

IBM

@server

iSeries

OS/400 ネットワーク・ファイル・システム・サポート

バージョン 5

SC88-4019-00
(英文原典：SC41-5714-02)





@server

iSeries

OS/400 ネットワーク・ファイル・システム・サポート

バージョン 5

SC88-4019-00

(英文原典：SC41-5714-02)

お願い

本書、および本書によってサポートされる製品 (プロダクト) をご使用になる前に、113 ページの『特記事項』を必ずお読みください。

本書は、SC88-5605-00 の改訂版です。

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)

原 典： SC41-5714-02
iSeries
OS/400 Network File System Support
Version 5

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 2002.8

この文書では、平成明朝体™W3、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™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 1999, 2002. All rights reserved.

© Copyright IBM Japan 2002

目次

図	v
表	vii
OS/400 ネットワーク・ファイル・システム・サポート (SC88-4019) について	ix
本書の対象読者	ix
前提条件および関連情報	x
iSeries ナビゲーター	x
第 1 章 ネットワーク・ファイル・システムとは何か	1
紹介	1
沿革	3
ファイル・システムとしてのネットワーク・ファイル・システム	3
状態なしのネットワーク・プロトコル	4
TULAB シナリオの概説	5
第 2 章 ネットワーク・ファイル・システムのクライアント / サーバー・モデル	7
ネットワーク・ファイルのクライアント / サーバー通信の設計	8
ネットワーク・ファイル・システムのプロセスのレイアウト	9
ネットワーク・ファイル・システムの階層について	9
ネットワーク・ファイル・システムのサーバーとしての iSeries	10
ネットワーク・ファイル・システムのサーバー側のデーモン	11
ネットワーク・ファイル・システムのクライアントとしての iSeries	12
ネットワーク・ファイル・システムのクライアント側のデーモン	13
NFS クライアント側のキャッシュ	14
第 3 章 NFS とユーザー定義ファイル・システム (UDFS)	17
ユーザー・ファイル・システム管理	17
ユーザー定義のファイル・システムの作成	17
ユーザー定義ファイル・システムの表示	19
ユーザー定義ファイル・システムの削除	20
ユーザー定義ファイル・システムのマウント	21
ユーザー定義ファイル・システムのマウント解除	22
ユーザー定義ファイル・システムの保管と復元	23
iSeries ナビゲーターでの UDFS 操作の実行	23
iSeries ナビゲーターでの UDFS のマウント	25
ネットワーク・ファイル・ファイル・システムにおけるユーザー定義ファイル・システムの機能	26

ユーザー定義ファイル・システムと補助記憶域ブールの使用	26
-----------------------------	----

第 4 章 ファイル・システムからサーバーをエクスポートする	29
エクスポートとは何か	29
なぜエクスポートが必要なのか	30
TULAB のシナリオ	31
どのファイル・システムをエクスポートできるか	31
ファイル・システムをエクスポートする方法	33
ファイル・システムをエクスポートする際の規則	33
CHGNFSEXP (ネットワーク・ファイル・システム・エクスポートの変更) コマンド	35
iSeries ナビゲーターからのエクスポート	39
エクスポート対象の検索	43
エクスポートに関する考慮事項	45
第 5 章 クライアントでファイル・システムをマウントする	47
マウントとは何か	47
なぜファイル・システムをマウントするのか	50
どのようなファイル・システムをマウントできるか	50
ファイル・システムをどこにマウントできるか	51
マウント・ポイント	54
ファイル・システムをマウントする方法	54
ADDMFS (マウント済みファイル・システムの追加) コマンド	54
RMVMFS (マウント済みファイル・システムの除去) コマンド	57
DSPMFSINF (マウント済みファイル・システム情報の表示) コマンド	59
第 6 章 ネットワーク・ファイル・システムを iSeries のファイル・システムと一緒に使用する	63
「ルート」ファイル・システム (/)	64
ネットワーク・ファイル・システムでの相違	64
オープン・システム・ファイル・システム (QOpenSys)	65
ネットワーク・ファイル・システムでの相違	65
ライブラリー・ファイル・システム (QSYS.LIB)	66
ネットワーク・ファイル・システムでの相違	66
文書ライブラリー・サービス・ファイル・システム (QDLS)	69
ネットワーク・ファイル・システムでの相違	69
光ディスク・ファイル・システム (QOPT)	70
ネットワーク・ファイル・システムでの相違	70
ユーザー定義ファイル・システム (UDFS)	72
ネットワーク・ファイル・システムでの相違	72
UNIX クライアントの管理者	73

ネットワーク・ファイル・システムでの相違	73
第 7 章 NFS の開始、遮断、および回復	75
TCP/IP の構成	75
不適切な開始および遮断によって起こる不具合	76
適切な開始シナリオ	77
STRNFSSVR (ネットワーク・ファイル・システム・サーバーの開始) コマンド	78
適切な遮断シナリオ	80
遮断に関する考慮事項	81
ENDNFSSVR (ネットワーク・ファイル・システム・サーバーの終了) コマンド	81
iSeries ナビゲーターでの NFS の開始または停止	83
ロックと回復	85
なぜファイルをロックする必要があるか	86
ファイルをロックする方法	86
状態なしのシステムと状態ありの操作	86
RLSIFSLCK (統合ファイル・システム・ロックの解除) コマンド	87
第 8 章 統合ファイル・システムの API とネットワーク・ファイル・システム	89
エラー状態	89
ESTALE エラー状態	89
EACCES エラー状態	89
API に関する考慮事項	89
ユーザー・データグラム・プロトコル (UDP) に関する考慮事項	90
クライアント・タイムアウトの解決	90
ネットワーク・ファイル・システムでの相違	90
open()、create()、および and mkdir() API	91
fcntl() API	91
未変更の API	91
第 9 章 ネットワーク・ファイル・システムの機密保護に関する考慮事項	93

信頼できる共同体	93
ネットワーク・データの暗号化	94
ユーザー権限	95
ユーザー ID (UID)	95
グループ ID (GID)	95
ユーザー ID のマッピング	96
適切な UID マッピング	99
ファイル・システムを安全にエクスポートする	101
エクスポート・オプション	102
付録 A. よく使用されるコマンドの要約	105
付録 B. /etc ファイルについて理解する	107
/etc ディレクトリー内のファイルを編集する	107
ファイルの編集 (EDTF) コマンドを使用してストリーム・ファイルを編集する	107
PC 基本エディターを使用してストリーム・ファイルを編集する	108
NFS から UNIX エディターを使用してストリーム・ファイルを編集する	108
/etc/exports ファイル	109
/etc/exports ファイルの項目の形式	109
HOSTOPT パラメーターによる /etc/exports の形式設定例	110
/etc/netgroup ファイル	111
/etc/rpcstab ファイル	112
/etc/statd ファイル	112
特記事項	113
コードの特記事項情報	115
プログラミング・インターフェース情報	115
商標	115
参考文献	117
索引	119



1.	ローカル・クライアントとそこから見たリモート・サーバー (データをエクスポートする前)	2	31.	ローカル・クライアントがリモート・サーバーからファイル・システムをマウントする	48
2.	ローカル・クライアントとそこから見たリモート・サーバー (データをエクスポートした後)	2	32.	マウントされたファイル・システムは、ローカル・クライアントのディレクトリーを覆い隠す	48
3.	ローカル・クライアントがリモート・サーバーからデータをマウントする	2	33.	高いレベルのディレクトリー上にマウントしたローカル・クライアント	49
4.	リモート・ファイル・システムはクライアント上で機能する	3	34.	/2 ディレクトリー上にマウントしたローカル・クライアント	49
5.	TULAB のネットワーク名前空間	6	35.	ローカル・クライアントおよびリモート・サーバーのビュー	52
6.	NFS クライアント / サーバー・モデル	7	36.	クライアントが TULAB2 から /classes/class1 をマウントする	52
7.	NFS クライアント / サーバー・プロトコルの内訳	9	37.	/classes/class1 ディレクトリーが /user/work を覆う	52
8.	NFS サーバー	11	38.	リモート・サーバーが /engdata をエクスポートする	53
9.	NFS クライアント	13	39.	ローカル・クライアントが /engdata をマウント・ポイント上にマウントする	53
10.	「ユーザー定義ファイル・システムの作成 (CRTUDFS)」画面の使用法	18	40.	/engdata ディレクトリーが /user/work を覆う	53
11.	「ユーザー定義 FS (DSPUDFS) の表示」の出力 (1/2)	19	41.	「マウント済み FS の追加 (ADDMFS)」画面の使用法	55
12.	「ユーザー定義 FS (DSPUDFS) の表示」の出力 (2/2)	20	42.	「マウント済み FS の除去 (RMVMFS)」画面	58
13.	「ユーザー定義 FS の削除 (DLTUDFS)」画面の使用法	21	43.	「マウント済み FS 情報の表示 (DSPMFSINF)」画面	60
14.	「CRTUDFS (UDFS の作成)」コマンドの Windows® の場合の表示	24	44.	「マウント済み FS 情報の表示 (DSPMFSINF)」出力画面 (1/2)	61
15.	「DSPUDFS (UDFS の表示)」コマンドの Windows の場合の表示	24	45.	「マウント済み FS 情報の表示 (DSPMFSINF)」出力画面 (2/2)	61
16.	ユーザー定義ファイル・システムのマウントの Windows 表示	25	46.	NFS サーバー経由でアクセスされる「ルート」(1) ファイル・システム	64
17.	/etc/exports ファイルを使用してファイル・システムをエクスポートする	30	47.	NFS サーバー経由でアクセスされる QOpenSys ファイル・システム	65
18.	"-I" オプションを使用して動的にファイル・システムをエクスポートする	30	48.	NFS サーバー経由でアクセスされる QSYS.LIB ファイル・システム	66
19.	サーバーが情報をエクスポートする前	32	49.	NFS サーバー経由でアクセスされる QDLS ファイル・システム	69
20.	サーバーが /classes/class2 をエクスポートした後	32	50.	NFS サーバー経由でアクセスされる QOPT ファイル・システム	70
21.	エクスポート前の TULAB2 のディレクトリー・ツリー	33	51.	NFS サーバー経由でアクセスされる UDFS ファイル・システム	72
22.	TULAB2 のディレクトリー・ブランチ /classes をエクスポート	34	52.	「NFS サーバーの開始 (STRNFSSVR)」画面の使用法	79
23.	TULAB2 のディレクトリー・ブランチ /classes/class1 をエクスポート	34	53.	「NFS サーバーの終了 (ENDNFSSVR)」画面の使用法	82
24.	「NFS エクスポートの変更 (CHGNFSEXP)」画面	37	54.	NFS サーバー・デーモンの開始または停止	84
25.	iSeries ナビゲーターのインターフェース	40	55.	「NFS プロパティ」ダイアログ・ボックス	85
26.	「NFS エクスポート」ダイアログ・ボックス	41	56.	「ファイル・システム・ロックの解除 (RLSIFSLCK)」画面の使用法	88
27.	「ホスト/ネットグループの追加」	42	57.	信頼できる共同体の外側にいるクライアントが機密保護上の抜け穴を作る	94
28.	「NFS クライアントのアクセスのカスタマイズ」ダイアログ・ボックス	43			
29.	「NFS エクスポート」ダイアログ・ボックス	44			
30.	ローカル・クライアントとファイル・システムをエクスポートしたリモート・サーバー	48			

表

1. ネットワーク・ファイル・システムのアプリケーションで使用するコマンド 105

OS/400 ネットワーク・ファイル・システム・サポート (SC88-4019) について

本書の目的は、ネットワーク・ファイル・システムとは何か、どのような処理を行うのか、そして、iSeries サーバー上でどのように機能するのかを説明することです。また、NFS を使用して、機密性が高く、便利な統合ファイル・システムを作成する方法について、具体的な例を挙げて説明しています。本書の対象読者は次のとおりです。

- ネットワーク・ファイル・システムを使用して、分散ネットワークを開発しているシステム管理者
- ネットワーク・ファイル・システムを操作するユーザーまたはプログラマー

第 1 章と第 2 章では、NFS を紹介するために、そのプロトコル、コンポーネント、およびアーキテクチャーに関する背景情報と概念について説明します。これは、iSeries サーバーの動作の仕組みについては理解しているが、NFS については理解していないユーザーにとっての背景情報となります。

本書のそれ以外の章 (第 3 章～第 9 章) では、NFS が実行できる処理、およびそれを最大限に活用する方法を詳細な実例を挙げて示します。本書の全体を通じて論じている内容は、機密性が高く、ユーザー・フレンドリーな分散名前空間を構築する方法です。これには、マウント、エクスポート、および以下のトピックに関して詳細な実例を取り上げ、情報を提供しています。

- クライアント / サーバーの関係において、NFS がどのように機能するか
- iSeries ファイル・システムにおける NFS の例外事項
- NFS の開始、遮断、および回復
- ファイルのロック
- 新しい統合ファイル・システムのエラー状態、およびそれと NFS との関係
- NFS の機密保護上の考慮事項に関するトラブルシューティング手順

本書の読者は、iSeries クライアント / サーバー・モデルについての経験を持っていることが前提ですが、ネットワーク・ファイル・システムについての経験は特に必要ありません。

本書の対象読者

本書は、iSeries サーバーにおけるネットワーク・ファイル・システムについて知りたいと思っている iSeries ユーザー、プログラマー、および管理者を対象としています。内容は次のとおりです。

- NFS に関する背景となる理論と概念、およびそれらがどのように機能するか
- NFS で使用可能なコマンド、iSeries 関連の表示、およびその他の操作の例
- NFS を使用して、機密性の高い効率的な名前空間を構築する技法

前提条件および関連情報

iSeries のテクニカル情報の検索は iSeries Information Center を使用して始めてください。

Information Center へのアクセスには次の 2 通りがあります。

- 次の Web サイトから

<http://www.ibm.com/eserver/iseries/infocenter>

- 注文した OS/400 に同梱されてくる CD-ROM から

iSeries Information Center および *PDF ライブラリー CD パッケージ*, SK88-8055-01。これには、ソフトコピー・ライブラリー CD-ROM の代わりとなる PDF バージョンの iSeries マニュアル *iSeries Information Center: 補足資料 PDF CD*, SK88-8056-01 も収録されています。

この iSeries Information Center には、アドバイザーを始め、Java、TCP/IP、Web サービス、ネットワークの機密保護、論理区画、クラスター化、CL コマンド、およびシステム・アプリケーション・プログラミング・インターフェース (API) といった重要なトピックが含まれています。それ以外にも、関連する IBM レッドブック、および Technical Studio、IBM ホーム・ページなど他の IBM Web サイトへのリンクも設定されています。

ハードウェアの新しいご注文をお受けするたびに、*iSeries* セットアップおよびオペレーション ID, SK88-8058-01 をお送りいたします。この CD-ROM には、IBM @server iSeries Access for Windows と EZ セットアップ・ウィザードが収録されています。iSeries Access は、PC を iSeries サーバーに接続するのに有効なクライアントおよびサーバーの機能セットを提供します。EZ セットアップ・ウィザードは、数多くの iSeries セットアップ作業を自動化するものです。

この他の関連情報については、117 ページの『参考文献』を参照してください。

iSeries ナビゲーター

IBM iSeries ナビゲーターは、iSeries サーバーを管理するのに有効なグラフィカル・インターフェースです。iSeries ナビゲーターの機能には、システムのナビゲーション、構成、計画機能、および作業を進めていくうえでの手引きとなるオンライン・ヘルプが含まれます。iSeries ナビゲーターを使用すると、サーバーの運用および管理が容易になり、生産性を上げることができます。これは、ユーザーが OS/400 オペレーティング・システムの新しい拡張機能にアクセスするための唯一のユーザー・インターフェースです。これには、中央サーバーから複数のサーバーを管理するためのマネージメント・セントラルも含まれています。

iSeries ナビゲーターに関するさらに詳しい情報については、iSeries Information Center および次の Web サイトを参照してください。

<http://www.ibm.com/eserver/iseries/navigator/>

第 1 章 ネットワーク・ファイル・システムとは何か

紹介

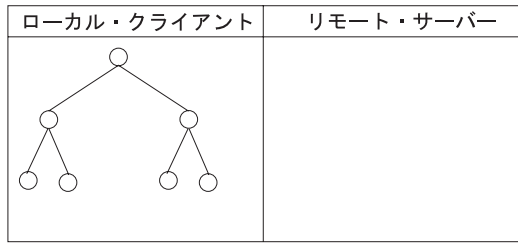
OS/400 ネットワーク・ファイル・システム・サポート は、ネットワーク・ファイル・アプリケーションおよびファイル・システムを操作するユーザーや管理者を支援する、iSeries サーバーのシステム機能を紹介します。ネットワーク・ファイル・システム (NFS**) を使用すると、すべてのユーザーが自分の必要とするデータにアクセスできるような分散ネットワーク・システムを構築できます。さらに、ネットワーク・ファイル・システムは、クライアント / サーバーの関係の中でデータを伝送する手段を提供します。

ネットワーク・ファイル・システムでは、ファイル・システムに格納されているリモート・オブジェクトをあたかもローカルであるかのように、つまりそれがローカル・ホストに存在するかのようにして扱うことができます。NFS を使用すると、ネットワーク内の全サーバーが、単一のセットのファイルを共有できます。そのため、個々のネットワーク・システムごとにファイルのコピーを重複して持つ必要がなくなります。NFS は、ユーザー、システム、およびデータの管理作業全般を支援します。

NFS は、ユーザーおよび管理者がネットワーク内にデータを分散させることができるように、以下の機能を提供します。

- **エクスポート**。ローカル・ファイル・システムをローカル・サーバーからエクスポートして、リモート・クライアントがアクセスできるようにします。この機能により、ファイル・システムの情報を中央で集中的に管理できます。NFS は、共通ディレクトリーをあらゆるサーバーで重複して持つのではなく、単一のサーバーからのディレクトリーの単一のコピーを、適切なすべてのクライアントと共有します。
- **マウント**。リモート・サーバーのファイル・システムを、ローカル・クライアントのディレクトリー上にマウントします。これにより、リモート・サーバーからエクスポートされたファイル・システムを iSeries サーバーが処理できるようになります。マウントされたファイル・システムは、あたかもローカル・サーバー上に存在するかのようにして扱うことができます。

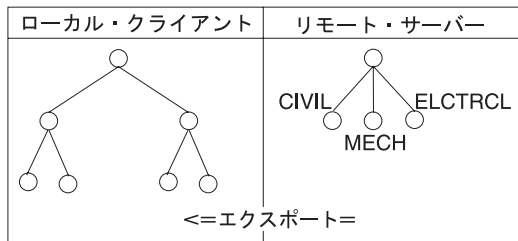
この後の図は、リモート NFS サーバーがディレクトリーをローカル・クライアントにエクスポートするプロセスを示しています。エクスポートされたディレクトリーをクライアントが一度認知した後、クライアントはそのディレクトリーを自分のローカル・ディレクトリー上にマウントします。こうして、リモート・サーバーのディレクトリーが、クライアント上でローカルに機能するようになります。



RV3N762-0

図1. ローカル・クライアントとそこから見たリモート・サーバー (データをエクスポートする前)

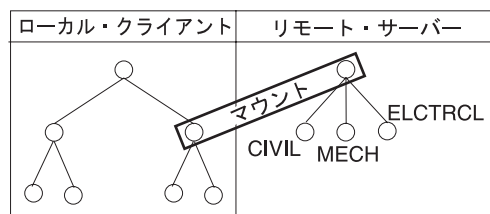
サーバーが情報をエクスポートする前は、クライアントはサーバー上のファイル・システムの存在を知りません。さらに、クライアントは、サーバー上にあるどのファイル・システムやオブジェクトについても知りません。



RV3N763-0

図2. ローカル・クライアントとそこから見たリモート・サーバー (データをエクスポートした後)

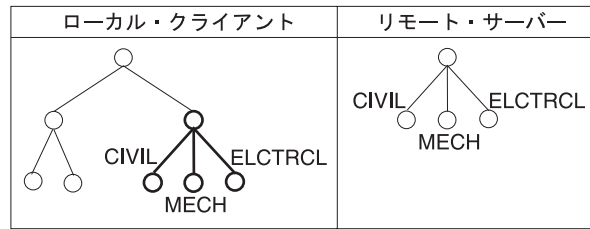
サーバーが情報をエクスポートした後、適切なクライアント(適切な権限を持つクライアント)は、サーバー上のファイル・システムの存在を認識できるようになります。さらに、クライアントは、サーバーからエクスポートされたファイル・システムやディレクトリーやオブジェクトをマウントすることができます。



