文字フィールドにコピーする場合、ABCDEFGHIJ は ABCDEF になります。コピーされるフィールドが、コピーの行われる先のフィールドより短い場合は、右側にブランクを埋め込まれます。たとえば、長さ 10 の文字フィールドが長さ 12 の文字フィールドにコピーされると、ABCDEFGHIJ は ABCDEFGHIJxx (x はブランク) になります。

数字フィールドをマップする場合、コピーされるフィールドがコピーの行われる先のフィールドより長いと、コピーされるフィールドは小数点の左側および右側が切り捨てられます。たとえば、長さ 9 で小数点以下の桁数が 4 のゾーン 10 進数フィールド 00115.1109 は、長さ 6 で小数点以下の桁数が 3 のゾーン 10 進数フィールドにコピーされると、115.110 になります。

小数点の左側で有効数字が切り捨てられなければならない場合は、値はコピーされず、フィールドはデフォルト値 (DFT キーワードが指定されている場合はそのパラメーター値、指定されていない場合は 0) に設定されます。また、浮動小数点数値指数が大きすぎて有効桁が失われることになる場合も、コピー先ファイル・フィールドはデフォルト値に設定されます。

数字フィールドをマップするとき、コピーされるフィールドがコピーの行われる先のフィールドより短い場合、コピーされるフィールドは、小数点の左側および右側にゼロを埋め込まれます。たとえば、長さ 7 で小数点以下の桁数が 5 のパック 10 進数フィールド 99.99998 は、長さ 10 で小数点以下の桁数が 6 のパック 10 進数フィールドにコピーされると、0099.999980 になります。

表 8. フィールド変換: (コピー元ファイル・データはコピー先ファイルで定義されているので、この表は、FMTOPT(*CVTFLOAT) または FMTOPT(*NULLFLAGS) に適用することはできません。)

コピー元フィールド	コピー先の文字フィールド、2 進文字フィールド、または 16 進数フィールド	コピー先のパック 10 進数フィールド	コピー先のゾーン 10 進数フィールド	コピー先の 2 進数 (小数部なし) フィールド	コピー先の浮動小数点フィールド	コピー先の 2 進数 (小数部あり) フィールド
文字、2 進文字、16 進数	X					
パック 10 進数		X	X	X	X	
ゾーン 10 進数		X	X	X	X	
2 進数 (小数点以下の桁数なし)		X	X	X	X	
浮動小数点		X	X	X	X	

表 8. フィールド変換 (続き): (コピー元ファイル・データはコピー先ファイルで定義されているので、この表は、FMTOPT(*CVTFLOAT) または FMTOPT(*NULLFLAGS) に適用することはできません。)

コピー元フィールド	コピー先の文字フィールド、2 進文字フィールド、または 16 進数フィールド	コピー先のパック 10 進数フィールド	コピー先のゾーン 10 進数フィールド	コピー先の 2 進数 (小数部なし) フィールド	コピー先の浮動小数点フィールド	コピー先の 2 進数 (小数部あり) フィールド
2 進数 (小数点以下の桁数あり)						X ¹
注:						
¹ 1 桁または 2 桁以上の小数部を持つ 2 進数字フィールドがコピーできる先は同じ桁数の小数部を持つ 2 進数フィールドに限られます。						

ソース・ファイルの順序番号フィールドおよび日付フィールドの追加または変更 (SRCOPT および SRCSEQ パラメーター)

ファイルのコピー時には、順序番号フィールド、日付フィールド順序番号、および日付フィールドを追加または変更することができます。

関連概念

25 ページの『ファイルをコピーする際の *ADD の指定』

*ADD を指定すると、各コピー対象レコードはメンバーの中の既存のレコードの終わりに追加されません。

装置ソース・ファイルからデータベース・ソース・ファイルへのコピー:

装置ソース・ファイルからデータベース・ソース・ファイルへコピーするとき、サーバーでは、レコードの先頭に順序番号フィールドおよび日付フィールドを追加します。

最初のレコードに順序番号 1.00 が割り当てられ、次には 2.00 が割り当てられる、というように増分値 1.00 ずつ番号が大きくなっていきます。9999 を超えるレコードがコピーされる場合は、順序番号は 1.00 に戻り、SRCOPT パラメーターおよび SRCSEQ パラメーターがコピー・コマンドで指定されない限り、大きくなり続けます。

同じファイルへの複数のコピーが MBROPT(*ADD) を指定して行われると、ファイル内には重複する順序番号があります。これについては、物理ファイル・メンバー再編成 (RGZPFM) コマンドを使用して修正することができます。

日付フィールドは、ゼロに初期設定されます。

装置との間でコピーするときは、装置ソース・ファイルを使用するより装置データ・ファイルを使用する方が効率的です。コピー機能によって、ソース順序番号フィールドおよび日付フィールドは、必要に応じて自動的に追加または除去されます。

データベース・ソース・ファイルから装置ソース・ファイルへのコピー:

装置ソース・ファイルへコピーするときは、サーバーはレコードの先頭の日付フィールドおよび順序番号フィールドを除去します。

装置との間でコピーするときは、装置ソース・ファイルを使用するより装置データ・ファイルを使用する方が効率的です。コピー機能によって、ソース順序番号フィールドおよび日付フィールドは、必要に応じて自動的に追加または除去されます。

データベース・ソース・ファイルからデータベース・ソース・ファイルへのコピー:

データベース・ソース・ファイルをデータベース・ソース・ファイルにコピーするときには、コピーされるレコードに順序番号を割り当てるために、SRCOPT(*SEQNBR) および SRCSEQ パラメーターを使用します。

CPYSRCF コマンドまたは CPYF コマンドを使用することによって、データベース・ソース・ファイル間でコピーすることができます。CPYSRCF コマンドの方が、パラメーターのデフォルトがデータベース・ソース・ファイルのコピーに適しているため、使用しやすいかもしれません。

順序番号を更新するために、SRCOPT(*SEQNBR) を指定する場合、サーバーは、SRCSEQ パラメーターを考慮します。SRCSEQ パラメーターは、コピーされる最初のレコードに割り当てられる開始値および増分値を指定します。これらのデフォルトは 1.00 および 1.00 です。4 桁以下の整数または 2 桁以下の小数部が、開始値および増分値として指定できます。小数部には小数点を使用しなければなりません。

たとえば、SRCSEQ(100.50) を指定した場合、コピーされるレコードは 100.00、100.50、101.00、101.50、... と続く順序番号を持つことになります。

10000 個以上のレコードを含むファイルを持っているとします。それぞれのレコードが固有の順序番号を持つように、小数桁の増分値を使用してください。開始値 .01 および増分値 .01 が指定された場合は、固有の順序番号を持ってコピーされる最大レコード数は、999 999 になります。最大順序番号が 9999.99 を超えると、そのコピーでの残りのレコードはすべて 9999.99 に設定されます。サーバーでは 1.00 には戻りません。

コピー先のデータベース・ソース・ファイルが到着順アクセス・パスしか持っていない場合、レコードは、常に物理的にファイルの終わりに入れられます。(ファイルにキー順アクセス・パスがないために、レコードはファイルのキーによるアクセス・パスの中に挿入できません。)

複合オブジェクトのコピー

ユーザー定義関数 (UDF)、ユーザー定義タイプ (UDT)、データ・リンク (DL)、ラージ・オブジェクト (LOB)、識別列、または ROWID を含むコピー元ファイルおよびコピー先ファイルの間で、コピーを行うことができます。

ユーザー定義関数を含むファイルのコピー:

ユーザー定義関数 (UDF) を含むファイルをコピーする場合、CPYF および CPYFRMQRYP コマンドで CRTFILE(*YES) を指定することができます。UDF は、新規コピー先ファイルでは作成されません。

iSeries サーバーには、ユーザー定義関数を含む DDM ファイルをコピーすることはできません。

関連情報

CPYF コマンド

CPYFRMQRYP コマンド

ユーザー定義タイプを含むファイルのコピー:

ユーザー定義タイプ (UDT) を含むファイルをコピーする場合、CPYF および CPYFRMQRYP コマンドで CRTFILE(*YES) を指定することができます。コピー元ファイルが UDT を含む SQL 表、ビュー、または索引である場合、これらのコマンドは SQL 表を作成します。

同じ (同一の) UDT との間でコピーを行う場合、FMTOPT(*MAP) を使用して UDT を他の UDT にコピーすることができます。ソース・タイプに互換性があれば、非 UDT から UDT にコピーすることもできます。同一でない UDT 間でコピーする場合、データ・マッピングを行うことはできません。また、UDT から非 UDT にコピーする場合も、データ・マッピングを行うことはできません。

iSeries サーバーには、ユーザー定義タイプを含む DDM ファイルをコピーすることはできません。

関連情報

CPYF コマンド

CPYFRMQRYP コマンド

データ・リンクを含むファイルのコピー:

データ・リンク (DL) を含むファイルをコピーする場合、CPYF および CPYFRMQRYP コマンドで CRTFILE(*YES) を指定することができます。コピー元ファイルが DL を含む SQL 表、ビュー、または索引である場合、これらのコマンドは SQL 表を作成します。

バージョン 4 リリース 3 以前の iSeries サーバーには、データ・リンクを含む DDM ファイルをコピーすることはできません。

DL は、その他の DL にのみマップできます。したがって、FMTOPT パラメーターで *NONE、*MAP、または *DROP を指定する場合、コピー元ファイルとコピー先ファイルには、対応する DL が存在していなければなりません。切り捨ては許されません。ただし、短い DL を長い DL に変換することはできます。

ファイルは、サーバー上で 1 度だけリンクできます。したがって、対応するコピー元ファイル・フィールドとコピー先ファイル・フィールドが両方とも FILE LINK CONTROL である場合、マッピングを実行するコピーや、様式が同一である必要のあるコピー (つまり、*NONE、*MAP、または *DROP が FMTOPT パラメーター上に指定されている) は、成功しません。*NOCHK パラメーター・オプションを使用して実行されるコピーには制限がありませんが、リンクされたファイルを参照する DL が、FILE LINK CONTROL である DL にコピーされると、エラーが発生します。

CPYF または CPYFRMQRYP コマンドで CRTFILE(*YES) を指定する場合に、コピー元ファイルに FILE LINK CONTROL DL フィールドが含まれていると、FMTOPT パラメーターを指定する方法によっては、以下の文のとおりになります。

- FMTOPT パラメーターで *NONE、*MAP、または *DROP を指定する場合、ファイルは作成されますが、エラー・メッセージが発行され、入出力は実行されません。
- FMTOPT パラメーターで *NOCHK または *CVTSRC を指定する場合、ファイルは作成され、入出力が試みられます。有効な LINK が含まれているレコードの入出力は失敗します。

以下の表は、様々な FMTOPT 値が使用される場合に CPYF コマンドと関連する LINK シナリオを示しています。

FMTOPT パラメーターが *MAP または *NONE である場合における、コピー元フィールドとコピー先フィールドの LINK 状況	リンクが実行される方法
FILE LINK CONTROL と FILE LINK CONTROL	許可されていません。ファイルは、1 度だけリンクできません。
NO LINK CONTROL と FILE LINK CONTROL (切り捨てなし)	リンクが実行されます。
FILE LINK CONTROL と NO LINK CONTROL (切り捨てなし)	リンクは実行されません。
NO LINK CONTROL と NO LINK CONTROL (切り捨てなし)	リンクは実行されません。

関連情報

CPYF コマンド

CPYFRMQRYF コマンド

ラージ・オブジェクトを含むファイルのコピー:

ラージ・オブジェクト (LOB) を含むファイルをコピーする場合、CPYF および CPYFRMQRYF コマンドで CRTFILE(*YES) を指定することができます。コピー元ファイルが LOB を含む SQL 表、ビュー、または索引である場合、これらのコマンドは SQL 表を作成します。

iSeries は、3 つのラージ・オブジェクト・データ・タイプ (バイナリー・ラージ・オブジェクト (BLOB)、単一バイトまたは混合文字ラージ・オブジェクト (CLOB)、および 2 バイト文字ラージ・オブジェクト (DBCLOB)) をサポートしています。ファイル・コピー (CPYF) コマンドを使用してこれらのオブジェクトを含むファイルをコピーする場合、以下の制約事項および要件を考慮する必要があります。

- 装置ファイルとの間でのコピーで、*PRINT にコピーする場合、または FMTOPT パラメーター上に *NOCHK または *CVTSRC の値を指定した場合、LOB データはコピーされません。これらの場合、"*POINTER" を含む LOB フィールドのデフォルト・バッファ値のみがコピーされます。このことは、LOB フィールドを含むファイルを同一ファイルにコピーする場合でも当てはまります。有効な LOB データがコピーされるのは、FMTOPT パラメーターに *NONE、*MAP、または *DROP を指定した場合だけです。
- テープまたはディスクにコピーする場合、LOB データはコピーされません。これらの場合、バッファ値 ("*POINTER" を含む) のみがテープまたはディスクに書き込まれます。さらに、テープまたはディスクから同じファイルにコピーする場合、エラーが生じることがあります。このエラーは、ファイルに "*POINTER" 値しか入っておらず、実際の LOB データへの有効なポインターが含まれていないために発生します。
- CPYF コマンドの MBROPT パラメーターに *UPDADD を指定した場合、コピー先ファイルには LOB フィールドを含めることができます。LOB フィールドは、重複キーが見つかった場合にも更新されません。
- CPYF コマンドの FMTOPT パラメーターに *CVTFLOAT または *NULLFLAGS を指定した場合、コピー先ファイルに LOB フィールドを含めることはできません。
- LOB フィールドを含むファイルを印刷したい場合は、CPYF コマンドの TOFILE パラメーターで *PRINT を指定します。印刷リストでは、LOB フィールド・データの代わりに "*POINTER" が表示され、その他の LOB でないフィールド・データも表示されます。TOFILE パラメーターに *PRINT が指

定されておらず、PRINT パラメーターに *COPIED、*EXCLUDE、または *ERROR を指定した場合、コピーするためには、FMTOPT パラメーター上に *NOCHK または *CVTSRC を指定しなければなりません。

- INCCHAR および INCREL パラメーターに LOB フィールドを指定することはできません。INCCHAR パラメーターに *RCD または *FLD を指定することができますが、固定バッファ長のみが比較され、実際の LOB データは比較されません。
- iSeries サーバーには、LOB フィールドを含む DDM ファイルをコピーすることはできません。

以下の表は、コピー操作中に LOB が他のデータ・タイプにマップされる方法を示しています。最初の表は、両方のフィールドに LOB フィールド・タイプが含まれている場合のマッピングを示しています。この表では、以下のガイドラインを考慮してください。

- DATE または TIME タイプと LOB の間でのマッピングは許可されていません。
- これらのマッピングは、特に注記がない限り FMTOPT(*MAP) についてのみ有効です。
- ラージ・オブジェクトについては、通常の文字データ (単一バイト、混合バイト、および 2 バイト) の制約事項と類似の制約事項があります。

表9. 両方のフィールドがラージ・オブジェクトである場合のコピー元ファイルおよびコピー先ファイルのマッピング

フィールド A のタイプ	フィールド B のタイプ	許可されている かどうか、およ びコピー方向	データの CCSID または属性		CCSID	変換が実行さ れる
			フィールド A	フィールド B		
BLOB	BLOB	Y* <—>	65535	65535	同じ	いいえ
CLOB	CLOB	Y* <—>	文字	文字	同じ	いいえ
CLOB	CLOB	Y* <—>	混用	混用	同じ	いいえ
DBCLOB	DBCLOB	Y* <—>	グラフィック	グラフィック	同じ	いいえ
DBCLOB	DBCLOB	Y* <—>	UCS2	UCS2	同じ	いいえ
CLOB	CLOB	Y <—>	文字	文字	異なる	はい
CLOB	CLOB	Y <—>	混用	混用	異なる	はい
DBCLOB	DBCLOB	Y <—>	グラフィック	グラフィック	異なる	はい
DBCLOB	DBCLOB	Y <—>	UCS2	UCS2	異なる	はい
CLOB	CLOB	Y <—>	文字	混用	異なる	はい
CLOB	DBCLOB	N	文字	グラフィック	異なる	—
CLOB	DBCLOB	Y <—>	混用	グラフィック	異なる	はい
CLOB	DBCLOB	Y <—>	文字	UCS2	異なる	はい
CLOB	DBCLOB	Y <—>	混用	UCS2	異なる	はい
DBCLOB	DBCLOB	Y <—>	グラフィック	UCS2	異なる	はい
BLOB	CLOB	Y <—>	65535	文字	異なる	いいえ
BLOB	CLOB	Y <—>	65535	混用	異なる	いいえ
BLOB	DBCLOB	N	65535	グラフィック	異なる	—
BLOB	DBCLOB	N	65535	UCS2	異なる	—
DBCLOB	DBCLOB	Y <—>	1200	1200	同じ	いいえ
CLOB	DBCLOB	Y <—>	文字	1200	異なる	はい
CLOB	DBCLOB	Y <—>	混用	1200	異なる	はい
DBCLOB	DBCLOB	Y <—>	グラフィック	1200	異なる	はい
BLOB	DBCLOB	N	65535	1200	異なる	—

表9. 両方のフィールドがラージ・オブジェクトである場合のコピー元ファイルおよびコピー先ファイルのマッピング (続き)

フィールド A のタイプ	フィールド B のタイプ	許可されている かどうか、およ びコピー方向	データの CCSID または属性		CCSID	変換が実行さ れる
			フィールド A	フィールド B		
CLOB	CLOB	Y <—>	1208	1208	同じ	いいえ
CLOB	CLOB	Y <—>	文字	1208	異なる	はい
CLOB	CLOB	Y <—>	混用	1208	異なる	はい
DBCLOB	CLOB	Y <—>	グラフィック	1208	異なる	はい
BLOB	CLOB	N	65535	1208	異なる	—

注: * これらのマッピングは、FMTOPT(*MAP)、FMTOPT(*NONE)、および FMTOPT(*DROP) の場合に有効です。

2 番目の表は、固定長データ・タイプとラージ・オブジェクトとの間のマッピングを示しています。

表10. 固定長データ・タイプとラージ・オブジェクトの間でのコピー元ファイルおよびコピー先ファイルのマッピング

フィールド A のタイプ	フィールド B のタイプ	許可されてい るかどうか、 およびコピー 方向	データの CCSID または属性		CCSID	変換が実行さ れる
			フィールド A	フィールド B		
文字	BLOB	Y <—>	文字	65535	異なる	いいえ
混用	BLOB	Y <—>	混用	65535	異なる	いいえ
択一	BLOB	Y <—>	択一	65535	異なる	いいえ
専用	BLOB	Y <—>	専用	65535	異なる	いいえ
グラフィック	BLOB	N	グラフィック	65535	異なる	—
UCS2	BLOB	N	UCS2	65535	異なる	—
文字	CLOB	Y <—>	文字	文字	同じ/異なる	いいえ/はい
混用	CLOB	Y <—>	混用	文字	異なる	はい
択一	CLOB	Y <—>	択一	文字	異なる	はい
専用	CLOB	Y <—>	専用	文字	異なる	はい
グラフィック	CLOB	N	グラフィック	文字	異なる	—
UCS2	CLOB	Y <—>	UCS2	文字	異なる	はい
文字	CLOB	Y <—>	文字	混用	異なる	はい
混用	CLOB	Y <—>	混用	混用	同じ/異なる	いいえ/はい
択一	CLOB	Y <—>	択一	混用	異なる	はい
専用	CLOB	Y <—>	専用	混用	異なる	はい
グラフィック	CLOB	Y <—>	グラフィック	混用	異なる	はい
UCS2	CLOB	Y <—>	UCS2	混用	異なる	はい
文字	DBCLOB	N	文字	グラフィック	異なる	—
混用	DBCLOB	Y <—>	混用	グラフィック	異なる	はい
択一	DBCLOB	Y <—>	択一	グラフィック	異なる	はい
専用	DBCLOB	Y <—>	専用	グラフィック	異なる	はい
グラフィック	DBCLOB	Y <—>	グラフィック	グラフィック	同じ/異なる	いいえ/はい
UCS2	DBCLOB	Y <—>	UCS2	グラフィック	異なる	はい

表 10. 固定長データ・タイプとラージ・オブジェクトの間でのコピー元ファイルおよびコピー先ファイルのマッピング (続き)

フィールド A のタイプ	フィールド B のタイプ	許可されているかどうか、およびコピー方向	データの CCSID または属性		CCSID	変換が実行される
			フィールド A	フィールド B		
文字	DBCLOB	Y <—>	65535 でない	UCS2	異なる	はい
混用	DBCLOB	Y <—>	65535 でない	UCS2	異なる	はい
択一	DBCLOB	Y <—>	65535 でない	UCS2	異なる	はい
専用	DBCLOB	Y <—>	65535 でない	UCS2	異なる	はい
グラフィック	DBCLOB	Y <—>	グラフィック	UCS2	異なる	はい
UCS2	DBCLOB	Y <—>	UCS2	UCS2	同じ/異なる	いいえ/はい
文字	DBCLOB	N	65535	UCS2	異なる	—
混用	DBCLOB	N	65535	UCS2	異なる	—
択一	DBCLOB	N	65535	UCS2	異なる	—
専用	DBCLOB	N	65535	UCS2	異なる	—
UTF8	BLOB	N	1208	65535	異なる	—
UTF8	CLOB	Y <—>	1208	文字	異なる	はい
UTF8	CLOB	Y <—>	1208	混用	異なる	はい
UTF8	DBLOB	Y <—>	1208	グラフィック	異なる	はい
UTF8	DBLOB	Y <—>	1208	UCS2	同じ	いいえ
UTF16	BLOB	N	1200	65535	異なる	—
UTF16	CLOB	Y <—>	1200	文字	異なる	はい
UTF16	CLOB	Y <—>	1200	混用	異なる	はい
UTF16	DBCLOB	Y <—>	1200	グラフィック	異なる	はい
UTF16	DBCLOB	Y <—>	1200	UCS2	同じ	いいえ
2 進文字	BLOB	Y <—>	65535	65535	同じ	いいえ
2 進文字	CLOB	Y <—>	65535	文字	異なる	いいえ
2 進文字	CLOB	Y <—>	65535	混用	異なる	いいえ
2 進文字	DBCLOB	N	65535	グラフィック	異なる	—
2 進文字	DBCLOB	N	65535	UCS2	異なる	—
2 進文字	UTF-8	N	65535	1208	異なる	—
2 進文字	UTF-16	N	65535	1200	異なる	—

3 番目の表は、可変長データ・タイプとラージ・オブジェクトの間のマッピングを示しています。

表 11. 可変長データ・タイプとラージ・オブジェクトの間でのコピー元ファイルおよびコピー先ファイルのマッピング

フィールド A のタイプ	フィールド B のタイプ	許可されているかどうか、およびコピー方向	データの CCSID または属性		CCSID	変換が実行される
			フィールド A	フィールド B		
VARLEN 文字	BLOB	Y <—>	文字	65535	異なる	いいえ
VARLEN 混用	BLOB	Y <—>	混用	65535	異なる	いいえ
VARLEN 択一	BLOB	Y <—>	択一	65535	異なる	いいえ

表 11. 可変長データ・タイプとラージ・オブジェクトの間のコピー元ファイルおよびコピー先ファイルのマッピング
(続き)

フィールド A の タイプ	フィールド B のタイプ	許可されている かどうか、およ びコピー方向	データの CCSID または属性		CCSID	変換が実行さ れる
			フィールド A	フィールド B		
VARLEN 専用	BLOB	Y <—>	専用	65535	異なる	いいえ
VARLEN グラフ ィック	BLOB	N	グラフィック	65535	異なる	—
VARLEN UCS2	BLOB	N	UCS2	65535	異なる	—
VARLEN 文字	CLOB	Y <—>	文字	文字	同じ/異なる	いいえ/はい
VARLEN 混用	CLOB	Y <—>	混用	文字	異なる	はい
VARLEN 択一	CLOB	Y <—>	択一	文字	異なる	はい
VARLEN 専用	CLOB	Y <—>	専用	文字	異なる	はい
VARLEN グラフ ィック	CLOB	N	グラフィック	文字	異なる	—
VARLEN UCS2	CLOB	Y <—>	UCS2	文字	異なる	はい
VARLEN 文字	CLOB	Y <—>	文字	混用	異なる	はい
VARLEN 混用	CLOB	Y <—>	混用	混用	同じ/異なる	いいえ/はい
VARLEN 択一	CLOB	Y <—>	択一	混用	異なる	はい
VARLEN 専用	CLOB	Y <—>	専用	混用	異なる	はい
VARLEN グラフ ィック	CLOB	Y <—>	グラフィック	混用	異なる	はい
VARLEN UCS2	CLOB	Y <—>	UCS2	混用	異なる	はい
VARLEN 文字	DBCLOB	N	文字	グラフィッ ク	異なる	—
VARLEN 混用	DBCLOB	Y <—>	混用	グラフィッ ク	異なる	はい
VARLEN 択一	DBCLOB	Y <—>	択一	グラフィッ ク	異なる	はい
VARLEN 専用	DBCLOB	Y <—>	専用	グラフィッ ク	異なる	はい
VARLEN グラフ ィック	DBCLOB	Y <—>	グラフィック	グラフィッ ク	同じ/異なる	いいえ/はい
VARLEN UCS2	DBCLOB	Y <—>	UCS2	グラフィッ ク	異なる	はい
VARLEN 文字	DBCLOB	Y <—>	65535 でない	UCS2	異なる	はい
VARLEN 混用	DBCLOB	Y <—>	65535 でない	UCS2	異なる	はい
VARLEN 択一	DBCLOB	Y <—>	65535 でない	UCS2	異なる	はい
VARLEN 専用	DBCLOB	Y <—>	65535 でない	UCS2	異なる	はい
VARLEN グラフ ィック	DBCLOB	Y <—>	グラフィック	UCS2	異なる	はい
VARLEN UCS2	DBCLOB	Y <—>	UCS2	UCS2	同じ/異なる	いいえ/はい
VARLEN 文字	DBCLOB	N	65535	UCS2	異なる	—
VARLEN 混用	DBCLOB	N	65535	UCS2	異なる	—

表 11. 可変長データ・タイプとラージ・オブジェクトの間のコピー元ファイルおよびコピー先ファイルのマッピング
(続き)

フィールド A の タイプ	フィールド B のタイプ	許可されている かどうか、およ びコピー方向	データの CCSID または属性		CCSID	変換が実行さ れる
			フィールド A	フィールド B		
VARLEN 択一	DBCLOB	N	65535	UCS2	異なる	-
VARLEN 専用	DBCLOB	N	65535	UCS2	異なる	-
VARLEN UTF8	BLOB	N	1208	65535	異なる	-
VARLEN UTF8	CLOB	Y <—>	1208	混用	異なる	はい
VARLEN UTF8	DBCLOB	Y <—>	1208	グラフィッ ク	異なる	はい
VARLEN UTF8	DBCLOB	Y <—>	1208	UCS2	異なる/同じ	はい/いいえ
VARLEN UTF16	BLOB	N	1200	65535	異なる	-
VARLEN UTF16	CLOB	Y <—>	1200	混用	異なる	はい
VARLEN UTF16	DBCLOB	Y <—>	1200	グラフィッ ク	異なる	はい
VARLEN UTF16	DBCLOB	Y <—>	1200	UCS2	異なる/同じ	はい/いいえ
VARLEN 2 進文 字	BLOB	Y <—>	65535	65535	同じ	いいえ
VARLEN 2 進文 字	CLOB	Y <—>	65535	文字	異なる	いいえ
VARLEN 2 進文 字	CLOB	Y <—>	65535	混用	異なる	いいえ
VARLEN 2 進文 字	DBCLOB	N	65535	グラフィッ ク	異なる	-
VARLEN 2 進文 字	DBCLOB	N	65535	UCS2	異なる	-
VARLEN 2 進文 字	UTF-8	N	65535	1208	異なる	-
VARLEN 2 進文 字	UTF-16	N	65535	1200	異なる	-

関連情報

CPYF コマンド

CPYFRMQRYP コマンド

識別列または ROWID 属性を含むファイルのコピー:

識別列または ROWID を含むファイルをコピーする場合、CPYF および CPYFRMQRYP コマンドに CRTFILE(*YES) を指定できます。

コピー元ファイルが識別列または ROWID を含む SQL 表、ビュー、または索引である場合、これらのコマンドは SQL 表を作成します。

識別列または ROWID を含むファイルにコピーする場合、値を提供するか、またはサーバーがフィールドに値を生成するようにできます。

バージョン 5 リリース 1 以前の iSeries サーバーには、識別列または ROWID を含む DDM ファイルをコピーすることはできません。

関連情報

CPYF コマンド

CPYFRMQRYP コマンド

異なるサーバー間でのコピー

このトピックで説明する CPYFRMIMPF および CPYTOIMPF コマンドを使用して、iSeries との間でデータをインポート (ロード) またはエクスポート (アンロード) することができます。

インポート・ファイルからのコピー (CPYFRMIMPF) コマンドによる、異なるサーバー間でのコピー:

インポート・ファイルからのコピー (CPYFRMIMPF) コマンド は、インポート・ファイルのデータをコピー先ファイルにマップまたは解析 (インポート) します。

インポート・ファイルのファイル・タイプによって、CPYFRMIMPF の実行時には異なる手順を使用します。

CPYFRMIMPF コマンドは、コピー中に複数のジョブを使用してインポート・ファイルからコピー先ファイルに情報をコピーする並列データ・ローダーもサポートしています。複数のジョブを使用するには、サーバーに対称マルチプロセッシング製品 (SMP) が導入されていなければなりません。

関連概念

『CPYFRMIMPF コマンドに関する注』

CPYFRMIMPF コマンドを使用してコピーを実行するのに必要な権限は、その他のすべてのコピーに必要な権限に類似しています。

74 ページの『CPYFRMIMPF コマンドに関する制約事項』

CPYFRMIMPF コマンドに適用される制約事項がいくつかあります。

77 ページの『CPYFRMIMPF コマンドで使用できる並列データ・ローダー・サポート』

インポート・ファイルからのコピー (CPYFRMIMPF) は、コピー中に複数のジョブを使用して、インポート・ファイルからコピー先ファイルへのデータの並列コピーをサポートします。これにより、他のプラットフォームからコピー先ファイルに迅速かつ容易にコピーを行うことができます。これは、特に、データウェアハウスを使用するユーザーにとって役立ちます。複数のジョブを使用するには、サーバーに対称マルチプロセッシング製品 (SMP) が導入されていなければなりません。

関連タスク

76 ページの『インポート・ファイルがストリーム・ファイルである場合の、iSeries へのデータのインポート』

インポート・ファイルからのコピー (CPYFRMIMPF) コマンドを使用して、ストリーム・ファイルからデータをインポートすることができます。

75 ページの『コピー元ファイルがデータベース・ファイルまたは DDM ファイルである場合の、iSeries へのデータのインポート』

インポート・ファイルからのコピー (CPYFRMIMPF) コマンドを使用して、データベース・ファイルまたは DDM ファイルからデータをインポートすることができます。

CPYFRMIMPF コマンドに関する注:

CPYFRMIMPF コマンドを使用してコピーを実行するのに必要な権限は、その他のすべてのコピーに必要な権限に類似しています。

コピー元ファイルには、以下のタイプのいずれかを使用できます。

- ストリーム・ファイル
- DDM ファイル
- テープ・ファイル
- ソース物理ファイル
- 分散物理ファイル
- プログラムで記述された物理ファイル
- 単一様式の論理ファイル
- 1 つのフィールドを持つ外部記述された物理ファイル。このフィールドは、数値データ・タイプ以外でなければなりません。

コピー先ファイルには、以下のタイプのいずれかを使用できます。

- ソース・ファイル
- DDM ファイル
- 分散物理ファイル
- プログラムで記述された物理ファイル
- 外部記述された物理ファイル

フィールド定義ファイルには、以下のタイプのいずれかを使用できます。

- ソース物理ファイル
- DDM ファイル
- プログラムで記述された物理ファイル
- 1 つのフィールドを持つ外部記述された物理ファイル

エラー・ファイルには、以下のタイプのいずれかを使用できます。

- ソース物理ファイル
- DDM ファイル
- プログラムで記述された物理ファイル
- 1 つのフィールドを持つ外部記述された物理ファイル

注: エラー・ファイルとコピー元ファイルの様式は同じでなければなりません。

関連概念

73 ページの『インポート・ファイルからのコピー (CPYFRMIMPF) コマンドによる、異なるサーバー間でのコピー』

インポート・ファイルからのコピー (CPYFRMIMPF) コマンド は、インポート・ファイルのデータをコピー先ファイルにマップまたは解析 (インポート) します。

関連情報

CPYFRMIMPF コマンド

CPYFRMIMPF コマンドに関する制約事項:

CPYFRMIMPF コマンドに適用される制約事項がいくつかあります。

- コピー元ファイルのデータ・タイプは、以下の 2 つのタイプのいずれかでなければなりません。
 - ソース物理ファイル

- データ・タイプが CHARACTER、IGC OPEN、IGC EITHER、IGC ONLY、GRAPHIC の、固定長または可変長である 1 つのフィールドを持つ物理ファイル
- コピーされるレコードの相対レコード番号は、コピー先ファイル内とコピー元ファイル内で同じになる可能性があります。
- コピーの前にコピー先ファイルを作成してください。
- RCDDEL *ALL は、CRLF、LF、CR、LF のいずれかが最初に現れる位置を検出します。プログラムの残りの部分では、この値が RCDDEL として使用されます。
- コピー先ファイルおよびコピー元ファイルに同じファイルを使用することはできません。
 - コピー元ファイルからレコードをインポートできない場合、エラー許可 (ERRLVL) パラメーターに基づいて処理は継続されます。ERRRCDFILE パラメーターを使用する場合に、ERRLVL(*NOMAX) をデフォルトとして想定するか、実際に指定すると、間違えたエラー・レコードが ERRRCDFILE に書き込まれることがあります。これには 2 つの理由があります。ERRLVL(*NOMAX) によってレコードがブロックされているか、要求を実行するためにコピー操作が複数のタスクを使用している可能性があります。エラー・レコード・ファイルに対する間違えたレコードの書き込みを防止するには、2 つの推奨方法があります。つまり、データを正しくコピーできることを確認した場合にのみ ERRLVL(*NOMAX) パラメーターを使用するか、*NOMAX の代わりに数値を指定するか、です。数値を指定すると、ERRLVL の数を超えるか、ファイルの終わりに達するまで、すべてのエラー・レコードが ERRRCDFILE に書き込まれます。

注: CPYFRMIMPF でファイル全体を処理するには、十分に大きい数値を指定する必要があることに注意してください。

- コピー元ファイルがソース・ファイルである場合、レコードの最初の 12 バイト (順序フィールドおよび日付フィールド) はコピーされません。コピー先ファイルがソース・ファイルである場合、コピー先ファイルのデータの最初の 12 バイト (順序フィールドおよび日付フィールド) はゼロに設定されます。

このコマンドは、ユーザー定義タイプ (UDT)、ユーザー定義関数 (UDF)、識別列、ROWID、およびラージ・オブジェクト (LOB) を含むファイルで使用することができます。データ・リンク (DL) を含むファイルでは使用できません。

65535 という JOBCCSID で CPYFRMIMPF コマンドを使用すると、不正確な結果または予測不能な結果になることがあります。

関連概念

73 ページの『インポート・ファイルからのコピー (CPYFRMIMPF) コマンドによる、異なるサーバー間でのコピー』

インポート・ファイルからのコピー (CPYFRMIMPF) コマンド は、インポート・ファイルのデータをコピー先ファイルにマップまたは解析 (インポート) します。

関連情報

CPYFRMIMPF コマンド

コピー元ファイルがデータベース・ファイルまたは DDM ファイルである場合の、iSeries へのデータのインポート:

インポート・ファイルからのコピー (CPYFRMIMPF) コマンドを使用して、データベース・ファイルまたは DDM ファイルからデータをインポートすることができます。

データベース・ファイルまたは DDM ファイルからデータをインポートするには、以下の手順に従ってください。

1. DB2 for iSeries 用の外部記述ファイルにコピーする予定のデータ用のインポート・ファイルを作成します。インポート・ファイルには、データベース・ソース・ファイル、1つのフィールドを持つ外部記述データベース・ファイル、またはプログラム記述の物理ファイルを使用できます。ファイルに1つのフィールドがある場合、そのデータ・タイプは、CHARACTER、IGC OPEN、IGC EITHER、IGC ONLY、GRAPHIC の、固定長または可変長でなければなりません。インポート・ファイルのレコード長は、区切り文字も含めて、サーバーに送信されるファイルの最長レコードを含むのに十分な長さでなければなりません。
2. インポート・ファイルまたはコピー元ファイルにデータを送信します。インポート・ファイルにデータを送信すると、ASCII と EBCDIC 間で必要なデータ変換が行われます。データをインポートするには、複数の方法があります。
 - TCP/IP ファイル転送 (テキスト転送)
 - CA/400 サポート (ファイル転送、ODBC)
 - CPYFRMTAP コマンド (テープ・ファイルからのコピー)
3. DB2 for iSeries 用の外部記述データベース・ファイルまたは DDM ファイルを作成します。このファイルには、インポート・ファイルの結果データが入ります。
4. CPYFRMIMPF コマンドを使用して、インポート・ファイルからコピー先ファイルへのコピー (レコードの変換または解析) を行います。大きいファイルをインポートする場合は、インポート・ファイルを複数の部分に分けて、それぞれの部分を N ウェイ・マルチプロセッサ・サーバーで並列に処理することができます。

関連概念

73 ページの『インポート・ファイルからのコピー (CPYFRMIMPF) コマンドによる、異なるサーバー間でのコピー』

インポート・ファイルからのコピー (CPYFRMIMPF) コマンドは、インポート・ファイルのデータをコピー先ファイルにマップまたは解析 (インポート) します。

77 ページの『CPYFRMIMPF コマンドで使用できる並列データ・ローダー・サポート』

インポート・ファイルからのコピー (CPYFRMIMPF) は、コピー中に複数のジョブを使用して、インポート・ファイルからコピー先ファイルへのデータの並列コピーをサポートします。これにより、他のプラットフォームからコピー先ファイルに迅速かつ容易にコピーを行うことができます。これは、特に、データウェアハウスを使用するユーザーにとって役立ちます。複数のジョブを使用するには、サーバーに対称マルチプロセッシング製品 (SMP) が導入されていなければなりません。

インポート・ファイルがストリーム・ファイルである場合の、iSeries へのデータのインポート:

インポート・ファイルからのコピー (CPYFRMIMPF) コマンドを使用して、ストリーム・ファイルからデータをインポートすることができます。

インポート・ファイルがストリーム・ファイルである場合、iSeries にデータをインポートするには、以下の手順に従ってください。

1. DB2 for iSeries 用の外部記述データベース・ファイルまたは DDM ファイルを作成します。このファイルには、インポート・ファイルの結果データが入ります。
2. CPYFRMIMPF コマンドを使用して、インポート・ファイルからコピー先ファイルへのコピー (レコードの変換または解析) を行います。大きいファイルをインポートする場合は、インポート・ファイルを複数の部分に分けることができます。分けられた複数の部分は、並列処理されます。

注: コピー元ストリーム・ファイル (FROMSTMF) パラメーターが指定されている場合、レコードは並列的にはコピーされません。

関連概念

73 ページの『インポート・ファイルからのコピー (CPYFRMIMPF) コマンドによる、異なるサーバー間でのコピー』

インポート・ファイルからのコピー (CPYFRMIMPF) コマンド は、インポート・ファイルのデータをコピー先ファイルにマップまたは解析 (インポート) します。

CPYFRMIMPF コマンドで使用できる並列データ・ローダー・サポート:

インポート・ファイルからのコピー (CPYFRMIMPF) は、コピー中に複数のジョブを使用して、インポート・ファイルからコピー先ファイルへのデータの並列コピーをサポートします。これにより、他のプラットフォームからコピー先ファイルに迅速かつ容易にコピーを行うことができます。これは、特に、データウェアハウスを使用するユーザーにとって役立ちます。複数のジョブを使用するには、サーバーに対称マルチプロセッシング製品 (SMP) が導入されていなければなりません。

コピーでのジョブの数は、Query 属性変更 (CHGQRYA) コマンドの DEGREE(*NBRTASKS) パラメーターによって決定されます。コピー元ファイル内のレコード数が 50,000 より少ない場合は、*NBRTASKS 値にかかわらずただ 1 つのジョブが使用されます。

CPYFRMIMPF コマンド (並列データ・ローダー・サポート付き) は、実質的に、インポート・ファイルをより小さい部分つまりブロックに分けます。小さく分けられた各部分は並列に実行依頼されるので、ファイル全体が同時に処理されます。(これにより、順次処理の待ち時間がなくなります。)

コピー元ファイルの相対レコード番号をコピー先ファイル内でもそのまま維持するには、コピーで 1 つのジョブのみを使用してください。この場合、DEGREE(*NONE) を指定してください。

- 1 注: コピー元ストリーム・ファイル (FROMSTMF) パラメーターが指定されている場合、レコードは並列的にはコピーされません。

関連概念

73 ページの『インポート・ファイルからのコピー (CPYFRMIMPF) コマンドによる、異なるサーバー間でのコピー』

インポート・ファイルからのコピー (CPYFRMIMPF) コマンド は、インポート・ファイルのデータをコピー先ファイルにマップまたは解析 (インポート) します。

関連タスク

75 ページの『コピー元ファイルがデータベース・ファイルまたは DDM ファイルである場合の、iSeries へのデータのインポート』

インポート・ファイルからのコピー (CPYFRMIMPF) コマンドを使用して、データベース・ファイルまたは DDM ファイルからデータをインポートすることができます。

インポート・ファイルからのデータの処理:

インポート・ファイルからのコピー (CPYFRMIMPF) は、インポート・ファイルからデータを読み取り、コピー先ファイルにデータをコピーします。インポート・ファイルのデータは、区切り文字によって、または固定様式として様式化できます。

78 ページの『区切り文字付きインポート・ファイルに関する注 (CPYFRMIMPF コマンド)』では、フィールドの始まりと終わりを定義する一連の文字 (区切り文字) について説明しています。コマンドのパラメーターは、どの文字を区切り文字として使用するかを定義します。

81 ページの『固定様式インポート・ファイル』では、インポート・ファイルの様式を定義するために、ユーザーが定義する必要のあるフィールド定義ファイルについて説明します。フィールド定義ファイルは、フィールドの始まりと終わり、および NULL であるかどうかを定義します。

関連概念

83 ページの『インポート・ファイルへのコピー (CPYTOIMPF) コマンドによる、異なるサーバー間でのコピー』

インポート・ファイルへのコピー (CPYTOIMPF) コマンドは、コピー元ファイル (通常はデータベース・ファイル) からインポート・ファイルにデータをコピーします。

関連情報

CPYFRMIMPF コマンド

区切り文字付きインポート・ファイルに関する注 (CPYFRMIMPF コマンド):

区切り文字付きインポート・ファイルの場合、インポート・ファイルのデータは、このトピックで説明される文字とデータ・タイプによって解釈されます。

ブランク

ブランクは、以下のように処理されます。

- 文字フィールドのすべての先行ブランクおよび末尾ブランクは、文字列区切り文字で囲まれていない限り、RMVBLANK パラメーターに基づいて廃棄されます。
- RMVBLANK *NONE を指定しない限り、ブランクのみで構成されるフィールドは、単一のブランク文字からなるフィールドとして解釈されます。
- 数字データ内にブランクを組み込むことはできません。
- ブランクをフィールドまたはレコードの区切り文字として選択することはできません。

RMVBLANK パラメーターには以下のオプションがあります。

- *NONE: すべての先行ブランクと末尾ブランクが保持されます。
- *LEADING: 先行ブランクが削除されます。これはデフォルト値です。
- *TRAILING: 末尾ブランクが削除されます。
- *BOTH: 先行ブランクと末尾ブランクが削除されます。

注: RMVBLANK パラメーターの指定に基づくブランクの削除は、文字列区切り文字よりも優先されます。

NULL フィールド

NULL フィールドは、以下のように定義されます。

- 2 つの隣接するフィールド区切り文字 (区切り文字の間にデータがない)。
- 2 つの隣接する文字列区切り文字 (区切り文字の間にデータがない)。
- 後にレコード区切り文字が続くフィールド区切り文字 (間にデータがない) 空文字列。

フィールドが NULL である場合、以下の説明が当てはまります。

- レコードの出力フィールドが NULL 可能でなく、インポートが NULL フィールドである場合、レコードはコピーされず、エラーが表示されます。

区切り文字

- フィールドまたはレコードの区切り文字にブランクを使用することはできません。
- 文字列区切り文字または文字列・エスケープ文字として、フィールド区切り文字、レコード区切り文字、小数点、日付区切り記号、または時刻区切り記号と同じ文字を使用することはできません。

- スtring・エスケープ文字 STRESCCHR(*NONE) が指定されている場合、String区切り文字ですべてのフィールドを囲むことができます。String区切り文字を文字String内に入れることはできません。
- フィールド区切り文字には、レコード区切り文字と同じ文字を使用することはできません。
- String区切り文字がStringに含まれる場合、String内のString区切り文字の前に、STRESCCHR パラメーターで指定されたString・エスケープ文字を使用してください。
- 区切り文字のデフォルトは、以下のとおりです。
 - String: *DBLQUOTE (二重引用符)
 - フィールド: , (コンマ)
 - 小数点: . (ピリオド)
 - レコード: *EOR (レコードの終わり)
- コピー元のデータ・タイプが CHARACTER、OPEN、EITHER、または ONLY である場合、すべての 2 バイト・データは、String区切り文字またはシフト文字の中に入っていなければなりません (OPEN、EITHER、 ONLY データ・タイプの場合)。

注: String区切り文字は、先行空白や末尾空白よりも優先順位が低くなっています。

String・エスケープ文字

String区切り文字がString内に含まれる場合、String区切り文字を (String区切り文字としてではなく) 通常の文字として扱うよう指示するために、STRESCCHR パラメーターを使用してString・エスケープ文字を指定できます。

STRESCCHR パラメーターには以下のオプションがあります。

- *NONE: String・エスケープ文字を使用しない。
- *STRDLM: String区切り文字をString・エスケープ文字として使用する。
- *character-value - character-value* をString・エスケープ文字として使用する。

数字フィールド

- 数字フィールドは、10 進様式または指数様式でインポートできます。
- 小数点の右側にあるデータは、出力データ様式によっては切り捨てられる場合があります。
- 小数点は、ピリオドまたはコンマのいずれかです (コマンド・オプション)。
- 符号付き数字フィールドがサポートされています (+ または -)。

文字または Varcharacter フィールド

- 出力フィールド内に入りきらないフィールドは切り捨てられます (右から)。サーバーは、診断メッセージを送信します。
- 空Stringは、区切り文字間にデータがない 2 つのString区切り文字として定義されます。
- ある文字を開始または終了のString区切り文字としてサーバーに認識させるには、RMVBLANK オプションの適用後にその文字がフィールド内の最初または最後の文字にならなければなりません。たとえば、区切り文字として ' が付いている 'abc' は、abc と同じです。

IGC または VarIGC フィールド

- データは、コピー元ファイルからコピー先ファイルにコピーされます。データのいずれかが無効である場合、マッピング・エラーが生成されます。

- シフトアウト文字とシフトイン文字との間にあるデータは、2 バイト・データとして処理されま
す。また、このデータは、区切り文字としては解析されません。この場合、シフト文字は "スト
リング区切り文字" になります。

グラフィック、VarGraphic フィールド

データは、コピー元ファイルからコピー先ファイルにコピーされます。

CCSID

- コピー元ファイルのデータは、コピー元ファイルの CCSID によってバッファーに読み取られま
す。バッファー内のデータは検査され、コピー先ファイルに書き込まれます。コピー先ファイル
の CCSID が使用されない限り、オープン・コピー先ファイルの CCSID は、コピー元ファイル
の値に設定されます。

コピー先ファイル CCSID が使用されると、データはその CCSID に変換されます。コピー先フ
ィールドの CCSID が 65535 であれば、コピー先 CCSID (TOCCSID) パラメーター
で指定された CCSID にデータが変換されます。

コピー元ファイルがテープ・ファイルであり、FROMCCSID(*FILE) が指定されている場合、以
下のいずれかの制限が適用されます。

- ジョブ CCSID が使用されます。
- コピー元ファイル CCSID がユーザーによって要求されます。
- コマンドで渡された文字データ (区切り文字) は、コピー元ファイルの CCSID に変換されま
す。これにより、コピー元ファイル・パラメーターの文字データとコマンド・パラメーターの文
字データとの互換性が実現します。

日付フィールド

- iSeries によってサポートされているすべての日付様式をインポートすることができます (*ISO、
*USA、*EUR、*JIS、*MDY、*DMY、*YMD、*JUL、および *YYMD)。
- 日付フィールドをタイム・スタンプ・フィールドにコピーすることができます。

時刻フィールド

- iSeries によってサポートされているすべての時刻様式をインポートすることができます (*ISO、
*USA、*EUR、*JIS、*HMS)。
- 時刻フィールドをタイム・スタンプ・フィールドにコピーすることができます。

日時区切り記号

日時フィールドで有効なすべての区切り記号がサポートされます。

タイム・スタンプ・フィールド

タイム・スタンプ・インポート・フィールドは、26 バイトでなければなりません。インポートで
は、時刻部分にピリオドがあり、タイム・スタンプの日付の部分と時刻の部分の間にダッシュがあ
ることが確認されます。

LOB フィールド

LOB データ・フィールドでは、LOB データを入れるための 2 次ストリーム・ファイルの使用が
必要になります。インポートされたデータ・ファイルには、LOB データが入っているストリー
ム・ファイルの名前がリストされます。データ・ファイル内の各フィールドのオフセットを記述し
たフィールド定義ファイル (FDF) には、LOB ファイル名フィールドのオフセット (LOB フィー
ルドそのものではない) も記述されます。

LOB データのインポート例:

FDF ファイル:

```
COLCHAR10      1      10      42
COLLOB         12      40      44
*END
```

データ・ファイルのレコード長は 44 文字になり、内容は以下のようなデータになります。

```
0      1      2      3      4
12345678901234567890123456789012345678901234
aaaaaaaaaa /lobdata/lob1.dat      N N
bbbbbb /lobdata/lob2.dat      N N
        /lobdata/lob3.dat      Y N
cccccccccc      N Y
```

注: インポートされたデータ・ファイルの各レコードは、以下のようになります。

1. 文字 1 から 10 は、COLCHAR10 という名前の列のデータであり、バイト位置 42 に NULL 標識が入ります。
2. 文字 12 から 40 は、COLLOB という名前の列のデータであり、バイト位置 44 に NULL 標識が入ります。COLLOB の値は、LOB データを含むファイルの名前です。
3. この例では、インポートする実際の LOB データを含んだデータ・ファイルが 3 つあります (lob1.dat、lob2.dat、lob3.dat)。

フィールド数のミスマッチ

コピー元ファイルとコピー先ファイルに同じ数のフィールドがない場合、データが切り捨てられてコピー先ファイル・サイズが小さくなるか、または余分のコピー先ファイル・フィールドに NULL 値が入ります。フィールドに NULL 値を入れることができない場合、エラー・メッセージが発行されます。

複数のジョブ

データをコピーするために使用されるジョブの数は、CHGQRYA コマンドの DEGREE(*NBRTASKS) パラメーターによって決まります。複数のジョブが使用される場合、サーバーは、データをコピーするためにバッチ・ジョブを使用します。ユーザーは、これらのバッチ・ジョブを変更、保留、または終了することができます。コピーは、開始したすべてのバッチ・ジョブが完了するまで完了しません。

相対レコード番号は、単一ジョブが使用され、インポート・ファイルに削除レコードが含まれていない場合にのみ維持されます。コピー元ファイルが分散物理ファイルまたは論理ファイルである場合、サーバーは、単一プロセスでコピーを実行します。

レコード数が 50,000 より少ないファイルは、1 つのジョブのみを使用します。

固定様式インポート・ファイル:

このトピックでは、固定様式ファイルを記述するフィールド定義ファイルの例を示します。

```
- *****/
- ****      Field Definition File      */
- *****/
- Description: This Field Definition File
- defines the import's file      */
- (FROMFILE) field start and end positions.      */
- *****/
- (FROMFILE) field start and end positions.      */
-FILE MYLIB/MYFILE      */
field1  1  12  13
field2  14  24  0
field3  25  55  56
field4  78  89  90
field5 100 109  0
field6 110 119 120
```

```
field7 121 221 0
*END
```

フィールド定義ファイル形式の簡単な説明を次に示します。

- = コメント行
*END = 定義の終了。これは必須です。

フィールド名	開始位置	終了位置	NULL 値位置
field1	1	12	13
field2	14	24	None
field3	25	55	56
field4	78	89	90
field5	100	109	None
field6	110	119	120
field7	121	221	None

フィールド名

この名前は、コピー先フィールドの名前です。

開始位置

これは、各レコードのインポート・ファイルでのフィールドの開始位置です。これは、バイト位置です。

終了位置

これは、各レコードのインポート・ファイルでのフィールドの終了位置です。これは、バイト位置です。

NULL 値位置

これは、各レコードのインポート・ファイルでのフィールドの NULL 値の位置です。値ゼロは、NULL が値を持たないことを指定します。インポート・ファイルでの値は、'Y' または 'N' です。

