



System i

プログラミング
光ディスク装置プログラミング

バージョン 6 リリース 1





System i

プログラミング
光ディスク装置プログラミング

バージョン 6 リリース 1

お願い

本書および本書で紹介する製品をご使用になる前に、53 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 (プロダクト番号 5761-SS1) のバージョン 6、リリース 1、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。

IBM 発行のマニュアルに関する情報のページ

<http://www.ibm.com/jp/manuals/>

こちらから、日本語版および英語版のオンライン・ライブラリーをご利用いただけます。また、マニュアルに関するご意見やご感想を、上記ページよりお送りください。今後の参考にさせていただきます。

(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： System i
Programming
Optical device programming
Version 6 Release 1

発 行： 日本アイ・ビー・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 2008.2

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 2006, 2008. All rights reserved.

© Copyright IBM Japan 2008

目次

光ディスク装置プログラミング	1
光ディスク装置プログラミングの PDF ファイル . . .	1
光ディスク装置プログラミングの概念	2
統合ファイル・システムと光ディスク装置のプログラ ミング	2
階層ファイル・システムと光ディスク装置のプログラ ミング	3
ボリューム、ディレクトリー、およびファイルの考 慮事項	4
i5/OS の場合の統合ファイル・システムのプログラミ ング	6
統合ファイル・システムの API	6
統合ファイル・システムの総称コマンド	14
例: 統合ファイル・システム	16
階層ファイル・システムのプログラミング	19
階層ファイル・システム API	20
ファイル・システム制御関数	25
標準属性	30
拡張属性	31
階層ファイル・システムを使用したファイル属性 のコピー	33
例: 光ディスク・ファイル・システムの場合の階 層ファイル・システム API のプログラミング	34
ヒント: 光ディスク装置プログラミング	36
メディア容量およびボリュームしきい値	37
ファイル単位でのメディア容量の管理	37
拡張バッファ入出力方式	38
「バッファ・データの強制」API	39
保留光ディスク・ファイルの管理	40
パス名の要件	40
例: 光ディスク記憶との中のスプール・ファイル の移動	41
光ディスク装置プログラミングの関連情報	51
付録. 特記事項	53
プログラミング・インターフェース情報	54
商標	54
使用条件	55

光ディスク装置プログラミング

一部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) は、i5/OS® 光ディスク・ファイル・システムとともに働きます。i5/OS 光ディスク・ファイル・システムは、CD-ROM、デジタル・ビデオ・ディスク (DVD)、Write Once Read Many (WORM) メディア、および光磁気メディアなどの光ディスク・メディアを使用する任意のデータ・ストレージ・システムから構成されます。

プログラマーは、これらの API を使用して、光ディスク・ファイルおよびディレクトリーの作成、アクセス、変更、または保守を行うことができます。これらの API を使用して、特定のビジネス・アプリケーション向けに光ディスク・サポートの使用法をカスタマイズすることもできます。

注: コーディング例をご使用になることで、51 ページの『コードに関するライセンス情報および特記事項』の条件に同意されたものとみなされます。

関連概念

アプリケーション・プログラミング・インターフェース

光ディスク装置プログラミングの PDF ファイル

本書の PDF ファイルを表示および印刷することができます。


本書の PDF 版を表示またはダウンロードするには、Optical device programming (光ディスク装置プログラミング) を選択してください。

PDF ファイルの保管

表示または印刷の目的でワークステーションに PDF を保管するには、以下のようになります。

1. ご使用のブラウザの PDF リンクを右クリックする。
2. PDF をローカルに保管するオプションをクリックする。
3. PDF を保管するディレクトリーに進む。
4. 「保存」をクリックする。

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ことが必要です。Adobe Web サイト (www.adobe.com/products/acrobat/readstep.html)  から、無償コピーをダウンロードできます。

関連資料

51 ページの『光ディスク装置プログラミングの関連情報』

Web サイトおよびその他のインフォメーション・センターのトピック・コレクションに、光ディスク装置プログラミングのトピックに関連する情報が含まれています。任意の PDF ファイルを表示または印刷することができます。

光ディスク装置プログラミングの概念

統合ファイル・システム・サポートまたは階層ファイル・システム (HFS) サポートを使用して、光ディスク・ファイルおよびディレクトリーを操作することができます。ボリューム、ディレクトリー、およびファイルについての考慮事項を理解すれば、光ディスク装置のプログラミングは容易になります。

以下の 2 つのカテゴリの API を使用して、光ディスク・ファイルおよびディレクトリーを操作することができます。

- 統合ファイル・システム・サポート。UNIX タイプ API および汎用コマンド・インターフェースにより構成されます。
- 階層ファイル・システム (HFS) サポート。API および汎用コマンドより構成されます。

並行して両カテゴリの API を使用できます。例えば、あるアプリケーションによって「HFS Open Stream File」API を使用して読み取りの目的で開かれている光ディスク・ファイルを、別のアプリケーションによって UNIX タイプ API のオープンを使用して読み取りの目的で開くことができます。

i5/OS 内には種々のファイル・システムが存在するため、どのファイル・システムをターゲットに呼び出しを行っているのかを HFS または統合ファイル・システムが区別するための何らかの手段を用意する必要があります。この都合から、呼び出すファイル・システムの名前はパス名パラメーターの最初に置き、頭にスラッシュを付けるよう定めています。HFS または統合ファイル・システムに処理依頼される要求のレシーバーとして光ディスク・ファイル・システムが識別されるように、パス名パラメーターの最初の部分は /QOPT でなければなりません。/QOPT に続く残りのパス・エレメントは、ボリューム/ディレクトリー/サブディレクトリー/ファイルです。パス名の例を次に示します。

/QOPT/CD001/Dir1/SubDir1/File

関連概念

アプリケーション・プログラミング・インターフェース

統合ファイル・システムと光ディスク装置のプログラミング

統合ファイル・システムは i5/OS オペレーティング・システムの一部であり、パーソナル・コンピューターや UNIX[®] オペレーティング・システムと同様にストリームの入出力およびストレージ管理をサポートすると同時に、システムに保管される情報全体について統合化構造を提供します。

統合ファイル・システムは 11 個のファイル・システムから構成され、ストレージ内の情報とのやりとりに関してそれぞれに独自の論理構造と規則のセットがあります。統合ファイル・システムの主要な機能には以下の項目があります。

- データの中の長い連続したストリングの収容が可能なストリーム・ファイルに情報を保管するためのサポート。データの中のこれらのストリングには、例えば、ドキュメントのテキスト、あるいは画像中の画素があります。ストリーム・ファイル・サポートは、クライアント/サーバー・アプリケーションでの効果的な使用を目的に設計されています。
- 共通 PC ファイル構造と同様にオブジェクトを編成することが可能な階層ディレクトリー構造。あるオブジェクトまでディレクトリーを介して指定されるパスにより、そのオブジェクトがアクセスされます。
- ユーザーおよびアプリケーションが、ストリーム・ファイルだけでなく、システムに保管されているデータベース・ファイル、文書、および他のオブジェクトにアクセスできる共通インターフェース。
- ご使用のシステム、統合 System x[™] または IBM[®] BladeCenter[®] サーバー、またはリモート Windows[®] サーバーにローカルに保管されるストリーム・ファイルの共通ビュー。ストリーム・ファイルは、ロー

カル・エリア・ネットワーク (LAN) サーバー、Novell NetWare サーバー、i5/OS オペレーティング・システムが稼働する他のリモート・システム、またはネットワーク・ファイル・システム・サーバーにリモートで保管することもできます。

統合ファイル・システムは、新たに出現する将来の情報処理形式 (クライアント/サーバー、オープン・システム、マルチメディアなど) をより適切にサポートするための追加機能により、i5/OS の既にある広範囲のデータ管理機能を拡張します。

統合ファイル・システムにより、以下のタスクの実行が可能になります。

- i5/OS データへの高速アクセスを達成します。特に、i5/OS ファイル・サーバーを使用するアプリケーション (IBM System i™ Access など) の場合。
- 画像、オーディオ、ビデオなどのタイプのストリーム・データをより効率的に処理します。
- POSIX や XPG などの UNIX ベースのオープン・システム規格のサポート用に、ファイル・システム・ベースおよびディレクトリー・ベースを使用します。また、このファイル構造およびディレクトリー構造は、ディスク・オペレーティング・システム (DOS) や Windows オペレーティング・システムなどの PC オペレーティング・システムのユーザーに対して親しみのある環境を提供します。
- 共通インターフェースを介してファイル・システム全体の管理が可能になると同時に、個別のファイル・システムとして処理される固有の機能 (レコード指向データベース・ファイル、UNIX ベースのストリーム・ファイルおよびファイル処理など) を使用してファイル・サポートに対するアクセスが得られます。

LAN 接続の光ディスク装置は、このインターフェースをサポートしていません。

関連概念

統合ファイル・システム

階層ファイル・システムと光ディスク装置のプログラミング

階層ファイル・システム (HFS) 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・プログラミング・インターフェース (API) および基礎となるファイル・システム (光ディスクまたはその他) サポートが含まれています。

HFS API により、直接接続の光ディスク・ライブラリー装置、LAN 接続の光ディスク・ライブラリー装置、CD-ROM、または DVD 装置上でデータを作成、保管、検索、および操作を行うアプリケーションを高水準言語で書くことが可能になります。

HFS API 光ディスク・サポートは次の 2 つの部分からなります。

- ファイルおよびディレクトリーとして知られるオブジェクトを処理するための、階層ファイル・システムへのアプリケーション・プログラミング呼び出しインターフェース。
- ファイルおよびディレクトリーが保管されるストレージ・デバイスを管理する、光ディスクまたはその他の登録済みファイル・システム。

HFS API 光ディスク機能には以下の項目があります。

- ディレクトリーの作成および削除
- ディレクトリーのオープン、読み取り、またはクローズ
- ファイルのオープン、読み取り、書き込み、またはクローズ
- ファイル内のバイトのロックまたはアンロック
- ファイル・サイズの取得または設定

- ファイルの名前変更、コピー、削除、または移動
- ディレクトリー項目属性の検索または変更

アプリケーションは、HFS API を使用して i5/OS システム上のストリーム・ファイル进行管理します。これらのストリーム・ファイルは、在来型の記録構造を持たないデータ・エレメントとして識別するために、オブジェクトと呼ばれることもあります。オブジェクトは既知の長さを持つ名前付きバイト・ストリームとして扱われ、そのサイズは数バイトから数メガバイトまでさまざまに異なります。

HFS API により、アプリケーションは、ストレージ・デバイス上のファイル・オブジェクトの作成と管理を行い、それらのファイル・オブジェクトに対する入出力操作を行うことができます。HFS API により、アプリケーションは、ディレクトリー・オブジェクトの作成と管理を行うことができます。ディレクトリー・オブジェクトは、類似のファイル・オブジェクトの論理的なグループ化と見なすことができます。これらのディレクトリー・オブジェクトには、そのディレクトリーに属するファイル・オブジェクトに関する情報が含まれています。ディレクトリーをディレクトリー内に含めることも可能で、結果として階層構造になります。

関連概念

アプリケーション・プログラミング・インターフェース

ボリューム、ディレクトリー、およびファイルの考慮事項

ボリューム、ディレクトリー、およびファイルについての以下の考慮事項を理解すれば、光ディスク装置のプログラミングは容易になります。

ボリュームの考慮事項

ボリュームについて言及している場合は、以下の用語に注意してください。

オンライン

ボリュームは、ドライブ内で読み取り/書き込みヘッドの下にマウントされています。

ニア・オンライン

ボリュームは、光ディスク・メディア・ライブラリー内にありますが、オンラインではありません。ボリュームは、ストレージ・スロット内またはオンライン・ボリュームの裏面にある可能性があります。

取り外し

ボリュームは、物理的には光ディスク・メディア・ライブラリー内にはありませんが、ボリュームが取り外されるときにそのボリュームに関するボリューム情報が保持されます。

オフライン

ボリュームは光ディスク装置内にありますが、その装置が電源オフか、オフに構成変更されたか、またはもう接続されていないかのいずれかです。

以下の光ディスク・ボリュームの特性を考慮してください。

- 光ディスク・ボリュームは、光ディスク・カートリッジの 1 つの面です。
- 光ディスク・カートリッジには、2 つのボリュームを持つものと、1 つのボリュームしか持たないものがあります。
- ボリューム名はすべて固有でなければなりません。
- 光ディスク・メディアの密度とタイプによって、ボリューム容量は数百メガバイトから数ギガバイトまでの範囲が可能です。

- 通常、ニア・オンライン・ボリュームは、10 秒未満でオンライン・ボリュームになります。これには、ボリュームをドライブにマウント済みであることが必要です。
- 光ディスク・メディア・ライブラリー内のドライブ数により、いつでもオンラインにできるボリューム数が決まります。ドライブにマウント (オンラインに) できるのは、一度に 1 つのボリュームだけです。ライブラリー内の残りのボリュームは、ニア・オンラインです。
- ボリュームは通常、互いに独立していますが、例外が 1 つあります。同じカートリッジ上にある 2 つのボリュームは、完全には独立させることができません。カートリッジ上の両ボリュームは、同時にはオンラインにできません。同じカートリッジ上の 2 つのボリューム間でコピーすることはできますが、要求されたファイルをすべてコピーするためには、カートリッジを何回か「裏返す」必要があります。
- 存在可能な除去ボリューム数には制限がありません。

アプリケーションがボリュームを管理する方法は、アプリケーションの要件によってほぼ完全に決まります。将来要求される取り出し時間に従い、データを戦略的にボリュームに書き込む必要があります。ニア・オンラインのボリュームがオンラインになるのを待機するのが望ましくない場合、アクセス対象になる可能性の最も高いボリュームがオンラインになるようにアプリケーションをセットアップする必要があります。

ディレクトリーの考慮事項

ボリューム上に作成可能なディレクトリー数に対する唯一の制限は、メディアの容量です。この制限は、光ディスク・ディレクトリー内に存在可能なファイル数にも適用されます。ボリュームに保管されるファイルに応じてディレクトリーが存在する必要はありません。必要であれば、すべてのファイルをボリュームのルート・ディレクトリーに保管することができます。ルート・ディレクトリーは、ボリュームが初期化されるときに作成される "/" ディレクトリーです。このルートは、他のディレクトリーのように作成も削除もできないため、従来の意味ではディレクトリーと見なされません。ルート・ディレクトリーは、初期化された光ディスク・ボリューム上に常に存在します。

ディレクトリーは、光ディスク・ファイルをより管理しやすいサブセットに分類するのに使用できます。ディレクトリーは、特定の期間、主題、特性、あるいはこれらの任意の組み合わせに基づいてファイルを収容することができます。例えば、ディレクトリー SPOOLFILES にサブディレクトリー YEAR_1994 および YEAR_1995 を従属させることができます。これをもう 1 ステップ進めると、これらのサブディレクトリー内に MONTH_MARCH および MONTH_APRIL という名前でサブディレクトリーを作成することができます。この構造の例を次に示します。

```
/SPOOLFILES /YEAR_1994 /MONTH_MARCH
86 Optical Support V5R3
|
/MONTH_APRIL /YEAR_1995 /MONTH_MARCH /MONTH_APRIL
```

次の例には、完全修飾ディレクトリー名が含まれています。

```
/SPOOLFILES
/SPOOLFILES/YEAR_1994
/SPOOLFILES/YEAR_1994/MONTH_MARCH
/SPOOLFILES/YEAR_1994/MONTH_APRIL
/SPOOLFILES/YEAR_1995
/SPOOLFILES/YEAR_1995/MONTH_MARCH
/SPOOLFILES/YEAR_1995/MONTH_APRIL
```

ディレクトリーはファイルを分類する場合に便利ですが、必須ではありません。ボリューム名と同様に、ディレクトリー名は同一ボリューム内では固有でなければなりません。例えば、ボリューム VOL001 は、DIR001 という名前のディレクトリーを 2 つ持つことはできません。ただし、ボリューム VOL001 は、DIR001 ディレクトリーと DIR000/DIR001 ディレクトリーを持つことが可能です。また、DIR001 ディレクトリーがボリューム VOL001 とボリューム VOL002 上に存在することも可能です。ディレクトリーの