RV3N764-0

図3. ローカル・クライアントがリモート・サーバーからデータをマウントする

マウント・コマンドを使用すると、特定のファイル・システム、ディレクトリー、またはオブジェクトが、クライアントからアクセスできるようになります。オブジェクトをマウントしても、そのオブジェクトがサーバーからクライアントにコピーされたり移動されたりするわけではありません。リモート・オブジェクトをローカルに使用できるようになるだけです。



RV3N765-0

図4. リモート・ファイル・システムはクライアント上で機能する

リモート・オブジェクトをローカルにマウントすると、マウント先にあったローカル・オブジェクトは、そのリモート・オブジェクトによって覆い隠されてしまいます。さらに、マウント・ポイント (サーバーへのマウントが開始される、クライアント上の地点) の下流にあるオブジェクトも、マウントされたオブジェクトによって覆い隠されます。マウントされたオブジェクトがクライアント上でローカルに機能する仕方は、サーバー上でリモートに機能する場合と同じです。

NFS のこれらの側面の詳細については、以下の項を参照してください。

- 29 ページの『第 4 章 ファイル・システムからサーバーをエクスポートする』
- 47 ページの『第 5 章 クライアントでファイル・システムをマウントする』

NFS は、TCP/IP ファイル・サーバー・サポート OS/400 用 (FSS/400用) と置き換わるシステム・アプリケーションです。FSS/400 用の作業に慣れているユーザーは、FSS/400用 と NFS の数多くの類似点に気付くはずですが。とはいえ、FSS/400 と NFS は相互に互換性のあるプロダクトではないことに注意してください。FSS/400 を NFS と同じサーバー上に置くことは可能ですが、一緒に稼働させることはできません。FSS/400 と NFS は同時に開始させないでください。

沿革

Sun Microsystems, Inc.** は、NFS を 1984 年にリリースしました。Sun は、NFS バージョン 2 を 1985 年に発表しました。1989 年には、NFS バージョン 2 について記述したコメント要求 (RFC) 規格 1094 が発表されました。X/Open は、NFS の規格である互換性のあるバージョンを 1992 年に発表しました。Sun は、NFS バージョン 3 プロトコルを 1993 年に発表しました。

Sun は NFS を UNIX** 環境で開発したため、UNIX® の多くの概念 (たとえば、UNIX 認証) が最終的なプロトコルに統合されました。それでも、依然として NFS はプラットフォームから独立しています。現在では、ほとんどすべての UNIX プラットフォームが NFS を使用しており、多くの PC、メインフレーム、およびワークステーションも NFS を使用しています。iSeries サーバーは NFS をインプリメンテーションすることにより、バージョン 2 とバージョン 3 のプロトコルをサポートします。

ファイル・システムとしてのネットワーク・ファイル・システム

iSeries のファイル・システムは、ユーザーおよびアプリケーションが記憶域の特定のセグメントにアクセスできるようにするためのサポートを提供しています。記憶域の論理単位には、以下のものがあります。

- ライブラリー

- ディレクトリー
- フォルダー

記憶域の論理単位には、以下のようなさまざまなタイプのデータを入れることができます。

- オブジェクト
- ファイル
- 文書

各ファイル・システムには、記憶域にある情報と対話するための論理構造および規則のセットがあります。それらの論理構造や規則は、ファイル・システムのタイプに応じて、ファイル・システムごとに異なります。データベース・ファイルやその他の各種オブジェクト・タイプにライブラリーを介してアクセスするという OS/400 サポートも、1 つのファイル・システムと考えることができます。同様に、フォルダーを介して文書にアクセスするという OS/400 サポートも、もう 1 つの別のファイル・システムと考えることができます。iSeries のファイル・システムの詳細については iSeries Information Center の、トピックである「**ファイル・システムおよび管理**」の「**統合ファイル・システムのファイル・システム**」を参照してください。

ネットワーク・ファイル・システムは、リモート・ファイルを「透過的に」アクセスしているように見える機能を提供します。つまり、ローカル・クライアント・ファイルと、リモート・サーバーからアクセスされるファイルとは、どちらも同じように操作できたり同じように機能したりし、双方を区別することはできません。この機能により、多数のネットワーク・クライアント全体で一貫した方式で機能するファイルおよびディレクトリーのセットを必要とするユーザーが、数多くの複雑なステップから解放されます。システム管理者の長期的な目標は、すべてのデータが自分のローカル・ワークステーション上に存在しており、自分のワークステーションで操作しているというような感覚をユーザーに抱かせる、透過的なネットワークを設計することです。また、効率的な NFS ネットワークでは、適切な人員が適切な量のデータに適切なときにアクセスできます。

ファイルやディレクトリーをクライアントが利用できるようにするには、広く普及したクライアント / サーバーの関係を経由してそれらをサーバーから**エクスポート**し、クライアントに**マウント**します。NFS クライアントは、同時に NFS サーバーとしても機能できます。同様に、NFS サーバーは、クライアントとしても機能できます。

状態なしのネットワーク・プロトコル

NFS には、クライアント / サーバー通信のためのリモート・プロシージャー呼び出し (RPC) が組み込まれています。RPC は、伝送制御プロトコル (TCP) やユーザー・データグラム・プロトコル (UDP) など、より単純なプロトコルを包含するハイエンドのネットワーク・プロトコルです。

NFS は、**状態なしのプロトコル**です。つまり、クライアント / サーバー通信に関する情報は一切保管したり保持したりしません。状態とは、何を行うのが正確に記述された要求に関する情報のことです。状態ありのプロトコルでは、ユーザーおよび要求についての情報を、多くのプロシージャーで使用するために保管します。

「状態なし」とは、ユーザーとその要求についての情報を何も保持しないことを言います。これを実現するためには、その要求に係る情報を、各要求に付随させ

て送信することが必要になります。NFS は状態なしのプロトコルなので、クライアントおよびサーバーがユーザーからの要求を処理するのに必要なすべての情報が、それぞれの RPC 要求に含まれています。

NFS を使用すると、ユーザーはネットワーク・インターフェースの詳細をバイパスすることができます。NFS は、データ通信の物理的および論理的な要素をアプリケーションから分離し、アプリケーションで各種の異なったトランスポートを使用できるようにします。

要約すれば、NFS プロトコルは、クライアント / サーバー・ネットワーク上で情報を転送する必要があるアプリケーションにとって便利なプロトコルです。RPC および NFS について詳しくは、9 ページの『ネットワーク・ファイル・システムの階層について』を参照してください。

TULAB シナリオの概説

本書では、**TULAB** という架空の名前空間を使用しながら、NFS アプリケーションの詳しい概念を説明します。**名前空間**とは、1 つ以上のサーバーが、規則正しく計画されたオブジェクト名を調べたり、管理したり、共用したりする分散ネットワーク空間のことです。

TULAB は、架空の工業大学にある、架空のコンピューター・ネットワーク環境です。これは、**ネットワーク管理者**と呼ばれる、ネットワーク構成や他のネットワーク関連情報を定義する担当者によって運営されています。この担当者は、ネットワーク・リソースが企業またはシステムでどのように利用されるかを管理します。TULAB の管理者である Chris Admin は、TULAB を使用する以下のような様々なユーザーのために、効率的で透過的な、またはシームレスな、分散型の名前空間を構築しようとしています。

- 工学系の学部学生
- 人文科学系の学部学生
- 工学系の学部卒業生
- TULAB のコンサルタント

各グループのユーザーは、いくつかのクライアント群を使用して作業します。各クライアント群は、TULAB サーバーからの異なったファイル・システムを必要とします。各ユーザー・グループは、それぞれ異なった許可と権限を持っているため、そのことが、機密保護が適切に保たれた名前空間を確立する際の課題になります。

Chris Admin は、NFS 名前空間の管理者が日常的に直面している一般的な問題に直面することでしょう。それだけでなく、一般的ではない、NFS に固有の状況にも対処しなければなりません。本書では、コマンドとそのパラメーターについて説明する際に、TULAB の場合の対応する実例を紹介します。また、NFS のアプリケーションについて説明する際には、TULAB の場合に Chris Admin が NFS をどのように構成するかを示します。

TULAB のネットワーク名前空間は、以下の 2 つの NFS サーバー・システムが関係する複雑なものです。

1. **TULAB1** - UNIX サーバー・システム
2. **TULAB2** - iSeries サーバー・システム

次の図に、TULAB 名前空間のレイアウトを示します。

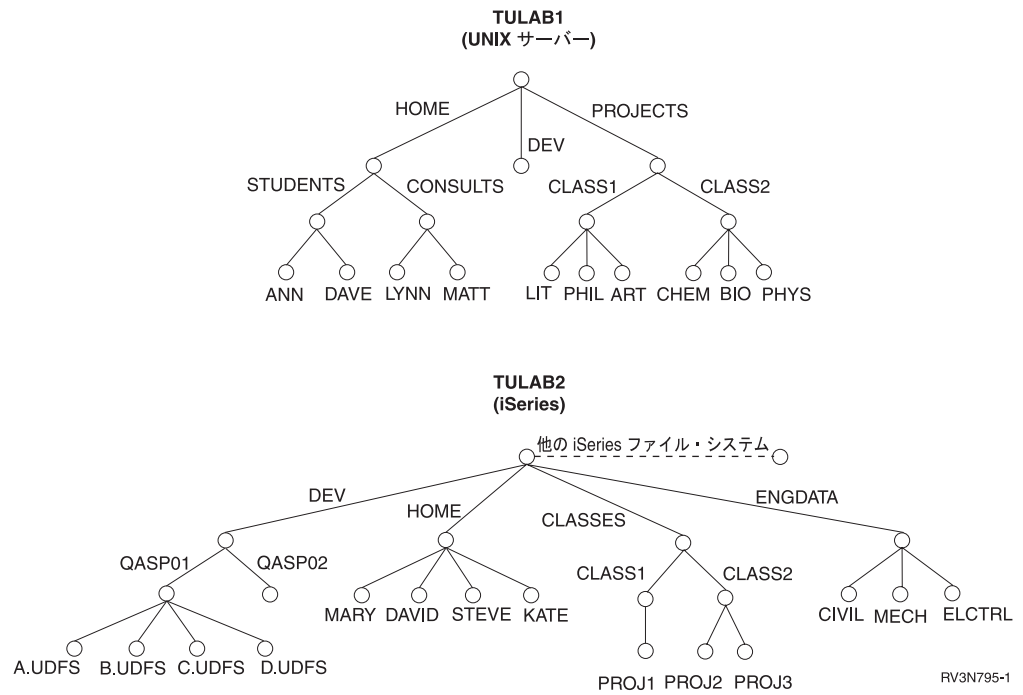


図 5. TULAB のネットワーク名前空間

第 2 章 ネットワーク・ファイル・システムのクライアント / サーバー・モデル

ネットワーク・ファイル・システムが iSeries サーバーにおいてどのように機能するかを理解するには、まず最初に、サーバーと各種クライアントとの間の通信の関係を理解しておく必要があります。クライアント / サーバー・モデルには、ローカル・ホスト (クライアント) と、ローカル・ホストから発行されるプロシーチャー呼び出しを処理する別のリモート・ネットワーク・システム (サーバー) が関係しています。別のシステムが要求を処理しているとはいえ、クライアントにとってそのプロシーチャーはローカルなものに見えます。しかし、場合によっては、1 台のコンピューターが NFS クライアントと NFS サーバーの両方の役割を果たすこともできます。

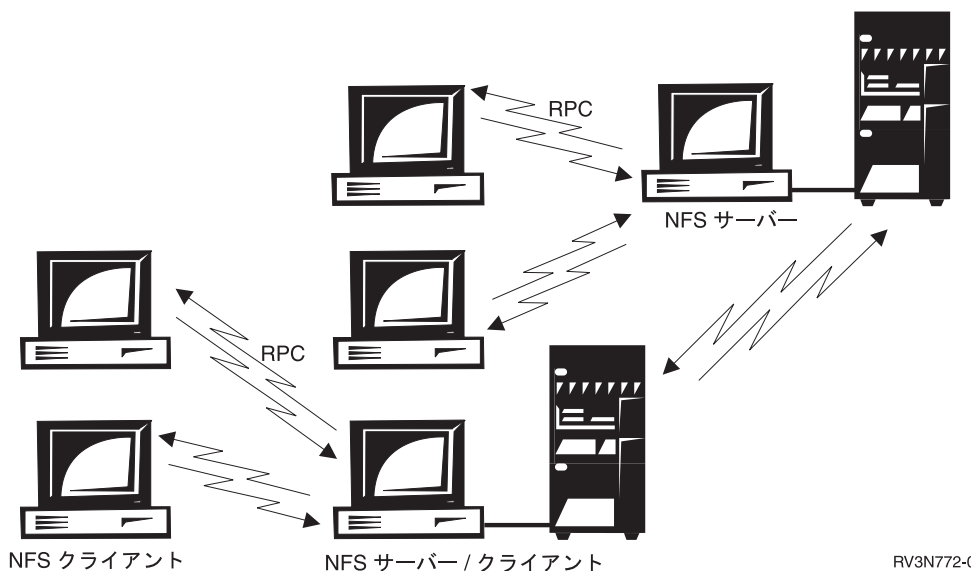


図 6. NFS クライアント / サーバー・モデル

クライアントでは利用できない各種の資源がサーバーに存在しているため、このような通信の関係が必要になります。必要な資源を所有しているホストがサーバーの役割を果たし、その資源を要求する元々の呼び出しを開始したホスト (クライアント) と通信します。NFS の場合、この資源は通常、共用ファイル・システム、ディレクトリー、またはオブジェクトです。

RPC は、そのようなクライアント / サーバーの関係を NFS 内に確立するためのメカニズムです。RPC は、プロシーチャー呼び出しに対して指定された引き数を、ネットワーク・データグラムと呼ばれるパケット・データに束ねます。NFS クライアントは、ジョブに適したサーバーに接続し、データグラムをそのサーバーに送信することによって、NFS サーバーとの RPC セッションを作成します。その後、サーバー上で引き数がアンパックされてデコードされます。命令はサーバーによって処理され、戻りメッセージ (必須) がクライアントに送り返されます。クライアントで

は、この返答が NFS に対する戻り値に変換されます。その後、処理がローカル・レベルで実施されたかのようにして、ユーザー・アプリケーションの実行が再開されます。

ネットワーク・ファイルのクライアント / サーバー通信の設計

クライアントおよびサーバー上のネットワーク・ファイル・システムの論理的なレイアウトの内訳には、数々のデーモン、キャッシュ、および NFS プロトコルが含まれています。各タイプのプロセスの概要は次のとおりです。

デーモンは、ネットワーク制御など、継続的な、またはシステム全体に関する機能を実行するプロセスです。NFS は、ユーザーからの要求を完了させるために様々なタイプのデーモンを使用します。

キャッシュは、頻繁にアクセスされる命令やデータを格納する、高速バッファ記憶域の一種です。キャッシュは、情報へのアクセス時間を短縮するために使用されます。**キャッシング**は、データをキャッシュに書き込む操作のことです。

NFS サーバーのデーモンについての情報は 11 ページの『ネットワーク・ファイル・システムのサーバー側のデーモン』を参照してください。NFS クライアントのデーモンについての情報は 13 ページの『ネットワーク・ファイル・システムのクライアント側のデーモン』を参照してください。クライアント側のキャッシュに関する詳細は、14 ページの『NFS クライアント側のキャッシュ』を参照してください。NFS プロトコルの詳細については、9 ページの『ネットワーク・ファイル・システムの階層について』で説明します。

ネットワーク・ファイル・システムのプロセスのレイアウト

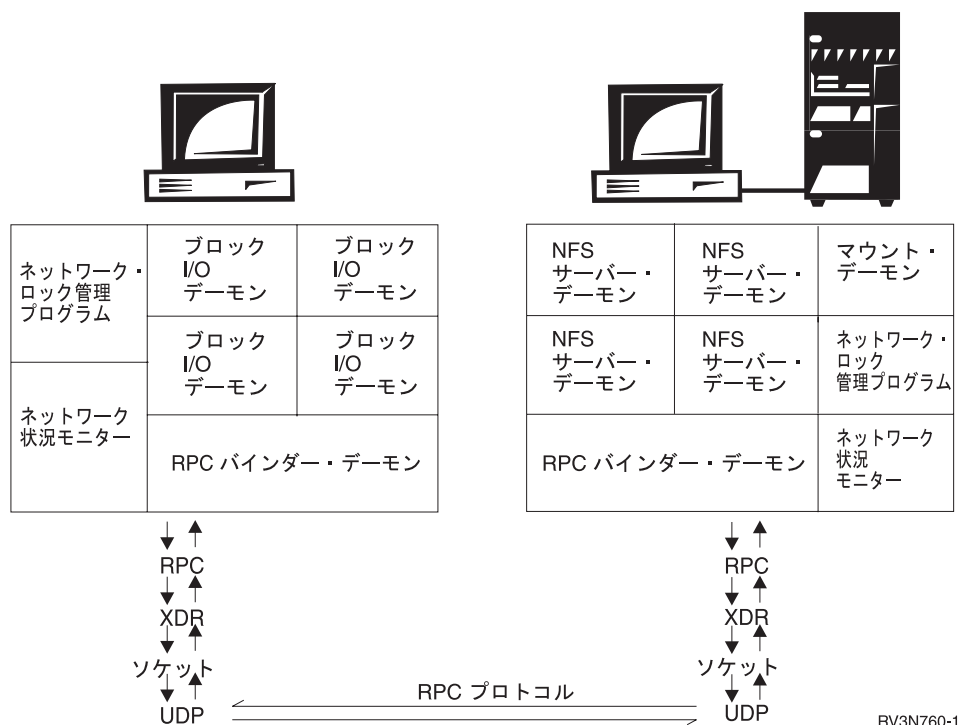


図7. NFS クライアント / サーバー・プロトコルの内訳

デーモンと呼ばれるローカル・プロセスは、クライアント側とサーバー側の両方に必要です。これらのデーモンは、ローカルおよびリモートの両方の要求を処理し、クライアント / サーバーの通信を取り扱います。NFS クライアントと NFS サーバーの両方に、ユーザーのタスクを実行する一式のデーモンがあります。それに加えて、NFS クライアントには、特定のタイプのデータをクライアント側にローカルに格納しておくためのデータ・キャッシュがあります。NFS クライアントのデータ・キャッシュの詳細については、14 ページの『NFS クライアント側のキャッシュ』を参照してください。

ネットワーク・ファイル・システムの階層について

単純なローエンド・プロトコルがいくつか集まって、NFS のように複雑なハイエンド・プロトコルを構成しています。NFS クライアントのコマンドでサーバーに接続するには、まず最初にリモート・プロシージャ呼び出し (RPC) プロトコルを使用する必要があります。その要求は、外部データ表示形式 (XDR) にエンコードされた後、ソケットを用いてサーバーに送信されます。クライアントとサーバーの間の実際の通信には、単純なユーザー・データグラム・パケット (UDP) プロトコルが使用されます。NFS の一部では、基本通信プロトコルとして伝送制御プロトコル (TCP) が使用されます。

NFS の動作は、ネットワーク・アプリケーションを特にサポートする論理上のクライアント / サーバー通信システムと見なすことができます。典型的な NFS の流れには、以下のステップが含まれます。

1. サーバーは、1 つまたは複数のクライアントからの要求を待機しています。
2. クライアントがサーバーに要求を送信し、ブロックします (応答を待つ)。
3. 要求が到着すると、サーバーはディスパッチ・ルーチン呼び出します。
4. ディスパッチ・ルーチンは、要求されたサービスを実行し、要求の結果を戻します。ディスパッチ・ルーチンは、サブルーチン呼び出して、特定の要求を処理することもできます。そのサブルーチンは、結果を直接クライアントに返す場合とディスパッチ・ルーチンに報告を戻す場合があります。
5. サーバーは、これらの結果をクライアントに戻します。
6. クライアントがブロックを解除します。

同時に複数の要求を実行することのオーバーヘッドは NFS サーバーにとって過大な負荷になるので、NFS サーバーは単スレッドになるように設計されます。したがって、NFS サーバーは、1 つのセッション当たり 1 つの要求だけしか処理できません。NFS サーバーを利用する複数のクライアントからの要求は待ち行列に入れられて、サーバーが受け取った順序で処理されます。スループットを向上させるには、複数の NFS サーバーが同一の待ち行列にある要求を処理するようにできます。