'Y' は、フィールドが NULL であることを意味しています。'N' は、フィールドが NULL ではないことを意味しています。

それぞれの列は、ブランク文字で区切らなければなりません。

各行は順番に配列する必要があります。

| フィールド定義ファイルを作成する代替的な方法として、実際の列名の代わりに *COL キーワードを使用
| することができます。 *COL は、順番にリストされたターゲット・ファイル内のすべての列に関するスト
| リーム・ファイル内のデータの位置を示します。以下の例は、*COL キーワードを使用して固定様式ファイ
| ルを記述する方法を示しています。

```
| *COL 1 12 13
| *COL 14 24 0
| *COL 25 55 56
| *COL 78 89 90
| *COL 100 109 0
| *COL 110 119 120
| *COL 121 221 0
| *END
```

| **CPYFRMIMPF コマンドのパフォーマンスを改善するためのヒント:**

CPYFRMIMPF コマンドのパフォーマンスを改善するには、以下の手順に従ってください。

1. TOFILE に関連する論理キー・ファイルを削除します。
2. TOFILE のすべての制約およびトリガーを除去します。
3. いくつかのレコードをコピーして、FROMFILE レコードが正しくコピーされることを確認します。いくつかのレコードをコピーするには、FROMRCDC およびレコード数オプションを使用します。
4. データが正しくコピーできたことを確認したら、ERRLVL(*NOMAX) パラメーターを使用します。
5. ERRLVL(*NOMAX) パラメーターを使用すると、レコード・ブロックによりパフォーマンスが向上します。レコードへの書き込みエラーがレコード・ブロック化中に発生する場合、完了メッセージ CPC2955 にリストされる、コピーされているレコードの数は正確でない可能性があります。

インポート・ファイルへのコピー (CPYTOIMPF) コマンドによる、異なるサーバー間でのコピー:

インポート・ファイルへのコピー (CPYTOIMPF) コマンドは、コピー元ファイル (通常はデータベース・ファイル) からインポート・ファイルにデータをコピーします。

この後で、インポート・ファイル (またはエクスポートされるファイル) を、ご使用のプラットフォームに移動することができます。これには、任意の方式 (TCP/IP ファイル転送 (テキスト転送)、CA/400 サポート (ファイル転送、ODBC)、または CPYTOTAP (テープ・ファイルへのコピー) コマンドなど) を使用できます。次に、サーバーは、2 つの方法のうちのいずれかでインポート・ファイルのデータを処理します。

さらに、ストリーム・ファイルを指定することもできます。このとき、CPYTOIMPF コマンドはストリーム・ファイルにデータをコピーします。

関連概念

77 ページの『インポート・ファイルからのデータの処理』

インポート・ファイルからのコピー (CPYFRMIMPF) は、インポート・ファイルからデータを読み取り、コピー先ファイルにデータをコピーします。インポート・ファイルのデータは、区切り文字によって、または固定様式として様式化できます。

関連資料

『CPYTOIMPF コマンドに関する注』

インポート・ファイルへのコピー (CPYTOIMPF) コマンドは、ユーザーのコピー元ファイルからデータを読み取り、そのデータをインポート・ファイルにコピーします。コピーに使用されるジョブの数は 1 つです。インポート・ファイルのデータは、区切り文字によって、または固定様式として様式化できます。

CPYTOIMPF コマンドに関する注:

インポート・ファイルへのコピー (CPYTOIMPF) コマンドは、ユーザーのコピー元ファイルからデータを読み取り、そのデータをインポート・ファイルにコピーします。コピーに使用されるジョブの数は 1 つです。インポート・ファイルのデータは、区切り文字によって、または固定様式として様式化できます。

85 ページの『区切り文字付きインポート・ファイルに関する注 (CPYTOIMPF コマンド)』には、フィールドの始まりと終わりを定義する一連の文字 (区切り文字) があります。詳しくは、84 ページの『CPYTOIMPF コマンドに関する制約事項』を参照してください。

コマンドのパラメーターは、どの文字を区切り文字として使用するかを定義します。固定様式のインポート・ファイルは、固定様式を使用します。詳細については、86 ページの『固定様式のインポート・ファイルへのデータのコピー (CPYTOIMPF コマンド)』を参照してください。

コピー元ファイル内のデータは、様式化されたデータベース・ファイルから読み取られ、コマンドのパラメーターによってインポート・ファイルに書き込まれます。

コピー元のライブラリーと同じ名前のユーザー・プロファイルが存在する場合は、少なくとも以下のいずれかの権限が必要です。

- その名前のユーザー・プロファイルに対するシステム権限 *ADD
- 管理権限

コピー元ファイルには、以下のいずれかを使用できます。

- ソース物理ファイル
- プログラムで記述された物理ファイル
- 分散物理ファイル
- 単一様式の論理ファイル
- 外部記述された物理ファイル

コピー先ファイルには、以下のいずれかを使用できます。

- ストリーム・ファイル
- ソース物理ファイル
- プログラムで記述された物理ファイル
- 1 つの非数字フィールドを持つ分散物理ファイル
- 1 つの非数字フィールドを持つ外部記述された物理ファイル

関連概念

83 ページの『インポート・ファイルへのコピー (CPYTOIMPF) コマンドによる、異なるサーバー間でのコピー』

インポート・ファイルへのコピー (CPYTOIMPF) コマンドは、コピー元ファイル (通常はデータベース・ファイル) からインポート・ファイルにデータをコピーします。

関連情報

CPYTOIMPF コマンド

CPYTOIMPF コマンドに関する制約事項:

CPYTOIMPF コマンドに適用される制約事項がいくつかあります。

- このコマンドでは、区切り文字の正しい使用法が制限されています。
- コピー先ファイル用のデータベース・ファイルのデータ・タイプには、以下のいずれかのタイプを使用できます。
 - CHARACTER、IGC OPEN、IGC EITHER、IGC ONLY、GRAPHIC、または可変長。データの長さは、コピー元ファイルのデータ、区切り記号、および任意のデータ変換を含むことのできる長さでなければなりません。
 - コピー先ファイルおよびコピー元ファイルに同じファイルを使用することはできません。
 - コピー元ファイルに複数様式論理ファイルを使用することはできません。
- コピー先ファイルのレコード長が十分な長さでないと、エラーが表示されます。
- IGCDTA は、CPYTOIMPF コマンドではサポートされていません。

このコマンドは、ユーザー定義タイプ (UDT) およびユーザー定義関数 (UDF) を含むファイルで使用することができます。ラージ・オブジェクト (LOB) またはデータ・リンク (DL) を含むファイルでは使用できません。

65535 という JOBCCSID で CPYTOIMPF コマンドを使用すると、不正確な結果または予測不能な結果になることがあります。

関連情報

CPYTOIMPF コマンド

区切り文字付きインポート・ファイルに関する注 (CPYTOIMPF コマンド):

このトピックでは、CPYTOIMPF コマンドを使用して区切り文字付きインポート・ファイルを処理する場合について説明します。

| ブランク

| RMVBLANK パラメーターを使用して、文字フィールドのブランク文字を処理することができます。RMVBLANK パラメーターには以下のオプションがあります。

- | • *NONE: すべての先行ブランクと末尾ブランクが保持されます。
- | • *LEADING: 先行ブランクが削除されます。これはデフォルト値です。
- | • *TRAILING: 末尾ブランクが削除されます。
- | • *BOTH: 先行ブランクと末尾ブランクが削除されます。

| 注: RMVBLANK パラメーターの指定に基づくブランクの削除は、ストリング区切り文字よりも優先されます。

NULL フィールド

フィールドが NULL である場合、フィールドには、2 つの隣接するフィールド区切り文字 (2 つの隣接フィールドの間にデータがない) が入っています。

区切り文字

- | • 区切り文字にブランクを使用することはできません。
- | • ストリング区切り文字またはストリング・エスケープ文字として、フィールド区切り文字、レコード区切り文字、小数点、日付区切り記号、または時刻区切り記号と同じ文字を使用することはできません。
- | • フィールド区切り文字には、レコード区切り文字と同じ文字を使用することはできません。
- | • ストリング区切り文字がストリングに含まれる場合、ストリング内のストリング区切り文字の前に、STRESCCHR パラメーターで指定されたストリング・エスケープ文字を使用してください。
- | • 区切り文字のデフォルトは、以下のとおりです。
 - | – ストリング: *DBLQUOTE (二重引用符)
 - | – フィールド: , (コンマ)
 - | – 小数点: . (ピリオド)
 - | – レコード: *EOR (レコードの終わり)

| ストリング・エスケープ文字

| ストリング区切り文字がストリング内に含まれる場合、ストリング区切り文字を (ストリング区切り文字としてではなく) 通常の文字として扱うよう指示するために、STRESCCHR パラメーターを使用してストリング・エスケープ文字を指定できます。

| STRESCCHR パラメーターには以下のオプションがあります。

- *NONE: ストリング・エスケープ文字を使用しない。
- *STRDLM: ストリング区切り文字をストリング・エスケープ文字として使用する。
- *character-value*: *character-value* をストリング・エスケープ文字として使用する。

数字フィールド

小数点は、ピリオドまたはコンマのいずれかです (コマンド・オプション)。

グラフィックス・フィールド

ストリング区切り文字は、すべてのグラフィック・データの前後に入れられます。グラフィック・データがファイル内に含まれており、ストリング区切り文字が *NONE 値である場合、エラーが出されます。

すべてのフィールド

SQL の CAST 機能は、コピー元ファイルからコピー先ファイルにデータをコピーします。コピー元ファイルに削除済みレコードが含まれていない限り、すべてのデータがコピーされ、コピー元ファイルとコピー先ファイルの相対レコード番号は同じになります。削除されたレコードはコピーされません。

CCSID

コピー元ファイルからのデータは、コピー先の CCSID に読み取られます。

- コピー元ファイル・フィールドの CCSID が 65535 であれば、FROMCCSID パラメーターで指定された CCSID からデータが変換されます。

日付フィールド

iSeries によってサポートされているすべての日付様式をエクスポートすることができます (*ISO、*USA、*EUR、*JIS、*MDY、*DMY、*YMD、*JUL、および *YYMD)。

時刻フィールド

iSeries によってサポートされているすべての時刻様式をエクスポートすることができます (*ISO、*USA、*EUR、*JIS、*HMS)。

日時区切り記号

日時フィールドで有効なすべての区切り記号がサポートされます。

タイム・スタンプ・フィールド

タイム・スタンプ・エクスポート・フィールドは、26 バイトでなければなりません。

固定様式のインポート・ファイルへのデータのコピー (CPYTOIMPF コマンド):

固定様式 (DATFMT(*FIXED)) でインポート・ファイルにデータをコピーすると、ファイルの各フィールドがコピーされます。コマンド NULLS(*YES) 上にヌル標識があると、コピー先ファイル内のフィールド・データの後に、フィールドが NULL であるかどうかを示す 'Y' または 'N' が入ります。

ファイルのオープン

アプリケーション・プログラムでファイルを使用したい場合は、名前でそのファイルを参照します。そこで、そのファイルに関するファイル記述が、プログラムとサーバーとの対話の方法を制御します。

アプリケーション・プログラムでどのようにファイル記述を使用するかに関しては、2 つのオプションがあります。

- 現在のファイル記述をそのまま使用する。この場合は、サーバーでは記述にまったく変更を加えず、そのまま使用します。

- ファイル記述に関連したパラメーターをいくつか、またはすべて変更する。ファイル記述に加える変更は、永続的なものでも、一時的なものでも構いません。永続的な変更に関する情報は、ご使用の装置に対応する資料を参照してください。

関連概念

3 ページの『ファイル・タイプ』

オープンされたファイルの有効範囲

ユーザー・デフォルト活動化グループ内でオープンされたファイルの有効範囲は、呼び出しプログラムの呼び出しレベル番号に限定されます (デフォルト)。呼び出しレベル番号は、サーバーが各呼び出しスタックに割り当てた固有の番号です。名前付き活動化グループ内でオープンされたファイルの有効範囲は、活動化グループ・レベルに限定されています (デフォルト)。

オープン操作の有効範囲は、オーバーライド・コマンドを使用して変更することができます。たとえば、オープン操作の有効範囲を、ジョブ・レベルに変更することができます。

関連概念

97 ページの『オーバーライドの使用』

オーバーライドを使用して、ファイル名やファイルに関連した装置名などのファイル属性を一時的に変更できます。

91 ページの『オープンされたファイルに関する情報の表示』

オープンされたファイルについての情報を表示するには、任意のコマンド行で `dspjob option(*opnf)` または `wrkjob option(*opnf)` と入力して Enter キーを押します。

一時ファイル記述を使用したファイルのオープン

オーバーライドを使用すると、アプリケーションの柔軟性が大幅に向上します。サーバーがオーバーライドを行なうのは、プログラムがファイルをオープンすることによって、ファイルに至るパスを最初に確立する時です。

オーバーライドは、次の 2 つの方法のいずれかで行うことができます。

- プログラム自体の中で指定され、オープン操作でパラメーターとして渡される情報による方法。
- アプリケーション・プログラムの実行時環境をセットアップするのに使用される入力ストリームの中でオーバーライド CL コマンドを使用する方法。

最初の方法を使用できるかどうかは、主にプログラムの作成に使用したプログラミング言語によります。プログラミング言語によっては、ユーザーにオープン処理に対する管理を許さないものがあります。そのようなプログラミング言語の場合、オープン処理はほぼ自動的に実行され、どの情報が渡されるかの制御が行われます。他のプログラミング言語の場合は、ユーザーがオープン処理をかなり制御することができます。

2 番目のオプションについては、使用するプログラミング言語には関係なく使用することができます。

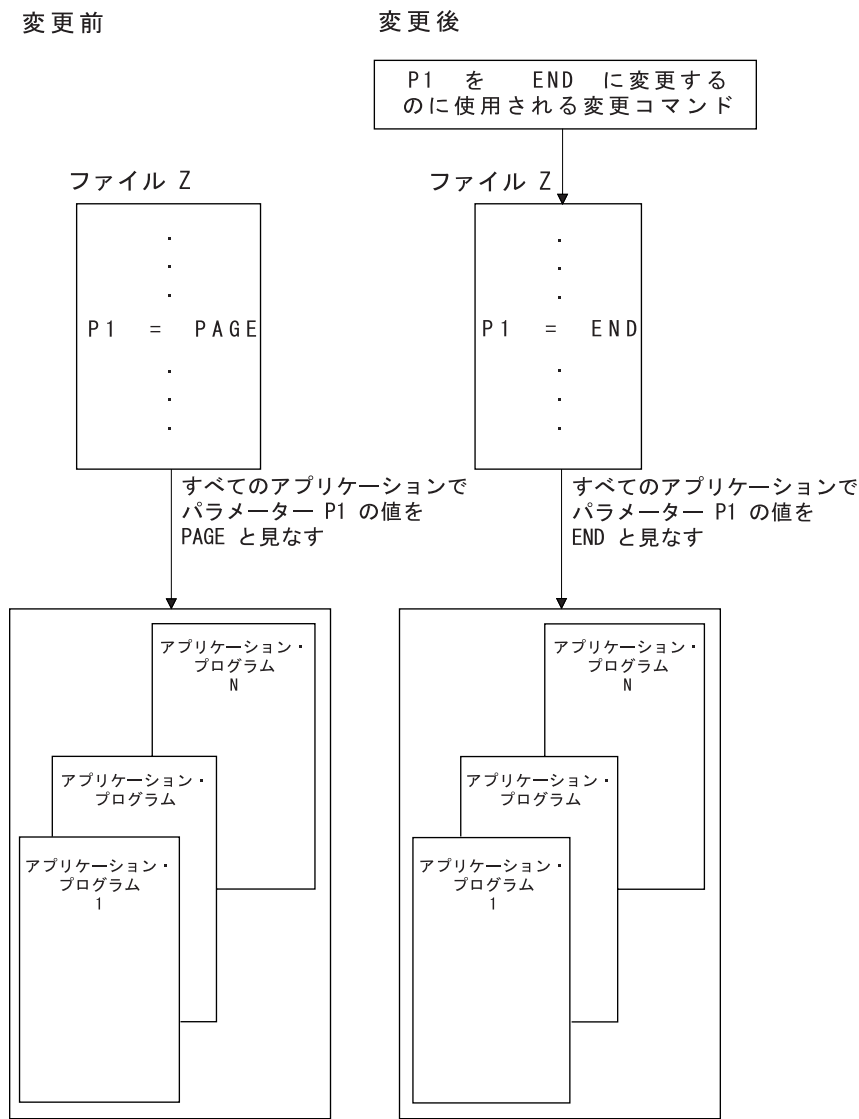
iSeries サーバーには、それぞれのファイル・タイプ用のオーバーライド CL コマンドがあります。アプリケーション・プログラムにオーバーライド・コマンドを組み込むことによって、プログラムで使用したいファイルの中のファイル記述を一時的に変更することができます。

これら 2 つのオプションは一緒に使用することができます。アプリケーションに含まれている情報でいくつかのパラメーターを変更し、オーバーライド・コマンドでその他の変更を行うことができます。両方のオプションは同じパラメーターを変更することができます。オペレーティング・システムは、次の順序に従ってファイルに一時的変更を行ないます。

1. ファイル記述から基本情報が提供されます。

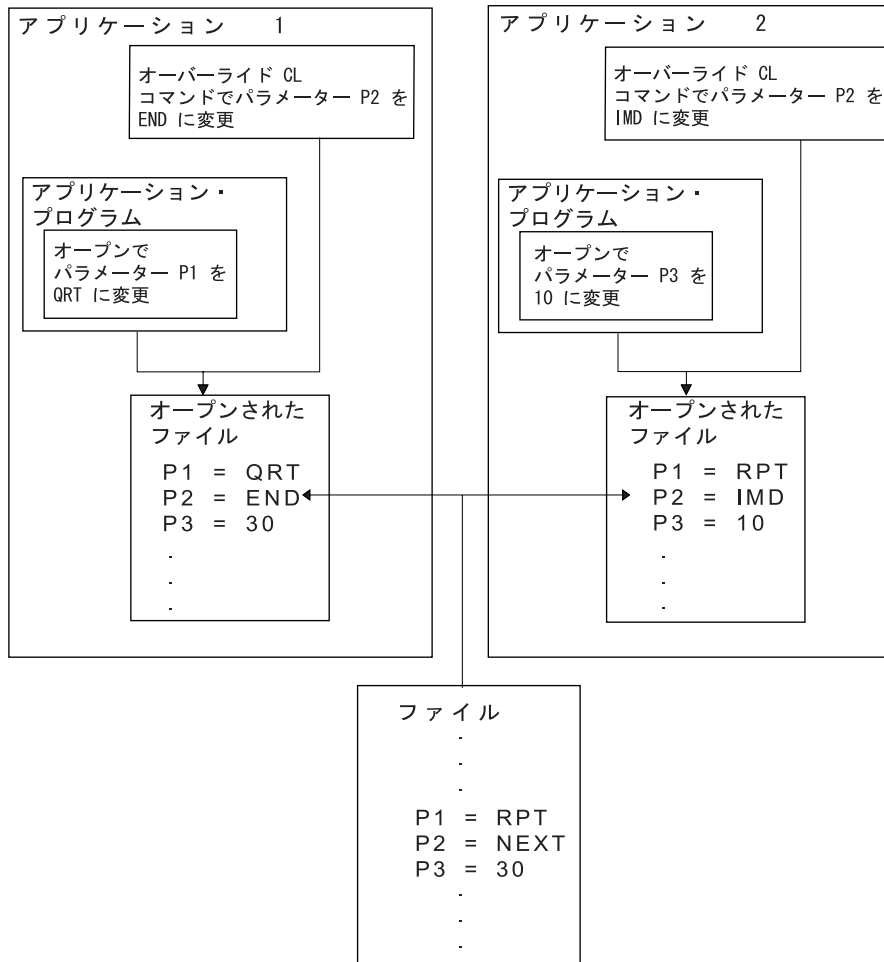
2. オープン処理時にアプリケーション・プログラムから受け取った変更情報が、基本情報に最初に適用されます。
3. オーバーライド・コマンドに含まれる変更情報は最後に適用されます。アプリケーションとオーバーライドの両方に、同じ情報を変更する変更情報がある場合は、オーバーライドの変更情報が優先されます。

オーバーライドを見ることができるのは、変更を加えたアプリケーションだけです。別のアプリケーション・プログラムから見た場合、ファイルは変更されていません。実際、2つのアプリケーション・プログラムが同じファイルを同時に使用し、それぞれが必要に応じてファイルを変更することもあり得ます。これらのアプリケーション・プログラムのどちらも、もう一方のアプリケーション・プログラムが一時変更を行っていることは認識しません。図1と89ページの図2は、永続変更および一時変更のプロセスを示しています。



RSLH143-2

図1. ファイルの永続的変更



RSLH196-1

図2. ファイルの一時的変更

アプリケーション・プログラムでは、ファイルをオープンすることによってアプリケーション・プログラムとファイルの間にいったん接続を確立してしまえば、入力操作または出力操作のいずれでもファイルの使用を行うことができます。データベース・ファイルの場合は、オープン処理によってアプリケーション・プログラムと実際のデータベース・ファイルの間にパスを確立します。装置ファイルの場合は、パスはアプリケーション・プログラムと実際の装置の間に確立されるか、あるいはスプーリング属性がその装置ファイルに対して活動化されている場合は、スプール・ファイルに至るパスが確立されます。すべての場合に、アプリケーション・プログラムは使用したい対象に接続し、この接続によって有効な入力操作または出力操作が決まります。どのファイル・タイプに対してもすべての操作が有効であるとは限りません。アプリケーション・プログラムでは、どのファイル・タイプを使用するのかを知り、そのファイル・タイプで有効な操作だけを使用しなければなりません。

DDM ファイルで *LIBL を使用する場合のオープンに関する考慮事項

ライブラリーに *LIBL を指定して DDM ファイルをオープンする場合、いくつかの考慮事項があります。

- サーバーは、まずライブラリー・リストで、指定されたメンバーを持つローカル・データベース・ファイルを検索します。ライブラリー・リストの中で、ローカル・データベース・ファイルより前に DDM ファイルを含むライブラリーがあったとしても、サーバーは、指定されたメンバーを持つローカル・データベース・ファイルを使用します。

そのため、*LIBL を使用する DDM ファイルをオープンしたい場合には、ライブラリー・リストのどこにも、同じ名前のローカル・データベース・ファイルや、指定したメンバーを含むローカル・データベース・ファイルがないことを確認する必要があります。

- 指定されたメンバーを含むローカル・データベース・メンバーが見つからない場合、サーバーは、ライブラリー・リストで、指定された名前を持つ最初のファイルを検索します。検出されたファイルが正しいタイプのものでなかった場合、または指定されたメンバーを含んでいない場合は、オープンが失敗します。

そのため、*LIBL を使用して DDM ファイルをオープンしたい場合には、ライブラリー・リスト内で、オープンしたい DDM ファイルが指定した名前を持つ最初のファイルになるようにしなければなりません。

ファイル記述変更の検出

外部記述ファイルを使用するプログラムをコンパイルするときは、高水準言語コンパイラーは、プログラムの中で指定されているファイルに関するレコード・レベル記述およびフィールド・レベル記述を抽出し、それらの記述をコンパイルされたプログラムの一部にします。プログラムを実行するときに、プログラムをコンパイルしたときの記述と現在の記述が同じかどうかを、確認検査することができます。

サーバーは対応するファイルを作成する際、各レコード様式ごとに固有のレベル識別コードを割り当てます。サーバーは、以下の情報を使用してレベル識別コードを決定します。

- レコード様式名
- フィールド名
- レコード様式の全長
- レコード様式内のフィールドの数
- フィールド属性 (たとえば、長さおよび小数点以下の桁数)
- レコード様式内でのフィールドの順番

注: 大規模レコード様式 (多くのフィールド) を持つファイルは、様式が多少異なっても、同じ様式のレベル識別コードを持つ場合があります。コピー元ファイルとコピー先ファイルのレコード様式名が同一の場合、これらのファイルをコピーしている時に問題が発生する可能性があります。

ディスプレイ・ファイル、プリンター・ファイル、および ICF ファイルでも標識と呼ばれる特殊フィールドの数および順序を使用して、レベル識別コードを決めることがあります。

レコード様式に関する DDS を変更し、前述のリストの中の項目のいずれかを変更した場合は、レベル識別コードが変更されます。

プログラムの実行時に、レコード様式識別コードを検査するには、ファイル作成コマンドまたはファイル変更コマンドで LVLCHK(*YES) を指定してください。

ファイルがオープンされて LVLCHK(*YES) が指定されると、オープンされたファイルのレベル識別コードとコンパイルされたプログラムの一部を成すファイル記述が比較されます。サーバーは、レベル識別コードの様式ごとの比較を行います。識別コードが異なる場合、またはプログラムの中で指定されている様式のいずれかが、ファイルの中に存在しない場合は、状態を識別するために、メッセージがプログラムに送られます。

識別コードが異なっている場合には、ファイル様式が変更されたことを意味します。この変更がプログラムで使用するフィールドに影響する場合は、プログラムをあらためてコンパイルしてそれが適切に実行できるようにしなければなりません。この変更がプログラムの使用するフィールドに影響しない場合には、プログ

ラムをコンパイルし直さずにそのファイルのために、オーバーライド・コマンドを入力し LVLCHK(*NO) を指定する方法をとることができます。LVLCHK(*NO) を指定すると、ファイルのオープン時にレベル識別コード検査は行われません。たとえば、あるフィールドがデータベース・ファイルのレコード様式の終わりに追加されたとしても、プログラムはその新しいフィールドを使いません。LVLCHK(*NO) を指定したデータベース・ファイル・オーバーライド (OVRDBF) コマンドを入力すれば、プログラムをコンパイルし直さずに実行することができます。

変更を検査する場合、いくつかの CL コマンドを使用することができます。ファイル・フィールド記述表示 (DSPFFD) コマンドを使用すると、レコード・レベルおよびフィールド・レベル記述を表示することができます。あるいは原始ステートメント入力ユーティリティー (SEU) があれば、ファイルの DDS を含むソース・ファイルを表示することができます。ファイルの中で定義されている様式レベル識別コードは、ファイル記述表示 (DSPFD) コマンドまたは DSPFFD コマンドによって表示することができます。プログラムが作成されたときに使用された様式レベル識別コードは、プログラム参照の表示 (DSPPGMREF) コマンドによって表示することができます。

ファイル記述に対する変更には、ファイルのオープン時にエラーを起こさないものもあります。これは、レコード様式識別コードが変更されなかったからか、または変更された様式をプログラムで使用しない場合に発生します。追加または削除された様式を使用しない既存のプログラムには影響を与えることなく、様式をファイルに追加またはファイルから除去することができます。

レベル識別コードが変更されなくても、ユーザーが追加または削除する DDS 機能によっては、ユーザーのプログラムの論理に変更を必要とするものがあります。したがって、追加または削除した機能を検討して、プログラムの論理に変更が必要かどうかを判別しなければなりません。

一般的に、ファイルの保全のためには、LVLCHK(*YES) の使用が勧められています。LVLCHK(*NO) を使用すると、予測し得ない結果を招く場合があります。

オープンされたファイルに関する情報の表示

オープンされたファイルについての情報を表示するには、任意のコマンド行で dspjob option(*opnf) または wrkjob option(*opnf) と入力して Enter キーを押します。

すると、次の画面が表示されます。

オープンされたファイルの表示

ジョブ : QPADEV0027 ユーザー : KELLYMR 番号 : 032138
 オープン・データ・パスの数 : 2

ファイル	ライブラリー	装置	メンバー/ 有効範囲	活動化グループ
QDUI80	QSYS	QPADEV0008	*ACTGRPDFN	0000000002 *DFTACTGRP
QDDSP0F	QSYS	QPADEV0008	*ACTGRPDFN	0000000002 *DFTACTGRP

続行するには、Enter キーを押してください。

F3= 終了 F5= 最新表示 F10= 入出力明細表示 F12= 取り消し F16= ジョブ・メニュー

「有効範囲」列は、オープンが限定されるレベルを識別します。*ACTGRPDFN は、オープンの有効範囲が活動化グループ・レベルに限定されていることを示します。ファイルがユーザー・デフォルト活動化グループでオープンされた場合、オープンの有効範囲は呼び出しプログラムの呼び出しレベル番号に限定されません。ファイルが名前付き活動化グループ内でオープンされた場合、オープンの有効範囲は活動化グループ・レベルに限定されます。*JOB は、オープンの有効範囲がジョブ・レベルに限定されていることを示します。オープン操作の有効範囲は、オーバーライド・コマンドを使用して変更することができます。

「活動化グループ」列は、活動化グループの番号と名前を識別します。*DFTACTGRP は、デフォルト活動化グループを示します。

関連概念

87 ページの『オープンされたファイルの有効範囲』

ユーザー・デフォルト活動化グループ内でオープンされたファイルの有効範囲は、呼び出しプログラムの呼び出しレベル番号に限定されます (デフォルト)。呼び出しレベル番号は、サーバーが各呼び出しスタックに割り当てた固有の番号です。名前付き活動化グループ内でオープンされたファイルの有効範囲は、活動化グループ・レベルに限定されています (デフォルト)。

97 ページの『オーバーライドの使用』

オーバーライドを使用して、ファイル名やファイルに関連した装置名などのファイル属性を一時的に変更できます。

オープンおよび入出力フィードバック域でのファイル状況のモニター

サーバーは、ファイルが正常にオープンされると、フィードバック域でファイルの状況をモニターします。サーバーがファイルに対する操作を行うと、フィードバック域が更新され、最新のファイル状況を反映します。これらのフィードバック域によって、ユーザーはより一層アプリケーションを管理することができ、エラー発生時に重要な情報を得ることができます。

フィードバック域はオープン時点で確立され、各オープン・ファイルごとに 1 つのフィードバック域があります。唯一の例外は共用ファイルの場合で、プログラムとファイルの間のデータ・パスだけでなく、フィードバック域も共用します。共用オープンの詳細については、94 ページの『ファイルの共用』を参照してください。

サーバーで使用される高水準言語によっては、操作実行の対象となっているファイルの状況およびその他の情報にアクセスできる場合もあります。ユーザーに関係のあるフィードバック域には次の 2 つがあります。

- オープン・フィードバック域

このフィードバック域には、正常にオープンされた後のファイルに関する一般的な情報が入ります。たとえば、ファイルの名前とライブラリーおよびファイル・タイプが含まれます。オープン・フィードバック域から取り出すことのできる情報の全リストについては、194 ページの『オープン・フィードバック域』を参照してください。また、ファイルに関する一般的な情報に加えて、サーバーがファイルをオープンした後の、ファイルに固有の情報も、オープン・フィードバック域に含まれます。利用可能なフィールドは、ファイル・タイプによって異なります。

さらに、オープン・フィードバック域には、ファイルに定義されている各装置や通信セッションに関する情報も含まれます。

- 入出力フィードバック域

入出力フィードバック域は次の 2 つの部分に分かれています。これらのフィードバック域は、入出力操作が正常に完了したときに更新されます。

- 共通域

共通域には、ファイルに対して実行された入出力操作に関する情報が入ります。これには、操作の数および最後に実行された操作の情報が含まれます。共通入出力フィードバック域から取り出すことのできる情報の全リストについては、211 ページの『入出力フィードバック域』を参照してください。

- ファイル依存フィードバック域

このフィードバック域には、ディスプレイ・ファイル、データベース・ファイル、プリンター・ファイル、および ICF ファイルの場合に、ファイル特定情報が入ります。たとえば、メジャーおよびマイナー戻りコードおよび装置から受け取ったデータ量があります。ファイル依存入出力フィードバック域から取り出すことのできる情報の全リストについては、220 ページの『ICF ファイルおよびディスプレイ・ファイルの入出力フィードバック域』、226 ページの『プリンター・ファイルの入出力フィードバック域』、および 227 ページの『データベース・ファイルの入出力フィードバック域』を参照してください。

上記の情報域は役立ちます。たとえば、装置ファイルでエラーが生じたときは、プログラムはファイル依存フィードバック域の中のメジャー/マイナー戻りコードに基づいて、定義済みエラー処理操作を決めることができます。データが通信装置から受信中で、相手側のアプリケーション・プログラムがエラーを送信した場合は、プログラムではデータの次のブロックが送られてエラーを示すまで、次の操作は待機するように決めることができます。おそらく、次の操作はファイルをクローズして、他方の側のアプリケーション・プログラムとの会話を打ち切るか、またはアプリケーションからの次の要求を待つということになります。

別の方法として、実際にオープンされたファイルのタイプを検出して、使用できる操作のタイプを判別することもできます。ファイル・タイプがプリンターの場合は、出力の操作しか使用できません。

関連概念

94 ページの『ファイルの共用』

iSeries のファイル管理では、いくつかのレベルに分かれた共用ファイルのサポートが提供されています。多数のユーザー、多数のジョブ、または同じジョブ内の多数のプログラムの間で、ファイルを共有することができます。

関連資料

194 ページの『オープン・フィードバック域』

オープン・フィードバック域は、オープン・データ・パス (ODP) の一部で、ファイルがオープンされた後にファイルに関する一般情報が入ります。そこには、ファイルに対して定義されている装置または通信セッションに関する情報の他に、ファイル・タイプによっては、ファイル固有の情報も入ります。この情報はオープン処理の過程で設定され、他の操作の実行に伴って更新されることがあります。

211 ページの『入出力フィードバック域』

iSeries では、OS/400 メッセージおよび入出力フィードバック情報を使用して、入出力操作の結果をプログラムに通知します。サーバーは、正常に行われたすべての入出力操作で入出力フィードバック域を更新しますが、ユーザーのプログラムがブロック化レコード入出力を使用している場合はその限りではありません。

220 ページの『ICF ファイルおよびディスプレイ・ファイルの入出力フィードバック域』

このトピックでは、ICF ファイルおよびディスプレイ・ファイルの入出力フィードバック域を表に示します。

226 ページの『プリンター・ファイルの入出力フィードバック域』

このトピックでは、プリンター・ファイルの入出力フィードバック域を表に示します。

ファイルの共用

iSeries のファイル管理では、いくつかのレベルに分かれた共用ファイルのサポートが提供されています。多数のユーザー、多数のジョブ、または同じジョブ内の多数のプログラムの間で、ファイルを共有することができます。

サーバーは、自動的に第 1 レベルのサポートを提供します。デフォルトでは、サーバーは 1 つのファイルを同時に複数のユーザーおよび複数のジョブに使用させます。サーバーは、ファイルの使用に際して、使用上の対立が起きないようにファイルおよびファイル関連リソースを割り振ります。同一ジョブ内で、1 つのプログラムが同じファイルを 2 回以上オープンする場合、または異なるプログラムが同じファイルをオープンする場合、プログラムはファイルを共用することができます。同じファイルが使用されている場合でも、各オープン操作ごとに、プログラムからデータまたは装置に至るパスが作成されるため、各オープンがそれぞれファイルの独立使用を表すことになります。

オープン・データ・パス

ジョブ内でさらに細かいレベルの共用を使用すると、複数のプログラムでデータまたは装置に至る同一パスを共用できるようになります。このパスは、**オープン・データ・パス**と呼ばれるもので、ファイルに関するすべての書き込みおよび読み取り操作が実行されるパスです。このレベルの共用は、ファイル作成コマンド、ファイル変更コマンド、およびファイル・オーバーライド・コマンドで **SHARE** パラメーターを指定することによって使用できるようになります。 **SHARE** パラメーターを使用すると、複数のプログラムでファイル状況、位置、および記憶域を共用することができます。このようにすると、ジョブに必要な主記憶域量が削減され、ファイルのオープンおよびクローズにかかる時間が節減されるため、パフォーマンスを向上させることができます。このような iSeries の共用レベルは、次の 2 つのモデルに基づいています。

- **オリジナル・プログラム・モデル**は、ソース・コードをコンパイルし、高水準言語プログラムを iSeries サーバー上に作成する一連の機能です。これは 統合言語環境 (Integrated Language Environment (ILE))[®] モデルが発表される前から存在しました。
- **ILE モデル**は、ILE 準拠の高水準言語の共通実行時環境および実行時バインド可のアプリケーション・プログラミング・インターフェース (API) を提供する、一連の構成およびインターフェースです。

オリジナル・プログラム・モデルの共用ファイル

オリジナル・プログラム・モデルでは、**SHARE(*YES)** パラメーターを使用することにより、同一ジョブで実行される 2 つ以上のプログラムで、1 つのオープン・データ・パス (ODP) を共用することができます。このパスは、プログラムをファイルに接続します。特に指定がない限り、オープン・データ・パスは、ファイルがオープンされるたびに新しく作成されます。ファイルが 2 回以上オープンされ、しかもオープン・データ・パスが、そのファイルに関して同じジョブでまだ活動状態にある場合は、そのファイルに関する活動 ODP をファイルの現在のオープンに使用できるように指定することができます。つまり、新しいオープン・データ・パスを作成する必要はなくなります。このようにすれば、2 回目以降のオープンに要する時間、およびジョブで必要とされる主記憶域量を減らすことができます。オープン・データ・パスを共用するためには、最初のオープンの際、および 2 回目以降に同一ファイルをオープンする際に、**SHARE(*YES)** を指定する必要があります。性能の良いアプリケーション・プログラムであれば、通常は、同一ジョブの中で複数のプログラムがオープンするデータベース・ファイルに対して、共用オープンを行います。他のファイルに関する **SHARE(*YES)** の指定は、アプリケーションによって異なります。

ILE モデルの共用ファイル

ILE モデルの場合、共用ファイルの有効範囲は、ジョブ・レベルまたは活動化グループ・レベルのどちらかに限定されます。**活動化グループ**とは、実行時ジョブのサブストラクチャーです。これは、1 つ以上のプ

プログラムに割り振られたサーバー・リソース (プログラムまたはプロシージャ変数の記憶、コミットメント定義、およびオープン・ファイル) によって構成されています。活動化グループは、ジョブ内にある縮小されたジョブに似ています。

有効範囲がジョブ・レベルに指定された共用ファイルは、どの活動化グループで実行されるプログラムでも共用することができます。有効範囲が活動化グループ・レベルに限定されている共用ファイルは、同じ活動化グループで実行されるプログラムでしか共用することができません。

共用ファイル: 考慮事項

ファイルを共用すると、ジョブ内のプログラムに、他の方法では不可能なやり方で対話させることができます。ただし、共用ファイルのオープン、読み取り操作と書き込み操作の実行、および共用ファイルのクローズが及ぼす影響について、理解しておく必要があります。

また、このサポートがどのように機能するのかという点や、各ファイル・タイプをご使用のプログラムで使用する際の規則について理解するため、すべてのファイル・タイプの該当する資料も参照してください。

注: ほとんどの高水準言語プログラムでは、ファイルが共用されているかどうかに関係なく、オープン操作またはクローズ操作を処理します。ファイルを高水準言語のプログラムで共用するには指定しないでください。SHARE パラメーターを用いて、ファイルが同一ジョブで共用されるように指示します。SHARE パラメーターは、ファイル作成コマンド、ファイル変更コマンド、およびファイル・オーバーライド・コマンドでのみ指定します。詳細については、ご使用の言語に該当する情報を参照してください。

関連概念

3 ページの『ファイル・タイプ』

92 ページの『オープンおよび入出力フィードバック域でのファイル状況のモニター』

サーバーは、ファイルが正常にオープンされると、フィードバック域でファイルの状況をモニターします。サーバーがファイルに対する操作を行うと、フィードバック域が更新され、最新のファイル状況を反映します。これらのフィードバック域によって、ユーザーはより一層アプリケーションを管理することができます。エラー発生時に重要な情報を得ることができます。

ジョブの中で共用されるファイルのオープンに関する考慮事項

SHARE(*YES) を指定して同一ジョブの中で共用されるファイルをオープンする場合には、考慮すべき点があります。

- あるジョブの中で共用ファイルが初めてオープンされるときは、そのファイルの後続のオープンに必要なすべてのオープン・オプションが必ず指定されるようにしなければなりません。共用ファイルの後続オープンに対して指定されたオープン・オプションが、共用ファイルの最初のオープンに対して指定されたオープン・オプションに一致しない場合は、エラー・メッセージがプログラムに送られます。(これについては、プログラムに変更を施すことによって、矛盾するオプションを除去して、訂正することができます。)

たとえば PGMA はジョブの中で FILE1 をオープンする最初のプログラムで、PGMA ではファイルを読み取る必要があるだけだとします。しかし、PGMA は、同一の共用ファイルからレコードを削除する PGMB を呼び出します。PGMB が共用ファイルからレコードを削除するためには、PGMA は、PGMA 自体でもレコードを削除することになるかのように、ファイルをオープンしなければならなくなります。これについては、高水準言語での正しい指定を使用することによって行なうことができます。(高水準言語によっては、これを行なうためには、実行されることのないファイル操作ステートメントを使用しなければならないこともあります。詳細については、ご使用の言語に該当する情報を参照してください。)

- 場合によっては、ジョブ内でファイルを共用できないことがあります。たとえば、あるプログラムでは、到着順にファイルのレコードを要求し、別のプログラムではキー順でレコードを要求するかもしれませんが。あるいは、出力を印刷するために同一ファイルを使用しますが、各プログラムからの出力は別々に生成させたいという場合も考えられます。このような場合には、オープン・データ・パスを共用しないでください。オーバーライド・コマンドで SHARE(*NO) を指定して、プログラムがジョブ内でファイルを共用しないようにします。
- 実動ライブラリーの中の共用ファイルの最初のオープンで、 UPDPROD(*NO) を指定してデバッグ・モードに入った場合、ファイルの後続共用オープンでは、元のオープン・データ・パスを共用するため、ファイルの変更が可能になってしまいます。これを防止するためには、プログラムをデバッグしている間にオーバーライド・コマンドで SHARE(*NO) を指定してから、ファイルをオープンしてください。
- 共用ファイルの最初のオープンに対してコミットメント制御を使用すると、後続の共用オープンでもすべて、コミットメント制御を使用する必要があります。
- プログラムまたはオーバーライド・コマンドでライブラリー名を指定しなかった場合 (*LIBL が使用されます)、サーバーは同じ共用ファイルが最後に *LIBL の名前でオープンされたときから、ライブラリー・リストが変更されていないものと見なします。ライブラリー・リストが変更されている場合は、オーバーライド・コマンドでライブラリー名を指定して、適切なファイルが開かれたかどうかを確認してください。
- 共用ファイルの最初のオープンで指定されたオーバーライドおよびプログラム仕様は処理されます。後続オープンで指定されたオーバーライドおよびプログラム仕様は、ファイル名またはオーバーライド・コマンドの SHARE または LVLCHK パラメーターで指定されている値を変更するものでない限り、無視されます。

ジョブの中で共用されるファイルの入出力に関する考慮事項

サーバーはファイルを共用するすべてのプログラムに対して同じ入出力域を使用するので、どのプログラムが操作をするかにかかわらず、操作の順序は順次となります。

たとえば、プログラム A がデータベース・ファイルからレコードを順次に読み取っていて、プログラム B を呼び出す直前にレコード 1 を読み取り、しかもプログラム B でもそのファイルを順次読み取る場合は、プログラム B は最初の読み取り操作でレコード 2 を読み取ります。プログラム B がそこで終了し、プログラム A が次のレコードを読み取る場合は、レコード 3 を読み取ることになります。プログラムがファイルを共用していなければ、プログラム A はレコード 1 およびレコード 2 を読み取り、プログラム B はレコード 1 を読み取るはずですが。

装置ファイルの場合、装置は最後に実行された入出力操作と同じ状態のままになっています。

ディスプレイ・ファイルおよび ICF ファイルの場合は、そのファイルをオープンする最初のプログラム以外のプログラムは、より多くのディスプレイまたはプログラム装置を獲得したり、すでに獲得されているディスプレイまたはプログラム装置をオープン・データ・パスに解放することがあります。ファイルを共用するすべてのプログラムは、新しく獲得された装置にはアクセスできますが、解放された装置にアクセスすることはできません。

ジョブの中で共用されるファイルのクローズに関する考慮事項

プログラムが共用ファイルをクローズするときに実行される処理は、その時点でそのオープン・データ・パスを共用する他のプログラムがあるかどうか依存します。

他にプログラムがある場合、実行される主な機能は、クローズを要求するプログラムをファイルから切り離すことです。接続しているファイルがデータベース・ファイルの場合は、プログラムが保留しているレコー

ドのロックもすべて解放されます。プログラムは、共用ファイルをあらためてオープンしない限り、その共用ファイルを使用することはできません。ファイルを共用する他のプログラムはまだ ODP に接続されていて、入出力操作を実行することができます。

ファイルをクローズするプログラムが、ファイルを共用している最後のプログラムである場合は、共用オプションなしでファイルが開かれた場合と同じ機能が、クローズ操作によってすべて実行されます。これには、ファイルに割り振られたすべてのリソースの解放、およびオープン・データ・パスの破棄が含まれます。

この最後のクローズ操作によって行われる機能は、ある特定の実行時エラーからの回復に必要な機能です。ユーザーのアプリケーション・プログラムが、そのようなエラーからの回復のために作成され、そのアプリケーション・プログラムが共用ファイルを使用する場合は、エラーの発生時にそのファイルに接続されているすべてのプログラムが、ファイルをクローズしなければならないことを意味します。したがって、呼び出しスタック内の前のプログラムに戻り、それらの各プログラム内でファイルをクローズする必要があるかもしれませぬ。

オーバーライドの使用

オーバーライドを使用して、ファイル名やファイルに関連した装置名などのファイル属性を一時的に変更できます。

オーバーライドは、プログラムの機能に小さな変更を加えるときに使用します。オーバーライドの場合は、プログラムの再コンパイルなしで、操作対象のデータを選択できます。下記のトピックでは、iSeries でオーバーライドを使用する方法について説明します。

関連概念

87 ページの『オープンされたファイルの有効範囲』

ユーザー・デフォルト活動化グループ内でオープンされたファイルの有効範囲は、呼び出しプログラムの呼び出しレベル番号に限定されます (デフォルト)。呼び出しレベル番号は、サーバーが各呼び出しスタックに割り当てた固有の番号です。名前付き活動化グループ内でオープンされたファイルの有効範囲は、活動化グループ・レベルに限定されています (デフォルト)。

91 ページの『オープンされたファイルに関する情報の表示』

オープンされたファイルについての情報を表示するには、任意のコマンド行で `dspjob option(*opnf)` または `wrkjob option(*opnf)` と入力して Enter キーを押します。

オーバーライド: 概要

オーバーライドは、ファイル名、ファイルに関連付けられた装置名またはリモート・ロケーション名、またはファイルの他の属性を一時的に変更する CL コマンドです。

オーバーライド・コマンドは、ディスプレイ装置から対話式に入力することもできますし、バッチ・ジョブの一環としてサブミットすることもできます。オーバーライド・コマンドは、制御言語 (CL) プログラムに組み込むことも、プログラム QCMDEXC を呼び出すことによって他のプログラムから出すこともできます。オーバーライドはどのように出されたにせよ、出されたジョブ、プログラム、またはディスプレイ装置でしか有効ではありません。さらに、他のジョブが同時に実行されていても、オーバーライドはそのような他のジョブにはまったく影響を及ぼしません。

アプリケーション・プログラムを作成するとき、ファイルはプログラムで指定されたファイル名によってそのプログラムに関連付けられます。サーバーでは、プログラムのコンパイル時または実行時に、これらのファイル名または指定ファイルの属性をユーザーがオーバーライドすることを認めています。

オーバーライドを使用すると、ファイルの作成時に指定されているファイル属性の全部ではありませんが、大部分を変更することができます。場合によっては、元のファイル定義には含まれない属性をオーバーライドの中で指定することもできます。

ファイルのオーバーライドは、ファイルの属性を永続的に変更するわけではありませんから、ファイルの変更とは異なります。たとえば、プリンター・ファイルに関して、コピー数を 2 部のかわりに 6 部要求するようにオーバーライドした場合は、プリンター・ファイルに関するファイル記述では 2 部の指定のままですが、印刷は 6 部行われます。サーバーはファイル・オーバーライド・コマンドを使用して、オープンするファイルおよびそのファイル属性を決めます。

サーバーは次の 3 つのオーバーライド機能を提供します。

- 102 ページの『オーバーライドの適用』
- 120 ページの『オーバーライドの削除』
- 121 ページの『オーバーライドの表示』

メッセージ・ファイルの場合は、オーバーライドの処理が他のファイルの場合とは異なる点はいくつかあります。オーバーライドできるのはメッセージ・ファイルの名前だけであり、属性はオーバーライドできません。

関連情報

制御言語 (CL)

オーバーライドを使用する利点:

オーバーライドが特に有効なのは、プログラムを再コンパイルせずに、プログラムの機能をわずかに変更する場合またはプログラムが操作するデータを選択する場合です。オーバーライドの値を使用すると、より多くの種類の状況で汎用プログラムを使用できるようになります。

オーバーライドを使用できる項目の例には、次の場合があります。

- 処理するファイルの名前の変更
- 処理するデータベース・ファイル・メンバーの選択
- 出力をスプーリングするかどうかの指示
- 他のテープ装置への出力の宛先変更
- 行/インチ (1pi) やコピーの数など、プリンターの特性変更
- ICF ファイルで使用するリモート・ロケーションの選択
- 通信セッションの特性変更

オーバーライド・コマンドの要約:

DLTOVR、DSPOVR、OVRDBF などの CL コマンドを使用して、ファイルに関するオーバーライド機能を処理することができます。

ユーザーは次の CL コマンドを使用して、ファイルに関するオーバーライド機能を処理することができます。

DLTOVR

呼び出しレベルで以前に指定した 1 つまたは複数のファイル・オーバーライド (メッセージ・ファイル・オーバーライドを含む) を削除する、オーバーライド削除コマンド。

DSPOVR

活動呼び出しレベル、活動化グループ・レベル、またはジョブのジョブ・レベルでのファイルのオーバーライドを表示する、オーバーライド表示コマンド。

OVRDBF

プログラム中で指定されたデータベース・ファイルのオーバーライド (置換)、プログラムが使用するデータベース・ファイルの特定のパラメーターのオーバーライド、または処理対象となるファイルおよびそのファイルの特定のパラメーターのオーバーライドを行う、データベース・ファイル・オーバーライド・コマンド。

OVRDSPF

プログラム中で指定されたディスプレイ・ファイルのオーバーライド (置換)、プログラムが使用するディスプレイ・ファイルでの特定のパラメーターのオーバーライド、または処理対象となるファイルおよびそのファイルの特定のパラメーターのオーバーライドを行う、ディスプレイ・ファイル・オーバーライド・コマンド。

OVRICFF

プログラムの中で指定されたファイルのオーバーライド、および処理対象となるファイルの特定のパラメーターのオーバーライドを行う、ICF ファイル・オーバーライド・コマンド。

OVRMSGF

プログラムの中で使用されるメッセージ・ファイルをオーバーライドするメッセージ・ファイル・オーバーライド・コマンド。このコマンドでオーバーライドを行う規則は、他のオーバーライド・コマンドとは異なっています。メッセージ・ファイルのオーバーライドについての詳細は、制御言語 (CL) を参照してください。

OVRPRTF

プログラム中で指定されたプリンター・ファイルのオーバーライド (置換)、プログラムが使用するプリンター・ファイルの特定のパラメーターのオーバーライド、または処理対象となるファイルおよびそのファイルの特定のパラメーターのオーバーライドを行う、プリンター・ファイル・オーバーライド・コマンド。

OVRSAVF

プログラムの中で指定されたファイルのオーバーライド (置換)、プログラムが使用するファイルでの特定の属性のオーバーライド、または処理対象となるファイルおよびファイルの特定の属性のオーバーライドを行う、保管ファイル・オーバーライド・コマンド。

OVRTAPF

プログラムの中で指定されたファイルのオーバーライド (置換)、プログラムが使用するファイルでの特定の属性のオーバーライド、または処理対象となるファイルおよびファイルの特定の属性のオーバーライドを行う、テープ・ファイル・オーバーライド・コマンド。

一部のコマンドに対するオーバーライドの影響:

コマンドの中には、オーバーライドをまったく無視するものや、特定のパラメーターのオーバーライドだけを許可するものがあります。

以下に列挙されているのはよく使用されるコマンドですが、いずれもオーバーライドを完全に無視します。

- ADDLFM
- ADDPFM
- ALCOBJ
- APYJRNCHG

- CHGOBJOWN
- CHGPTR
- CHGSBSD
- CHGXXXF (すべてのファイル変更コマンド)
- CLRPFM
- CLRSAVF
- CPYIGCTBL
- CRTDUPOBJ
- CRTAUTHLR
- CRTSBSD
- CRTTAPF
- DLCOBJ
- DLTF
- DLTAUTHLR
- DSPDBR
- DSPFD
- DSPFFD
- DSPJRN
- EDTOJAUT
- EDTDLOAUT
- ENDJRNPF
- GRTOJAUT
- INZPFM
- MOVOBJ
- RGZPFM
- RMVJRNCHG
- RMVM
- RNMOBJ
- RTVMBRD
- RVKOBJAUT
- SBMDBJOB
- SIGNOFF
- STRDBRDR
- STRJRNPF

注: 保管操作および復元操作は、それぞれの媒体 (テープ、ディスク、保管ファイル) に関連するすべてのファイルのオーバーライドを無視します。

経路指定終了ステップまたはジョブ終了処理の一環としてオープンされるサーバー・ファイルの場合、オーバーライドは適用されません。たとえば、ジョブ・ログ・ファイルに対してオーバーライドを指定することはできません。サーバー・ファイル内の何かをオーバーライドする必要がある時は、オーバーライド・コマンド以外のコマンドによって変更できる場合もあります。たとえば、あるジョブ・ログに関する出力待ち行