命名規則については、40 ページの『パス名の要件』を参照してください。

ファイルの考慮事項

光ディスク・ファイルのサイズは、アプリケーションおよびそれらのファイルのユーザーの要件によってほぼ完全に決まります。光ディスク・ファイル (HFS または統合ファイル・システムを介してアクセス可能) のサイズは、ボリュームの容量に応じて 0 バイトから 4 294 705 152 バイトまでの範囲が可能です。メディアのターゲット部分の物理サイズは、使用可能なフリー・スペース量によって制限されます。

アプリケーションにとって最適なファイル・サイズを選択する際は、以下の考慮事項に特別な注意を払ってください。

- システム上のシステム・ディスク装置または主記憶装置の量
- データの読み取り方法 (順次かランダムか)
- 通常、ファイル全体を検索するか、ごく一部のみ検索するか
- ファイルは、ボリュームに書き込まれた後で更新されるかどうか

一般に、ファイルが大きくなるほど、パフォーマンスおよびメディア使用効率がよくなります。大きなファイルが使用されるほど、ファイルのディレクトリ情報に取られるメディア・スペースは少なくなり、実データで使用されるメディア・スペースが増えます。また、ファイル・サイズとパフォーマンスの関連は、線形の比較ではありません。20 KB のデータの書き込み時間は、10 KB のデータの書き込みの 2 倍かかるわけではありません。パフォーマンス (KB/秒) は、読み取りまたは書き込みが行われるデータ量が増えるにつれて向上します。

i5/OS の場合の統合ファイル・システムのプログラミング

統合ファイル・システム・サポートでは、光ディスク・ファイルおよびディレクトリを保守するのに使用できる UNIX タイプ・インターフェースを提供しています。LAN 接続の光ディスク装置は、このインターフェースをサポートしていません。

光ディスク・サポート用の統合ファイル・システム・サポートは、UNIX タイプ API および汎用コマンドにより構成されます。

すべてのファイル・システムと同様に、光ディスク・ファイル・システムには、統合ファイル・システムを介して光ディスク機能にアクセスするアプリケーションに対する固有の規則と制約事項があります。UNIX タイプ API と総称コマンドのいくつかはサポートされていません。その他は、部分的にしかサポートされないか、または制限が付きます。

統合ファイル・システムの API

UNIX タイプ API は、Integrated Language Environment® (ILE) C for i5/OS プログラムで使用可能な C 言語関数です。

次の表は、UNIX タイプの光ディスク・ファイル・システム API (サポートされるものとサポートされないものの両方) のクイック・リファレンスです。

表 1. 光ディスクにおける UNIX タイプ API のインプリメンテーション

UNIX タイプ API	サポート	注記および使用上の注意
--------------	------	-------------

表1. 光ディスクにおける UNIX タイプ API のインプリメンテーション (続き)

<p>access (ファイル・アクセス可能性の判別)</p>	<p>あり</p>	<p>親の光ディスク・ボリュームに対する *X 権限が必要です。非 Universal Disk Format (UDF) ボリュームの場合、他の権限は不要です。UDF フォーマットのボリュームの場合、以下の許可規則が適用されます。</p> <ul style="list-style-type: none"> • 検査対象のオブジェクトに先行するパス内の各ディレクトリーに対する *X 権限が必要。 • R_OK が指定された場合、*R 権限が必要。 • W_OK が指定された場合、*W 権限が必要。 • X_OK が指定された場合、*X 権限が必要。 • R_OKIX_OK が指定された場合、*RX 権限が必要。 • W_OKIX_OK が指定された場合、*WX 権限が必要。 • R_OKIW_OK が指定された場合、*RX 権限が必要。 • F_OK が指定された場合、権限は不要。
<p>accessx (who パラメーターに基づくファイル・アクセス可能度の判別)</p>	<p>あり</p>	<p>親の光ディスク・ボリュームに対する *X 権限は不要です。UDF ボリュームの場合、以下の許可規則が適用されます。</p> <p>who パラメーターに有効な値は、以下のとおりです。</p> <ul style="list-style-type: none"> • ACC_INVOKER • ACC_SELF • ACC_ALL • ACC_OTHERS <ol style="list-style-type: none"> 1. R_OK が指定された場合、*R 権限が必要。 2. W_OK が指定された場合、*W 権限が必要。 3. X_OK が指定された場合、*X 権限が必要。 <p>権限検査は互いに排他的です。</p>

表1. 光ディスクにおける UNIX タイプ API のインプリメンテーション (続き)

chdir (現行ディレクトリーの変更)	あり	親の光ディスク・ボリュームに対する *X 権限が必要です。 非 UDF ボリュームの場合、他の権限は不要です。 UDF フォーマットのボリュームの場合、パス内の各ディレクトリーに対する *X 権限が必要です。
chmod (ファイル許可の変更)	あり	UDF フォーマットの光ディスク・ボリュームの場合のみサポートされます。親の光ディスク・ボリュームに対する *CHANGE 権限が必要です。オブジェクトに先行するパス内の各ディレクトリーに対する *X 権限が必要です。この操作を実行するには、ファイルの所有者であるか、または *ALLOBJ 特殊権限を持つ必要があります。
chown (ファイルの所有者およびグループの変更)	あり	UDF フォーマットの光ディスク・ボリュームの場合のみサポートされます。親の光ディスク・ボリュームに対する *CHANGE 権限が必要です。オブジェクトに先行するパス内の各ディレクトリーに対する *X 権限が必要です。この操作を実行するには、ファイルの所有者であるか、または *ALLOBJ 特殊権限を持つ必要があります。非 UDF フォーマットのボリューム上のファイルおよびディレクトリーは、QDFTOWN ユーザー・プロファイルによって所有されます。
close (ファイル記述子のクローズ)	あり	
closedir (ディレクトリーのクローズ)	あり	
creat (ファイルの作成または再書き込み)	あり	親の光ディスク・ボリュームに対する *CHANGE 権限が必要です。非 UDF ボリュームの場合、他の権限は不要です。UDF フォーマットのボリュームの場合、パス内の各ディレクトリーに対する *X 権限、および親ディレクトリーに対する *WX 権限が必要です。 親ディレクトリーの変更および修正タイム・スタンプは更新されません。
dup (オープン・ファイル記述子の複写)	あり	

表1. 光ディスクにおける UNIX タイプ API のインプリメンテーション (続き)

dup2 (オープン・ファイル記述子の他の記述子への複写)	あり	
fchmod (記述子によるファイル許可の変更)	あり	UDF フォーマットの光ディスク・ボリュームの場合のみサポートされます。この操作を実行するには、ファイルの所有者であるか、または *ALLOBJ 特殊権限を持つ必要があります。
fchown (記述子によるファイルの所有者およびグループの変更)	あり	UDF フォーマットの光ディスク・ボリュームの場合のみサポートされます。この操作を実行するには、ファイルの所有者であるか、または *ALLOBJ 特殊権限を持つ必要があります。非 UDF フォーマットのボリューム上のファイルおよびディレクトリーは、QDFTOWN ユーザー・プロファイルによって所有されます。
fcntl (ファイル制御コマンドの実行)	なし	
fpathconf (記述子による構成可能パス名変数の取得)	あり	
fstat (記述子によるファイル情報の取得)	あり	ファイルに対するユーザーの権限にかかわらず、所有者、グループ、およびその他のモード・ビットは常にオンです。 ファイル・アクセスのタイム・スタンプは変更されません。
fsync (ファイルへの変更の同期化)	あり	UDF フォーマットのボリュームの場合、データは光ディスクに書き込まれます。非 UDF フォーマットのボリュームの場合、データは、保留光ディスク・ファイルを介して回復可能な内部ディスク・ストレージに書き込まれます。
ftruncate (ファイルの切り捨て)	あり	
getcwd (現行ディレクトリーの取得)	あり	親の光ディスク・ボリュームに対する *X 権限が必要です。非 UDF ボリュームの場合、他の権限は不要です。UDF フォーマットのボリュームの場合、オブジェクトに先行するパス名内の各ディレクトリーに対する *RX 権限が必要です。
getegid	あり	
geteuid	あり	
getgid	あり	
getgrid	あり	

表 1. 光ディスクにおける UNIX タイプ API のインプリメンテーション (続き)

getgrnam	あり	
getgroups	あり	
getpwnam	あり	
getpwuid	あり	
getuid	あり	
ioctl (ファイル入出力制御要求の 実行)	なし	
link (ファイルへのリンクの作成)	なし	QOPT はリンクをサポートしません。
lseek (ファイル読み取り/書き込み オフセットの取得)	あり	
lstat (ファイルまたはリンク情報の 取得)	あり	<p>ファイル・アクセスのタイム・スタンプは変更されません。</p> <p>親の光ディスク・ボリュームに対する *X 権限が必要です。非 UDF ボリュームの場合、他の権限は不要です。UDF フォーマットのボリュームの場合、オブジェクトに先行するパス内の各ディレクトリーに対する *X 権限、およびオブジェクトに対する *R 権限が必要です。</p>
mkdir (ディレクトリーの作成)	あり	<p>親の光ディスク・ボリュームに対する *CHANGE 権限が必要です。非 UDF ボリュームの場合、他の権限は不要です。UDF フォーマットのボリュームの場合、パス内の各ディレクトリーに対する *X 権限、および親ディレクトリーに対する *WX 権限が必要です。</p> <p>親ディレクトリーの変更および修正タイム・スタンプは更新されません。</p> <p>所有者 ID およびグループ ID は設定されません。</p>

表1. 光ディスクにおける UNIX タイプ API のインプリメンテーション (続き)

open (ファイルのオープン)	あり	<p>書き込みアクセス用にファイルがオープンされる場合、親の光ディスク・ボリュームに対する *CHANGE 権限が必要です。</p> <p>読み取りアクセス用にファイルがオープンされる場合、親の光ディスク・ボリュームに対する *USE 権限が必要です。</p> <p>UDF フォーマットのボリュームの場合、さらに以下の許可規則が適用されます。</p> <ul style="list-style-type: none"> • オブジェクトが O_RDONLY でオープンされる場合、*R 権限が必要。 • オブジェクトが O_WRONLY でオープンされる場合、*W 権限が必要。 • オブジェクトが O_RDWR でオープンされる場合、*RW 権限が必要。 • オブジェクトが存在せず、O_CREAT が指定された場合、親ディレクトリーに対する *WX が必要。
opendir (ディレクトリーのオープン)	あり	<p>親の光ディスク・ボリュームに対する *USE 権限が必要です。</p> <p>UDF フォーマットのボリュームの場合、オブジェクトに先行するパス内の各ディレクトリーに対する *X 権限、およびオープンされるオブジェクトに対する *R 権限が必要です。</p>
pathconf (構成パス名変数の取得)	あり	
Qp0lGetPathFromFileId	あり	

表1. 光ディスクにおける UNIX タイプ API のインプリメンテーション (続き)

Qp0lRenameKeep	一部あり	<p>QOPT はディレクトリーの名前変更をサポートしません。オブジェクトはファイルでなければなりません。</p> <p>親の光ディスク・ボリュームに対する *CHANGE 権限が必要です。非 UDF ボリュームの場合、他の権限は不要です。UDF フォーマットのボリュームの場合、パス内の各ディレクトリーに対する *X 権限、親ディレクトリーに対する *WX 権限、およびファイルに対する *W 権限が必要です。ボリュームを名前変更する場合、ボリュームのルート (/) ディレクトリーに対する *RWX がが必要です。</p> <p>新旧のファイルは、同じディレクトリー内になければなりません。</p>
Qp0lRenameUnLink	一部あり	<p>QOPT はディレクトリーの名前変更をサポートしません。オブジェクトはファイルでなければなりません。</p> <p>親の光ディスク・ボリュームに対する *CHANGE 権限が必要です。非 UDF ボリュームの場合、他の権限は不要です。UDF フォーマットのボリュームの場合、パス内の各ディレクトリーに対する *X 権限、親ディレクトリーに対する *WX 権限、およびファイルに対する *W 権限が必要です。ボリュームを名前変更する場合、ボリュームのルート (/) ディレクトリーに対する *RWX がが必要です。</p> <p>新規のパスによって識別されるオブジェクトが存在してはなりません。</p>
read (ファイルからの読み取り)	あり	<p>ファイル・アクセス時刻は更新されません。</p> <p>Universal Disk Format (UDF) でフォーマット設定されているボリューム上のファイルからの読み取りの場合、読み取られる範囲にあるバイト・ロックは無視されます。同じことが readv() の場合にも当てはまります。</p>
readdir (ディレクトリー項目の読み取り)	あり	<p>ディレクトリー・アクセス時刻は更新されません。</p>
readlink (シンボリック・リンクの値の読み取り)	なし	<p>QOPT にはシンボリック・リンクがありません。</p>

表1. 光ディスクにおける UNIX タイプ API のインプリメンテーション (続き)

<p>rename (ファイルまたはディレクトリーの名前変更)</p>	<p>一部あり</p>	<p>QOPT はディレクトリーの名前変更をサポートしません。オブジェクトはファイルまたはボリュームでなければなりません。</p> <p>親の光ディスク・ボリュームに対する *CHANGE 権限が必要です。非 UDF ボリュームの場合、他の権限は不要です。UDF フォーマットのボリュームの場合、パス内の各ディレクトリーに対する *X 権限、親ディレクトリーに対する *WX 権限、およびファイルに対する *W 権限が必要です。ボリュームを名前変更する場合、ボリュームのルート (/) ディレクトリーに対する *RWX がが必要です。</p> <p>新規のパスによって識別されるオブジェクトが存在してはなりません。</p>
<p>rewinddir</p>	<p>あり</p>	
<p>rmdir (ディレクトリーの除去)</p>	<p>あり</p>	<p>親の光ディスク・ボリュームに対する *CHANGE 権限が必要です。非 UDF ボリュームの場合、他の権限は不要です。UDF フォーマットのボリュームの場合、パス内の各ディレクトリーに対する *X 権限、および親ディレクトリーに対する *WX 権限が必要です。</p> <p>親ディレクトリーの変更および修正タイム・スタンプは更新されません。</p> <p>ディレクトリーが使用中の場合は、操作は許可されません。</p>
<p>stat (ファイル情報の取得)</p>	<p>あり</p>	<p>ファイル・アクセスのタイム・スタンプは変更されません。</p> <p>親の光ディスク・ボリュームに対する *X 権限が必要です。非 UDF ボリュームの場合、他の権限は不要です。UDF フォーマットのボリュームの場合、オブジェクトに先行するパス内の各ディレクトリーに対する *X 権限、およびオブジェクトに対する *R 権限が必要です。光ディスク・ボリュームに対して発行された場合、戻されるサイズは、ボリューム容量または 2 147 483 647 のいずれか小さい方です。</p>

表1. 光ディスクにおける UNIX タイプ API のインプリメンテーション (続き)

symlink (シンボリック・リンクの作成)	なし	QOPT はシンボリック・リンクをサポートしません。
sysconf (システム構成変数の取得)	あり	
unlink (ファイルへのリンクの除去)	あり	親の光ディスク・ボリュームに対する *CHANGE 権限が必要です。非 UDF ボリュームの場合、他の権限は不要です。UDF フォーマットのボリュームの場合、パス内の各ディレクトリーに対する *X 権限、および親ディレクトリーに対する *RX 権限が必要です。 親ディレクトリーの変更および修正タイム・スタンプは更新されません。 ジョブでファイルをオープンしている場合は、ファイルへのリンクを除去できません。
unmask (ジョブへの権限マスクの設定)	あり	
utime (ファイルのアクセスおよび変更の時刻の設定)	なし	QOPT は、ファイルのアクセスおよび変更の時刻の設定をサポートしません。
write (ファイルへの書き込み)	あり	ファイルの変更および修正タイム・スタンプは、ファイルがクローズされるときに更新されます。Universal Disk Format (UDF) でフォーマット設定されているボリューム上のファイルに書き込む場合、書き込まれる範囲にあるバイト・ロックは無視されます。同じことが writev() の場合にも当てはまります。