ネットワーク・ファイル・システムのサーバーとしての iSeries

NFS サーバーは数多くの別個のエンティティで構成され、それらのエンティティが協働してリモート呼び出しやローカル要求を処理します。それには、以下のようなものがあります。

- **NFS サーバー・デーモン**。これらのデーモンは、リモート・クライアントからローカル・ファイルへのアクセス要求を処理します。特定のデーモンの複数インスタンスを同時に実行できます。
- **エクスポート・コマンド**。このコマンドは、ローカル・ディレクトリーをリモート・クライアントからアクセスできるようにします。
- **/etc/exports ファイル**。このファイルには、NFS サーバーが始動時に自動的にエクスポートしたローカル・ディレクトリー名が入っています。管理担当者がこのファイルを作成して保守します。このファイルは、エクスポート・コマンドによって読み取られます。/etc/exports ファイルについては、109 ページの『/etc/exports ファイル』および 29 ページの『第 4 章 ファイル・システムからサーバーをエクスポートする』を参照してください。
- **エクスポート・テーブル**。このテーブルには、現時点でサーバーからエクスポートされているすべてのファイル・システムが入っています。エクスポート・コマンドが、/etc/exports ファイルからエクスポート・テーブルを作成します。ユーザーは、エクスポート・コマンドを使用してエクスポート・テーブルを動的に更新することができます。

「ネットワーク・ファイル・システムのエクスポートの変更 (CHGNFSEXP)」および「ファイル・システムのエクスポート (EXPORTFS)」コマンドに関する説明と、それらのコマンドと /etc/exports ファイルとの関係については、29 ページの『第 4 章 ファイル・システムからサーバーをエクスポートする』を参照してください。

ネットワーク・ファイル・システムのサーバー側のデーモン



NFS サーバー・ デーモン	NFS サーバー・ デーモン	マウント・ デーモン
NFS サーバー・ デーモン	NFS サーバー・ デーモン	ネットワーク・ ロック 管理プログラム
RPC バインダー・デーモン		ネットワーク 状況 モニター

RV3N781-0

図8. NFS サーバー

NFS は、他の RPC ベースのサービスと同様の仕方でサーバー側のデーモンを使用して、送られてくる要求を処理します。NFS は、一部のデーモンについては複数のコピーを使用して、全体としてのパフォーマンスや効率を向上させることもあります。

RPC バインダー・デーモン

RPC デーモンは、UNIX 版の多くの NFS が使用しているポート・マッパー・デーモンに類似しています。クライアントは、RPC バインダー・デーモンを使用して、指定された RPC サービスのポートを決定します。ローカル・サービスは、初期設定時にローカル RPC バインダー・デーモン (ポート・マッパー) に自らを登録します。iSeries では、ユーザーは、RPC バインダーを使用して、自分の RPC プログラムを登録できます。

NFS サーバー・デーモン

NFS サーバー・デーモンが最も差し迫って必要であるのは、NFS RPC 要求のマルチスレッド化の必要性に重点がおかれているからです。ユーザー・レベルのプロセスでデーモンを実行することにより、サーバーは複数の独立したプロセス・スレッドを持つことができるようになります。この方法により、サーバーは、複数の NFS 要求を同時に処理できます。あるデーモンが 1 つの要求の処理を完了すると、そのデーモンは、新しい要求を待っているデーモンの列の末尾に戻ります。このスケジューリング方法を使用すると、少なくとも 1 つのサーバー・デーモンが待ち行列で待っている限り、サーバーはいつでも新しい要求を受け入れることができます。このデーモンのインスタンスを複数用意すれば、複数のタスクを同時に実行できます。

マウント・デーモン

各 NFS サーバー・システムは、クライアント・システムからの要求を聴取するマウント・デーモンを 1 つ実行します。このデーモンは、クライアントからのマウントおよびマウント解除要求に応じて役割を果たします。マウント・デーモンがクライアントのマウント要求を受け取ると、デーモンはエクスポート・テーブルを調べ

ます。マウント・デーモンは、エクスポート・テーブルとマウント要求を比較して、クライアントがそのマウントの実行を許可されているかどうかを検査します。マウントが許可されていれば、マウント・デーモンは、要求を出したクライアントに不透明なデータ構造、ファイル・ハンドルを送信します。この構造は、クライアントから要求されたマウント・ポイントを一意的に記述しています。こうして、後で要求を行うときに、マウントされたファイル・システムのルートをクライアントが提示できるようになります。

ネットワーク状況モニター・デーモン

ネットワーク状況モニター (NSM) は、ネットワーク・ホストに関する情報をアプリケーションに提供する、**状態あり**の NFS サービスです。ネットワーク・ロック管理プログラム (NLM) ・デーモンは、NSM を多用して、ロックを設定したホストと、そのようなロックを保持しているホストの記録をとります。

NSM サーバーは、ホストごとに 1 つ用意されます。これは、クライアントの状態について記録を保ち、状態が変化するとき (通常、クラッシュから回復した後) に、関係する宛先にそのことを通知します。

NSM デーモンは、状態が変化した後で通知すべきホストの情報を入れた通知リストを保持しています。ローカルの状態が変化した後、NSM は通知リストに載っている各ホストに、ローカル NSM の新しい状態を通知します。NSM は、別のホストから状態変化の通知を受け取ると、その状態変化をローカルのネットワーク・ロック管理プログラム・デーモン (NLMD) に通知します。

ネットワーク・ロック管理プログラム・デーモン

ネットワーク・ロック管理プログラム (NLM) デーモンは、NFS ファイルのバイト範囲ロックの通知を提供する、**状態あり**のサービスです。NLM は、要求と要求をまたいで状態を保持し、クラッシュ前の状態を保持 (不揮発性記憶装置を利用) しているネットワーク状況モニター (NSM) を利用します。

NLM は、以下の 2 つのタイプのバイト範囲ロックをサポートしています。

1. **モニターされるロック**。これは、システム障害が発生した場合の信頼性を高めるために役立ちます。NLM サーバーがいったんクラッシュしてから回復すると、クラッシュ前にサーバーが保持していたロックはクライアントの介入なしで元通りに復元されます。同様に、クライアントがクラッシュして回復すると、NLM サーバーは古いロックをすべて解除します。モニターされるロックを作成するには、クライアントとサーバーの両方でネットワーク状況モニター (NSM) が機能していなければなりません。
2. **モニターされないロック**。このロックは、クラッシュ後に解除したり、始動後に再設定したりするのに明示的な処置が必要です。これはロックをモニターするのとは逆の選択肢で、クライアント・システムとサーバー・システムの両方に NSM が必要です。

ネットワーク・ファイル・システムのクライアントとしての iSeries

いくつかのエンティティが協働することにより、NFS クライアント上のローカル・ジョブとサーバーとの通信が実現されます。それは、以下のようなプロセスです。

- **RPC バインダー・デーモン**。このデーモンは、RPC プロトコルを使用して、ローカルおよびリモート・デーモンと通信します。クライアントは、このデーモンによって、NFS サービスを探します。
- **ネットワーク状況モニターおよび ネットワーク・ロック・マネージャー**。この 2 つのデーモンは、クライアントには必須ではありません。しかし、多くのクライアント・アプリケーションは、リモート・ファイルの各部分に対するバイト範囲ロックを、ユーザーに通知せずにクライアントのために設定します。このため、NSM および NLM デーモンは、NFS のクライアントとサーバーの両方に用意することをお奨めします。
- **ブロック I/O デーモン**。このデーモンは、データ・キャッシュを管理します。したがって、このデーモンの動作は「状態あり」です。キャッシングを実行し、クライアント側の NFS 要求をリモート NFS サーバーに転送するのを支援します。このデーモンのインスタンスを複数用意すれば、複数のタスクを同時に実行できます。
- **データ・キャッシュと属性キャッシュ**。この 2 つのキャッシュは、情報をクライアント側に格納してクライアント / サーバーの対話を減らすことにより、NFS のパフォーマンスを向上させます。属性キャッシュはファイルおよびディレクトリーの属性情報をクライアント上にローカルに格納し、データ・キャッシュは頻繁に使用されるデータをクライアント上に格納します。
- **マウント・コマンドとマウント解除コマンド**。ユーザーは、これらのコマンドを使用してクライアント名前空間にファイル・システムをマウントしたり、そのマウントを解除したりできます。これらは汎用のツールで、NFS で使用するだけでなく、その他のローカル・ファイル・システムをマウントしたりマウント解除したりするためにも使用します。「マウント済みファイル・システムの追加 (ADDMFS)」コマンド、および「マウント済みファイル・システムの除去 (RMVMFS)」コマンドについては、47 ページの『第 5 章 クライアントでファイル・システムをマウントする』を参照してください。

ネットワーク・ファイル・システムのクライアント側のデーモン



ネットワーク・ロック管理プログラム	ブロック I/O デーモン	ブロック I/O デーモン
ネットワーク状況モニター	ブロック I/O デーモン	ブロック I/O デーモン
	RPC バインダー・デーモン	

RV3N780-0

図9. NFS クライアント

NFS クライアントは、要求を処理し、リモート・サーバーとの間でデータを転送するために、RPC デーモンの他にもう 1 つのデーモンを持っています。それは、ブ

ロック I/O デーモンです。NFS では、標準的なクライアント / サーバー・モデルとは違って、NFS クライアント上のプロセスが、クライアントのブロック I/O デーモンからは独立して、独自に一部の RPC 呼び出しを発行することがあります。NFS クライアントは、オプションとしてネットワーク・ロック管理プログラム (NLM) およびネットワーク状況モニター (NSM) をローカルに使用することができますが、標準的な操作ではこれらのデーモンは必要ありません。ユーザー・アプリケーションは、しばしばユーザーに知らせずにバイト範囲ロックを設定するので、NLM および NSM の両方をクライアント上でも使用することをお奨めします。

ブロック I/O デーモン

ブロック I/O デーモンは、リモート・ファイルに対するクライアントからの要求、またはサーバーでの操作を処理します。ブロック I/O デーモンは、クライアントから、サーバー上のリモート・ファイルに対して出されるデータ要求を処理する場合があります。NFS クライアント上、またはクライアントでもある NFS サーバー上だけで実行されるこのデーモンは、ユーザーのデータ・キャッシュを管理します。ブロック I/O デーモンは「状態あり」で、クライアント・アプリケーションの要求をキャッシュまたは NFS サーバーのどちらかに転送します。ユーザーは、一定の間隔を指定して、ブロック I/O デーモンによってキャッシュされるすべてのデータを更新できます。ユーザーは、複数のデーモンを始動することにより、別々の操作を同時に実行できます。

NFS クライアント側のキャッシュ

ファイルのデータや属性をキャッシングすることは、管理担当者が NFS のパフォーマンスをチューニングする手段になります。情報をキャッシングすることより、書き込みを遅延させたり、先読みを実行したりすることができます。

NFS におけるクライアント側でのキャッシングは、サーバーに送信される RPC 要求の数を削減します。NFS クライアントでキャッシュに入れられたデータは、リモート・ディスクからではなく、ローカル・メモリーから読み出すことができます。利用可能なキャッシング・スキーマは、アクセス先のファイル・システムによって異なります。また、複数のクライアントが同時に変更や更新を行うデータの保全性および整合性を保証できない一部のキャッシング・スキーマは、使用が禁止されています。標準の NFS キャッシュ・ポリシーは、望ましいパフォーマンスを確保できるとともに、クライアント / サーバー通信の關係に状態を持ち込むことは避けています。

クライアント側のキャッシュには、ディレクトリーおよびファイル属性キャッシュと、データ・キャッシュという 2 つのタイプがあります。

ディレクトリーおよびファイル属性キャッシュ

ファイル・システムのすべての操作で、ファイルおよびディレクトリーに入っているデータが使用されるとは限りません。多くの操作では、ファイルやディレクトリーの属性 (長さ、所有者、変更日など) が取得されたり設定されたりします。このような属性だけを対象にする操作は、頻繁に発生し、ファイルやディレクトリーに入っているデータには影響を与えないため、キャッシングされた情報を利用できる第一の候補になります。

ファイル属性は、クライアント側のファイルおよびディレクトリー・キャッシュに保存されます。これは、属性の取得あるいは設定を行う操作のたびに NFS サーバ

一への接続処理を行わなくてもすむようにするためです。サーバーがファイルの属性を読み取ったとき、クライアント側では、通常 30 秒の最低期間、その属性は有効な状態を保ちます。この期間は、MOUNT コマンドの `acregmin` オプションを使用して設定できます。ファイルを変更すると、その属性が更新され、それにより、ローカル・コピーの属性が変更されて、キャッシュの有効期間が最低期間 1 つ分の長さだけ延長されます。

ファイルの属性は最大期間 (通常は 60 秒) の間には変更されない状態になっています。この期間は、MOUNT コマンドの `acregmax` オプションを使用して設定できます。この期間を過ぎると、サーバーは、キャッシュからファイル属性を除去し、変更済みファイル属性をサーバーに書き戻します。

ファイルをオープンする時点で属性を強制的に最新値に更新するには、MOUNT コマンドの `nocto` オプションを使用します。`noac` オプションを指定すると、属性のローカル・キャッシングが停止され、MOUNT コマンドの `acregmin`、`acregmax`、`acdirmin`、および `acdirmax` オプションが無効になります。

同じメカニズムがディレクトリーの属性に対しても使用されますが、有効期間はもっと長くなります。キャッシュに保存されているディレクトリー属性の最低および最高有効期間は、MOUNT コマンドの `acdirmin` および `acdirmax` オプションによって設定されます。

属性のキャッシングを利用すると、ファイルまたはディレクトリーに対して複数の変更を加える際に、毎回サーバー上の属性を取得したり設定したりする必要がなくなります。中間状態の属性はキャッシュに入れられ、属性キャッシュの最大期間が経過した時点で、すべての更新がサーバーに書き込まれます。頻繁にアクセスされるファイルおよびディレクトリーの属性はクライアント上にローカルにキャッシングされるので、一部の NFS 要求は RPC 呼び出しを行わずに実行することができます。この種のクライアント / サーバー対話を回避することにより、属性のキャッシングは NFS のパフォーマンスを向上させます。

ADDMFS および MOUNT コマンドの詳細については、47 ページの『第 5 章 クライアントでファイル・システムをマウントする』を参照してください。ADDMFS および MOUNT コマンドのオプションについて詳しくは、iSeries Information Center のトピック「CL」を参照してください。

データ・キャッシュ

データ・キャッシュは、頻繁に使用される情報をクライアント上にローカルに記憶するという点で、ディレクトリーおよびファイル属性キャッシュとよく似ています。ただし、データ・キャッシュは、ファイルやディレクトリーの属性ではなく、頻繁に使用されるデータや使用される可能性の高いデータを記憶します。クライアントが一度読み取ったことのある情報を取り出すためにサーバーにアクセスする必要が生じた場合に、データ・キャッシュからデータが提供されます。この操作によって、NFS のパフォーマンスが向上します。

ユーザーがリモート・オブジェクトを要求すると、いつもその要求はサーバーに送られます。もしその要求が少量のデータ、たとえば 1 バイト (B) のデータを読み取るという要求である場合、サーバーは 4 キロバイト (KB) のデータを戻します。この「余分な」データは、クライアントのキャッシュに記憶されます。そのデータは近いうちにクライアントによって読み取られる可能性が高いからです。

特定の期間内に各ユーザーが同じデータを頻繁にアクセスする場合、クライアントはこの情報をキャッシュに入れることにより、クライアント / サーバー間の対話を回避することができます。この種のキャッシュは、あるファイルの特定の「領域」にあるデータを頻繁に使用するユーザーにも有効です。これは**局所性**と呼ばれる状態で、サーバーから取り出される 1 次データだけでなく、その周囲のもっと大きなデータ・ブロックが関係しています。ユーザーがある特定の領域からのデータを頻繁に要求する場合は、データのブロック全体が取り出されてキャッシュに入れられます。同じユーザーがすぐにその周辺データのアクセス要求を出す確率は高いのですが、その情報はすでにクライアント上のローカル・キャッシュに入れているので、NFS のパフォーマンスが向上します。

クライアントのタイムアウト

クライアントにキャッシュがロードされていない場合は、すべての要求がサーバーに送られます。この場合、それぞれのクライアント操作を処理するのに余分の時間がかかります。ユーザーは、コマンドを再送信するまでのタイムアウト値を、マウント・コマンドで指定することができます。マウント・コマンドは、低速のサーバーと存在しないサーバーとを見分けることができないので、コマンドを再試行することになります。

省略時の再試行値は、2 秒です。この制限時間内にサーバーが応答しないと、クライアントはコマンドを再試行し続けます。ネットワーク環境では、これは、iSeries クライアントからの反復要求による過負荷を招くおそれがあります。この問題は、マウント・コマンドで指定するタイムアウト値を 5 ～ 10 秒に増やすことで解決できます。

第 3 章 NFS とユーザー定義ファイル・システム (UDFS)

ユーザー定義ファイル・システム (UDFS) は、エンド・ユーザー・インターフェースによってユーザーが直接に管理するタイプのファイル・システムです。これは、iSeries サーバーのコードが作成するシステム定義ファイル・システム (SDFS) と対照的です。QDLS、QSYS.LIB および QOPT は、いずれも SDFS の実例です。

UDFS は、ユーザーが特定の補助記憶域プール (ASP) 上に独自のファイル・システムを作成し、管理できるという概念を iSeries に導入します。ASP は、補助記憶域を構成するディスク装置またはディスク装置サブシステムから定義される記憶装置です。ASP は、特定のオブジェクトを特定のディスク装置に配置することにより、他のディスク装置のディスク媒体の障害によってデータが損失するのを防ぐ手段を提供します。

ブロック特殊ファイル (*BLKSF オブジェクト) の概念により、ユーザーは UDFS を単一のエンティティと見なすことができます。その内容は、その UDFS をローカル名前空間にマウントして初めて見えるようになります。マウントされていない UDFS は、ユーザーにとっては単一の不透明なエンティティに見えます。統合ファイル・システム・インターフェースから UDFS 内の個々のオブジェクトへのアクセスは、その UDFS がマウントされて初めて可能になります。

UDFS サポートでは、どの ASP がファイル・システムを格納するかを選んだり、大文字 / 小文字の区別などのファイル・システム属性を管理したりすることができます。マウント済みの UDFS を NFS クライアントにエクスポートして、それらのクライアントもエクスポート側の ASP に保存されたデータを共用できるようにすることが可能です。この章では、NFS によって使用できるように、UDF を作成し、操作する方法について説明します。

ユーザー・ファイル・システム管理

UDFS は、ファイルを管理する新しいストラテジーをユーザーに提供します。それには、UDFS に固有の新しい CL コマンドおよび変更された CL コマンドが含まれます。

UDFS に関連する各種の CL コマンドと、そのパラメーターおよびオプションについては、iSeries Information Center のトピック「CL」を参照してください。

ユーザー定義のファイル・システムの作成

「ユーザー定義ファイル・システムの作成」コマンドは、ファイル・システムを作成します。その内容を残りの統合ファイル・システムの名前空間から見えるようにするには、「ADDMFS (マウント済みファイル・システムの追加)」または MOUNT コマンドを使用します。UDFS は、オブジェクト・タイプ *BLKSF、つまりブロック特殊ファイルで表現されます。ユーザーは、自分で選んだ ASP 内に UDFS を作成することができ、大文字 / 小文字を区別するかどうかを指定することができます。

制約事項:

1. このコマンドを使用するには、*IOSYSCFG 特殊権限が必要です。

CRTUDFS 画面

```

                                     ユーザー定義 FS の作成 (CRTUDFS)

      選択項目を入力して、実行キーを押してください。

ユーザー定義ファイル・システム . > '/DEV/QASP02/KATE.UDFS'
データの共通認可 . . . . . *INDIR          名前 , *INDIR, *RWX, *RW...
オブジェクトの共通認可 . . . . . *INDIR          *INDIR, *NONE, *ALL...
      値の続きは+
オブジェクトの値の監査 . . . . . *SYSVAL        *SYSVAL, *NONE, *USRPRF...

                                     追加のパラメーター

大文字小文字の区別 . . . . . *MIXED          *MIXED, *MONO
省略時のファイル形式 . . . . . *TYPE2        *TYPE1, *TYPE2
テキスト ' 記述 ' . . . . . *BLANK

                                                                 終了

F3= 終了   F4=プロンプト   F5= 最新表示   F12= 取り消し
F13= この画面の使用法     F24= キーの続き
```