列を変更するには、ジョブ変更 (CHGJOB) コマンドでジョブに関する出力待ち行列の名前を指定する OUTQ パラメーターを使用して、出力待ち行列をサインオフの前に変更することができます。ジョブ・ログに関するプリンター・ファイルに出力待ち行列に対する値 *JOB が入っている場合は、その出力待ち行列がジョブに対して指定された出力待ち行列になります。

次のコマンドでは、SRCFILE および SRCMBR パラメーターに対してだけオーバーライドが使用できません。

- CRTCMD
- CRTICFF
- CRTDSPF
- CRTLF
- CRTXXXPGM
- CRTPRTF
- CRTSRCPF
- CRTTBL
- CRTPF
- すべてのプログラム作成コマンド。これらのコマンドでも、オーバーライドを使用して、どのファイルをコンパイル済みプログラムによってオープンするかを決めます。

次のコマンドでは、TOFILE、MBR、SEQONLY、LVLCHK、および INHWRT パラメーターに対してオーバーライドが使用できません。

OPNQRYF

以下のコマンドではオーバーライドを行うことができますが、MBR を *ALL に変えることはできません。

- CPYFRMPCD
- CPYTOPCD

次のコマンドでは、そこで使用するディスプレイ・ファイルにオーバーライドを適用することはできません。これらのコマンドが使用するプリンター・ファイルに対しては、ファイル・タイプまたはファイル名を変更してはなりません。これらのコマンドによって使用されるプリンター・ファイルに対して実行できる変更にはいくつかの制約がありますが、可能な指定を複数組み合わせた場合、常に適切な報告書が生成されることをサーバーは保証できません。

DMPOBJ および DMPSYSOBJ

(前記の制限に加えて、これらのコマンドではダンプ対象ファイルに対するオーバーライドは使用できません。)

DSPXXXXXX

(すべての表示コマンド。ファイルに関する情報を表示する表示コマンドでは、そのファイルに対するオーバーライドは使用できません。)

DSPIGCDCT

EDTIGCDCT

GO (メッセージ・ファイルをオーバーライドすることができます。)

PRTXXXXXX

(すべての印刷コマンド。)

QRYDTA

TRCXXX

(すべてのトレース・コマンド。)

WRKXXXXXX

(すべての処理コマンド。)

関連概念

118 ページの『プログラムのコンパイル時におけるオーバーライドの適用』

プログラムのコンパイル時にオーバーライドを適用する目的は、ソース・ファイルを選択するため、あるいは、入出力操作に使用するレコード様式を定義するためにコンパイラーが使用する外部データ定義を提供するためです。

マルチスレッド化されたジョブでのオーバーライドの使用:

マルチスレッド化されたジョブの中で OVRDBF、OVRPRTF、OVRMSGF、および DLTOVR コマンドを使用できますが、いくつかの制約事項があります。

制約事項は次のとおりです。

- データベース・ファイル・オーバーライド (OVRDBF) コマンド。このコマンドは、マルチスレッド化されたジョブの初期スレッドから実行することができます。2 次スレッドで実行されるオープン操作に影響を与えるのは、有効範囲がジョブまたは ILE 活動化グループに限定されたオーバーライドだけです。
- プリンター・ファイル・オーバーライド (OVRPRTF) コマンド。このコマンドは、マルチスレッド化されたジョブの初期スレッドから実行することができます。2 次スレッドで実行されるオープン操作に影響を与えるのは、有効範囲がジョブまたは ILE 活動化グループに限定されたオーバーライドだけです。
- メッセージ・ファイル・オーバーライド (OVRMSGF) コマンド。このコマンドは、マルチスレッド化されたジョブの初期スレッドから実行することができます。このコマンドは、初期スレッド内でのメッセージ・ファイルの参照にのみ適用されます。2 次スレッドで実行されるメッセージ・ファイルの参照には影響を与えません。
- オーバーライド削除 (DLTOVR) コマンド。このコマンドは、マルチスレッド化されたジョブの初期スレッドから実行することができます。

これ以外のオーバーライド・コマンドは、マルチスレッド化されたジョブでは許可されておらず、無視されます。

オーバーライドの適用

実行できるオーバーライドには、ファイル・オーバーライドとプログラム装置項目オーバーライドという 2 つの一般的なタイプがあります。

ファイル・オーバーライド

ファイル・オーバーライドは、以下の事柄をオーバーライドします。

- ファイル属性
- ファイル名
- ファイル属性およびファイル名の両方
- ファイル・オープンの有効範囲
- ファイル・タイプ

ファイル・タイプのオーバーライドの詳細については、128 ページの『ファイルの指定変更』を参照してください。

プログラム装置項目オーバーライド

プログラム装置項目オーバーライドは、ユーザーのプログラムが通信するリモート・サーバーまたは装置のそれぞれとアプリケーションとの間のリンクを提供する ICF ファイルの属性をオーバーライドします。プログラム装置項目のオーバーライドの詳細については、「[ICF Programming](#)」



を参照してください。

関連概念

3 ページの『ファイル・タイプ』

128 ページの『ファイルの指定変更』

ファイル指定変更とは、オーバーライドを使用して、データ入力またはデータ出力を異なるタイプの装置に指定することです。たとえば、ディスクットに指定したデータの代わりにプリンターに指定変更します。ただし、オーバーライドをこのように使用することについては、関連する 2 つの装置のそれぞれ異なる特性をプログラムで適応させることができなければならないため、上記のリストに示したオーバーライド用例の場合に比べて注意が必要になります。

関連情報

[ICF Programming \(PDF\)](#)

ファイル属性のオーバーライド:

ファイル・オーバーライドの最も単純な形式は、ファイル属性のオーバーライドです。

ファイル属性は、以下の操作の結果として作成されます。

- ファイル作成コマンドおよびメンバー追加コマンド。最初に、これらのコマンドがファイル属性を作成します。
- ファイルを使用するプログラム。ユーザー・プログラムではコンパイル時に、ファイル属性をいくつか指定することができます。(指定することができる属性は、プログラムの作成に使用する高水準言語によって異なります。)
- オーバーライド・コマンド。ユーザー・プログラムの中で使用されているファイル記述とファイル・パラメーターの組み合わせから成るファイル属性を、プログラム実行時にこれらのコマンドでオーバーライドすることができます。

たとえば、以下のような属性を持つプリンター・ファイルを作成したとします。

- ページ・サイズ: 60 × 80
- 行/インチ (lpi): 6
- 印刷出力のコピー数: 2
- ファイル区切りページ数: 2
- オーバーフロー行番号: 55

この場合、プリンター・ファイル作成 (CRTPRTF) コマンドは、次のようになります。

```
CRTPRTF FILE(QGPL/OUTPUT) SPOOL(*YES) +  
  PAGESIZE(60 80) LPI(6) COPIES(2) +  
  FILESEP(2) OVRFLW(55)
```

ユーザーのアプリケーション・プログラムの中では、プリンター・ファイル OUTPUT は、オーバーフロー行番号 58、ページ・サイズ 66 × 132 として指定されています。

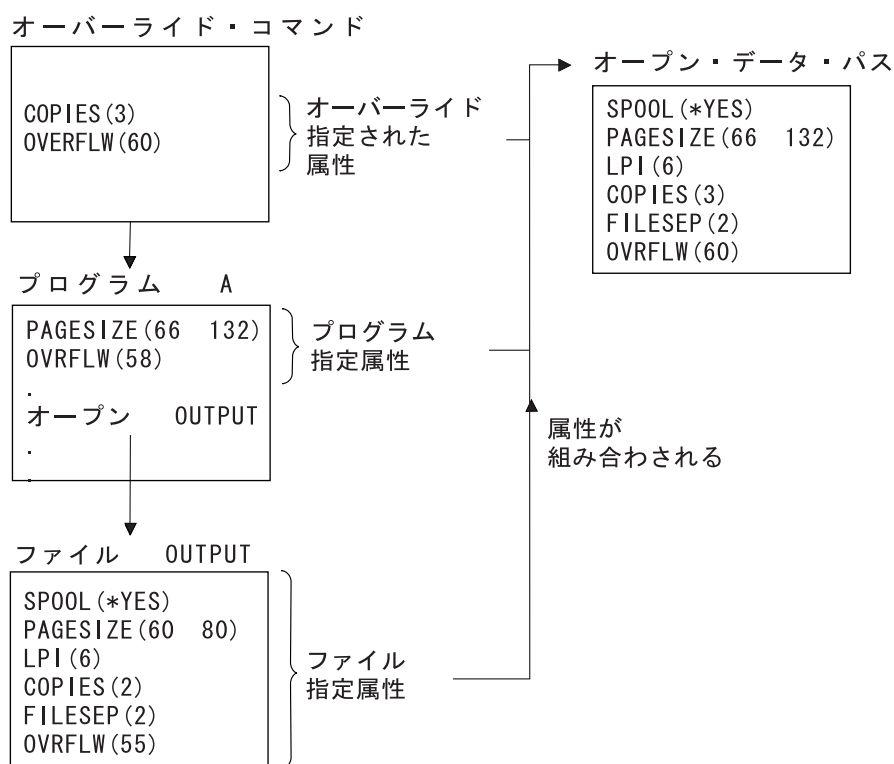
しかし、アプリケーション・プログラムを実行する前に、印刷出力のコピー数を 3 に、オーバーフロー行を 60 にそれぞれ変更したいとします。この場合、オーバーライド・コマンドは以下のようになります。

OVRPRTF FILE(OUTPUT) COPIES(3) OVRFLW(60)

そこでアプリケーション・プログラムを呼び出すと、印刷出力のコピーが 3 部生成されます。

アプリケーション・プログラムで OUTPUT ファイルをオープンすると、サーバーはファイル指定属性、プログラム指定属性、およびオーバーライド指定属性を組み合わせて、オープン・データ・パスを形成します。プログラムが実行される際、サーバーはオープン・データ・パスを使用します。サーバーはまず、ファイル指定属性とプログラム指定属性を組み合わせます。次いで、これらの組み合わせられた属性をオーバーライド属性と組み合わせます。この例では、OUTPUT ファイルがオープンされ、出力操作が実行されると、ページ・サイズが 66 × 132、6 行/インチ、コピー数 3、ファイル区切り 2 ページ、および 60 行オーバーフローで、スプール出力が生成されます。

図 3 はこの例を示しています。



RV2H087-0

図 3. ファイル属性のオーバーライド

ファイル名のオーバーライド:

プログラムによって使用されるファイルを変更するもう 1 つの単純な形式は、ファイル名のオーバーライドです。プログラムのコンパイル後にファイルが移動または名前変更された場合などには、これが役立ちます。

たとえば、プリンター・ファイル OUTPUT (OUTPUT はアプリケーション・プログラムの中で指定) ではなく、プリンター・ファイル REPORTS を使用して、アプリケーション・プログラムによる出力を印刷させたいとします。プログラムを実行する前に、次のコマンドを入力してください。

OVRPRTF FILE(OUTPUT) TOFILE(REPORTS)

ファイル REPORTS を使用する前に、あらかじめファイルが CRTPRTF コマンドで作成されていなければなりません。

ファイル名およびファイル属性のオーバーライド:

この様式のファイルのオーバーライドは、ファイル属性のオーバーライドとファイル名のオーバーライドとを組み合わせたものです。この様式のオーバーライドを使用すると、プログラムの中で使用するファイルをオーバーライドすることができますし、オーバーライド・ファイルの属性をオーバーライドすることもできます。

たとえば、プリンター・ファイル OUTPUT (OUTPUT はアプリケーション・プログラムの中で指定) ではなく、プリンター・ファイル REPORTS を使用して、アプリケーション・プログラムによる出力を印刷させたいとします。アプリケーション・プログラムにプリンター・ファイル REPORTS を使用させるだけでなく、コピーも 3 部生成したいとします。ファイル REPORTS は、次のコマンドで作成されたと想定します。

```
CRTPRTF FILE(REPORTS) SPOOL(*YES) +  
  PAGESIZE(68 132) LPI(8) OVRFLW(60) +  
  COPIES(2) FILESEP(1)
```

プログラムを実行する前に、次のコマンドを入力します。

```
OVRPRTF FILE(OUTPUT) TOFILE(REPORTS) COPIES(3)
```

そこでアプリケーション・プログラムを呼び出すと、プリンター・ファイル REPORTS を使用して、出力のコピーが 3 部生成されます。

これは次の 2 つのオーバーライド・コマンドとは等しくない ことに注意してください。

オーバーライド 1 OVRPRTF FILE(OUTPUT) TOFILE(REPORTS)

オーバーライド 2 OVRPRTF FILE(REPORTS) COPIES(3)

個々のファイルのオープンの場合、オーバーライドは各呼び出しレベルごとに 1 つだけ適用されます。したがって、プログラムによって使用されるファイルをオーバーライドし、1 つの呼び出しレベルからオーバーライド・ファイルの属性もオーバーライドしたい場合は、単一のコマンドを使用しなければなりません。2 つのオーバーライドを使用すると、最初のオーバーライドがプリンター・ファイル REPORTS を使用して出力を印刷します。サーバーは 2 つ目のオーバーライドを無視します。

オープン・ファイルの有効範囲のオーバーライド:

適切なオーバーライド・コマンドでオープン有効範囲 (OPNSCOPE) パラメーターを使用すれば、ファイル・オープン操作の有効範囲を変更することができます。OPNSCOPE パラメーターの値は *JOB または *ACTGRPDFN (デフォルト) のどちらかです。

このパラメーターを使用して、オープン操作の有効範囲を、呼び出しレベルまたは活動化グループ・レベルからジョブ・レベルに変更することができます。

たとえば、以下のオーバーライド・コマンドでは、BILLING ファイルのオープン操作の有効範囲をジョブ・レベルに限定されます。

```
OVRDBF FILE(BILLING) OPNSCOPE(*JOB)
```

サーバーがオーバーライドを処理する方法:

統合言語環境では、オーバーライドの有効範囲を呼び出しレベル、活動化グループ・レベル (デフォルト)、およびジョブ・レベルに限定することができます。

図 4 は、統合言語環境で稼働しているジョブを示しています。

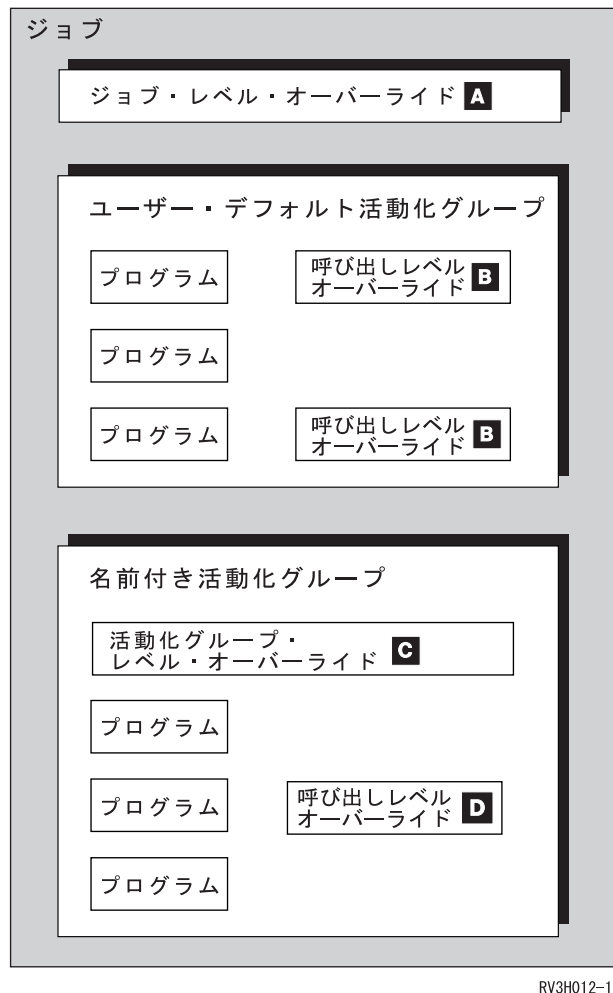


図 4. 統合言語環境内にあるジョブ

以下の説明で、参照キーとは、図 4 にある対応する参照キーのことです。

ジョブとは、サーバーが実行する作業単位のことです。対話式ジョブは、ユーザーがサインオンした時に開始し、サインオフしたときに終了します。有効範囲がジョブ・レベルに限定されているオーバーライド (A) は、ジョブ内の任意の活動化グループで稼働中のすべてのプログラムに有効です。特定のジョブ・レベルのファイルに対しては、活動状態のオーバーライドが 1 つだけ許可されます。複数のオーバーライドが指定された場合、効果があるのは最新のものです。有効範囲がジョブ・レベルに限定されているオーバーライドは、以下のいずれかのイベントが発生するまで有効です。

- ジョブが終了する
- サーバーが明示的にオーバーライドを削除する
- 同じファイルに対して、オーバーライドが別のジョブ・レベルのオーバーライドに置換される

これは、オーバーライドが指定された呼び出しレベルにかかわらず適用されます。たとえば、呼び出しレベル 3 で発行されたオーバーライドの有効範囲をジョブ・レベルに限定した場合、このオーバーライド

は、呼び出しレベル 3 が削除されても引き続き有効です。オーバーライドの有効範囲をジョブ・レベルに限定するには、オーバーライド・コマンドに `OVRSCOPE(*JOB)` を指定します。

ユーザー・デフォルト活動化グループで指定されているオーバーライド (**B**) は、有効範囲を呼び出しレベルまたはジョブ・レベルに限定することができます。それらの有効範囲をユーザー・デフォルト活動化グループ・レベルに限定することはできません。しかしながら、名前付き活動化グループ・レベルで指定されたオーバーライド (**C** および **D**) は、有効範囲を呼び出しレベル、活動化グループ・レベル、またはジョブ・レベルに限定することができます。有効範囲が名前付き活動化グループ・レベルに限定されているオーバーライド (**C**) は、オーバーライドが置換または削除されるまで、あるいは名前付き活動化グループが削除されるまで有効です。

有効範囲が名前付き活動化グループ内呼び出しレベルに限定されたオーバーライド (**D**) は、置換または削除されるまで、あるいはオーバーライドが指定されたプログラムが終了するまで有効です。オーバーライドの有効範囲を呼び出しレベルに限定するには、オーバーライド・コマンドに `OVRSCOPE(*CALLLVL)` を指定します。

有効範囲が名前付き活動化グループ・レベルに限定されているオーバーライドは、名前付き活動化グループで稼働中のプログラムにのみ適用されます。他の名前付き活動化グループまたはユーザー・デフォルト活動化グループで稼働中のプログラムには効果はありません。

呼び出しレベルは、あるプログラムがジョブ内の別のプログラムを呼び出すときに、関連するプログラムの間で従属関係を識別します。呼び出しレベルに有効範囲が限定されたオーバーライドは、指定された時点から、オーバーライドが置換、削除されるまで、またはオーバーライドが指定されたプログラムの終了まで有効です。これは、オーバーライドがユーザー・デフォルト活動化グループで出された場合にも、名前付き活動化グループで出された場合にもあてはまります。

たとえば、次のようになります。

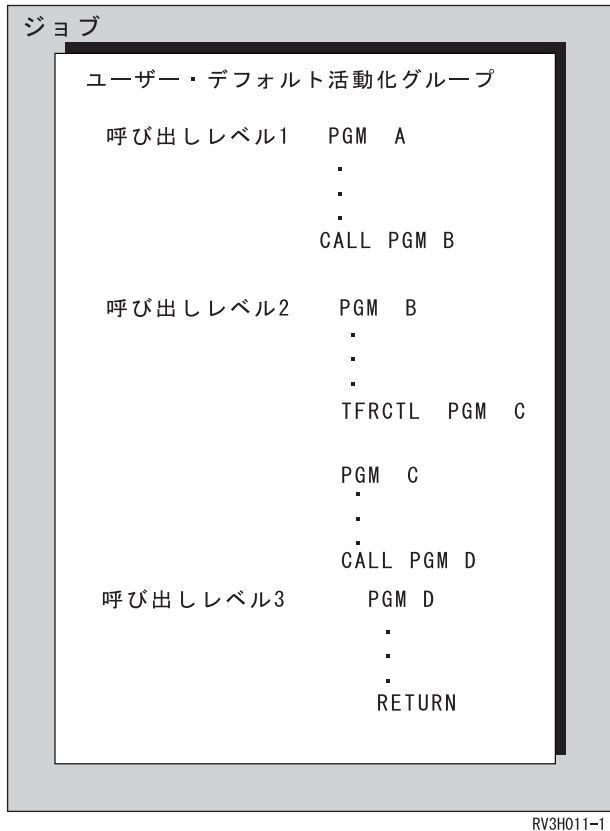


図5. ジョブ内の呼び出しレベル

ジョブ処理 (WRKJOB)、活動ジョブ処理 (WRKACTJOB)、ジョブ表示 (DSPJOB) などのコマンドにはいくつかのオプションがあり、これによって活動ジョブの呼び出しスタックを表示させることができます。呼び出しスタックに表示されたプログラムとそのプログラムの呼び出しレベルの間には、1対1の関係があります。呼び出しスタックの最初 (リストの先頭) に表示されたプログラム名は、そのジョブの呼び出しレベル1であるプログラムの名前です。呼び出しレベル1は、ジョブに対する最下位の呼び出しレベルです。2番目に表示されるプログラム名は、そのジョブに関する呼び出しレベル2のプログラムです。最後に表示されるプログラム名は、そのジョブに関する最上位の呼び出しレベルのプログラム名です。

図5の例では、PGMCへの制御権移動 (TFRCTL) によって、呼び出しスタックのPGMBがPGMCに置換されます。CALLコマンドは他のプログラムを呼び出しスタックに入れます。RETURNコマンドはスタックからプログラムを除去します。

オーバーライド処理の優先順位:

オーバーライドの中には、他のオーバーライドより優先順位が高いものがあります。

オープン操作が以下の順序で発生したときに、オーバーライドがサーバーによって処理されます。

1. 活動化グループ内の最も古いプロシーチャーのレベルと等しいかそれ以上の呼び出しレベルのオーバーライドがまず適用されます。
2. オープン操作が発行されたのと同じ活動化グループ内で指定された活動化グループ・レベルのオーバーライドが適用されます。
3. 活動化グループ内の最も古いプロシーチャーのレベルの下位にある呼び出しレベルのオーバーライドが適用されます。

4. ジョブ・レベルのオーバーライドが適用されます。

サーバーがオーバーライドを処理する方法: シナリオ:

- I オーバーライドの有効範囲が活動化グループに限定されている場合、その活動化グループ内の最も古いプロ
- I シージャーの呼び出しレベルに達するまで、サーバーはそれらのオーバーライドを処理しません。

以下の例は、複数の活動化グループでオーバーライドが実行される方法を示しています。

注: コード例を使用することにより、お客様は 239 ページの『コードに関する特記事項』の条件に同意することになります。

呼び出しレベル 2	プログラム A (ユーザー・デフォルト活動化グループ内) OVRPRTF FILE(YYY) FOLD(*YES) OVRSCOPE(*CALLLVL) CALL PGM B
呼び出しレベル 3	プログラム B (活動化グループ 8 内) OVRPRTF FILE(ZZZ) TOFILE(YYY) DEV(P1) LPI(6) + OVRSCOPE(*CALLLVL) CALL PGM C
呼び出しレベル 4	プログラム C (ユーザー・デフォルト活動化グループ内) OVRPRTF FILE(ZZZ) CPI(12) OVRSCOPE(*CALLLVL) CALL PGM D
呼び出しレベル 5	プログラム D (活動化グループ 21 内) OVRPRTF FILE(YYY) DEV(P2) OVRSCOPE(*JOB) CALL PGM E
呼び出しレベル 6	プログラム E (活動化グループ 21 内) OVRPRTF FILE(ZZZ) LPI(12) OVRSCOPE(*ACTGRPDFN) CALL PGM F
呼び出しレベル 7	プログラム F (活動化グループ 8 内) OVRPRTF FILE(ZZZ) LPI(9) OVRSCOPE(*CALLLVL) CALL PGM G
呼び出しレベル 8	プログラム G (活動化グループ 8 内) OVRPRTF FILE(ZZZ) DUPLEX(*NO) + OVRSCOPE(*ACTGRPDFN) CALL PGM H
呼び出しレベル 9	プログラム H (活動化グループ 8 内) OVRPRTF FILE(YYY) LPI(5) OVRSCOPE(*ACTGRPDFN) CALL PGM I
呼び出しレベル 10	プログラム I (活動化グループ 8 内) OPEN FILE(ZZZ)

図 6. 複数の活動化グループ内でのオーバーライド処理の例: 複数の活動化グループでオーバーライドが実行される方法についての例。

プログラム I がファイル ZZZ をオープンすると、ファイル ZZZ は次のような属性を持つようになります。

CPI(12)

呼び出しレベル 4 から

FILE(YYY)

呼び出しレベル 3 から

LPI(5) 呼び出しレベル 9 から

FOLD(*YES)

呼び出しレベル 2 から

DEV(P2)

呼び出しレベル 5 から

サーバーは次の順序でオーバーライドを処理します。

1. ファイル ZZZ は、呼び出しレベル 10 でオープンする。サーバーでは、有効範囲が呼び出しレベルに限定され、呼び出しレベル 10 で発行された任意のオーバーライドを検索します。そのようなオーバーライドはありません。
2. サーバーでは、有効範囲が呼び出しレベルに限定された適用可能なオーバーライドを探すために、そのすぐ前の呼び出しレベル (レベル 9) を検索する。そのようなオーバーライドはありません。(呼び出しレベル 9 で出されたオーバーライドは、ファイル YYY に対するものなので適用されません。)
3. サーバーでは、有効範囲が呼び出しレベルに限定された適用可能なオーバーライドを探すために、呼び出しレベル 8 を検索する。ファイル ZZZ に対するオーバーライドはありますが、有効範囲は活動化グループ・レベルに限定されています。サーバーでは、活動化グループ 8 の中で最も古いプロシージャのレベルと等しいか、またはそれ以上の呼び出しレベルのすべてのオーバーライドを処理し終わらなければこのオーバーライドを処理しません。この例の場合、活動化グループ 8 の中で最も古いプロシージャの呼び出しレベルは 3 です。したがって、サーバーは、呼び出しレベル 8 で出された活動化グループの一時的変更を処理する前に、3 またはそれ以上の呼び出しレベルで出されたすべての呼び出しレベルを処理します。
4. サーバーでは、有効範囲が呼び出しレベルに限定された適用可能なオーバーライドを探すために、呼び出しレベル 7 を検索する。呼び出しレベル 7 で発行されたオーバーライドは有効範囲が呼び出しレベルに限定されているので、処理されます。LPI(9) 属性がファイル ZZZ に割り当てられています。
5. サーバーでは、有効範囲が呼び出しレベルに限定された適用可能なオーバーライドを探すために、呼び出しレベル 6 を検索する。呼び出しレベル 6 は、活動化グループ 21 の中にあることに注意してください。ファイル ZZZ に対するオーバーライドはありますが、有効範囲は活動化グループ 21 の活動化グループ・レベルに限定されます。オーバーライドの有効範囲が活動化グループ 8 ではない活動化グループに限定されているので、サーバーではこのオーバーライドをすべて無視します。
6. サーバーでは、有効範囲が呼び出しレベルに限定された適用可能なオーバーライドを探すために、呼び出しレベル 5 を検索する。そのようなオーバーライドはありません。(呼び出しレベル 5 で発行されたオーバーライドは、ファイル YYY に対するものなので適用されません。)
7. サーバーでは、有効範囲が呼び出しレベルに限定された適用可能なオーバーライドを探すために、呼び出しレベル 4 を検索する。呼び出しレベル 4 で発行されたオーバーライドは、有効範囲が呼び出しレベルに限定されているので処理されます。CPI(12) 属性がファイル ZZZ に割り当てられています。
8. サーバーでは、有効範囲が呼び出しレベルに限定された適用可能なオーバーライドを探すために、呼び出しレベル 3 を検索する。呼び出しレベル 3 で発行されたオーバーライドは、有効範囲が呼び出しレベルに限定されているので処理されます。オープンされるファイルは YYY から ZZZ に変更されていることに注意してください。DEV(P1) 属性がファイル YYY に割り当てられています。LPI(9) 属性は LPI(6) に変更され、ファイル YYY に割り当てられています。

呼び出しレベル 3 は、活動化グループ 8 の中で最も古いプロシージャの呼び出しレベルです。ですから、活動化グループ 8 の活動化グループ・レベルに有効範囲が限定された (ファイル YYY に対する) 任意のオーバーライドが処理されます。呼び出しレベル 9 で発行されたオーバーライドが処理されます。これにより、LPI(6) 属性が LPI(5) に変更されます。

9. サーバーでは、有効範囲が呼び出しレベルに限定された適用可能なオーバーライドを探すために、呼び出しレベル 2 を検索する。呼び出しレベル 2 で発行されたオーバーライドが処理されます。これにより、FOLD(*YES) 属性がファイル YYY に割り当てられます。
10. サーバーでは、有効範囲が呼び出しレベルに限定された適用可能なオーバーライドを探すために、呼び出しレベル 1 を検索する。そのようなオーバーライドはありません。
11. サーバーでは、有効範囲がジョブ・レベルに限定された適用可能なオーバーライドを探すために、ジョブ・レベル 1 を検索する。呼び出しレベル 5 で発行されたオーバーライドは有効範囲がジョブ・レベルに限定されており、ファイル YYY に対してのものなので、処理されます。これにより、DEV(P1) 属性は DEV(P2) に変更されます。

オーバーライドの処理: 一般的な原則:

サーバーは、いくつかの一般的な原則に従ってオーバーライドを処理します。

- 適用されるオーバーライドには、ファイルがアプリケーション・プログラムによってオープンされる時点、またはファイルをオープンするプログラムのコンパイル時、あるいはまたは一定のサーバー・コマンドが使用される時に有効なものをすべて含みます。(103 ページの『ファイル属性のオーバーライド』、118 ページの『プログラムのコンパイル時におけるオーバーライドの適用』、99 ページの『一部のコマンドに対するオーバーライドの影響』を参照してください。) したがって、適用するオーバーライドについてはいずれも、ファイルがプログラムによってオープンされる前、またはファイルをオープンするプログラムがコンパイルされる前のいずれかに指定しなければなりません。必ずしもプログラムで使用されるすべてのファイルにオーバーライドを提供する必要はありません。オーバーライドが提供されていないファイルの名前はすべて、実際のファイル名として使用されます。
- 有効範囲がジョブ・レベルに限定されたオーバーライド・コマンドは、置換、削除、またはオーバーライドが指定されたジョブが終了するまで有効です。オーバーライドの削除の詳細については、120 ページの『オーバーライドの削除』を参照してください。
- 各レベル (ジョブ・レベル、活動化グループ・レベル、または呼び出しレベル) では、ファイルに対して 1 つの活動状態のオーバーライドしか存在しません。同一ファイルに対して複数のオーバーライドが同一レベルに存在する場合は、最新のオーバーライドが活動状態となります。

同一ファイルに対して複数のオーバーライドが存在するときのオーバーライドの処理方法の例については、114 ページの『同一呼び出しレベルの同一ファイルへのオーバーライド: シナリオ』を参照してください。

- 有効範囲がジョブ・レベルに限定されたオーバーライド・コマンドは、オーバーライドが指定された呼び出しレベルまたは活動化グループにかかわらず、ジョブ内で稼働しているすべてのプログラムに適用されます。
- 有効範囲が活動化グループ・レベルに限定されたオーバーライド・コマンドは、オーバーライドが指定された呼び出しレベルにかかわらず、その活動化グループ内で稼働しているすべてのプログラムに適用されます。
- 対話式に入力されたオーバーライド・コマンド (有効範囲が呼び出しレベルに限定されている) は、コマンド処理プログラムの呼び出しプログラムの呼び出しレベルで存在します。たとえば、コマンド入力画面から入力されたオーバーライド (有効範囲が呼び出しレベルに限定されている) は、コマンド入力画面から呼び出されたコマンド処理プログラムで削除または置換することはできません。
- CL プログラムの中でコーディングされたオーバーライド (有効範囲が呼び出しレベルに限定されている) の呼び出しレベルは、その CL プログラムの呼び出しレベルです。
- バッチ・ジョブの中のプログラム外のオーバーライド (有効範囲が呼び出しレベルに限定されている) は、そのバッチ・ジョブのコマンド処理プログラムの呼び出しレベルを取ります。

- オーバーライド・コマンド (有効範囲が呼び出しレベルに限定されている) が QCMDEXC を呼び出すことによって実行される場合、そのオーバーライドは QCMDEXC を呼び出したプログラムの呼び出しレベルを取ります。例については、115 ページの『CL プログラム・オーバーライド』を参照してください。
- 呼び出し操作からの終了 (ENDPGM、RETURN、または異常終了) は、その呼び出しレベルに有効範囲が限定されたオーバーライドを削除します。その呼び出しレベルで出されたオーバーライドでも、有効範囲が活動化グループ・レベルまたはジョブ・レベルに限定されたオーバーライドは削除されません。

例については、『オーバーライドに対する終了の影響についてのシナリオ』を参照してください。

- TFRCTL コマンドは、あるプログラムを同一呼び出しレベルの別のプログラムによって置き換えます。制御権の転送先のプログラムは、TFRCTL コマンドが入っていたプログラムと同一呼び出しレベルで実行されます。制御権を別のプログラムに転送する元のプログラムの中のオーバーライド・コマンドは、制御権の転送の間に削除されません。

例については、113 ページの『オーバーライドに対する TFRCTL の影響についてのシナリオ』を参照してください。

- 単一のファイルに対して、いくつかのオーバーライド (呼び出しレベルあたり 1 つ、活動化グループ・レベルで 1 つ、およびジョブ・レベルで 1 つ) が許可されます。オーバーライドは、108 ページの『オーバーライド処理の優先順位』の優先順位に従って処理されます。

オーバーライドの処理の例については、109 ページの『サーバーがオーバーライドを処理する方法: シナリオ』を参照してください。

- 低い呼び出しレベル、活動化グループ・レベル、およびジョブ・レベルのオーバーライドがオーバーライド処理をしてしまうのを防ぐため、現在のオーバーライドに SECURE(*YES) を指定してください。例については、115 ページの『オーバーライドに対するファイルの保護』を参照してください。

オーバーライドに対する終了の影響についてのシナリオ:

呼び出し操作からの終了 (ENDPGM、RETURN、または異常終了) は、その呼び出しレベルに有効範囲が限定されたオーバーライドを削除します。その呼び出しレベルで出されたオーバーライドでも、有効範囲が活動化グループ・レベルまたはジョブ・レベルに指定されたオーバーライドは削除されません。

たとえば、RETURN コマンドは有効範囲が呼び出しレベルに限定されたすべてのオーバーライドを削除します。したがって、RETURN コマンドまたは ENDPGM コマンドで終了する呼び出されたプログラムの中のオーバーライド (有効範囲が呼び出しレベルに限定されている) は、呼び出し側プログラムには適用されません。ただし、制御権伝送 (TFRCTL) コマンドを使用するプログラムの場合は異なります。

113 ページの図 7 では、RETURN コマンドによりプログラム B にある最初のオーバーライドが削除され、FILE X がプログラム A でオープンされます。しかしながら、二番目のオーバーライドがジョブ・レベルに有効範囲が指定されているので、RETURN コマンドはそのオーバーライドを削除しません。プログラム A で、OPEN FILE A コマンドが処理されると、FILE B がプログラム A の中でオープンされます。

注: コード例を使用することにより、お客様は 239 ページの『コードに関する特記事項』の条件に同意することになります。

```

プログラム A
.
.
.
CALL PGM(B)

          プログラム B
オーバーライド 1  OVRDBF FILE(X) FILE(Y)
オーバーライド 2  OVRDBF FILE(A) TOFILE(B) +
                  OVRSCOPE(*JOB)
.
.
.
RETURN

OPEN FILE X
.
.
.
OPEN FILE A

```

図7. オーバーライドに対する出口の影響の例： オーバーライドに対する出口の影響を概説する例。

オーバーライドに対する TFRCTL の影響についてのシナリオ:

TFRCTL コマンドは、あるプログラムを同一呼び出しレベルの別のプログラムによって置き換えます。制御権の転送先のプログラムは、TFRCTL コマンドが入っていたプログラムと同一呼び出しレベルで実行されます。制御権を別のプログラムに転送する元のプログラムの中のオーバーライド・コマンドは、制御権の転送の間に削除されません。

次の例では、プログラム A はプログラム B に制御権を転送し、プログラム B はプログラム A と同じ呼び出しレベルで実行されます。データベース・ファイル・オーバーライド (OVRDBF) コマンドによって、ファイルはオープンされてプログラム A およびプログラム B の両方に使用される時に、メンバーの最後のレコードに位置指定されます。

注: コード例を使用することにより、お客様は 239 ページの『コードに関する特記事項』の条件に同意することになります。

CALL PGM(A)
プログラム A

OVRDBF FILE(INPUT) POSITION(*END)

(INPUT はオープンされ、メンバーの最後のレコードに位置指定されて、
処理後クローズされます。) TFRCTL PGM(B)
プログラム B

(INPUT はオープンされ、メンバーの最後のレコードに位置指定されます。)

図 8. TFRCTL コマンドの例： 以下の図のように、TFRCTL コマンドは、あるプログラムを同一呼び出しレベルの別のプログラムに置き換えます。

同一呼び出しレベルの同一ファイルへのオーバーライド： シナリオ：

2 つのオーバーライドが同一ファイル名に対して同一呼び出しレベルで入力されると、2 番目のオーバーライドが最初のオーバーライドに置き換わります。これを利用すれば、最初のオーバーライドを削除しなくても、単一呼び出しレベルでオーバーライドを置き換えることができます。

次の例では、プログラムがファイル A をオープンしようとする時、オーバーライド 2 によってファイル A がファイル B にオーバーライドされます。各呼び出しレベルごとにオーバーライドは 1 つしか適用できませんから、サーバーはオーバーライド 1 を無視し、プログラムによってオープンされるファイルはファイル B になります。

注： コード例を使用することにより、お客様は 239 ページの『コードに関する特記事項』の条件に同意することになります。

```
          プログラム A
          .
          .
          .
オーバーライド 1   OVRDBF FILE(B) TOFILE(C)
オーバーライド 2   OVRDBF FILE(A) TOFILE(B)
          .
          .
          .
          OPEN FILE A
          .
          .
          .
```

図 9. オーバーライドを使ったオープン・ファイル・コマンドへのサーバー応答の例： 各呼び出しレベルごとに適用できるオーバーライドは 1 つだけです。

ファイル C をオープンするには、2 つのデータベース・ファイル・オーバーライド (OVRDBF) コマンドを、次のコマンドで置き換えてください。

```
OVRDBF FILE(A) TOFILE(C)
```


これによって、ファイルが作成されたときと同じ呼び出しレベルまたはジョブ・レベルでのオーバーライドの適用が妨げられることはありません。どの属性が先になるかにかかわらず、オーバーライドのファイル属性がファイル作成ステートメントの対応する属性にとって代わります。

関連概念

120 ページの『オーバーライドの削除』

呼び出されたプログラムが呼び出しプログラムに制御権を戻すと、呼び出されたプログラムの呼び出しレベルで指定されていたオーバーライドは、すべて削除されます。

CL プログラム・オーバーライド:

CL プログラムでファイルをオーバーライドしてから高水準言語プログラムを呼び出す場合、そのオーバーライドは高水準言語プログラムの場合にも有効のままになります。しかし、高水準言語プログラムで、ファイルをオーバーライドする CL プログラムを呼び出す場合は、制御権が高水準言語プログラムに戻ったときに、サーバーは自動的にオーバーライドを削除します。

オーバーライドに対するファイルの保護:

時には、自分のプログラムを呼び出す人またはプログラムによって、自分が指定したファイル名またはファイル属性を変更されたくない場合があります。保護を必要とする各ファイルごとに、ファイル・オーバーライド・コマンドで `SECURE(*YES)` パラメーターをコーディングすることによって、追加のファイル・オーバーライドを防止することができます。これにより、ファイルは低位の呼び出しレベル、活動化グループ・レベル、およびジョブ・レベルでのオーバーライドから保護されます。

- | 以下の例は、他のユーザーやプログラムが自分のファイルの名前または属性を変更できないようにする方法
- | を示しています。

注: コード例を使用することにより、お客様は 239 ページの『コードに関する特記事項』の条件に同意することになります。

```

オーバーライド 1  OVRPRTF FILE(PRINT1) SPOOL(*NO)

オーバーライド 2  OVRDBF FILE(NEWEMP) TOFILE(OLDEMP) +
                   MBR(N67)
                   CALL PGM(CHECK)

                   プログラム CHECK

オーバーライド 3  OVRDBF FILE(INPUT) +
                   TOFILE(NEWEMP) MBR(N77) +
                   SECURE(*YES)
                   CALL PGM(EREPORT)

                   プログラム EREPORT
                   (NEWEMP および PRINT1 がオープンされます。)

オーバーライド 4  OVRDBF FILE(INPUT) +
                   TOFILE(NEWEMP) MBR(N77)
                   CALL PGM(ELIST)

                   プログラム ELIST
                   (OLDEMP および PRINT1 がオープンされます。)

```

図 10. 保護ファイルの例： 保護ファイルは、自分のプログラムを呼び出す人またはプログラムによって、自分が指定したファイル名またはファイル属性を変更されたくない場合に使用します。

例では、プログラム EREPORT が呼び出されると、ファイル INPUT および PRINT1 をオープンしようと試みます。EREPORT は、実際にはファイル NEWEMP、メンバー N77 をオープンします。オーバーライド 3 で SECURE(*YES) が指定されているため、サーバーはオーバーライド 2 を適用しません。例では、プログラム ELIST が呼び出された場合も、ファイル INPUT および PRINT1 をオープンしようと試みます。ELIST は、実際にはファイル OLDEMP、メンバー N67 をオープンします。オーバーライド 4 はオーバーライド 3 と同じ名前を持ち、オーバーライド 3 と同じ呼び出しレベルであるため、オーバーライド 3 を置換します。こうして、ファイルは低い呼び出しレベルでのオーバーライドから保護されなくなり、サーバーは、プログラム ELIST にオーバーライド 2 を適用します。

PRINT1 が影響を受けるのはオーバーライド 1 だけであり、これはプログラム EREPORT および ELIST の両方に対して有効です。

プリンター・ファイルの総称によるオーバーライドの使用:

OVRPRTF コマンドを使用すると、ジョブの中のすべてのプリンター・ファイルに対して、同じ値のセットを用いて 1 回でオーバーライドを行うことができます。この総称によるオーバーライドがなければ、プリンター・ファイルのそれぞれに対して別々のオーバーライドを行わなければならないことになります。

*PRTF を使用して OVRPRTF コマンドを適用するシナリオについては、以下のトピックを参照してください。

例: *PRTF を指定した OVRPRTF の適用:

OVRPRTF コマンドは、*PRTF をファイル名として指定することによって、すべてのプリンター・ファイルに適用することができます。プリンター・ファイル名に対して同一呼び出しレベルで他にオーバーライドがない場合は、*PRTF 指定による OVRPRTF コマンドが適用されます。

注: コード例を使用することにより、お客様は 239 ページの『コードに関する特記事項』の条件に同意することになります。

以下の例は、*PRTF がどのように機能するかを示しています。

```
オーバーライド 1   OVRPRTF FILE(OUTPUT) COPIES(6) +
                    LPI(6)
オーバーライド 2   OVRPRTF FILE(*PRTF) COPIES(1) +
                    LPI(8)
                    CALL PGM(X)
```

図 11. OVRPRTF コマンドおよび *PRTF パラメーターの例: 以下の図は、OVRPRTF コマンドおよび *PRTF パラメーターの例です。

プログラム X でファイル OUTPUT をオープンすると、オープンされたファイルは次の属性を持つこととなります。

COPIES(6)

オーバーライド 1 から

LPI(6) オーバーライド 1 から

プログラム X でファイル PRTOUT (つまり OUTPUT 以外のプリンター・ファイル) をオープンすると、オープンされたファイルは次の属性を持つこととなります。

COPIES(1)

オーバーライド 2 から

LPI(8) オーバーライド 2 から

例: 複数の呼び出しレベルからの *PRTF 指定による OVRPRTF の適用:

このトピックの例では、複数の呼び出しレベルから *PRTF 値を使用して、どのようにプリンター・ファイル・オーバーライドが適用されるかを示しています。

注: コード例を使用することにより、お客様は 239 ページの『コードに関する特記事項』の条件に同意することになります。

```
                    プログラム A
オーバーライド 1   OVRPRTF FILE(*PRTF) COPIES(1)
オーバーライド 2   OVRPRTF FILE(PRT2) COPIES(2)
オーバーライド 3   OVRPRTF FILE(PRT4) COPIES(2)
                    CALL PGM(B)

                    プログラム B
オーバーライド 4   OVRPRTF FILE(*PRTF) LPI(4)
オーバーライド 5   OVRPRTF FILE(PRT3) LPI(8)
オーバーライド 6   OVRPRTF FILE(PRT4) LPI(8)
                    CALL PGM(X)
```

図 12. プリンター・ファイル・オーバーライドの例: *PRTF 値を使用することによって、複数呼び出しレベルからプリンター・ファイル・オーバーライドを適用できます。

プログラム X でファイル PRT1 をオープンすると、オープンされたファイルは次の属性を持つことになります。

COPIES(1)

オーバーライド 1 から

LPI(4) オーバーライド 4 から

PRT1 に対する特定のオーバーライドが見つからないため、*PRTF オーバーライド (1 および 4) が適用されます。

プログラム X でファイル PRT2 をオープンすると、オープンされたファイルは次の属性を持つことになります。

COPIES(2)

オーバーライド 2 から

LPI(4) オーバーライド 4 から

プログラム B では PRT2 に関して特定のオーバーライドが見つからないため、オーバーライド 4 が適用されます。プログラム A ではオーバーライド 2 が PRT2 に対して指定されており、このオーバーライドが適用されます。

プログラム X でファイル PRT3 をオープンすると、オープンされたファイルは次の属性を持つことになります。

COPIES(1)

オーバーライド 1 から

LPI(8) オーバーライド 5 から

プログラム B ではオーバーライド 5 が PRT3 に対して指定されており、このオーバーライドが適用されます。プログラム A では PRT3 に関しては特定のオーバーライドが見つからないため、オーバーライド 1 が適用されます。

プログラム X でファイル PRT4 をオープンすると、オープンされたファイルは次の属性を持つことになります。

COPIES(2)

オーバーライド 3 から

LPI(8) オーバーライド 6 から

プログラム B では、オーバーライド 6 が PRT4 に対して指定されており、オーバーライド 6 が適用されます。プログラム A では、オーバーライド 3 が PRT4 に対して指定されており、オーバーライド 3 が適用されます。

プログラムのコンパイル時におけるオーバーライドの適用:

プログラムのコンパイル時にオーバーライドを適用する目的は、ソース・ファイルを選択するため、あるいは、入出力操作に使用するレコード様式を定義するためにコンパイラーが使用する外部データ定義を提供するためです。

ソース・ファイルのオーバーライドは、他のオーバーライドと同様に処理されます。オーバーライドでは、他のファイル、データベース・ファイルの他のメンバー、ディスクまたはテープの他のラベルを選択したり、その他のファイル属性を変更したりする場合があります。

コンパイルされているプログラム内で使用されるファイルが、プログラムの中で外部記述ファイルとして使用されている場合、そのようなファイルにもオーバーライドを適用することができます。これらのファイルはコンパイル時にはオープンされないため、オーバーライドは通常の方法では適用できません。オーバーライドがコンパイル時に使用されるのは、ファイル名、およびプログラムで入出力操作を使用する場合にレコード様式およびフィールドを定義するのに使用されるライブラリーを決めるためです。その他のファイル属性がオーバーライドで指定されていても、コンパイル時には無視されます。プログラムのソース・ステートメントの中で指定されているファイル名が、アプリケーションで必要とするレコード様式が入っているファイル名ではない場合に限り、コンパイル時にファイル・オーバーライドが活動状態になることが必要です。

コンパイルされたプログラムの実行時にオープンされるファイル名は、プログラム・ソース・ステートメントで参照するファイル名によって決まり、プログラムの実行時に有効なオーバーライドがあればそれによって変更されます。この場合、コンパイル時に使用されるファイル名は保持されません。実際にオープンされるファイルの中のレコード様式は、プログラムのコンパイル時に使用されたレコード様式と互換性がなければなりません。レコード互換性を確保する最も容易な方法は、コンパイル時に活動状態にあったものと同じオーバーライドを実行時に活動状態にすることであるのは明らかです。プログラムで外部記述データを使用し、異なるフィールド・レベル・ファイルが実行時に使用される場合は、オーバーライドで LVLCHK(*NO) を指定する必要があるのが普通です。詳細については、128 ページの『ファイルの指定変更』を参照してください。

次の例では、プログラムのコンパイル時にどのようにオーバーライドが行われるかを示しています。

```

オーバーライド 1   OVRDBF FILE(RPGSRC) +
                   TOFILE(SRCPGMS) MBR(INVN42)
オーバーライド 2   OVRPRTF FILE(OUTPUT) TOFILE(REPORTS)
                   CALL PGM(A)

                   プログラム A
オーバーライド 3   OVRPRTF FILE(LISTOUT) +
                   TOFILE(OUTPUT)
オーバーライド 4   OVRDBF FILE(RPGSRC) WAITFILE(30)
                   CRTRPGPGM PGM(INVENTORY) +
                   SRCFILE(RPGSRC)
                   RETURN
オーバーライド 5   OVRPRTF FILE(LISTOUT) +
                   TOFILE(REPORTS) LPI(8)
                   CALL PGM(INVENTORY)

```

図 13. プログラムのコンパイル時のオーバーライドの例： プログラムのコンパイル時にどのようにオーバーライドが行われるかを示しています。

プログラム INVENTORY では、プリンター・ファイル LISTOUT ではなくプリンター・ファイル REPORTS をオープンし、8 行/インチで出力を作成します。

プログラム INVENTORY は、データベース・ファイル SRCPGMS 中のメンバー INVN42 から作成 (コンパイル) されます。オーバーライド 4 は最初に適用され、オプション・ファイルの属性をオーバーライドします。オーバーライド 1 は最後に適用され、ファイル RPGSRC をデータベース・ファイル SRCPGMS のメンバー INVN42 にオーバーライドします。

プログラム INVENTORY は、ファイル REPORTS からのプリンター様式で作成されます。プログラム INVENTORY のソース (ファイル SRCPGMS およびメンバー INVN42 から取り出された) には、プリン

ター・ファイル LISTOUT に対するオープンが含まれていると仮定します。オーバーライド 3 は最初に適用され、ファイル LISTOUT が OUTPUT にオーバーライドされます。オーバーライド 2 は最後に適用され、OUTPUT が REPORTS にオーバーライドされます。他の属性もここで指定できますが、コンパイル時に使用されるのはレコード様式だけですから、その必要はありません。

実行時にはプログラム A は終了しているため、オーバーライド 3 はもう活動していません。したがって、オーバーライド 2 は LISTOUT に対して影響していません。しかし、オーバーライド 5 が実行時に活動していて、LISTOUT を REPORTS で置き換え、1 インチ当たり 8 行を指定します。同じファイルがコンパイル時と実行時に使用されるため、レベル検査はそのままにしておくことができます。

関連資料

99 ページの『一部のコマンドに対するオーバーライドの影響』

コマンドの中には、オーバーライドをまったく無視するものや、特定のパラメーターのオーバーライドだけを許可するものがあります。

オーバーライドの削除

呼び出されたプログラムが呼び出しプログラムに制御権を戻すと、呼び出されたプログラムの呼び出しレベルで指定されていたオーバーライドは、すべて削除されます。

有効範囲が活動化グループ・レベルまたはジョブ・レベルに限定されたオーバーライドは、これには該当しません。有効範囲が活動化グループ・レベルに限定されたオーバーライドは、明示的に削除、置換されるか、または指定された活動化グループが削除されるまで有効です。有効範囲がジョブ・レベルに限定されたオーバーライドは、明示的に削除、置換されるか、またはオーバーライドが指定されたジョブが終了するまで有効です。

制御権が別のプログラムに転送 (TFRCTL コマンド) される時、転送プログラムの呼び出しレベルの中のオーバーライドは削除されません。

また、オーバーライド削除 (DLTOVR) コマンドを使用して、サーバー上でオーバーライドを明示的に削除することもできます。DLTOVR コマンドにより、呼び出しレベル、活動化グループ・レベル、またはジョブ・レベルに有効範囲が限定されるオーバーライドを削除することができます。活動化グループ・レベルに有効範囲が限定されたオーバーライドを削除するには、OVRSCOPE(*ACTGRPDFN) がデフォルトであるため、OVRSCOPE パラメーターに値を指定する必要はありません。有効範囲がジョブ・レベルに限定されたオーバーライドを削除するには、DLTOVR コマンドに OVRSCOPE(*JOB) を指定する必要があります。

オーバーライドを識別するには、オーバーライド・コマンドの FILE パラメーターで指定されているファイルを使用します。FILE パラメーターに値 *ALL を指定することにより、現在のレベル (呼び出しレベル、活動化グループ・レベル、またはジョブ・レベル) すべてのオーバーライドを削除することができます。