統合ファイル・システムの総称コマンド

総称コマンドは、光ディスク・サポートへのインターフェースを提供するシステム制御言語 (CL) コマンドです。

表2 は、統合ファイル・システムに関連する総称 CL コマンドのクイック・リファレンスです。

総称コマンドを発行するために必要な権限については、「機密保護解説書」を参照してください。

表2. 光ディスクにおける総称コマンドのインプリメンテーション

総称コマンド	サポート	注記および制約事項
ADDLNK	なし	
CHGAUD	なし	

表2. 光ディスクにおける総称コマンドのインプリメンテーション (続き)

総称コマンド	サポート	注記および制約事項
CHGAUT	あり	UDF フォーマットの光ディスク・ボリュームの場合のみサポートされます。 ¹
CHGCURDIR	あり	
CHGOWN	あり	UDF フォーマットの光ディスク・ボリュームの場合のみサポートされます。
CHGPGP	あり	UDF フォーマットの光ディスク・ボリュームの場合のみサポートされます。
CHKIN	なし	
CHKOUT	なし	
CPY	あり	
CRTDIR	あり	/QOPT または次のレベルのディレクトリー (ボリュームを表す) を作成しようとする、コマンドは失敗します。
DSPAUT	あり	
DSPCURDIR	あり	
DSPLNK	あり	
ENDJRN	なし	
MOV	一部あり	ファイルまたはサブディレクトリーを含む場合、QOPT はディレクトリーの移動をサポートしません。QOPT はボリュームの移動をサポートしません。
RMVDIR	一部あり	QOPT は RMVLNK(*YES) をサポートしません。
RMVLNK	あり	
RNM	一部あり	QOPT はディレクトリーの名前変更をサポートしません。
RST	一部あり	QOPT は、SUBTREE (*STG) を使用したボリューム全体の復元をサポートします。
RTVCURDIR	あり	
SAV	一部あり	QOPT は、SUBTREE (*STG) を使用したボリューム全体の保管をサポートします。
SAVRST	なし	
STRJRN	なし	
WRKAUT	あり	UDF フォーマットの光ディスク・ボリュームの場合のみサポートされます。 ^{1, 2}
WRKLNK	あり	

注:

1. この操作を実行するには、ファイルの所有者であるか、または *ALLOBJ 特殊権限を持つ必要があります。

QOPT は、光ディスク・ファイルおよびディレクトリーに関連したオブジェクト・レベルの権限の保守も遵守も行いません。したがって、オブジェクト・レベルの権限の変更または取り消しを試みることはできません。新規オブジェクト権限 (OBJAUT) パラメーターに使用可能な唯一の値は、*SAME です。

新規データ権限 (DTAAUT) パラメーターに *EXCLUDE を指定することはできません。コマンド・パラメーターの規則では、新規データ権限パラメーターに *EXCLUDE を指定する場合は、新規オブジェクト権限パラメーターに値 *NONE を指定する必要があります。

所有者、グループ、または他のユーザーに関連する権限を取り消す場合、新規データ権限パラメーターの値として *NONE を指定することができます。この場合、指定されたユーザーおよびユーザーのデータ権限は、許可ユーザーのリストから除去されます。

QOPT は専用認可リストの保守も遵守も行いません。新規データ権限を所有者、グループ、またはその他 (*PUBLIC) のいずれでもないユーザーに割り当てることはできません。

2. QOPT は専用認可リストの保守も遵守も行いません。新規ユーザーを追加し (WRKAUT 画面のオプション 1)、新規データ権限を所有者、グループ、またはその他 (*PUBLIC) のいずれでもないユーザーに割り当てることはできません。

許可ユーザーのリストからユーザーを除去するためのオプション 4 は、サポートされていません。除去対象のユーザーに対してオプション 2 を選択し、プロンプト表示 (F4) を行ってください。新規データ権限パラメーター (DTAAUT) に *NONE を設定し、新規オブジェクト権限パラメーター (OBJAUT) に (*SAME) を設定する必要があります。

例: 統合ファイル・システム

以下のプログラミング例では、QOPT 物理ファイル・システムに関連する、統合ファイル・システムの UNIX タイプ API の使用法を示します。この例は、Integrated Language Environment (ILE) C (i5/OS オペレーティング・システム用) で書かれています。

プログラミング例では、以下の機能を示しています。

- 光ディスク・ディレクトリー項目の検索
- 光ディスク・ファイルの作成
- ファイルの書き込み
- ファイルのクローズ
- ファイルのオープン
- ファイルの読み取り
- ファイルへのオフセットの変更

コーディング例

このプログラム例は、さまざまな統合ファイル・システム API の使用法を示しています。本プログラムは C 言語で書かれています。

注: 以下のコード例をご使用になることで、51 ページの『コードに関するライセンス情報および特記事項』の条件に同意されたものとみなされます。

```
/******  
/* This program demonstrates the use of various integrated file      */  
/* system functions applied to the QOPT physical file system          */  
/* including:                                                         */  
/*   chdir()   - change current directory                            */  
/*   close()   - close file                                          */  
/*   closedir() - close directory                                    */  
/*   creat()   - create file                                         */  
/*   lseek()   - seek file (change file offset)                     */  
/*   open()    - open file                                           */  
/*   opendir() - open directory                                     */  
/*   read()    - read file                                           */  
/*   readdir() - read directory entry                               */  
/*   rewinddir() - rewind directory entries                         */  
/*   stat()    - directory statistics                               */  
/*   write()   - write file                                          */  
/******  
#include <stdio.h>  
#include <unistd.h>  
#include <sys/types.h>  
#include <dirent.h>  
#include <sys/stat.h>  
#include <fcntl.h>  
  
void main (void)  
{  
    /******  
    /* local variables, contents and defines                          */  
    /******  
    char path[294];           /* optical path                        */  
    DIR *dirP;                /* pointer to the directory */  
    int filedes;              /* open file descriptor     */  
    struct dirent *direntP;   /* directory entry structure */  
    struct stat info;         /* dir/file information     */  
    int volume_number;        /* what it says...         */  
    int rc = 0;               /* function return codes   */  
    int kk = 0;               /* local counter           */  
    char data[] = "The quick red fox jumped over the fence";  
  
    /******  
    /* Retrieve the list of volumes from the QOPT physical file      */  
    /* system by opening the QOPT pfs root directory and reading the */  
    /* directory entries.                                           */  
    /******  
    memset(path,              /* clear path name         */  
           0x00,  
           sizeof(path));  
    strcpy(path,              /* set physical file system */  
           "/QOPT");  
    rc = stat("/QOPT", &info); /* determine number of files */  
    if (rc != 0)  
        perror("stat() failed:");  
  
    dirP = opendir(path);     /* open the directory      */  
    if (dirP == NULL)  
        perror("opendir() failed:");  
  
    for (kk = 1; kk <= info.st_nlink; kk++)  
    {  
        direntP = readdir(dirP);  
        if (direntP == NULL)  
            perror("readdir() failed:");  
        printf("%d) %s\n", kk, direntP->d_name);  
    }  
}
```

```

/*****/
/* Prompt user for the volume they want to work with and make it */
/* the current directory. */
/*****/
printf("\nEnter the number the volume you want to work with:\n");
scanf("%d", &volume_number);

rewinddir(dirP); /* beginning of directory */
for (kk = 1; kk <= volume_number; kk++)
direntP = readdir(dirP); /* get requested dir. entry */

strcat(path, "/");
strcat(path, direntP->d_name);
rc = chdir(path); /* set current working dir. */
if (rc != 0)
perror("chdir() failed:");
if (getcwd(path, sizeof(path)) == NULL)
perror("getcwd() failed:");
printf("\nThe current working directory is: %s\n", path);

rc = closedir(dirP); /* close the directory */
if (rc != 0)
perror("closedir() failed:");

/*****/
/* Create and open a file write only. If the file exists it */
/* will be truncated. The owner will have read, write, and */
/* execute authority to the file. */
/*****/
strcat(path, "/");
printf("\nEnter a file name:\n");
scanf("%s", &path[strlen(path)]);

filedes = creat(path, S_IRWXU);
if (filedes == -1)
{
perror("creat() failed");
return;
}

rc = write(filedes, data, sizeof(data));
if (rc == -1)
perror("write() failed:");

close(filedes);

/*****/
/* Read back the file and print it. */
/*****/
memset(data, 0x00, sizeof(data));
filedes = open(path, O_RDWR);
if (filedes == -1)
{
perror("open() failed");
return;
}

read(filedes, data, sizeof(data));
if (filedes == -1)
{
perror("read() failed");
return;
}
printf("\nThe data written to file is: %s\n", data);

/*****/

```



```

/* Change the offset into the file and change part of it. Read */
/* the entire file, print it out and close the file.          */
/*****
lseek(filedes, 4, SEEK_SET);
rc = write(filedes, "slow old ", 9);
if (rc == -1)
{
perror("write() failed");
return;
}
lseek(filedes, 18, SEEK_SET);
rc = write(filedes, "went under ", 11);
if (rc == -1)
{
perror("write() failed");
return;
}

lseek(filedes, 0, SEEK_SET);
read(filedes, data, sizeof(data));
if (filedes == -1)
{
perror("read() failed");
return;
}
printf("\nThe data now is: %s\n", data);

close(filedes);

printf("Done...\n");
return;
}

```

関連概念

UNIX タイプの API

階層ファイル・システムのプログラミング

階層ファイル・システム (HFS) API を使用して、直接接続または LAN 接続の光ディスク装置を対象に読み取りまたは書き込みを行うことができます。HFS API は、i5/OS オペレーティング・システムの一部です。

光ディスク・サポート用の HFS API サポートは、以下の 2 つの部分から構成されます。

- ファイルおよびディレクトリーとして知られるオブジェクトを処理するための、階層ファイル・システムへのアプリケーション・プログラミング呼び出しインターフェース。
- ファイルおよびディレクトリーが保管されるストレージ・デバイスを管理する、光ディスクまたはその他の登録済みファイル・システム。

HFS API サポートには以下の光ディスク機能があります。

- ディレクトリーの作成および削除
- ディレクトリーのオープン、読み取り、またはクローズ
- ファイルのオープン、読み取り、書き込み、またはクローズ
- ファイル内のバイトのロックまたはアンロック
- ファイル・サイズの取得または設定
- ファイルの名前変更、コピー、削除、または移動
- ディレクトリー項目属性の検索または変更

階層ファイル・システム API

光ディスク・ファイル・システムの場合、階層ファイル・システム (HFS) API の使用法は一般の API 使用法と比較して異なっています。

HFS がサポートする API はすべてのファイル・システムに共通ですが、それらの API に関する解釈や制約事項は各ファイル・システムで異なります。次の表は、各 HFS API の光ディスクについての解釈を要約したものです。いくつかの API については、LAN 接続の光ディスク装置と直接接続の光ディスク装置とで制約事項の違いがあります。直接接続の光ディスク装置の例として、CD、DVD、および SCSI 接続の光ディスク・ライブラリーがあります。LAN 接続の光ディスク装置の例として、イーサネットまたはトークンリング接続の光ディスク・ライブラリーがあります。

表 3. 光ディスク HFS API の制約事項

HFS API	直接接続の使用上の注意	LAN 接続の使用上の注意
ファイル・ポインターの変更 (QHFCHGFP)	なし。	なし。
ストリーム・ファイルのクローズ (QHFCLOSF)	なし。	なし。
ファイル・システムの制御 (QHFCFLFS)	<p>以下の要求をサポートします。</p> <ul style="list-style-type: none"> • SAV は、保留光ディスク・ファイルを保管します。 • RLS は、保留光ディスク・ファイルを解放します。 • SRD/VOL は、光ディスク・ボリュームから読み取られたセクターを戻します。 • SRD/DEV は、光ディスク装置から読み取られたセクターを戻します。 • RTV/VOL は、ボリューム固有の情報を戻します。 • GET は、最小限のデータ・キャッシュを使用してメディアからファイル・データを直接に読み取ります。UDF フォーマットのボリュームの場合、GET には、ファイルに先行するパス内の各ディレクトリーに対する *X 権限、およびファイルに対する *R 権限が必要です。 	<p>以下の要求をサポートします。</p> <ul style="list-style-type: none"> • UPD/LAN は、LAN ボリュームのリストの動的索引リフレッシュを実行します。 • UPD/VOL は、ボリューム固有の情報を戻します。 • RTV/VOL は、ボリューム固有の情報を戻します。 • RTV/DIR は、指定されたディレクトリーのサブディレクトリーおよびファイル項目を戻します。

表 3. 光ディスク HFS API の制約事項 (続き)

HFS API	直接接続の使用上の注意	LAN 接続の使用上の注意
<p>ストリーム・ファイルのコピー (QHFCPYSF)</p>	<p>ソース・ファイルが QOPT ファイル・システム内にある場合、ソース光ディスク・ボリュームに対する *USE 権限が必要です。</p> <p>ターゲット・ファイルが QOPT ファイル・システム内にある場合、ターゲット光ディスク・ボリュームに対する *CHANGE 権限が必要です。コピー情報パラメーター (バイト 1) のオプション 2 はサポートされません (コピーの付加)。指定された場合は、CPF1F62 が戻されます。</p> <p>操作が完了すると、QCRTDTTM、QACCDTTM、および QWRDTTM が現在日付に設定されます。</p> <p>QOPT と QDLS の両ファイル・システム間でコピーする場合、グローバル光ディスク属性 CPYATR に応じてファイル属性がオプションでコピーされます。この属性は、CHGOPTA コマンドを使用して表示または変更できます。</p> <p>QOPT と QDLS の両ファイル・システム間でコピーする場合、ファイル許可はコピーされません。これらのファイル・システム間で許可を保存する必要がある場合は、コピー (CPY) CL コマンドを使用してください。</p> <p>UDF フォーマットのボリューム上にソース・ファイルがある場合、ファイルに先行するパス内の各ディレクトリーに対する *X 権限が必要です。ファイルに対する *R 権限も必要です。</p> <p>UDF フォーマットのボリューム上にターゲット・ファイルがある場合、親ディレクトリーに対する *WX 権限、および親ディレクトリーに先行するパス内の各ディレクトリーに対する *X 権限が必要です。</p>	<p>ソース・ファイルが QOPT ファイル・システム内にある場合、ソース光ディスク・ボリュームに対する *USE 権限が必要です。</p> <p>ターゲット・ファイルが QOPT ファイル・システム内にある場合、ターゲット光ディスク・ボリュームに対する *CHANGE 権限が必要です。コピー情報パラメーター (バイト 1) のオプション 2 はサポートされません (コピーの付加)。</p> <p>直接接続のライブラリー内のボリュームから LAN 接続の光ディスク装置内のボリュームへのコピーは、サポートされません。</p>
<p>ディレクトリーの作成 (QHFCRTDR)</p>	<p>操作が完了すると、QCRTDTTM、QACCDTTM、および QWRDTTM が現在日付に設定されます。</p> <p>操作が完了すると、QFILSIZE および QALCSIZE が 0 に設定されます。</p> <p>光ディスク・ボリュームに対する *CHANGE 権限が必要です。</p> <p>光ディスクのルート・ディレクトリーの作成はサポートされません。</p> <p>ディレクトリーのボリューム部分の作成はサポートされません。</p> <p>属性情報テーブルで渡される属性はサポートされず、結果は CPF1F71 エラー・メッセージになります。属性情報テーブル・パラメーターの長さは 0 でなければなりません。</p> <p>光ディスク属性 OPT.CHGATDTTM が作成されます。これは、ディレクトリー属性が変更された最後の時刻を示します。この日付は現在日付に設定されます。ユーザーが属性を指定しても、無視されます。</p> <p>UDF フォーマットのボリュームの場合、親ディレクトリーに対する *WX 権限が必要です。親ディレクトリーに先行するパス内の各ディレクトリーに対する *X 権限も必要です。ディレクトリーの所有者はディレクトリーを作成するユーザーとなり、所有者データ権限は *RWX に設定されます。1 次グループおよび 1 次グループ・データ権限は、親ディレクトリーと同じになります。*PUBLIC データ権限は、親ディレクトリーと同じになります。</p>	<p>操作が完了すると、QCRTDTTM、QACCDTTM、および QWRDTTM が現在日付に設定されます。</p> <p>操作が完了すると、QFILSIZE および QALCSIZE が 0 に設定されます。</p> <p>光ディスク・ボリュームに対する *CHANGE 権限が必要です。</p> <p>光ディスクのルート・ディレクトリーの作成はサポートされません。</p> <p>ディレクトリーのボリューム部分の作成はサポートされません。</p> <p>すべての標準属性は無視されます。</p> <p>属性情報テーブル・パラメーターの長さは 0 に設定されなければなりません。</p>