図 10. 「ユーザー定義ファイル・システムの作成 (CRTUDFS)」画面の使用法

CRTUDFS コマンドを使用するときには、以下のような多くのパラメーターおよびオプションを指定することができます。

- 必須指定の UDFS パラメーターでは、新しい UDFS の名前を指定します。この項目は /DEV/QASPXX/name.udfs という形式で指定しなければなりません。ここで、XX はシステムの有効な補助記憶域プール (ASP) 番号の 1 つで、name はユーザー定義ファイル・システムの名前です。パス名の残りの部分はすべて、上記の例のように指定する必要があります。パス名の name の部分は、指定した QASPXX ディレクトリー内で固有でなければなりません。
- CRTUDFS コマンドの DTAAUT パラメーターでは、新しい UDFS についてユーザーに与えられるデータの共通認可を指定します。
- CRTUDFS コマンドの OBJAUT パラメーターでは、新しい UDFS についてユーザーに与えられるオブジェクトの共通認可を指定します。
- CRTUDFS コマンドの CRTOBJAUD パラメーターでは、新しい UDFS 内に作成されるオブジェクトの監査値を指定します。
- CRTUDFS コマンドの CASE パラメーターでは、新しい UDFS で大文字 / 小文字を区別するかどうかを指定します。*MONO または *MIXED のどちらかの値を指定できます。*MONO の値を使用すると、大文字 / 小文字を区別しない UDFS を作成できます。*MIXED の値を使用すると、大文字 / 小文字を区別する UDFS を作成できます。
- DFTFILEFMT では、UDFS で作成されるストリーム・ファイル (*STMF) の形式を指定します。*TYPE1 または *TYPE2 のどちらかの値を指定できます。
*TYPE1 *STMF の値は、OS/400® のバージョン 4 リリース 4 より前のリリースの *STMF オブジェクトと同じ値を持ちます。

- CRTUDFS コマンドの TEXT パラメーターでは、新しい UDFS のテキスト記述を指定します。

例

例 1: TULAB2 のシステム ASP 内に UDFS を作成する

```
CRTUDFS UDFS('/DEV/QASP01/A.udfs) CASE(*MONO)
```

このコマンドは、A.udfs という名前の大文字 / 小文字を区別しないユーザー定義ファイル・システムをシステム補助記憶域プール (ASP) qasp01 内に作成します。

例 2: TULAB2 のユーザー ASP 内に UDFS を作成する

```
CRTUDFS UDFS('/DEV/QASP02/kate.udfs') CASE(*MIXED)
```

このコマンドは、kate.udfs という名前の大文字 / 小文字を区別するユーザー定義ファイル・システムを補助記憶域プール (ASP) qasp02 内に作成します。

ユーザー定義ファイル・システムの表示

「ユーザー定義ファイル・システムの表示 (DSPUDFS)」コマンドは、マウントされているものでも、されていないものでも、既存の UDFS の属性を表示します。

DSPUDFS 画面

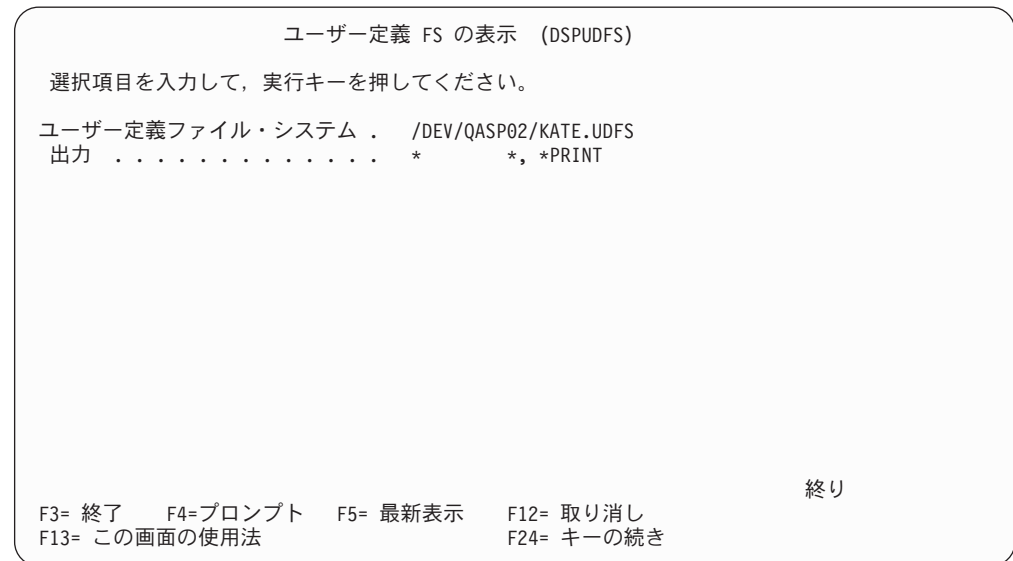


図 11. 「ユーザー定義 FS (DSPUDFS) の表示」の出力 (1/2)

DSPUDFS コマンドを使用するときに指定する必要があるパラメーターは、次の 1 つだけです。

- UDFS パラメーターでは、表示する UDFS の名前を指定します。この項目は /DEV/QASP02/name.udfs という形式で指定する (またはこのパス名に解決される) 必要があります。ここで、XX はシステムの有効な補助記憶域プール (ASP) 番号の 1 つで、name はユーザー定義ファイル・システムの名前です。パス名の残りの部分はすべて、上記の例のように指定する必要があります。

DSPUDFS コマンドを正しく使用すると、指定した UDFS に関する情報が画面に表示されます。

```
                ユーザー定義 FS の表示
ユーザー定義ファイル・システム : /DEV/QASP01/KATE.UDFS
所有者 . . . . . : NLS001
コード化文字セット ID . . . . . : 5035
大文字と小文字を区別 . . . . . : *MIXED
省略時のファイル形式 . . . . . : *TYPE2
作成日/時刻 . . . . . : 03/12/02 13:52:57
変更日/時刻 . . . . . : 03/12/02 13:52:57
取り付け位置のパス . . . . . : 装てんされていない
記述 . . . . . :
                                                                終り
続行するには、実行キーを押してください。
F3= 終了   F12= 取り消し
(C) COPYRIGHT IBM CORP. 1980, 2000.
```

図 12. 「ユーザー定義 FS (DSPUDFS) の表示」の出力 (2/2)

例

TULAB2 のユーザー ASP 内に UDFS を作成する

```
DSPUDFS UDFS('/DEV/QASP02/kate.udfs')
```

このコマンドは 補助記憶域プール (ASP) qasp02 内の kate.udfs という名前のユーザー定義ファイル・システム (UDF) の属性を表示します。

ユーザー定義ファイル・システムの削除

「ユーザー定義ファイル・システムの削除 (DLTUDFS)」コマンドは、マウントされていない既存の UDFS とその中のすべてのオブジェクトを削除します。指定した UDFS がマウントされている場合、このコマンドは実行できません。UDFS を削除すると、その UDFS 内のすべてのオブジェクトも削除されます。UDFS 内のいずれかのオブジェクトを削除するのに必要な権限を持っていないユーザーがこのコマンドを使用した場合、その UDFS 内のオブジェクトは 1 つも削除されません。

制約事項:

1. 削除しようとする UDFS は、マウントされてはなりません。
2. *IOSYSCFG 特殊権限を持つユーザーだけがこのコマンドを使用できます。

DLTUDFS 画面

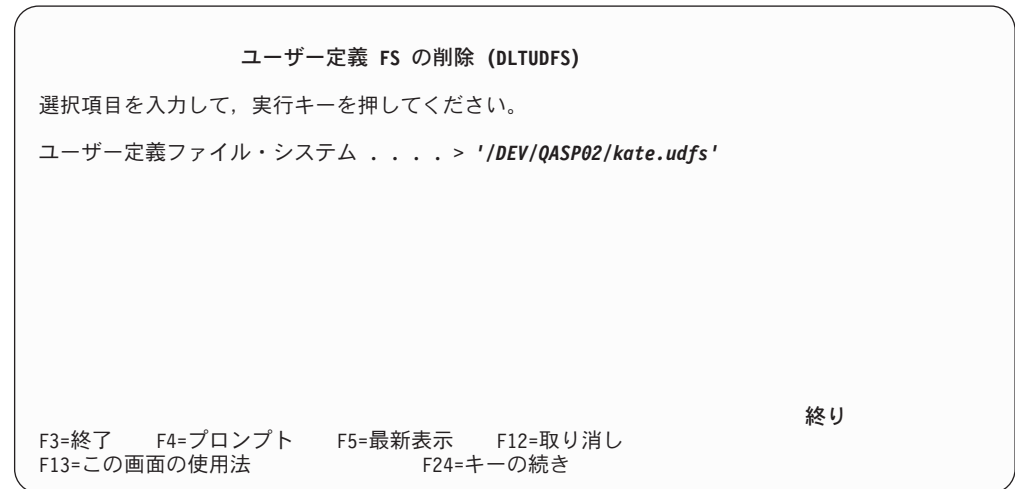


図 13. 「ユーザー定義 FS の削除 (DLTUDFS)」画面の使用法

DLTUDFS コマンドを使用するときに指定する必要があるパラメーターは、以下の 1 つだけです。

- UDFS パラメーターでは、削除する UDFS (マウントされていないもの) の名前を指定します。

この項目は /DEV/QASPXX/name.udfs という形式で指定しなければなりません。ここで、XX はシステムの有効な補助記憶域プール (ASP) 番号の 1 つで、name は UDFS の名前です。パス名の残りの部分はすべて、上記の例のように指定する必要があります。このパラメーターには、'*' や '?' などのワイルドカード文字は指定できません。指定した UDFS が現時点でマウントされている場合、このコマンドは実行できません。

例

TULAB2 のユーザー ASP 内にある UDFS をマウント解除して削除する

```
UNMOUNT TYPE(*UDFS) MFS('/DEV/QASP02/kate.udfs')
```

このコマンドは kate.udfs という名前のユーザー定義ファイル・システム (UDFS) を統合ファイル・システム名前空間からマウント解除します。UDFS を削除するには、まずその UDFS をマウント解除しなければなりません。UDFS をマウント解除した後は、DLTUDFS コマンドを使用して、その UDFS とその中のすべてのオブジェクトを削除できます。

```
DLTUDFS UDFS('/DEV/QASP02/kate.udfs')
```

このコマンドは補助記憶域プール (ASP) qasp02 内の kate.udfs という名前のユーザー定義ファイル・システム (UDF) を削除します。

ユーザー定義ファイル・システムのマウント

「マウント済みファイル・システムの追加 (ADDMFS)」および「MOUNT」コマンドは、ファイル・システム内のオブジェクトを、統合ファイル・システムの名前空

間からアクセスできるようにします。UDFS をマウントするには、ADDMFS コマンドに TYPE (*UDFS) を指定する必要があります。

ADDMFS コマンド (またはその別名 MOUNT) を使用すれば、UDFS、NFS、または NetWare いずれのファイル・システムでも動的にマウントできます。UDFS を NFS クライアントに正しくエクスポートできるようにするには、以下のステップに従ってください。

1. ブロック特殊ファイルをローカルにマウントする (タイプ *UDFS)。
2. パスを UDFS マウント・ポイント (ステップ 1 でマウント先にしたディレクトリー) にエクスポートする。

これらの一連のステップに従えば、名前空間のリモート・ビューとローカル・ビューを確実に同一にすることができます。これ以降、エクスポートした UDFS ファイル・システムは、リモート NFS クライアントによってマウントできます (タイプ *NFS)。ただし、このファイル・システムは事前にローカル名前空間にマウントしておく必要があります。

ADDMFS/MOUNT 画面

「マウント済みファイル・システムの追加 (ADDMFS)」および MOUNT コマンドの画面については、58 ページの『RMVMFS/UNMOUNT 画面』を参照してください。

例

TULAB2 に UDFS をマウントおよびエクスポートする

```
MOUNT TYPE(*UDFS) MFS('/DEV/QASP02/kate.udfs') MNTOVRDIR('/usr')
```

このコマンドは、kate.udfs という名前のユーザー定義ファイル・システム (UDFS) を TULAB2 の統合ファイル・システム名前空間のディレクトリー /usr にマウントします。

```
CHGNFSEXP OPTIONS('-I -O ACCESS=Prof:1.234.5.6')  
DIR('/usr')
```

このコマンドは、kate.udfs という名前のユーザー定義ファイル・システムをエクスポートして、適切なクライアント Prof および 1.234.5.6 から使用可能にします。

MOUNT および ADDMFS コマンドの詳細については、47 ページの『第 5 章 クライアントでファイル・システムをマウントする』を参照してください。

EXPORTFS および CHGNFSEXP コマンドの詳細については、29 ページの『第 4 章 ファイル・システムからサーバーをエクスポートする』を参照してください。

ユーザー定義ファイル・システムのマウント解除

「マウント済みファイル・システムの除去 (RMVMFS)」または「UNMOUNT」コマンドは、マウントされているファイル・システムを、統合ファイル・システムの名前空間からアクセスできないようにします。マウント解除コマンドを使用した時点でファイル・システム内のいずれかのファイルが使用中である (たとえば、ファイルがオープンされている) 場合は、ユーザーにエラー・メッセージが戻されます。

マウントしたファイル・システムそのものの上にさらにマウントした場合は、そのファイル・システムが覆われていない状態にしてからでないと、マウント解除できません。

注: クライアントによってマウントされたエクスポート済みの UDFS をマウント解除すると、次にクライアントが発行した操作がサーバーに到達した時点で、リモート・クライアントは操作が失敗したことを表す ESTALE 戻りコードを受け取ります。

RMVMFS/UNMOUNT 画面

「RMVMFS (マウント済みファイル・システムの除去)」および「UNMOUNT」コマンドの画面については、57 ページの『RMVMFS (マウント済みファイル・システムの除去) コマンド』を参照してください。

UNMOUNT および RMVMFS コマンドの詳細については、47 ページの『第 5 章 クライアントでファイル・システムをマウントする』を参照してください。

ユーザー定義ファイル・システムの保管と復元

ユーザーは、すべての UDFS とその関連権限を保管したり復元したりすることができます。「保管 (SAV)」コマンドを使用すると、UDFS 内のオブジェクトを保管することができます。また、「復元 (RST)」コマンドを使用すると、UDFS オブジェクトを復元することができます。この 2 つのコマンドは、UDFS がマウントされていてもマウントされていない場合でも正しく機能します。

iSeries ナビゲーターでの UDFS 操作の実行

iSeries ナビゲーターは iSeries サーバーのためのグラフィカル・ユーザー・インターフェース (GUI) で、ユーザー定義ファイル・システムへの簡単で便利なアクセス手段を提供します。iSeries ナビゲーターを使用するユーザーは、UDFS の作成、削除、マウントおよびマウント解除を、Windows ** クライアントから実行できます。

新規ユーザー定義ファイル・システム (UDFS) を作成するには次の手順に従ってください。

1. 「iSeries ナビゲーター」で、ご使用のサーバーを展開します。
2. 「ファイル・システム」を展開します。
3. 「統合ファイル・システム」を展開します。
4. 「ルート」を展開します。
5. 「Dev」を展開します。
6. 新規の UDFS を格納したい補助記憶域プール (ASP) をクリックします。
7. 「ファイル」メニューから「新規の UDFS」を選択します。
8. 「新しいユーザー定義ファイル・システム」ダイアログで、UDFS 名、記述 (オプション)、監査値、デフォルトのファイル形式および新しい UDFS のファイルのファイル名が大文字と小文字を区別されるかどうかを指定します。

以下に、ユーザーが PC から iSeries に接続した場合に、iSeries ナビゲーターで表示される画面の例を示します。

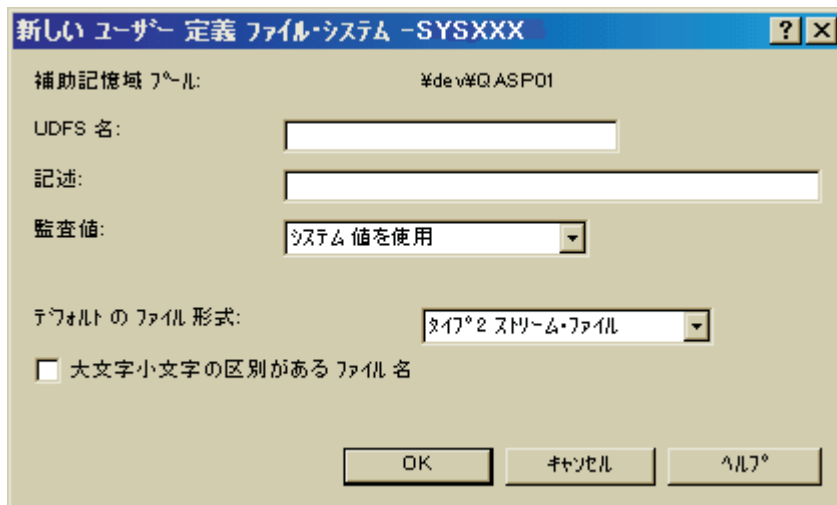


図 14. 「CRTUDFS (UDFS の作成)」コマンドの Windows® の場合の表示

iSeries ナビゲーターでは、名前、監査値、大文字 / 小文字の区別およびその他の UDFS オプションを指定できます。CRTUDFS の詳細については、17 ページの『ユーザー定義のファイル・システムの作成』を参照してください。

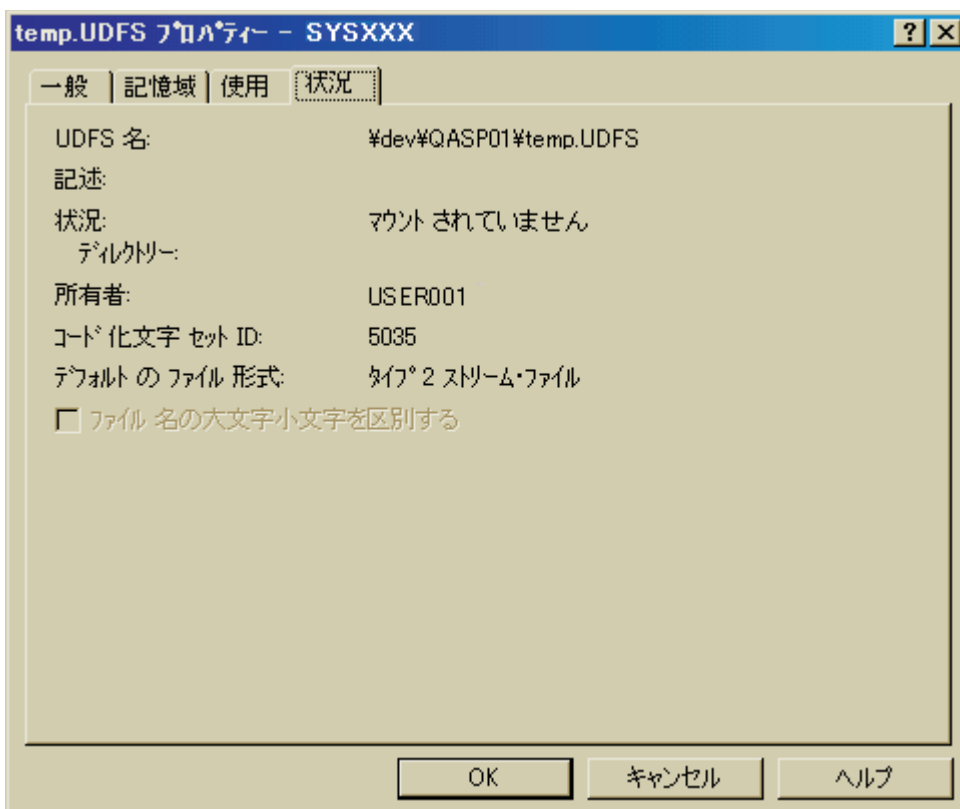


図 15. 「DSPUDFS (UDFS の表示)」コマンドの Windows の場合の表示

iSeries ナビゲーターで、UDFS のプロパティを表示できます。DSPUDFS コマンドについて詳しくは、19 ページの『ユーザー定義ファイル・システムの表示』を参照してください。

iSeries ナビゲーターでの UDFS のマウント

iSeries ナビゲーターを使用して iSeries サーバーにアクセスすると、ユーザー定義ファイル・システムを動的にマウントできます。

ユーザー定義ファイル・システム (UDFS) をマウントするには次のようにします。

1. 「iSeries ナビゲーター」で、ご使用のサーバーを展開します。
2. 「ファイル・システム」を展開します。
3. 「統合ファイル・システム」を展開します。
4. 「ルート」を展開します。
5. 「Dev」を展開します。
6. マウントしたい UDFS が入っている補助記憶域プール (ASP) をクリックします。
7. iSeries ナビゲーターの右側のペインにある **UDFS 名**カラムでマウントしたい UDFS を右マウス・ボタン・クリックして、「マウント」を選択します。