関連概念

114 ページの『同一呼び出しレベルの同一ファイルへのオーバーライド: シナリオ』

2 つのオーバーライドが同一ファイル名に対して同一呼び出しレベルで入力されると、2 番目のオーバーライドが最初のオーバーライドに置き換わります。これを利用すれば、最初のオーバーライドを削除しなくても、単一呼び出しレベルでオーバーライドを置き換えることができます。

オーバーライドの表示

オーバーライド表示 (DSPOVR) コマンドを使用すると、ジョブ・レベル、活動化グループ・レベル、および 1 つのジョブに対する複数の呼び出しレベルごとに、ファイル・オーバーライドを表示することができます。すべてのファイル・オーバーライドも表示できますし、個々のファイルに関するファイル・オーバーライドだけを表示することもできます。

ファイル・オーバーライドは、組み合わせてから表示することができます。この組み合わせオーバーライドは、ジョブ・レベルから現レベルまたは任意の指定呼び出しレベルまでのオーバーライドを組み合わせた結果で、ファイルが特定の呼び出しレベルで使用される時に適用される、複合オーバーライドを作ります。現行の呼び出しレベルは、現在実行中のプログラムの呼び出しレベルです。このプログラムは、呼び出しスタック上に表示される最後のプログラム名です。このコマンドは、バッチ環境と対話式環境のどちらからでも要求することができます。また、この機能は、ジョブ処理メニュー (WRKJOB コマンドを使用) のオプション 15 (ファイル・オーバーライドの表示) からでも、あるいはジョブ表示メニュー (DSPJOB コマンドを使用) でオプション 15 (ファイル・オーバーライドの表示) を選択することによってもアクセスできます。

関連情報

DSPOVR コマンド

例: 特定の活動化グループに対するすべてのオーバーライドの表示: シナリオ: 特定の活動化グループに対するすべてのオーバーライドを表示するには、次のように入力します。

```
DSPOVR FILE(REPORTS) ACTGRP(*)
```

これにより、オーバーライドが出される活動化グループの REPORTS ファイルに対してのすべてのオーバーライドが表示されます。ACTGRP(*) はデフォルトですが、ここでは例として示されています。コマンドが出された活動化グループ以外の活動化グループを指定するには、ACTGRP パラメーターに活動化グループの名前を指定してください。

例: 1 つのファイルに対する組み合わせファイル・オーバーライドの表示: 個々のファイルの、指定の呼び出しレベルにおける組み合わせファイル・オーバーライドを表示するには、次のように入力します。

```
DSPOVR FILE(REPORTS) MRGOVR(*YES) LVL(3)
```

このコマンドでは、呼び出しレベル 3 でファイル REPORTS に関して組み合わせられたオーバーライドを示す表示画面が、各キーワードとパラメーターのテキスト記述で生成されます。ジョブ・レベル、活動化グループ・レベル、および呼び出しレベル 1、2、および 3 で該当するオーバーライドは、いずれも組み合わせオーバーライドを形成するのに使用されますが、それよりも上位の呼び出しレベルのオーバーライドは無視されます。指定された呼び出しレベルが活動状態でない場合は、現行レベルまでの該当するオーバーライドがすべて使用されます。

例: 1 つのファイルに対するすべてのファイル・オーバーライドの表示: 指定の呼び出しレベルまでにおける個々のファイルに対するすべてのファイル・オーバーライドを表示するには、次のように入力します。

```
DSPOVR FILE(REPORTS) MRGOVR(*NO) LVL(2)
```

このコマンドでは、ファイル名、オーバーライドが要求された呼び出しレベル、オーバーライドのタイプ、およびキーワード・パラメーター様式でオーバーライド・パラメーターを示す表示画面を生成します。ファイル・オーバーライドが、ファイルに関して指定された呼び出しレベル以下で見つからない場合は、エスケープ・メッセージ CPF9842 が送られます。CL プログラムの中で DSPOVR を使用している場合は、ファイルに関してオーバーライドがない場合のプログラム打ち切りを防止するために、DSPOVR コマンドに続けて MONMSG コマンドを追加することができます。この技法については、この章の後の部分で登場するいくつかの例の中で図示されています。

関連情報

制御言語 (CL)

例: すべてのファイルに対する組み合わせファイル・オーバーライドの表示: すべてのファイルの、現行の呼び出しレベルにおける組み合わせファイル・オーバーライドを表示するには、次のように入力します。

```
DSPOVR FILE(*ALL) MRGOVR(*YES) LVL(*)
```

このコマンドでは、ファイル名、オーバーライドのタイプ、およびコマンドで入力されたキーワードとパラメーターだけがキーワード・パラメーター様式で示された組み合わせオーバーライドを示す表示画面を生成します。これは、パラメーターを指定しないで **DSPOVR** を入力した時に生じるものと同じです。表示されるのは、パラメーターが指定されたキーワードだけです。関連したテキスト記述は表示されません。999より大きな呼び出しレベルのオーバーライドは、表示されません。

例: WRKJOB を使用したオーバーライドの表示: オーバーライドが **DSPOVR** コマンドによってではなく、ジョブを処理するシステム・インターフェースの1つでオプション (たとえば、**WRKJOB**) を用いて表示されるときは、ジョブ・レベルから現行レベルまでのすべてのファイル・オーバーライドが表示されません。これは、次のように入力した場合と同じになります。

```
DSPOVR FILE(*ALL) MRGOVR(*NO) LVL(*)
```

これにより、ファイル名、オーバーライドが要求されたレベル (呼び出しレベル、活動化グループ・レベル、またはジョブ・レベル)、オーバーライドのタイプ、および各オーバーライドに関するオーバーライド・パラメーターをキーワード・パラメーター様式で示す表示画面が生成されます。

オーバーライド表示機能では内部制御ブロックのコピーを使用するため、オーバーライド表示機能が呼び出された時と出力が生成された時の間に削除されたオーバーライドは、出力に反映されない可能性があります。これは、他のジョブでのオーバーライドが表示されている場合にのみ起こります。

包括的な例: オーバーライドの表示: 次の例は、さまざまな様式のオーバーライド表示コマンドが実行できることを示すためのものです。**DSPOVR** コマンドは、対話式に入力するか、あるいはまた **CL** プログラム、または **QCMDXEC** を介して任意の高水準言語プログラムに一時的に追加し、プログラムの呼び出し時またはファイルのオープン時に、適正なオーバーライドが有効になるかを確認検査するために使用されるのが普通です。コマンド 1、2、3、および 18 は、呼び出しレベル 1 で入力されるものと想定します。

注: コード例を使用することにより、お客様は 239 ページの『コードに関する特記事項』の条件に同意することになります。


```

プログラム A (ユーザー・デフォルト活動化グループ内)
コマンド 1  OVRPRTF FILE(PRTA) COPIES(3)
コマンド 2  OVRDBF FILE(DBC) WAITFILE(*IMMED)
コマンド 3  CALL PGM(B)

          プログラム B (活動化グループ 5 内)
コマンド 4  OVRPRTF FILE(PRTB) TOFILE(PRTA) COPIES(6) +
          OVRSCOPE(*CALLLVL)
コマンド 5  OVRDBF FILE(DBC) WAITFILE(60) OVRSCOPE(*CALLLVL)
コマンド 6  OVRDBF FILE(DBE) TOFILE(DBF) OVRSCOPE(*JOB)
コマンド 7  DSPOVR FILE(PRTB) MRGOVR(*YES)
コマンド 8  CALL PGM(C)

          プログラム C (活動化グループ 5 内)
コマンド 9  CALL PGM(QCMDEXC) PARM('OVRDSPF FILE(DSPE) +
          TOFILE(DSPF) OVRSCOPE(*CALLLVL)' 50)
コマンド 10 OVRDBF FILE(DBC) TOFILE(DBD) OVRSCOPE(*CALLLVL)
コマンド 11 DSPOVR FILE(DBC) MRGOVR(*NO) LVL(3)
コマンド 12 DSPOVR FILE(DBD) MRGOVR(*NO) LVL(2)
コマンド 13 MONMSG MSGID(CPF9842)
コマンド 14 OVRDSPF FILE(CREDITS) TOFILE(DEBITS)
コマンド 15 CALL PGM(QCMDEXC) PARM('DSPOVR FILE(*ALL) MRGOVR(*YES) +
          LVL(*) OUTPUT(*)' 47)
コマンド 16 RETURN

コマンド 17 DSPOVR FILE(*ALL) MRGOVR(*NO)
コマンド 18 RETURN
コマンド 19 DSPOVR FILE(*ALL) MRGOVR(*NO) LVL(2) OUTPUT(*)

```

図 14. オーバーライド表示の例： この例は、オーバーライドを表示する方法を概説しています。

コマンド 1 では、ファイル PRTA の COPIES 属性の値を、レベル 1 で 3 にオーバーライドさせます。

コマンド 2 では、ファイル DBC の WAITFILE 属性を、レベル 1 で *IMMED にオーバーライドします。

コマンド 3 では、プログラム B を呼び出して新しい呼び出しレベル 2 を作成します。

コマンド 4 では、ファイル PRTB からファイル PRTA へ、レベル 2 でオーバーライドします。また、このコマンドは、COPIES 属性の値を 6 にオーバーライドします。

コマンド 5 では、ファイル DBC の WAITFILE 属性の値を、レベル 2 で 60 にオーバーライドさせます。

コマンド 6 では、ファイル DBE からファイル DBF へオーバーライドし、オーバーライドの有効範囲はジョブ・レベルに限定されます。

コマンド 7 では、ファイル PRTB に関するレベル 2 の組み合わせオーバーライドを、124 ページの図 15 に示すように、各キーワードとパラメーターのテキスト記述で表示します。オーバーライド後のファイルはコマンド 4 で PRTA になり、COPIES 属性はコマンド 1 で 3 になります。

表示 印刷装置ファイル一時変更

```
ファイル . . . . . : PRTB
呼び出しレベル . . . . . : *
組合せ . . . . . : *YES

一時変更中のファイル . . . . . : FILE      キーワード  値
印刷装置ファイルへの一時的変更 . . . . . : TOFILE   PRTB
ライブラリー . . . . . :                PRTA
コピー枚数 . . . . . : COPIES      *LIBL
                                         3
```

続行するには、Enter キーを押してください。

F3= 終了 F12= 取り消し

図 15. 印刷装置ファイル一時変更の表示： この例は、プリンター・ファイル・オーバーライドの表示コマンドを示しています。

コマンド 8 では、プログラム C を呼び出して、新しい呼び出しレベル 3 を作成します。

コマンド 9 では、ファイル DSPE からファイル DSPF に、レベル 3 でオーバーライドします。QCMDEXC プログラムに対する呼び出しを介して行われるオーバーライドでは、QCMDEXC プログラムを呼び出したプログラムの呼び出しレベルを取ります。

コマンド 10 では、ファイル DBC からファイル DBD へオーバーライドが生じます。

コマンド 11 では、ジョブ・レベルからレベル 3 までのファイル DBC に関するすべてのオーバーライドを、125 ページの図 16 に示すように、表示します。コマンド 10、5、および 2 によって指定されたオーバーライドは、キーワード・パラメーター様式で表示されます。この様式の DSPOVR コマンドでは、ファイル指定変更に関係なく、選択されたファイルに関するすべてのオーバーライドを示します。表示されている 3 つのオーバーライドは、レベル 3 における名前変更のため、組み合わせられません。

すべてのファイル一時変更の表示

呼び出しレベル : 3

オプションを入力して、Enter キーを押してください。

5= 一時変更明細の表示

OPT	ファイル	レベル	タイプ	キーワード指定
-	DBC	3	DB	TOFILE(*LIBL/DBD)
-		2	DB	WAITFILE(60)
-		1	DB	WAITFILE(*IMMED)

F3= 終了 F5= 最新表示 F12= 取り消し

図 16. すべてのファイル一時変更の表示 (1 つのファイル)：この例は、すべてのファイル・オーバーライドの表示を説明しています。

コマンド 12 では、ファイル DBD に関するジョブ・レベルからレベル 2 のすべてのファイル・オーバーライドを表示しようと試みます。ファイル DBD に関するオーバーライドはレベル 1 でもレベル 2 でも存在しないため、オーバーライドが見つからない旨のエスケープ・メッセージ (CPF9842) が送られます。

コマンド 13 では、前のコマンドでのメッセージ CPF9842 をモニターします。モニターでは取るべき処置は指定しませんが、メッセージが送られる場合は機能チェックを防止します。

コマンド 14 では、ディスプレイ・ファイル CREDITS からディスプレイ・ファイル DEBITS へオーバーライドが生じます。オーバーライドの有効範囲は、活動化グループ 5 の活動化グループ・レベルに限定されています。OVRSCOPE(*ACTGRPDEFN) がデフォルトとなります。

コマンド 15 では、すべてのファイルに関するジョブ・レベルからレベル 3 の組み合わせオーバーライドを、126 ページの図 17 で示すように、キーワード・パラメータ様式で表示します。ファイル DBC は、コマンド 10 でファイル DBD にオーバーライドされます (したがって、コマンド 5 およびコマンド 2 は無効)。ファイル DSPE は、コマンド 9 でファイル DSPF にオーバーライドされます。ファイル PRTB は、コマンド 4 および 1 でファイル PRTA および COPIES(3) にオーバーライドされます。ファイル DBE は、コマンド 6 でファイル DBF にオーバーライドされます。ファイル DEBITS は、コマンド 14 でファイル CREDITS にオーバーライドされます。

すべての組合せファイル一時変更の表示

呼び出しレベル : *

オプションを入力して、Enter キーを押してください。

5= 一時変更明細の表示 8= 関係するファイル一時変更の表示

OPT	ファイル	タイプ	キーワード指定
-	DSPE	DSP	TOFILE(*LIBL/DSPF)
8	PRTB	PRT	TOFILE(*LIBL/PRTA) COPIES(3)
-	DBC	DB	TOFILE(*LIBL/DBD)
-	PRTA	PRT	COPIES(3)
-	DBE	DB	TOFILE(*LIBL/DBF)
-	CREDITS	DSPF	TOFILE(*LIBL/DEBITS)

F3= 終了 F5= 最新表示 F11= すべてのファイル一時変更 F12= 取り消し

図 17. すべての組み合わせファイル一時変更の表示： この例は、すべての組み合わせファイル・オーバーライドの表示を説明しています。

PRTB の行に 5 と入力すれば、124 ページの図 15 のような明細画面が表示されます。同じ行に 8 と入力すれば、図 18 に示すように、コマンド 4 と 1 を別の行に示す画面が表示されます。これらのオーバーライドは、組み合わせられて PRTB オーバーライドを形成します。

関係するファイル一時変更の表示

ファイル : PRTB
呼び出しレベル : *

オプションを入力して、Enter キーを押してください。

5= 一時変更明細の表示

OPT	レベル	タイプ	キーワード指定
-	2	PRT	TOFILE(*LIBL/PRTA) COPIES(6)
-	1	PRT	COPIES(3)

F3= 終了 F5= 最新表示 F12= 取り消し F14= 前の一時変更の表示

図 18. 関係するファイル一時変更の表示： この例は、関係するファイル・オーバーライドの表示を説明しています。

コマンド 16 によって、レベル 2 へ戻り、レベル 3 は削除されます。レベル 3 で発行された、有効範囲が呼び出しレベルに限定されているオーバーライドは、暗黙的に削除されます。コマンド 14 で発行されたオーバーライドは有効範囲が活動化グループ・レベルに限定されているので、削除されません。

コマンド 17 では、ジョブ・レベルから現呼び出しレベル (レベル 2) までで出されたすべてのオーバーライドを、図 19 で示すように表示します。コマンド 1、2、4、5、6、および 14 で指定されたオーバーライドが、キーワード・パラメーター様式で表示されます。呼び出しレベル 3 はもはや活動状態ではないので、コマンド 10 で出されたオーバーライドは表示されません。この画面で F11 を押すと、126 ページの図 17 で示される画面に似た画面を表示することができます。

すべてのファイル一時変更の表示

呼び出しレベル : *

オプションを入力して、Enter キーを押してください。
5= 一時変更明細の表示

OPT	ファイル	レベル	タイプ	キーワード指定
-	CREDITS	*ACTGRP	PRT	TOFILE(*LIBL/DEBITS)
-	PRTB	2	PRT	TOFILE(*LIBL/PRTA) COPIES(6)
-	DBC	2	DB	WAITFILE(60)
-		1	DB	WAITFILE(*IMMED)
-	PRTA	1	PRT	COPIES(3)
-	DBE	*JOB	DB	TOFILE(*LIBL/DBF)

F3= 終了 F5= 最新表示 F11= すべての組合せ済みファイル一時変更 F12= 取り消し

図 19. すべてのファイル一時変更の表示 (すべてのファイル)： この例は、すべてのファイル・オーバーライドの表示を説明しています。

コマンド 18 によって、レベル 1 へ戻り、レベル 2 は削除されます。レベル 2 で出された、有効範囲が呼び出しレベルに限定されているオーバーライドは、暗黙的に削除されます。コマンド 14 で行われたオーバーライド (有効範囲が活動化グループに限定されている) は、活動化グループ 5 が終了したときに暗黙的に削除されます。この例では、活動化グループ 5 は非持続性の活動化グループで、コマンド 18 が処理されたときに終了すると想定されています。ただし、コマンド 6 で生じたオーバーライドは削除されません。

コマンド 19 では、ジョブ・レベルから呼び出しレベル 2 で出されたすべてのオーバーライドが、キーワード・パラメーター様式で表示されます。レベル 2 はもう活動状態ではないため、有効範囲がジョブ・レベル (コマンド 6) に限定されたオーバーライドおよびコマンド 1 および 2 においてレベル 1 で指定されたオーバーライドだけが表示されます。

オーバーライドの表示: ヒント: この節の最初の 2 つの例のように、呼び出しレベルを指定するとき、オーバーライド・コマンドを最初に入力した呼び出しレベルはレベル 1 でない可能性があることに注意してください。ユーザー・プロファイル内で指定されている最初のプログラムと最初のメニューの内容、および他のプログラムまたはメニューの内容によっては、レベル 3 またはレベル 4 で最初のオーバーライド・コマンドが入力された可能性があります。より低い呼び出しレベルでどのプログラムが実行中かを確認するには、WRKJOB を入力してオプション 11 (呼び出しスタック) を選択することができます。

表示したいものが正確にわからない場合は、普通、パラメーターを指定しないでオーバーライド表示画面を要求するのが賢明です。その理由は、基本オーバーライド表示画面上のオプションを使用すれば、必要なオーバーライドの明細表示画面を選択することができるからです。次の特定のオプションを使用することができます。

- すべてのオーバーライドの組み合わせ表示画面で、122 ページの『例: WRKJOB を使用したオーバーライドの表示』のように、組み合わせていない表示画面を要求することができます。
- 組み合わせられていないすべてのオーバーライドの表示画面で、組み合わせ表示画面を要求することができます。
- すべてのオーバーライドの組み合わせ表示画面で、任意のオーバーライドについて、121 ページの『例: 1 つのファイルに対する組み合わせファイル・オーバーライドの表示』にあるコマンドに相当する組み合わせ明細画面を要求することができます。
- すべてのオーバーライドの組み合わせ画面で、組み合わせ表示画面に含まれるそれぞれのオーバーライドのすべてについて、それぞれが要求されたレベル (呼び出しレベルまたはジョブ・レベル) を示す画面を要求することができます。
- 含まれるオーバーライドの表示画面かすべてのオーバーライドの (組み合わせでない) 表示画面のいずれかで、特定のファイルに関する単一呼び出しレベルのオーバーライドの明細表示画面を要求することができます。

ファイルの指定変更

ファイル指定変更とは、オーバーライドを使用して、データ入力またはデータ出力を異なるタイプの装置に指定することです。たとえば、ディスクに指定したデータの代わりにプリンターに指定変更します。ただし、オーバーライドをこのように使用することについては、関連する2つの装置のそれぞれ異なる特性をプログラムで適応させることができなければならないため、上記のリストに示したオーバーライド用例の場合に比べて注意が必要になります。

異なるタイプのファイルをオーバーライドするには、新しいタイプのファイルのオーバーライド・コマンドを使用してください。たとえば、ディスク・ファイルをプリンター・ファイルにオーバーライドする場合は、プリンター・ファイル・オーバーライド (OVRPRTF) コマンドを使用してください。

この節は、アプリケーション・プログラムを使用する場合にのみ適用されます。サーバー・コードがファイルの指定変更をサポートする場合と、サポートしない場合があります。サーバー・コードが処理するオーバーライドの規則については、99 ページの『一部のコマンドに対するオーバーライドの影響』を参照してください。

また、OVRDBF コマンドを使用して、あるファイルを分散データ管理 (DDM) ファイルに指定変更することもできます。リモート・サーバーが別の iSeries サーバーである場合は、この章で述べられている通常の規則はすべて適用されます。リモート・サーバーが iSeries サーバーでも System/38™ でもない場合は、通常は、満了日またはファイル終了遅延を指定すべきではありません。

あるプログラムの中で使用されるファイルを同一タイプの別のファイルで置き換えると、新しいファイルは元のファイルと同じ方法で処理されます。フィールド・レベル・ファイル、または外部記述データを含む他のファイルを指定変更する場合、通常は、LVLCHK(*NO) を指定するか、プログラムを再コンパイルする必要があります。レベル検査をオフにする場合でも、ファイルには、プログラムのレコードと互換性のあるレコード様式を使用する必要があります。レコード様式に互換性がない場合、結果は予測することができません。

*FILE 以外の TOFILE パラメーター値を持つオーバーライドでは、高位呼び出しレベルで適用されるデータベース・メンバー仕様を除去します。メンバー名は、ファイル名またはライブラリーに対する変更を伴うか、または低位の呼び出しレベルにおける別のオーバーライドで指定されない限り、デフォルトとして *FIRST を取ります。

異なるタイプのファイルに変更すると、装置依存特性、およびサーバーが順次に読み取りまたは書き出しするレコードは無視されます。一部の装置パラメーターについては、新しい装置ファイルの中またはオーバー

ライドの中で指定しなければならない場合もあります。その他については、サーバーはデフォルトを使用します。特定の指定変更組み合わせの影響については、この節で後述します。

最終ファイル・タイプとは異なるファイル・タイプのオーバーライドで指定された属性は、いずれも無視されます。この規則の例外は、パラメーター **SPOOL**、**SHARE**、および **SECURE** です。サーバーは、装置タイプには関係なく、ファイルに適用されるどんなオーバーライドからのパラメーターも受け入れます。

ファイルの指定変更についての詳細は、以下のトピックを参照してください。

関連概念

102 ページの『オーバーライドの適用』

実行できるオーバーライドには、ファイル・オーバーライドとプログラム装置項目オーバーライドという 2 つの一般的なタイプがあります。

関連情報

分散データ管理

ファイル指定変更の計画:

このトピックでは、ファイル指定変更の有効な組み合わせを表に示します。

この表を使用するには、オーバーライドされるファイル・タイプを「変更元ファイル」欄で、オーバーライドした後のファイル・タイプを「変更先ファイル欄」で識別してください。両欄の交わる部分に **I** または **O** または両方が指定されています。これは、これらの 2 つのファイル・タイプの間で入力ファイルまたは出力ファイルとして使用される時、置き換えが有効であることを意味します。

たとえば、ディスク出力ファイルをテープ出力ファイルに、ディスク入力ファイルをテープ入力ファイルにオーバーライドすることができます。この表は、ファイル・タイプの置換についてのみ扱ったものです。つまり、入力ファイルを出力ファイルにオーバーライドすることによって、プログラム機能を変更することはできません。

表 12. ファイル指定変更: 有効なファイル指定変更を、次の表で要約しています。

コピー先ファイル	コピー元ファイル					
	プリンター	システム間通信機能 (ICF)	ディスク	ディスプレイ	データベース	テープ
プリンター	O*	O	O	O	O	O
ICF	O	I/O O I	O I	I/O O I	O I	O I
ディスク	O	O I	O I	O I	O I	O I
ディスプレイ	O	I/O O I	O I	I/O O I	O I	O I
データベース	O	O I	O I	O I	O I	O I
テープ	O	O I	O I	O I	O I	O I

注:

- I= 入力ファイル O= 出力ファイル I/O= 入出力ファイル
- *= 別のタイプのプリンターへの指定変更

ファイルの指定変更: ヒント:

指定変更の組み合わせによっては、装置の特定特性のために特殊な問題が生じる場合があります。

特に、以下の点に注意してください。

- 保管ファイルは指定変更しないでください。
- 順序外処理データベース・ファイルが指定変更できるのは、別のデータベース・ファイルまたは DDM ファイルだけです。
- 複数の装置 (MAXDEV または MAXPGMDEV > 1) を使用するディスプレイ・ファイルまたは ICF ファイルは、ディスプレイ・ファイルまたは ICF ファイルにだけ指定変更できます。
- ディスプレイ・ファイルを他のファイル・タイプに、または別のファイル・タイプをディスプレイ・ファイルに指定変更するには、入力専用または出力専用フィールドがある場合はオーバーライドを活動状態にして、プログラムを再コンパイルすることが必要になります。ディスプレイ・ファイルでは、これらのフィールドが使用しないレコード・バッファからこれらのフィールドは省かれますが、他のファイル・タイプではそのようなことはないからです。

指定変更されたファイルに対するデフォルトのアクション:

このトピックでは、サーバーがファイルを指定変更するとき取る特定のデフォルトについて、および指定変更のそれぞれの組み合わせでどのデフォルトがサーバーによって無視されるかについて説明します。

コピー元

プリンター

コピー先

ICF: レコードは一度に 1 つずつファイルに書き込まれます。プリンター制御情報は無視されます。

ディスプレイ: レコードは、それぞれ前のレコードに重ね書きされながら、ディスプレイに書き込まれます。プログラムで記述されたファイルの場合、Enter キーを使用して各レコードを要求することができます。プリンター制御情報は無視されます。

データベース: レコードは、順次にデータベースに書き込まれます。プリンター制御情報は無視されます。

ディスク: ディスク上に書き込まれるデータ量は、ディスクの交換タイプによって異なります。ディスク・ラベル情報は、ディスク・ファイルの中またはオーバーライド・コマンドで指定されなければなりません。プリンター制御情報は無視されます。

テープ: レコードは、順次にテープに書き込まれます。テープ・ラベル情報は、テープ・ファイルの中またはオーバーライド・コマンドで指定されなければなりません。プリンター制御情報は無視されます。

コピー元

ICF 入力

コピー先

ディスプレイ: レコードは、一度に 1 つずつディスプレイから取り出されます。各レコードのデータを入力し、レコードが完成したら、Enter キーを押してください。

データベース: レコードは、データベースから取り出されます。

ディスク: レコードは、順次に取り出されます。ディスク・ラベル情報は、ディスク・ファイルの中またはオーバーライド・コマンドで指定されなければなりません。

テープ: レコードは、順次に取り出されます。テープ・ラベル情報は、テープ・ファイルの中またはオーバーライド・コマンドで指定されなければなりません。

コピー元

ICF 出力

コピー先

プリンター: レコードは印刷され、改行または切り捨ては、プリンター・ファイルに指定されているとおりに実行されます。

ディスプレイ: レコードは、それぞれ前のレコードに重ね書きされながら、ディスプレイに書き込まれます。

データベース: レコードは、順次にデータベースに書き込まれます。

ディスク: ディスク上書きされるデータ量は、ディスクの交換タイプによって異なります。ディスク・ラベル情報は、ディスク・ファイルの中またはオーバーライド・コマンドで指定されなければなりません。

テープ: レコードは、順次にテープに書き込まれます。テープ・ラベル情報は、テープ・ファイルの中またはオーバーライド・コマンドで指定されなければなりません。

コピー元

ICF 入出力

コピー先

ディスプレイ: 入力レコードは、ディスプレイから一度に 1 つずつ取り出されます。各レコードのデータを入力し、レコードが完成したら、Enter キーを押してください。出力レコードは、前の入力または出力レコードに重ね書きされながらディスプレイに書き込まれます。入力レコードと出力レコードは本来、相互に独立しているので任意に組み合わせることができます。

コピー元

ディスク入力

コピー先

ICF: レコードは、一度に 1 つずつ ICF ファイルから取り出されます。

ディスプレイ: レコードは、一度に 1 つずつディスプレイから取り出されます。各レコードのデータを入力し、レコードが完成したら、Enter キーを押してください。非フィールド・レベル装置ファイルが指定されなければなりません。ディスクのラベル情報は無視されます。

データベース: レコードは、順次に取り出されます。ディスクのラベル情報は無視されます。

テープ: レコードは、順次に取り出されます。ラベル値がプログラムの中で指定されている場合は、その値がテープ・ファイルのラベルとして使用されます。

コピー元

ディスク出力

コピー先

ICF: レコードは、一度に 1 つずつ ICF ファイルに書き込まれます。

データベース: レコードは、順次にデータベースに書き込まれます。

ディスプレイ: レコードは、それぞれ前のレコードに重ね書きされながら、ディスプレイに書き込まれます。各出力レコードは、Enter キーを使用して要求することができます。

プリンター: レコードは印刷され、改行または切り捨ては、プリンター・ファイルに指定されているとおりに実行されます。

テープ: レコードは、順次にテープに書き込まれます。

コピー元

ディスプレイ入力

コピー先

ICF: レコードは、一度に 1 つずつ ICF ファイルから取り出されます。

ディスク: レコードは、順次に取り出されます。ディスク・ラベル情報は、ディスク・ファイルの中またはオーバーライド・コマンドで指定されなければなりません。

データベース: 入力レコードが取り出されます。

テープ: レコードは、順次に取り出されます。テープ・ラベル情報は、テープ・ファイルの中またはオーバーライド・コマンドで指定されなければなりません。

コピー元

ディスプレイ出力

コピー先

ICF: レコードは、一度に 1 つずつ ICF ファイルに書き込まれます。

データベース: レコードは、順次にデータベースに書き込まれます。

ディスク: ディスク上書きされるデータ量は、ディスクの交換タイプによって異なります。ディスク・ラベル情報は、ディスク・ファイルの中またはオーバーライド・コマンドで指定されなければなりません。

テープ: レコードは、順次にテープに書き込まれます。テープ・ラベル情報は、テープ・ファイルの中またはオーバーライド・コマンドで指定されなければなりません。

プリンター: レコードは印刷され、改行または切り捨ては、プリンター・ファイルに指定されているとおりに実行されます。

コピー元

ディスプレイ入出力

コピー先

ICF: 入力レコードは、一度に 1 つずつ ICF ファイルから取り出されます。出力レコードは一度に 1 つずつ ICF ファイルに書き込まれます。入力レコードと出力レコードの関係は、アプリケーション・プログラムによって決まります。

コピー元

データベース入力 (順次に処理される)

コピー先

ICF: レコードは、一度に 1 つずつ ICF ファイルから取り出されます。

ディスプレイ: レコードは、一度に 1 つずつディスプレイから取り出されます。各レコードのデータを入力し、レコードが完成したら、Enter キーを押してください。非フィールド・レベル装置ファイルが指定されなければなりません。

ディスク: レコードは、順次に取り出されます。ディスク・ラベル情報は、ディスク・ファイルの中またはオーバーライド・コマンドで指定されなければなりません。

テープ: レコードは、順次にテープから取り出されます。テープ・ラベル情報は、テープ・ファイルの中またはオーバーライド・コマンドで指定されなければなりません。

コピー元

データベース出力 (順次に処理される)

コピー先

プリンター: 印刷される文字数は、指定されるページ・サイズで決定されます。折り畳みが指定されている場合は、すべてのレコードが印刷されます。

ICF: レコードは、一度に 1 つずつ ICF ファイルに書き込まれます。

ディスプレイ: レコードは、それぞれ前のレコードに重ね書きされながら、ディスプレイに書き込まれます。各出力レコードは、Enter キーを使用して要求することができます。

ディスクット: ディスクット上に書き出されるデータ量は、ディスクットの交換タイプによって異なります。ディスクット・ラベル情報は、ディスクット・ファイルの中またはオーバーライド・コマンドで指定されなければなりません。

テープ: レコードは、順次にテープに書き込まれます。テープ・ラベル情報は、テープ・ファイルの中またはオーバーライド・コマンドで指定されなければなりません。

コピー元

テープ入力

コピー先

ICF: レコードは、一度に 1 つずつ ICF ファイルから取り出されます。

ディスプレイ: レコードは、一度に 1 つずつディスプレイから取り出されます。各レコードのデータを入力し、レコードが完成したら、Enter キーを押してください。非フィールド・レベル装置ファイルが指定されなければなりません。テープのラベル情報は無視されます。

データベース: レコードは、順次に取り出されます。1 つのレコードが単一のフィールドとして読み取られます。テープのラベル情報は無視されます。

ディスクット: レコードは、順次に取り出されます。ラベル値がプログラムの中で指定されている場合は、その値がディスクット・ファイルのラベルとして使用されます。

コピー元

テープ出力

コピー先

プリンター: レコードは印刷され、改行または切り捨ては、プリンター・ファイルに指定されているとおりに実行されます。

ICF: レコードは、一度に 1 つずつ ICF ファイルに書き込まれます。テープのラベル情報は無視されます。

ディスクット: ディスクット上に書き出されるデータ量は、ディスクットの交換タイプによって異なります。ラベル値がプログラムの中で指定されている場合は、その値がディスクット・ファイルのラベルとして使用されます。

ディスプレイ: レコードは、それぞれ前のレコードに重ね書きされながら、ディスプレイに書き込まれます。各出力レコードは、Enter キーを使用して要求することができます。

データベース: レコードは、順次にデータベースに書き込まれます。

パフォーマンス

コピー操作のパフォーマンスを改善するために、いくつかの指針に従うことができます。

前のトピックで説明した指針のほかに、分散ファイルをコピーするときには、コピー・コマンドのパフォーマンスに影響するさまざまな要因を理解しておく必要があります。DB2 マルチシステム機能は、分散ファイル (複数の iSeries サーバーにまたがっているファイル) をサポートしています。分散ファイルをコピーする場合は、コピー・コマンドのパフォーマンスに影響するさまざまな要因に精通している必要があります。分散ファイルへのコピーや分散ファイルからのコピーを行うときに適用される制限についても知っている必要があります。分散ファイルのコピーについては、DB2 Multisystem を参照してください。

キー順アクセス・パスの回避

キー順アクセス・パスの保守を必要とするコピーは、到着順アクセス・パスとの間でのコピーよりも低速です。

到着順がキー順アクセス・パスと同じになるようにコピー元ファイルを再編成して、コピーのパフォーマンスを向上させることができます。また、キー順アクセス・パスが使用されないように FROMRCD または TORCD パラメーターを使用してレコードを選択しても、コピーのパフォーマンスを向上させることができます。

コピー先ファイルに対して作成する論理アクセス・パスを少なくしてください。これにより、コピー・プロセスが多くのアクセス・パスを更新しなくてもよくなるので、コピーのパフォーマンスが向上します。

ファイル内のレコードの長さを短くするほど、コピーは速くなります。

指定するパラメーターを少なくする

一般に、指定するオプションのコピー・パラメーターを少なくすれば、コピーのパフォーマンスを向上させることができます。

コピー操作のパフォーマンスに影響するパラメーターは以下のとおりです。

- ERRLVL
- FMTOPT
- INCCHAR
- INCREL
- PRINT
- SRCOPT

COMPRESS 機能の使用はパフォーマンスにそれほど影響しません。削除済みレコードがコピー先ファイルで必要な場合、たとえば相対レコード番号が同じである必要があるときは、COMPRESS(*NO) を要求してください。

レコード・フォーマット・レベル ID の検査

CPYF コマンドでレコード・フォーマット・レベル ID を正しく設定することによっても、コピーのパフォーマンスを向上させることができます。

CPYF を使用して、おそらく同一と思われる 2 つのファイル間でデータを移動している場合、コピー・パフォーマンスを最適化するためには、レコード・フォーマット・レベル ID も同一でなければなりません。レコード・フォーマット・レベル ID が同一でないと、CPYF は、すべてのレコードで各フィールドおよび列を検査する、長いコード・パスを通ることになります。そうすると、拡張検査が必要ない場合でも、CPYF が機能を完了するのにかかる時間に影響が及ぶことがあります。

2 つのファイルに異なる属性がある場合、それらのファイルのレコード・フォーマット・レベル ID も異なっているはずですが、レコード・フォーマット・レベル ID が同一だと、フィールドおよび列レベルの検査が実行されず、その結果、データ変換が不適切に行われたり、まったく行われなかったりします。FMTOPT (*NOCHK) を指定すると、レコード・フォーマット・レベル ID の値に関係なく、フィールドおよび列レベルの検査を回避することができます。ただし、FMTOPT (*NOCHK) を指定すると、データの特定の属性 (ヌル値など) が失われます。

ファイルのコピー時におけるエラーの防止

コピーの前に特定の条件や状況を計画するとき、多くのコピー・エラーを防止することができます。

関連概念

43 ページの『レコードの印刷 (PRINT、OUTFMT、および TOFILE(*PRINT) パラメーター)』
コピーされたすべてのレコード、除外されたすべてのレコード、または ERRLVL 出力エラーが生じたすべてのレコードを印刷することができます。これを行うには、コピー・コマンド上に PRINT 特殊値を指定します。

コピー時における回復可能エラーの制限

データベース・ファイルとの間のコピー、またはテープ・ファイルからのコピーを行うときは、コピーが打ち切られる前に受け入れられることのできる回復可能エラーの数を限定することができます。この限界を指定するには ERRLVL パラメーターを指定します。

このパラメーターは以下のタイプのエラーに適用されます。

CPF4826

媒体エラー

CPF5026

このメンバーのアクセス・パスの中の重複キー。(注: CPYF で MBROPT(*UPDADD) を指定した場合、コピー・コマンドは CPF5026 を ERRLVL エラーと見なしません。)

CPF5027

他のジョブが使用中のレコード。(注: CPYF で MBROPT(*UPDADD) を指定した場合、コピー・コマンドは CPF5027 だけを ERRLVL エラーと見なします。)

CPF5029

データまたはキーの変換エラー

CPF502D

参照保全制約の違反

CPF502E

参照保全制約の妥当性検査を行うことはできませんでした。

CPF5030

メンバーの部分損傷

CPF5034

別のメンバーのアクセス・パスの中の重複キー

CPF5036

無効な長さのテープ・ブロック読み取り

CPF504B

データ・リンク・エラー

CPF504C

データ・リンク作成エラー

CPF5097

浮動小数点キー・フィールドでは許可されない *NAN (非数値) 値

ERRLVL パラメーターは、コピーされた各ラベル対または各メンバーで許可される回復可能エラーの最大数を指定します。ERRLVL に指定される値は、コピーされる各メンバーまたはラベルの対ごとに、コピー元ファイルとコピー先ファイルの両方を合わせて許される合計エラー数を示します。エラーが発生するたびに、以下の処理が行われます。

1. そのラベル対またはメンバーのカウントが 1 つ増えます。
2. TOFILE(*PRINT)、PRINT(*COPIED)、または PRINT(*EXCLD) が指定されている場合は、直前に読み取りまたは書き出しされた良好なレコードを識別するメッセージが、すべてのコピー・リストに印刷されます。
3. PRINT(*ERROR) が指定されている場合は、エラー・レコードが印刷されます。
4. コピーが継続します。
5. コピー元ファイル・メンバーが、限界値を超えることなく完全にコピーされた場合は、カウンターは 0 にリセットされ、次のメンバーのコピーが開始されます。
6. あるメンバーのコピー中に限界値を超えた場合は、コピーしなければならないレコードがまだ残っていたり追加メンバーがあっても、コピーは打ち切れ、メッセージが送られます。

コピー元データベース・ファイル (オープン Query ファイルも含めて) の場合、回復可能エラーとは次のようなエラーです。

- データの変換 (マップ) 時に生じるエラー
- ディスク (補助記憶域) の損傷に起因するエラー

コピー元テープ・ファイルの場合、回復可能エラーとは次のようなエラーです。

- 有効でないブロック長
- 装置のテープ・ボリュウムからの媒体読み取り操作のエラー

コピー先物理ファイルの場合、回復可能エラーとは次のようなエラーです。

- データの変換時に生じるエラー
- 複数の同一キーが検出されるときに生じるエラー

エラーを発生させるレコードは、コピー先ファイルにコピーされません。書き込みエラーの場合、そのレコードは PRINT(*COPIED) および PRINT(*EXCLD) 印刷出力に印刷されます。この印刷出力の後にはメッセージが表示されます。このメッセージは、レコードが実際にはコピーされなかったことを示します。PRINT(*ERROR) が指定された場合、書き込みエラーを発生させたレコードは *ERROR リストに印刷されます。このとき、エラーが発生したことを示すメッセージが表示されます。読み取りエラーの場合は、コピー印刷出力 (TOFILE(*PRINT)、PRINT(*COPIED)、PRINT(*EXCLD)、または PRINT(*ERROR)) にそのレコードを印刷することができません。ただし、レコードの読み取りができなかったことを示すメッセージが指定のすべての印刷出力に印刷されます。

ファイルの一部をディスクから読み取れない場合、データベース・ファイルの内容に対する部分オブジェクト損傷が発生します。ファイルがこのように損傷している場合は、良好なレコードをコピーし、損傷のためにコピーされなかったレコードを手操作で追加することによって、エラー状態のレコードをバイパスすることができます。

ERRLVL パラメーターの値には関係なく、回復可能エラーは必ず応答 『C』 (『Cancel』 (取り消し) を示す) を伴ってジョブ・ログに記録されます。

制約関係があるファイルについては、ERRLVL パラメーターが影響を及ぼすのはコピー先ファイルのみです。ERRLVL パラメーターが 0 に設定されている場合、コピー先ファイルの制約関係に違反するすべてのレコードは、ファイルにコピーされません。コピー操作は終了します。ERRLVL が 0 より大きい場合は、コピー先ファイルの制約関係に違反するすべてのレコードは、コピー先ファイルにコピーされません。しかし、コピー操作は、違反 (回復可能エラー) の発生が ERLVL 値に到達するまで継続されます。この値を超えるとコピー操作は終了します。

ERRLVL パラメーターは、検査保留状況にある制約関係があるファイルを非検査保留状況へ戻すのに使用することができます。これを行うには、コピー元の従属ファイルと同じ制約を持つコピー先の従属ファイルを設定してください。その後、ERRLVL(*NOMAX) を設定した CPYF コマンドを使用して、すべての有効なレコードをコピーしてください。コピー先ファイルは空にしておいてください。コピー先ファイルの制約を検査保留状況にするコピー元ファイルからのレコードは、コピー先ファイルへ挿入されません。ERRLVL が *NOMAX に設定されている場合は、コピー元ファイルにあるすべてのレコードが処理されません。

他のコピー・コマンド (CPYSRCF、CPYFRMTAP、および CPYTOTAP) には ERLVL パラメーターがないので、回復可能エラーが 1 つ通知されると、すぐに終了します。

ファイルのコピー時における日付、時刻、およびタイム・スタンプ・エラーの防止

FMTOPT(*MAP)、*BLDKEY を指定した FROMKEY、*BLDKEY を指定した TOKEY、および INCREL パラメーターを使用する場合、2 桁の年フィールドまたは値は、年フィールドが 40 から 99 の範囲のときには 1900 年代、年フィールドが 00 から 39 の範囲のときには 2000 年代と見なされます。

たとえば、12/31/91 は 1991 年の 12 月 31 日となりますが、12/31/38 は 2038 年の 12 月 31 日となります。

しかし、1940 年 1 月 1 日から 2039 年 12 月 31 日の範囲でない内部日付を持つ 2 桁の年フィールドを含むコピー元ファイルでは、入力マッピング・エラーが生じてコピー操作は失敗します。

FMTOPT(*MAP) を使用してコピー元ファイル・フィールドの年フィールドを 4 桁から 2 桁に変換またはコピーする場合、コピー元ファイル・フィールドは 1940 年 1 月 1 日から 2039 年 12 月 31 日までの値でなければなりません。そうでない場合、マッピング・エラーが生じてコピー先ファイル・フィールドにはデフォルト値がセットされます。

同様に、*BLDKEY を指定した FROMKEY または *BLDKEY を指定した TOKEY のレコード選択入力カストリングとして 4 桁の年フィールドを使用する場合、対応するコピー元ファイル・フィールドが 2 桁の年フィールドを持つ場合に 4 桁の年フィールドの値は上記の範囲の値でなければなりません。そうでない場合、マッピング・エラーが生じます。INCREL レコード選択はこの規則の例外です。この範囲外の 4 桁の年フィールド値を、対応する 2 桁の年フィールドに対して使用することもできます。

さまざまなフィールド・タイプおよび様式を処理する方法の詳細については、『コピーを使用したマッピングに関する考慮事項』を参照してください。

コピーを使用したマッピングに関する考慮事項: 日付、時刻、タイム・スタンプ・フィールドに文字フィールドをマップする場合に、フォーマット様式がその文字フィールドで使用されていると、先行ゼロは月、日、時間の箇所から省略されることがあります。マイクロ秒は文字フィールドから切り捨てられるかまたは全部省略されることがあります。

時刻フィールドにマップする場合、秒の箇所 (および対応する区切り記号) は文字フィールドから省略されることがあります。

*USA 様式の値を使用する場合、先行ブランクのある AM または PM が必要です。これらの規則は、CPYF コマンドで *BLDKEY を指定した FROMKEY、*BLDKEY を指定した TOKEY、または INCREL パラメーターを使用するときに入力される日付、時刻、またはタイム・スタンプの値にも適用されます。日付、時刻、およびタイム・スタンプの他のすべてのインスタンスは、先行ゼロが必要な場合があり、切り捨ては行われません。

TOKEY パラメーターの両様式 (*BLDKEY または非 *BLDKEY) を使用する場合、コピー元フィールド・データは日時順に比較するために特定様式でなければなりません。日付フィールドの場合、日時順に比較するために *ISO または *JIS 様式を使用しなければなりません。時刻フィールドの場合、日時順に比較するために *HMS、*ISO、*EUR、または *JIS 様式を使用しなければなりません。他の日付または時刻フィールド (日付用 (*MDY、*DMY、*YMD、*JUL、*EUR、または *USA) または時刻用 (*USA)) の場合、TOKEY パラメーターが直接的な文字比較を実行するので日時順の比較ができません。日付順の比較ができない場合、通知メッセージが送られ、コピー操作は継続します。

日付、時刻、またはタイム・スタンプ・フィールドを持つファイルにデータをコピーするとき、およびコピー元ファイルが装置ファイルであるか、または FMTOPT(*NOCHK) か FMTOPT(*CVTSRC) が指定されているとき、日付、時刻、またはタイム・スタンプ・フィールドにコピーされたデータがコピー先ファイル・フィールド様式および区切り記号属性に対して有効でない場合に出力マッピング・エラーが生じることがあります。これが生じる場合、レコードをコピーすることはできません。なお、CPYF または CPYFRMQRYP コマンドが使用される場合、そのレコードを回避し、コピー操作を継続するためにゼロ以外のエラー・レベル (ERRLVL パラメーター) を指定することができます。これらのインスタンスを日付、時刻、またはタイム・スタンプ・フィールドにコピーするとき、コピー元ファイル・データがコピー先ファイルに対して有効であるかどうか重要です。

ファイルのコピー時における位置エラーの防止

コピー対象の最初のレコードがコピー元ファイル・メンバー内に見つからない場合、ファイル・コピー機能の位置エラーが発生します。

これが発生する可能性があるのは、CPYF、CPYSRCF、または CPYTOTAP コマンドを使用している場合です。以下のいずれかの条件が満たされる場合、コピー元ファイル・メンバーの位置エラーを受け取ることがあります。

- FROMKEY パラメーターが指定されていて、メンバー内のすべてのレコードが FROMKEY 値よりも小さいか、メンバーが空である。
- FROMRCD パラメーターがメンバーの終わりを超えて指定されたか、メンバーが空である。
- コピー元ファイル・メンバー位置の値 (OVRDBF コマンドの POSITION パラメーター) がメンバーの終わりを超えているか、またはコピー元ファイルのアクセス・パスに対して有効でないか、あるいはメンバーが空である場合。

メンバー位置エラーが発生すると、そのメンバーをコピー先ファイルに追加することはできません。また、メンバーに関する情報は印刷リストに追加されません。

複数のメンバーがかかわるコピー操作中にメンバー位置エラーが発生すると、コピー操作は次のメンバーに対して継続されます。

すべてのメンバーについてメンバー位置エラーが発生すると、印刷リストは生成されず、コピー先ファイルを作成することはできません。

ファイルのコピー時における割り振りエラーの防止

ファイルをコピーするとき、iSeries はコピー元ファイルとコピー先ファイルをロックします。ファイル・コピー時の割り振りエラーを防ぐために、これらのファイルに対してより強いロックを設定することができます。

データベース・ファイルがコピーされる時、各コピー元ファイル・メンバーには読み取り専用 (*SHRRD) ロック状態が割り振られます。装置ファイルがコピーされる時、メンバーには読み取り専用 (*SHRRD) ロック状態が割り振られます。メンバーは、コピーされる時にものみ割り振られます。読み取り専用ロック状態では、他のユーザーは、コピー中のファイルの読み取りおよび更新を行えます。

一般的に、コピー先メンバーには更新専用 (*SHRUPD) ロック状態が割り振られます。ただし、MBROPT(*REPLACE) が指定されている場合は、コピー先になるメンバーには排他 (*EXCL) ロック状態が割り振られ、コピー先になるファイルの中のレコードは削除されます。

物理ファイルから別の物理ファイルにコピーするときは、より強力なロックをメンバーに施して、内部システム機能でコピーを実行することができるようにすることもできます。

- コマンドにより、コピー元ファイル・メンバーに読み取り可能排他 (*EXCLDRD) ロック状態が割り振ることができます。
- コマンドにより、コピー先ファイル・メンバーに排他 (*EXCL) ロック状態が割り振ることができます。

実行するコピーのタイプによって、これらのより強力なロックが必要となります。これらのロックを取得できない場合は、コピー・コマンドを実行して ERRVLV パラメーターに値 1 (または 0 以外の有効な値) を指定してください。これらの値では、より強力なロックは必要ありません。

ファイルのコピー時には、さまざまな理由で割り振りエラーが発生する可能性があります。たとえば、コピー時にコピー先ファイルを使用する機能は使用しないでください。

ファイルのコピー時における割り振りエラーの理由: 別のジョブにより、あるメンバーに強力すぎるロック状態が割り振られた場合、エラー・メッセージが出されてコピー操作が打ち切られる可能性があります。ファイルを含むライブラリーがコピー操作中に名前変更された場合も同様です。

コピー・コマンドの実行時には、(タイムアウトのない *EXCL ロックと同様のロック) コピー先ファイルがロックされてアクセスできなくなることがあります。コピー先ファイルを使用しなければならない機能を使用しようとする時、そのコピー・コマンドが完了するまで、そのワークステーションはロックされます。たとえば、コピー中のコピー先ファイルに対して以下の機能を使用しないでください。

WRKACTJOB

- オプション 11 (ロックの作業)
- オプション 5 (ジョブ・メンバー・ロックの作業)
- オプション 8 (オブジェクト・ロックの作業)

DSPDBR

DSPFD

DSPFFD

WRKJOB

- オプション 12 (活動状態のロックの作業)
- オプション 5 (ジョブ・メンバー・ロックの作業)
- F10 (活動中のオープン・ファイルの表示)

WRKLIB

- コピー先ファイルが入っているライブラリー

DSPLIB

コピー先ファイルが入っているライブラリー
WRKOBJLCK
WRKRCDLCK

コピー先ファイルに関する何らかの情報を表示したい場合は、その要件をあらかじめ予想し、ERRLV(1)を指定することによりコピー・コマンドで、ブロック・レコード単位の操作を強制的に使用させるようにしなければなりません。

これが原因で問題が発生するとあらかじめ予想される場合は、Allocate Object (ALCOBJ) コマンドを使用して、ファイルおよびメンバーを事前割り振りすることができます。

関連情報

制御言語 (CL)

制約関係に起因するコピー・エラーの防止

制約関係とは、従属ファイルと親ファイルのデータ保全性を保証する機構です。ファイルのコピー時には、制約関係に注意する必要があります。

制約関係が従属ファイルと親ファイル間に存在するのは、従属ファイルの外部キー・アクセス・パスの非 NULL 外部キー値すべてが、親ファイルの親キー・アクセス・パスの親キー値と一致する時です。物理データ・ファイルは、親ファイルにも従属ファイルにもなることができます。しかし、ソース物理ファイルは親ファイルにも従属ファイルにもなることができません。

以下にリストされているコピー・コマンドでは、次のような関係が許可されます。

- CPYF - コピー元ファイルまたはコピー先ファイルは、親または従属ファイルになることができます。
- CPYFRMQRYP - コピー先ファイルは、親または従属ファイルになることができます。
- CPYFRMTAP - コピー先ファイルは、親または従属ファイルになることができます。
- CPYTOTAP - コピー元ファイルは、親または従属ファイルになることができます。

関連概念

『検査保留状況にないファイルのコピー』

親または従属ファイルに、検査保留状況にない制約関係が確立されている場合は、このトピックに説明されている規則が適用されます。

141 ページの『検査保留状況にあるファイルのコピー』

親または従属ファイルに、検査保留状況にある制約関係が確立されている場合は、このトピックに説明されている規則が適用されます。

検査保留状況にないファイルのコピー

親または従属ファイルに、検査保留状況にない制約関係が確立されている場合は、このトピックに説明されている規則が適用されます。

- コピー元ファイルに制約関係が確立されている場合は、それが親ファイルであろうと従属ファイルであろうと、コピー元ファイルからすべてのレコードがコピーされます。
- コピー先ファイルに制約関係が確立されているか使用可能にされている場合は、制約関係が検査保留状況にならないようにするために、以下の規則を適用します。
 - 親ファイルは、レコードが取り除かれたメンバーを持つことはできません。
 - 親ファイルは、親キー・アクセス・パスに同じ値の複数の親キー値を持つことはできません (キーは固有でなければなりません)。すなわち、コピー先ファイルが制約関係にある親ファイルである場合、コピーによって重複キー・レコードをコピー先ファイルにコピーすることはできません。