表 3. 光ディスク HFS API の制約事項 (続き)

HFS API	直接接続の使用上の注意	LAN 接続の使用上の注意
ディレクトリーの削除 (QHFDLTDR)	<p>光ディスクのルート・ディレクトリーの削除はサポートされません。</p> <p>パスのボリューム部分の削除はサポートされません。</p> <p>光ディスク・ボリュームに対する *CHANGE 権限が必要です。</p> <p>UDF フォーマットのボリュームの場合、親ディレクトリーに対する *WX 権限、および親ディレクトリーに先行するパス内の各ディレクトリーに対する *X 権限が必要です。削除されるディレクトリーに対する *W 権限も必要です。</p>	<p>光ディスクのルート・ディレクトリーの削除はサポートされません。</p> <p>パスのボリューム部分の削除はサポートされません。</p> <p>光ディスク・ボリュームに対する *CHANGE 権限が必要です。</p>
ストリーム・ファイルの削除 (QHFDLTSF)	<p>光ディスク・ボリュームに対する *CHANGE 権限が必要です。</p> <p>UDF フォーマットのボリュームの場合、親ディレクトリーに対する *WX 権限が必要です。親ディレクトリーに先行するパス内の各ディレクトリーに対する *X 権限も必要です。削除されるファイルに対する *W 権限も必要です。</p>	<p>光ディスク・ボリュームに対する *CHANGE 権限が必要です。</p>
ファイル・サイズの取得 (QHFGETSZ)	なし	なし
ファイル・サイズの設定 (QHFSETSZ)	なし	サポートされません

表 3. 光ディスク HFS API の制約事項 (続き)

HFS API	直接接続の使用上の注意	LAN 接続の使用上の注意
ストリーム・ファイルのオープン (QHFOPNFSF)	<p>オープン情報パラメーター:</p> <ul style="list-style-type: none"> 書き込み専用または読み取り/書き込みのアクセス・モード (バイト 6) を指定してオープンする場合は、ボリュームに対する *CHANGE 権限が必要です。 読み取り専用のアクセス・モード (バイト 6) を指定してオープンする場合は、ボリュームに対する *USE 権限が必要です。 ロック・モード (バイト 5) は異なるオープン・インスタンスをまたがって強制されます。同一ジョブが 1 つのファイルを複数回オープンすると、これらのオープンによるロックが競合する可能性があります。 <p>書き込み操作のオープン要求で QALCSIZE が指定された場合、光ディスク・メディアが検査されて、十分なスペースが使用可能かどうか調べられます。そうでない場合、エラー・メッセージ CPF1F62 が戻されます。</p> <p>QALCSIZE 以外の標準属性はすべて無視されます。</p> <p>ファイルが作成されると、QCRTDTTM、QACCDTTM、および QWRDTTM が現在日付に設定されます。ファイルが更新されると、QWRDTTM が現在日付に設定されます。ファイルが読み取られる場合、タイム・スタンプは変更されません。QACCDTTM は、ファイルが作成された後で変更されることはありません。常に QCRTDTTM と等しくなります。</p> <p>以下の許可規則は、UDF フォーマットのボリュームの場合のみ適用されます。</p> <ul style="list-style-type: none"> 読み取り (READ) 用にファイルを開く場合、ファイルに先行するパス内の各ディレクトリーに対する *X 権限、およびファイルに対する *R 権限が必要です。 書き込み (WRITE) 用に既存のファイルを開く場合、ファイルに先行するパス名内の各ディレクトリーに対する *X 権限、およびファイルに対する *W 権限が必要です。 読み取り/書き込み (READ/WRITE) 用に既存のファイルを開く場合、ファイルに先行するパス名内の各ディレクトリーに対する *X 権限、およびファイルに対する *RW 権限が必要です。 ファイルを作成する場合、親ディレクトリーに対する *WX 権限が必要です。 ファイルを作成する場合、ファイルの所有者はファイルを作成するユーザーとなり、所有者データ権限は *RWX に設定されます。1 次グループおよび 1 次グループ・データ権限は、親ディレクトリーと同じになります。*PUBLIC データ権限は、親ディレクトリーと同じになります。 	<p>オープン情報パラメーター:</p> <ul style="list-style-type: none"> バイト 3 (ライトスルー・フラグ) はサポートされません。 バイト 7 (実行するオープン操作のタイプ) はサポートされません。 読み取り専用のアクセス・モード (バイト 6) を指定してオープンする場合は、ボリュームに対する *USE 権限が必要です。 <p>ファイル・オープンの試みが読み取り専用アクセスに対するものでない限り、属性は許容されず、エラー・メッセージ CPF1F71 が出力されます。属性情報テーブル・パラメーターの長さは 0 でなければなりません。</p> <p>ファイル・オープンの試みが読み取り専用アクセスに対するものである場合、属性は許容されますが、無視されます。</p>
ストリーム・ファイルの読み取り (QHFRDSF)	なし。	なし。
ディレクトリー項目属性の検索 (QHFRVAT)	<p>光ディスク・ボリュームに対する *USE 権限が必要です。</p> <p>UDF フォーマットのボリュームの場合、ファイルに先行するパス名内の各ディレクトリーに対する *X 権限、および読み取られるファイルまたはディレクトリーに対する *R 権限が必要です。</p>	<p>ユーザーは、LAN 標準属性である QFILSIZE、QCRTDTTM、および QWRDTTM しか検索できません。</p> <p>光ディスク・ボリュームに対する *USE 権限が必要です。</p> <p>属性情報テーブル・パラメーターの長さは 0 に設定されなければなりません。</p>

表 3. 光ディスク HFS API の制約事項 (続き)

HFS API	直接接続の使用上の注意	LAN 接続の使用上の注意
ストリーム・ファイルの書き込み (QHFWRTSF)	なし。	なし。
ディレクトリー項目属性の変更 (QHFCGAT)	<p>QFILATTR は、変更可能な唯一の標準属性です。その他は、指定されてもすべて無視されます。</p> <p>読み取り専用フラグ (QFILATTR 属性のバイト 1) は、ファイルに対してのみ設定できます (ディレクトリーに対しては設定できません)。ディレクトリーに対して指定しても、無視されます。</p> <p>変更フラグ (QFILATTR 属性のバイト 5) は、0 または 1 のいずれかに設定できます。このフラグは、ファイルの作成または書き込みのたびに、自動的に (1) に設定されます。</p> <p>OPT.CHGATDTM を指定しても、無視されます。</p> <p>光ディスク・ボリュームに対する *CHANGE 権限が必要です。</p> <p>UDF フォーマットのボリュームの場合、ファイルに先行するパス名内の各ディレクトリーに対する *X 権限、およびファイルに対する *W 権限が必要です。</p>	API はサポートされません。
ディレクトリーのクローズ (QHFCLODR)	なし。	API はサポートされません。
バッファ・データの強制 (QHFFRCFSF)	<p>ボリューム・メディア・フォーマットが *UDF の場合、データは光ディスク・メディアに書き込まれます。</p> <p>ボリューム・メディア・フォーマットが *UDF ではない場合、データは、光ディスク・メディアではなく、内部ディスク・ストレージに書き込まれます。</p> <p>読み取り専用アクセスでオープンされたファイルの場合、この API は無効です。</p>	API はサポートされません。
ストリーム・ファイルの範囲のロックおよびアンロック (QHFLULSF)	なし。	API はサポートされません。
ストリーム・ファイルの移動 (QHFMVVSF)	<p>ソース・ファイルが QOPT ファイル・システム内にある場合、光ディスク・ソース・ボリュームに対する *CHANGE 権限が必要です。</p> <p>ターゲット・ファイルが QOPT ファイル・システム内にある場合、光ディスク・ターゲット・ボリュームに対する *CHANGE 権限が必要です。</p> <p>QOPT と QDLS の両ファイル・システム間で移動する場合、グローバル光ディスク属性 CPYATR に応じてファイル属性がオプションでコピーされます。この属性は、CHGOPTA コマンドを使用して表示または変更できます。</p> <p>UDF フォーマットのボリューム上にソース・ファイルがある場合、親ディレクトリーに対する *WX 権限、および親ディレクトリーに先行するパス名内の各ディレクトリーに対する *X 権限が必要です。ファイルに対する *RW 権限も必要です。</p> <p>UDF フォーマットのボリューム上にターゲット・ファイルがある場合、親ディレクトリーに対する *WX 権限、およびファイルに先行するパス名内の各ディレクトリーに対する *X 権限が必要です。</p>	API はサポートされません。

表 3. 光ディスク HFS API の制約事項 (続き)

HFS API	直接接続の使用上の注意	LAN 接続の使用上の注意
ディレクトリーのオープン (QHFOPNDR)	<p>ファイル・システムのルート (/QOPT) をオープンすると、「ディレクトリー項目の読み取り」時に直接接続と LAN 接続の両方のボリュームを戻すことができます。</p> <p>ファイル・システムのルートをオープンする場合、ロック・モードは無視されます。</p> <p>ロック・モードの「非ロック」はサポートされません。これが要求されると、ロック・モードの「否認なし」で代替されます。</p> <p>光ディスク・ボリュームに対する *USE 権限が必要です。</p> <p>UDF フォーマットのボリュームの場合、オープンされるディレクトリーに先行するパス名内の各ディレクトリーに対する *X 権限、およびオープンされるディレクトリーに対する *R 権限が必要です。</p>	API はサポートされません。
ディレクトリー項目の読み取り (QHRDDR)	<p>QOPT ファイル・システム名なしで QNAME が戻されます。</p> <p>QNAME は、LAN 接続のボリュームに対して設定される唯一のフィールドです。</p> <p>QWRDITM は、常に QCRTDITM と等しくなります。</p> <p>ファイルおよびディレクトリーの場合、QACCDITM は、常に QCRTDITM と等しくなります。</p> <p>ボリュームの場合、QACCDITM は、最新ボリューム参照日と等しくなります。</p>	API はサポートされません。
ストリーム・ファイルの名前変更 (QHRNMSF)	<p>光ディスク・ボリュームに対する *CHANGE 権限が必要です。</p> <p>UDF フォーマットのボリュームの場合、親ディレクトリーに対する *WX 権限、および親ディレクトリーに先行するパス名内の各ディレクトリーに対する *X 権限が必要です。名前変更されるファイルに対する *W 権限も必要です。</p>	API はサポートされません。
ディレクトリーの名前変更 (QHRNMDR)	API はサポートされません。	API はサポートされません。

ファイル・システム制御関数

ファイル・システム制御 (QHCTLFS) API を使用すると、光ディスク・サポートで光ディスク・ファイル・システムに固有の操作を実行できます。

以下に示す関数は、HFS API を介した光ディスク固有のものであり、それ以外には使用できません。直接接続および LAN 接続の光ディスク装置に対して、それぞれ別の関数が使用可能です。

直接接続の光ディスク装置用のファイル・システム制御関数

以下のファイル・システム制御関数は、直接接続の光ディスク装置に対して使用可能です。

- **SAV**. 保留光ディスク・ファイルを保管します。
- **RLS**. 保留光ディスク・ファイルを解放します。
- **SRD/VOL**. 光ディスク・ボリュームに対するセクター読み取りを実行します。
- **SRD/DEV**. 光ディスク装置に対するセクター読み取りを実行します。
- **RTV/VOL**. ボリューム固有の情報を戻します。
- **GET**. 最小限のキャッシュを使用してメディアからファイル・データを直接に読み取ります。

保留光ディスク・ファイル保管関数

QHCTLFS API を使用して、保留光ディスク・ファイルを保管します。プロセスは、保留光ディスク・ファイルを保管できるように、そのファイルに対する読み取りアクセスが許可されていなければなりません。

QHCTLFS プログラムの入力バッファの構文は次のとおりです。

```
'SAV' + '/' + held-file-path + '//' + destination-file-path
```

例えば、次のとおりです。

- 入力データ・バッファ: SAV/VOLUME1/DIRECTORY1/FILE1//VOLUME2/DIRECTORY2/FILE2
- 入力データ・バッファ長: 54

この関数は、「保留光ディスク・ファイル処理 (WRKHLDOPTF)」画面のオプションにより使用することも可能です。ただし、「保留光ディスク・ファイル処理 (WRKHLDOPTF)」画面の保管オプションと異なり、ファイル・システム制御 API の保留光ディスク・ファイル保管関数は、保留ファイルを保管後に自動的に解放しません。したがって、後で、明示的な保留光ディスク・ファイルの解放要求が必要になります。

保留光ディスク・ファイル解放関数

ファイルの保留状況をクリアし、光ディスクに書き込む義務から光ディスク・ファイル・システムを解放するには、QHCTLFS API を使用します。プロセスは、保留光ディスク・ファイルを解放できるように、そのファイルに対する読み取り/書き込みアクセスが許可されていなければなりません。このことは、現在そのファイルが他のアクティブなジョブによってロックされているとはならないことを意味します。

QHCTLFS プログラムの入力バッファの構文は次のとおりです。

```
'RLS' + '/' + held-file-path
```

例えば、次のとおりです。

- 入力データ・バッファ: RLS/VOLUME1/DIRECTORY1/FILE1
- 入力データ・バッファ長: 28

この関数は、「保留光ディスク・ファイル処理 (WRKHLDOPTF)」画面のオプションにより使用することも可能です。

セクター読み取り関数

QHCTLFS API は、光ディスク・メディアのセクター読み取りを実行します。セクター読み取り関数は、光ディスク・メディア上でデータが保管されている場所をアプリケーションが正確に認識している場合に有効です。セクター読み取り関数は、ファイルのオープンもクローズもなしに、またすべての HFS API とは独立して実行することができます。複数のセクターを一度に読み取ることもできます。

ファイル・システム制御のセクター読み取り関数を発行するための入力バッファには、次の 2 つのバリエーションがあります。

```
SRD/VOL/volume_name/starting sector/number of sectors
```

```
SRD/DEV/device_name/starting sector/number of sectors
```

どちらも、ユーザーが要求したセクターの範囲を戻します。光ディスク・ボリュームからも、光ディスク装置からもセクターを要求できます。例えば、アプリケーションが光ディスク・ボリューム VOL01 のセクター 1000 から始まる 5 つのセクターの読み取りを必要とする場合、次のように要求します。

```
SRD/VOL/VOL01/1000/5
```


注: DEV は、スタンドアロン CD および DVD 装置に対して有効です。

ボリューム情報検索関数

QHCTLFS API は、特定のボリュームに関する情報を検索します。

QHCTLFS プログラムの入力バッファ・フォーマットは次のとおりです。

```
RTV/VOL/volume_name
```

出力バッファに戻される情報のフォーマットは、ボリューム属性の出力ファイル構造 (QAMODVA) と同じです。

ファイル・データ取得関数

QHCTLFS HFS API を使用して、ファイルのデータ・ブロックを出力バッファに直接に読み込むことができます。この関数により、ファイル全体を順次に読み取る場合、または大きなデータ・ブロックを読み取る場合にパフォーマンスが向上します。光ディスク・ファイル・システムは、通常のオープン、読み取り、およびクローズの各ストリーム・ファイル HFS API を使用する場合には、データをコピーしたり、キャッシュに入れたりしません。ファイルのランダム読み取り操作を行う場合は、オープン、読み取り、およびクローズの各ストリーム・ファイル API が依然として最高のパフォーマンスを提供します。

この API を使用する場合は、以下の制約事項が適用されます。

- 出力バッファを 512 バイト境界に位置合わせします。
- 最大読み取りサイズは、16 384 000 バイトです。
- HFS API は、ファイルに対する共用非更新 (*SHRNUP) アクセスを必要とします。
- 呼び出し側プログラムは、ユーザー状態 (システム状態ではなく) でなければなりません。
- HFS API は、ボリュームに対する *USE 権限を必要とします。

QHCTLFS プログラムの入力バッファの構文は次のとおりです。

```
'GET' + '/' + entire path + '//' + bytes to read + '/' + file offset
```