ヒント：ドラッグ・アンド・ドロップしたい場合には、同じサーバー上の統合ファイル・システム内で UDFS をフォルダーまでドラッグすることによって、マウントできます。UDFS を /dev、/dev/QASPxx、別のサーバー、またはデスクトップにドロップすることはできません。

次の画面は、ユーザー定義ファイル・システムをマウントするための「iSeries ナビゲーター」ダイアログ・ボックスです。

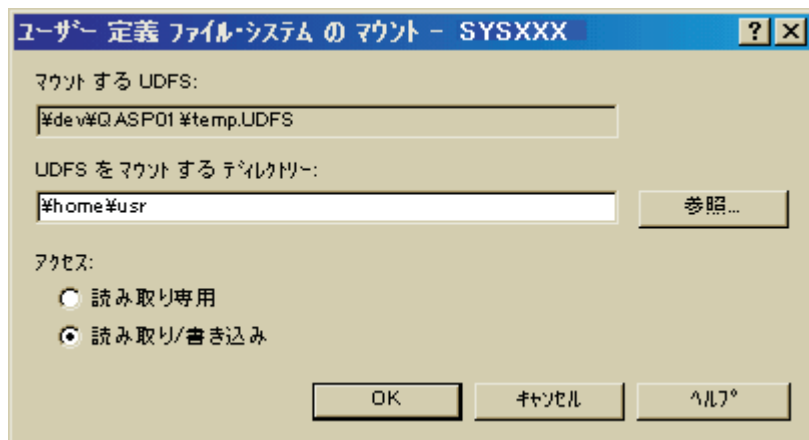


図 16. ユーザー定義ファイル・システムのマウントの Windows 表示

ネットワーク・ファイル・ファイル・システムにおけるユーザー定義ファイル・システムの機能

UDFS の内容をエクスポートするには、まず最初にその UDFS をローカル名前空間にマウントする必要があります。ブロック特殊ファイル (*BLKFS) はマウントされると、“root” または QOpenSys ファイル・システムと類似した動作をします。UDFS の内容は、サーバーからエクスポートされた時点で、リモート・クライアントに見えるようになります。

マウント解除された UDFS (*BLKSF object) またはそれが常駐している ASP はエクスポートできます。ただし、リモート NFS クライアントからの、そのようなオブジェクトの使用には制限があります。ほとんどの UNIX クライアントからは、最低限のマウントおよび表示機能の使用になります。*BLKSF オブジェクトを iSeries クライアントにマウントしたり、NFS がマウントされているディレクトリーでそれらを操作したりすることはできません。したがって、/DEV またはそれに含まれるオブジェクトをエクスポートすると、管理が困難になります。次のセクションでは、このようなシナリオに対処する方法について説明します。

ユーザー定義ファイル・システムと補助記憶域プールの使用

このシナリオには、1 人の仕事熱心なユーザー、連絡不行届きのシステム管理者、そして ASP の問題をネットワーク・ファイル・システム経由で解決する方法が関係しています。

Jeff という名前の 1 人のユーザーは、リモート NFS クライアント上の自分のアカウントにログインするたびに、TULAB2 名前空間にアクセスして操作の対象にしています。この名前空間には、/DEV/QASP02/ として名前空間に接続された ASP 内に、いくつかのユーザー定義ファイル・システム (A.udfs、B.udfs、C.udfs および D.udfs) が存在しています。Jeff は毎日、これらのディレクトリーをいつもの形式で作業の対象にしています。

ある日、システム管理者が、これらの UDFS を削除し、QASP02 をサーバーから物理的に除去してしまいます。その後、Jeff がログインすると、いつもの UDFS を見付けることができません。そこで、有能なユーザーである Jeff は、CRTDIR または MKDIR コマンドを使用して /DEV/QASP02/ ディレクトリーを作成し、以前のサブディレクトリーに似せていくつかのサブディレクトリーも作成します。Jeff は、A.udfs、B.udfs、C.udfs および D.udfs を 1.udfs、2.udfs、3.udfs および 4.udfs に置き換えます。

これは、ネットワークにとって問題になります。なぜなら、ユーザーに間違った印象を与え、サーバーにとって不都合だからです。本物の ASP ディレクトリー (/DEV/QASPXX) は、IPL 時にシステムによってのみ作成されるものなので、Jeff が新しく作成したディレクトリーは本物の ASP の代用にはなりません。また、それは本物の ASP ディレクトリー (/DEV/QASPXX) ではないので、Jeff が作成した新しい項目は、すべてシステム ASP 上のディスク容量や他の資源を占めることになり、Jeff が存在すると思込んでいる QASP02 上にはありません。

さらに、Jeff が作成したオブジェクトは UDFS ではないため、彼が予想するのとは異なる特性を持つことがあります。たとえば、彼の作成した偽物の /DEV/QASP02 ディレクトリーでは、CRTUDFS コマンドを使用することができません。

このとき、システム管理者が、IPL を実行するためのシステム遮断をせずに、本物の ASP を追加することに決めたとします。次回の IPL の際に、新しい ASP は、Jeff の偽物の /dev/qasp02 ディレクトリー上にマウントされます。Jeff や他の多くのユーザーは、自分たちのディレクトリーに突然アクセスできなくなるので、パニックに陥ります。システムが実行したマウントによって、ディレクトリーが「覆い隠されて」しまったのです。この新しい ASP は、RMVMFS コマンドや UNMOUNT コマンドを使用してもマウント解除することができません。サーバー側の Jeff やその他のユーザーには、偽物の ASP ディレクトリー (1.udfs、2.udfs、3.UDFS および 4.udfs など) にある自分達のディレクトリーやオブジェクトにアクセスする手段がありません。

ネットワーク・ファイル・システムの回復

NFS プロトコルは、マウント・ポイントをまたいで適用されることはありません。前述の問題の解決策を理解するには、この概念が重要になります。システムが実行したマウントによって覆い隠された偽物の ASP と偽物の UDFS は、サーバー側のユーザーからは見えませんが、これらのオブジェクトは依然として存在しており、NFS を使用するリモート・クライアントからはアクセスすることができます。回復プロセスには、クライアント側とサーバー側の両方で行う処置が含まれます。

1. 管理者は、偽物の ASP (および、その「下流」にあるすべてのもの) の上のディレクトリーを、EXPORTFS コマンドでエクスポートすることができます。/DEV をエクスポートすると、その下にある偽物の ASP ディレクトリーがエクスポートされ、そのディレクトリーの上にマウントされてしまった本物の ASP ディレクトリーはエクスポートされません。NFS はマウント・ポイントを越えて有効になることはないので、NFS は下にあるディレクトリーおよびオブジェクトだけを認識します。

```
EXPORTFS OPTIONS('-I -O ROOT=TUclient52X') DIR('/DEV')
```

2. エクスポートされたディレクトリーをクライアントがマウントできるようになったので、それをクライアント上の便宜的なディレクトリー (/tmp など) 上にマウントします。

```
MOUNT TYPE(*NFS) MFS('TULAB2:/DEV') MNTOVRDIR('/tmp')
```

3. マウントされたファイル・システムに対してクライアントで WRKLNK コマンドを使用すれば、偽物の ASP ディレクトリーにクライアントからアクセスできるようになり、接続されたディレクトリーを保守できるようになります。

```
WRKLNK '/tmp/*'
```

4. 次に、サーバー側では、何かの都合のよいディレクトリー (たとえば /safe など) をエクスポートする必要があります。このディレクトリーが、偽物の ASP ディレクトリーとその内容の永続的な所在場所になります。

```
EXPORTFS OPTIONS ('-I -O ROOT=TUclient52X') DIR('/safe')
```

5. クライアントは、サーバーからディレクトリー /safe をマウントすることにより、偽物の ASP ディレクトリーとその内容のための最終的な保管場所を提供できます。

```
MOUNT TYPE(*NFS) DIR('/safe') MNTOVRDIR('/user')
```

6. 最後に、クライアント側で、サーバー上の偽物の ASP ディレクトリーと、偽物のオブジェクト 1.udfs、2.udfs、3.udfs および 4.udfs を、クライアントにマウントした /safe ディレクトリーにコピーすることができます。

```
COPY OBJ('/temp/*') TODIR('/user')
```

これで、偽物の QASP02 ディレクトリーと、その下に作成された偽物のオブジェクトを、サーバーのユーザーがアクセスできるようになります。ただし、今度は、オブジェクトはサーバーの /safe ディレクトリー上にあります。

第 4 章 ファイル・システムからサーバーをエクスポートする

ネットワーク・ファイル・システムの主要な機能は、各種のローカル・ファイル・システム、ディレクトリー、およびオブジェクトを、エクスポート・コマンドによってリモート・クライアントが利用できるようにすることです。エクスポートは、クライアントとサーバー間の「透過的な」関係をセットアップするための最初の主要なステップです。

サーバーからのエクスポートが行われる前には、リモート・クライアントは、ローカル・サーバー上の特定のファイル・システムを「見る」つまりアクセスすることはできません。さらに言えば、リモート・クライアントからは、サーバー上のファイル・システムの存在をまったく認知することができません。クライアントは、どのような方法によっても、サーバーのファイル・システムをマウントして、操作の対象にすることができません。エクスポートの後には、サーバーから許可されたクライアントは、サーバーのファイル・システムをマウントして操作の対象にすることができるようになります。エクスポートされてマウントされたファイル・システムは、あたかもそれがローカル・ワークステーション上に存在しているかのように稼働します。エクスポート機能により、NFS サーバーの管理者は、正確にどのファイル・システムをアクセス可能にするか、およびどのクライアントからそのファイル・システムをアクセスできるようにするかなど、様々な管理を行うことができます。

エクスポートとは何か

エクスポートとは、ユーザーがローカル・サーバーのファイル・システムをリモート・クライアントからアクセスできるようにする処理のことです。リモート・クライアントは、適切な権限とアクセス ID を持っていれば、エクスポートされたサーバー・ファイル・システムを見たりアクセスしたりすることができます。

CHGNSFSEXP コマンドまたは EXPORTFS コマンドのどちらかを使用して、リモート・クライアントにエクスポートするためのエクスポート・テーブルに、`/etc/exports` ファイルからディレクトリー名を追加することができます。また、これらのコマンドを使用することにより、`/etc/exports` ファイルをバイパスして動的に NFS サーバーからエクスポートを実行することもできます。

ネットワークを経由してエクスポートできるようなファイル・システムを持つホスト・システムが、NFS サーバーになります。サーバーは、これらのファイル・システムをネットワーク・システム全体に公示するわけではありません。そうではなく、エクスポートされたファイル・システムについてのオプションと、関連するアクセス権限や制限のリストを、`/etc/exports` ファイルに保持します。`/etc/exports` ファイルは、エクスポート・コマンドにより、エクスポート・テーブルに組み入れられます。エクスポート・コマンドはエクスポート・オプションを読み取り、コマンドが使用される時点で、エクスポートするファイル・システムに対してそのオプションを適用します。ファイル・システムをエクスポートするもう 1 つの方法として、個々のファイル・システムごとに、エクスポート・コマンドの「-I」オプション

を使用する方法もあります。そのコマンドは、`/etc/exports` ファイルにある情報を処理しません。

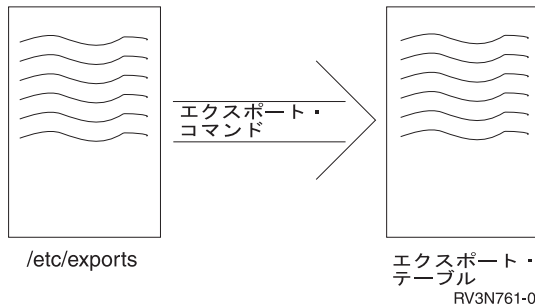


図 17. `/etc/exports` ファイルを使用してファイル・システムをエクスポートする

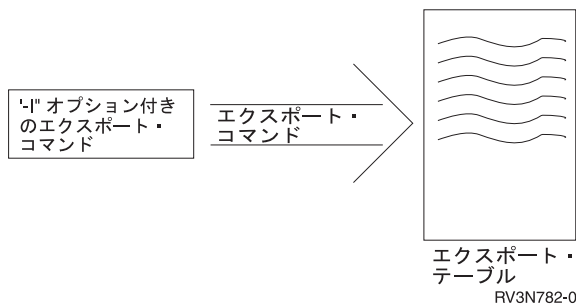


図 18. `"-f"` オプションを使用して動的にファイル・システムをエクスポートする

マウント・デーモンは、エクスポートされたファイル・システムをマウントする要求がクライアントから発行されるたびに、エクスポート・テーブルを調べます。適切な権限を持つユーザーは、`/etc/exports` ファイルの項目を追加、削除、または変更することにより、任意のファイル・システムをエクスポートするようにそのファイルを更新することができます。その後、エクスポート・コマンドを使用してエクスポート・テーブルを更新します。ほとんどのシステム管理者は、NFS サーバーが、開始時にただちに処理する `/etc/exports` ファイルの存在を確認できるようにそのサーバーを構成します。これを実現するために、管理者は、「`STRNFSSVR` (ネットワーク・ファイル・システム・サーバーの開始)」コマンドに `*ALL` を指定します。サーバーは、この方法で `/etc/exports` ファイルを見つけた後、エクスポート・コマンドを使用してエクスポート・テーブルを作成します。これで、そのファイル・システムは即時にクライアントから使用できるようになります。

なぜエクスポートが必要なのか

エクスポート機能は、任意のファイル・システムをクライアントからアクセスできるように指定する簡単な手段をシステム管理者に提供します。管理者は随意にエクスポートを実行することにより、ファイル・システムを利用可能にする相手とその方法を指定して、特定のユーザーやユーザー・グループの必要を満たすことができます。

エクスポートという効率的なシステムを活用すれば、クライアント・システムの各グループに必要な構成ファイルと始動ファイル、保存ファイル、およびアプリケー

ションを、各グループが 1 組だけ持つようにすることができます。エクスポート・コマンドを使用すると、これらのタイプのデータをクライアントからアクセスできるように任意の時点で設定できます。

ファイル・システムをエクスポートする手段には安全性の低いものが多い中で、エクスポート・コマンドのオプションを使用する方法は、ファイル・システムをエクスポートするための安全性の高い手段です。エクスポートされたファイル・システムの利用を、ネットワーク名前空間の信頼できる共同体に属するシステム・グループだけに限定することができます。

エクスポート機能は、クライアントのセットアップから、機密データをアクセスするのに必要な権限の判別まで、名前空間を管理するための簡単で効率的な手段を提供できます。/etc/exports ファイルを適切に使用すると、名前空間の安全性と機密性を高めるだけでなく、すべてのユーザーの必要を満たすことができます。

TULAB のシナリオ

TULAB では、工学系の学部学生が、工学系の学部卒業生と一緒に作業しています。どちらのグループの学生も、NFS を経由してサーバー上の自分のホーム・ディレクトリーおよびアプリケーションにアクセスできます。学生たちの研究テーマは、地元の橋の構造についての歴史の変遷を比較検討することです。学生たちは、キャンパスの同じ建物内の様々な部屋で研究を行います。Chris Admin は、全員がすべてのデータをアクセスできるようにするのではなく、学生とコンピューターの 2 つのグループがそれぞれ適切なデータをアクセスできるようにするための手段を必要としています。

Chris Admin は、橋の構造の安全性に関する統計レコードの入ったデータベース・ファイルだけを含むディレクトリーをエクスポートできます。この操作は、未知のユーザーが機密データにアクセスするという危険性を心配せずに実行できます。Chris Admin は、エクスポート・コマンドを使用して、いくつかの選ばれたクライアント・システムだけがこれらのファイルにアクセスできるように指定できます。このようにして、2 つのグループの学生たちが、別々のワークステーション上で同じデータをマウントしてアクセスできるようになります。

どのファイル・システムをエクスポートできるか

エクスポートできるのは、以下のようなファイル・システムです。

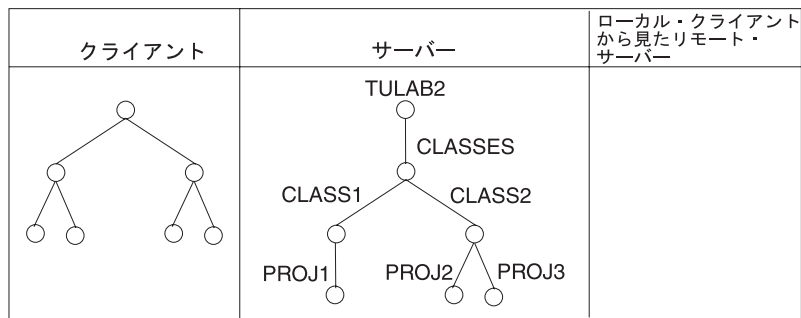
- 「ルート」 (/)
- QOpenSys
- QSYS.LIB
- QDLS
- QOPT
- UDFS

リモート・サーバーからマウントしたファイル・システムをエクスポートすることはできません。これには、ディレクトリー・ツリー全体だけでなく、個々のファイルも含まれます。iSeries サーバーは、33 ページの『ファイル・システムをエクスポートする際の規則』で詳述するエクスポートに関する一般的な規則に従います。この一連の規則には、ファイル・システムのエクスポートに関する一般的な NFS 「下流」規則も含まれています。つまり、ファイル・システムをエクスポートする

と必ず、階層上でそのファイル・システムより下（「下流」）にあるすべてのディレクトリおよびオブジェクトもエクスポートすることになります。

CHGNFSEXP または EXPORTFS コマンドは、ローカル・ユーザーのビューの一部を選択して、それをリモート・クライアントのビューの一部とするための主要な方法です。

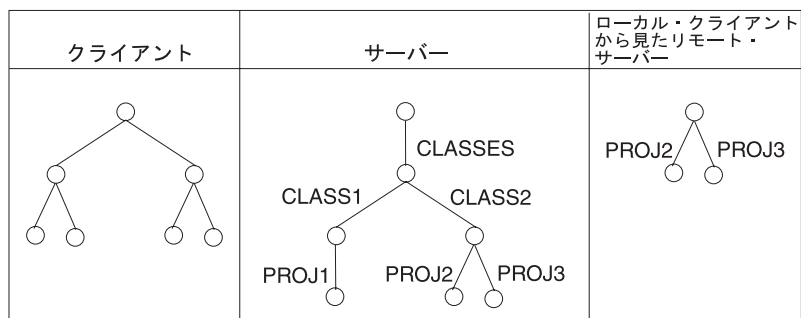
エクスポートされたファイル・システムは、`/etc/exports` ファイルから作成されるエクスポート・テーブルにリストされます。これらのファイル・システムは、クライアントが適切なアクセス権を持っている場合に、そのクライアントをマウントするためにアクセスできます。ユーザーは、エクスポートされたデータのマウントやアクセスを、あたかもそのデータがクライアント・システム上にローカルに存在しているかのようにして実行することができます。サーバーの管理者は、ファイル・システムのエクスポートを動的に変更することにより、ファイル・システムの異なったビューをリモート・クライアントに提示することもできます。



RV3N766-0

図 19. サーバーが情報をエクスポートする前

エクスポートの前には、リモート・クライアントはローカル・サーバーのファイル・システムを全く見ることはできません。



RV3N767-0

図 20. サーバーが `/classes/class2` をエクスポートした後

エクスポートの後には、リモート・クライアントはエクスポートされたファイル・システム PROJ2 および PROJ3 を見ることはできます。

サーバー上のすべてのファイル・システムがリモート・クライアントに見えるようになるわけではありません。エクスポートされたファイル・システムだけを、エクスポート・コマンドまたは `/etc/exports` ファイルで指定された適切な権限を持つクライアントから、マウントのために利用できます。リモート・クライアントは、

自分自身のローカル・ファイルと様々なリモート・サーバーからエクスポートされたファイルだけしか見ることができません。リモート・クライアントからサーバーのデータにアクセスできるようにするには、まずそのデータをエクスポートし、次にマウントしなければなりません。

ファイル・システムをエクスポートする方法

NFS サーバーのファイル・システムをクライアント上でマウントするためにエクスポートするには、「CHGTFSEXP (ネットワーク・ファイル・システム・エクスポートの変更)」、または EXPORTFS CL コマンドを使用します。