- 従属ファイルの非 NULL 外部キー値は、対応する親キー値を常に持たなければなりません。すなわち、コピー先ファイルが制約関係にある従属ファイルの場合、コピー操作によって対応する親キー・レコードを持たない非 NULL 外部キー・レコードを従属ファイルにコピーすることはできません。

コピー操作により、親または従属のコピー先ファイルが破壊されていないことを保証します。レコードによって制約関係が検査保留状況にならない限り、レコードをコピー先ファイルにコピーすることができます。制約関係の規則と一致しないレコードをコピーしようとする場合、ERRLVL パラメーターが 0 より大きい値で指定されていないと (CPYF および CPYFRMQRYF コマンドのみ)、コピー操作は終了します。

上記の規則を回避するには、コピー操作の前に含まれている制約を使用不能にし、コピーを実行し、制約を再び使用可能にします。しかし、制約の規則が満たされないと、ファイルは検査保留状況になります。

関連概念

140 ページの『制約関係に起因するコピー・エラーの防止』

制約関係とは、従属ファイルと親ファイルのデータ安全性を保証する機構です。ファイルのコピー時には、制約関係に注意する必要があります。

検査保留状況にあるファイルのコピー

親または従属ファイルに、検査保留状況にある制約関係が確立されている場合は、このトピックに説明されている規則が適用されます。

- コピー元ファイルに検査保留の制約関係が確立されているならば、データ・アクセスが制限されます。コピー元ファイルが親ファイルならば、コマンドはデータを読み取り、コピー先ファイルにコピーします。コピー元ファイルが従属ファイルならば、データを読み取ることができないので、コピー先ファイルにはコピーされません。
- コピー先ファイルに検査保留状況の制約関係が確立されているならば、データ・アクセスが制限されます。コピー先ファイルが親ファイルならば、新しいレコードを追加できます (MBROPT(*ADD) を指定できます)。コピー先ファイルが親ファイルならば、そのファイルは消去できません (MBROPT(*REPLACE) を指定できません)。コピー先ファイルが従属ファイルならば、どの MBROPT パラメーター・キーワードを使用していても、コピーを実行できません。

上記の規則を回避するには、コピー操作の前に含まれている制約を使用不能にし、コピーを実行し、制約を再び使用可能にします。しかし、制約の規則が一致しないと、ファイルは検査保留状況になります。

関連概念

140 ページの『制約関係に起因するコピー・エラーの防止』

制約関係とは、従属ファイルと親ファイルのデータ安全性を保証する機構です。ファイルのコピー時には、制約関係に注意する必要があります。

ファイルに対する権限に関係したコピー・エラーの防止

このトピックの表では、コピー元ファイルおよびコピー先ファイルに対する必要な権限を要約しています。

表 13. コピー操作の実行に必要な権限

	コピー元ファイル	コピー先ファイル
DDM ファイル	*OBJOPR *READ	*OBJOPR ¹ *ADD
装置ファイル ²	*OBJOPR *READ	*OBJOPR *READ
論理ファイル	*OBJOPR ³ *READ	許可されていません
物理ファイル	*OBJOPR *READ	*OBJOPR ¹ *ADD

表 13. コピー操作の実行に必要な権限 (続き)

	コピー元ファイル	コピー先ファイル
:		
1	これは、MBROPT(*ADD) で必要な権限です。 MBROPT(*REPLACE) が指定されている場合は、*OBJMGT および *DLT も必要とされます。 MBROPT(*UPDADD) が指定された場合、*UPD 権限も必要になります。	
2	ファイルとして使用される装置の場合は、*OBJOPR および *READ 権限も必要です。	
3	コピーされる論理ファイル・メンバーの場合は、基礎になる物理ファイル・メンバーに対する *READ 権限も必要です。	

コピー先ファイルが存在せず、CRTFILE(*YES) を指定してコピー・コマンドでコピー先ファイルを作成するようにした場合は、 CRTPF コマンドに対する操作権限も持っていなければなりません。

セキュリティ

iSeries には、データやファイルへのアクセス制御に役立つさまざまなセキュリティ機能があります。

これらのトピックでは、いくつかのファイル・セキュリティ機能について説明します。それぞれのトピックで、オブジェクト操作権、オブジェクト存在権、オブジェクト管理権、オブジェクト参照権、オブジェクト変更権という各オブジェクト権限タイプを取り上げます。データ権限や、データに対するユーザー・アクセスを制限するべき理由についても説明します。さらに、ファイル操作に必要な権限や、ファイルの作成時にファイルやデータへのアクセスを制限する方法に関するトピックもあります。

関連情報

iSeries 機密保護解説書 (PDF)

システム・セキュリティの計画とセットアップ

オブジェクト権限

ユーザーに対して認可できる、ファイルに関連した権限には、いくつかのタイプがあります。また、SQL GRANT および REVOKE ステートメントを使用して、これらの iSeries 権限を SQL 表 (表内の個々の列を含む) に割り当てたり、そこから除去したりすることができます。

関連情報

SQL 解説書

オブジェクト操作権

オブジェクト操作権は、該当オブジェクトに対するユーザーのデータ権限に従ってユーザーがオブジェクト記述を表示し、そのオブジェクトを使用できるようにします。

オブジェクト操作権は、次のような場合に必要です。

- 処理するファイルをオープンする。この場合は、ファイルに対する読み取り権限も必要です。スプーリングを使用していない装置ファイルの場合は、装置に対するオブジェクト操作権およびすべてのデータ権限も持っていなければなりません。
- ファイル記述を使用するプログラムをコンパイルする。
- ファイル記述を表示する。
- ファイルを削除する。
- ファイルの所有権を移す。

- 権限の認可および取り消しを行う。
- ファイル記述を変更する。
- ファイルを移動および名前変更する。

オブジェクト存在権

オブジェクト存在権によって、ファイルの存在状況を変更することができます。

オブジェクト存在権は、次のような場合に必要です。

- ファイルを削除する。
- ファイルの記憶域を保管、復元、および解放する。
- ファイルの所有権を移す。

オブジェクト管理権

オブジェクト管理権によって、ファイルに対するさまざまな操作を実行することができます。

オブジェクト管理権は、次のような場合に必要です。

- 権限の認可および取り消しを行う。 ユーザーは自分が持っている権限だけに限り、その認可および取り消しを行うことができます。
- ファイル記述を変更する。
- ファイルを移動および名前変更する。
- 別のデータベース・ファイルからデータベース・ファイルを参照する。
- トリガーをデータベース・ファイルへ追加および除去する。
- 参照制約および固有制約をデータベース・ファイルに追加する。
- データベース・ファイルに対する参照制約および固有制約を除去する。
- データベース・ファイルの属性を変更する。
- SQL パッケージの属性を変更する。

オブジェクト参照権

オブジェクト参照権によって、別のデータベース・ファイルからデータベース・ファイルを参照することができます。参照されたデータベース・ファイルで実行できる操作は、参照しているデータベース・ファイルによって判別されます。

オブジェクト変更権

オブジェクト変更権は、データベース・ファイルまたは SQL パッケージの属性を変更することを許可します。

オブジェクト変更権は、次の場合に必要とされます。

- トリガーをデータベース・ファイルへ追加および除去する。
- 参照制約および固有制約をデータベース・ファイルに追加する。
- データベース・ファイルに対する参照制約および固有制約を除去する。
- データベース・ファイルの属性を変更する。
- SQL パッケージの属性を変更する。

データ権限

データ権限を使用すると、ファイル内のデータに対するアクセスを制限することができます。

関連する操作を実行するために、以下の権限が必要です。

実行 プログラムを実行するため、またはライブラリーにあるオブジェクトを見つけるための権限。

読み取り

入力のためのファイルのオープン、ファイルを使用したプログラムのコンパイル、あるいはファイル記述の表示などのための権限。

追加 ファイルに新しいレコードを追加するための権限。

更新 更新のためにデータベース・ファイルをオープンする権限。

削除 削除のためにデータベース・ファイルをオープンする権限。

データベース・ファイルおよび保管ファイル以外のファイルの場合は、実行権、追加権、更新権、および削除権は無視されます。

ファイル操作に必要な権限

このトピックでは、ファイル機能に必要なファイルのオブジェクト権限およびデータ権限をリストしています。

表 14 では、ファイル機能に必要な、ファイルのオブジェクト権限をリストしています。145 ページの表 15 では、ファイル機能に必要なデータ権限をリストしています。この内容は、前の 2 つの節で示されたのと同じものですが、ここでは、権限ごとに機能をリストするのではなく、機能ごとに権限をリストしています。

表 14. ファイル操作に必要なオブジェクト権限： ファイル機能に必要なファイルのオブジェクト権限

機能	オブジェクト操作権	オブジェクト存		オブジェクト参照権	オブジェクト変更権
		在権	オブジェクト管理権		
ファイルのオープン、入出力、クローズ ¹	X				
ファイル記述を使用したプログラムのコンパイル	X				
ファイル記述の表示	X				
ファイルの削除	X	X			
保管/復元		X			
所有権の移動	X	X			
権限の認可/取り消し	X		X		
ファイル記述の変更	X		X		
ファイルの移動	X		X		
ファイルの名前変更	X		X		
ファイルの置換	X	X	X		
別のファイルを参照 ²			X	X	
ファイル制約を追加または除去 ³			X		X
トリガーを追加または除去 ⁴			X		X

表 14. ファイル操作に必要なオブジェクト権限 (続き): ファイル機能に必要なファイルのオブジェクト権限

機能	オブジェクト存		オブジェクト参照		オブジェクト変更権
	オブジェクト操作権	在権	オブジェクト管理権	権	
属性の変更 ⁵			X		X
注:					
1	スプーリングを使用していない装置ファイルの場合は、その装置に対してオブジェクト操作権および全データ権限も持っていなければなりません。				
2	データベース・ファイルに対してのみ。				
3	データベース・ファイルに対してのみ。親ファイルには、オブジェクト管理権またはオブジェクト参照権が必要です。従属ファイルには、オブジェクト管理権またはオブジェクト変更権が必要です。				
4	データベース・ファイルに対してのみ。ファイルには、オブジェクト管理権またはオブジェクト変更権が必要です。				
5	データベース・ファイルおよび SQL パッケージに対してのみ。ファイルには、オブジェクト管理権またはオブジェクト変更権が必要です。				

表 15. ファイル操作に必要なデータ権限: ファイル機能に必要なデータ権限。

機能	実行	読み取り	追加	更新	削除
ファイルのオープン、入出力、クローズ ¹		X	X ²	X ³	X ³
ファイル記述を使用したプログラムのコンパイル		X			
プログラムの実行またはライブラリーにあるオブジェクトの検出	X				
ファイル記述の表示		X			
ファイルの置換		X			
トリガーを追加または除去 ⁴		X	X ⁵	X ⁶	X ⁷
注:					
1	スプーリングを使用していない装置ファイルの場合は、その装置に対してオブジェクト操作権および全データ権限も持っていなければなりません。				
2	データベース、および保管ファイルへの出力を目的としたオープン。				
3	データベース・ファイルに対する更新、または削除を目的としたオープン。				
4	データベース・ファイルに対してのみ。				
5	トリガーを挿入するのに、読み取り権に加えて必要とされる追加権。				
6	トリガーを更新するのに、読み取り権に加えて必要とされる更新権。				
7	トリガーを削除するのに、読み取り権に加えて必要とされる削除権。				

ファイル作成時のファイルおよびデータへのアクセス制限

権限を指定すると、ファイルへのアクセスを制御できるようになります。ファイル作成時に共通権限を指定するには、作成コマンドで AUT パラメーターを使用してください。

共通権限とは

共通権限とは、ユーザーがファイルに対して特定の権限を持っていないか、ファイルに対して特定の権限を持っているグループのメンバーでない場合に使用される権限です。つまり、ユーザーがファイルに対して特定の権限を持っている場合、またはユーザーが特定の権限を持つグループのメンバーである場合は、ユーザーがファイルに対して操作を実行するときに、共通権限は検査されません。共通権限は次のように指定することができます。

- ***LIBCRTAUT**。ファイルに対して特定のユーザー権限またはグループ権限を持っていないすべてのユーザーが、作成中のファイルが所属するライブラリーによって決定される権限を持ちます。ライブラリーの値は、このライブラリーに対して共通権限を設定する **CRTAUT** コマンドで指定されます。
- ***CHANGE**。ファイルに対して特定のユーザー権限またはグループ権限を持っていないすべてのユーザーが、ファイルを使用する権限を持ちます。*CHANGE の値はデフォルトの共通権限です。*CHANGE は、すべてのユーザーに対してオブジェクト操作権および全データ権限を認可します。
- ***USE**。ファイルに対して特定のユーザー権限またはグループ権限を持っていないすべてのユーザーが、ファイルを使用する権限を持ちます。*USE は、ユーザーにオブジェクト操作権、実行権およびデータ読み取り権を認可します。
- ***EXCLUDE**。ファイルを変更または使用することができるのは、所有者、セキュリティー担当者、特定の権限を持つユーザー、または特定の権限を持つグループのメンバーであるユーザーだけに限られます。
- ***ALL**。ファイルに対して特定のユーザー権限またはグループ権限を持っていないすべてのユーザーが、全データ権限および全オブジェクト権限を持ちます。
- **権限リスト名**。権限リストとは、ユーザーとそのユーザーが持つ権限のリストです。このリストを使用すれば、ユーザーおよびユーザーが持っている様々な権限をまとめることができます。

既存のファイルに対する権限の指定および変更

既存のファイルに対する共通権限を指定または変更する場合は、オブジェクト権限編集 (EDTOBJAUT)、オブジェクト権限認可 (GRTOBJAUT)、オブジェクト権限取り消し (RVKOBJAUT) のいずれかのコマンドを使用してファイルの共通権限を認可したり取り消したりすることができます。

関連概念

3 ページの『ファイル・タイプ』

関連情報

iSeries 機密保護解説書 (PDF)

システム・セキュリティーの計画とセットアップ

データベース・ファイル管理のトラブルシューティング

iSeries は、エラーの発生時にサーバー・レポートおよびエラー・メッセージを提供します。エラーの内容に応じて、操作を行うことができます。

ファイル・エラー検出およびサーバーによる処理

エラーは、ファイルのオープン時、プログラム装置の獲得または解放時、ファイルに対する入出力操作中、およびファイルのクローズ時に検出されることがあります。サーバーでは、必要に応じて、正常に実行されなかった操作を再試行限度まで再び自動的に実行を試みます。この再試行が正常に完了すれば、オペレーターも、プログラムによる処置も必要ありません。

サーバーがエラーを報告する方法

サーバーは、次のいずれかまたはすべての方法を使用して、プログラムの処理に影響するエラーを報告します。

- 通知メッセージ、状況メッセージ、診断メッセージ、またはエスケープ・メッセージが、ファイルを使用するプログラムのプログラム・メッセージ待ち行列に送られます。これらのメッセージは、ジョブに対して設定されているメッセージ記録レベルによっては、ジョブ・ログにも記録されます。詳しくは、『サーバーによるファイル内のメッセージおよびメッセージ・モニター』を参照してください。
- 高水準言語でファイルの状況コードが戻されます。
- システム間通信機能 (ICF) ファイル、ディスプレイ・ファイル、およびプリンター・ファイルの場合、メジャーおよびマイナー戻りコードが、入出力フィードバック域の中で戻されます。詳しくは、149 ページの『サーバーによるファイル内のメジャーおよびマイナー戻りコード』を参照してください。
- 通知メッセージ、状況メッセージ、診断メッセージ、またはエスケープ・メッセージが、オペレーター・メッセージ待ち行列 (QSYSOPR) またはヒストリー・メッセージ待ち行列 (QHST) に送られます。
- 問題分析および解決プログラムによる使用に備えて、エラーに関する情報がサーバー・エラー・ログに保管されます。
- 警報メッセージが、ネットワークの中の別のサーバーのオペレーターに送られる場合もあります。
- 通常のプログラムの流れが中断し、制御がエラー処理サブルーチンに移されるか、または他の言語操作が行われます。実行時エラーの処理方法の詳細については、対応する高水準言語の資料を参照してください。

エラー回復を試みるプログラムにとって重要なのは、上記のうち一部だけです。

エラーを受け取ったときに行う処置

エラーを受け取ったときに行う処置については、150 ページの『ファイル・サーバー・エラーからの回復』を参照してください。

回復不能エラー

すべてのファイル・エラーに対して、プログラムによるエラー回復が可能なわけではありません。エラーの中には永続的なものもあります。この場合、ファイル、装置、またはプログラムは、何らかの修正処置が取られるまで機能することができません。この修正処置には、装置をオフに構成変更した上で再度オンに構成変更することによってリセットしたり、装置構成またはアプリケーション・プログラムのエラーを修正するなどの措置を伴うことがあります。メッセージおよび戻りコードの中には、通信回線の状況の変化または不測の事態に備えて講じられたサーバー処置など、エラーではなく通知としてユーザーまたはアプリケーション・プログラムに状態を知らせるものもあります。多くの場合、アプリケーション・プログラムでエラー状態の有無をテストし、オペレーターによる介入を伴わずプログラムの続行を可能にする、あらかじめ計画された回復処置を取ることが可能です。

関連情報

制御言語 (CL)

システム操作の基本

サーバーによるファイル内のメッセージおよびメッセージ・モニター

画面に表示されるメッセージは、オペレーターや新しいアプリケーション・プログラムのテストを行っているプログラマーにとって、主要な情報源になります。メッセージは、ファイル状況コード、標識、およびメ

ジャー/マイナー戻りコードよりも、含まれている情報の特定性が高いのが普通です。制御言語を使用するとメッセージをモニターできるので、制御言語 (CL) プログラムによってメッセージを処理して修正処置を取ることができます。

CL に関するトピック内のメッセージのセクションで、メッセージ・タイプとメッセージ・モニターに関する詳細情報が提供されています。ほとんどの高水準言語では、ファイル状況コードおよび戻りコード (次の節で説明します) の方が、情報源としてより便利です。

メッセージ番号が種類別に割り当てられているため、関連メッセージのグループをプログラムで容易にモニターすることができます。以下の表は、ファイル・エラー・メッセージに割り当てられたメッセージ番号の範囲を示しています。

表 16. OS/400 ファイル管理メッセージ番号範囲： 関連メッセージのグループをプログラムで容易にモニターするために、ファイル・エラー・メッセージに割り当てられているメッセージ番号の範囲。

メッセージ ID	操作	メッセージ・タイプ
CPF4001-40FF	オープン	診断メッセージおよび状況メッセージ
CPF4101-43FF	オープン	ファイルを使用不可にするエスケープ・メッセージ
CPF4401-44FF	クローズ	診断メッセージおよび状況メッセージ
CPF4501-46FF	クローズ	ファイルを使用不可にするエスケープ・メッセージ
CPF4701-48FF	入出力、獲得、および解放	取り消しのデフォルト応答を行なうファイルまたは装置を使用不能にしない状況またはエスケープ通知
CPF4901-49FF	入出力、獲得、および解放	無視または進行のデフォルト応答を行なう通知
CPF5001-50FF	入出力、獲得、および解放	取り消しのデフォルト応答を行なう通知
CPF5101-53FF	入出力、獲得、および解放	ファイルまたは装置を使用不可にするエスケープ・メッセージ
CPF5501-56FF	入出力、獲得、および解放	ファイルまたは装置を使用不可にするエスケープ・メッセージ

状況メッセージの中には、たとえば CPF4018 のように、補足説明を提供する診断メッセージが先行するものがあります。ジョブのメッセージ記録レベルによっては、診断メッセージがジョブ・ログに記録される場合もあります。CL プログラムで CPF4018、CPF5041、または類似のメッセージをモニターする場合は、付随する診断メッセージをプログラム・メッセージ待ち行列から取り出すことができます。

エラーが発生し、エスケープ・メッセージが出され、しかもそのメッセージがモニターされない場合、プログラムは打ち切れ、オペレーターに対してメッセージが表示されます。状況メッセージをモニターすることもできますが、これらのモニターを行わなければプログラムは続行されます。CL 以外のほとんどの高水準言語は、発生する可能性の高いすべてのファイル・エラーをモニターするだけでなく、標準の回復手順をいくつか提供しています。エラーの重大度によっては、高水準言語はプログラムを打ち切り、独自のメッ

ページを出すこともあります。その代わりに、アプリケーション・プログラマーがエラー回復ルーチンをコーディングし、その特定のアプリケーションで予測されるエラーを処理することができます。

これらのエラー処理ルーチン内では、エラーの原因を判別するために、ファイルの状況コードやメジャーおよびマイナー戻りコードを調べる必要があるのが普通です。ご使用の言語の資料には、ファイル状況コードおよびメジャー/マイナー戻りコードにアクセスする方法が説明してあります。また、各言語の資料には、それぞれの言語で定義されているファイル状況コードについての説明も含まれています。

関連情報

制御言語 (CL)

サーバーによるファイル内のメジャーおよびマイナー戻りコード

ICF ファイル、ディスプレイ・ファイル、およびプリンター・ファイルの場合に、メジャーおよびマイナー戻りコードが、エラーおよびある種の状況を報告するのに使用されます。その他のファイルの場合には使用されません。これは通常 4 文字で示され、最初の 2 文字でメジャー・コードを、残りの 2 文字でマイナー・コードを表します。

メジャー・コードはエラーの一般的なタイプを示し、マイナー・コードは詳細を示します。マイナー・コードは、ゼロ以外の場合は組み合わされているメジャー・コードに関係なく、同一のまたは類似の意味を持っています。

アプリケーション・プログラムでは、各入出力操作後に、戻りコードをテストすることができます。メジャー戻りコードが 00 の場合、操作は正常に完了し、マイナー戻りコードには、次は読み取り操作と書き出し操作のどちらを実行すべきかを指示する情報が入っています。メジャー戻りコードが 04 以上の場合は、エラーが生じたことを示します。プログラムでは、プログラムによる回復が試みられる特定のエラーの有無をテストすることができます。アプリケーション・プログラムでは、メジャー・コードとマイナー・コードを 1 つの単位として比較することによって、特定の状態の有無をテストすることも、メジャー・コードだけをテストすることによって、状態のクラスを識別することもできます。

ほとんどのメジャーおよびマイナー戻りコードには、関連するメッセージ番号があり、代表的な回復処置が類似しています。個々の言語は状況コードのファイルを作成します。これらのファイルは、メジャーおよびマイナー戻りコードに基づいて設定されます。

表 17では、メジャー戻りコードの意味が定義されています。ディスプレイ・ファイルに関して使用されるメジャーおよびマイナー戻りコードの定義、およびそれぞれに関連するメッセージ番号については、


Application Display Programming  を参照してください。プリンター・ファイルに関する同様の特定の定義および ICF ファイルで有効なそれぞれの通信タイプに関する同様の特定の定義については、印刷のトピック、および各通信タイプの解説資料で扱われています。

表 17. メジャー戻りコードの定義：メジャー戻りコードとその定義が概説されています。

コード	定義
00	プログラムによって要求された操作は正常に完了しました。マイナー・コードには方向転換などの状況情報が含まれます。
02	入力操作は正常に完了しましたが、ジョブは終了中です (制御付き)。マイナー・コードには状況情報が含まれます。
03	入力操作は正常に完了しましたが、受け取ったデータはありません。マイナー・コードには状況情報が含まれます。

表 17. メジャー戻りコードの定義 (続き): メジャー戻りコードとその定義が概説されています。

コード	定義
04	データが読み取り待機状態にあるときに出力操作が試行されたため、エラーが発生しました。
08	装置がすでに獲得されているか、セッションがすでに確立されているため、獲得操作が失敗しました。
11	装置もセッションも送信勧誘されていなかったために、送信勧誘済みプログラム装置からの読み取り操作は正常に行われませんでした。
34	入力の例外が発生しました。データ長またはレコード様式がプログラムに許容されませんでした。
80	永続 (回復不能) サーバー・エラーまたはファイル・エラーが発生しました。問題を修正するためにプログラマーの処置が必要です。
81	永続 (回復不能) 装置エラーまたはセッション・エラーが、入出力操作中に生じました。
82	オープン操作または獲得操作中に、装置エラーまたはセッション・エラーが発生しました。回復はおそらく可能です。
83	入出力操作中に、装置エラーまたはセッション・エラーが発生しました。回復はおそらく可能です。

関連情報

[アプリケーション表示プログラミング \(PDF\)](#)

[印刷](#)

ファイル・サーバー・エラーからの回復

この節では、受け取る可能性のあるエラーについて、およびエラーから回復するために行う処置について説明します。

サーバーによるエラーの正常完了

メジャーおよびマイナー戻りコード 0000 は、ユーザーのプログラムによって要求された操作が、正常に完了したことを示します。

ほとんどの場合、サーバーはメッセージを出しません。一部の場合にサーバーは、診断メッセージを使用して、システムで処理できない何らかの異常事態についてユーザーに通知することがありますが、一部の条件の下でエラーとされるものについては通知しない場合があります。たとえば、システムが無効なパラメーターを無視したり、何らかのデフォルトのアクションを行ったりする場合があります。

通信装置の場合は、メジャー戻りコード 00 は、データを受信して正常に完了したことを示しますが、これにはアプリケーション・プログラムで次に実行が予想される操作を示すマイナー戻りコードが付随します。ゼロ以外のマイナー・コードはエラーを示しません。メッセージは出されません。

サーバーによるエラーの例外を伴う完了

サーバーは、アプリケーション・プログラムからの特定の応答に該当する状態に対して、特定のメジャー戻りコード (02、03、0309 など) をいくつか割り当てます。

メジャー戻りコード 02 は、要求された入力操作は正常に完了したものの、サーバーが制御付きでジョブを終了中であることを示します。アプリケーション・プログラムでは、可能な限り迅速に処理を完了する必要があります。制御付き取り消しは、順序正しく終了するための時間をプログラムに与えるためのものです。ENDJOB コマンドで指定した時間内にプログラムが終了しない場合、サーバーはそれ以上の通知をせずにジョブを終了します。

メジャー戻りコード 03 は、入力操作は正常に完了したものの、データの転送は行われなかったことを示します。アプリケーションによって、これはエラー状態の場合もあれば、ユーザーがデータを入力しないで機能キーを押した場合もあります。このコードはまた、すべてのデータが処理されたこと、およびアプリケーション・プログラムが完了処理に進むべきであることを示すこともあります。いずれにせよ、プログラムの入力バッファの内容は無視されることになります。

メジャーおよびマイナー・コード 0309 は、サーバーがデータを受け取らなかったこと、および、制御付きの方法でジョブを終了中であることを示します。メジャーおよびマイナー・コード 0310 は、指定された待機時間が終了したのでデータはないことを示します。メジャー・コード 02 または 03 に付随するその他のマイナー戻りコードは、メジャー・コード 00 の場合と同じで、通信状況および次に実行される操作を示します。

メジャー戻りコード 04 は、出力例外が生じたことを示します。特に、データの受信をすべき時に、プログラムがデータの送信を試行した場合です。これはおそらく、前の正常な完了でマイナー戻りコードが適正に処理されなかった結果です。ユーザーのプログラムは、単に受信データを受け取ってから書き出し操作を繰り返すだけで、回復できます。

メジャー戻りコード 34 は、入力例外が生じたことを示します。受け取ったデータが長すぎるか、レコード様式に矛盾しているかどちらかです。マイナー戻りコードで、受け取ったデータの誤りであった箇所、およびデータが切り捨てられたのか拒否されたのかを示します。ユーザーのプログラムでは、おそらく例外を処理して続行できると考えられます。データが拒否された場合は、別のレコード様式を指定することによって、そのデータを読み取ることができる可能性があります。

このグループの他の 2 つの戻りコード 0800 および 1100 は、両方ともアプリケーション・プログラミン
グ・エラーの結果であるのが普通ですが、まだ回復可能です。0800 は、装置がすでに獲得されているか、セッションがすでに確立されているために獲得操作が正常に行われなかったことを示します。1100 は、装置を送信勧誘せずにプログラムで送信勧誘済み装置からの読み取りを試みたことを示します。いずれの場合も無効な要求は無視され、プログラムは続行することができます。

メジャー・コード 02 の場合、またはメジャー・コード 03 に付随するほとんどのマイナー・コードの場合、メッセージは出されませんが、このグループの他の例外には CPF4701 から CPF47FF または CPF5001 から CPF50FF の範囲のメッセージが出されるのが普通です。

永続サーバー・エラーまたはファイル・エラー

メジャー戻りコード 80 は、ファイルに影響する重大なエラーを示します。

アプリケーション・プログラムでファイルをクローズし、もう 1 度オープンしてからファイルの使用を再度試みなければなりません。エラーの原因となっている問題を見つけて修正しない限り、回復はしないでしよう。共用ファイルの中のエラー状態をリセットするために、そのファイルをクローズしてからもう一度オープンする場合は、オープン・データ・パスを共用するすべてのプログラムがそのファイルをクローズしなければなりません。したがって、呼び出しスタック内にある前のプログラムに戻り、それらの各プログラムで共用ファイルをクローズしなければならない場合があります。オペレーターまたはプログラマーは付随メッセージのテキストを参照して、特定のエラーに対して適した処置を判別しなければなりません。

このグループのいくつかのマイナー戻りコードは、特別な重要性を持ちます。メジャーおよびマイナー・コード 8081 は、重大なサーバー・エラーを示しており、この場合はおそらく APAR が必要になります。メジャーおよびマイナー戻りコードとともに送られるメッセージの中で、問題分析 (ANZPRB) コマンドを実行して詳細な情報を入手するよう指示される場合もあります。

メジャーおよびマイナー・コード 80EB は、誤ったオプションまたは矛盾したオプションが、装置ファイルの中で、またはオープン操作におけるパラメーターとして指定されたことを示します。大抵の場合は、ファイルをクローズし、プログラムを打ち切り、オーバーライド・コマンドを用いて無効パラメーターを修正して、再度プログラムを実行することができます。オーバーライド・コマンドは、コマンドが出されたジョブにのみ作用します。これによって変更を簡単にテストできますが、変更を永続的にするためには、装置ファイルを後で適切に変更または再作成することが必要です。

入出力操作での永続装置エラーまたはセッション・エラー

メジャー戻りコード 81 は、装置またはセッションに影響する重大なエラーを示します。

これには、装置、通信回線、または通信コントローラーに影響するハードウェア障害が含まれます。また、不意に切断または電源切断を生じた装置が原因のエラー、および装置によって発見され、逆にサーバーに報告された異常事態も含まれます。マイナー戻りコードと付随メッセージの両方で、問題の原因についてより特定された情報が与えられます。

ファイルのタイプによって異なりますが、プログラムはファイルをクローズしてからもう一度オープンするか、装置を解放してから再度獲得するか、あるいはセッションを獲得し直すかしなければなりません。共用ファイルの中のエラー状態をリセットするために、そのファイルをクローズしてからもう一度オープンする場合は、オープン・データ・パスを共用するすべてのプログラムがそのファイルをクローズしなければなりません。場合によっては、装置をオフに構成変更してからオンに構成変更し直すことで装置をリセットするよう、メッセージの中で指示されることもあります。エラーの原因となっている問題が見つかって修正されない限り、プログラムでその障害装置を使用することができる可能性はほとんどありませんが、代替装置が使用可能な場合、プログラム内での回復が可能なこともあります。

このグループのマイナー戻りコードの中には、メジャー戻りコード 82 の場合と同じものがあります。装置障害または回線障害はいつでも起こり得ますが、メジャー・コード 81 は入出力操作の際に発生します。これは、プログラムがすでに、装置またはセッションとの間にリンクを確立していることを意味します。このため、プログラムは既に何らかのデータを転送済みの場合がありますが、再始動の際にプログラムは先頭から開始します。結果的に、データの重複が起こります。

メジャー・コード 81 に伴うメッセージ番号は、入出力操作またはクローズ操作を示す範囲にある場合があります。クローズ操作での装置障害は、ファイルのクローズに固有の処置ではなく、データの最終ブロックの送信における障害の結果である可能性もあります。クローズ操作でのエラーは、ファイルが部分的にしかクローズされないという結果を招くことがあります。ユーザーのエラー回復プログラムでは、2 度目のクローズ操作によってクローズ障害に回答しなければなりません。2 度目のクローズ操作は、エラーに関係なく常に完了します。

オープン操作または獲得操作での装置エラーまたはセッション・エラー

メジャー戻りコード 82 は、装置エラーまたはセッション・エラーがオープン操作時または獲得操作時に生じたことを示します。マイナー戻りコードと付随メッセージの両方で、問題の原因についてより特定された情報が与えられます。

このグループのマイナー戻りコードの中には、メジャー戻りコード 81 の場合と同じものがあります。装置障害または回線障害はいつでも起こり得ますが、メジャー・コード 82 は装置またはセッションがユーザー

のプログラムで最初に使用を試みたときに使用不能であったことを示します。したがって、データは転送されませんでした。問題は、構成エラーまたはインストール・エラーの結果である可能性があります。

マイナー戻りコードによっては、ユーザーのプログラムでエラーを回復し、一定の待ち時間の後、正常に行われなかった操作を再試行できるものもあります。試行回数はユーザーのプログラムの中で指定しなければなりません。また、代替またはバックアップ装置またはセッションを代わりに使用できる場合もあります。

メジャー・コード 82 に伴うメッセージ番号は、オープン操作または獲得操作を示す範囲にある場合があります。操作がオープン操作だった場合は、エラーから回復するためには、部分的にオープンしているファイルをクローズしてから、オープンし直すことが必要です。操作が獲得操作だった場合は、再度獲得を試みる前に解放操作を実行することが必要な場合もあります。いずれの場合もファイル待機時間を十分長く指定して、サーバーがエラーから回復できるようにしなければなりません。

入出力操作での回復可能装置エラーまたはセッション・エラー

メジャー戻りコード 83 は、装置へのデータ送信または装置からのデータ受信にエラーが生じたことを示します。アプリケーション・プログラムによる回復が可能です。マイナー戻りコードと付随メッセージの両方で、問題の原因についてより特定された情報が与えられます。

このグループのエラーのほとんどは、有効でないコマンドまたはデータを装置に送ったか、有効なデータには違いないが、送った時点が正しくないか、または送り先の装置がそのデータを処理することができないことの結果です。アプリケーション・プログラムは、正常に行われなかった操作またはデータ項目をスキップして次に移ることによって、あるいはまた適切なデフォルトに置き換えることによって回復できることもあります。アプリケーションで論理エラーが生じている可能性もあります。

リファレンス

ナビゲーション・バーまたはこのトピックのリストからリファレンス・トピックを選択すれば、特定のリファレンス情報 (文字のサポート、フィールドバック域のレイアウト、その他のサポート上の考慮事項など) にすばやくアクセスできます。

2 バイト文字セットのサポート

この節では、2 バイト文字を使用する場合に必要な情報を扱います。

DBCS のプリンターとスプーリングのサポートについては、印刷を参照してください。

2 バイト文字セットについての基本情報

中国語、日本語、韓国語などの言語は、1 バイト・コードでは表しきれない多くの異なる文字を使用しています。このような言語に対するコード化文字セットを作成するため、サーバーでは各文字を表すのに 2 バイトを使用します。2 バイト・コードでコード化された文字は、2 バイト文字と呼ばれます。

154 ページの図 20 は、1 バイト・コード体系でコード化された英数字および 2 バイト・コード体系でコード化された 2 バイト文字を示しています。

2 バイト文字は 1 バイト文字と同時に 1 つのアプリケーション・プログラムで使用することができます。たとえば、データベースへの 2 バイト文字データと 1 バイト文字データの保管、2 バイト文字のテキストおよびフィールドを用いた表示画面の作成、2 バイト文字の報告書の印刷などを行うことができます。

1 バイト・コード (SBCS)	2 バイト・コード (DBCS)
A — X'C1'	A — X'42C1'
B	B — X'42C2'
1 — X'C2'	1 — X'42F1'
2	2 — X'42F2'
— X'F1'	あ — X'4481' (日本語)
— X'F2'	美 — X'457D' (日本語)
	ㄱ — X8877' (韓国語)
	橋 — X'4589' (日本語)
	进 — X'4F99' (中国語(簡体字))
	進 — X'5B70' (中国語(繁体字))

X'hhhh' は、コードが 16 進値 "hhhh" を持っていることを示す。

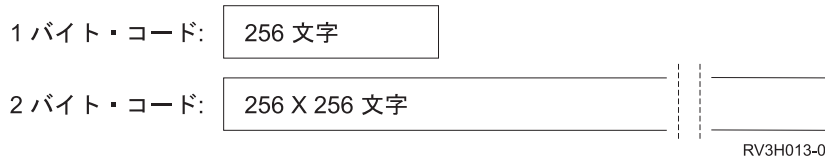


図 20. 1 バイト・コード体系と 2 バイト・コード体系

DBCS コード体系:

IBM では、2 種類の DBCS コード体系をサポートします。1 つはホスト・サーバー用であり、もう 1 つはパーソナル・コンピューター用です。

IBM ホスト・コード体系は次のコード範囲を持ちます。

第 1 バイト

16 進数 41 から 16 進数 FE

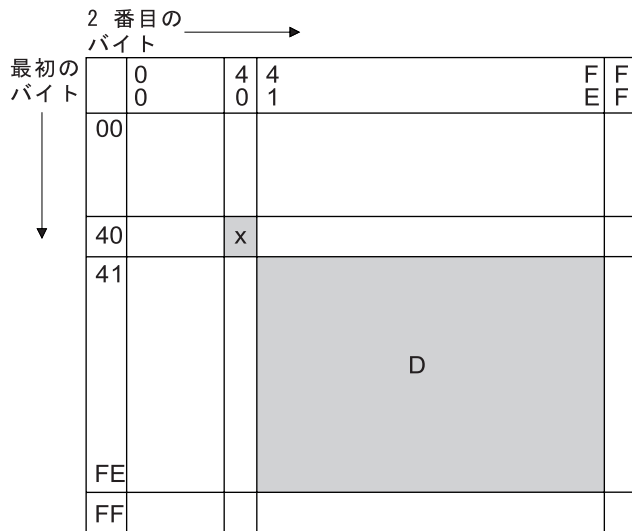
第 2 バイト

16 進数 41 から 16 進数 FE

2 バイト・ブランク

16 進数 4040

155 ページの図 21 では、縦軸を第 1 バイト、横軸を第 2 バイトとして 256 × 256 の交点つまりコード・ポイントを示しています。各コード・ポイントが 1 つの文字を表します。右下の部分は 2 バイト・コード域として有効なコード域 (D) で、X は 2 バイト文字ブランクを表わします。



D: 2 バイト・コード域
 x: 2 バイト文字ブランク

RSLH712-4

図 21. IBM ホスト・コード体系

16 進数の 41 から FE の値を DBCS コードの第 1 バイトと第 2 バイトに割り当てることによって、コードをいくつかの区にグループ化することができます (それぞれの区には 192 のコード・ポイントが含まれています)。たとえば、第 1 バイトが 16 進数 42 であるコード・グループは、42 区と呼ばれます。42 区は対応する 1 バイト EBCDIC コード・ページの英数字と同じですが、2 バイト・コードで表した英数字です。たとえば文字 A は、1 バイト EBCDIC コードでは 16 進数 C1 として表され、IBM ホスト・コードでは 16 進数 42C1 として表されます。

iSeries サーバーでは、次の 2 バイト文字セットがサポートされています。

- IBM 日本語文字セット
- IBM 韓国語文字セット
- IBM 中国語 (簡体字) 文字セット
- IBM 中国語 (繁体字) 文字セット

次の表は、おのおのの文字セットごとのコード範囲およびその文字セットでサポートされる文字の数を示しています。

表 18. IBM 日本語文字セット

区	内容	文字数
40	4040 で表されるスペース (ブランク)	1
41 から 44	漢字以外の文字 <ul style="list-style-type: none"> • ギリシャ文字、ロシア文字、ローマ数字 (41 区) • 英数字および関連記号 (42 区) • カタカナ、ひらがな、および特殊記号 (43 から 44 区) 	549
45 から 55	基本漢字	3226
56 から 68	拡張漢字	3487

表 18. IBM 日本語文字セット (続き)

区	内容	文字数
69 から 7F	ユーザー定義文字	4370 以下
80 から FE	未使用	

IBM 定義文字の総数: 7263

表 19. IBM 韓国語文字セット

区	内容	文字数
40	4040 で表されるスペース (ブランク)	1
41 から 46	ハングル文字/韓国語漢字以外の文字 (ラテン語アルファベット、ギリシャ文字、ローマ文字、日本語かな、数字、特殊記号)	939
47 から 4F	未使用	
50 から 6C	韓国語漢字	5265
6D から 83	未使用	
84 から D3	ハングル文字 (Jamo も含む)	2672
D4 から DD	ユーザー定義文字	1880 以下
DE から FE	未使用	

IBM 定義文字の総数: 8877

表 20. IBM 中国語 (簡体字) 文字セット

区	内容	文字数
40	4040 で表されるスペース (ブランク)	1
41 から 47	中国語文字以外の文字 (ラテン語アルファベット、ギリシャ文字、ロシア文字、日本語かな、数字、特殊記号)	712
48 から 6F	中国語文字: レベル 1 およびレベル 2	3755 と 3008
70 から 75	未使用	
76 から 7F	ユーザー定義文字	1880 以下
80 から FE	未使用	

IBM 定義文字の総数: 7476

表 21. IBM 中国語 (繁体字) 文字セット

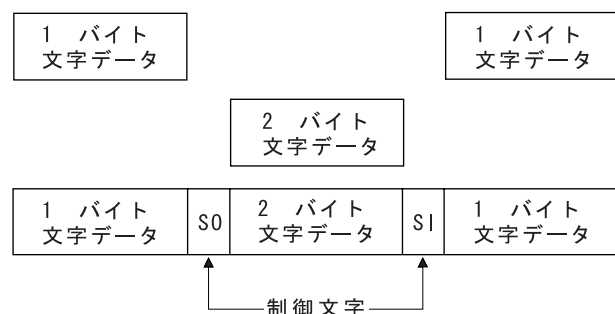
区	内容	文字数
40	4040 で表されるスペース (ブランク)	1
41 から 49	中国語文字以外の文字 (ラテン語アルファベット、ギリシ ャ文字、ローマ文字、日本語かな、数字、特殊記号)	1003
4A から 4B	未使用	
4C から 68	1 次中国語文字	5402
69 から 91	2 次中国語文字	7654
92 から C1	未使用	
C2 から E2	ユーザー定義文字	6204 以下
E3 から FE	未使用	

IBM 定義文字の総数: 14060

このコード体系は、iSeries サーバー、System/36™、System/38、およびシステム/370 サーバーに適用されま
す。パーソナル・システム/55 では、IBM パーソナル・コンピューター DBCS コード体系と呼ばれる異な
る DBCS コード体系が使用されます。IBM パーソナル・コンピューター DBCS コード体系の詳細につ
いては、IBM PS/55 の資料を参照してください。

シフト制御 2 バイト文字:

IBM ホスト・コード体系が使用される場合、サーバーは、2 バイト文字ストリングの始まりと終わりを識
別するのにシフト制御文字を使用します。シフトアウト (SO) 文字である 16 進数の 0E は、2 バイト文
字ストリングの始まりを示します。シフトイン (SI) 文字である 16 進数の 0F は、2 バイト文字の終わ
りを示します。



RSLH713-1

図 22. シフトアウト文字とシフトイン文字の配置

各シフト制御文字は英数字 1 文字分と同じスペース (1 バイト) を占有します。これに対して、2 バイト
文字は英数字 2 文字分のスペースを占有します。

2 バイト文字をグラフィック・フィールドかまたはグラフィック・データ・タイプの変数に保管するとき、シフト制御文字を使用して 2 バイト文字を囲む必要はありません。

無効な 2 バイト・コードおよび未定義の 2 バイト・コード:

無効な 2 バイト・コードとは、有効な 2 バイト・コード範囲にない 2 バイト・コードのことです。DBCS コード体系に関するトピックの中の IBM ホスト・コード体系に、有効な 2 バイト・コードの範囲が示されています。これに対して、未定義の 2 バイト・コードの場合はその 2 バイト・コードは有効ですが、そのコードに対してグラフィック記号が定義されていません。

関連資料

154 ページの『DBCS コード体系』

IBM では、2 種類の DBCS コード体系をサポートします。1 つはホスト・サーバー用であり、もう 1 つはパーソナル・コンピューター用です。

2 バイト・データの使用:

この節では、2 バイト・データを使用できる場合について、および使用上の制約事項について説明します。

2 バイト・データを使用できる場合:

2 バイト・データは、ファイル内のデータとして、メッセージのテキストとして、オブジェクト記述のテキストとして使用できるほか、高水準言語プログラムにより処理されるリテラル、定数、およびデータとしても使用できます。

- ファイルのデータとして。
 - データベース・ファイルのデータ。
 - ディスプレイ・ファイルの中の入力可能フィールドに入力されるデータおよび出力可能フィールドに表示されるデータ。
 - プリンター・ファイルの出力可能フィールドに印刷されるデータ。
 - ディスプレイ・ファイルおよびプリンター・ファイルでリテラルとして使用されるデータ。
- メッセージのテキストとして。
- オブジェクト記述のテキストとして。
- 高水準言語プログラムにより処理されるリテラル、定数、およびデータとして。

2 バイト・データが表示できるのは DBCS ディスプレイ装置だけであり、印刷できるのは DBCS プリンターだけです。2 バイト・データはディスク、テープ、ディスクおよび光ディスク装置に書き出すことができます。

2 バイト・データを使用できない場合:

2 バイト・データは、iSeries オブジェクト名として、および制御言語 (CL) その他の高水準言語のコマンド名または変数名として使用できません。さらに、英数字入出力ワークステーション上で表示または印刷出力することもできません。

2 バイト文字のサイズ:

2 バイト文字は通常、表示または印刷されるときに 1 バイト文字の 2 倍の幅になります。

フィールド長は通常は使用される 1 バイト文字の数として識別されるので、2 バイト・データ・フィールドの長さを計算するときは、2 バイト文字の幅を考慮してください。2 バイト・データが入るフィールドの長さの計算についての詳細は、DDS concepts を参照してください。

2 バイト文字の処理

2 バイト文字は数が多いため、各 2 バイト文字を識別するために、サーバーでは各英数字を識別するのに必要とされるよりも多くの情報を必要とします。

2 バイト文字には、基本 2 バイト文字と拡張 2 バイト文字の 2 つのタイプがあります。これらの文字は通常、その表示または印刷が行われる装置によって処理されます。

基本 2 バイト文字:

基本文字とは、DBCS 可能装置のハードウェアに入っている、頻繁に使用される 2 バイト文字のことです。

基本文字とは、DBCS 可能装置のハードウェアに入っている、頻繁に使用される 2 バイト文字のことです。装置に保管されている 2 バイト文字の数は、サポートされている言語および装置の記憶域サイズによって異なります。DBCS 可能装置では、基本文字の表示および印刷は、オペレーティング・システムの外字処理機能を使用しないで行うことができます。

拡張 2 バイト文字:

外字の処理の場合、装置はサーバーの援助を必要とします。サーバーから装置にあらかじめ文字の形を知らせてやらなければ、装置では文字を表示または印刷することができません。外字は DBCS 可能装置ではなく、DBCS フォント表に保管されています。

外字を表示または印刷するときは、装置はオペレーティング・システムの制御下で DBCS フォント表から外字を受け取ります。

外字処理は、DBCS フォント表に保管されている文字を DBCS 可能装置で使用可能にするのに必要な、オペレーティング・システムの機能です。

外字処理を要求するには、2 バイト・データを処理するディスプレイ・ファイルまたはプリンター・ファイルを作成するファイル作成コマンド (CRTDSPF コマンドまたは CRTPRTF コマンド) で 2 バイト外字パラメーター IGCEXNCHR(*YES) を指定してください。IGCEXNCHR(*YES) はデフォルト値なので、サーバーはそれ以外の指定がない限り、自動的に外字を処理します。ファイル変更 (CHGDSPF または CHGPRTF) コマンドまたはファイル・オーバーライド (OVRDSPF または OVRPRTF) コマンドを使用して、このファイル属性を変更することができます。たとえば、ディスプレイ・ファイル DBCSDSPF をオーバーライドして外字を処理できるようにするには、次のように入力してください。

```
OVRDSPF DSPF(DBCSDSPF) IGCEXNCHR(*YES)
```

注:

1. サーバーは、英数字ファイルの処理の場合には、IGCEXNCHR パラメーターを無視します。
2. 日本語 5583 プリンターを使用して外字を印刷する時は、IBM Advanced DBCS Printer Support for iSeries ライセンス・プログラムの漢字印刷機能を使用しなければなりません。このユーティリティの使用方法については、「*Kanji Print Function User's Guide and Reference*」を参照してください。

拡張 2 バイト文字が処理されなかったときに起こる現象:

このトピックでは、拡張 2 バイト文字が処理されない場合の結果について説明します。

外字が処理されない場合は、次のような結果になります。

- 基本 2 バイト文字は表示および印刷されます。

- 画面では、外字を表示するはずの位置にサーバーが未定義文字を表示します。
- 印刷出力では、外字を印刷するはずの位置にサーバーが未定義文字を印刷します。
- 外字はたとえ表示または印刷されなくとも、サーバーに正しく保管されます。

DBCS 装置ファイル・サポート

この節では、DBCS 可能装置ファイルについて説明し、DBCS 可能装置ファイルの処理に関する考慮事項について説明しています。

データ記述仕様 (DDS) はファイルの記述に使用される言語ですが、DBCS 可能装置ファイルに使用することができます。

DBCS ファイルとは:

DBCS ファイルとは、2 バイト・データが入っているか、または 2 バイト・データを処理するのに使用されるファイルのことです。それ以外のファイルは、英数字ファイルと呼ばれます。

次のタイプの装置ファイルは、DBCS ファイルにすることができます。

- ディスプレイ
- プリンター
- テープ
- ディスケット
- ICF

DBCS ファイルの指定が必要な場合:

このトピックでは、ファイルが DBCS であることを指定する必要がある状況について説明します。

次のような状況の 1 つまたは複数に該当する場合には、ファイルが DBCS ファイルであることを指定しなければなりません。

- ファイルが 2 バイト文字を含む入力データを受け取るか、2 バイト文字を含む出力データを表示または印刷する。
- ファイルに 2 バイト文字リテラルが含まれている。
- ファイルがその処理時にファイルで使用される DDS に、2 バイト文字リテラルを含む (固定情報フィールドやエラー・メッセージなど)。
- ファイルの DDS に DBCS キーワードが含まれる。
- ファイルに 2 バイト・データを保管する (データベース・ファイル)。

DBCS ファイルの指定の方法:

装置ファイルが DBCS ファイルである場合は、サーバーが 2 バイト・データを正しく処理するために、その旨を指定しなければなりません。このトピックに示されている方法で、DBCS ファイルを指定することができます。

- DDS による方法:
 - DDS は次のデータ・タイプのフィールドを提供します。
 - *DBCS 専用*フィールドは、2 バイト文字だけを表示して受け取ります。DBCS 専用フィールドの 2 バイト文字は常にシフトアウト文字とシフトイン文字で囲まれており、このシフトアウト文字とシフトイン文字は対になっていなければなりません。

- *DBCS* 混用フィールドは、1 バイト文字および 2 バイト文字の両方を表示して受け取ります。2 バイト文字はシフトアウト文字とシフトイン文字で囲まれており、このシフトアウト文字とシフトイン文字は対になっていなければなりません。
- *DBCS* 択一フィールドは、1 バイト文字と 2 バイト文字の両方ではなくいずれか一方を表示して受け取ります。2 バイト文字はシフトアウト文字とシフトイン文字で囲まれます。
- *DBCS* グラフィック・フィールドは、2 バイト文字だけを表示して受け取ります。*DBCS* グラフィック・フィールドの文字は、シフトアウト文字およびシフトイン文字を持ちません。なお、*iSeries* の *DBCS* グラフィック・フィールドは、システム/370 の *DBCS* フィールドと同等です。
- *ICF* ファイルでは、*DBCS* 混用データ・タイプ (タイプ O) を用いてフィールドを定義します。
- プリンター・ファイルでは、*DBCS* 混用データ・タイプ (タイプ O) および *DBCS* グラフィック・データ・タイプ (タイプ G) を用いてフィールドを定義します。
- ディスプレイ・ファイルでは、*DBCS* 専用データ・タイプ (タイプ J)、*DBCS* 択一データ・タイプ (タイプ E)、*DBCS* 混用データ・タイプ (タイプ O) または *DBCS* グラフィック・データ・タイプ (タイプ G) を用いてフィールドを定義します。
- *DDS* のデフォルト (*DFT*) やエラー・メッセージ (*ERRMSG*) などのキーワードで指定されるリテラルなど、処理時にファイルで使用される 2 バイト文字リテラルを使用していきます。

注: また、2 バイト文字リテラルをファイル内のテキストおよび注記として使用することもできます (たとえば *DDS* キーワード *TEXT* を使用)。ただし、サーバーでは、*DBCS* の使用が 2 バイト文字の注記だけのファイルについては、これを *DBCS* ファイルとは見なしません。