次の例では、ファイル FILE.XXX の先頭 (オフセット=0) から 15 MB を読み取ります。

- 入力データ・バッファ: GET/VOL1/DIR1/SUBDIR1/FILE.XXX//15728640/0
- 入力データ・バッファ長: 42

読み取られたバイト数は、「戻されたデータの長さ」パラメーターに戻されます。上記の例で、FILE.XXX のサイズがわずか 50 KB であれば、このフィールドには 51200 が戻されます。したがって、この要求を発行する前にファイル・サイズを知る必要はありません。同様に、「戻されたデータの長さ」パラメーターに 15728640 が戻された場合、ファイルは少なくとも 15 MB のサイズがあります。全データを検索するには、さらに読み取り操作が必要になる可能性があります。

読み取りバイト数が 4096 の倍数であることは、必須ではありません。ただし、数が 4096 の倍数でない場合、要求したバイト数を超えてデータが出力バッファに読み込まれることもあります。この理由は、装置が 4096 バイトのブロック単位で入出力を行うためです。したがって、この問題を避けるため、データの読み取りは 4096 バイトの倍数で行うことをお勧めします。

ファイル・システム制御 (GET 関数) からのエラー

次の表は、この API を使用して発生する可能性のある一般的なアプリケーション・エラーのいくつかを示しています。

表 4. GET 関数の一般的なエラー

メッセージ	エラー
OPT1812 with 6030 as unexpected return code (OPT1812 が予期しない戻りコード 6030 を戻しました)	ファイル・オフセットがファイルの終わりを超えています。
OPT1812 with A950 as unexpected return code (OPT1812 が予期しない戻りコード A950 を戻しました)	出力バッファが 512 バイト境界上に位置合わせされていません。
OPT1860	読み取りバイト数がバッファ・サイズを超えています。
OPT1812 with C060 as unexpected return code (OPT1812 が予期しない戻りコード C060 を戻しました)	16 384 000 バイトより大きいサイズを読み取ろうとしました。
CPF1F48	入力バッファが無効です。構文を確認してください。

LAN 接続の光ディスク装置用のファイル・システム制御関数

以下のファイル・システム制御関数は、LAN 接続のメディア・ライブラリーに対して使用可能です。

- UPD/LAN - LAN ボリューム・リストの動的リフレッシュを実行します。
- UPD/VOL - ボリューム固有の情報を戻します。
- RTV/VOL - ボリューム固有の情報を戻します。
- RTV/DIR - 指定されたディレクトリーのサブディレクトリーおよびファイル項目を戻します。

ボリューム情報更新関数

QHCTLFS API を使用して、特定のボリュームに関する情報を検索するか、あるいは LAN 上の使用可能なボリュームの内部リストを更新します。

QHCTLFS プログラムの入力バッファ・フォーマットは次のとおりです。

UPD/VOL/volume_name

これは、以下のことを実行します。

- UPD/VOL/volume-name: この入力バッファ・フォーマットを使用すると、ボリューム上のフリー・スペース量、合計ユーザー・スペース、メディア・タイプ、および裏面のボリューム ID が戻されます。フォーマットは以下のとおりです。
 - バイト (1 から 32): 裏面のボリューム ID。
 - バイト (33): 予約済み。
 - バイト (34 から 37): ボリューム上のユーザー・フリー・スペース。これは、4 バイトの 2 進数フィールドです。
 - バイト (38 から 41): ボリューム上の合計フリー・スペース。これは、ボリューム上のユーザー・フリー・スペースとボリューム上の予約済みスペースを合わせたものです。ボリューム上の予約済みスペースは、そのボリュームのボリュームフルしきい値を設定する際に判別されます。これは、4 バイトの 2 進数フィールドです。
 - バイト (42): メディア・タイプ。これは、1 バイトの 2 進数フィールドで、以下のいずれかの値になります。
 - 0 = 無効なメディアまたは 3431 スタンドアロン・ドライブ
 - 1 = Write Once Read Many (WORM) メディア
 - 2 = 再書き込み可能メディア

- バイト (43): ボリューム上のフリー・スペースの大きさ。これは、1 バイトの 2 進数フィールドで、以下のいずれかの値になります。
 - 0 = スペース・フィールドはバイト数単位です。
 - 1 = スペース・フィールドは K バイト数 (1024) 単位です。
 - 2 = スペース・フィールドは M バイト数 (1048576) 単位です。
- バイト (44): ボリューム上の合計スペースの大きさ。これは、1 バイトの 2 進数フィールドで、以下のいずれかの値になります。
 - 0 = スペース・フィールドはバイト数単位です。
 - 1 = スペース・フィールドは K バイト数 (1024) 単位です。
 - 2 = スペース・フィールドは M バイト数 (1048576) 単位です。
- UPD/LAN: この入力バッファ・フォーマットを使用すると、活動状態のすべてのサーバー上で使用可能なボリュームの内部リストが更新されます。この関数は、データ・サーバーに対してカートリッジの追加または除去を行った後で実行できます。

ボリューム情報検索関数

QHCTLFS API は、特定のボリュームに関する情報を検索します。

QHCTLFS プログラムの入力バッファ・フォーマットは次のとおりです。

RTV/VOL/volume_name

出力バッファに戻される情報のフォーマットは、ボリューム属性の出力ファイル構造 (QAMODVA) と同じです。

システムは、すべての光ディスク装置タイプのボリューム用にフォーマット QAMODVA を使用します。フォーマットは同じですが、LAN ボリュームの場合はすべてのフィールドに値が入るわけではありません。

ディレクトリー情報検索関数

QHCTLFS API を使用して、特定のディレクトリーのファイルおよびサブディレクトリーのリストを検索します。

QHCTLFS プログラムの入力バッファは次のとおりです。

RTV/DIR/volume_name/directory_name

ディレクトリー情報は、次のフォーマットで出力バッファに戻されます。

- CBdirectoryBCBdirectoryBCBfilenameBCBfilenameBB。ここで、
 - C
 - D = ディレクトリー項目
 - F = ファイル名項目
 - B = EBCDIC ブランク
 - BB = スtringの終了を示すための 2 桁の EBCDIC ブランク

出力バッファは、少なくとも 31 KB の長さが必要です。

標準属性

ファイルおよびディレクトリーのディレクトリー項目には、それらに関連した、属性と呼ばれる情報が含まれています。各属性は、名前と値で構成されます。標準属性は、ディレクトリーまたはファイルを作成すると自動的に生成される属性です。標準属性名は、識別しやすいように、文字 Q で始まります。

すべてのファイル・システムが標準属性を使用します。光ディスク・ファイル・システムでは、固有に解釈されるものがいくつかあります。LAN 接続の光ディスク装置では、標準属性の解釈が、直接接続の光ディスク装置とは異なります。

QALCSIZE 属性

出力フィールドとして、QALCSIZE は、ファイルによって光ディスクに割り振られるバイト数を示します。ディレクトリーの場合は常に 0 です。

書き込み要求時に QALCSIZE 属性が「ストリーム・ファイルのオープン」で指定されると、指定された量を割り振るのに使用可能な十分なスペースがメディアにあるかどうか調べられます。光ディスク・ボリューム上で十分なスペースが使用可能でない場合、メッセージ CPF1F61 「**No free space available on media (メディア上に使用可能なフリー・スペースがありません)**」が出力されます。この属性の使用法についての詳細は、37 ページの『メディア容量およびボリュームしきい値』を参照してください。

QACCDTTM 属性

QACCDTTM 属性は、光ディスク・ファイル・システムではサポートされていません。これは、ファイル作成日時 (QCRTDTTM) 属性と常に同じです。

QCRTDTTM 属性

QCRTDTTM 属性は、ファイルまたはディレクトリーの作成日を示します。

QWRDTTM 属性

QWRDTTM 属性は、データが光ディスク・ファイルに書き込まれた最後の日時を示します。ファイル属性が最後に書き込まれた日時は反映されません。

QFILATTR 属性

QFILATTR 属性は、直接接続の光ディスク・サポート装置によってのみサポートされます。ファイル・フラグの光ディスクの場合の解釈は以下のとおりです。

- **読み取り専用ファイル:** i5/OS オペレーティング・システムは、光ディスク・ファイル・システムを介してこの属性の完全サポートを提供しています。この属性をオン (1) に設定すると、ファイルの削除も上書きもできません。
- **隠しファイル:** i5/OS オペレーティング・システムは、ユーザー・アプリケーションでの管理用にこの属性を保守しますが、光ディスク・ファイル・システムによる完全なサポートはしていません。この属性をオン (1) に設定しても、光ディスク・ファイル・システムはそのファイルを隠しファイルとして認識しません。ユーザー・アプリケーションは、この属性がオンに設定されているファイルに対して特別なアクセスを必要としません。
- **システム・ファイル:** i5/OS オペレーティング・システムは、ユーザー・アプリケーションでの管理用にこの属性を保守しますが、光ディスク・ファイル・システムによる完全なサポートはしていません。こ

の属性をオン (1) に設定しても、光ディスク・ファイル・システムはそのファイルをシステム・ファイルとして認識しません。ユーザー・アプリケーションは、この属性がオンに設定されているファイルに対して特別なアクセスを必要としません。

- **変更ファイル:** i5/OS オペレーティング・システムでは、この属性が光ディスク・ファイル・システムによってサポートされます。ファイルの作成または書き込み時に、自動的にオン (1) が設定されます。「ディレクトリー項目属性の変更 (QHFCHGAT)」API の使用によってのみ、オフ (0) が設定できます。

拡張属性

拡張属性は、標準ではないため階層ファイル・システム (HFS) に認識されないファイルおよびディレクトリーの特別な属性です。これらは、通常はビジネス・アプリケーションにより定義されますが、特別な意味を持つものとして光ディスク・ファイル・システムによって認識されるものもあります。

OPT.CHGATDTTM 属性

OPT.CHGATDTTM 属性は、ファイル属性が書き込まれた最後の日時を示します。ディレクトリー項目属性の検索 (QHFRTVAT) コマンドを介して拡張属性としてユーザー・アプリケーションに戻されます。

QOPT.IOMETH 属性

QOPT.IOMETH は、光ディスク・ファイル・システムのための特別な属性です。直接接続の光ディスク・サポート装置によってのみサポートされます。LAN サポートでは無視されます。また、メディア・フォーマットが Universal Disk Format の場合、システムはこの属性を無視します。

アプリケーションによる「ストリーム・ファイルのオープン」要求時に、この名前の拡張属性が属性情報テーブル (AIT) 内の属性名フィールドとして渡されると、光ディスク・ファイル・システムは特殊方式の入出力が要求されていることを認識します。光ディスク・ファイル・システムは、AIT の属性値フィールドから特殊方式の入出力を検索します。

現在、光ディスク・ファイル・システムでサポートされる特殊方式の入出力は 1 つだけです。この方式の入出力を要求する場合は、QOPT.IOMETH 属性用の属性値フィールドに値 (EXPNBUFF) を指定します。光ディスク・ソフトウェアは、この特殊な拡張属性を、通常の拡張属性ではなく、要求された入出力方式として認識します。この後は決してファイルと関連付けられることはなく、ファイルの属性が検索される場合も表示されません。そのプロセスのすべての読み取り操作は、ファイルがクローズされるまで、拡張バッファ入出力を使用します。拡張バッファ入出力を使用する場合の方法および制約事項を以下に示します。拡張バッファ入出力を使用すべきかどうかを判断するには、『拡張バッファ入出力方式』のトピックを参照してください。

属性情報テーブルの HFS 属性は、いくつかのフィールドから構成されます。これらのフィールドと、拡張バッファ入出力用にファイルをオープンするときに指定する値は、次の表に要約されています。

表 5. 拡張バッファ属性定義

フィールド	データ・タイプ (注参照)	EBIO の場合の値
属性名	CHAR(*)	QOPT.IOMETH
属性値	CHAR(*)	EXPNBUFF
属性名の長さ	BIN(4)	0000000B
属性値の長さ	BIN(4)	00000008

表 5. 拡張バッファ属性定義 (続き)

フィールド	データ・タイプ (注参照)	EBIO の場合の値
注:		
<ul style="list-style-type: none"> CHAR(*) は、可変バイト数の文字情報を示します。 BIN(4) は、4 バイトの 2 進情報を示します。 文字フィールドはすべて大文字で設定します。 		

属性フィールドの値の他に、属性情報テーブルを作成するために次の 2 つの追加フィールドが必要です。

- テーブルに定義された属性の数
- 各属性へのテーブル・オフセット (バイト数)

「ストリーム・ファイルのオープン (QHFOPNNSF)」API は、入力として 10 バイトのオープン情報を必要とします。拡張バッファ入出力用にファイルをオープンしようとする場合、オープン情報は以下の制約事項に従わなければなりません。

- ファイルが存在する場合のアクションは、そのファイルのオープンでなければならない。
- ファイルが存在しない場合のアクションは、エラーを戻すことでなければならない。
- ファイルのロック・モードは、「書き込み拒否」または「読み取り/書き込み拒否 (排他的)」でなければならない。
- ファイルのアクセス・モードは、「読み取り専用」でなければならない。

属性情報テーブルに拡張バッファ入出力属性が存在し、これらの制約事項のいずれかに従っていない場合、OPT1133 メッセージが出力されます。このメッセージは、オープン情報のどのフィールドがエラーで渡されたかを示します。

属性のフォーマット、属性情報テーブル、あるいは「ストリーム・ファイルのオープン」API については、『API トピック』を参照してください。

拡張バッファ入出力の制約事項

前述の拡張バッファ入出力用にファイルをオープンする場合の制約事項に加えて、ファイルのオープン後は、以下の API を拡張バッファ入出力用に使用することはできません。

- ストリーム・ファイルの範囲のロックまたはアンロック
- ストリーム・ファイル・サイズの設定
- ストリーム・ファイルの書き込み

関連概念

アプリケーション・プログラミング・インターフェース

関連タスク

38 ページの『拡張バッファ入出力方式』

QOPT.IOMETH 拡張属性を使用し、階層ファイル・システム (HFS) を介してストリーム・ファイルをオープンすると、大容量の光ディスク・ファイルのデータの一部 (全部ではなく) を主として読み取るアプリケーションのパフォーマンスが向上します。入出力のこの方式は、拡張バッファ入出力と呼ばれます。

階層ファイル・システムを使用したファイル属性のコピー

階層ファイル・システム (HFS) API をサポートするファイル・システムの間で、ファイル属性をコピーすることができます。

QOPT と QDLS の両ファイル・システム間での属性のコピー

階層ファイル・システムを使用して QOPT と QDLS の両ファイル・システム間でファイルをコピーする場合、ターゲット・ファイルには、デフォルト・ファイル属性またはソース・ファイルのファイル属性のいずれかが割り当てられます。これは、光ディスク属性変更 (CHGOPTA) コマンドの属性コピー (CPYATR) グローバル値に指定する値によって決まります。

CHGOPTA コマンドで CPYATR グローバル値が *NO と指定された場合、QOPT と QDLS の両ファイル・システム間でコピーされるファイルに対して、デフォルト・ファイル属性が作成されます。

CHGOPTA コマンドで CPYATR グローバル値が *YES と指定された場合、QOPT と QDLS の両ファイル・システム間のコピーでは、ソース・ファイルのファイル属性がターゲット・ファイルにコピーされます。

QDLS から QOPT への属性のコピー

QDLS から QOPT へのコピー操作または移動操作の場合、以下のデフォルト属性がターゲット・ファイルに割り当てられます。

- 標準ファイル属性:
 - 作成日時は現在日時に設定されます。
 - 変更日時は現在日時に設定されます。
 - アクセス日時は現在日時に設定されます。
 - QFILATTR 標準属性は 00000 に設定され、それぞれの意味は、「ファイルは読み取り専用ではない」、「ファイルは隠しファイルではない」、「ファイルはシステム・ファイルではない」、「ファイルはディレクトリーではない」、および「ファイルは、最後にアーカイブまたは作成されて以降変更されていない」です。
- DIA 文書属性はコピーされません。
- ユーザー定義の拡張属性はコピーされません。