エクスポートの手順に関する特定の考慮事項については、93 ページの『第 9 章 ネットワーク・ファイル・システムの機密保護に関する考慮事項』を参照してください。

ファイル・システムをエクスポートする際の規則

NFS サーバーのファイル・システムをエクスポートしてクライアントからアクセスできるようにするには、以下の 4 つの概念上の規則が適用されます。

1. 任意のファイル・システム、またはファイル・システムの適切なサブセットは、NFS を実行しているシステムからのみエクスポートできます。ファイル・システムの適切なサブセットとは、エクスポートしようとしているファイル・システムのパスより下から始まるファイルまたはディレクトリーです。この機能により、大きなファイル・システムの全体を一度にエクスポートする代わりに、そのファイル・システムの特定の部分だけをエクスポートすることができます。

たとえば、/usr はファイル・システムで、/usr/public_html ディレクトリーはそのファイル・システムの一部だとします。このとき、/usr/public_html (および、このディレクトリー内のすべてのオブジェクト、ファイル、サブディレクトリー) は、/usr ファイル・システムの適切なサブセットです。

エクスポート処理は、川にたとえることができます。川に物を投げ入れると、それは下流に向かって流れていきます。ファイル・システムをエクスポートすると、「下流」にあるすべてのディレクトリー、ファイル・システム、およびその他のオブジェクトも一緒にエクスポートされることになります。エクスポートも、川が上流から下流に向かって流れるのと同じです。

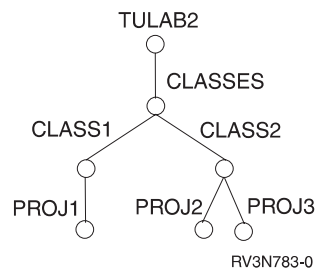


図 21. エクスポート前の TULAB2 のディレクトリー・ツリー

どのサーバー・ファイル・システムについても、エクスポートを開始すると、エクスポート・ポイントより下のすべてのオブジェクトが一緒にエクスポートされ

ます。これには、ディレクトリー、ファイル、およびオブジェクトが含まれます。

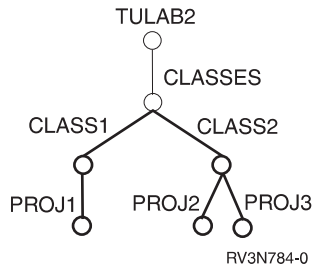


図 22. TULAB2 のディレクトリー・ブランチ /classes をエクスポート

たとえば、TULAB2 から /classes をエクスポートすると、/classes より下にあるすべてのもの、つまり /classes/class1、/classes/class2、およびその関連サブディレクトリーもエクスポートされます。もしクライアントが /classes/class1 に対するアクセスだけを必要としているのであれば、エクスポートしたデータが多過ぎることになります。

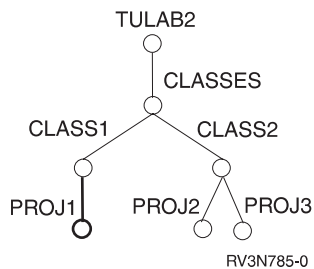


図 23. TULAB2 のディレクトリー・ブランチ /classes/class1 をエクスポート

もっと賢明な方法は、クライアントが必要とするものだけをエクスポートすることです。クライアントが /classes/class1 へのアクセスだけを必要としているのであれば、/classes/class1 だけをエクスポートします。このようにすると、クライアントの名前空間に無駄がなくなるので、クライアント上での作業が簡単になります。それに加えて、NFS サーバーにおける機密保護上の問題からの保護にもなります。

2. すでにエクスポートしたファイル・システムのサブディレクトリーをエクスポートすることはできません。ただし、そのサブディレクトリーが、別のローカル・ファイル・システム上に存在している場合は、エクスポートできます。

たとえば、/bin をすでにエクスポートした場合に、/bin/there が /bin のサブディレクトリーであるとして。その場合に、/bin/there をエクスポートすることはできません。ただし、そのサブディレクトリーが別のローカル・ファイル・システム上に存在している場合は、エクスポートできます。

3. すでにエクスポートしたファイル・システムの親ディレクトリーをエクスポートすることはできません。ただし、その親ディレクトリーが、別のローカル・ファイル・システム上に存在している場合は、エクスポートできます。

たとえば、ファイル・システム /home/sweet/home をすでにエクスポートした場合に、/home/sweet が /home/sweet/home の親ディレクトリーであるとして。

その場合に、/home/sweet をエクスポートすることはできません。ただし、その親ディレクトリーが別のローカル・ファイル・システム上に存在している場合は、エクスポートできます。

4. ローカル・ファイル・システムだけをエクスポートできます。リモート・サーバー上に存在するファイル・システム、またはそのファイル・システムのサブセットはエクスポートできません。もちろん、そのリモート・サーバー自身からエクスポートすることはできます。

たとえば、/help が、現在アクセスしているサーバーとは異なる別のサーバー上に存在しているとします。/help をエクスポートするには、そのリモート・サーバーにアクセスできなければなりません。

もっと複雑な例として、ローカル・クライアントがリモート・サーバーからマウントしたファイル・システムをエクスポートしようとする場合を考えます。たとえば、ローカル・クライアント上に /home/troupe というファイル・システムが存在し、リモート・サーバー上に /remote1 というファイル・システムが存在するとします。クライアントが /remote1 を /home/troupe にマウントすると、そのクライアントは、/home/troupe をエクスポートできなくなります。そのディレクトリーは、実際にはローカル・クライアント上ではなくリモート・サーバー上に存在するからです。

第 1 の規則により、大きなファイル・システムのうち、選択した部分だけをエクスポートすることができます。単一のファイルをエクスポートしてマウントすることができます。この機能は、ローカル・ディスクを持たないクライアントで多用されます。第 2 および第 3 の規則は、ローカル・ファイル・システムを 1 通りの方法でエクスポートできること、そしてその方法だけが有効であることを示しています。あるファイル・システムのサブディレクトリーをいったんエクスポートすると、上流にさかのぼってそのファイル・システム全体をエクスポートするようなことはできません。また、あるファイル・システム全体を公用にした場合に、そのエクスポートの下流の流れを制限して、少数のファイルだけを含むようにすることもできません。

サブディレクトリーをエクスポートすることは、リレーショナル・データベースにビューを作成することと似ています。データベースのうちユーザーが見る必要のある部分を選び、無関係な情報または機密情報は隠すようにします。このようにして、システム管理者は機密資料へのアクセスを制限できます。

CHGNFSEXP (ネットワーク・ファイル・システム・エクスポートの変更) コマンド

目的

「ネットワーク・ファイル・システム・エクスポートの変更 (CHGNFSEXP)」コマンドは、現時点で NFS クライアントにエクスポートされているディレクトリー・ツリーのリスト (エクスポート・テーブル) にディレクトリー名を追加します (エクスポート)。また、このコマンドでは、現時点で NFS クライアントにエクスポートされているディレクトリー・ツリーのリストから項目を除去することもできます (アンエクスポート)。OPTIONS リストのフラグには、CHGNFSEXP コマンドで実行する処理を指定します。CHGNFSEXP オプションの完全な解説については、iSeries Information Center の「CL」トピックを参照してください。

ファイル・システムとその内容をエクスポートするためのディレクトリーおよびオプションのリストは、`/etc/exports` ファイルに格納されます。`CHGNFSEXP` コマンドでは、`'-A'` フラグを使用すれば、そのファイルに指定されているすべてのディレクトリー・ツリーをエクスポートできます。また、`CHGNFSEXP` にディレクトリー名を指定すれば、単一のディレクトリー・ツリーをエクスポートできます。エクスポートするディレクトリー・ツリーが `/etc/exports` ファイルにある場合、それがエクスポートされる際のオプションをそのファイル内で指定することができます。または、`'-I'` フラグを使用してオプションを指定変更し、新しいオプションを `CHGNFSEXP` コマンドに指定することもできます。

さらに、`/etc/exports` ファイルに事前に定義していないディレクトリー・ツリーをエクスポートするために、`'-I'` およびオプションを `CHGNFSEXP` コマンドに指定できます。ディレクトリー・ツリーをアンエクスポートするには、`CHGNFSEXP` コマンドで `'-U'` フラグを使用します。以前にエクスポートした任意のファイル・システム (リモート・クライアントがマウントしているものでもかまわない) をアンエクスポートできます。その場合は、そのファイル・システム内のオブジェクトに対する操作を次に実行しようとした時点で、NFS サーバーからリモート・クライアントに `ESTALE` エラー番号が送られます。

特定のクライアント・グループに対してエクスポートするには、`/etc/netgroup` ファイルを使用します。このファイルには、ファイル・システムのエクスポート先になるクライアント・グループの別名が入っています。ネットグループの外にある他のシステムは、ファイル・システムにアクセスすることができません。`/etc/netgroup` ファイルについて詳しくは、111 ページの『`/etc/netgroup` ファイル』を参照してください。

このコマンドは、以下の代替コマンド名を使用して呼び出すことができます。

- `EXPORTFS`

`/etc/exports` ファイルを編集する方法については、107 ページの『ファイルの編集 (EDTF) コマンドを使用してストリーム・ファイルを編集する』を参照してください。

制約事項:

エクスポートの項目を変更する際には、以下の制約事項が適用されます。

1. このコマンドを使用するユーザーには、`*IOSYSCFG` 特殊権限が必要です。
2. このコマンドを使用するユーザーには、`/etc` ディレクトリーに対する読み取り (`*R`) および書き込み (`*W`) データ権限が必要です。

`CHGNFSEXP` および `EXPORTFS` コマンドとその関連パラメーターの詳細については、iSeries Information Center のトピック「CL」を参照してください。

CHGNFSEXP/EXPORTFS 画面

NFS エクスポートの変更 (CHGNFSEXP)

選択項目を入力して、実行キーを押してください。

NFS エクスポート・オプション > '-I -0 RO,ANON=199,ACCESS=Prof:1.234.5.6'

ディレクトリー > '/engdata/mech'

追加のパラメーター

ホスト・オプション :

ホスト名	TULAB1	Character value, *DFT
データ・ファイル CCSID	850	1-65533, *BINARY, *ASCII...
パス名 CCSID	850	1-65533, *ASCII, *JOBCCSID
強制同期書き出し	*SYNC	*SYNC, *ASYNC

値の続きは +

終り

F3=終了 F4=プロンプト F5=最新表示 F12=取り消し
F13=この画面の使用法 F24= キーの続き

図 24. 「NFS エクスポートの変更 (CHGNFSEXP)」画面

CHGNFSEXP または EXPORTFS コマンドを使用するときには、次のような多くのパラメーターおよびオプションを指定することができます。

- エクスポート・オプション・リストには、いくつかのフラグを指定し、必要ならその後に、エクスポートするディレクトリー・ツリーの特性を表す文字列のリストを指定します。それぞれのフラグは、マイナス記号 '-' とその後の 1 文字で構成されます。ここで指定するオプションは、CHGNFSEXP コマンドの OPTIONS パラメーターとして設定されます。各フラグはスペースで区切って指定します。各フラグの特定の組み合わせだけが認められます。有効でない組み合わせが検出されると、エラーが戻されます。
- ディレクトリー項目は、エクスポートしたいディレクトリーの名前です。ここで指定するパス名は、CHGNFSEXP コマンドの DIR パラメーターとして設定されます。この項目は、エクスポートする (NFS クライアントが利用できるようにする) またはアンエクスポートする (NFS クライアントが利用できないようにする) 既存のディレクトリーを指定します。このディレクトリーとして、すでにエクスポートしたディレクトリーのサブディレクトリーまたは親ディレクトリーを指定することはできません (別のファイル・システムに属しているものは例外)。OPTIONS パラメーターに -A フラグを指定した場合、このパラメーターは指定できません。
- HOSTOPT パラメーターには、ディレクトリー・ツリーのエクスポート先の NFS クライアントに関する追加情報を指定する 4 つの要素があります。ディレクトリー・ツリーのエクスポート先のホスト名に対して HOSTOPT パラメーターを指定しないと、そのホストに対しては HOSTOPT パラメーターの各要素の省略時値が使用されます。
 1. 追加オプションの指定の対象にするホストの名前。ここに指定するホストは、エクスポートされたディレクトリー・ツリーにアクセスできるホストとして、上の OPTIONS -0 リストに指定しておく必要があります。ここで指定する

オプションを関連付ける、単一のホストのアドレスを表す別名である単一のホスト名、またはネットグループ名を指定します。インターネット・アドレスに名前を割り当てるには、「TCP/IP の構成」メニュー (CFGTCPコマンド) の「TCP/IP ホスト・テーブル項目の処理」オプションを使用します。また、リモート・ネーム・サーバーを使用して、リモート・システム名をインターネット・アドレスにマッピングすることもできます。

2. ネットワーク・データ・ファイルの CCSID は、指定したホスト名 (またはネットグループ名) との間で送受信するファイルのデータに対して使用されます。HOSTOPT パラメーターで指定しなかったホストについては、省略時のネットワーク CCSID (バイナリー、変換なし) が使用されます。

新規に作成されるファイルは、値が *BINARY でなければ、この CCSID でタグ付けされます。*BINARY 値のファイルはネットワーク・パス名の CCSID でタグ付けされます。

3. ネットワーク・パス名の CCSID は、指定したホスト名 (またはネットグループ名) との間で送受信するファイルのパス名構成要素に対して使用されます。HOSTOPT パラメーターで指定しなかったホストについては、省略時のネットワークパス名 CCSID (ASCII) が使用されます。
4. 書き込みモードは、指定したホスト名 (またはネットグループ名) に関して、書き込み要求を同期的に処理するか非同期的に処理するかを指定します。省略時値の *SYNC は、データを即時にディスクに書き込むことを意味します。*ASYN 値は、データを即時にディスクに書き込むことを保証しませんが、サーバーのパフォーマンスを改善するために利用できます。

注: ネットワーク・ファイル・システムのプロトコルでは、これまで、同期書き込みを採用してきました。同期書き込みの場合は、サーバーによってデータが書き込まれる後、までサーバーの制御がクライアントに戻されません。非同同期書き込みの場合は、実際に書き込まれたかどうかは保証されません。

例

例 1: /etc/exports から全項目をエクスポートする

```
CHGNFSEXP OPTIONS('-A')
CHGNFSEXP '-A'
```

この 2 つのコマンドはどちらも、/etc/exports ファイルにあるすべての項目をエクスポートします。

例 2: オプションを指定して 1 つのディレクトリーをエクスポートする

```
CHGNFSEXP OPTIONS('-I -O RO,ANON=199,ACCESS=Prof:1.234.5.6')
DIR('/engdata/mech')
HOSTOPT((TULAB1 850 850))
```

このコマンドは、パス名 /engdata/mech より下のディレクトリー・ツリーを読み取り専用でエクスポートします。さらに、このディレクトリー・ツリーをマウントすることを 2 つのクライアントだけに許可します。キーワードを必要としない定位置パラメーターを指定しています。HOSTOPT パラメーターを使用して、ホスト TULAB1 のコード・ページを指定しています。

例 3: 多数のネットグループにディレクトリーをエクスポートする

```
CHGNFSEXP OPTIONS('-I -O ACCESS=students:LabSystems:Profs, ANON=-1')
DIR('/home')
```

このコマンドは、パス名 /home より下のディレクトリー・ツリーを、
/etc/netgroup ファイルに定義されている次の 3 つのネットグループにエクスポートします。

1. students
2. LabSystems
3. Profs

ただし、これらのネットグループに属するすべてのメンバーが /home ディレクトリーに対する読み取り/書き込みアクセス権限を持つとは限りません。個別の、そしてグループごとの許可がまだこのパスに適用されています。また、このコマンドには、ANON=-1 という指定もあります。これは、無名ユーザーがこのシステムにアクセスすることを許可しないという指定です。

例 4: エクスポートで読み取り専用許可を強制する

```
CHGNFSEXP OPTIONS('-I -O R0, RW=students:LabSystems, ANON=-1,
ACCESS=Profs:LabSystems:students')
DIR('/classes')
```

このコマンドは、パス名 /classes より下のディレクトリー・ツリーをネットグループ Profs、LabSystems、および students にエクスポートします。しかし、students および LabSystems だけが、エクスポートされたツリーに対する読み取り/書き込みアクセス権限を持ちます。また、このコマンドには、ANON=-1 という指定もあります。これは、無名ユーザーがこのシステムにアクセスすることを許可しないという指定です。

iSeries ナビゲーターからのエクスポート

ローカル・ファイル・システムは、iSeries ナビゲーターを使用してエクスポートすることもできます。iSeries ナビゲーターでエクスポートするには、次のステップに従ってください。

1. 「iSeries ナビゲーター」で、ご使用のサーバーを展開します。
2. 「ファイル・システム」を展開します。

次の画面は、iSeries ナビゲーターのインターフェースの例です。

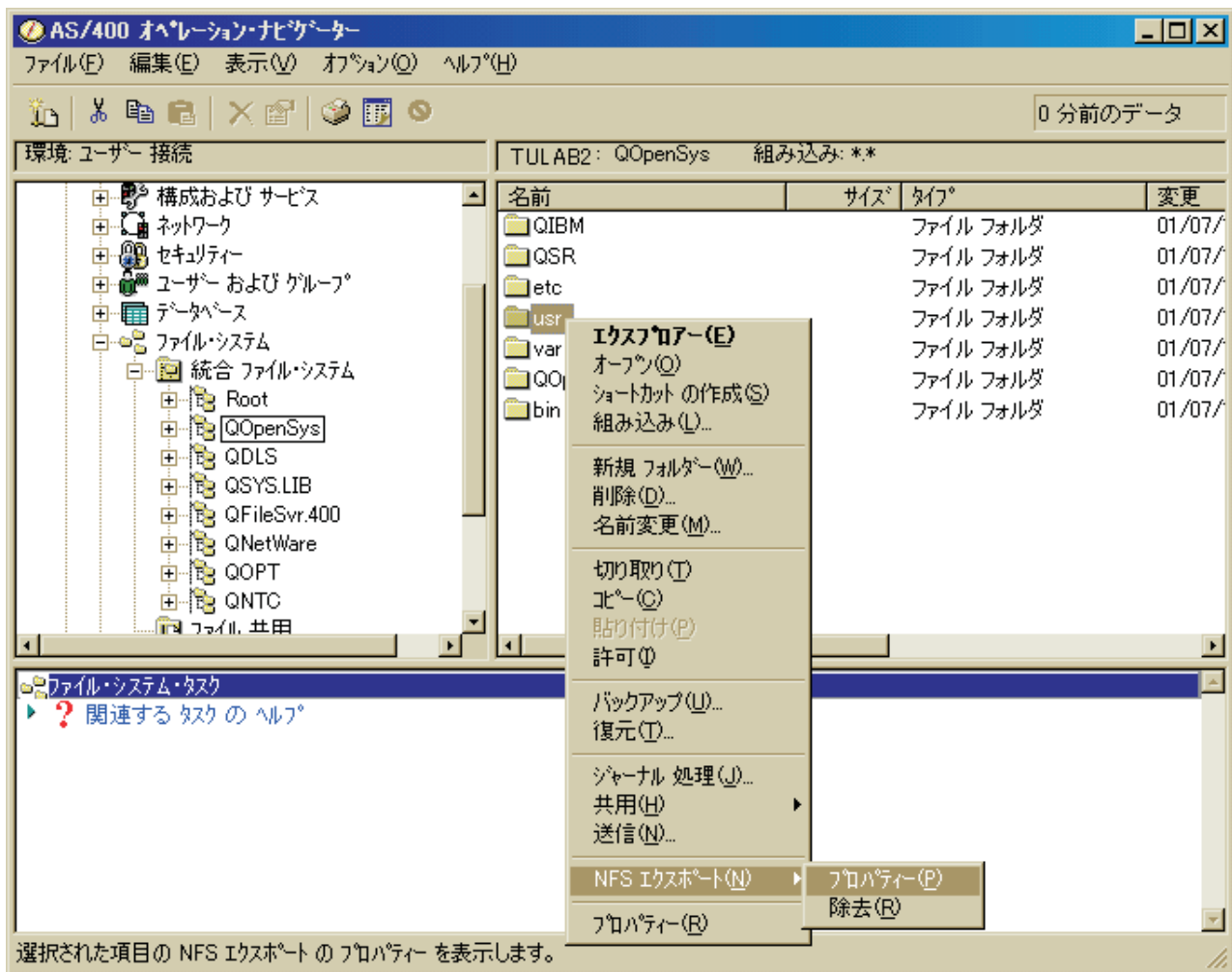


図 25. iSeries ナビゲーターのインターフェース

3. 「統合ファイル・システム」を展開します。エクスポートしたいファイル・システムまたはフォルダが見えるまで展開を続行します。
4. ファイル・システムまたはフォルダを右マウス・ボタン・クリックし、「NFS エクスポート」->「プロパティー」と選択します。次の図は、iSeries ナビゲーターで表示されるダイアログ・ボックスです。