- ディスプレイ・ファイルおよびプリンター・ファイルに *DDS* の代替データ・タイプ (*IGCALTTYP*) キーワードを指定します。このキーワードを指定すると、ディスプレイ・ファイルおよびプリンター・ファイルを、英数字と 2 バイト文字の両方のアプリケーションで使用することができます。*IGCALTTYP* キーワードを有効にすると、2 バイト・データをファイルで使用することができます。

*IGCDTA(*YES)* の値を指定して、ディスプレイ・ファイルまたはプリンター・ファイルの作成、変更、またはオーバーライドを行うことにより、*IGCALTTYP* キーワードを有効にしてください。次の装置ファイル・コマンドで *IGCDTA(*YES)* を指定することにより、ディスプレイ・ファイルおよびプリンター・ファイルの *IGCALTTYP* キーワードを有効にすることができます。

- ディスプレイ・ファイル作成 (*CRTDSPF*)
- プリンター・ファイル作成 (*CRTPRTF*)
- ディスプレイ・ファイル変更 (*CHGDSPF*)
- プリンター・ファイル変更 (*CHGPRTF*)
- ディスプレイ・ファイル・オーバーライド (*OVRDSPF*)
- プリンター・ファイル・オーバーライド (*OVRPRTF*)

*IGCDTA(*NO)* を指定すると、*IGCALTTYP* キーワードは効力を持たないので、ファイルで使用できるのは、英数字データだけになります。ファイルの変更またはオーバーライドによって、*IGCALTTYP* キーワードを有効にしても、ファイルの *DDS* が変更されることはありません。

DDS ですでに *DBCS* 機能を指定してある場合は、*IGCALTTYP* 機能を使用する場合を除き、ファイル作成コマンドで *IGCDTA(*YES)* を指定する必要はありません。その代わりに、ファイルに *DDS* で示していない *DBCS* 機能がある時は、*IGCDTA(*YES)* を指定してください。たとえば、ファイルに 2 バイト・データを入れる予定である場合は、ファイル作成コマンドで *IGCDTA(*YES)* を指定してください。

- 次の装置ファイル作成コマンドに *IGCDTA(*YES)* を指定します。

- ディスプレイ・ファイル作成 (CRTDSPF)
- プリンター・ファイル作成 (CRTPRTF)
- テープ装置ファイル作成 (CRTTAPF)
- 次のデータベース・ファイル作成コマンドに IGCDTA(*YES) を指定します。
 - 物理ファイル作成 (CRTPF)
 - ソース物理ファイル作成 (CRTSRCPF)

DBCS のファイルの指定が正しくない場合:

ファイルが DBCS ファイルであることを正しく指定しないと、エラーが発生する可能性があります。

- プリンター・ファイルの場合、プリンター・データ管理機能は、プリンターに対する出力データには 2 バイト・データは入っていないと想定します。最終結果は、データの印刷が行われるプリンターのタイプ、および使用しているプリンター・ファイルの印刷不能文字置き換えパラメーターの状況によって異なります。

印刷不能文字置き換えオプションが選択されている場合は、プリンター・データ管理機能では、シフト制御文字を印刷不能文字と解釈して、ブランクで置き換えます。2 バイト・データ自体は英数字データとして解釈され、プリンターはそのまま印刷しようと試みます。印刷された 2 バイト・データは、意味をなしません。

印刷不能文字置き換えオプションが選択されておらず、プリンターが英数字プリンターである場合は、2 バイト・データは制御文字も含め、そのままプリンターに送られます。ほとんどの英数字プリンターではシフト制御文字はサポートされず、プリンターでエラーが起こります。

印刷不能文字置き換えオプションが選択されておらず、プリンターが DBCS 装置である場合は、2 バイト・データは外字データを除いて印刷されます。ファイルが DBCS であることが指定されていないので、サーバーは外字処理を実行しません。したがって外字は、未定義 2 バイト文字を表す記号で印刷されます。

- ディスプレイ・ファイルの場合、ディスプレイ・データ管理機能は、ディスプレイに対する出力データには 2 バイト・データが入っていないものと想定します。最終の結果は、ディスプレイが英数字ディスプレイか DBCS ディスプレイかによって異なります。

ディスプレイが英数字ディスプレイの場合は、2 バイト・データは英数字データとして解釈されます。シフト制御文字は、ブランクとして表示されます。表示された 2 バイト・データは意味をなしません。

ディスプレイが DBCS ディスプレイの場合は、2 バイト・データは、外字を除いて表示されます。サーバーでは、データに対して外字処理を実行しません。したがって、外字は未定義 2 バイト文字を表す記号で表示されます。

- ソース・ファイルが DBCS ファイルとして指定されていない場合は、サーバーは DBCS テキストのリテラルを、2 バイト文字リテラルとして認識しません。

プリンター・ファイルで DBCS を使用可能にする:

サーバーは多くの場合、最終的に印刷または表示されるデータの生成にプリンター・ファイルを使用しています。このような場合、IBM 提供のプリンター・ファイルのいずれかを使用して、データはまずスプール・ファイルに入れます。データはそのスプール・ファイルから取り出され、ユーザーの要求に応じて表示または印刷されます。

データに 2 バイト文字が入っている場合、そのデータをスプール・ファイルに入れるのに使用されるプリンター・ファイルは、2 バイト・データを処理することができなければなりません。プリンター・ファイルで 2 バイト・データを処理することができるのは、そのファイルに関して、IGCDTA パラメーターに *YES が指定されているときです。ほとんどの場合、サーバーでは 2 バイト・データがあることを認識し、使用されるプリンター・ファイルで 2 バイト・データの処理を間違いなく行なうために、適切な措置を講じます。

しかし、サーバーで 2 バイト・データがあることを認識できず、2 バイト・データの処理ができないプリンター・ファイルの使用を試みる場合があります。このような場合は、ディスプレイまたはプリンターでの出力は、判読できないものになります。2 バイト文字を含むオブジェクト記述を英数字装置で表示または印刷しようとすると、このようなことが起こります。

2 バイト文字を表示または印刷する場合に、正しい結果が得られるようにするには、次のような注意が必要です。ユーザー側で処置が必要になるのは、1 バイト文字の言語が 2 次言語として導入されている場合です。DBCS バージョンのプロダクトの一部として配布されたプリンター・ファイルは、常に DBCS データの処理が可能です。

プロダクトまたは機能の導入後に、次の推奨処置を実行する必要があります。

1. サーバーに接続されているプリンターおよびディスプレイ装置すべてで、DBCS の使用が可能である場合は、すべてのプリンター・ファイルを 2 バイト・データ用として使用可能にすることができます。1 バイト・バージョンの 2 次言語機能の一部として配布された IBM 提供のプリンター・ファイルの場合は、次のコマンドを出すことによって、すべてのプリンター・ファイルを 2 バイト文字使用可能にすることができます。

```
CHGPRTF FILE(*ALL/*ALL) IGCDTA(*YES)
```

このコマンドが完了すると、すべてのライブラリーのプリンター・ファイルはすべて、2 バイト・データが使用可能になります。この変更は永続変更です。

2. サーバーに接続されているプリンターおよびディスプレイ装置のすべてが DBCS 可能とは限らない場合は、すべての IBM 提供プリンター・ファイルを使用可能にすることはできません。

その代わりにサーバーのライブラリー探索機能を使用して、どのプリンター・ファイルを特定のジョブで使用するかということ制御します。2 バイト・データが使用される可能性がある時は、ジョブに関するライブラリー・リストで、DBCS 可能なプリンター・ファイルがまず見つかるようにするべきです。これに対して、1 バイト・データしか使用されないと予想される場合は、DBCS 可能になっていないプリンター・ファイルがまず見つかるように、ライブラリー・リストをセットアップすべきです。このようにすれば、プリンター・ファイル機能は処理されるデータのタイプに一致することになります。使用するプリンター・ファイルのタイプに関する決定は、処理されるデータのタイプに基づいてなされます。実際にデータを表示または印刷するのに使用される装置によって、この決定が影響を受けることもあります。

場合によっては、プリンター・ファイルを永続的に変更するのではなく、一時的に DBCS 使用可能にする方がよい場合もあります。ある特定のジョブで、OVRPRTF コマンドを使用することによって、一時的な変更を行うことができます。

ある特定のプリンター・ファイルを一時的に使用可能にするには、次のコマンドが使用できます。

```
OVRPRTF FILE(filename) IGCDTA(*YES)
```

このコマンドで、filename とは DBCS を使用可能にしたいプリンター・ファイルの名前です。

DBCS ディスプレイのサポート

この節では、2 バイト文字の表示について説明します。

シフト制御 2 バイト文字の挿入:

DBCS 専用フィールドには、サーバーによって自動的にシフト制御文字が挿入されます。

混用フィールドまたは択一フィールドにシフト制御文字を挿入するには、以下のステップを実行してください。

1. カーソルを 2 バイト・データを挿入したいフィールドに位置付けます。
2. シフト制御文字挿入キーを押します (使用する DBCS ディスプレイ装置の使用者の手引きに従ってください)。

サーバーは次に示すように、一対のシフト制御文字を同時に挿入します (ここでは、 0_E はシフトアウト文字を表し、 0_F はシフトイン文字を表しています)。

0_E0_F

カーソルは、サーバーによってシフトイン文字の下に置かれ、キーボードは挿入モードになります。シフト制御文字の間に 2 バイト文字を挿入します。2 バイト文字を挿入するには、カーソル位置から 2 バイト文字の入力を始めます。たとえば、2 バイト文字ストリング D1D2D3 は次のように入力します (ここでは、 0_E はシフトアウト文字を表し、 0_F はシフトイン文字を表し、D1、D2、および D3 は 3 つの 2 バイト文字を表しています)。

$0_ED1D2D30_F$

すでにフィールドにシフト制御文字が入っているかどうかを知るには、シフト制御文字表示キーを押してください。

DBCS グラフィック・フィールドに 2 バイト文字を保管する場合は、シフト制御文字の使用は必要ありません。シフト制御文字をグラフィック・フィールドに挿入しないでください。

表示される拡張 2 バイト文字の数:

サーバーでは、日本語表示画面にさまざまな外字を 1 時点で 512 文字まで表示することができます。それ以上の外字は、未定義文字として表示されますが、サーバーには正しく保管されています。

表示画面上の DBCS 入力フィールドの数:

DBCS 入力フィールドを使用すると、1 つの画面で許される入力フィールドの合計数に影響を及ぼします。ローカル 5250 ディスプレイの場合は、入力フィールドは最大 256 まで指定することができます。しかし、DBCS フィールドを使用する場合、3 つ使用するごとにフィールド最大数は 1 つずつ減少します。

たとえば、画面上に DBCS フィールドが 9 つある場合、フィールド最大数は $256 - (9/3) = 253$ 入力フィールドになります。

英数字ワークステーションで 2 バイト・データを表示した場合の影響:

英数字ディスプレイ装置では、2 バイト・データを正しく表示することはできません。

英数字ディスプレイ装置で 2 バイト・データを表示しようとする、次のような結果になります。

- サーバーはそのディスプレイ装置に照会メッセージを送り、2 バイト・データを使用するそのプログラムを続行したいのか、取り消したいのかを問い合わせます。

- プログラムを使用し続けたい場合は、サーバーはシフト制御文字を無視し、2 バイト文字を 1 バイト文字であるかのように解釈します。表示される 2 バイト・データは意味をなしません。

DBCS ファイルのコピー

スプール DBCS ファイルも、非スプール DBCS ファイルも同様にコピーすることができます。

関連概念

60 ページの『UCS-2 グラフィック・フィールドの制約事項』

コピー元またはコピー先が UCS-2 グラフィック・フィールドの場合、いくつかの制約事項があります。

スプール DBCS ファイルのコピー:

スプール・ファイル・コピー (CPYSPLF) コマンドを使用することによって、2 バイト文字データを持つスプール・ファイルをコピーします。ただし、ファイルのコピー先のデータベース・ファイルは、IGCDTA(*YES) の値を指定して作成されていなければなりません。

2 バイト・データが入っているデータベース・ファイルに、スプール・ファイルをコピーする場合に、シフトアウト文字用に余分の桁が必要になります。このシフトアウト文字は、レコードの制御情報とユーザー・データの間に入られます。次の表は、制御文字 (CTLCHAR) キーワードに対して指定される値に基づいた、シフトアウト文字の桁番号を示してあります。

CTLCHAR の値	シフトアウト文字の桁番号
*NONE	1
*FCFC	2
*PRTCTL	5
*S36FMT	10

非スプール DBCS ファイルのコピー:

ファイル・コピー (CPYF) コマンドを使用することによって、2 バイト文字データを 1 つのファイルから別のファイルへコピーすることができます。

2 バイト・データベース・ファイルから英数字データベース・ファイルへデータをコピーするときは、次のパラメーターのいずれか 1 つを CPYF コマンドで指定してください。

- 両方のファイルが共にソース・ファイルの場合、または両方のファイルが共にデータベース・ファイルの場合は、FMTOPT(*MAP) パラメーターまたは FMTOPT(*NOCHK) パラメーターのいずれかを指定することができます。
- 一方のファイルがソース・ファイルで、他方のファイルがデータベース・ファイルである場合は、FMT(*CVTSRC) パラメーターを指定してください。

DBCS ファイルを英数字ファイルへコピーする時は、ファイル・タイプの違いを示す通知メッセージが、サーバーによって出されます。

物理ファイルまたは論理ファイルから物理ファイルへのコピーにおいては、コピー元ファイルとコピー先ファイルに同じ名前を持つフィールドがあっても、フィールドのデータ・タイプが次表のような時は、ファイル・コピー機能の FMTOPT(*MAP) オプション、または FMTOPT(*NOCHK) オプションのいずれかを指定しなければなりません。

コピー元ファイルのフィールドのデータ・タイプ

A (文字)
O (DBCS 混用)
O (DBCS 混用)
E (DBCS 択一)
J (DBCS 専用)
O (DBCS 混用)
E (DBCS 択一)
G (DBCS グラフィック)
G (DBCS グラフィック)
G (DBCS グラフィック)
G (UCS-2 グラフィック)
G (UCS-2 グラフィック)
G (UCS-2 グラフィック)
G (UCS-2 グラフィック)
G (UCS-2 グラフィック)
A (文字 (CCSID 非 65535))
O (DBCS 混用 (CCSID 非 65535))
E (DBCS 択一 (CCSID 非 65535))
J (DBCS 専用 (CCSID 非 65535))
G (DBCS グラフィック)
A (UTF-8)
A (UTF-8)
A (UTF-8)
A (UTF-8)
A (UTF-8)
A (UTF-8)
A (文字 (CCSID 非 65535))
O (DBCS 混用 (CCSID 非 65535))
E (DBCS 択一 (CCSID 非 65535))
J (DBCS 専用 (CCSID 非 65535))
G (DBCS グラフィック非 65535)
G (UCS-2 グラフィック)
G (UTF-16)
G (UTF-16)
G (UTF-16)
G (UTF-16)
G (UTF-16)
G (UTF-16)
G (UTF-16)
G (UTF-16)
A (文字 (CCSID 非 65535))
O (DBCS 混用 (CCSID 非 65535))
E (DBCS 択一 (CCSID 非 65535))
J (DBCS 専用 (CCSID 非 65535))
G (DBCS グラフィック非 65535)
G (UCS-2 グラフィック)

コピー先ファイルのフィールドのデータ・タイプ

J (DBCS 専用)
J (DBCS 専用)
E (DBCS 択一)
J (DBCS 専用)
G (DBCS グラフィック)
G (DBCS グラフィック)
G (DBCS グラフィック)
J (DBCS 専用)
O (DBCS 混用)
E (DBCS 択一)
A (文字 (CCSID 非 65535))
O (DBCS 混用 (CCSID 非 65535))
E (DBCS 択一 (CCSID 非 65535))
J (DBCS 専用 (CCSID 非 65535))
G (DBCS グラフィック)
G (UCS-2 グラフィック)
G (UCS-2 グラフィック)
G (UCS-2 グラフィック)
G (UCS-2 グラフィック)
G (UCS-2 グラフィック)
A (文字 (CCSID 非 65535))
O (DBCS 混用 (CCSID 非 65535))
E (DBCS 択一 (CCSID 非 65535))
J (DBCS 専用 (CCSID 非 65535))
G (DBCS グラフィック非 65535)
G (UTF-16)
G (UCS-2 グラフィック)
A (UTF-8)
A (UTF-8)
A (UTF-8)
A (UTF-8)
A (UTF-8)
A (UTF-8)
A (文字 (CCSID 非 65535))
O (DBCS 混用 (CCSID 非 65535))
E (DBCS 択一 (CCSID 非 65535))
J (DBCS 専用 (CCSID 非 65535))
G (DBCS グラフィック非 65535)
A (UTF-8)
G (UCS-2 グラフィック)
G (UTF-16)
G (UTF-16)
G (UTF-16)
G (UTF-16)
G (UTF-16)
G (UTF-16)
G (UTF-16)
G (UTF-16)

DBCS 専用フィールドまたは DBCS グラフィック・フィールドにコピーするために CPYF コマンドで FMTOPT(*MAP) を使うときは、コピー元ファイルの対応フィールドが以下のものであってはなりません。

- 2 バイト未満の文字フィールド
- 奇数バイトの長さの文字フィールド
- 奇数バイトの長さの DBCS 混用フィールド

注: UCS-2 グラフィック・フィールドから、またはそこに対して FMTOPT(*MAP) によるコピーを行う場合、CCSID 65535 を指定できるのは DBCS グラフィック・タイプだけです。UCS-2 グラフィックは、CCSID 65535 であってはなりません。

コピー元フィールドに上記のいずれかを指定すると、エラー・メッセージが出されます。

2 バイト・データを 1 つのデータベース・ファイルから別のデータベース・ファイルへコピーする場合、FMTOPT(*MAP) パラメーターを指定すれば 2 バイト・データは正しくコピーされます。サーバーは 2 バイト・データに対する埋め込みおよび切り捨てを正しく実行して、データ保全性を確保します。

DBCS 混用フィールドをグラフィック・フィールドにコピーするのに CPYF コマンドで FMTOPT(*MAP) を指定する場合、DBCS 混用フィールドに SBCS データ (ブランクを含む) があると、変換エラーが発生します。

アプリケーション・プログラムでの DBCS に関する考慮事項

この節では、2 バイト・データを処理するアプリケーションを作成する際の考慮事項について説明します。

2 バイト・データを処理するアプリケーション・プログラムの設計:

2 バイト・データを処理するアプリケーション・プログラムを設計する場合、このトピックに説明されているような考慮事項があります。

英数字データを処理するアプリケーション・プログラムを設計する場合と同じ要領に加えて、次の補足事項も含めて 2 バイト・データを処理するアプリケーション・プログラムを設計してください。

- データベース・ファイルの中で使用されている 2 バイト・データを識別する。
- 2 バイト・データを使用する画面様式および印刷様式を設計する。
- 必要に応じて、対話式アプリケーション・プログラムの 2 バイト・データ入力手段として、DBCS 変換を用意する。DBCS 変換用の DDS キーワード (IGCCNV) を使用して、ディスプレイ・ファイルに DBCS 変換を指定してください。DBCS ワークステーションには各種の 2 バイト・データ入力手法が用意されているので、2 バイト・データの入力に iSeries の DBCS 変換機能を使用する必要はありません。
- プログラムで使用される 2 バイト文字メッセージを作成する。
- 外字処理を指定して、サーバーですべての 2 バイト・データが印刷および表示されるようにする。詳細については、159 ページの『拡張 2 バイト文字』を参照してください。
- 2 バイト文字を追加して定義する必要があるかどうかを判別します。文字作成ユーティリティ (CGU) を使用して、ユーザー定義文字を定義して維持することができます。CGU については、*適用業務開発ツール (ADTS)・セット/400: 文字作成ユーティリティ (CGU) のマニュアル*を参照してください。

2 バイト・データを処理するアプリケーション・プログラムを作成する場合には、2 バイト文字データが必ず 2 バイト単位で処理されるように注意し、1 つの 2 バイト文字を分割しないようにしてください。

英数字アプリケーション・プログラムの DBCS アプリケーション・プログラムへの変更:

英数字アプリケーション・プログラムで外部記述ファイルを使用している場合は、外部記述ファイルを変更するだけで、そのアプリケーション・プログラムを DBCS アプリケーション・プログラムに変更することができます。

アプリケーション・プログラムを変換するには、以下のステップを実行してください。

1. 変更したい英数字ファイルのソース・ステートメントの複製コピーを作成する。

2. 英数字固定情報およびリテラルを、2 バイト文字固定情報およびリテラルに変更する。
3. ファイルの該当フィールドを混用 (O) データ・タイプに変更するか、または代替データ・タイプ (IGCALTTYP) DDS キーワードを指定することにより、2 バイト・データと英数字データの両方をこれらのフィールドに入れることができるようにする。2 バイト・データにはより多くのスペースが必要なので、フィールドの長さの変更が必要になることがあります。
4. 変換されたファイルを別のライブラリーに保管する。そのファイルには、英数字バージョンと同じ名前を付けてください。
5. 変更されたファイルをジョブで使用する場合、そのファイルを使用するジョブのライブラリー・リストをライブラリー・リスト変更 (CHGLIBL) コマンドを使用して変更します。これによって、英数字バージョンのファイルが保管されているライブラリーよりも前に DBCS ファイルが保管されているライブラリーが検査されます。

DBCS フォント表

DBCS フォント表には、サーバーで使用される 2 バイト外字のイメージが入っています。サーバーは、これらのイメージを使用して、外字を表示および印刷します。

下記の DBCS フォント表は、ユーザーが保管または復元することのできるオブジェクトです。これらのフォント表は、DBCS 言語バージョンの OS/400 ライセンス・プログラムと共に配布されます。

QIGC2424

外字を 24 × 24 ドット・マトリックス・イメージで表示および印刷するのに使われる日本語 DBCS フォント表。サーバーはこの表を日本語ディスプレイ装置、ディスプレイ装置に接続されたプリンター、5227-1 型プリンター、および 5327-1 型プリンターで使用します。

QIGC2424C

外字を 24 × 24 ドット・マトリックス・イメージで印刷するのに使われる中国語 (繁体字) DBCS フォント表。サーバーはこの表を 5227-3 型プリンターおよび 5327-3 型プリンターで使用します。

QIGC2424K

外字を 24 × 24 ドット・マトリックス・イメージで印刷するのに使われる韓国語 DBCS フォント表。サーバーはこの表を 5227-2 型プリンターおよび 5327-2 型プリンターで使用します。

QIGC2424S

外字を 24 × 24 ドット・マトリックス・イメージで印刷するのに使われる中国語 (簡体字) DBCS フォント表。サーバーはこの表を 5227-5 型プリンターで使用します。

QIGC3232

文字を 32 × 32 ドット・マトリックス・イメージで印刷するのに使われる日本語 DBCS フォント表。サーバーはこの表を 5583 プリンターおよび 5337-1 型プリンターで使用します。

QIGC3232S

文字を 32 × 32 ドット・マトリックス・イメージで印刷するのに使われる中国語 (簡体字) DBCS フォント表。サーバーはこの表を 5337-R05 型プリンターで使用します。

すべての DBCS フォント表のオブジェクト・タイプは *IGCTBL です。ユーザー定義文字を DBCS フォント表に追加する場合の詳細については、「適用業務開発ツール (ADTS)・セット/400: 文字作成ユーティリティー (CGU)」のマニュアルを参照してください。

DBCS フォント表に関するコマンド:

このトピックにリストされているコマンドを使用して、DBCS フォント表を管理および使用することができます。

- DBCS フォント表検査 (CHKIGCTBL)
- DBCS フォント表コピー (CPYIGCTBL)
- DBCS フォント表削除 (DLTIGCTBL)
- 文字作成ユーティリティ開始 (STRCGU)
- フォント管理援助機能開始 (STRFMA)

DBCS フォント表の存在の有無の探索:

DBCS フォント表検査 (CHKIGCTBL) コマンドを使用して、DBCS フォント表がサーバーに存在するかどうかを調べることができます。

たとえば、表 QIGC2424 が存在するかどうかを調べるには、次のように入力してください。

```
CHKIGCTBL IGCTBL(QIGC2424)
```

表が存在しない場合は、サーバーはメッセージを出して応答します。表が存在する場合は、サーバーは戻るだけでメッセージは出しません。

新しいタイプの DBCS ワークステーションの追加時には、表の存在の有無を検査して、その装置で使用する表がサーバーに存在することを確認してください。

関連情報

DBCS フォント表検査 (CHKIGCTBL) コマンド

DBCS フォント表のテープまたはディスクへのコピー:

DBCS フォント表コピー (CPYIGCTBL) コマンドを使用して、DBCS フォント表をテープまたはディスクにコピーすることができます。

システム保管 (SAVSYS) コマンドを使用すると DBCS フォント表も保管されるので、通常のサーバー・バックアップを行う場合には CPYIGCTBL コマンドを使用する必要はありません。

関連情報

DBCS フォント表コピー (CPYIGCTBL) コマンド

DBCS 表をテープまたはディスクにコピーする時点:

以下の場合には、DBCS フォント表をテープまたはディスクにコピーしてください。

- その表の削除前。
- 新しいユーザー定義文字の表への追加後。
- 別のサーバーでの表の使用を計画した時。

DBCS 表をテープまたはディスクにコピーする方法:

DBCS フォント表をテープまたはディスクへコピーするには、以下のステップを実行してください。

1. テープまたはディスクを、*DATA 様式に初期設定しておきます。必要に応じて、ディスク初期設定 (INZDKT) コマンドで FMT(*DATA) パラメーターを指定し、テープまたはディスクを初期設定します。テープの初期設定の詳細な方法については、テープ・ファイルのトピックを参照してください。
2. 初期設定済みテープまたはディスクをサーバーにロードします。

3. CPYIGCTBL コマンドを次のように入力してください。
 - a. OPTION(*OUT) 値を選択します。
 - b. DEV パラメーターに、表をコピーしたい装置を指定します。
 - c. SELECT および RANGE パラメーターを使用して、サーバーからコピーしたい表の部分を指定します。SELECT および RANGE パラメーターの値の選択に関する詳細については、CL のトピックの CPYIGCTBL コマンドの説明を参照してください。

下に挙げるのは、DBCS フォント表を取り外し可能媒体にコピーするのに使用する CPYIGCTBL コマンドの 2 つの例です。

- DBCS フォント表 QIGC2424 をディスクットにコピーするには、次のように入力してください。

```
CPYIGCTBL IGCTBL(QIGC2424) OPTION(*OUT) +  
DEV(QDKT)
```

- DBCS フォント表 QIGC2424 からユーザー定義文字だけをテープにコピーするには、次のように入力してください。

```
CPYIGCTBL IGCTBL(QIGC2424) OPTION(*OUT) +  
DEV(QTAP01) SELECT(*USER)
```

4. Enter キーを押します。サーバーは DBCS フォント表を、指定された媒体にコピーします。
5. サーバーが表のコピーを終了したら、テープまたはディスクットを取り外します。

DBCS フォント表のテープまたはディスクットからのコピー:

DBCS フォント表コピー (CPYIGCTBL) コマンドを使用して、DBCS フォント表をテープまたはディスクットからサーバーへコピーすることができます。

次の条件が満たされる場合、サーバーは DBCS フォント表の内容をコピーする時にその表を自動的に再作成します。

- 指定された表が、サーバーに存在していない。
- 表のコピー元の媒体に、IBM 定義の 2 バイト文字のすべてが入っている。
- SELECT(*ALL) または SELECT(*SYS) が CPYIGCTBL コマンドで指定されている。

関連情報

DBCS フォント表コピー (CPYIGCTBL) コマンド

DBCS 表をテープまたはディスクットからコピーする方法:

DBCS フォント表を、テープまたはディスクットからサーバーにコピーするには、次のようにしてください。

1. サーバーに、表のコピー元の取り外し可能媒体をロードします。
2. CPYIGCTBL コマンドを次のように入力してください。
 - a. OPTION(*IN) の値を選択します。
 - b. DEV パラメーターを使用して、DBCS フォント表のコピー元の装置を選択します。
 - c. SELECT および RANGE パラメーターを使用して、テープまたはディスクットからコピーしたい表の部分を指定します。CPYIGCTBL コマンドの説明と、SELECT および RANGE パラメーターの値の選択に関する指示は、CL に関するトピックを参照してください。

以下は、DBCS フォント表をサーバーにコピーするのに使用するコマンドの 2 つの例です。

- DBCS フォント表 QIGC2424 をディスクットからコピーするには、次のように入力してください。


```
CPYIGCTBL IGCTBL(QIGC2424) OPTION(*IN) +  
DEV(QDKT)
```

- DBCS フォント表 QIGC2424 からユーザー定義文字だけをテープからコピーし、表の中のユーザー定義文字をテープからのユーザー定義文字で置き換えるには、次のように入力してください。

```
CPYIGCTBL IGCTBL(QIGC2424) OPTION(*IN) +  
DEV(QTAP01) SELECT(*USER) RPLIMG(*YES)
```

3. Enter キーを押します。サーバーは DBCS フォント表を、テープまたはディスクからシステムにコピーします。
4. サーバーが表のコピーを終了したら、テープまたはディスクを取り外します。

DBCS フォント表の削除:

DBCS フォント表削除 (DLTIGCTBL) コマンドを使用して、DBCS フォント表をサーバーから削除することができます。

関連情報

DBCS フォント表削除 (DLTIGCTBL) コマンド

DBCS フォント表の削除時点:

使用していない DBCS フォント表を削除して、記憶域スペースを解放します。たとえば、サーバーで日本語プリンター 5583 または 5337 を使用する予定がない場合はフォント表 QIGC3232 は必要がなく、削除することができます。

DBCS フォント表の削除方法:

表を削除するときは、以下のステップを実行してください。

1. 必要に応じて、表をテープまたはディスクにコピーします。詳細については、169 ページの『DBCS フォント表のテープまたはディスクへのコピー』を参照してください。表を削除する前に取り外し可能媒体にコピーしない場合は、将来使用できる表のコピーを持たないこととなります。
2. その表を使用するすべての装置をオフに構成変更します。
3. DLTIGCTBL コマンドを入力します。

たとえば、DBCS フォント表 QIGC3232 を削除するには、次のように入力してください。

```
DLTIGCTBL IGCTBL(QIGC3232)
```

4. Enter キーを押します。サーバーは照会メッセージ CPA8424 をシステム・オペレーター・メッセージ待ち行列に送って、DBCS 表の削除をするかどうかを確認します。
5. 照会メッセージに応答します。サーバーは表の削除を完了すると、メッセージを出します。

注: DBCS フォント表を使用している装置が、現在オンに構成変更されている場合は、その表を削除してはなりません。また、影響を受ける制御装置がオンに構成変更されていないことを確認してください。装置および制御装置がオンに構成変更されているとき、表を削除しようとした場合は、影響を受ける装置で外字を表示または印刷しようと次に試みたとき、サーバーではそれらの装置と同じ制御装置に接続されている装置、および制御装置について、損傷していると報告します。そのような損傷が報告された場合は、次のステップを実行してください。

1. 構成変更 (VRYCFG) コマンドを使用して、影響を受けた装置をオフに構成変更します。
2. 影響を受けた制御装置をオフに構成変更します。
3. 影響を受けた制御装置をオンに構成変更します。
4. 影響を受けた装置をオンに構成変更します。

5. 通常のシステム作業を続けます。

DBCS フォント表用の文字作成ユーティリティの開始:

STRCGU コマンドを使用して、文字作成ユーティリティを開始することができます。

使用されているパラメーターに応じて、CGU メインメニューを呼び出すか、または特定の CGU 機能を指定することができます。詳細については、*適用業務開発ツール (ADTS) ・セット /400: 文字作成ユーティリティ (CGU)* を参照してください。

ユーザー定義の 2 バイト文字のコピー:

STRFMA コマンドを使用して、iSeries の DBCS フォント表とパーソナル・システム/55、5295 ディスプレイ、または InfoWindow[®] 3477 ディスプレイのユーザー・フォント・ファイルとの間でユーザー定義 2 バイト文字をコピーすることができます。

関連情報

STRFMA コマンド

DBCS フォント・ファイル

サーバー提供 DBCS フォント表に加えて、サーバーは DBCS フォント・ファイルも用意しています。これらの DBCS フォント・ファイルは、使用頻度の高い 2 バイト文字が入っている物理ファイルです。

文字作成ユーティリティを使用するときは、これらのファイルの文字を新しいユーザー定義文字の基礎として使用することができます。これらのファイルは変更されないように、読み取り専用権限で提供されます。文字作成ユーティリティまたは IBM Advanced DBCS Printer Support for iSeries ライセンス・プログラムを使用しない場合は、スペース節減のためにこれらのファイルを削除しても構いません。これらのファイルは、すべて QSYS ライブラリーに存在しています。

次の DBCS フォント・ファイルは、DBCS 言語バージョンの OS/400 ライセンス・プログラムとともに配布されます。これらのフォント・ファイルは、CGU および iSeries IBM Advanced DBCS Printer Support for iSeries ライセンス・プログラムで参照されます。

QCGF2424

日本語 DBCS 基本文字イメージのコピーを保管するために使用される日本語 DBCS フォント・ファイル。

QCGF2424K

韓国語 DBCS 基本文字イメージのコピーを保管するために使用される韓国語 DBCS フォント・ファイル。

QCGF2424C

中国語 (繁体字) DBCS 基本文字イメージのコピーを保管するために使用される中国語 (繁体字) DBCS フォント・ファイル。

QCGF2424S

中国語 (簡体字) DBCS 基本文字イメージのコピーを保管するために使用される中国語 (簡体字) DBCS フォント・ファイル。

DBCS ソート表

DBCS ソート表には、サーバーで使用されるすべての 2 バイト文字のソート情報および照合順序が入っています。サーバーはこれらの表を使用して、ソート・ユーティリティを用いた 2 バイト文字のソートを行います。

DBCS ソート表は、保管、復元、および削減することができるオブジェクトです。文字作成ユーティリティを使用すれば、DBCS フォント表のイメージ項目に対応する項目を追加、削除、および変更することもできます。日本語用の場合だけに限られますが、データ・ファイルとの間で DBCS マスター・ソート表の相互コピーもできます。

次の DBCS ソート表は、DBCS 言語バージョンの OS/400 ライセンス・プログラムとともに配布されます。

QCGMSTR

日本語 2 バイト文字セットに関するソート情報を保管するのに使用される日本語 DBCS マスター・ソート表

QCGACTV

日本語 2 バイト文字セットに関する照合順序を保管するのに使用される日本語 DBCS 活動ソート表

QCGMSTRC

中国語 (繁体字) 2 バイト文字セットに関するソート情報を保管するのに使用される中国語 (繁体字) DBCS マスター・ソート表

QCGACTVC

中国語 (繁体字) 2 バイト文字に関する照合順序を保管するのに使用される中国語 (繁体字) DBCS 活動ソート表

QCGACTVK

韓国語漢字を同じ発音のハングル文字にマップするのに使用される韓国語 DBCS 活動ソート表

QCGMSTRS

中国語 (簡体字) 2 バイト文字セットに関するソート情報を保管するのに使用される中国語 (簡体字) DBCS マスター・ソート表

QCGACTVS

中国語 (簡体字) 2 バイト文字セットに関する照合順序を保管するのに使用される中国語 (簡体字) DBCS 活動ソート表

日本語、韓国語、中国語 (簡体字)、および中国語 (繁体字) の 2 バイト文字をソートすることができます。これらの言語にはそれぞれ 2 つの DBCS ソート表、つまり DBCS マスター・ソート表と DBCS 活動ソート表があります。ただし、韓国語の場合は DBCS 活動ソート表があるだけです。DBCS マスター・ソート表には、すべての定義済み DBCS 文字に関するソート情報が入っています。日本語、中国語 (簡体字)、および中国語 (繁体字) の DBCS ソート表は、マスター・ソート表情報から作成され、その言語の 2 バイト文字用の照合順序が入っています。これらの照合順序には、1 バイト英数字セットに関する EBCDIC 照合順序および ASCII 照合順序の場合に類似した目的があります。韓国語文字の場合は、ハングル文字にはそれらの照合順序および DBCS コードの両方が発音に従って割り当てられています。したがって別個の照合順序は必要でなく、さらに韓国語文字のそれぞれは DBCS 活動ソート表 QCGACTVK を使用して、同一発音のハングル文字にマップされています。

どの DBCS ソート表も、そのオブジェクト・タイプは *IGCSRT です。

DBCS ソート表に関するコマンド:

このトピックにリストされているコマンドを使用して、DBCS ソート表を管理および使用することができます。

- オブジェクト検査 (CHKOBJ)
- オブジェクト保管 (SAVOBJ)

- オブジェクト復元 (RSTOBJ)
- DBCS ソート表コピー (CPYIGCSRT) (日本語表専用)
- DBCS ソート表削除 (DLTIGCSRT)
- 文字作成ユーティリティー開始 (STRCGU) (CGU については、**適用業務開発ツール (ADTS) ・セット /400: 文字作成ユーティリティー (CGU)** を参照)

サーバーにおける DBCS ソート表の使用:

ソート表はテープまたはディスクに保管し、システムから削除し、その後サーバーに復元することができます。

日本語 DBCS マスター・ソート表は、System/36 または Application System/Entry (AS/Entry) システムとの間で共用することができるように、データ・ファイルとの間でコピーが可能です。文字作成ユーティリティーを使用してユーザー定義文字を作成するにつれ、各ユーザー定義文字に関するソート情報を追加し、その文字を DBCS 照合順序に追加することもできます。

DBCS ソート表の存在の有無の探索:

オブジェクト検査 (CHKOBJ) コマンドを使用して、DBCS ソート表がサーバーに存在するかどうかを調べることができます。

たとえば、ソート表 QCGMSTR が存在するかどうかを調べるには、次のように入力してください。

```
CHKOBJ OBJ(QSYS/QCGMSTR) OBJTYPE(*IGCSRT)
```

表が存在しない場合は、サーバーはメッセージを出して応答します。表が存在する場合は、サーバーは戻りだけでメッセージは出しません。

初めて 2 バイト文字をソートする場合には、DBCS 活動ソート表の存否を調べてください。文字をソートするには、その DBCS 言語に関する DBCS 活動表が存在しなければなりません。

関連情報

オブジェクト検査 (CHKOBJ) コマンド

DBCS ソート表のテープまたはディスクへの保管:

オブジェクト保管 (SAVOBJ) コマンドを使用して、DBCS ソート表をテープまたはディスクに保管することができます。オブジェクト・タイプに *IGCSRT を指定してください。

DBCS ソート表は SAVSYS コマンドを使用するときに保管されるので、通常のサーバー・バックアップを実行する際に SAVOBJ コマンドを使用する必要はありません。

DBCS ソート表のテープまたはディスクへの保管時点

次の場合には、DBCS ソート表をテープまたはディスクに保管してください。

- その表の削除前。
- 文字作成ユーティリティーの使用による情報の表への追加、更新または変更後。
- 別の iSeries サーバーでの表の使用を計画した時。

関連資料

177 ページの『DBCS ソート表の削除方法』

関連情報

オブジェクト保管 (SAVOBJ) コマンド

DBCS ソート表のテープまたはディスクからの復元:

RSTOBJ コマンドを使用して、DBCS ソート表をテープまたはディスクからサーバーに復元することができます。

テープまたはディスク上の表は、以前に SAVOBJ コマンドを使用して保管されたものでなければなりません。オブジェクト・タイプに *IGCSRT を指定してください。指定された表がサーバーに存在しないと、システムは自動的に DBCS ソート表を再作成します。

これらの表は、その存在をサーバーが認識するためには、QSYS ライブラリーに復元されなければなりません。そのため、RSTOBJ は *IGCSRT オブジェクトを QSYS ライブラリーに対してのみ復元します。また、そのオブジェクトが QSYS ライブラリーに存在していない場合にだけ復元します。

関連情報

RSTOBJ コマンド

日本語 DBCS マスター・ソート表のデータ・ファイルへのコピー:

文字作成ユーティリティーによって、CPYIGCSRT コマンドを使用して、日本語 DBCS マスター・ソート表 (QCGMSTR) をデータ・ファイルへコピーしてください。

このデータ・ファイルは、その後で、System/36 サーバーまたは AS/Entry システムに転送して、その日本語マスター・ソート表を置き換えることができます。

日本語 DBCS マスター・ソート表のデータ・ファイルへのコピー時点:

次の場合には、日本語 DBCS マスター・ソート表を、データ・ファイルにコピーしてください。

- その表を System/36 または AS/Entry で使用するために転送することを計画したとき。日本語 DBCS マスター・ソート表は、必ず日本語 DBCS フォント表とともに移送しなければなりません。
- その表の削除前に、SAVOBJ コマンドの代替として。そうすれば、ファイルを保持することも、ディスクまたはテープに保管することもできます。

日本語 DBCS マスター・ソート表のデータ・ファイルへのコピー方法:

注: この節では、System/36 と表記されている場合にはすべて AS/Entry システムにも適用されます。

日本語 DBCS マスター・ソート表をデータ・ファイルにコピーするには、以下のステップを実行してください。

1. どのデータ・ファイルにコピーするかを決めてください。ファイルは存在していなくてもかまいません。自動的に作成されます。
2. CPYIGCSRT コマンドを次のように入力してください。
 - a. OPTION(*OUT) 値を選択します。
 - b. FILE パラメーターを使用して、マスター表のコピー先にしたいデータ・ファイルの名前を指定してください。 System/36 で使用するためにマスター表をそこへ移送する場合は、ファイル名 #KAMAST を指定するか、マスター・ファイルを System/36 に読み込む時点で、ファイルの名前を変更しなければなりません。ファイルをディスクにコピーするには、iSeries の CPYF コマンドを使用し、ファイルをディスクから System/36 にコピーするには、 System/36 の TRANSFER コマンドを使用してください。

- c. MBR パラメーターを使用して、マスター表のコピー先にしたいデータ・ファイル・メンバーの名前を指定してください。 System/36 で使用するためにマスター表を移送する場合は、 MBR パラメーターに *FILE を指定する必要があります。
3. Enter キーを押します。サーバーでは、ファイルおよびメンバーが存在しない場合はそれらを作成し、存在する場合は既存のメンバーに重ね書きします。
4. ここでこのファイルをSystem/36 に移送して、#KAMAST ファイルに置き換える場合は、 SRTXBLD プロシージャーも使用して、新しいマスター表を反映するように、活動表を更新しなければなりません。

日本語 DBCS マスター・ソート表のデータ・ファイルからのコピー:

CPYIGCSRT コマンドを使用して、データ・ファイルから日本語 DBCS マスター・ソート表 (QCGMSTR) をコピーすることができます。

日本語 DBCS マスター・ソート表のデータ・ファイルからのコピー時点:

次の場合には、日本語 DBCS マスター・ソート表をデータ・ファイルからコピーしてください。

- CPYIGCSRT コマンドの使用を計画するとき。この場合は、AS/Entry マスター・ソート・ファイル (#KAMAST) をSystem/36 から、または AS/Entry を iSeries サーバーにコピーすることができます。CPYIGCSRT コマンドは、 #KAMAST ファイルから iSeries マスター・ソート表 (QCGMSTR) にソート情報をコピーします。コピー操作が完了した後で、#KAMAST ファイルを iSeries サーバーから削除してください。
- マスター表のバージョンをデータ・ファイルにコピーし、そのバージョンを復元したいとき。

注: 日本語 DBCS マスター・ソート表は、必ず日本語 DBCS フォント表と共に移行またはコピーしなければなりません。

日本語 DBCS マスター・ソート表のデータ・ファイルからのコピー方法:

日本語 DBCS マスター・ソート表をデータ・ファイルからコピーするには、以下のステップを実行してください。

1. CPYIGCSRT コマンドを次のように入力してください。
 - a. OPTION(*IN) 値を選択します。
 - b. FILE パラメーターを使用して、移行済みSystem/36 または AS/Entry マスター・ファイル、または CPYIGCSRT コマンドで OPTION(*OUT) を用いて以前にファイルにコピーした iSeries マスター表が入っているデータ・ファイルの名前を指定してください。 System/36 または AS/Entry のマスター・ファイルを移行するには、 System/36 または AS/Entry 上で IFORMAT パラメーターを指定した TRANSFER コマンドを使用して、 #KAMAST マスター・ファイルをディスクに保管します。 iSeries のファイル・コピー (CPYF) コマンドを使用して、ディスクからマスター・ファイル #KAMAST をコピーしてください。ここで説明されているように、CPYIGCSRT コマンドを使用して、データをファイルから iSeries 日本語 DBCS マスター・ソート表にコピーしてください。
 - c. MBR パラメーターを使用して、マスター表データのコピー元にしたいデータ・ファイル・メンバーの名前を指定してください。
2. Enter キーを押します。既存の日本語 DBCS マスター・ソート表の中の情報はオーバーライドされていても、その表が存在していなければ、このコマンドを使用することはできません。
3. 日本語 DBCS 活動表を更新して、新しくコピーした情報を反映するには、 System/36 または AS/Entry 環境の SRTXBLD プロシージャー、または OPTION(5) を指定する STRCGU コマンドを使用してください。これがあらかじめ行われていないと、ソート・ユーティリティーを使用して日本語 2 バイト文字をソートすることはできません。

DBCS ソート表の削除:

DLTIGCSRT コマンドを使用して、サーバーから DBCS ソート表を削除することができます。

DBCS ソート表の削除時点:

使用していない DBCS ソート表を削除して、ディスク・スペースを解放します。しかし、まず SAVOBJ コマンドを使用して表のコピーを必ず保管してください。以下のいずれかの条件が該当する場合、DBCS 言語に関する DBCS マスター・ソート表を削除してください。

1. 文字作成ユーティリティーを使用して、その言語の新しい文字を作成しない。
2. その言語の文字をソートするのにソート・ユーティリティーを使用しない。

DBCS 言語の文字をソートするのにソート・ユーティリティーを使用しない場合は、その言語の DBCS 活動ソート表を削除すべきです。この言語に関してソート・ユーティリティーを使用するためには、DBCS 活動ソート表がサーバー上になければなりません。

DBCS ソート表の削除方法:

表を削除するときは、以下のステップを実行してください。

1. 必要に応じて、表をテープまたはディスケットに保管します。表を削除する前に取り外し可能媒体に保管しない場合は、将来使用できる表のコピーを持たないこととなります。
2. DLTIGCSRT コマンドを入力してください。

たとえば、DBCS ソート表 QCGACTV を削除するには次のように入力してください。

```
DLTIGCSRT IGCSRT(QCGACTV)
```

3. Enter キーを押します。サーバーは表の削除を完了すると、メッセージを出します。

関連資料

174 ページの『DBCS ソート表のテープまたはディスケットへの保管』

オブジェクト保管 (SAVOBJ) コマンドを使用して、DBCS ソート表をテープまたはディスケットに保管することができます。オブジェクト・タイプに *IGCSRT を指定してください。

DBCS 変換辞書

DBCS 変換辞書は、英数字項目およびそれらに関連する DBCS 語を集めたものです。サーバーは、DBCS 変換を実行する時点で、この辞書を参照します。

どの DBCS 変換辞書も、そのオブジェクト・タイプは *IGCDCT です。DBCS 変換には、サーバー提供辞書とユーザー作成辞書が使用されます。

関連資料

187 ページの『DBCS 変換機能の処理方法』

DBCS 変換は、ユーザーとユーザーが読み項目を入力するサーバーの間における対話式機能です。関連する DBCS 文字をサーバーが表示し、使用する語をユーザーが選択します。

サーバー提供の DBCS 変換辞書 (日本語専用):

QSYSIGCDCT はライブラリー QSYS に入っているサーバー提供の辞書であり、英数字で表された日本語読みの項目およびそれらの項目に関連する DBCS 語が集められています。サーバーは、DBCS 変換を実行する際に 2 番目にこの辞書をチェックします。

QSYSIGCDCT には次の項目が入っています。

- 人名
 - 姓
 - 名
- 組織名
 - 上場している私企業
 - 公共企業体
 - 中央政府および地方自治体の代表的な機関
 - 代表的な機関
 - ほとんどすべての総合大学および単科大学
- 住所
 - 各県の行政単位名
 - 11 大都市の町名および街路名
- 企業で広く使用されている部課名および職位名などの業務用語
- 基本 2 バイト文字を含め、IBM の定義による単一 2 バイト文字

この辞書の項目については、追加も削除もできません。ただし、関連する漢字を並べ替えて、DBCS 変換時に最も使用頻度の高い語が最初に表示されるようにすることができます。

関連資料

179 ページの『DBCS 変換辞書の編集』

ユーザー作成の DBCS 変換辞書:

ユーザー作成辞書には、ユーザーが含めたいと思う読み項目および関連 DBCS 語を入れることができます。業務に固有の語、または常時使用するが、サーバー提供辞書には含まれていない語を含むユーザー辞書を作成することもできます。

ユーザーは、1 つまたは複数の DBCS 変換辞書を任意の名前を付けて作成し、任意のライブラリーに保管することができます。しかし、辞書の数がどんなに多くても、それらの辞書にどんな名前が付けられていても、サーバーは DBCS 変換の実行時にユーザーのライブラリー・リストの中で QUSRIGCDCT という名前の最初のユーザー辞書だけを参照します。ライブラリー・リストは必ず適正に指定して、サーバーで正しい辞書をチェックできるようにしてください。

サーバーは DBCS 変換の実行時に、QSYSIGCDCT よりも前に QUSRIGCDCT をチェックします。

DBCS 変換辞書に関するコマンド:

このトピックにリストされているコマンドを使用すれば、DBCS 変換辞書に対してオブジェクト管理機能を実行することができます。

次のコマンドを使用すれば、DBCS 変換辞書に対してオブジェクト管理機能を実行することができます。これらのコマンドを入力するときは、OBJTYPE(*IGCDCT) パラメーターを指定してください。

- CHGOBJOWN: DBCS 変換辞書の所有者を変更します。
- CHKOBJ: DBCS 変換辞書を検査します。
- CRTDUPOBJ: 辞書の重複オブジェクトを作成します。
- DMPOBJ: DBCS 変換辞書をダンプします。
- DMPSYSOBJ: サーバー提供の辞書をダンプします。

- DSPOBJAUT: 辞書に対するユーザーの権限を表示します。
- GRTOBJAUT: 辞書を使用する権限を認可します。
- MOVOBJ: 辞書を別のライブラリーに移動します。
- RNMOBJ: 辞書名を変更します。
- RSTOBJ: 辞書を復元します。
- RVKOBJAUT: 辞書を使用する権限を取り消します。
- SAVOBJ: 辞書を保管します。
- SAVCHGOBJ: 変更された辞書を保管します。

次のコマンドを使用すると、サーバーは DBCS 変換辞書を保管または復元します。

- RSTLIB: 辞書が保管されているライブラリーを復元します。
- SAVLIB: 辞書が保管されているライブラリーを保管します。
- SAVSYS: サーバーを保管するときに QSYSIGCDCT (サーバー DBCS 変換辞書) を保管します。

次のコマンドを使用して、辞書を作成、編集、表示、および削除することができます。

- CRTIGCDCT: DBCS 変換辞書を作成します。
- EDTIGCDCT: DBCS 変換辞書を編集します。
- DSPIGCDCT: DBCS 変換辞書を表示します。
- DLTIGCDCT: DBCS 変換辞書を削除します。

DBCS 変換辞書の作成:

DBCS 変換辞書を作成するには、以下のステップを実行してください。

1. DBCS 変換辞書作成 (CRTIGCDCT) コマンドを使用してください。
2. 辞書に QUSRIGCDCT と命名して、その辞書が DBCS 変換の過程で使用できるようにしてください。サーバーはユーザーのライブラリー・リストを探索して、その辞書が最初のユーザー作成辞書であることが判明するとその辞書を使用します。

辞書の作成中は、アプリケーション・プログラムによって変換に使用されるのを防ぐために、辞書を別の名前にすることができます。そのような場合は、後でオブジェクト名変更 (RNMOBJ) コマンドを使用して、辞書名を変更してください。

たとえば、ユーザー DBCS 変換辞書を作成して、ライブラリー DBCSLIB に保管するには次のように入力してください。

```
CRTIGCDCT IGCDCT(DBCSLIB/QUSRIGCDCT)
```

3. EDTIGCDCT コマンドを使用して、作成後の辞書に項目および関連語を入れてください。辞書に項目を入れる方法については、『DBCS 変換辞書の編集』を参照してください。

DBCS 変換辞書の編集:

DBCS 変換辞書編集 (EDTIGCDCT) コマンドを使用して、DBCS 変換辞書を編集してください。編集を使用してユーザー定義文字を辞書に追加し、ユーザーが DBCS 変換を使用して文字を入力し、DBCS 変換辞書の中の用語を個別の要求に合わせて並べ替えられるようにしてください。

関連資料

177 ページの『サーバー提供の DBCS 変換辞書 (日本語専用)』

QSYSIGCDCT はライブラリー QSYS に入っているサーバー提供の辞書であり、英数字で表された日

本語読みの項目およびそれらの項目に関連する DBCS 語が集められています。 サーバーは、DBCS 変換を実行する際に 2 番目にこの辞書をチェックします。

DBCS 変換辞書の要件:

DBCS 変換辞書の編集集中に使用する必要のあるディスプレイ装置は、EDTIGCDCT コマンドで ENTRY パラメーターとして指定した値によって異なります。

- ENTRY パラメーターに特定のストリングを指定した場合、または 2 バイト文字を表示したい場合は、DBCS ディスプレイ装置を使用しなければなりません。
- ENTRY パラメーターに特定のストリングを指定しなかった場合、または 2 バイト文字を表示しない場合は、DBCS ディスプレイ装置か 24 行 x 80 桁の英数字ディスプレイ装置のいずれかを使用してください。

関連情報

EDTIGCDCT コマンド

制御言語 (CL)

DBCS 変換辞書の操作:

ユーザー作成 DBCS 変換辞書に対して、このトピックにリストされている編集操作を行うことができます。

- 辞書に項目を追加する (作成後の辞書に対する最初の項目追加も含まれます)。辞書には最大 99,999 項目を入れることができます。
- 辞書から項目を削除する。
- 読み項目に関連する DBCS 語の置き換えなど、辞書の中の項目を変更する。
- 読み項目に関連する DBCS 語を移動して、DBCS 変換時に現れる順序を並べ替える。

サーバー提供辞書 QSYSIGCDCT に対して実行することのできる唯一の編集機能は、読み項目に対する漢字の移動です。語を移動して、DBCS 変換時に現れる順序を並べ替えてください。

DBCS 変換辞書の編集に使用される画面:

EDTIGCDCT コマンドを入力すると、サーバーではそのコマンドで ENTRY パラメーターとして指定された値に応じて、DBCS 変換辞書の処理画面か関連語句の編集画面かどちらかを表示します。

関連情報

EDTIGCDCT コマンド

制御言語 (CL)

DBCS 変換辞書の処理画面:

このトピックの図を使用して、編集する項目の選択、項目の削除など、読み項目を処理することができます。EDTIGCDCT コマンドの ENTRY パラメーターとして *ALL または総称ストリングを入力した場合は、サーバーは「DBCS 変換辞書の処理」画面を表示します。

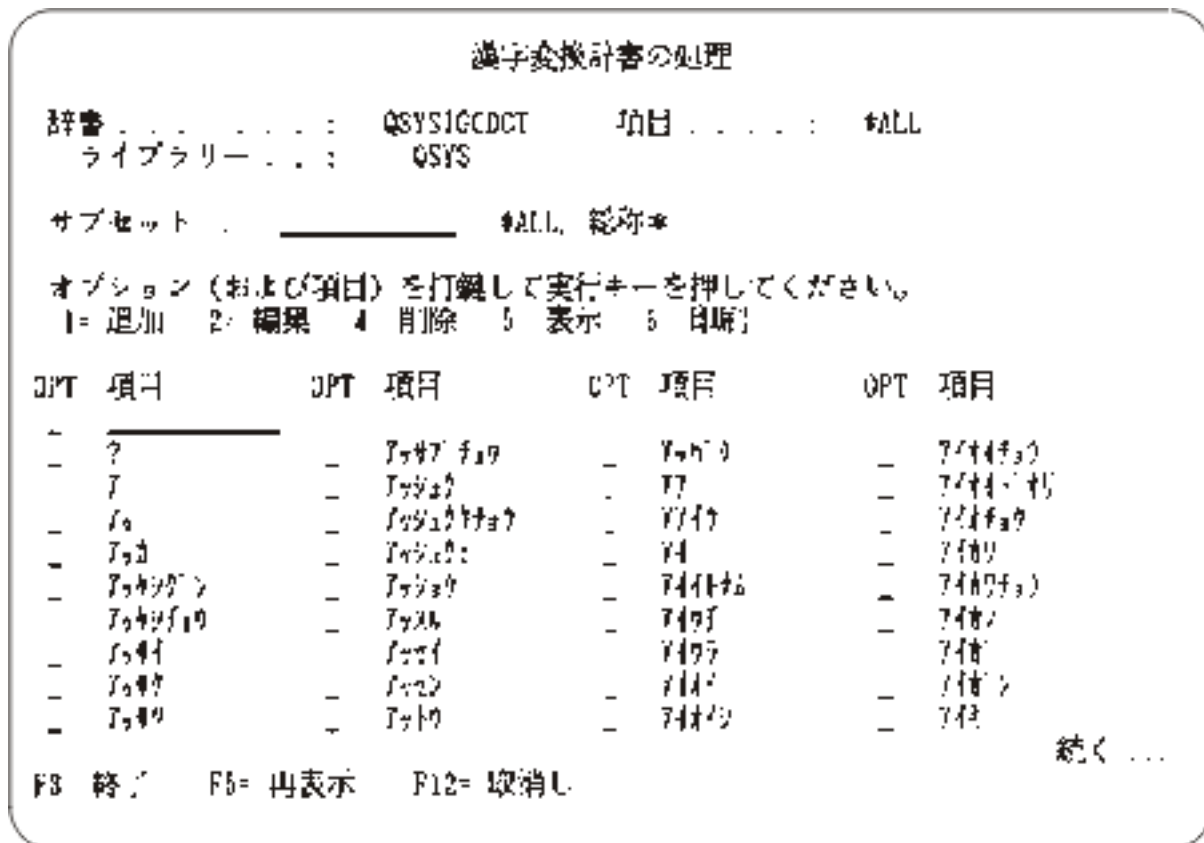


図 23. DBCS 変換辞書の処理画面

関連情報

EDTIGCDCT コマンド

制御言語 (CL)

DBCS 変換辞書の関連語句の編集画面:

この画面を使用して、読み項目に関連する DBCS 語を処理できます。ある特定のストリングを ENTRY パラメーターとして入力した場合は、サーバーで関連語句編集画面を表示します。また、「DBCS 変換辞書の処理」画面から編集対象項目を選択した場合も、サーバーは「関連語句の編集」画面を表示します。

183 ページの図 24 は「関連語句の編集」画面の例です。

関連情報

EDTIGCDCT コマンド

制御言語 (CL)

DBCS 変換辞書の編集操作の例:

この節では、EDTIGCDCT 画面を使用して実行できる編集操作の例を示します。

DBCS 変換辞書の編集の開始:

編集操作のタイプがどれであれ、EDTIGCDCT コマンドを入力して、辞書の編集を開始してください。たとえば、辞書に最初の項目を入れるには次のように入力してください。

```
EDTIGCDCT IGCDCT(DBCSLIB/QUSRIGCDCT) +  
ENTRY(*ALL)
```

また、ストリング ABC で始まる項目を編集するには次のように入力してください。

```
EDTIGCDCT IGCDCT(DBCSLIB/QUSRIGCDCT) +  
ENTRY('ABC*')
```

DBCS 変換辞書への最初の項目の追加:

最初の項目を辞書の中に追加するには、以下のステップを実行してください。

1. EDTIGCDCT コマンドを入力する時、ENTRY(*ALL) を指定してください。たとえば、ライブラリー DBCSLIB の中に保管されている辞書 QUSRIGCDCT を編集するには、次のように入力してください。

```
EDTIGCDCT IGCDCT(DBCSLIB/QUSRIGCDCT) +  
ENTRY(*ALL)
```

サーバーでは、「DBCS 変換辞書の処理」画面を表示します。

2. このリストの最初のオプション・フィールドに 1 を入力し、辞書に追加したい英数カナ項目を入力フィールドに入力してください。

「関連語句の編集」画面に「データの始め」と「データの終り」という 2 つのデータ行が表示されま

す。

3. 「データの始め」 行の横の *NBR* フィールドに 1 を入力して、1 行挿入してください。
4. Enter キーを押します。サーバーでは空白行を 1 行表示します。

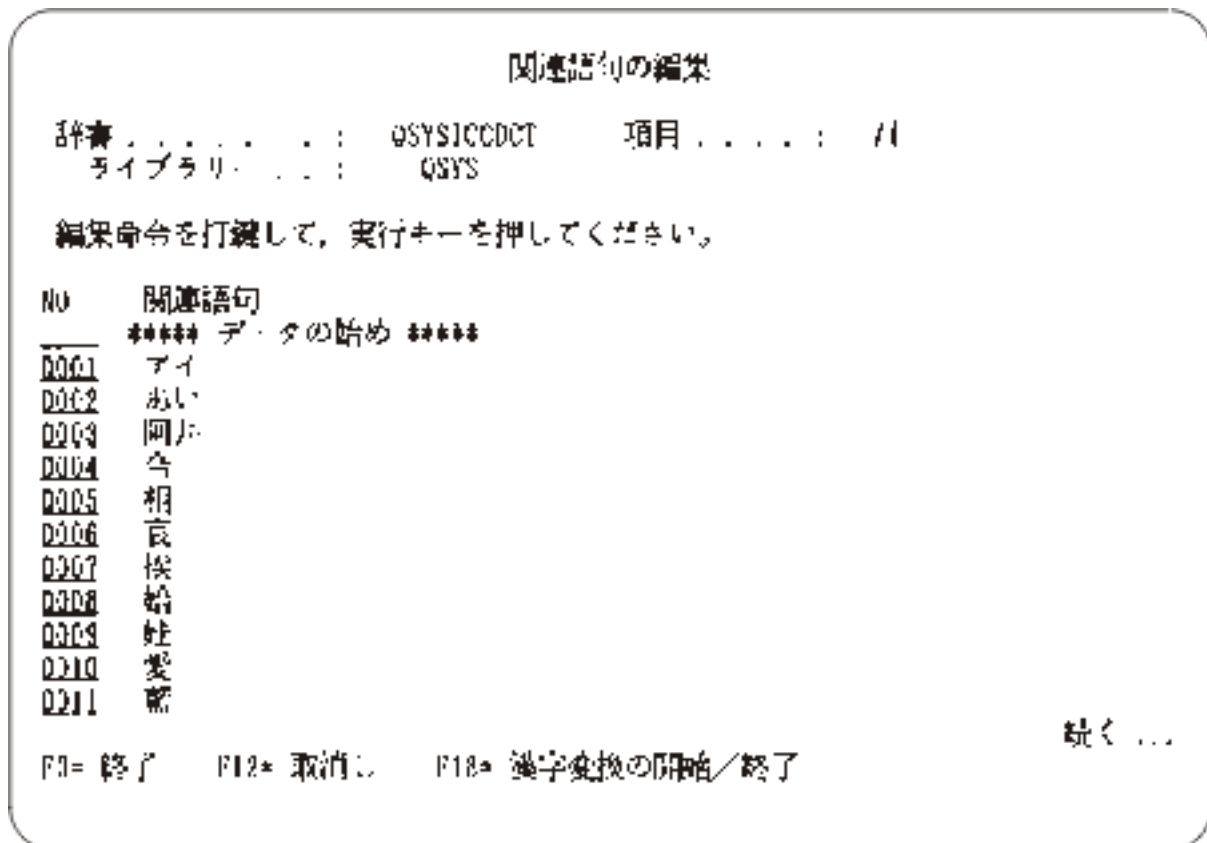


図 24. 「関連語句の編集」画面

5. 上記のブランク行に、新しい読み項目に関連付けられる DBCS 語を入力してください。

挿入された行にデータを入力し、カーソルをその行に置いたままにして、Enter キーを押すと新しい行がもう 1 行下に現れます。この行に別の DBCS 語を入力することもできれば、その行をブランクのままにして Enter キーを押すことによって、削除することもできます。

6. この最初の項目の入力を終える時は、F12 を押して「辞書入力終了」画面を表示してください。Y オプションを選択してその項目を保管してから、「DBCS 変換辞書の処理」画面に戻ってください。オプション 1 を再び入力し、入力フィールドに別の英数カナ項目を入力して項目の辞書への追加を続けるか、または F3 を押して辞書の編集を終了してください。

DBCS 変換辞書の関連語句の移動:

読み項目に関連する語を移動すると、DBCS 変換時に語が現れる順序が変更されます。語を移動するには、以下のステップを実行してください。

1. EDTIGCDCT コマンドで特定の項目を入力するか、「DBCS 変換辞書の処理」画面から編集対象項目を選択して、DBCS 語を移動したい項目の「関連語句の編集」画面を表示してください。
2. 画面が表示されたら、移動したい DBCS 語の横の NBR フィールドに M と入力してください。
3. 語を移動したい行のその前の行の NBR フィールドに A を入力してください。
4. Enter キーを押します。サーバーは、M の印のついた行の語を、A の印のついた行の直後の行に移動します。

DBCS 変換辞書での項目の削除:

図 25 の画面で削除したい項目の横の入力フィールドに、 4 を入力してください。

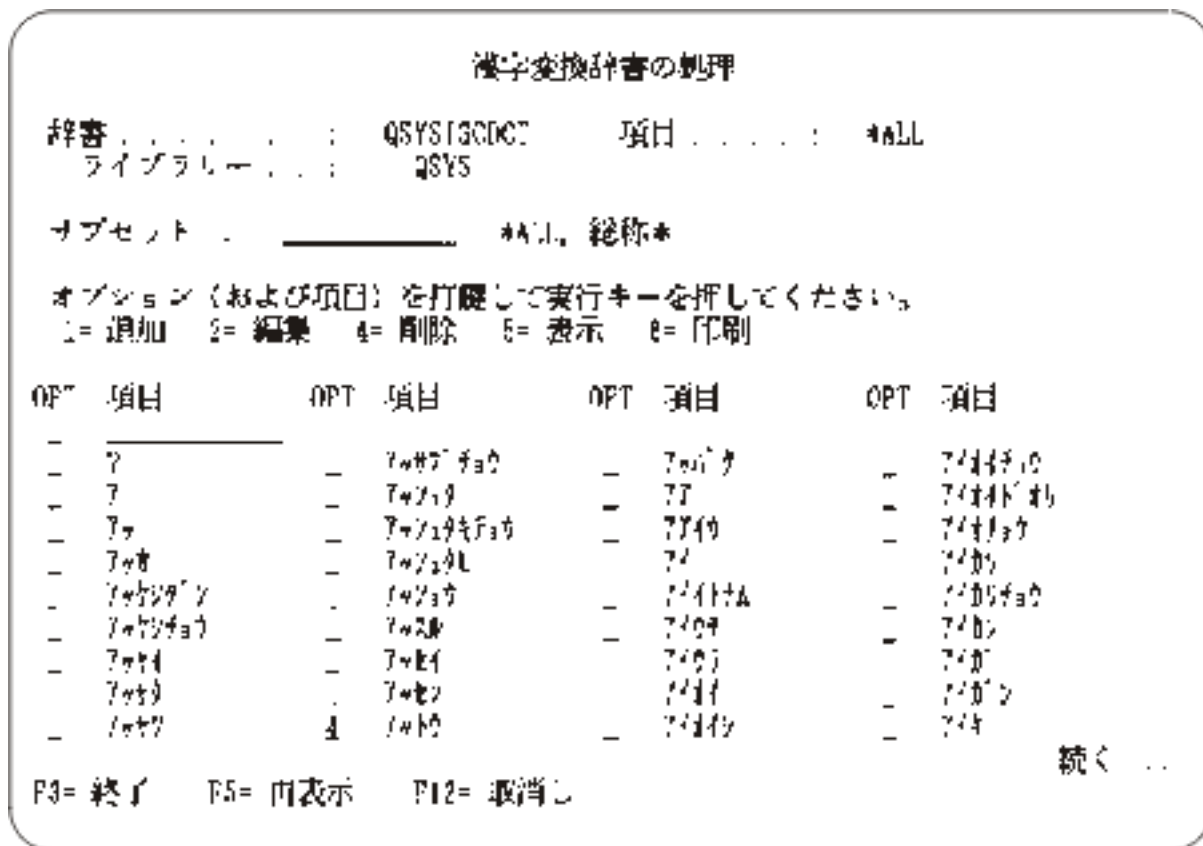
DBCS 変換辞書の編集処理の終了:

編集操作を終了するには、F3 を押してください。「辞書入力終了」画面が表示されますから、項目を保管するかどうかの選択ができます。そこでサーバーは、「コマンド入力」画面などの基本作業画面に戻ります。

DBCS 変換辞書の編集に関する提案:

DBCS 変換辞書を編集するときには、いくつかの考慮事項があります。

- 「関連語句の編集」画面で DBCS 変換を用いれば、関連語をユーザー作成辞書に入力することができます。この手順については、186 ページの『DBCS 変換 (日本語専用)』を参照してください。



たとえば、辞書 QUSRIGCDCT の項目 ABC およびその関連語を印刷するには、次のように入力してください。

```
DSPIGCDCT IGCDCT(DBCSLIB/QUSRIGCDCT) +  
ENTRY(ABC) OUTPUT(*PRINT)
```

サーバー提供辞書 QSYSIGCDCT の項目のすべておよび関連語を表示するには、次のように入力してください。

```
DSPIGCDCT IGCDCT(QSYS/QSYSIGCDCT)
```

186 ページの図 26 は、DSPIGCDCT コマンドによって作成される画面の例です。読み項目および関連語が示されています。

関連情報

制御言語 (CL)

DBCS 変換辞書表示 (DSPIGCDCT) コマンド

DBCS 変換辞書の削除:

DBCS 変換辞書削除 (DLTIGCDCT) コマンドを使用して、DBCS 変換辞書をサーバーから削除することができます。

辞書を削除するためには、辞書に対するオブジェクト存在権限、および辞書が保管されているライブラリーに対するオブジェクト操作権を持っていないければなりません。

辞書を削除する時は、正しいライブラリー名を間違いなく指定してください。大勢のユーザーが、それぞれ QUSRIGCDCT と命名された辞書をライブラリーの中に持っている可能性があります。ライブラリー名を指定しないと、サーバーはライブラリー・リストの中の最初の DBCS 変換辞書を削除します。

たとえば、ライブラリー DBCSLIB の中の DBCS 変換辞書を削除するには、次のように入力してください。

```
DLTIGCDCT IGCDCT(DBCSLIB/QUSRIGCDCT)
```

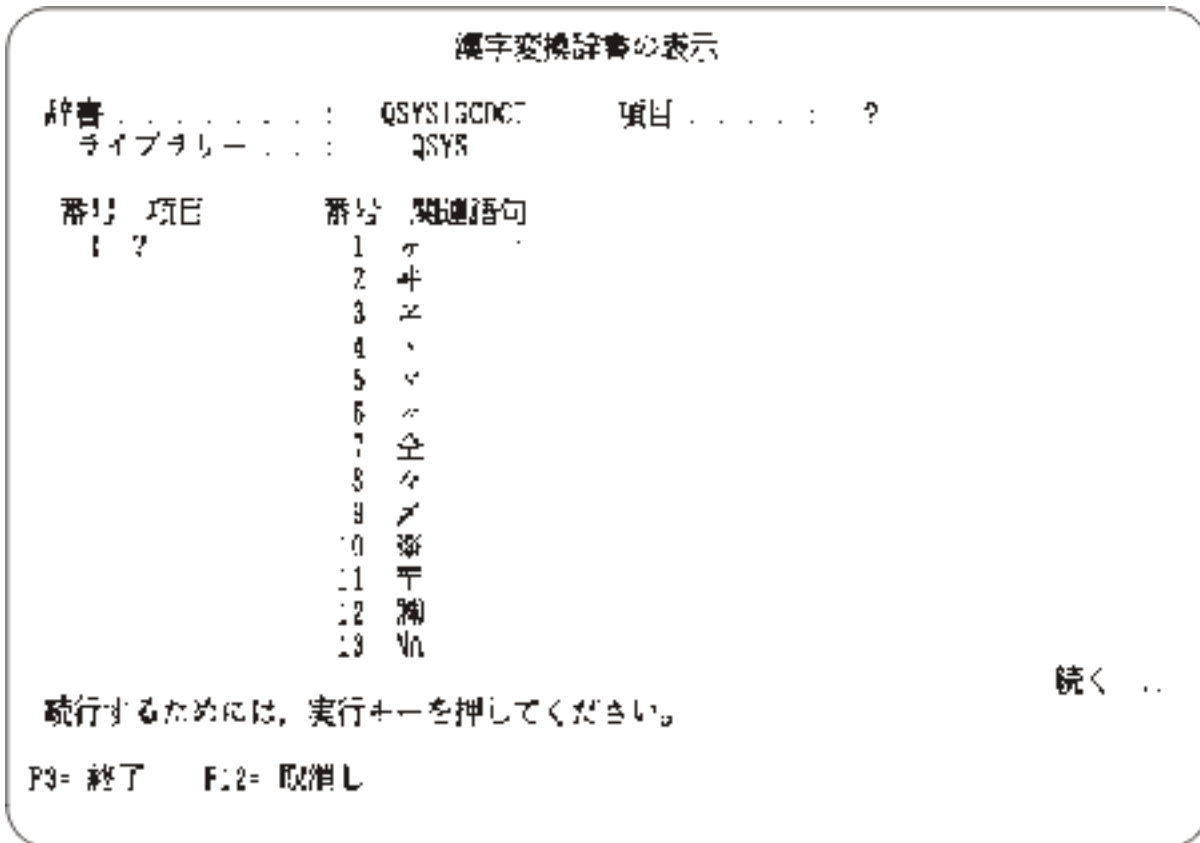


図 26. DSIPIGCDCT コマンドによって作成される画面

関連情報

制御言語 (CL)

DBCS 変換辞書削除 (DLTIGCDCT) コマンド

DBCS 変換 (日本語専用)

DBCS ディスプレイ装置を使用して 2 バイト文字データを入力する場合は、そのディスプレイ装置でサポートされている各種のデータ入力手法を使用できます。あるいは、iSeries の DBCS 変換サポートの使用を選択することもできます。


DBCS 変換を用いると、読み項目または DBCS コードを入力して、その項目またはコードを関連 DBCS 語に変換することになります。 DBCS 変換は日本語文字セット用として意図されており、他の 2 バイト文字セットには適用されません。

特に DBCS 変換を用いると、以下のデータを変換できます。

- 読みストリングから漢字への変換
- 英語の英数字から 2 バイト英数字への変換
- 英数字カタカナから 2 バイト文字のひらがな、およびカタカナへの変換
- DBCS コード から対応する 2 バイト文字への変換
- DBCS 番号 から対応する 2 バイト文字への変換

DBCS 変換を使用できる場合:

このトピックでは、DBCS 変換を使用できる状況をリストします。

- 特定の SEU 画面の入力フィールドにデータを入力するとき。DBCS 変換を使用できるフィールドについては、V5R1 補足マニュアル Web サイト  の *ADTS for AS/400®: 原始ステートメント入力ユーティリティー 資料* (英語) を参照してください。
- QCMDEXEC を使用して 2 バイト・データを要求するプロンプトを出すとき。この手順に関する説明については、制御言語 (CL) を参照してください。
- ユーザー作成アプリケーション・プログラムの中の DBCS ディスプレイ・ファイルの入力フィールドにデータを入力するとき。DDS キーワード IGCCNV を用いて、DBCS 変換を指定してください。このキーワードについては、DDS concepts を参照してください。
- DBCS 変換辞書の編集 (EDTIGCDCT コマンド) 時に表示される「関連語句の編集」画面上で関連語句を編集するとき。

関連資料

179 ページの『DBCS 変換辞書の編集』

DBCS 変換機能の処理方法:

DBCS 変換は、ユーザーとユーザーが読み項目を入力するサーバーの間における対話式機能です。関連する DBCS 文字をサーバーが表示し、使用する語をユーザーが選択します。

サーバーでは、DBCS 変換辞書をチェックすることによって、ある読み項目に関連する語を判別します。サーバーは DBCS 変換の実行時に、2 つの DBCS 変換辞書をチェックします。まず最初に、ユーザーのライブラリー・リストを探索して見つけた最初のユーザー作成辞書 QUSRIGCDCT をチェックします。次に、ライブラリー QSYS に保管されているシステム提供辞書 QSYSIGCDCT をチェックします。(QSYSIGCDCT には、日本語 2 バイト文字だけが入っています。) ユーザーは他にもユーザー辞書を作成し、それらの辞書に QUSRIGCDCT 以外の名前を付けることができますが、DBCS 変換を実行する時にサーバーが参照するのは、ユーザーのライブラリー・リストにある QUSRIGCDCT という名前の最初のユーザー作成辞書に限られます。

サーバーは辞書をチェックした上で、その読み項目に関連する語を表示します。そこで、ユーザーは自分が選択した語の下にカーソルを置いて、Enter キーを押します。サーバーはユーザーが DBCS 変換を開始した時点でカーソルが位置していた語を入力します。

関連資料

177 ページの『DBCS 変換辞書』

DBCS 変換辞書は、英数字項目およびそれらに関連する DBCS 語を集めたものです。サーバーは、DBCS 変換を実行する時点で、この辞書を参照します。

DBCS 変換の使用:

DBCS 変換用の辞書をカスタマイズすることができます。

DBCS 変換時に使用されるユーザー定義辞書を変更することができます。ユーザー定義辞書を変更するにあたっては、その前にアプリケーション・プログラムを打ち切るか、またはサーバーが実行しているコマンドを打ち切ってください。その上で、(CHGLIBL コマンドを用いて) ライブラリー・リストを変更することによって、使用されている辞書を変更してください。

DBCS 変換用として、ユーザー自身の DBCS 変換辞書を作成することができます。サーバー提供辞書は、英数字で表された日本語の読み付き項目、および項目に関連する日本語の漢字を集めたものです。この手順の説明については、179 ページの『DBCS 変換辞書の作成』を参照してください。

ユーザー作成辞書が見つからない場合、サーバーは QSYSIGCDCT だけを参照します。DBCS 変換辞書の作成および使用の詳細については、177 ページの『DBCS 変換辞書』を参照してください。

DBCS 変換の実行:

このトピックでは、DBCS 変換を使用して 1 つの読み項目に関連する DBCS 文字に変換する方法について説明します。2 バイト文字データを入力したい各フィールドごとに、それぞれ別個に DBCS 変換を開始しなければなりません。

注: DBCS 変換は、日本語データ項目のためのものです。他の国語での使用には制限があります。

DBCS 変換を実行する時は、Help キーを押すことによってその機能に関する情報を表示することができます。ヘルプは、DBCS 変換が終了するまで使用できます。

1. 2 バイト文字を入力したいフィールドにカーソルを置いてください。シフト制御文字がまだ挿入されていない場合は、そのフィールドに挿入してください。シフト制御文字の挿入方法については、164 ページの『シフト制御 2 バイト文字の挿入』を参照してください。
2. カーソルをシフトイン文字の下、シフト制御文字間のブランク、または 2 バイト文字の下に位置付けてください。
3. DBCS 変換の開始に使用される機能キーを押してください。

SEU では、「関連語句の編集」画面 (EDTIGCDCT コマンドを使用する時に表示されます) からと同じように、F18 を押してください。サーバーが次のようなプロンプト行を表示します。

$\bar{A} \quad \text{-----} \quad \bar{B} \quad \text{-----} \quad \bar{C}$

4. 次の値を入力してください。
 - a. A のフィールドには、次のいずれかの値を入力してください。
 - I ステップ 2 で下にカーソルを位置付けた文字の前に、変換した語を挿入します。
 - R ステップ 2 で下にカーソルを位置付けた文字を、変換した項目で置き換えます。
 - b. B のフィールドには、次のいずれかの値を入力してください。
 - 1) 変換される読みのストリング。読みストリングには、最大 12 文字まで入れることができます。
 - 2) 2 バイト文字の 4 文字 DBCS コード。
 - 3) 2 バイト文字の 2 から 5 桁の DBCS 番号。
 - c. C のフィールドには、次の変換コードの内の 1 つを入力してください。

項目なし

DBCS 変換辞書を参照して、フィールド B の項目を読みから 2 バイト文字に変換します。

- G フィールド B の 2 桁から 5 桁の漢字番号を、それが表す文字に変換します。
- H フィールド B の項目を 2 バイトのひらがな、上段シフトの英字、数字、または特殊文字に変換します。
- K フィールド B の項目を 2 バイトのひらがな、下段シフトの英字、数字、または特殊文字に変換します。
- X 4 文字の DBCS コードを、それが表す文字に変換します。

5. Enter キーを押します。サーバーが次のようなプロンプト行を表示します。

$\bar{A} \quad \text{-----} \quad \bar{B} \quad \text{-----} \quad \bar{C} \quad \text{-----} \quad \bar{D} \quad \text{-----} \quad ^+$

6. サーバーが D のフィールドにフィールド B の項目に関連する語を表示します。

表示されている最後の語の後にプラス (+) 符号が表示される場合は、サーバーではまだ表示する語が続きます。次ページ・キーを押して、追加の項目を表示してください。その後で前に表示された語に戻るには、前ページ・キーを押してください。

語が反転イメージで表示された場合、その語の中に組み込みブランクが入っています。

7. 必要とする DBCS 文字の下にカーソルを置いて、その語を選択してください。
8. Enter キーを押します。フィールド A に入力したのがどちらであるかに応じて、サーバーは、ステップ 2 でカーソルを位置付けた語の前に挿入するか、その語を置き換えるかのどちらかの方法で語を入力します。
9. 次のいずれかの操作を行ってください。
 - a. DBCS 変換の使用を続けます。フィールドへのデータ入力が終わるまで、ステップ 4 (188 ページ) からステップ 8 を繰り返します。
 - b. 変換を開始するのに使用したのと同じ機能キーを押して、DBCS 変換を終了します。フィールドの終わりに達すると、サーバーが自動的に変換を終了します。

SEU では、「関連語句の編集」画面 (EDTIGCDCT コマンドを使用する時に表示されます) からと同じように、F18 を押してください。

注: DBCS 変換が終了するまで、他のサーバー機能は一切実行できません。たとえば、F3 キーを使用して SEU 画面を終了することはできません。

DBCS 変換の例:

この節では、DBCS 変換の例を示します。

1 つの英数カナ項目から 2 バイト項目への変換:

この例は、1 つの項目を変換してフィールドに入力する方法を示しています。

1. 2 バイト文字データを入力したいと思うフィールドに、カーソルを置いてください (190 ページの図 27 を参照)。
2. シフト制御文字をフィールドに挿入してください。シフト制御文字を挿入する方法については、164 ページの『シフト制御 2 バイト文字の挿入』を参照してください。
3. DBCS 変換の開始に使用される機能キーを押してください。この例の画面の場合は、該当する機能キーは F18 です。190 ページの図 28 のように、サーバーはプロンプト行を表示します。

変換の開始時に、カーソルはシフトイン文字の下に置かれていたため、変換は自動的に I (変換済みの語を挿入) に設定されます。

4. 変換対象の読み項目を 2 番目のフィールドに入力してください。

3 番目のフィールドはブランクにしておいてください。191 ページの図 29 の画面例を参照してください。

5. Enter キーを押します。サーバーが関連する DBCS 文字を表示します。
6. 入力したい漢字が最初に表示されている漢字でない場合は、その下にカーソルを置いてください。191 ページの図 30 の画面例では、最初に表示されている漢字が入力したい漢字になっています。
7. Enter キーを押します。192 ページの図 31 に示すように、漢字がフィールドに入力されます。

カーソルをここに置きます。

日付 : 91/05/23 人 事 情 報 保 守 プログラム名 : EMPMAINT
画面名 : EMPMAINTE

社員番号 : 12002 氏名 _____ 性別 ____ 年齢 ____
フリガナ _____

現住所 _____
都道府県名 _____ 市町村名 _____

本籍地 _____
都道府県名 _____ 市町村名 _____

職位コード _ 職位名称 _____

部課コード _ 部課名称 _____

給与 _____ 趣味 _____

F3 : 終了 F18: カナ漢字変換

HRSL321-0

図 27. 画面例 1

シフト制御文字がこのフィールドに
挿入されたことがわかります。

日付 : 91/05/23 人 事 情 報 保 守 プログラム名 : EMPMAINT
画面名 : EMPMAINTE

社員番号 : 12002 氏名 ^{PP} _____ 性別 ____ 年齢 ____
フリガナ _____

現住所 _____
都道府県名 _____ 市町村名 _____

本籍地 _____
都道府県名 _____ 市町村名 _____

職位コード _ 職位名称 _____

部課コード _ 部課名称 _____

給与 _____ 趣味 _____

F3 : 終了 F18: カナ漢字変換

↓ _____

プロンプト行。

HRSL322-0

図 28. 画面例 2

日付 : 91/05/23 人 事 情 報 保 守 プログラム名 : EMPMAINT
画面名 : EMPMAINTE

社員番号 : 12002 氏名 _____ 性別 ____ 年齢 ____
フリガナ _____

現住所 _____
都道府県名 _____ 市町村名 _____

本籍地 _____
都道府県名 _____ 市町村名 _____

職位コード _ 職位名称 _____

部課コード _ 部課名称 _____

給与 _____ 趣味 _____

F3 : 終了 F18: カナ漢字変換

↓ 774 _____

英数字エントリーをここに入力します。

HRSL323-0

図 29. 画面例 3

日付 : 91/05/23 人 事 情 報 保 守 プログラム名 : EMPMAINT
画面名 : EMPMAINTE

社員番号 : 12002 氏名 _____ 性別 ____ 年齢 ____
フリガナ _____

現住所 _____
都道府県名 _____ 市町村名 _____

本籍地 _____
都道府県名 _____ 市町村名 _____

職位コード _ 職位名称 _____

部課コード _ 部課名称 _____

給与 _____ 趣味 _____

F3 : 終了 F18: カナ漢字変換

↓ 774 _____ 新井 荒井 新居 荒居 荒粗

カーソルをここに置きます。

HRSL324-0

図 30. 画面例 4

システムが語をフィールドに入力します。

日付 : 91/05/23	人事情報保守	プログラム名 : EMPMAINT	画面名 : EMPMAINTE
社員番号 : _____	氏名 <small>フリガナ</small> <u>新井</u>	性別 _____	年齢 _____
現住所	_____	_____	_____
都道府県名 _____	市町村名 _____	_____	_____
本籍地	_____	_____	_____
都道府県名 _____	市町村名 _____	_____	_____
職位コード _____	職位名称 _____	_____	_____
部課コード _____	部課名称 _____	_____	_____
給与 _____	趣味 _____	_____	_____
		F3 : 終了	F18 : カナ漢字変換
↑ <u>1074</u>			

HRSL325-0

図 31. 画面例 5

多数の英数カナ項目から DBCS への同時変換:

この例は、多数の英数カナ項目を DBCS に同時に変換する方法を示しています。

DBCS 変換を各英数字ごとに連続的に開始する必要はありません。その代わりに、次のステップを実行することができます。

1. フィールド B に収まる限り、英数カナ項目をいくつでも入力してください。各項目は空白で区切ります。フィールド B には、英数字 12 文字分のスペースがあります。

これらが変換対象の項目です。

I	XXX	YYY	ZZZ	_____	
A	B	C			D

サーバーは項目を入力した順序で、一度に 1 項目ずつ変換します。サーバーは項目を 1 つ変換すると、その項目に関連する漢字をフィールド D に表示します。

2. 使用したい漢字の下にカーソルを置いてください。
3. Enter キーを押します。するとサーバーがフィールド B を調整し、次の項目がフィールドの左端位置に移動します。その項目に関連する漢字がフィールド D に表示されます。

この時点で、フィールド B の終わりに、追加の変換対象読み項目を入力することができます。

英数字空白から DBCS 空白への変換:

この例は、DBCS 変換を使用して英数字空白 (1 桁幅) を DBCS 空白 (2 桁幅、つまり 2 バイト文字相当幅) に変換する方法を示しています。

空白を変換するには、以下のステップを実行してください。

1. フィールド B に 1 つまたは複数の空白を入力してください。

\bar{A} \bar{B} \bar{C} D

2. Enter キーを押します。サーバーでは、ユーザーがフィールド B に入力した英数字ブランクと同数の DBCS ブランクを、フィールド D に表示します。DBCS ブランクは反転表示されます。
3. Enter キーをもう一度押してください。サーバーは、ユーザーが DBCS 変換を開始したフィールドに DBCS ブランクを入力します。

英数カナ項目または変換コードから DBCS への変更:

この例は、英数カナ項目または変換コードを DBCS に変更する方法を示しています。

変換の過程で表示された関連語の中に読み項目に対する適切な置き換え語がなく、(別のタイプの変換または別の読み項目を用いることによって) 変換をあらためて試みたい場合は、以下のステップを実行してください。

1. カーソルをフィールド B に移動してください。たとえば、次のようにします。

ここにカーソルを移動します。
|
XXXXXX

\bar{A} \bar{B} \bar{C} D

2. 以下のステップのいずれか 1 つを実行してください。
 - a. 英数カナ項目を入力したいフィールドの最初の桁の下に、カーソルを置いてください。
 - b. 別の読み項目を入力してください。
 - c. フィールド C の変換コードを、たとえば H から K へというように変更してください。
3. Enter キーを押します。
4. DBCS 変換を続けてください。

DBCS 変換の使用による DBCS 変換辞書への語句入力:

この例は、「関連語句の編集」画面で DBCS 文字を入力するとき、どのように DBCS 変換を使用できるかを示しています。

DBCS 変換を開始するには、以下のステップを実行してください。

1. DBCS 文字を入力したい桁にカーソルを位置付けてください。
2. F18 を押してください。サーバーが、画面の最下部に変換プロンプト行を表示します。

188 ページの『DBCS 変換の実行』で説明されている指示事項にしたがって、DBCS 変換を実行してください。

注: DBCS 変換の開始および終了は、各行のデータごとに別々に行わなければなりません。

DBCS 変換の使用に関する考慮事項:

DBCS 変換の使用に関しては、いくつかの考慮事項があります。

- DBCS 変換を実行することができるのは、DBCS ディスプレイ装置で 5556 キーボードを使用する場合に限られます。
- DBCS 変換を使用して文字を挿入または置き換えることができるのは、2 バイト文字の挿入先の行に十分なスペースがある場合だけです。

- 文字を挿入するために使用可能なスペースは、その行のブランクでない最後の文字から、画面の右端までの文字数に相当します。
- 文字を置き換えるために使用可能なスペースは、カーソル位置 (カーソルの付いている文字を含む) から、そのフィールドの DBCS 部分の終わりまでの文字数に相当します。

十分なスペースがないと、次のような結果が発生します。

- 使用可能なスペースがないところで、ストリングの挿入または置き換えを試みた場合は、サーバーからメッセージが送られます。
- メッセージを無視して Enter キーをもう一度押した場合は、サーバーは挿入または置き換えストリングの右側から、超過分の文字を切り捨てます。

フィードバック域のレイアウト

ここで示す表は、オープンされたファイルに関連するオープン・フィードバック域および入出力フィードバック域の説明です。

次の情報が、これらのフィードバック域の中の各項目について示されています。

- オフセット。フィードバック域の開始から各項目の記憶位置までのバイト数です。
- データ・タイプ
- 長さ。長さはバイト数で与えられます。
- 内容。項目の説明およびその有効値です。
- ファイル・タイプ。各項目で有効なファイル・タイプです。

ユーザーが使用している高水準言語で提供されているサポートによって、この情報にアクセスする方法、およびデータ・タイプの表し方が決まります。詳細については、ご使用の高水準言語の資料を参照してください。

オープン・フィードバック域

オープン・フィードバック域は、オープン・データ・パス (ODP) の一部で、ファイルがオープンされた後にファイルに関する一般情報が入ります。そこには、ファイルに対して定義されている装置または通信セッションに関する情報の他に、ファイル・タイプによっては、ファイル固有の情報も入ります。この情報はオープン処理の過程で設定され、他の操作の実行に伴って更新されることがあります。

表 22. オープン・フィードバック域

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
0	文字	2	<p>オープン・データ・パス (ODP) タイプ:</p> <p>DS スプールされていないディスプレイ、テープ装置、ICF、保管、プリンター・ファイル、またはスプールされていないディスケット・ファイル</p> <p>DB データベース・メンバー</p> <p>SP スプールされているプリンターまたはディスケット・ファイル、またはインライン・データ・ファイル</p>	全タイプ
2	文字	10	<p>オープンされているファイルの名前。 ODP タイプが DS の場合は、これは装置ファイルまたは保管ファイルの名前です。 ODP タイプが SP の場合は、これは装置ファイルまたはインライン・データ・ファイルの名前です。 ODP タイプが DB の場合は、これはメンバーが属しているデータベース・ファイルの名前です。</p>	全タイプ

表 22. オープン・フィールドバック域 (続き)

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
12	文字	10	ファイルが入っているライブラリーの名前。インライン・データ・ファイルの場合は、値は *N です。	全タイプ
22	文字	10	スプール・ファイルの名前。スプール入力または出力レコードが入っているデータベース・ファイルの名前	スプールされているプリンターまたはディスクット、またはインライン・データ
32	文字	10	スプール・ファイルが入っているライブラリーの名前	スプールされているプリンターまたはディスクット、またはインライン・データ
42	2 進数	2	スプール・ファイル番号	スプールされているプリンターまたはディスクット
44	2 進数	2	最大レコード長	全タイプ
46	2 進数	2	最大キー長	データベース
48	文字	10	メンバー名 <ul style="list-style-type: none"> • ODP タイプが DB の場合は、オフセット 2 で名前を指定されたファイルの中のメンバー名です。ファイルが MBR(*ALL) にオーバーライドされている場合は、最後のレコードを提供したメンバー名です。 • ODP タイプが SP の場合は、オフセット 22 で名前を指定されたファイルの中のメンバー名です。 	データベース、プリンター、ディスクット、およびインライン・データ
58	2 進数	4	予約済み	
62	2 進数	4	予約済み	
66	2 進数	2	ファイル・タイプ	全タイプ
			1 ディスプレイ	
			2 プリンター	
			4 ディスクット	
			5 テープ	
			9 保管	
			10 DDM	
			11 ICF	
			20 インライン・データ	
			21 データベース	

表 22. オープン・フィールドバック域 (続き)

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
68	文字	3	予約済み	
71	2 進数	2	表示画面上の行数または印刷ページ上の行数	ディスプレイ、プリンター
			ヌル・フィールド・バイト・マップの長さ	データベース
73	2 進数	2	表示画面上の桁数または印刷行の文字数	ディスプレイ、プリンター
			ヌル・キー・フィールド・バイト・マップの長さ	データベース
75	2 進数	4	オープン時点のメンバー内レコード数。結合論理ファイルの場合は、プライマリーの中のレコード数です。提供されるのは、ファイルが入力用にオープンされている場合だけです。	データベース、インライン・データ
79	文字	2	アクセス・タイプ: AR 到着順 KC キーによる、重複キー許容。重複キーには、先変更先出し (FCFO) 順でアクセスします。 KF キーによる、重複キー許容。重複キーには、先入れ先出し (FIFO) 順でアクセスします。 KL キーによる、重複キー許容。重複キーには、後入れ先出し (LIFO) 順でアクセスします。 KN キーによる、重複キー許容。重複キーにアクセスする順は、次のいずれかになります。 • 先入れ先出し (FIFO) 順 • 後入れ先出し (LIFO) 順 • 先変更先出し (FCFO) 順	データベース
81	文字	1	KU キーによる、固有 重複キー指示。セットされるのは、アクセス・パスが KC、KF、KL、KN、または KU の場合だけです。 D アクセス・パスが KF または KL の場合は、重複キーが許容されます。 U 重複キーは許容されません。すべてのキーは固有で、アクセス・パスは KU です。	データベース
82	文字	1	ソース・ファイル指示 Y ファイルはソース・ファイルです。 N ファイルはソース・ファイルではありません。	データベース、テープ、ディスク、およびインライン・データ
83	文字	10	予約済み	
93	文字	10	予約済み	

表 22. オープン・フィールドバック域 (続き)

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
103	2 進数	2	オープン・フィールドバック域のボリューム・ラベル・フィールドまでのオフセット	ディスクett、テープ
105	2 進数	2	ブロック化レコード入出力を使用するとき、1 つのブロックの中で読み取りまたは書き出しできる最大レコード数	全タイプ
107	2 進数	2	オーバーフロー行番号	プリンター
109	2 進数	2	ブロック化レコード入出力増分。ブロックの中の次のレコードをアドレッシングするのに、各レコードの開始に追加されなければならないバイト数です。	全タイプ
111	2 進数	4	予約済み	
115	文字	1	その他のフラグ	
			ビット 1: 予約済み	全タイプ
			ビット 2: ファイル共用可能	
			0 ファイルは共用可能としてオープンされませんでした。	
			1 ファイルは共用可能としてオープンされました (SHARE(*YES))。	
			ビット 3: コミットメント制御	データベース
			0 ファイルはコミットメント制御下ではありません。	
			1 ファイルはコミットメント制御下にあります。	
			ビット 4: コミットメント・ロック・レベル	データベース
			0 変更されたレコードだけがロックされます (LCKLVL(*CHG))。 このビットがゼロでオフセット 132 の文字のビット 8 が 1 の場合は、アクセスされたすべてのレコードがロックされますが、ファイル内の現在の位置が変わるとロックは解放されます (LCKLVL (*CS))。	
			1 アクセスされたレコードはすべてロックされます (LCKLVL (*ALL))。	
			ビット 5: メンバー・タイプ	データベース
			0 メンバーは物理ファイル・メンバーです。	
			1 メンバーは論理ファイル・メンバーです。	

表 22. オープン・フィールドバック域 (続き)

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
			ビット 6: フィールド・レベル記述 0 ファイルにはフィールド・レベル記述は入りません。 1 ファイルにはフィールド・レベル記述は入りません。	データベース以外のすべてのタイプ
			ビット 7: DBCS またはグラフィック可能ファイル 0 ファイルは DBCS またはグラフィック可能のフィールドを含みません。 1 ファイルは DBCS またはグラフィック可能のフィールドを含みます。	データベース、ディスプレイ、プリンター、テープ、ディスク、および ICF
			ビット 8: ファイル終了遅延 0 ファイル終了遅延処理は行われていません。 1 ファイル終了遅延処理が行われています。	データベース
116	文字	10	要求元装置の名前。ディスプレイ・ファイルの場合は、これは要求元装置であるディスプレイ装置記述の名前です。ICF ファイルの場合は、これはリモート・ロケーション *REQUESTER に関連するプログラム装置名です。 このフィールドが指定されるのは、装置かりリモート・ロケーション名 *REQUESTER がどちらかが、オープン操作または獲得操作によって、ファイルに接続されているときだけです。それ以外の場合は、このフィールドには *N が入っています。	ディスプレイ、ICF
126	2 進数	2	ファイル・オープン・カウント。ファイルが共用可能としてオープンされなかった場合は、このフィールドには 1 が入ります。ファイルが共用可能としてオープンされた場合は、このフィールドには現在このファイルに接続されているプログラムの数が入ります。	全タイプ
128	2 進数	2	予約済み	
130	2 進数	2	オープンされた基礎になる物理メンバー数。論理メンバーの場合は、これは論理メンバーがそれに対してオープンされた物理メンバーの数です。物理メンバーの場合は、このフィールドはいつも 1 に設定されています。	データベース
132	文字	1	その他のフラグ ビット 1: 複数メンバー処理 0 指定されているメンバーだけが処理されます。 1 すべてのメンバーが処理されます。	データベース

表 22. オープン・フィードバック域 (続き)

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
			ビット 2: 結合論理ファイル 0 ファイルは結合論理ファイルではありません。 1 ファイルは結合論理ファイルです。	データベース
			ビット 3: ローカルまたはリモート・データ (DDM ファイル) 0 データはローカル・サーバーに保管されます。 1 データはリモート・サーバーに保管されます。	データベース
			ビット 4: リモートのSystem/38 または iSeries のデータ (DDM ファイル)。適用できるのは、ビット 3 の値が 1 の場合だけです。 0 データはリモートのSystem/38 または iSeries サーバー上にあります。 1 データはリモートのSystem/38 または iSeries サーバー上にありません。	データベース
			ビット 5: 標識域の分離 0 標識はプログラムの入出力バッファの中です。 1 標識はプログラムの入出力バッファの中ではありません。ファイルの作成時に、DDS キーワード INDARA が使用されました。	プリンター、ディスプレイ、および ICF
			ビット 6: ユーザー・バッファ 0 サーバーでプログラムの入出力バッファを作成します。 1 ユーザー・プログラムが入出力バッファを提供します。	全タイプ
			ビット 7: 予約済み	

表 22. オープン・フィードバック域 (続き)

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
			<p>ビット 8: 追加のコミットメント・ロック・レベル標識記号。これはオフセット 115 の文字のビット 3 が 1 の場合のみ有効です。</p> <p>オフセット 115 の文字のビット 4 が 0 の場合:</p> <p>0 変更されたレコードだけがロックされます (LCKLVL(*CHG))。</p> <p>1 アクセスされたすべてのレコードがロックされますが、ファイル内の現在の位置が変わるとロックは解放されます (LCKLVL(*CS))。</p> <p>オフセット 115 の文字のビット 4 が 1 の場合:</p> <p>0 アクセスされたレコードはすべてロックされます (LCKLVL (*ALL))。</p> <p>1 予約済み</p>	データベース
133	文字	2	オープン識別コード。この値は、フル・オープン操作 (SHARE(*NO)) または SHARE(*YES) を指定してオープンされているファイルを最初にオープンするときに固有のものです。これは、ディスプレイ・ファイルおよび ICF ファイルの場合に使用されますが、ファイル・タイプすべてに関して設定されます。この値を使用して、ファイルと関連データ待ち行列の項目との一致を調べることができます。	全タイプ
135	2 進数	2	このフィールド値は、最大レコード様式長であり、この長さは、データとファイル固有の情報 (先頭文字用紙制御、オプション標識、応答標識、ソース順序番号、プログラム/サーバー間データなど) の両方の長さからなります。値がゼロの場合は、オフセット 44 のフィールドを使用してください。	プリンター、ディスクケット、テープ、および ICF
137	2 進数	2	バッファの文字データのコード化文字セット識別コード (CCSID)。	データベース
139	文字	1	その他のフラグ	データベース
			<p>ビット 1: 空文字 (null) 可能フィールド・ファイル</p> <p>0 ファイルは空文字 (null) 可能フィールドを含みません。</p> <p>1 ファイルは空文字 (null) 可能フィールドを含みます。</p>	
			<p>ビット 2: 可変長フィールド・ファイル</p> <p>0 ファイルは可変長フィールドを含みません。</p> <p>1 ファイルは可変長フィールドを含みます。</p>	データベース

表 22. オープン・フィードバック域 (続き)

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
			ビット 3: 可変長レコード処理 0 可変長レコード処理を行いません。 1 可変長レコード処理を行います。	データベース
			ビット 4: CCSID 文字置換 0 CCSID データ変換中に置換文字は使用されません。 1 CCSID データ変換中に置換文字が使用されることがあります。	データベース、ディスプレイ
			ビット 5: ジョブ・レベル・オープン標識 0 この ODP は、有効範囲がジョブ・レベルに限定されていません。 1 この ODP は、有効範囲がジョブ・レベルに限定されています。	全タイプ
			ビット 6 から 8: 予約済み	
140	文字	6	予約済み	
146	2 進数	2	この ODP に対して定義されている装置数。ディスプレイ・ファイルの場合は、これは、ディスプレイ・ファイル作成 (CRTDSPF) コマンドの DEV パラメーターで定義された装置数によって決まります。ICF ファイルの場合は、これは、ICF プログラム装置項目追加 (ADDICFDEVE) コマンドまたは ICF プログラム装置項目オーバーライド (OVRICFDEVE) コマンドを用いて定義または獲得されたプログラム装置数によって決まります。それ以外のファイルの場合は、すべて値 1 になります。	全タイプ
148	文字		装置名定義リスト。	全タイプ

関連概念

92 ページの『オープンおよび入出力フィードバック域でのファイル状況のモニター』

サーバーは、ファイルが正常にオープンされると、フィードバック域でファイルの状況をモニターします。サーバーがファイルに対する操作を行うと、フィードバック域が更新され、最新のファイル状況を反映します。これらのフィードバック域によって、ユーザーはより一層アプリケーションを管理することができ、エラー発生時に重要な情報を得ることができます。

装置定義リスト:

オープン・フィードバック域の装置定義リスト部分は、配列構造です。配列の中の各項目には、ファイルに接続されている各装置または通信セッションに関する情報が入ります。この配列の中の項目数は、オープン・フィードバック域のオフセット 146 における数値によって決まります。

装置定義リストは、オープン・フィードバック域のオフセット 148 から始まります。装置定義リストに関して示されているオフセットは、オープン・フィードバック域の初めからでなく、装置定義リストの初めからになっています。

表 23. 装置定義リスト

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
0	文字	10	プログラム装置名。データベース・ファイルの場合は、値は DATABASE です。スプールされているプリンターまたはディスク・ファイルの場合、値は *N です。保管ファイルの場合、値は *NONE です。ICF ファイルの場合は、値は、ADDICFDEVE コマンドまたは OVRICFDEVE コマンドに由来するプログラム装置の名前です。それ以外のファイルの場合は、すべて値は装置記述の名前です。	インライン・データ以外のすべてのタイプ
10	文字	50	予約済み	
60	文字	10	装置記述名。スプールされているプリンターまたはディスク・ファイルの場合、値は *N です。保管ファイルの場合、値は *NONE です。それ以外のファイルの場合は、すべて値は装置記述の名前です。	データベースおよびインライン・データ以外のすべてのタイプ
70	文字	1	装置クラス	データベースおよびインライン・データ以外のすべてのタイプ
			16 進 01 ディスプレイ	
			16 進 02 プリンター	
			16 進 04 ディスク	
			16 進 05 テープ	
			16 進 09 保管	
			16 進 0B ICF	

表 23. 装置定義リスト (続き)