ファイル名 (QNAME) およびファイル・サイズ (QFILSIZE) は維持されます。

QOPT から QDLS への属性のコピー

QOPT から QDLS へのコピー操作または移動操作の場合、以下のデフォルト属性が作成されます。

- 標準ファイル属性:
 - 作成日時は現在日時に設定されます。
 - 変更日時は現在日時に設定されます。
 - アクセス日時は現在日時に設定されます。
 - QFILATTR 標準属性は 00000 に設定され、それぞれの意味は、「ファイルは読み取り専用ではない」、「ファイルは隠しファイルではない」、「ファイルはシステム・ファイルではない」、「ファイルはディレクトリーではない」、および「ファイルは、最後にアーカイブまたは作成されて以降変更されていない」です。
- DIA 文書属性:

- DIA.CA04C700 (テキスト記述) はファイル名に設定されます。
 - DIA.CA04C701 (プロファイル GCID) は、コード・ページ 697 および文字セット 500 に設定されます。
 - DIA.CA04C706 (ファイル・タイプ) は 000E (PC ファイル) に設定されます。
 - DIA.CA04C720 (ライブラリー割り当て文書名) は、このファイルを表すように割り当てられます。
 - DIA.CA04C708 (最終変更日時) は現在日時に設定されます。
 - DIA.CA04C707 (作成日時) は現在日時に設定されます。
 - DIA.CA04C710 (NLS 情報) は、ジョブの言語 ID と、国 ID または地域 ID に設定されます。
 - DIA.CA04C740 (ファイル日時) は現在日時に設定されます。
- ユーザー定義の拡張属性はコピーされません。

ファイル名 (QNAME) およびファイル・サイズ (QFILSIZE) は維持されます。

例: 光ディスク・ファイル・システムの場合の階層ファイル・システム API のプログラミング

以下の階層ファイル・システム (HFS) の例は、光ディスク・ファイル・システムのプログラムを作成するのに役立ちます。

このトピックでは、ILE RPG プログラム言語で HFS API を使用方法を示します。

プログラミング例では、以下の機能を示しています。

- 配列からパス名を検索する。
- HFS API を呼び出して、ストリーム・ファイルをオープンする。
- HFS API を呼び出して、プログラムにパラメーターとして渡された 256 バイトのバッファーを書き込む。
- HFS API を呼び出して、ストリーム・ファイルをクローズする。

注: 以下のコード例をご使用になることで、51 ページの『コードに関するライセンス情報および特記事項』の条件に同意されたものとみなされます。

パスの取得およびサブルーチンの呼び出し

この例では、パスを取得し、サブルーチンを呼び出します。

```

E              AR      1  5 36
C      *ENTRY      PLIST
* 2 PARAMETERS - A DATA BUFFER ID AND AN INDEX TO THE ARRAY
C              PARM      DATAIN  256
C              PARM      IDX      10
* MOVE THE ARRAY ELEMENT TO A FIELD CALLED "PATH"
C              MOVE AR,IDX  PATH
* EXECUTE SUBROUTINES TO OPEN, WRITE AND CLOSE A FILE
C              EXSR OPNSF
C  RTCD      IFEQ 0
C              EXSR WRTSF
C              EXSR CLOSF
C              END
C              SETON                      LR
* TABLE/ARRAY . . . . . :  AR
** /QOPT/MYVOL1/DIRA/FILE

```



```

/QOPT/MYVOL1/DIRA/SUBDIRB/FILE
/QOPT/MYVOL1/DIRA/SUBDIRB/C/FILE
/QOPT/MYVOL1/DIRA/SUBDIRB/C/D/FILE
/QOPT/MYVOL1/DIRA/SUBDIRB/C/D/E/FILE

```

ファイル・オープン用のデータ構造の定義

この例では、HFS でデータ構造を定義します。

```

* PATH LENGTH PARAMETER
IPATHLN    DS
I          B  1    40PATHL

* OPEN INFORMATION PARAMETER
IOPNINF    DS
I          1  1  EXISTS
I          2  2  NOTTHR
I          3  3  SYNASY
I          4  4  RSV1
I          5  5  SHAREM
I          6  6  ACCESS
I          7  7  OTYPE
I          8 10  RSV3

* ATTRIBUTE LENGTH PARAMETER
IATTRLN    DS
I          B  1    40ATTRL

* RETURN CODE PARAMETER
IRETCOD    DS
I          B  1    40RCLEN
I          B  5    80RTCD
I          9 15   CONDITN
I          16 16   RSV
I          17 272  MSG

* BYTES TO READ/WRITE PARAMETER
IBYTRDW    DS          B  1    40BYT2RW

* BYTES ACTUALLY READ/Written PARAMETER
IBYTACT    DS          B  1    40BYTARW

```

光ディスク・ファイルのオープン

この例では、光ディスク・ファイルをオープンします。

```

* PARAMETER LIST FOR QHFOPNSF CALL
C          POPNSF    PLIST
C          PARM          FHDLE 16
C          PARM          PATH  36
C          PARM          PATHL
C          PARM          OPNINF
C          PARM          ATRTBL 1
C          PARM          ATTRLN
C          PARM          ACTION 1
C          PARM          RETCD

C* OPEN FILE SUBROUTINE
C          OPNSF      BEGSR

C* FILL IN THE PATH AND ATTRIBUTE LENGTHS
C          Z-ADD36      PATHL  SET PATH LEN=36
C          Z-ADD*ZEROS  ATTRL  ZERO ATTRIBUTE LENGTH

C* FILL IN THE OPNINF PARAMETER
C          MOVE '0'     EXISTS  1  FAIL IF EXISTS
C          MOVE '1'     NOTTHR  1  CREATE IF NOT THERE
C          MOVE '0'     SYNASY  1  ASYNCHRONOUS
C          MOVE *BLANKS  RSV1    1
C          MOVE '1'     SHAREM  1  DENY NONE
C          MOVE '2'     ACCESS  1  READ/WRITE
C          MOVE '0'     OTYPE   1  NORMAL

```

```

C          MOVE *BLANKS   RSV3   3
C* CALL THE API TO OPEN THE STREAM FILE
C          CALL 'QHFOPNSF'POPNSF   50
C          OPNEND      ENDSR

```

光ディスクへのファイルの書き込み

この例では、光ディスクにファイルを書き込みます。

```

* PARAMETER LIST FOR QHFRDSF OR QHFWRTSF CALL
C          PRWSF      PLIST
C          PARM          FHDLE  16
C          PARM          DATAIN
C          PARM          BYT2RW
C          PARM          BYTARW
C          PARM          RETCD
C* CALL API TO WRITE TO THE FILE
C          WRTSF      BEGSR
C          Z-ADD256          BYT2RW SET WRITE LENGTH=256
C          CALL 'QHFWRTSF'PRWSF   50
C          WRTEND      ENDSR

```

光ディスク・ファイルのクローズ

この例では、光ディスク・ファイルをクローズします。

```

* PARAMETER LIST FOR QHFCLOSF CALL
C          PCLOSF      PLIST
C          PARM          FHDLE  16
C          PARM          RETCD
C* CALL API TO CLOSE THE FILE
C          CLOSF      BEGSR
C          CALL 'QHFCLOSF'PCLOSF   50
C          CLSEND      ENDSR
C* END OF SAMPLE RPG CALL TO THE HFS API

```

関連概念

アプリケーション・プログラミング・インターフェース

ヒント: 光ディスク装置プログラミング

このトピックの手法は、多くの場合、お客様のビジネス向けにカスタム光ディスク・プログラムを設計する際に役立ちます。

このトピックでは、アプリケーション・プログラマーがアプリケーションを最適化できるように、光ディスク・ファイル・システムによるファイル・データの管理方法について説明します。アプリケーションにはさまざまな要件があるため、このトピックでは、光ディスク・アプリケーションを作成する最良の方法を提案するわけではありません。しかし、すべてのアプリケーション・プログラマーに役立つ説明を提供します。

このトピックを使用して、HFS API または UNIX タイプ API のいずれかを介して光ディスク・ファイル管理を処理するための最良の方法を決定してください。このトピックは、直接接続の光ディスク・サポート向けのアプリケーションにのみ使用してください。

注: このトピックの概念は、光ディスク LAN サポートには適用されません。

メディア容量およびボリュームしきい値

光ディスク・ファイル・システムでは論理しきい値機能が用意され、アプリケーションが絶対ボリューム容量に達するのを防ぐのに役立ちます。論理しきい値は、ボリュームが初期化されるときに定義され、ボリュームごとに固有です。光ディスク・ボリューム変更 (CHGOPTVOL) コマンドを使用して、このしきい値を変更できます。

注: 論理ボリュームしきい値は、ハイパフォーマンス光ディスク・ファイル・システム (HPOFS) メディア・フォーマットに対してのみ適用できます。Universal Disk Format (UDF) メディア・フォーマットの場合、論理ボリュームしきい値は常に 100% であり、変更できません。

メディアがフルになったときの状態に対処する方法を工夫しておく必要があります。これは、特に Write Once Read Many (WORM) メディアへの書き込みの場合に当てはまります。以下の質問について検討しておく必要があります。

- ボリュームしきい値をどのように使用するか？
- ボリュームがフルになったら、何をするか？
- ボリューム・フル状態にどのように備えるか？

論理ボリュームしきい値は、HPOFS メディア・フォーマットに対してのみ適用できます。UDF メディア・フォーマットの場合、論理ボリュームしきい値は常に 100% であり、変更できません。

ボリュームしきい値は、アプリケーションが実際のボリューム・フル状態に対する備えを可能にするために用意されています。WORM メディアがフルになると、書き込み操作を行うことができなくなります。アプリケーションの要件に応じて、さまざまな方法でしきい値を使用して、メディアが物理的にフルになるのに備えることができます。

例えば、アプリケーションがスプール・ファイルのグループを光ディスクに書き込むとします。各グループが書き込まれた後で、書き込まれたばかりのスプール・ファイルへの索引を含む追加ファイルが書き込まれます。この索引がないと、スプール・ファイルが役に立たなくなる可能性があります。アプリケーションがメディア容量を管理できなければ、索引ファイルを書き込む前に、ボリュームはスペースを使い尽くす可能性があります。スペースを使い尽くすのを避ける方法の 1 つは、ボリュームしきい値を 99% に設定することです。メッセージ「No space available (使用可能なスペースがありません)」が出力されたら、アプリケーションは、しきい値を 100% に増やして、必要な追加ファイルを書き込むことができます。

ファイル単位でのメディア容量の管理

アプリケーションは、ファイル単位でメディア容量を管理することが必要になることもあります。

ファイルがボリュームに適合するかどうか判断する際に、次の方式が役立ちます。

- クローズ操作のエラーの処理。

光ディスク・ボリュームがしきい値 95% で初期化され、ボリュームしきい値に達するまでアプリケーションがファイルを書き込むとします。しきい値に達すると、アプリケーションは、メッセージ CPF1F61 「No free space available on media (メディア上に使用可能なフリー・スペースがありません)」を受け取ります。この時点で、CHGOPTVOL コマンドを使用してボリュームしきい値を 97% (あるいは、最大 100% までの任意の値) に増やすことができます。その後、ファイルのクローズを試みることができます。

- 「ストリーム・ファイルのオープン」HFS API での QALCSIZE の指定。

ファイルがボリュームに収まるかどうかを判断するもう 1 つの方法は、「ストリーム・ファイルのオープン」で割り振りサイズ (QALCSIZE) を指定することです。「ストリーム・ファイルのオープン」により、システムは属性 QALCSIZE に値を渡すことができます。この属性が有効なのは、作成または置換用にオープン操作が行われる場合であり、その他の場合には無視されます。QALCSIZE の値を指定すると、指定した値はボリューム上で使用可能なスペースと比較されます。使用可能なスペースが QALCSIZE より小さい場合、システムはメッセージ CPF1F61 を出力します。オープン操作を実行するためには、使用可能なスペースが QALCSIZE を超えていなければなりません。ファイルの最初のオープン・インスタンスのみがこの属性を受け付けます。複数回のファイルのオープンで指定されると、システムは追加の属性を無視します。

注: これにより、オープン操作の時点で光ディスク・ボリュームにスペースが実際に割り振られるわけではありません。ボリュームを検査して、要求されたバイト数が使用可能かどうかを確認するだけです。

この方法の使用には以下のような欠点があります。

1. オープン要求を出す時点で、作成するファイルのサイズを知っておく必要があります。
2. 複数のジョブが同一メディアに書き込もうとする場合、データが書き込まれる時点までスペースが依然として使用可能である保証がありません。

オープン要求を行う前にファイルのサイズが分かり、ファイルのオープン中にそのボリュームに書き込むジョブが他にない場合、この方法はファイルの作成前にメディア容量を検査するのに最適です。

- ボリューム上の使用可能なスペースの検索。

別の方法は、ボリューム上で使用可能なスペースをアプリケーションに検索させることです。出力ファイル・サポートを指定した光ディスク表示 (DSPOPT) コマンドを使用することにより、これが可能です。出力ファイルを読み取り、メディア上で使用可能と想定されるバイト数を検索することができます。

拡張バッファ入出力方式

QOPT.IOMETH 拡張属性を使用し、階層ファイル・システム (HFS) を介してストリーム・ファイルを開くと、大容量の光ディスク・ファイルのデータの一部 (全部ではなく) を主として読み取るアプリケーションのパフォーマンスが向上します。入出力のこの方式は、*拡張バッファ入出力*と呼ばれます。

拡張バッファ入出力は、ハイパフォーマンス光ディスク・ファイル・システム (HPOFS) または ISO 9660 フォーマットのメディアにアクセスするときに、HFS API アプリケーションのみが使用可能です。メディア・フォーマットが UDF の場合、この属性は無視されます。

注: HFS API を使用して、光ディスク・ファイル・データは、i5/OS 主記憶域にある仮想光ディスク・ファイルのバッファに入れられます。拡張バッファ入出力がオプションとして選択されていない場合、このバッファのサイズは、実際の光ディスク・ファイルのサイズに等しくなります。例えば、ファイルが HFS API 「ストリーム・ファイルのオープン」を使用してオープンされる場合、光ディスク・メディア上の 100 MB ファイルは、100 MB バッファを持ちます。光ディスク・バッファに関連するオーバーヘッド操作のパフォーマンス・コストは、バッファ・サイズに比例します。100 MB ファイルの 1 バイトを読み取るのにかかる時間は、50 KB ファイルの 1 バイトを読み取る時間よりもかなり多くなります。

光ディスク・ファイルが拡張バッファ入出力用にオープンされる場合、バッファのサイズはゼロで始まり、アプリケーションの要求に応じてデータがバッファに読み取られるにつれて拡張されます。サイズ拡張の最小量は 256 KB です。バッファは、要求されたデータが、まだバッファに含まれていない論理

256 KB ページ内に含まれている場合のみ拡張されます。これらの理由により、拡張バッファ入出力用にオープンされた 100 MB ファイルの 1 バイトを読み取るのにかかる時間は、同じ方法でオープンされた 50 KB ファイルの 1 バイトを読み取るのにかかる時間とほぼ同じになります。

拡張バッファ入出力が有効な状況

拡張バッファ入出力は、以下の条件のいずれかに当てはまる場合、読み取り操作のパフォーマンスを向上させるオプションとして考慮することをお勧めします。

- 読み取り対象の光ディスク・ファイルの標準的なサイズが 256 KB を超える。
- ストリーム・ファイルのオープンとクローズの間に光ディスク・ファイルから読み取るデータ量は、ファイル・データ全体のごく一部である。その割合を正確に規定することは不可能ですが、割合が小さくなるほど、達成されるパフォーマンス向上の度合いは大きくなります。例えば、拡張バッファ入出力を使用して 50 MB ファイルのうちの 25 KB を読み取るアプリケーションは、同じファイルの 45 MB を読み取るアプリケーションよりも大きなパフォーマンス向上が得られます。一度に 40 KB ずつの複数回の読み取りで 50 MB のファイル例全体を読み取るアプリケーションの場合、拡張バッファ入出力を使用してもパフォーマンス向上は得られません。
- ファイルが拡張バッファ入出力用にオープンされている間は、アプリケーションが「ストリーム・ファイル・サイズの設定」、「バイト範囲のロック/アンロック」、または「ストリーム・ファイルの書き込み」のいずれの API も発行しない。