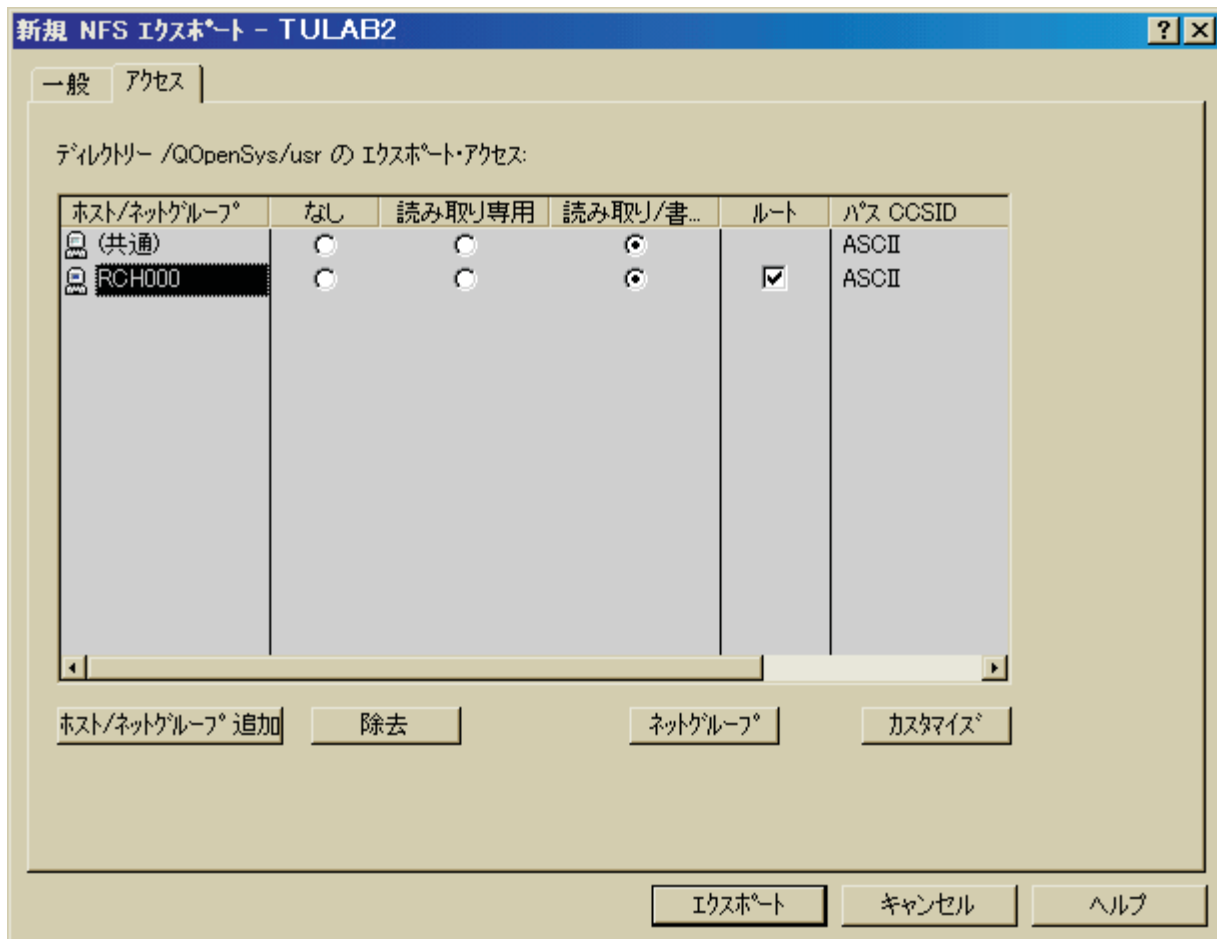


図 26. 「NFS エクスポート」ダイアログ・ボックス

5. 「一般」タブの「匿名ユーザー・マッピング」フィールドには、NFS サーバーにログオンする不明または匿名のユーザーに割り当てられるユーザー ID を入力します。
6. 「アクセス」タブをクリックして、エクスポートに適切なアクセスの設定値を選択します。
7. 「ホスト/ネットグループの追加」をクリックして、特定の特権を持つ別のクライアントを追加します。次の図は、「ホスト/ネットグループの追加」ダイアログ・ボックスを示しています。



図 27. 「ホスト/ネットワークグループの追加」

8. 「OK」をクリックして、選択したクライアントを「ホスト/ネットワークグループの追加」ダイアログ・ボックスに追加します。再度「新規 NFS エクスポート」ダイアログが表示されます。
9. 「カスタマイズ」をクリックして、パス *CCSID* およびデータ *CCSID* オプションを構成します。次の図は、「NFS クライアントのアクセスのカスタマイズ」ダイアログ・ボックスを示しています。

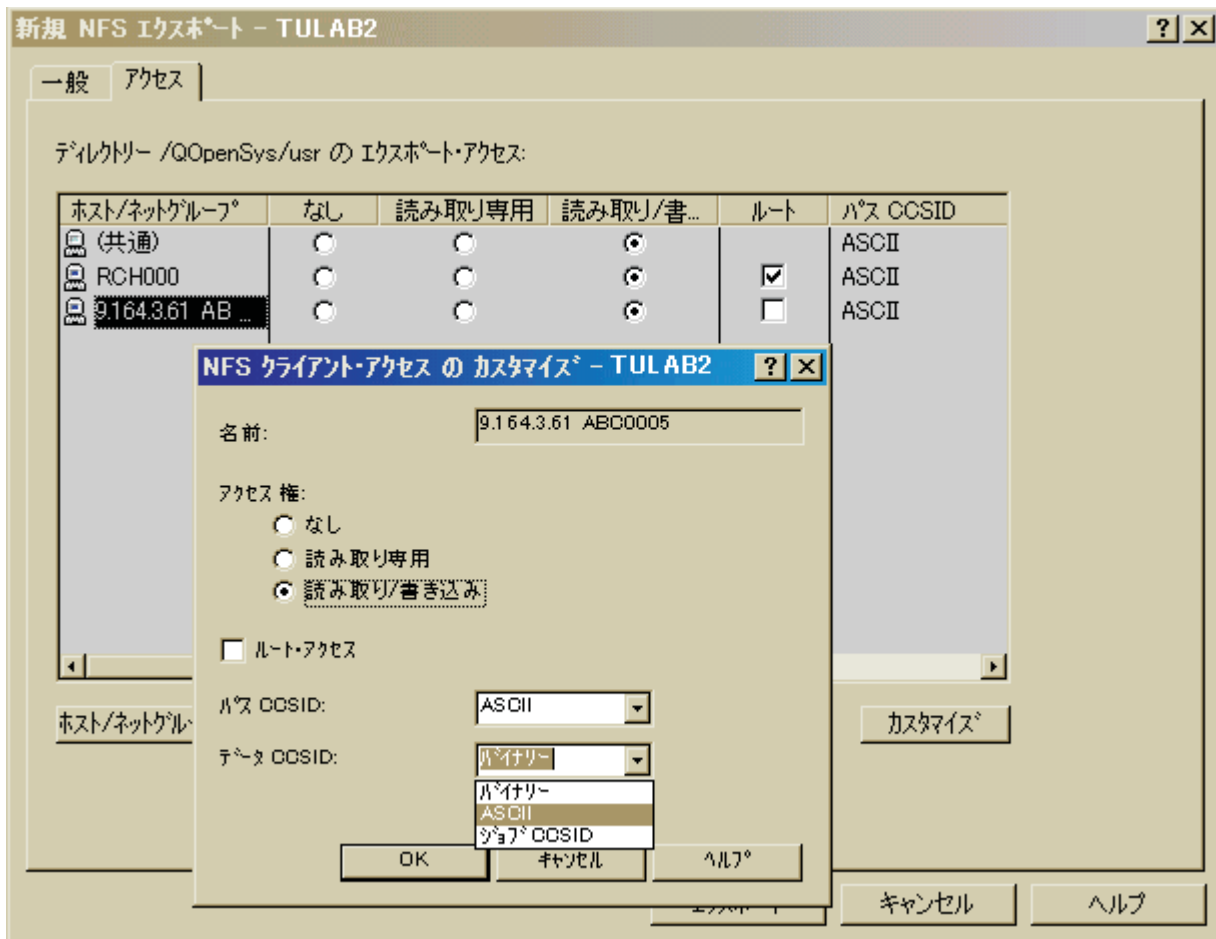


図 28. 「NFS クライアントのアクセスのカスタマイズ」ダイアログ・ボックス

10. 「OK」をクリックして、カスタマイズした設定値をエクスポート機能に追加します。再度「新規 NFS エクスポート」ダイアログが表示されます。
11. 「エクスポート」をクリックします。
即時に新しいエクスポートが行われます。新規または変更されたエクスポートで、/ETC/EXPORTS ファイルを更新するオプションも使用できます。

エクスポート対象の検索

現在、サーバーにエクスポートされている項目を知りたい場合がよくあります。それを知るためには、次の 3 通りの方法があります。

1. iSeries ナビゲーターを使用する。
2. ネットワーク・ファイル・システムのエクスポート項目の検索 (QZNFRTVE) API を使用する。
3. ネットワークで、UNIX の showmount コマンドを使用する。

iSeries ナビゲーターの使用

現在、サーバーにエクスポートされている項目は、iSeries ナビゲーターを使用して見つけることができます。検索は次のように行います。

1. 「iSeries ナビゲーター」で、ご使用のサーバーを展開します。

2. 「ネットワーク」を展開します。
3. 「サーバー」を展開します。
4. 「TCP/IP」をクリックします。右側のパネルに、NFS の状況と他のサーバーが表示されます。
5. 「NFS」を右マウス・ボタン・クリックし、「エクスポート」を選択します。ここで、新しいエクスポートを追加したり、リストから項目を除去したりできます。

次の図は、NFS をエクスポートするためのダイアログ・ボックスです。

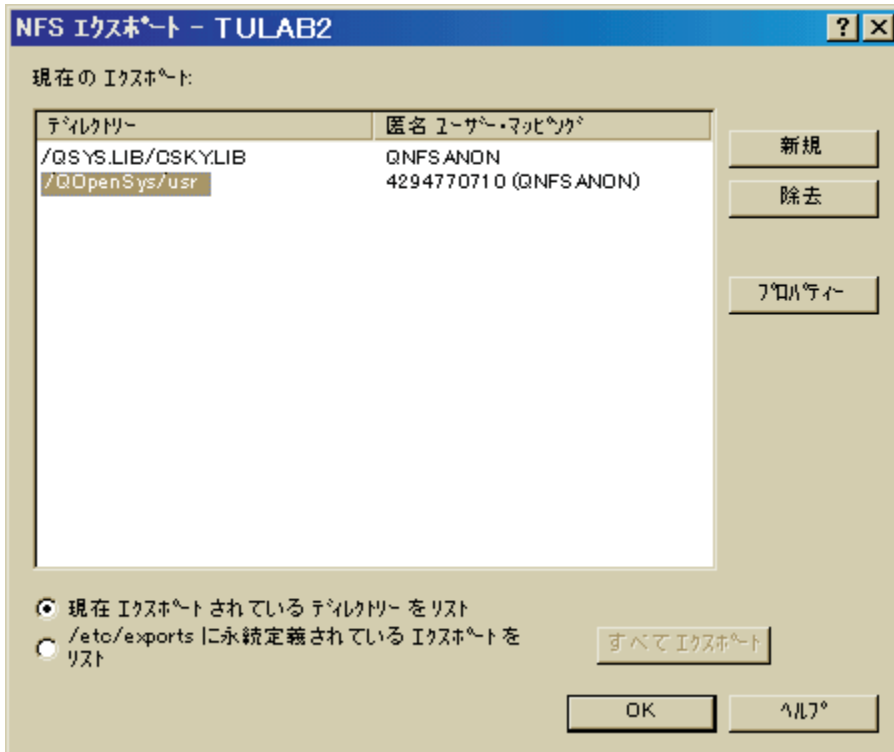


図 29. 「NFS エクスポート」ダイアログ・ボックス

6. 「OK」をクリックします。

「ネットワーク・ファイル・システムのエクスポート項目の検索 (QZNFRTVE)」API

ある iSeries に現在、エクスポートされている項目を検索する 2 番目の方法は、「ネットワーク・ファイル・システムのエクスポート項目の検索 (QZNFRTVE)」API を使用することです。このプログラミング・インターフェースを使用すると、エクスポートされた 1 つまたは複数の項目に関する情報を検索できます。この API を呼び出すには、コマンド行から次のように入力します。

```
CALL QZNFRTVE
```

詳細メッセージを表示している場合は、この入力によって、すべての現行エクスポートがリストされます。個々のエクスポートについて PF1 を押すと、その項目の構成の詳細が表示されます。QZNFRTVE API について詳しくは、iSeries Information Center のトピック「API」を参照してください。

UNIX showmount コマンド

3 番目の方法では、UNIX システムからネットワークにエクスポートされた項目が表示されます。マウント・デーモンが、エクスポート項目を戻すサービスを行います。TULAB1 で発行する以下のコマンドは、このサービスを使用して、TULAB2 にエクスポートされた項目を使用する方法を示しています。

```
> showmount -e tulab2
export list for tulab2:
/classes/class1 host1,host2
/engdata/civil (everyone)
```

エクスポートに関する考慮事項

マウントされたファイル・システムのループ

ユーザーおよび管理者は、すでにマウントされたネットワーク・ファイル・システムはエクスポートできないという NFS の制限に困難を感じる場合があります。NFS では、マウントされたファイル・システムのエクスポートは許されません。これは、マウントされたファイル・システムがループに陥るおそれがあるからです。この問題は、NFS がマウントを許可したため、新しいクライアントがその名前空間をマウントした場合に発生することになります。新しいクライアントは、マウント・ポイントを検出するためには、マウント・ポイントを通り抜ける必要があることに気付きます。決してディレクトリーの末尾に到達することがないので、パスが永久に循環してしまいます。マウント・デーモンは終わりのないストリングに陥り、その結果、クライアントに障害が発生します。

マウントされたファイル・システムのループの解決

NFS は、すでにエクスポートされた NFS ファイル・システムのエクスポートを許可しないので、この問題が起きることはありません。

記号リンク

ネットワーク・ファイル・システムは、記号リンクの使用をサポートします。記号リンクは、ファイル内に含まれているパスの形式でのパス名の表現です。実際のパスは、ファイルの内容に基づいてパス検索を行うことで決定されます。

2 つのパス名のうちの 1 つが記号リンクであるために、その 2 つが同一のオブジェクトを指すようになることがあります。たとえば、ディレクトリー /nancy に、/home/grad/kathryn への記号リンクが入っている場合、/home/grad/kathryn にあるオブジェクトへのパスは、同時に /nancy でもあります。2 つのパス名は同一です。ユーザーがこの 2 つの項目をエクスポートした場合、どちらも正しくエクスポートされます。/home/grad/kathryn のエクスポートは /nancy のエクスポートと同じことです。NFS は、オブジェクト をエクスポートするのであり、パス名 はエクスポートしないので、エクスポート・テーブルに保管する必要があるのは、最後に起きたオブジェクトのエクスポートだけです。

記号リンクについて詳しくは、Iseries Information Center のトピック「ファイル・システムおよび管理」の「記号リンク」を参照してください。

第 5 章 クライアントでファイル・システムをマウントする

マウント・コマンドは、リモート・ファイル・システムを、NFS クライアントのローカル・ディレクトリー上に配置します。エクスポートの後にファイル・システムをマウントすることは、クライアントとサーバーとの間の「透過的な」関係をセッアップするための 2 番目の主要なステップです。

マウントにより、サーバーがエクスポートしたさまざまなファイル・システムを実際に利用できるようになります。クライアントは、マウント・コマンドを使用して、エクスポートされたファイル・システムをローカル・ファイル・システムの全体または一部の上にマップすることができます。この処理はシームレスに行われるため、ローカル・アプリケーションは、リモート・サーバーからマウントされたファイル・システムとローカルに存在するファイル・システムとをほとんど区別できません。複数のクライアントが、同時に 1 つまたは複数のファイル・システムをマウントして操作の対象にできます。

リモート・サーバーからファイル・システムがエクスポートされると、クライアントは、それらのアクセス可能なファイル・システムをマウントして、自分のローカル名前空間の一部にすることができます。クライアントは、エクスポートされたサーバー・ファイル・システムの全部または一部を動的にマウントしたりマウント解除したりできます。クライアントがファイル・システムを自分のローカル名前空間にマウントした後、マウント・ポイントより下にあるローカル・ファイル・システムの情報は「覆い隠されて」しまいます。つまり、「覆われた」または「隠された」ファイル・システムは、リモート・ファイル・システムがマウント解除されるまではアクセス不能です。

マウントとは何か

マウントは、ローカル・クライアントからリモート・サーバーのファイル・システムへのアクセスを提供するための、クライアント側での操作です。マウント・コマンドは、ファイル・システムをクライアント上にコピーするのではなく、リモート・ファイル・システムがあたかも物理的にクライアント上に存在しているかのように見えるようにします。実際には、そのファイル・システムはサーバー上にだけ存在していて、クライアントはそれをアクセスしているに過ぎません。しかし、インターフェースは、マウントされたファイル・システムがローカルに存在するという印象を与えるように設計されています。ほとんどの場合に、アプリケーションもユーザーもその違いを見分けることはできません。

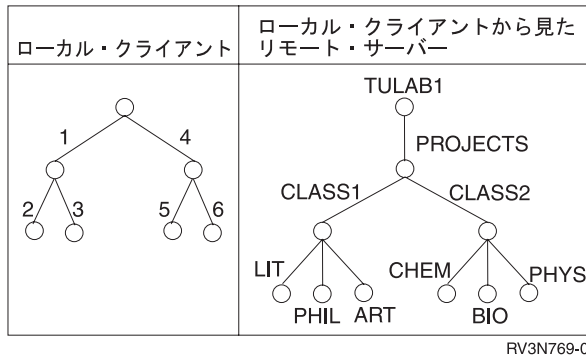


図 30. ローカル・クライアントとファイル・システムをエクスポートしたリモート・サーバー

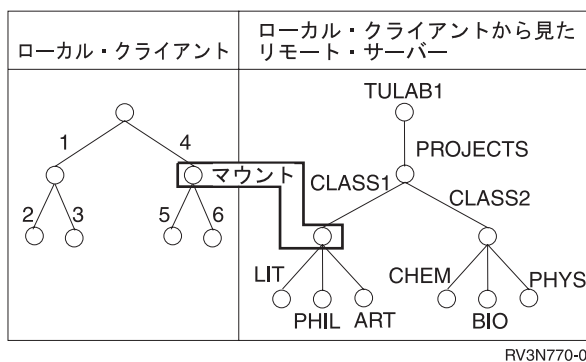


図 31. ローカル・クライアントがリモート・サーバーからファイル・システムをマウントする

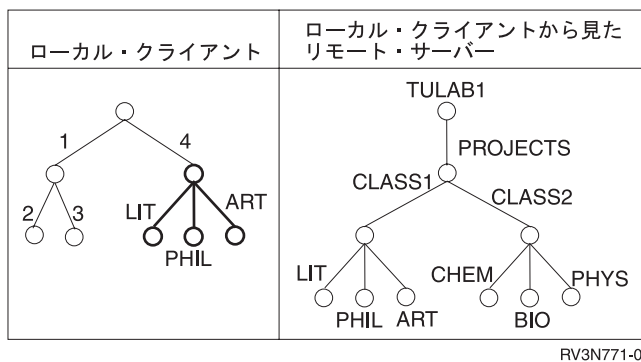


図 32. マウントされたファイル・システムは、ローカル・クライアントのディレクトリーを覆い隠す

適切な権限を持っている限り、NFS クライアントは、NFS サーバーからエクスポートされた任意のファイル・システム、またはファイル・システムの一部をマウントできます。マウントは、リモート・サーバーからエクスポートされたディレクトリーを選択し、ローカル・クライアントの統合ファイル・システムの名前空間からそのディレクトリーをアクセスできるようにするという、クライアント側での処理です。