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
71	文字	1	装置タイプ	
			16 進 02 5256 プリンター	
			16 進 07 5251 ディスプレイ	
			16 進 08 スプール	
			16 進 0A BSCEL	
			16 進 0B 5291 ディスプレイ	
			16 進 0C 5224/5225 プリンター	
			16 進 0D 5292 ディスプレイ	
			16 進 0E APPC	
			16 進 0F 5219 プリンター	
			16 進 10 5583 プリンター (DBCS)	
			16 進 11 5553 プリンター	
			16 進 12 5555-B01 ディスプレイ	
			16 進 13 3270 ディスプレイ	
			16 進 14 3270 プリンター	
			16 進 15 グラフィック可能装置	
			16 進 16 金融機関ディスプレイ	
			16 進 17 3180 ディスプレイ	
			16 進 18 保管ファイル	
			16 進 19 3277 DHCF 装置	

表 23. 装置定義リスト (続き)

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
71	文字	1	装置タイプ 16 進 1A 9347 テープ装置 16 進 1B 9348 テープ装置 16 進 1C 9331-1 ディスケット装置 16 進 1D 9331-2 ディスケット装置 16 進 1E システム間通信サポート 16 進 1F 非同期通信サポート 16 進 20 SNUF 16 進 21 4234 (SCS) プリンター 16 進 22 3812 (SCS) プリンター 16 進 23 4214 プリンター 16 進 24 4224 (IPDS™) プリンター 16 進 25 4245 プリンター 16 進 26 3179-2 ディスプレイ 16 進 27 3196-A ディスプレイ 16 進 28 3196-B ディスプレイ 16 進 29 5262 プリンター 16 進 2A 6346 テープ装置 16 進 2B 2440 テープ装置 16 進 2C 9346 テープ装置	

表 23. 装置定義リスト (続き)

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
71	文字	1	装置タイプ	
			16 進 2D	
			6331 ディスケット装置	
			16 進 2E	
			6332 ディスケット装置	
			16 進 30	
			3812 (IPDS) プリンター	
			16 進 31	
			4234 (IPDS) プリンター	
			16 進 32	
			IPDS プリンター (型式不明)	
			16 進 33	
			3197-C1 ディスプレイ	
			16 進 34	
			3197-C2 ディスプレイ	
			16 進 35	
			3197-D1 ディスプレイ	
			16 進 36	
			3197-D2 ディスプレイ	
			16 進 37	
			3197-W1 ディスプレイ	
			16 進 38	
			3197-W2 ディスプレイ	
			16 進 39	
			5555-E01 ディスプレイ	
			16 進 3A	
			3430 テープ装置	
			16 進 3B	
			3422 テープ装置	
			16 進 3C	
			3480 テープ装置	
			16 進 3D	
			3490 テープ装置	
			16 進 3E	
			3476-EA ディスプレイ	
			16 進 3F	
			3477-FG ディスプレイ	
			16 進 40	
			3278 DHCF 装置	

表 23. 装置定義リスト (続き)

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
71	文字	1	装置タイプ	
			16 進 41 3279 DHCF 装置	
			16 進 42 ICF 金融機関用装置	
			16 進 43 小売業用通信装置	
			16 進 44 3477-FA ディスプレイ	
			16 進 45 3477-FC ディスプレイ	
			16 進 46 3477-FD ディスプレイ	
			16 進 47 3477-FW ディスプレイ	
			16 進 48 3477-FE ディスプレイ	
			16 進 49 6367 テープ装置	
			16 進 4A 6347 テープ装置	
			16 進 4D ネットワーク仮想端末ディスプレイ	
			16 進 4E 6341 テープ装置	
			16 進 4F 6342 テープ装置	
			16 進 50 6133 ディスケット装置	
			16 進 51 5555-C01 ディスプレイ	
			16 進 52 5555-F01 ディスプレイ	
			16 進 53 6366 テープ装置	
			16 進 54 7208 テープ装置	
			16 進 55 6252 (SCS) プリンター	

表 23. 装置定義リスト (続き)

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
71	文字	1	装置タイプ	
			16 進 56 3476-EC ディスプレイ	
			16 進 57 4230 (IPDS) プリンター	
			16 進 58 5555-G01 ディスプレイ	
			16 進 59 5555-G02 ディスプレイ	
			16 進 5A 6343 テープ装置	
			16 進 5B 6348 テープ装置	
			16 進 5C 6368 テープ装置	
			16 進 5D 3486-BA ディスプレイ	
			16 進 5F 3487-HA ディスプレイ	
			16 進 60 3487-HG ディスプレイ	
			16 進 61 3487-HW ディスプレイ	
			16 進 62 3487-HC ディスプレイ	
			16 進 63 3935 (IPDS) プリンター	
			16 進 64 6344 テープ装置	
			16 進 65 6349 テープ装置	
			16 進 66 6369 テープ装置	
			16 進 67 6380 テープ装置	
			16 進 68 6378 テープ装置	
			16 進 69 6390 テープ装置	

表 23. 装置定義リスト (続き)

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
71	文字	1	装置タイプ 16 進 70 6379 テープ装置 16 進 71 9331-11 ディスケット装置 16 進 72 9331-12 ディスケット装置 16 進 73 3570 テープ装置 16 進 74 3590 テープ装置 16 進 75 6335 テープ装置	
72	2 進数	2	表示画面上の行数	ディスプレイ
74	2 進数	2	表示画面の各行あたりの桁数	ディスプレイ

表 23. 装置定義リスト (続き)

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
76	文字	2	ビット・フラグ ビット 1: 明滅機能 0 表示は明滅することができません。 1 表示は明滅することができます。 ビット 2: 装置ロケーション 0 ローカル装置 1 リモート装置 ビット 3: 獲得状況。装置がオープン時に暗黙に獲得されても、このビットはセットされます。 0 装置は獲得されません。 1 装置は獲得されます。 ビット 4: 送信勧誘状況 0 装置は送信勧誘されません。 1 装置は送信勧誘されます。 ビット 5: データ使用可能状況 (装置が送信勧誘される場合のみ) 0 データは使用不能です。 1 データは使用可能です。 ビット 6: トランザクション状況。 0 トランザクションは開始されていません。呼び出し要求がまだ送られていないか、切断要求がすでに送信または受信されたか、あるいはトランザクションがすでに完了していません。 1 トランザクションは開始されています。トランザクションは活動状態です。呼び出し要求が送られるか受け取られ、トランザクションは終了していません。	ディスプレイ

表 23. 装置定義リスト (続き)

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
76	文字	2	ビット・フラグ ビット 7: 要求元装置 0 要求元装置ではありません。 1 要求元装置です。 ビット 8: DBCS 装置 0 装置は 2 バイト文字データを処理することができません。 1 装置は 2 バイト文字データを処理することができます。 ビット 9 から 10: 予約済み ビット 11: DBCS キーボード 0 キーボードは 2 バイト文字データを入力することができません。 1 キーボードは 2 バイト文字データを入力することができます。 ビット 12 から 16: 予約済み	ディスプレイ
78	文字	1	同期レベル 16 進 00 トランザクションは SYNLVL(*NONE) で作成されました。確認処理は許されません。 16 進 01 トランザクションは SYNLVL(*CONFIRM) で作成されました。確認処理は可能です。 16 進 02 トランザクションは SYNLVL(*COMMIT) で作成されました。	ICF
79	文字	1	会話タイプ 16 進 D0 基本会話 (CNVTYPE(*USER)) 16 進 D1 マップ式会話 (CNVTYPE(*SYS))	ICF
80	文字	50	予約済み	

ボリューム・ラベル・フィールド:

このトピックでは、ボリューム・ラベル・フィールドとそのプロパティを表に示します。

表 24. ボリューム・ラベル・フィールド

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
0	文字	128	現行ボリュームのボリューム・ラベル	ディスク、テープ
128	文字	128	オープンされたファイルの見出しラベル 1	ディスク、テープ
256	文字	128	オープンされたファイルの見出しラベル 2	テープ

入出力フィールドバック域

iSeries では、OS/400 メッセージおよび入出力フィールドバック情報を使用して、入出力操作の結果をプログラムに通知します。サーバーは、正常に行われたすべての入出力操作で入出力フィールドバック域を更新しますが、ユーザーのプログラムがブロック化レコード入出力を使用している場合はその限りではありません。

この場合、サーバーがフィールドバック域を更新するのは、1 ブロックのレコードが読み取りまたは書き出しされたときだけです。情報の中には、ブロックの中の最後のレコードを反映するものがあります。それ以外の情報では (入出力操作のカウントなど)、レコードのブロックに対する操作回数は反映しますが、レコード数は反映しません。プログラムがブロック化レコード入出力を使用するかどうかの判別には、ご使用の高水準言語の資料を参照してください。

入出力フィールドバック域は、共通域とファイル依存域の 2 つの部分から成っています。

関連概念

92 ページの『オープンおよび入出力フィールドバック域でのファイル状況のモニター』

サーバーは、ファイルが正常にオープンされると、フィールドバック域でファイルの状況をモニターします。サーバーがファイルに対する操作を行うと、フィールドバック域が更新され、最新のファイル状況を反映します。これらのフィールドバック域によって、ユーザーはより一層アプリケーションを管理することができ、エラー発生時に重要な情報を得ることができます。

共通入出力フィールドバック域:

このトピックでは、iSeries の共通入出力フィールドバック域を表に示します。

表 25. 共通入出力フィールドバック域

オフセット	データ・タイプ	長さ	内容
0	2 進数	2	ファイル依存フィールドバック域までのオフセット
2	2 進数	4	書き出し操作カウント。書き出し操作が正常に完了したときだけ更新されます。ブロック化レコード入出力操作の場合、このカウントはブロック数であって、レコード数ではありません。
6	2 進数	4	読み取り操作カウント。読み取り操作が正常に完了したときだけ更新されます。ブロック化レコード入出力操作の場合、このカウントはブロック数であって、レコード数ではありません。
10	2 進数	4	書き出し読み取り操作カウント。書き出し読み取り操作が正常に完了したときだけ更新されます。

表 25. 共通入出力フィールドバック域 (続き)

オフセット	データ・タイプ	長さ	内容
14	2 進数	4	他の操作カウント。書き出し、読み取り、または読み書き操作以外に正常に行われた操作の回数。操作が正常に完了したときだけ更新されます。このカウントには、更新、削除、データ強制終了、ボリューム強制終了、データ終了変更、レコード・ロック解放、および装置獲得/解放操作が含まれます。
18	文字	1	予約済み
19	文字	1	現操作
			16 進 01 読み取りまたはブロック読み取りまたは送信を勧誘した装置からの読み取り
			16 進 02 直接読み取り
			16 進 03 キーによる読み取り
			16 進 05 書き出しまたはブロック書き出し
			16 進 06 書き出し読み取り
			16 進 07 更新
			16 進 08 削除
			16 進 09 データ強制終了
			16 進 0A ボリューム強制終了
			16 進 0D レコード・ロック解放
			16 進 0E データ終了変更
			16 進 0F 削除済みレコード配置
			16 進 11 装置解放
			16 進 12 装置獲得

表 25. 共通入出力フィールドバック域 (続き)

オフセット	データ・タイプ	長さ	内容
20	文字	10	<p>処理されたばかりのレコード様式の名前で、この名前は次の 2 つのどちらかにより決定します。</p> <ul style="list-style-type: none"> 入出力要求で指定 デフォルトまたは様式選択処理で決定 <p>ディスプレイ・ファイルの場合は、デフォルトの名前はファイルの中の唯一のレコード様式の名前か、画面に書き出された入力可能フィールドの入っているレコードの前のレコード様式名のどちらかです。ディスプレイ・ファイルでは、複数の様式を同時に画面上で備えることがあるため、この様式が最後のカーソル位置に入力された様式を表さない場合もあります。</p> <p>ICF ファイルの場合は、様式名は使用されている様式選択オプションに基づいて、サーバーによって決められます。詳しくは、ICF Programming  を参照してください。</p> <p>装置クラス</p> <p>バイト 1</p> <p>16 進 00 データベース</p> <p>16 進 01 ディスプレイ</p> <p>16 進 02 プリンター</p> <p>16 進 04 ディスクット</p> <p>16 進 05 テープ</p> <p>16 進 09 保管</p> <p>16 進 0B ICF</p> <p>16 進 00 キーのないファイル</p> <p>16 進 01 キー付きファイル</p> <p>バイト 2 (バイト 1 に 16 進 00 が入っていない場合)</p> <p>16 進 02 5256 プリンター</p>
30	文字	2	

表 25. 共通入出力フィードバック域 (続き)

オフセット	データ・タイプ	長さ	内容
30	文字	2	装置クラス
			16 進 07 5251 ディスプレイ
			16 進 08 スプール
			16 進 0A BSCEL
			16 進 0B 5291 ディスプレイ
			16 進 0C 5224/5225 プリンター
			16 進 0D 5292 ディスプレイ
			16 進 0E APPC
			16 進 0F 5219 プリンター
			16 進 10 5583 プリンター (DBCS)
			16 進 11 5553 プリンター
			16 進 12 5555-B01 ディスプレイ
			16 進 13 3270 ディスプレイ
			16 進 14 3270 プリンター
			16 進 15 グラフィック可能装置
			16 進 16 金融機関ディスプレイ

表 25. 共通入出力フィールドバック域 (続き)

オフセット	データ・タイプ	長さ	内容
30	文字	2	装置クラス
			16 進 17 3180 ディスプレイ
			16 進 18 保管ファイル
			16 進 19 3277 DHCF 装置
			16 進 1A 9347 テープ装置
			16 進 1B 9348 テープ装置
			16 進 1C 9331-1 ディスケット装置
			16 進 1D 9331-2 ディスケット装置
			16 進 1E システム間通信サポート
			16 進 1F 非同期通信サポート
			16 進 20 SNUF
			16 進 21 4234 (SCS) プリンター
			16 進 22 3812 (SCS) プリンター
			16 進 23 4214 プリンター
			16 進 24 4224 (IPDS) プリンター
			16 進 25 4245 プリンター
			16 進 26 3179-2 ディスプレイ
			16 進 27 3196-A ディスプレイ
			16 進 28 3196-B ディスプレイ
			16 進 29 5262 プリンター

表 25. 共通入出力フィードバック域 (続き)

オフセット	データ・タイプ	長さ	内容
30	文字	2	装置クラス
			16 進 2A 6346 テープ装置
			16 進 2B 2440 テープ装置
			16 進 2C 9346 テープ装置
			16 進 2D 6331 ディスケット装置
			16 進 2E 6332 ディスケット装置
			16 進 30 3812 (IPDS) プリンター
			16 進 31 4234 (IPDS) プリンター
			16 進 32 IPDS プリンター (型式不明)
			16 進 33 3197-C1 ディスプレイ
			16 進 34 3197-C2 ディスプレイ
			16 進 35 3197-D1 ディスプレイ
			16 進 36 3197-D2 ディスプレイ
			16 進 37 3197-W1 ディスプレイ
			16 進 38 3197-W2 ディスプレイ
			16 進 39 5555-E01 ディスプレイ
			16 進 3A 3430 テープ装置
			16 進 3B 3422 テープ装置
			16 進 3C 3480 テープ装置
			16 進 3D 3490 テープ装置

表 25. 共通入出力フィールドバック域 (続き)

オフセット	データ・タイプ	長さ	内容
30	文字	2	装置クラス
			16 進 3E 3476-EA ディスプレイ
			16 進 3F 3477-FG ディスプレイ
			16 進 40 3278 DHCF 装置
			16 進 41 3279 DHCF 装置
			16 進 42 ICF 金融機関用装置
			16 進 43 小売業用通信装置
			16 進 44 3477-FA ディスプレイ
			16 進 45 3477-FC ディスプレイ
			16 進 46 3477-FD ディスプレイ
			16 進 47 3477-FW ディスプレイ
			16 進 48 3477-FE ディスプレイ
			16 進 49 6367 テープ装置
			16 進 4A 6347 テープ装置
			16 進 4D ネットワーク仮想端末ディスプレイ
			16 進 4E 6341 テープ装置
			16 進 4F 6342 テープ装置
			16 進 50 6133 ディスケット装置
			16 進 51 5555-C01 ディスプレイ
			16 進 52 5555-F01 ディスプレイ

表 25. 共通入出力フィードバック域 (続き)

オフセット	データ・タイプ	長さ	内容
30	文字	2	装置クラス
			16 進 53 6366 テープ装置
			16 進 54 7208 テープ装置
			16 進 55 6252 (SCS) プリンター
			16 進 56 3476-EC ディスプレイ
			16 進 57 4230 (IPDS) プリンター
			16 進 58 5555-G01 ディスプレイ
			16 進 59 5555-G02 ディスプレイ
			16 進 5A 6343 テープ装置
			16 進 5B 6348 テープ装置
			16 進 5C 6368 テープ装置
			16 進 5D 3486-BA ディスプレイ
			16 進 5F 3487-HA ディスプレイ
			16 進 60 3487-HG ディスプレイ
			16 進 61 3487-HW ディスプレイ
			16 進 62 3487-HC ディスプレイ
			16 進 63 3935 (IPDS) プリンター
			16 進 64 6344 テープ装置
			16 進 65 6349 テープ装置
			16 進 66 6369 テープ装置

表 25. 共通入出力フィードバック域 (続き)

オフセット	データ・タイプ	長さ	内容
30	文字	2	装置クラス 16 進 67 6380 テープ装置 16 進 68 6378 テープ装置 16 進 69 6390 テープ装置 16 進 70 6379 テープ装置 16 進 71 9331-11 ディスケット装置 16 進 72 9331-12 ディスケット装置 16 進 73 3570 テープ装置 16 進 74 3590 テープ装置 16 進 75 6335 テープ装置
32	文字	10	装置名。操作が完了したばかりの装置の名前です。指定されるのは、ディスプレイ、プリンター、テープ装置、ディスク装置、および ICF ファイルの場合だけです。スプールされているプリンターまたはディスク・ファイルの場合、値は *N です。ICF ファイルの場合は、値はプログラム装置名です。それ以外のファイルの場合は、値は装置記述名です。
42	2 進数	4	最後の入出力操作で処理されたレコードの長さ (ICF、ディスプレイ、テープ装置、またはデータベース・ファイルの場合のみ) です。ICF 書き出し操作では、これはデータのレコード長です。ICF 読み取り操作では、これは最後の読み取り操作に関連するレコードのレコード長です。
46	文字	80	予約済み
126	2 進数	2	ブロック化レコードに対する読み取り要求で検索されるか、またはブロック化レコードに対する書き出しまたはデータ強制終了もしくはボリューム強制終了要求で送られたレコード数。指定されるのは、データベース、ディスク、およびテープ・ファイルの場合だけです。

表 25. 共通入出力フィールドバック域 (続き)

オフセット	データ・タイプ	長さ	内容
128	2 進数	2	出力の場合、このフィールドの値はレコード様式長であり、先頭文字用紙制御、オプション標識、ソース順序番号、およびプログラム/サーバー間データの長さも含まれます。この値がゼロの場合は、オフセット 42 のフィールドを使用してください。 入力の場合は、このフィールドの値はレコード様式長であり、応答標識およびソース順序番号の長さも含まれます。この値がゼロの場合は、オフセット 42 のフィールドを使用してください。
130	文字	2	予約済み
132	2 進数	4	現行ブロック・カウント。すでに読み取りまたは書き出しが行われたテープ・データ・ファイルのブロック数です。テープ・ファイル専用。
136	文字	8	予約済み

ICF ファイルおよびディスプレイ・ファイルの入出力フィールドバック域:

このトピックでは、ICF ファイルおよびディスプレイ・ファイルの入出力フィールドバック域を表に示します。

表 26. ICF ファイルおよびディスプレイ・ファイルの入出力フィールドバック域

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
0	文字	2	フラグ・ビット ビット 1: 読み取り取り消し標識 0 読み取り取り消し操作で読み取り要求が取り消されませんでした。 1 読み取り取り消し操作で読み取り要求が取り消されました。 ビット 2: データ戻り標識 0 読み取り取り消し操作で入力バッファの内容が変更されませんでした。 1 読み取り取り消し操作で、非待機指定読み取り操作に由来するデータが入力バッファに入りました。 ビット 3: コマンド・キー標識 0 この標識を設定する条件が生じませんでした。 1 ページ印刷、Help、Home、次ページ、前ページ、または Clear キーが押されました。キーの活動化は DDS キーワードで指定されましたが、応答標識は指定されていません。 ビット 4 から 16: 予約済み	ディスプレイ

表 26. ICF ファイルおよびディスプレイ・ファイルの入出力フィールドバック域 (続き)

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
2	文字	1	<p>アテンション標識バイト (AID)。このフィールドでは、どの機能キーが押されたかを識別します。</p> <p>ICF ファイルの場合は、このフィールドには 16 進 F1 が入り、ディスプレイ装置で Enter キーが押された場合を模倣します。</p> <p>ディスプレイ・ファイルの場合は、このフィールドには装置から戻された 1 バイトの 16 進数値が入ります。</p> <p>16 進コード 機能キー</p> <p>16 進 31 1</p> <p>16 進 32 2</p> <p>16 進 33 3</p> <p>16 進 34 4</p> <p>16 進 35 5</p> <p>16 進 36 6</p> <p>16 進 37 7</p> <p>16 進 38 8</p> <p>16 進 39 9</p> <p>16 進 3A 10</p> <p>16 進 3B 11</p> <p>16 進 3C 12</p> <p>16 進 B1 13</p> <p>16 進 B2 14</p>	

表 26. ICF ファイルおよびディスプレイ・ファイルの入出力フィールドバック域 (続き)

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
2	文字	1	16 進 B3 15	ファイル・タイプ ディスプレイ、 ICF
			16 進 B4 16	
			16 進 B5 17	
			16 進 B6 18	
			16 進 B7 19	
			16 進 B8 20	
			16 進 B9 21	
			16 進 BA 22	
			16 進 BB 23	
			16 進 BC 24	
			16 進 BD Clear	
			16 進 F1 実行/レコード前進	
			16 進 F3 Help (オペレーター・エラー・モードではない)	
			16 進 F4 前ページ	
			16 進 F5 次ページ	
			16 進 F6 ページ印刷	
			16 進 F8 レコード・バックスペース	
			16 進 3F 自動入力 (選択ライト・ペン用)	

表 26. ICF ファイルおよびディスプレイ・ファイルの入出力フィールドバック域 (続き)

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
3	文字	2	カーソル位置 (行および桁)。データをプログラムに戻すサブファイル操作ではない入力操作で更新されます。たとえば、16 進 0102 は 1 行目の 2 桁目を意味します。10 行目の 33 桁目は 16 進 0A21 になります。	ディスプレイ
5	2 進数	4	実データ長。ICF ファイルの詳細については、ICF Programming  を参照してください。ディスプレイ・ファイルの場合、これは入出力操作によって処理されたレコード様式の長さです。	ディスプレイ、ICF
9	2 進数	2	サブファイル・レコードの相対レコード番号。サブファイル・レコード操作の場合に更新されます。入力操作の場合は、データがプログラムに戻された場合に限り更新されません。ディスプレイ上に複数のサブファイルが表示されている場合、このオフセットには、最後に更新されたサブファイルの相対レコード番号が含まれます。	ディスプレイ
11	2 進数	2	最も低いサブファイル。最後の書き出し操作が SFLDSP を指定して、サブファイル制御レコードに対してなされた場合は、最も上のサブファイル表示域に現在表示されている最も低い相対レコード番号を示します。画面送り戻し操作の場合に更新されます。別のレコードに対する書き出し操作で 0 にリセットされます。メッセージ・サブファイルの場合は、設定されません。	ディスプレイ
13	2 進数	2	サブファイルの中のレコード総数。サブファイル・レコードに対する PUT 相対操作で更新されます。SFLINZ キーワードがオプションで選択されているサブファイル制御レコードに対して書き込みまたは書き込み/読み取り操作を行うと、この数値はゼロにセットされます。ディスプレイに表示されている複数のサブファイルに対してレコードが置かれる場合は、このオフセットにはすべてのサブファイルのレコード数が含まれます。これは、SFLINZ キーワードがオプションで選択されたサブファイル制御レコードに対して書き込みまたは書き込み/読み取り操作が行われていないと想定されるからです。	ディスプレイ
15	文字	2	活動ウィンドウ内のカーソル位置 (行および桁)。データをプログラムに戻すサブファイル操作ではない入力操作で更新されます。たとえば、16 進数 0203 は活動ウィンドウの上部左端にある 2 行目の 3 桁目を意味します。	ディスプレイ

表 26. ICF ファイルおよびディスプレイ・ファイルの入出力フィールドバック域 (続き)



オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
17	文字	17	予約済み	
34	文字	2	メジャー戻りコード	ディスプレイ、 ICF
			00 操作は正常に完了しました。	
			02 入力操作は正常に完了しましたが、ジョブ (制御付き) は取り消されました。	
			03 入力操作は正常に完了しましたが、データは受信されませんでした。	
			04 出力例外	
			08 装置はすでに獲得されています。	
			11 送信を勧誘した装置からの読み取りは正常に行われませんでした。	
			34 入力例外	
			80 永続サーバー・エラーまたはファイル・エラー	
			81 永続セッション・エラーまたは装置エラー	
			82 獲得操作またはオープン操作が正常に行われませんでした。	
			83 回復可能セッション・エラーまたは装置エラー	
36	文字	2	マイナー戻りコード。ディスプレイ・ファイルの値については、Application Display Programming  を参照してください。ICF ファイルの値については、ICF Programming  および該当する通信タイプのプログラマーの手引きを参照してください。	ディスプレイ、 ICF
38	文字	8	システム・ネットワーク体系 (SNA) センス戻りコード。一部の戻りコードの場合、このフィールドには、エラーの理由に関する詳細な情報が入ることがあります。SNA センス・コードの説明については、該当する SNA 資料を参照してください。	ICF
46	文字	1	安全標識:	ICF
			0 テキスト終結 (ETX) 制御文字は受け取られていません。	
			1 ETX 制御文字は受け取られています。	
47	文字	1	予約済み	

表 26. ICF ファイルおよびディスプレイ・ファイルの入出力フィールドバック域 (続き)

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
48	文字	1	リモート・システム/アプリケーションからの書き出し要求 (RQSWRT) コマンド	ICF
			0 RQSWRT は受け取られていません。	
			1 RQSWRT は受け取られています。	
49	文字	10	リモート・サーバーから受け取ったレコード様式名	ICF
59	文字	4	予約済み	
63	文字	8	モード名	ICF
71	文字	9	予約済み	

関連概念

92 ページの『オープンおよび入出力フィールドバック域でのファイル状況のモニター』サーバーは、ファイルが正常にオープンされると、フィールドバック域でファイルの状況をモニターします。サーバーがファイルに対する操作を行うと、フィールドバック域が更新され、最新のファイル状況を反映します。これらのフィールドバック域によって、ユーザーはより一層アプリケーションを管理することができ、エラー発生時に重要な情報を得ることができます。

プリンター・ファイルの入出力フィールドバック域:

このトピックでは、プリンター・ファイルの入出力フィールドバック域を表に示します。

表 27. プリンター・ファイルの入出力フィールドバック域

オフセット	データ・タイプ	長さ	内容
0	2 進数	2	ページの中の現在行番号
2	2 進数	4	現ページ・カウント
6	文字	1	ビット 1: スプール・ファイルの削除状況:
			1 スプール・ファイルは削除されました。
			0 スプール・ファイルは削除されていません。
			ビット 2 から 8: 予約済み
7	文字	27	予約済み
34	文字	2	メジャー戻りコード
			00 操作は正常に完了しました。
			80 永続サーバー・エラーまたはファイル・エラー
			81 永続装置エラー
			82 オープン操作は行われませんでした。
			83 回復可能装置エラーが起きました。
36	文字	2	マイナー戻りコード。

関連概念

92 ページの『オープンおよび入出力フィードバック域でのファイル状況のモニター』
 サーバーは、ファイルが正常にオープンされると、フィードバック域でファイルの状況をモニターしま
 す。サーバーがファイルに対する操作を行うと、フィードバック域が更新され、最新のファイル状況
 を反映します。これらのフィードバック域によって、ユーザーはより一層アプリケーションを管理す
 ることができ、エラー発生時に重要な情報を得ることができます。

関連情報

印刷

データベース・ファイルの入出力フィードバック域:

このトピックでは、データベース・ファイルの入出力フィードバック域を表に示します。

表 28. データベース・ファイルの入出力フィードバック域

オフセット	データ・タイプ	長さ	内容
0	2 進数	4	キーおよびヌル・キー・フィールド・パイ ト・マップを含むデータベース・フィード バック域のサイズ。
4	文字	4	ビット 1 から 32: 各ビットは、JFILE キーワードの中 の結合論理ファイルを表します。 0 JDFTVAL がファイルに対 して指定されていません。 1 JDFTVAL がファイルに対 して指定されています。
8	2 進数	2	データベース・ファイル用の入出力フィード バック域の先頭から、キー値に続くヌル・キ ー・フィールド・バイト・マップまでのオフ セット (この区域ではオフセット 34 で開始 する)。
10	2 進数	2	ロックされたレコードの数
12	2 進数	2	フィールドの最大数
14	2 進数	4	フィールド・マッピング・エラー・ビットマ ップまでのオフセット
18	文字	1	ファイルの現在位置標識 ビット 1: ファイルの現在位置の GET-NEXT-KEY EQUAL 操作の場 合の有効性。 0 ファイル位置は有効ではあ りません。 1 ファイル位置は有効です。 ビット 2 から 8: 予約済み

表 28. データベース・ファイルの入出力フィールドバック域 (続き)

オフセット	データ・タイプ	長さ	内容
19	文字	1	<p>現行レコード削除済み標識</p> <p>ビット 1 から 2: 予約済み</p> <p>ビット 3: 次メッセージ標識</p> <p>0 次メッセージはファイルの 終わりではありません。</p> <p>1 次メッセージはファイルの 終わりである可能性があります。</p> <p>ビット 4: 削除済みレコード標識</p> <p>0 現ファイル位置は活動レコ ードの中です。</p> <p>1 現ファイル位置は削除済み レコードの中です。</p> <p>ビット 5: 書き出し操作キー・フィールドバック 標識</p> <p>0 キー・フィールドバックは最 後の書き出し操作によって 与えられません。</p> <p>1 キー・フィールドバックは最 後の書き出し操作によって 与えられます。</p> <p>ビット 6: ファイル位置変更済み標識。設定さ れるのは、読み取り操作および位置 指定入出力操作の場合だけです。書 き出し、更新、および入出力削除操 作では設定されません。</p> <p>0 ファイル位置は変更されま せませんでした。</p> <p>1 ファイル位置は変更されま した。</p>

表 28. データベース・ファイルの入出力フィールドバック域 (続き)

オフセット	データ・タイプ	長さ	内容
19	文字	1	<p>現行レコード削除済み標識</p> <p>ビット 7:</p> <p>保留例外標識。入力専用としてオープンされ、SEQONLY(*YES N) (ここで、N は 1 より大) であるファイルの場合に有効です。</p> <p>0 保留状態の検索エラーは存在していません。</p> <p>1 保留状態の検索エラーが存在しています。</p> <p>ビット 8:</p> <p>重複キー標識</p> <p>0 最後に行った読み取りまたは書き出し操作のキーは重複キーではありませんでした。</p> <p>1 最後に行った読み取りまたは書き出し操作のキーは重複キーでした。</p>
20	2 進数	2	<p>キー・フィールド数。このオフセットは、2 進操作に使用してください。次のオフセット (オフセット 21) は、文字操作に使用してください。これらのオフセットはオーバーラップし、同じ値になります (キー・フィールドの数は 32 以下であり、オフセット 20 の最下位バイトだけが使用されます)。</p>
21	文字	1	キー・フィールド数。
22	文字	4	予約済み
26	2 進数	2	キー長
28	2 進数	2	データ・メンバー番号
30	2 進数	4	データ・メンバーの中の相対レコード番号
34	文字	*	キー値
*	文字	*	ヌル・キー・フィールド・バイト・マップ

属性取得フィールドバック域:

属性取得操作を実行することにより、特定のディスプレイ装置または ICF セッションに関する特定の情報を得ることができます。

表 29. 属性の取得

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
0	文字	10	プログラム装置名。	ディスプレイ、 ICF
10	文字	10	装置記述名。この項目に対応する装置記述の名前。	ディスプレイ、 ICF

表 29. 属性の取得 (続き)

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
20	文字	10	ユーザー ID	ディスプレイ、 ICF
30	文字	1	装置クラス D ディスプレイ I ICF U 不明	ディスプレイ、 ICF
31	文字	6	装置タイプ 3179 3179 ディスプレイ 317902 3179-2 ディスプレイ 3180 3180 ディスプレイ 3196A 3196-A1/A2 ディスプレイ 3196B 3196-B1/B2 ディスプレイ 3197C1 3197-C1 ディスプレイ 3197C2 3197-C2 ディスプレイ 3197D1 3197-D1 ディスプレイ 3197D2 3197-D2 ディスプレイ 3197W1 3197-W1 ディスプレイ 3197W2 3197-W2 ディスプレイ 3270 3270 ディスプレイ 3476EA 3476-EA ディスプレイ 3476EC 3476-EC ディスプレイ 3477FA 3477-FA ディスプレイ 3477FC 3477-FC ディスプレイ 3477FD 3477-FD ディスプレイ 3477FE 3477-FE ディスプレイ 3477FG 3477-FG ディスプレイ 3477FW 3477-FW ディスプレイ 525111 5251 ディスプレイ 5291 5291 ディスプレイ 5292 5292 ディスプレイ 529202 5292-2 ディスプレイ	

表 29. 属性の取得 (続き)

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
			5555B1 5555-B01 ディスプレイ	ディスプレイ、 ICF
			5555C1 5555-C01 ディスプレイ	
			5555E1 5555-E01 ディスプレイ	
			5555F1 5555-F01 ディスプレイ	
			5555G1 5555-G01 ディスプレイ	
			5555G2 5555-G02 ディスプレイ	
			DHCF77 3277 DHCF 装置	
			DHCF78 3278 DHCF 装置	
			DHCF79 3279 DHCF 装置	
			3486BA 3486-BA ディスプレイ	ディスプレイ、 ICF
			3487HA 3487-HA ディスプレイ	
			3487HC 3487-HC ディスプレイ	
			3487HG 3487-HG ディスプレイ	
			3487HW 3487-HW ディスプレイ	
			APPC 拡張プログラム間通信装置	
			ASYN 非同期通信装置	
			BSC 2 進データ同期通信装置	
			BSC BSC 通信装置	
			FINANC ICF 金融機関用通信装置	
			INTRA システム間通信装置	
			LU1 LU1 通信装置	
			RETAIL 小売業用通信装置	
37	文字	1	SNUF SNA アップライン機能通信装置 要求元装置 このフラグは、この項目が *REQUESTER 装置を定義しているかどうかを示します。	ディスプレイ、 ICF
			N *REQUESTER 装置ではありません (通信元装 置)。	
			Y *REQUESTER 装置です (通信宛先装置)。	

表 29. 属性の取得 (続き)

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
38	文字	1	獲得状況。オープン時に暗黙に獲得されている場合でも設定。 N 装置は獲得されません。 Y 装置は獲得されます。	ディスプレイ、ICF
39	文字	1	送信勧誘状況 Y 装置は送信勧誘されます。 N 装置は送信勧誘されません。	ディスプレイ、ICF
40	文字	1	データ使用可能状況 Y 送信勧誘データは使用可能です。 N 送信勧誘データは使用不能です。	ディスプレイ、ICF
41	2 進数	2	表示画面上の行数	ディスプレイ
43	2 進数	2	表示画面上の桁数	ディスプレイ
45	文字	1	ディスプレイの明滅 Y 表示は明滅することができます。 N 表示は明滅することができません。	ディスプレイ
46	文字	1	オンライン/オフライン状況 O ディスプレイはオンライン状況です。 F ディスプレイはオフライン状況です。	ディスプレイ
47	文字	1	ディスプレイ・ロケーション L ローカル・ディスプレイ R リモート・ディスプレイ	ディスプレイ
48	文字	1	ディスプレイ・タイプ A 英数字またはカタカナ I DBCS G グラフィック DBCS	ディスプレイ
49	文字	1	ディスプレイのキーボード・タイプ A 英数字またはカタカナ・キーボード I DBCS キーボード	ディスプレイ
50	文字	1	トランザクション状況。すべての通信タイプ。 N トランザクションは開始されていません。呼び出し要求がまだ送られていないか、切断要求がすでに送信または受信されたか、あるいはトランザクションがすでに完了していません。 Y トランザクションは開始されています。トランザクションは活動状態です。呼び出し要求が送られるか受け取られ、トランザクションは終了していません。	ICF

表 29. 属性の取得 (続き)

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
51	文字	1	同期レベル APPC および INTRA。 0 同期レベル 0 (SYNLVL(*NONE))。 1 同期レベル 1 (SYNLVL(*CONFIRM))。 2 同期レベル 2 (SYNLVL(*COMMIT))。	ICF
52	文字	1	使用する会話。 APPC のみ。 M マップ式会話 B 基本会話	ICF
53	文字	8	リモート・ロケーション名。すべての通信タイプ。	ICF
61	文字	8	ローカル LU 名。 APPC のみ。	ICF
69	文字	8	ローカル・ネットワーク ID。 APPC のみ。	ICF
77	文字	8	リモート LU 名。 APPC のみ。	ICF
85	文字	8	リモート・ネットワーク ID。 APPC のみ。	ICF
93	文字	8	モード。 APPC のみ。	ICF
101	文字	1	コントローラーの情報 N ディスプレイは、非プログラマブル・ワークステーションの拡張インターフェースをサポートするコントローラーに接続されていません。 1 ディスプレイは、非プログラマブル・ワークステーションの拡張インターフェースをサポートするコントローラー (タイプ 1) に接続されています。注を参照してください。 2 ディスプレイは、非プログラマブル・ワークステーションの拡張インターフェースをサポートするコントローラー (タイプ 2) に接続されています。注を参照してください。 3 ディスプレイは、非プログラマブル・ワークステーションの拡張インターフェースをサポートするコントローラー (タイプ 3) に接続されています。注を参照してください。	ディスプレイ
102	文字	1	ディスプレイのカラー機能 Y カラー表示 N モノクロ表示	ディスプレイ
103	文字	1	ディスプレイの罫線サポート Y ディスプレイは、罫線をサポートします。 N ディスプレイは、罫線をサポートしていません。	ディスプレイ

表 29. 属性の取得 (続き)

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
104	文字	1	16 進 00 リセット状態 16 進 01 送信状態 16 進 02 受信据え置き状態 16 進 03 割り振り解除据え置き状態 16 進 04 受信状態 16 進 05 確認状態 16 進 06 送信確認状態 16 進 07 割り振り解除確認状態 16 進 08 コミット状態 16 進 09 送信コミット状態 16 進 0A 割り振り解除コミット状態 16 進 0B 割り振り解除状態 16 進 0C 必須ロールバック状態	ICF
105	文字	8	LU.6 会話相関子	ICF
113	文字	31	未使用	ディスプレイ、 ICF

注: 次の情報は、ICF セッションまたはリモート表示セッションで統合サービス・デジタル網 (ISDN) を使用する場
合のみ、提供されます。なお、情報の受け入れ域が小さすぎると、一部の情報が利用できないことがあります。

144	2 進数	2	ISDN リモート番号の長さ (バイト数)。 ISDN リモ ート番号タイプ、ISDN リモート番号計画、および ISDN リモート番号の 3 のフィールド・タイプの長さの合計 です。 ISDN リモート番号の右側がブランクで埋めら れていても、その埋め込みはこの合計数には含まれま せん。 ISDN を使わない場合、このフィールドは 0 に なっています。	ディスプレイ、 ICF
-----	------	---	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------

表 29. 属性の取得 (続き)

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
146	文字	2	ISDN リモート番号タイプ (10 進) 00 不明 01 国際 02 国内 03 ネットワーク指定 04 加入者 06 省略	ディスプレイ、 ICF
148	文字	2	ISDN リモート番号計画 (10 進) 00 不明 01 ISDN/電話 03 データ 04 テレックス** 08 国内標準 09 私用	ディスプレイ、 ICF
150	文字	40	ISDN リモート番号 (EBCDIC)。フィールドを満たす必要があるときは、右側がブランクで埋められます。	ディスプレイ、 ICF
190	文字	4	予約済み	ディスプレイ、 ICF
194	2 進数	2	ISDN リモート・サブアドレスの長さ (バイト)。ISDN リモート・サブアドレス・タイプおよび ISDN リモート・サブアドレスの 2 つのフィールドの長さの合計です。ISDN リモート・サブアドレスの右側がブランクで埋め込まれていても、その長さは合計に含まれません。ISDN を使わない場合は、このフィールドは 0 になっています。	ディスプレイ、 ICF
196	文字	2	ISDN リモート・サブアドレス・タイプ (10 進) 00 NSAP 01 ユーザー指定	ディスプレイ、 ICF
198	文字	40	ISDN リモート・サブアドレス (16 進数値の EBCDIC 表現。右側にゼロ埋め込み)	ディスプレイ、 ICF
238	文字	1	予約済み	ディスプレイ、 ICF
239	文字	1	ISDN 結合 0 着信 ISDN 呼び出し 1 発信 ISDN 呼び出し	ディスプレイ、 ICF
240	2 進数	2	その他 非 ISDN 結合 ISDN リモート・ネットワーク・アドレスの長さ (バイト)。ISDN リモート・ネットワーク・アドレスの右側がブランクで埋め込まれていても、その長さは含まれません。 ISDN を使わない場合は、このフィールドは 0 になっています。	ディスプレイ、 ICF

表 29. 属性の取得 (続き)

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
242	文字	32	ISDN リモート・ネットワーク・アドレスの長さ (EBCDIC)。フィールドを満たす必要があるときは、右側がブランクで埋められます。	ディスプレイ、 ICF
274	文字	4	予約済み	ディスプレイ、 ICF
278	文字	2	ISDN リモート・アドレス拡張の長さ (バイト数)。 ISDN リモート・アドレス拡張のタイプおよび ISDN リモート・アドレス拡張の 2 つのフィールドの長さの合計です。 ISDN リモート・アドレス拡張の右側がブランクで埋め込まれていても、その長さは含まれません。	ディスプレイ、 ICF
280	文字	1	ISDN を使わない場合または ISDN リモート・アドレス拡張がない場合、このフィールドは 0 になっています。 ISDN リモート・アドレス拡張タイプ (10 進)。 0 ISO 8348/AD2 に従って割り当てられるアドレス。 2 ISO 8348/AD2 に従って割り当てられないアドレス。 その他 予約済み	ディスプレイ、 ICF
281	文字	40	ISDN リモート・アドレス拡張 (16 進数値の EBCDIC 表現。右側にゼロ埋め込み)	ディスプレイ、 ICF
321	文字	4	予約済み	ディスプレイ、 ICF
325	文字	1	X.25 呼び出しタイプ (10 進)。 0 着信相手選択接続機能 (SVC) 1 発信 SVC 2 非 X.25 SVC その他 予約済み	ディスプレイ、 ICF
注: 次の情報は、受信したプログラム開始要求の結果としてユーザーのプログラムが開始されたときのみ使用できます。なお、情報の受け入れ域が小さすぎると、一部の情報が利用できないことがあります。				
326	文字	64	トランザクション・プログラム名。経路指定リストに開始用の別のプログラムがある場合でも、受信したプログラム開始要求の結果として開始用に指定されるプログラムの名前。	ICF
390	2 進数	1	保護 LUWID フィールドの長さ。有効な値は 0 から 26 の範囲です。	ICF
391	2 進数	1	修飾 LU-NAME の長さ。有効な値は 0 から 17 の範囲です。	ICF
392	文字	17	様式の中のネットワーク修飾保護 LU-NAME (様式は netid.luname)。ネットワーク修飾保護 LU-NAME がない場合、このフィールドはブランクになります。	ICF
409	文字	6	保護 LUWID インスタンス番号。	ICF
415	2 進数	2	保護 LUWID 順序番号。	ICF

表 29. 属性の取得 (続き)

オフセット	データ・タイプ	長さ	内容	ファイル・タイプ
注: 以下の情報は、保護会話がリモート・サーバー上で開始されたときのみ利用可能です。すなわち、会話が *COMMIT の SYNCLVL で開始したときです。なお、情報の受け入れ域が小さすぎると、一部の情報が利用できないことがあります。				
417	2 進数	1	無保護 LUWID フィールドの長さ。有効な値は 0 から 26 の範囲です。	ICF
418	2 進数	1	修飾 LU-NAME の長さ。有効な値は 0 から 17 の範囲です。	ICF
419	文字	17	様式の中のネットワーク修飾無保護 LU-NAME (様式は netid.luname)。ネットワーク修飾無保護 LU-NAME がない場合は、このフィールドはブランクになります。	ICF
436	文字	6	無保護 LUWID インスタンス番号。	ICF
442	2 進数	2	無保護 LUWID 順序番号。	ICF

注:

タイプ 1

V2R2 で利用可能なコントローラー。これらは、ウィンドウ[®]および継続カーソル進行などをサポートします。

タイプ 2

V2R3 で利用可能なコントローラー。これらは V2R2 の機能すべてに加えて、メニュー・バー、継続入力フィールド、編集マスク、および単純ホットスポットをサポートします。


タイプ 3

V3R1 で利用可能なコントローラー。これらは、V2R2 および V2R3 の機能すべてをサポートします。さらに、ウィンドウの下部枠のテキストをサポートします。


データベース・ファイル管理の関連情報

このトピックでは、データベース・ファイル管理についての追加情報を提供する iSeries 関連資料およびトピックをリストします。


計画、インストール、マイグレーション

- i5/OS[™] グローバリゼーションのトピックは、データ処理管理者、システム・オペレーターとシステム管理者、アプリケーション・プログラマー、エンド・ユーザー、IBM 営業担当員、およびシステム・エンジニアに対して、iSeries サーバーの各国語サポート機能を理解し使用するのに必要な情報を提供しています。また、このトピックは iSeries ユーザーが、iSeries 各国語サポート (NLS) および iSeries サーバーの多国語サポート機能の計画、導入、構成、および使用するためのものでもあります。さらに、多国語データのデータベース管理および多国語システム用アプリケーション・プログラムについての考慮事項の説明も提供しています。
- 「装置構成」  は、システム・オペレーターまたはシステム管理者に対して、初期ローカル・ハードウェア構成を行う方法と、その構成の変更方法を説明しています。また、装置構成に関する概念の説明、および 9406、9404、9402 システム装置の装置構成に関する計画情報も扱われています。


アプリケーション開発

- 適用業務開発ツール (ADTS)・セット/400: 文字作成ユーティリティー (CGU) は、アプリケーション・プログラマーまたはシステム・プログラマーに対して、アプリケーション開発ツールの文字作成ユーティリティー (CGU) を使用して、システムで 2 バイト文字セット (DBCS) の作成および維持を行うための情報を提供しています。
- V5R1 補足マニュアル Web サイト  の ADTS for AS/400: 原始ステートメント入力ユーティリティー (SEU) 資料 (英語) を参照してください。この資料は、アプリケーション・プログラマーまたはシステム・プログラマーに対して、アプリケーション開発ツールの原始ステートメント入力ユーティリティー (SEU) を使用して、ソース・メンバーの作成および編集を行うための情報を提供しています。


システム管理

- バックアップおよび回復の手引きは、システム・プログラマーに対して、バックアップと回復の実施方法を計画するために必要な情報を提供しています。また、バックアップと回復の実施方法を実現するための手順、ディスク装置障害からの回復方法、および災害時からの回復方法について説明しています。
- 実行管理機能のトピックは、実行管理環境の作成および変更方法について説明します。
- 「iSeries 機密保護解説書」  は、システム・プログラマーに対して、セキュリティの計画、設計、および監査についての情報を提供しています。また、セキュリティ・システム値、ユーザー・プロフィール、およびリソースのセキュリティについての情報も含まれます。

通信と接続

- ICF Programming  は、アプリケーション・プログラマーに対して、iSeries の通信および ICF ファイルを使用するアプリケーション・プログラムを作成するのに必要な情報を提供しています。また、データ記述仕様 (DDS) のキーワード、システム提供の様式、戻りコード、ファイル転送サポート、およびプログラム例も扱っています。

プログラム・イネーブラー

- DDS concepts は、アプリケーション・プログラマーに対して、(論理および物理) データベース・ファイル、およびユーザー・プログラム外部にある装置ファイル (ディスプレイ用、プリンター用、および ICF 用) を記述するのに必要な項目とキーワードを詳しく説明しています。
- データベース・プログラミングは、アプリケーション・プログラマーまたはシステム・プログラマーに対して、iSeries データベース編成の詳細を提供しています。説明には、システムでデータベース・ファイルの作成、記述、および操作を行う方法などが含まれます。
- Application Display Programming  は、アプリケーション・プログラムの画面の作成および維持、オンライン・ヘルプ情報の作成、および iSeries サーバー上のディスプレイ・ファイルの処理についての情報を提供しています。
- 印刷は、プリンターを理解し制御するための情報を提供しています。印刷エレメントと概念、プリンター・ファイル・サポート、印刷スプーリング・サポート、プリンター接続、高機能印刷、およびパーソナル・コンピューターでの印刷などの内容が含まれています。
- テープ・ファイルでは、磁気テープ装置ファイルの作成と操作に関する情報を提供しています。
- CL プログラミングのトピックには、オブジェクトおよびライブラリー、制御言語 (CL) プログラミング、フロー制御およびプログラム間通信、CL プログラムのオブジェクト処理、および CL プログラム作成に関する一般的な説明を含む、プログラミング上のトピックについての広範囲の説明が記載されています。他のトピックには、事前定義メッセージと即時メッセージ、メッセージ処理、ユーザー定義の

コマンドおよびメニューの定義と作成、およびアプリケーションのテスト (デバッグ・モード、ブレイクポイント、トレース、および表示機能を含む) が含まれます。

- iSeries Information Center 内の制御言語 (CL) のトピックには、制御言語 (CL) とそのコマンドに関する説明があります。各コマンドが定義され、構文図、パラメーター、デフォルト値、およびキーワードが含まれています。

システム管理

- 分散データ管理は、アプリケーション・プログラマーまたはシステム・プログラマーに対して、リモート・ファイル処理についての情報を提供しています。また、i5/OS の分散データ管理機能 (DDM) に対するリモート・ファイルの定義方法、DDM ファイルの作成方法、DDM を介してサポートされるファイル・ユーティリティー、および他のシステムとの関連における i5/OS の DDM の要件について説明しています。

コードに関する特記事項

IBM は、お客様に、すべてのプログラム・コードのサンプルを使用することができる非独占的な著作使用权を許諾します。お客様は、このサンプル・コードから、お客様独自の特別のニーズに合わせた類似のプログラムを作成することができます。

強行法規で除外を禁止されている場合を除き、IBM、そのプログラム開発者、および供給者は「プログラム」および「プログラム」に対する技術的サポートがある場合にはその技術的サポートについて、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。

IBM、そのプログラム開発者、または供給者は、いかなる場合においてもその予見の有無を問わず、以下に対する責任を負いません。

1. データの喪失、または損傷。
2. 直接損害、特別損害、付随的損害、間接損害、または経済上の結果的損害
3. 逸失した利益、ビジネス上の収益、あるいは節約すべかりし費用

国または地域によっては、法律の強行規定により、上記の責任の制限が適用されない場合があります。

付録. 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒106-0032
東京都港区六本木 3-2-31
IBM World Trade Asia Corporation
Licensing

以下の保証は、国または地域の法律に沿わない場合は、適用されません。 IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとしします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation
Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

- 1 本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム
- 1 契約の契約条項、IBM プログラムのご使用条件、IBM 機械コードのご使用条件、またはそれと同等の条項
- 1 に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。 © Copyright IBM Corp. _年を入れる_. All rights reserved.

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

プログラミング・インターフェース情報

本書「データベース・ファイル管理」には、プログラムを作成するユーザーが IBM i5/OS のサービスを使用するためのプログラミング・インターフェースが記述されています。

商標

以下は、IBM Corporation の商標です。

- | AS/400
- | DB2
- | DB2 Universal Database
- | i5/OS
- | IBM
- | IBM (ロゴ)
- | InfoWindow
- | Integrated Language Environment
- | IPDS
- | iSeries
- | System/370
- | Systems Application Architecture

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

使用条件

これらの資料は、以下の条件に同意していただける場合に限りご使用いただけます。

個人使用: これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、非商業的な個人による使用目的に限り複製することができます。ただし、IBM の明示的な承諾をえずに、これらの資料またはその一部について、二次的著作物を作成したり、配布 (頒布、送信を含む) または表示 (上映を含む) することはできません。

商業的使用: これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、お客様の企業内に限り、複製、配布、および表示することができます。ただし、IBM の明示的な承諾をえずにこれらの資料の二次的著作物を作成したり、お客様の企業外で資料またはその一部を複製、配布、または表示することはできません。

ここで明示的に許可されているもの以外に、資料や資料内に含まれる情報、データ、ソフトウェア、またはその他の知的所有権に対するいかなる許可、ライセンス、または権利を明示的にも黙示的にも付与するものではありません。

資料の使用が IBM の利益を損なうと判断された場合や、上記の条件が適切に守られていないと判断された場合、IBM はいつでも自らの判断により、ここで与えた許可を撤回できるものとさせていただきます。

お客様がこの情報をダウンロード、輸出、または再輸出する際には、米国のすべての輸出入関連法規を含む、すべての関連法規を遵守するものとします。

IBM は、これらの資料の内容についていかなる保証もしません。これらの資料は、特定物として現存するままの状態を提供され、第三者の権利の不侵害の保証、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任なしで提供されます。



Printed in Japan