関連資料

31 ページの『拡張属性』

拡張属性は、標準ではないため階層ファイル・システム (HFS) に認識されないファイルおよびディレクトリーの特別な属性です。これらは、通常はビジネス・アプリケーションにより定義されますが、特別な意味を持つものとして光ディスク・ファイル・システムによって認識されるものもあります。

「バッファ・データの強制」API

「バッファ・データの強制」API は、ファイル情報とディレクトリー情報を同期的に表示させます。

「バッファ・データの強制」(QHFFRCSF) または fsync() API を使用すると、光ディスク・ファイル・データの光ディスク・メディアへの書き込み中に、光ディスク・ファイル・データを不揮発性メディアに書き込むことができます。

光ディスク・ファイルの作成または更新の際、ファイルが正常にクローズされるまでは、データが光ディスクに存在する保証はありません。ただし、光ディスク・ファイル・データは、QHFFRCSF または fsync() API のいずれかを使用して、不揮発性ストレージに同期的に書き込むことができます。不揮発性ストレージのタイプは、光ディスク・メディア・フォーマットによって異なります。

ハイパフォーマンス光ディスク・ファイル・システム (HPOFS) の場合、すべてのファイル・データは内部ディスク・ストレージに書き込まれます。停電や他の予期しないエラーの発生によりファイルをクローズできない場合、データは保留光ディスク・ファイルを使用して回復することができます。

Universal Disk Format (UDF) の場合、すべてのファイル・データは、強制操作が発行されたときに光ディスクに書き込まれます。停電や他の予期しないエラーの発生によりファイルがクローズできない場合でも、回復は必要ありません。ただし、データが強制された後で書き込み操作が行われ、しかもクローズ操作が正常に終わらない場合、ファイル・データは予測不能です。強制操作に続く書き込み操作は非同期であるため、データは光ディスクに書き込まれない可能性があります。

保留光ディスク・ファイルの管理

光ディスク・メディアへの書き込み中のエラーのため保留されている仮想ファイルを、別のボリュームに保管することができます。保留光ディスク・ファイルとは、光ディスク・メディアに正常に書き込まれなかった仮想ファイルのことです。

仮想ファイルは、非 UDF フォーマットのボリューム上でファイルのクローズ操作中にエラーが発生すると、保留になります。アプリケーション・インターフェースおよび光ディスク・ユーティリティを使用して、これらの仮想ファイルを管理できます。UDF フォーマットのボリュームにアーカイブできなかったファイルの場合、保留ファイルの作成は行われません。

光ディスク・ボリュームがしきい値 95% で初期化され、ボリュームしきい値に達するまでアプリケーションがファイルを書き込むとします。しきい値に達すると、アプリケーションは、メッセージ CPF1F61 「No free space available on media (メディア上に使用可能なフリー・スペースがありません)」を受け取りません。この例では、絶対ボリューム容量に達し、ファイルはボリュームに収めるには大きすぎます。ボリュームしきい値を増やしても効果がないため、別の解決策が必要です。クローズ要求が失敗した場合、仮想ファイルは保留状態になります。保留光ディスク・ファイル処理コマンドを使用して、この仮想ファイルを別のボリュームに保管することができます。必要であれば、ファイルを別の名前で保管することができます。保管要求は、ファイル・システム制御関数を使用して実行することもできます。

パス名の要件

パス名という用語は、ファイル・システム名、ボリューム名、ディレクトリー名、およびファイル名を表します。

直接接続装置内のボリュームのパス名

直接接続装置内にあるボリュームのパス名では、区切り文字としてスラッシュ (/) が使用されます。パス名は、スラッシュで始まり、294 文字以内でなければなりません。直接接続装置上のパス名のフォーマットについては、以下の例を参照してください。

```
/QOPT/VOL_NAME/DIRECTORY_NAME/SUB_DIR1/.../SUB_DIRn/FILE_NAME
```

QOPT は、光ディスク・ファイル・システムを示します。HFS API または UNIX タイプ API を使用して光ディスク・サポートを呼び出す場合、これを使用して光ディスク・ファイル・システムを修飾する必要があります。ファイル・システム名に続くパスの部分は、289 文字を超えてはなりません。パス名の使用規則については、以下の項目を参照してください。

- パス名は、以下にリストされている文字を除いて、任意の EBCDIC 文字から構成できます。
 - X'00' から X'3F'
 - X'FF'
 - 引用符 (")
 - アスタリスク (*)
 - より小 (<) およびより大 (>) 記号
 - 疑問符 (?)
 - ハイフン (-)
 - 円記号 (¥)

統合ファイル・システム API を使用して UDF フォーマットのボリュームにアクセスする場合は、無効な文字は、X'00' から X'3F'、X'FF'、およびバックスラッシュのみです。

- ボリューム ID は、HPOFS メディア・フォーマットの場合は最大 32 文字、UDF メディア・フォーマットの場合は最大 30 文字が可能です。この ID には、英字 (A から Z)、数字 (0 から 9)、ハイフン (-)、下線 (_)、またはピリオド (.) のみを含めます。先頭文字は英字または数字でなければならず、また ID にはブランクを含めてはなりません。
- パス名には 1 つ以上のディレクトリーを含めることができますが、必須ではありません。すべてのサブディレクトリーを合わせた文字数の総数は、256 文字を超えてはなりません。
- ファイル名は、パス内の最後のエレメントです。パス内のディレクトリーの長さにより、ファイル名の長さが制限されます。ディレクトリー名とファイル名を合わせて、256 文字を超えてはなりません。ディレクトリー名の直前のスラッシュは、この 256 文字の一部と見なされます。

LAN 接続装置内のボリュームのパス名

LAN 接続光ディスク装置内にある光ディスク・ボリュームのパス名では、区切り文字としてスラッシュ (/) が使用されます。パス名は、スラッシュで始まり、261 文字以内でなければなりません。LAN 接続光ディスク装置での光ディスク・ボリュームのパス名のフォーマット例は、次のとおりです。

```
/QOPT/VOL_NAME/DIRECTORY_NAME/SUB_DIR1/.../SUB_DIRn/FILE_NAME
```

QOPT は光ディスク・ファイル・システムを表し、HFS API または統合ファイル・システム API を使用して光ディスク・サポートを呼び出す場合、これを使用して光ディスク・ファイル・システムを修飾する必要があります。ファイル・システム名に続くパスの部分は、256 文字を超えてはなりません。LAN 接続装置上のパス名の使用規則については、以下の項目を参照してください。

- パス名に使用可能な文字については、IBM 3995 Optical Library Dataserver Operator Guide for C-Series

Models  を参照してください。

- ボリューム名は必須で、最大 32 文字を含むことができます。
- パス名には 1 つ以上のディレクトリーを含めることができますが、必須ではありません。すべてのサブディレクトリーを合わせた文字数の総数は、254 文字を超えてはなりません。
- ファイル名は、パス内の最後のエレメントです。ファイル名の長さは、パス内のボリュームおよびディレクトリーの長さによって制限されます。ボリューム名、ディレクトリー名、およびファイル名を合わせて、256 文字を超えてはなりません。ボリューム名およびディレクトリー名の直前のスラッシュは、この 256 文字の一部と見なされます。

例: 光ディスク記憶との間のスプール・ファイルの移動

これらの基本的な光ディスク・プログラミング例では、API を使用して制御言語 (CL) プログラムを作成します。

ストリーム・ファイルのコピー: コマンド・ソース

注: 以下のコード例をご使用になることで、51 ページの『コードに関するライセンス情報および特記事項』の条件に同意されたものとみなされます。

```

/*****/
/*                               */
/* COMMAND NAME:  CPYSTRF          */
/*                               */
/* COMMAND TITLE: Copy Stream File */
/*                               */
/* COMMAND DESCRIPTION: Copy stream file between two file systems */
/*                               */
/*****/
CMD PROMPT('Copy Stream File')

```

```

        PARM      KWD(SRCFILE) TYPE(*CHAR) LEN(300) MIN(1) +
MAX(1) PROMPT('Source file name')      +
        VARY(*YES)
PARM      KWD(TGTFILE) TYPE(*CHAR) LEN(300) MIN(1) +
MAX(1) PROMPT('Target file name')      +
        VARY(*YES)
PARM      KWD(RPLFILE)TYPE(*CHAR) LEN(6) DFT(*NO) +
        SPCVAL((*NO '0  ') (*YES '1  ')) +
        PROMPT('Replace existing file')

```

ストリーム・ファイルのコピー: CL プログラム・ソース

この CL 例は、ファイル・システム間でストリーム・ファイルをコピーするのに使用できます。

```

/*****/
/*          */
/* PROGRAM: CPYSTRF (Copy Stream File)          */
/*          */
/*          */
/* DESCRIPTION:                                  */
/* This is the CL program for sample CL command CPYSTRF. This */
/* program can be used to copy stream files between file      */
/* systems. The actual copy is done by making a call to      */
/* the HFS API program QHFCPYSF (Copy stream file).          */
/*          */
/*          */
/* INPUT PARAMETERS:                              */
/* - Complete source path                               */
/* Example: /filesystem/directory1/directoryx/file          */
/*          /QDLS/DIRA/DIRB/FILE01                        */
/*          - or -                                       */
/*          /filesystem/volume/directory1/directoryx/file   */
/*          /QOPT/VOLN01/DIRA/DIRB/FILE01                  */
/* - Complete target path                               */
/* Note: Except for the file the path must already exist.    */
/* Example: /filesystem/directory1/directoryx/file          */
/*          /QDLS/DIRA/DIRB/FILE01                        */
/*          - or -                                       */
/*          /filesystem/volume/directory1/directoryx/file   */
/*          /QOPT/VOLN01/DIRA/DIRB/FILE01                  */
/* - Replace existing target file                       */
/* *YES - replace existing file                          */
/* *NO - do not replace existing file                    */
/*          */
/*          */
/* LOGIC:                                             */
/* - Separate source file length and value              */
/* - Ensure source path is converted to upper case      */
/* - Separate target file length and value              */
/* - Ensure target path is converted to upper case      */
/* - Call copy stream file                              */
/*          */
/*          */
/* EXAMPLE:                                           */
/* The example will copy document THISWEEK from folder BILLS */
/* to optical volume YEAR1993. The document will be put into */
/* directory /BILLS/DEC as file WEEK50.                 */
/* Folders are stored in file system DLS (document library services)*/
/*          */
/*          CPYSTRF SRCFILE('/QDLS/BILLS/THISWEEK')          */
/*          TGTFILE('/QOPT/YEAR1993/BILLS/DEC/WEEK50')      */
/*          RPLFILE(*NO)                                    */
/*          */
/*****/
PGM PARM(&SRCFILE &TGTFILE &CPYINFO);

/*****/

```



```

/* Input parameters */
/*****/
DCL VAR(&SRCFILE); TYPE(*CHAR) LEN(300)
DCL VAR(&TGTFILE); TYPE(*CHAR) LEN(300)
DCL VAR(&CPYINFO); TYPE(*CHAR) LEN(6)

/*****/
/* Program variables */
/*****/
DCL VAR(&SRCLLEN); TYPE(*CHAR) LEN(4) +
VALUE(X'00000000')
DCL VAR(&TGTLLEN); TYPE(*CHAR) LEN(4) +
VALUE(X'00000000')
DCL VAR(&ERRCODE); TYPE(*CHAR) LEN(4) +
VALUE(X'00000000')
DCL VAR(&COUNT); TYPE(*DEC) LEN(5 0)
DCL VAR(&TBL); TYPE(*CHAR) LEN(10) +
VALUE('QSYSTRNTBL')
DCL VAR(&LIB); TYPE(*CHAR) LEN(10) +
VALUE('QSYS ')

/*****/
/* Monitor for any messages sent to this program */
/*****/
MONMSG MSGID(CPF0000) EXEC(GOTO CMDLBL(DONE))
MONMSG MSGID(OPT0000) EXEC(GOTO CMDLBL(DONE))

/*****/
/* The HFS API needs to be passed the file and the file length. */
/* By coding the VARY(*YES) parameter on the command definition */
/* for the source and target file we are passed the length of */
/* entered value as a 2 byte binary field which precedes the */
/* actual value entered. */
/*****/

/*****/
/* Separate source file length and file value. Ensure source */
/* file is upper case. */
/*****/
CHGVAR VAR(%SST(&SRCLLEN 3 2)) VALUE(%SST(&SRCFILE 1 2))
CHGVAR VAR(%SST(&SRCFILE 1 300)) VALUE(%SST(&SRCFILE 3 298))

CHGVAR VAR(&COUNT); VALUE(%BIN(&SRCLLEN 3 2))
CALL QDCXLATE (&COUNT +
&SRCFILE +
&TBL +
&LIB)

/*****/
/* Separate target file length and file value. Ensure target */
/* file is upper case. */
/*****/
CHGVAR VAR(%SST(&TGTLLEN 3 2)) VALUE(%SST(&TGTFILE 1 2))
CHGVAR VAR(%SST(&TGTFILE 1 300)) VALUE(%SST(&TGTFILE 3 298))

CHGVAR VAR(&COUNT); VALUE(%BIN(&TGTLLEN 3 2))
CALL QDCXLATE (&COUNT +
&TGTFILE +
&TBL +
&LIB)

/*****/
/* Call the copy stream file HFS API to copy the source file to */
/* the target file. */
/*****/
CALL QHFCPYSF (&SRCFILE +
&SRCLLEN +

```

```

&CPYINFO      +
&TGTFILE      +
&TGTLLEN      +
&ERRCODE)

```

```

SNDPGMMSG MSG('CPYSTRF completed successfully')
RETURN

```

```

DONE:
SNDPGMMSG MSGID(OPT0125) MSGF(QSYS/QCPFMSG)      +
MSGDTA(CPYSTRF) MSGTYPE(*ESCAPE)
RETURN

```

```

ENDPGM

```

光ディスクへのデータベース・ファイルのコピー: コマンド・ソース

```

/*****/
/*                                          */
/* COMMAND NAME:      CPYDBOPT          */
/*                                          */
/* COMMAND TITLE:    Copy Database to Optical */
/*                                          */
/* DESCRIPTION:      Copy database file to an optical file */
/*                                          */
/*****/
CPYDBOPT:  CMD          PROMPT('Copy DB to Optical')

      PARM      KWD(FRMFILE) TYPE(QUAL1) MIN(1)      +
              PROMPT('From file')

      PARM      KWD(FRMMBR) TYPE(*NAME) LEN(10)      +
              SPCVAL((*FIRST)) EXPR(*YES) MIN(1)    +
              PROMPT('From member')

      PARM      KWD(TGTFILE) TYPE(*CHAR) LEN(300)    +
              MIN(1) EXPR(*YES)                    +
              PROMPT('Target file')
QUAL1:  QUAL    TYPE(*NAME) LEN(10)
QUAL    TYPE(*NAME) LEN(10) DFT(*LIBL)              +
              SPCVAL((*LIBL) (*CURLIB))            +
              PROMPT('Library')

```

光ディスクへのデータベース・ファイルのコピー: CL プログラム・ソース

この CL 例は、データベース・ファイルのメンバーを光ディスク記憶にコピーするのに使用できます。

```

/*****/
/*                                          */
/* PROGRAM: CPYDBOPT (Copy Database to Optical) */
/*                                          */
/*                                          */
/* DESCRIPTION:                                          */
/* This is the CL program for sample CL command CPYDBOPT. This */
/* program can be used to copy a member from a database file to */
/* optical storage.                                          */
/*                                          */
/*                                          */
/* DEPENDENCIES:                                          */
/* - The sample command and program CPYSTRF exists.      */
/* - There is an existing folder named OPTICAL.FLR      */
/* This folder is used for temporary storage when copying */
/* from database to optical. It is assumed that this folder is */
/* empty and that the user will delete anything which gets */
/* copied into it.                                          */
/*                                          */
/*                                          */
/*****/

```

```

/* INPUT PARAMETERS: */
/* -From file */
/* - From member */
/* - Complete target path */
/* Assumption: - Except for the file the complete path currently */
/* exists. */
/* - File does not currently exist. */
/* Example: /filesystem/volume/directory1/directoryx/file */
/* /QOPT/VOLN01/DIRA/DIRB/FILE01 */
/* */
/* */
/* LOGIC: */
/* - Separate file and library */
/* - Copy file to folder */
/* - Build source file */
/* - Copy file from Document Library Service (DLS) to OPT */
/* */
/* */
/* EXAMPLE: */
/* The example will copy member MYMEMBER in file MYFILE in library */
/* MYLIB to optical storage. It will be stored as file */
/* MYFILE.MYMEMBER in directory /MYLIB on volume VOLN01. */
/* */
/* CPYDBOPT FRMFILE(MYLIB/MYFILE) */
/* FRMMBR(MYMEMBER) */
/* TGTFILE('/QOPT/VOLN01/MYLIB/MYFILE.MYMEMBER') */
/* */
/*****/
PGM PARM(&FROMFILE &FROMMBR &TGTFILE);

/*****/
/* Input parameters */
/*****/
DCL VAR(&FROMFILE); TYPE(*CHAR) LEN(20)
DCL VAR(&FROMMBR); TYPE(*CHAR) LEN(10)
DCL VAR(&TGTFILE); TYPE(*CHAR) LEN(300)

/*****/
/* Program variables */
/*****/
DCL VAR(&FILE); TYPE(*CHAR) LEN(10)
DCL VAR(&LIB); TYPE(*CHAR) LEN(10)
DCL VAR(&SRCFILE); TYPE(*CHAR) LEN(28) +
VALUE('/QDLS/OPTICAL.FLR/xxxxxxxxxx')

/*****/
/* Monitor for all messages sent to this program */
/*****/
MONMSG MSGID(CPF0000) EXEC(GOTO CMDLBL(DONE))
MONMSG MSGID(IWS0000) EXEC(GOTO CMDLBL(DONE))
MONMSG MSGID(OPT0000) EXEC(GOTO CMDLBL(DONE))

/*****/
/* Separate file and library names then copy the DB file to a */
/* PC folder. */
/*****/
CHGVAR VAR(&FILE); VALUE(%SST(&FROMFILE 1 10))
CHGVAR VAR(&LIB); VALUE(%SST(&FROMFILE 11 10))

CPYTOPCD FROMFILE(&LIB/&FILE); +
TOFLR(OPTICAL.FLR) +
FROMMBR(&FROMMBR); +
TRNTBL(*NONE)

/*****/
/* Complete the source file path name with the member and copy */
/* the stream file from DLS to optical */

```

```

/*****/
CHGVAR VAR(%SST(&SRCFILE 19 10)) VALUE(&FROMMBR);

CPYSTRF SRCFILE(&SRCFILE);          +
        TGTFILE(&TGTFILE);

SNDPGMMSG MSG('CPYDBOPT completed successfully')
RETURN

DONE:
SNDPGMMSG MSGID(OPT0125) MSGF(QSYS/QCPFMSG)          +
        MSGDTA(CPYDBOPT) MSGTYPE(*ESCAPE)
RETURN

ENDPGM

```

光ディスクへのスプール・ファイルのコピー: コマンド・ソース

```

/*****/
/*
/* COMMAND NAME: CPYSPLFOPT          */
/*
/* COMMAND TITLE: Copy Spooled File to Optical */
/*
/* DESCRIPTION: Copy spooled file to an optical file */
/*
/*****/
CPYSPLFO: CMD PROMPT('Copy Spooled File to Optical')

PARM KWD(FRMFILE) TYPE(*NAME) LEN(10) +
      MIN(1) +
      PROMPT('From file')

PARM KWD(TGTFILE) TYPE(*CHAR) LEN(300) +
      MIN(1) EXPR(*YES) +
      PROMPT('Target file')

PARM KWD(JOB) TYPE(Q2) +
      DFT(*) SNGVAL(*) +
      MIN(0) MAX(1) +
      PROMPT('Jobname')

PARM KWD(SPLNBR) TYPE(*CHAR) LEN(5) +
      SPCVAL((*ONLY) (*LAST)) DFT(*ONLY) +
      PROMPT('Spool number')

Q2: QUAL TYPE(*NAME) LEN(10) +
      MIN(1) +
      EXPR(*YES)

QUAL TYPE(*NAME) LEN(10) +
      EXPR(*YES) +
      PROMPT('User')

QUAL TYPE(*CHAR) LEN(6) +
      RANGE(000000 999999) +
      EXPR(*YES) FULL(*YES) +
      PROMPT('Number')

```

光ディスクへのスプール・ファイルのコピー: CL プログラム・ソース

この CL 例は、スプール・ファイルを光ディスク記憶にコピーするのに使用できます。

```

/*****/
/*
/* PROGRAM: CPYSPLFOPT (Copy Spooled File to Optical) */
/*

```

```

/* */
/* DESCRIPTION: */
/* This is the CL program for sample CL command CPYSPLFOPT. This */
/* program can be used to copy a spooled file to optical storage. */
/* */
/* */
/* DEPENDENCIES: */
/* - The sample command and program CPYDBOPT exists. */
/* - The sample command and program CPYSTRF exists. */
/* - There is an existing folder named OPTICAL.FLR */
/* This folder is used for temporary storage when copying */
/* from spooled files to optical. It is assumed that this folder */
/* is empty and that the user will delete anything which gets */
/* copied into it. */
/* - This CL program uses the CL command CPYSPLF to copy the */
/* spooled files to a physical file before copying them to */
/* optical. When you use the CPYSPLF command to copy */
/* a spooled file to a physical file, certain information can */
/* be lost or changed. Before using this command please */
/* refer to the CL Reference Book for the limitations and */
/* restrictions of the CPYSPLF command. */
/* - There is an existing file named LISTINGS in library QUSRSYS. */
/* It is assumed that this file contains no existing members */
/* and that any members that are created will be deleted by the */
/* user. The record length of the file is 133. */
/* */
/* */
/* INPUT PARAMETERS: */
/* - From file */
/* Specify the name of the spooled file to be copied. */
/* - Target file */
/* Assumption: Except for the file the path must already exist. */
/* Example: /filesystem/volume/directory1/directoryx/file */
/* /QOPT/VOLN01/DIRA/DIRB/FILE01 */
/* - Job */
/* Specify the name of the job that created the spooled file */
/* which is to be copied. The possible values are: */
/* The job that issued this command is the job that */
/* created the spooled file. */
/* - or - */
/* job-name Specify the name of the job that created the */
/* spooled file. */
/* user-name Specify the user name that identifies the user */
/* profile under which the job was run. */
/* job-number Specify the system assigned job number. */
/* - Spool number */
/* If there are multiple files for a job specify the files */
/* spool number. */
/* */
/* */
/* LOGIC: */
/* - Separate job into its three parts: job name, user, job number */
/* - Copy spooled files to database */
/* - Copy database to optical */
/* */
/* */
/* EXAMPLE: */
/* The example will copy spooled file QSYSPRT spool number 2 which */
/* the current process has printed to optical storage. */
/* It will be stored on volume YEAR92 in directory */
/* /DEC/WEEK01/MONDAY as file INVOICES */
/* */
/* CPYSPLFO SPLFILE(QSYSPRT) */
/* TGTFILE('/QOPT/YEAR92/DEC/WEEK01/MONDAY/INVOICES') */
/* SPLNBR(2) */
/* */
/* */
/*****

```

```

PGM PARM(&FROMFILE &TGTFILE &JOB &SPLNBR);

/*****/
/* Input parameters */
/*****/
DCL VAR(&FROMFILE); TYPE(*CHAR) LEN(10)
DCL VAR(&TGTFILE); TYPE(*CHAR) LEN(300)
DCL VAR(&JOB); TYPE(*CHAR) LEN(26)
DCL VAR(&SPLNBR); TYPE(*CHAR) LEN(5)

/*****/
/* Program variables */
/*****/
DCL VAR(&JNAME); TYPE(*CHAR) LEN(10)
DCL VAR(&JUSER); TYPE(*CHAR) LEN(10)
DCL VAR(&JNUM); TYPE(*CHAR) LEN(6)

/*****/
/* Monitor for all messages that can be signalled */
/*****/
MONMSG MSGID(CPF0000) EXEC(GOTO CMDLBL(DONE))
MONMSG MSGID(OPT0000) EXEC(GOTO CMDLBL(DONE))

/*****/
/* Separate each part of the job name and call the copy spool */
/* file command using the current job or the specified name. */
/*****/
CHGVAR VAR(&JNAME); VALUE(%SST(&JOB 1 10))
CHGVAR VAR(&JUSER); VALUE(%SST(&JOB 11 10))
CHGVAR VAR(&JNUM); VALUE(%SST(&JOB 21 6))

IF COND(&JNAME *EQ '*') THEN(DO)
  CPYSPLF FILE(&FROMFILE); +
  TOFILE(QUSRSYS/LISTINGS) +
  TOMBR(&FROMFILE); +
  SPLNBR(&SPLNBR); +
  CTLCHAR(*FCFC)
  ENDDO
ELSE DO
  CPYSPLF FILE(&FROMFILE); +
  TOFILE(QUSRSYS/LISTINGS) +
  TOMBR(&FROMFILE); +
  JOB(&JNUM/&JUSER/&JNAME); +
  SPLNBR(&SPLNBR); +
  CTLCHAR(*FCFC)
  ENDDO

/*****/
/* Copy the database file to optical storage */
/*****/
CPYDBOPT FRMFILE(QUSRSYS/LISTINGS) +
  FRMMBR(&FROMFILE); +
  TGTFILE(&TGTFILE);

SNDPGMMSG MSG('CPYSPLFOPT completed successfully')
RETURN

DONE:
SNDPGMMSG MSGID(OPT0125) MSGF(QSYS/QCPFMSG) +
  MSGDTA(CPYSPLFOPT) MSGTYPE(*ESCAPE)

RETURN

ENDPGM

```

データベースへの光ディスクのコピー: コマンド・ソース

```
/*
/* *****
/*
/* COMMAND NAME:  CPYOPTDB
/*
/*
/* COMMAND TITLE: Copy Optical to Database
/*
/*
/*
/* DESCRIPTION:   Copy optical file to database file
/*
/*
/* *****
CPYOPTDB:  CMD          PROMPT('Copy Optical to DB ')

      PARM          KWD(SRCFILE) TYPE(*CHAR) LEN(300)  +
                        MIN(1)  EXPR(*YES)              +
                        PROMPT('Source file')

      PARM          KWD(TOFILE) TYPE(QUAL1) MIN(1)      +
                        PROMPT('To file')

      PARM          KWD(TOMBR) TYPE(*NAME)  LEN(10)     +
                        SPCVAL((*FIRST)) EXPR(*YES) MIN(1)  +
                        PROMPT('To member')

QUAL1:    QUAL      TYPE(*NAME) LEN(10)

      QUAL          TYPE(*NAME) LEN(10) DFT(*LIBL)      +
                        SPCVAL((*LIBL) (*CURLIB))      +
                        PROMPT('Library')
```

データベースへの光ディスクのコピー: CL プログラム・ソース

この CL 例は、光ディスク・ボリュームのファイルをデータベース上の既存のファイルのメンバーにコピーするのに使用できます。

```
/*
/* *****
/*
/* PROGRAM: CPYOPTDB (Copy Optical to Database)
/*
/*
/*
/* DESCRIPTION:
/* This is the CL program for sample CL command CPYOPTDB. This
/* program can be used to copy a file which is on optical
/* storage to a member of an existing file.
/*
/*
/*
/* DEPENDENCIES:
/* - The sample command and program CPYSTRF exist.
/* - There is an existing folder named OPTICAL.FLR
/* This folder is used for temporary storage when copying
/* from optical to database. It is assumed that this folder is
/* empty and that the user will delete anything which gets
/* copied into it.
/*
/*
/* INPUT PARAMETERS:
/* - Complete source path
/* Example: /filesystem/volume/directory1/directoryx/file
/*          /QOPT/VOLN01/DIRA/DIRB/FILE01
/* - To file
/* Assumptions:
/* - Target library already exists.
/* - Target file already exists and has the same attributes
/* as that which contained the original file.
/* - To member
/*
/*
/*
```

```

/* LOGIC: */
/* - Build target file */
/* - Copy file from OPT to Document Library Services (DLS) */
/* - Separate file and library */
/* - Copy from folder to database file */
/* */
/* */
/* EXAMPLE: */
/* The example will copy file invoices which is in directory */
/* DEC on volume YEAR1992. INVOICES was originally a spooled file */
/* which had a record length of 133. It will be placed in file */
/* LISTINGS which is in library QUSRSYS as member INVODEC92. */
/* */
/* CPYDBOPT TGTFILE('/QOPT/YEAR1992/DEC/INVOICES') */
/* TOFILE(QUSRSYS/LISTINGS) */
/* TOMBR(INVODEC92) */
/* */
/*****/
PGM PARM(&SRCFILE &TOFILE &TOMBR);

/*****/
/* Input parameters */
/*****/
DCL VAR(&SRCFILE); TYPE(*CHAR) LEN(300)
DCL VAR(&TOFILE); TYPE(*CHAR) LEN(20)
DCL VAR(&TOMBR); TYPE(*CHAR) LEN(10)

/*****/
/* Program variables */
/*****/
DCL VAR(&FILE); TYPE(*CHAR) LEN(10)
DCL VAR(&LIB); TYPE(*CHAR) LEN(10)
DCL VAR(&TGTFILE); TYPE(*CHAR) LEN(28) +
VALUE('/QDLS/OPTICAL.FLR/xxxxxxxxxx')

/*****/
/* Monitor for all messages signalled */
/*****/
MONMSG MSGID(CPF0000) EXEC(GOTO CMDLBL(DONE))
MONMSG MSGID(IWS0000) EXEC(GOTO CMDLBL(DONE))
MONMSG MSGID(OPT0000) EXEC(GOTO CMDLBL(DONE))

/*****/
/* Build the target file name and copy the stream file from */
/* optical to DLS */
/*****/
CHGVAR VAR(%SST(&TGTFILE 19 10)) VALUE(&TOMBR);

CPYSTRF SRCFILE(&SRCFILE); +
TGTFILE(&TGTFILE);

/*****/
/* Separate the file and library names. Copy the folder to DB. */
/*****/
CHGVAR VAR(&FILE); VALUE(%SST(&TOFILE 1 10))
CHGVAR VAR(&LIB); VALUE(%SST(&TOFILE 11 10))

CPYFRMPCD FROMFLR(OPTICAL.FLR) +
TOFILE(&LIB/&FILE); +
FROMDOC(&TOMBR); +
TOMBR(&TOMBR); +
TRNTBL(*NONE)

SNDPGMMSG MSG('CPYOPTDB completed successfully')
RETURN

```

DONE:

SNDPGMMSG MSGID(OPT0125) MSGF(QSYS/QCPFMSG) +
MSGDTA(CPYOPTDB) MSGTYPE(*ESCAPE)
RETURN

ENDPGM

光ディスク装置プログラミングの関連情報

Web サイトおよびその他のインフォメーション・センターのトピック・コレクションに、光ディスク装置プログラミングのトピックに関連する情報が含まれています。任意の PDF ファイルを表示または印刷することができます。

Web サイト

- 光ディスク・ストレージ (www-03.ibm.com/servers/eserver/series/optical) 
- 光ディスク・ストレージ: System i アプリケーション・ソフトウェア (www.ibm.com/servers/eserver/series/optical/applications/applications.htm) 

その他の情報

- 統合ファイル・システム
- 光ディスク記憶
- アプリケーション・プログラミング・インターフェース

関連資料

1 ページの『光ディスク装置プログラミングの PDF ファイル』
本書の PDF ファイルを表示および印刷することができます。

コードに関するライセンス情報および特記事項

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権を許諾します。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。

いかなる場合においても、IBM および IBM のサプライヤーならびに IBM ビジネス・パートナーは、その予見の有無を問わず発生した以下のものについて賠償責任を負いません。

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。

〒106-8711
東京都港区六本木 3-2-12
IBM World Trade Asia Corporation
Intellectual Property Law & 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。

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 (光ディスク装置プログラミング) 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。

BladeCenter
i5/OS
IBM
IBM (ロゴ)
Integrated Language Environment
System i
System x

Adobe、Adobe ロゴ、PostScript、および PostScript ロゴは、Adobe Systems Incorporated または子会社の米国およびその他の国における商標または登録商標です。

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。

UNIX は、The Open Group の米国およびその他の国における登録商標です。

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、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