多くの UNIX 版の NFS では、リモート・ファイル・システムをクライアント上の /etc/fstab ファイルに指定することにより、IPL 時に自動的にマウントさせること

ができます。しかし、iSeries 版の NFS では、IPL 時に作動するプログラムが用意されていて、通常は、ユーザーが選択できる各種のシステム・タスクを開始するために使用されます。IPL 時に始動するこのプログラムの名前は、QSTRUPPGM というシステム値に入っています。プログラムの省略時名は QSTRUP で、QSYS ライブラリーに入っています。ユーザーは、このプログラムを編集して「ADDMFS (マウント済みファイル・システムの追加)」または MOUNT コマンドを組み込むことにより、始動時に自動的にリモート・ファイル・システムをマウントするようになります。マウント・コマンドを使用する前に STRNFSSVR *ALL コマンドを使用することをお奨めします。さらに、エクスポートされたリモート・ファイル・システムは、IPL 後の任意の時点で ADDMFS または MOUNT コマンドを使用する、適切な権限を持つクライアントによって明示的にマウントすることができます。

ファイル・システムをクライアントにマウントすると、マウント・ポイントより下に存在するファイル・システム、ディレクトリー、またはオブジェクトはすべて、マウントされたファイル・システムによって「覆い隠されて」しまいます。つまり、ファイル・システムをマウントしたときに、マウント・ポイントより下流にファイル・システム、ディレクトリー、またはオブジェクトがローカルに存在すると、それらはアクセス不能になります。

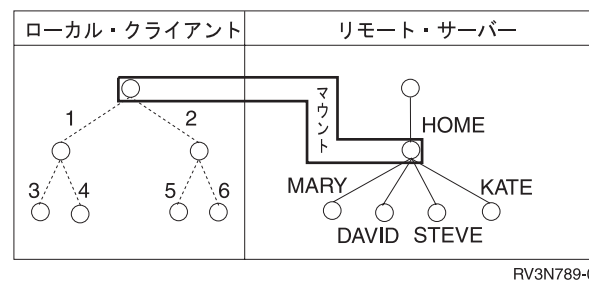


図 33. 高いレベルのディレクトリー上にマウントしたローカル・クライアント

エクスポートに関する「下流」の規則に類似した、マウントに関する「下流」の法則があります。リモート・ファイル・システムをローカル・ディレクトリーにマウントすると、マウント・ポイントの「下流」にあるすべてのオブジェクトは「覆い隠されます」。したがって、それらのオブジェクトはローカル名前空間からはアクセス不能になります。ローカル・ディレクトリー・ツリーの高いレベルにマウントすると、自分のローカル・オブジェクトの大部分がアクセス不能になります。

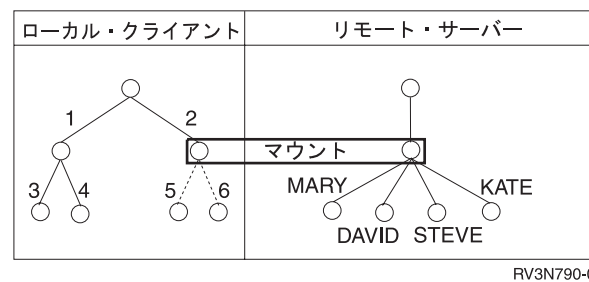


図 34. /2 ディレクトリー上にマウントしたローカル・クライアント

リモート・ファイル・システムをローカル・ディレクトリー・ツリーの低い地点にマウントすれば、自分のローカル・オブジェクトの中に「覆い隠される」ものがあるとしても、それほど多くならずに済みます。

マウントに伴うこの「覆い隠し」により、新しくマウントしたファイル・システムより下にあるローカル・データは、ユーザーから「見えなく」なります。つまり、統合ファイル・システムの名前空間に登場しなくなります。「覆い隠された」ファイル・システムは、マウントされたファイル・システムがマウント解除されるまで、アクセス不能です。クライアントは、以前にマウントした任意のファイル・システムを動的にマウント解除することができます。この処理は、マウントによって隠されていたファイル・システム、ディレクトリー、またはオブジェクトから「覆いを取り去る」こととなります。また、マウントされていたその特定のファイル・システムについて、サーバーとの接続を（そしてもちろん、アクセスも）切断することとなります。

なぜファイル・システムをマウントするのか

ファイル・システムをマウントすると、以前にはサーバー上でしか利用できなかったファイルやその他のオブジェクトを、クライアント・システムで処理できるようになります。これは、必要な情報または重要な情報を幅広い範囲のユーザーおよびシステムに分散させるために役立ちます。さらに、マウント機能は、ネットワークの始動時に必要な反復を減らすことができます。各クライアントは始動時の構成をサーバーから直接マウントすることができ、必要ならサーバーの再構成を自由に行うこともできます。

マウント機能があるため、リモート・クライアントでのディレクトリー構造を決定するのが容易になり、自由度も増します。必要に応じてファイル・システムを動的にマウントしたりマウント解除したりできます。さらに、ユーザーはマウント・コマンドにパラメーターやオプションを指定することにより、クライアントおよびサーバーに最大限の資源と最大限の機密保護を提供できます。

ときには、クライアントの名前空間にある情報が複雑になりすぎたり多くなりすぎたりして圧倒されることがあります。マウント解除コマンドは、ファイル・システムを一度に 1 つずつ、ゆっくりと切り離していくための簡単な方法です。すべてのファイル・システムをマウント解除するには、「UNMOUNT」または「RMVMS (マウント済みファイル・システムの除去)」コマンドの TYPE パラメーターに *ALL という値を指定します。

ファイル・システムをマウントおよびマウント解除する方法の詳細については、54 ページの『ADDMFS (マウント済みファイル・システムの追加) コマンド』 および 57 ページの『RMVMS (マウント済みファイル・システムの除去) コマンド』 を参照してください。

どのようなファイル・システムをマウントできるか

ユーザーは、以下の 3 種類のファイル・システムを iSeries にマウントすることができます。

ネットワーク・ファイル・システム

ユーザーは大部分のファイル・システムを始動時にマウントするとはいえ、

ファイル・システムを動的にマウントしたりマウント解除したりする必要が生じることもあります。サーバーからエクスポートされたリモート・ファイル・システムは、ローカル・クライアントが適切なアクセス権限を持っている限り、いつでもマウントすることができます。

ユーザー定義ファイル・システム

iSeries では、UDFS は、統合ファイル・システムの名前空間内の隠されたオブジェクトとして見えます。UDFS の内容は、統合ファイル・システムの名前空間内にマウントされて初めてアクセスできるようになります。

UDFS は、始動時にマウントすることができますが、任意の時点でユーザーが動的にマウントしたりマウント解除したりすることもできます。

Novell** NetWare** ファイル・システム

ユーザーは、NetWare ファイル・システムも、動的にマウントしたりマウント解除したりすることができます。NetWare ファイル・システムの詳細については、以下の資料を参照してください。

- iSeries Information Center 内の「iSeries 上の NetWare」
- iSeries Information Center 内の「NetWare file system」

ファイル・システムをどこにマウントできるか

NFS ファイル・システムは、別のクライアント・ファイル・システムの全部または一部の上にマウントすることができます。これが可能なのは、マウント・ポイントとして使用されるディレクトリーは、それが実際にどこに存在していても同じに見えるからです。

クライアントにとっては、NFS ファイル・システムは「通常の」ローカル・ファイル・システムのように見え、そのように機能します。ユーザーは、ネットワーク・ファイル・システムを、以下の iSeries クライアント・ファイル・システムの上にマウントできます。

- 「ルート」(しかし、ルート・ディレクトリーそのものの上ではない)
- QOpenSys
- NFS
- UDFS

エクスポートされたファイル・システムをクライアントがマウントすると、新しくマウントされたファイル・システムの下にあるものはすべて覆い隠されます。このことは、リモート・ファイル・システムをローカル・ディレクトリー上にマウントする場合にも、以前にマウントしたりリモート・ファイル・システム上にマウントする場合にも当てはまります。そのようにして覆い隠されたファイル・システムは、「上を覆っている」ファイル・システムをすべてマウント解除するまでは、アクセス不能です。

たとえば、TULAB2 が /classes/class1 をエクスポートし、それには /classes/class1/proj1 ディレクトリーが含まれているとします。一方、リモート・クライアントにはローカル・ディレクトリー /user があり、それには /user/work ディレクトリーが含まれ、さらにそれには /user/work/time ディレクトリーが含まれているとします。クライアントが /classes/class1/ を /user/work 上にマウントすると、ローカル・ディレクトリー・ツリー上でマウント・ポイントより「下流」にあるものはすべて、マウントされたファイル・システムによって完

全に覆い隠されます。この例の場合、マウント・ポイントは、/user/work です。したがって、/user/work ディレクトリーには proj1 だけが含まれることとなります。クライアントが、「覆い隠された」データ (/user/work/time) をアクセスしようとすると、ユーザーにエラー・メッセージが戻されます。マウントされたファイル・システムによって覆い隠されたデータは、そのファイル・システムがマウント解除されるまではアクセス不能です。

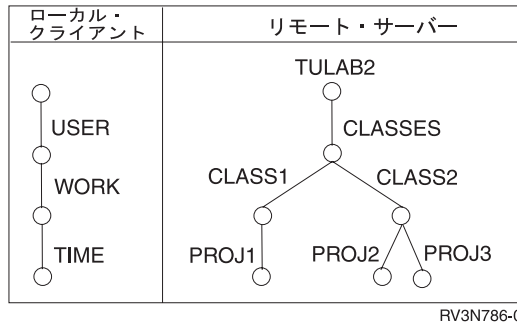


図 35. ローカル・クライアントおよびリモート・サーバーのビュー

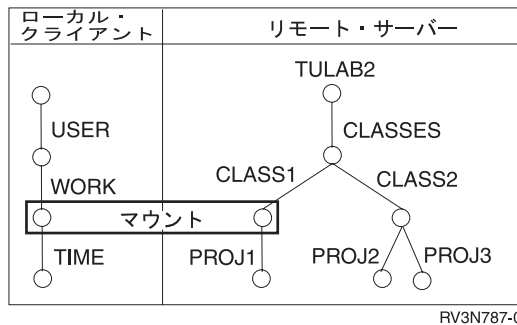


図 36. クライアントが TULAB2 から /classes/class1 をマウントする

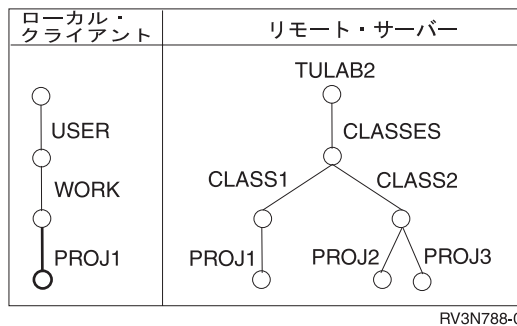


図 37. /classes/class1 ディレクトリーが /user/work を覆う

引き続きこの例で考えます。もう 1 つのリモート・ファイル・システム /engdata が、TULAB2 からエクスポートされたとします。エクスポートに関する「下流」規則に従って、/engdata のすべてのサブディレクトリーも一緒にエクスポートされます。クライアントは、エクスポートされたサブディレクトリーを、すでに存在するマウント・ポイント上にマウントすることができます。/engdata を /user/work デ

ディレクトリー上にマウントすると、/user/work のすべての内容が、/user/work/proj1 を含めて、マウントにより覆い隠されます。したがって、それらはアクセス不能になります。クライアント上の新しいローカル・ディレクトリー・ツリーは、以下の図に示すとおり、/user/work と、そのさまざまな内容およびサブディレクトリーということになります。

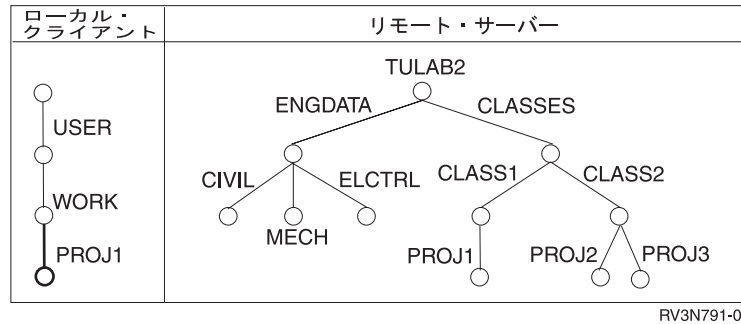


図 38. リモート・サーバーが /engdata をエクスポートする

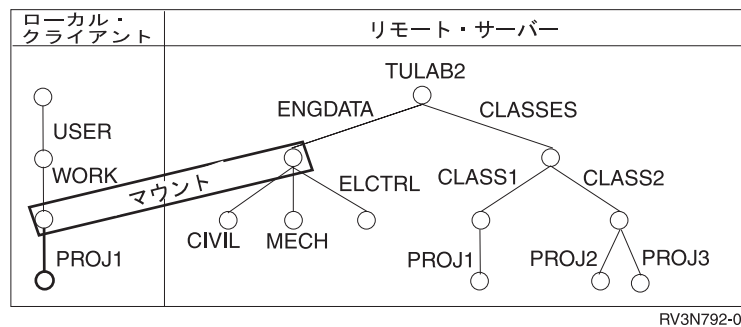


図 39. ローカル・クライアントが /engdata をマウント・ポイント上にマウントする

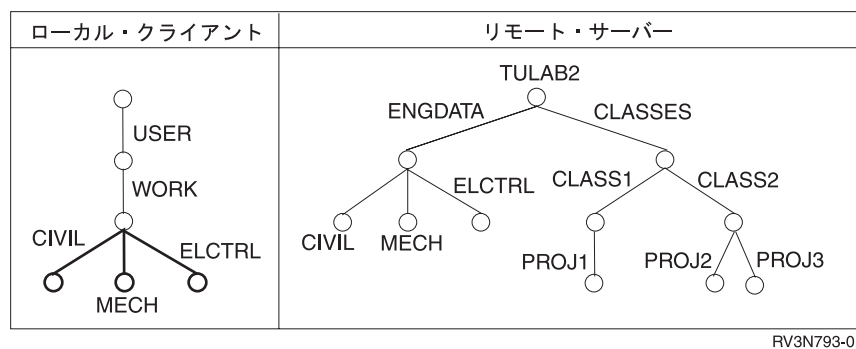


図 40. /engdata ディレクトリーが /user/work を覆う

注: NFS クライアントは、常に ファイル・システムの最新のビューを見ます。クライアントがファイル・システムを動的にマウントまたはマウント解除した場合、次の最新表示の後にその変更がクライアントの名前空間に反映されます。

いくつかの異なるリモート・ファイル・システムを、同じマウント・ポイント上にマウントすることも可能です。ユーザーは、各種のリモート・サーバーからのファ

イル・システムを、前もってマウント解除せずに、ローカル・クライアントの同じディレクトリー上にマウントすることができます。いくつか「積み重ねられた」ファイル・システムをマウント解除すると、最後にマウントされたものが最初に除去されます。

マウント・ポイント

マウント・ポイントは、ローカル・クライアントおよびリモート・サーバーの名前空間のうち、エクスポートされたファイル・システムをユーザーがマウントした領域にマークを付けます。マウント・ポイントは、ファイル・システムのマウント元であるサーバー上の地点と、マウント先 であるクライアント上の地点を示します。

たとえば、システムが TULAB1 から /home/consults ディレクトリーをエクスポートし、それをリモート・クライアントの /test ディレクトリー上にマウントするとします。この場合、クライアントのマウント・ポイントは、/test です。元のディレクトリー /test およびその内容は覆い隠され、サーバー TULAB1 の名前空間内のウィンドウとなります。リモート・ファイル・システムとマウント・ポイントまたはパスとの対応付けを調べる場合は、DSPMFSINF コマンドを使用します。例：

```
DSPMFSINF OBJ('/test')
```

このコマンドについて詳しくは、59 ページの『DSPMFSINF (マウント済みファイル・システム情報の表示) コマンド』を参照してください。

ファイル・システムをマウントする方法

リモート・サーバーのファイル・システムをローカル名前空間からアクセスできるようにするには、MOUNT および「ADDMFS (マウント済みファイル・システムの追加) CL コマンドを使用します。UNMOUNT および「RMVMFS (マウント済みファイル・システムの除去) コマンドは、マウントされたファイル・システムを名前空間から除去します。「DSPMFSINF (マウント済みファイル・システム情報の表示) コマンドは、マウントされたファイル・システムについての情報を提供します。これらのコマンドは、メニューから使用することもできます。任意のコマンド行から「GO CMDMFS (マウント済みファイル・システムのコマンド・メニューへ進む)」を入力します。

NFS ファイル・システムをマウントする前に、ご使用のサーバーに正しい TCP/IP 構成が存在することを確認する必要があります。iSeries Information Center のトピック「TCP/IP の構成」を参照してください。

ADDMFS (マウント済みファイル・システムの追加) コマンド

目的

「マウント済みファイル・システムの追加 (ADDMFS)」コマンドは、ファイル・システム内のオブジェクトを、統合ファイル・システムの名前空間からアクセスできるようにします。マウントできるファイル・システムは、以下のいずれかです。

1. ローカル・サーバー上のユーザー定義ファイル・システム (*UDFS)
2. ローカルのネットワーク・ファイル・システム・クライアントを経由してアクセスされるリモート・ファイル・システム (*NFS)
3. ローカルまたはリモートの NetWare ファイル・システム (*NETWARE)

- ADDMFS コマンドの MFS パラメーター (必須) では、マウントするファイル・システムのパス名を指定します。ローカルのブロック特殊ファイル (*BLKSF) へのパス、リモートの NFS パス名、または NetWare ファイル・システムのパスを指定できます。
- ADDMFS コマンドの MNTOVRDIR パラメーター (必須) では、ファイル・システムのマウント先になる既存のディレクトリー名を指定します。このディレクトリーは、マウント済みファイル・システムによって「覆われる」こととなります。
- マウント・オプション・リストには、マウント・オプションの文字ストリングを指定します。各キーワードはコンマで区切ります。キーワードによっては、キーワードの後に等号 '=' と値を指定します。特に指定しないキーワードについては、そのオプションの省略時値が使用されます。オプション・リストには、スペースを含めることができます。
- ADDMFS コマンドのコード化文字セット ID (CCSID) パラメーターでは、NFS 用の 1 対の CCSID を指定します。この CCSID 値は完全な単一バイトまたは 2 バイト CCSID のためのものです。
- データ・ファイルの CCSID は、リモート・サーバーにあるデータ・ファイルについて、どの CCSID を想定するかを指定します。1 文字当たりのバイト数が元のデータと等しい CCSID を指定してください。
- パス名の CCSID は、リモート・サーバーのパス名について、どの CCSID を想定するかを指定します。これは、リモート・サーバーのパス名に対して想定される CCSID です。

MOUNT コマンドの例

例 1: ユーザー定義ファイル・システムをマウントする

```
ADDMFS TYPE(*UDFS) MFS('/DEV/QASP02/PROJ.UDFS')
      MNTOVRDIR('/REPORT')
```

このコマンドは、ユーザー定義ファイル・システム PROJ.UDFS をディレクトリー /report 上にマウントします。このコマンドでは、その他のパラメーターに省略時値を使用しています。

例 2: TULAB2 からネットワーク・ファイル・システムをマウントする

```
ADDMFS TYPE(*NFS) MFS('TULAB2:/QSYS.LIB/SCHOOL.LIB')
      MNTOVRDIR('/HOME')
```

このコマンドは、リモート・システム TULAB2 のリモート・ファイル・システム /qsys.lib/school.lib を、ローカル・クライアントのディレクトリー /home 上にマウントします。

例 3: オプションを指定してネットワーク・ファイル・システムをマウントする

```
ADDMFS TYPE(*NFS) MFS('TULAB2:/QSYS.LIB/WORK.LIB')
      MNTOVRDIR('/HOME')
      OPTIONS('ro, nosuid, rsize=256, retrans=10')
      CCSID(*JOBCCSID)
```

このコマンドは、リモート・システム TULAB2 のファイル・システム /qsys.lib/work.lib を、ローカル・クライアントのディレクトリー /HOME 上にマウントします。さらに、以下の事項も指定します。

- 読み取り専用としてマウントする。

- setuid の実行を許可しない。
- 読み取りバッファを 256 バイトに設定する。
- 再送信回数を 10 回に設定する。

リモート・システムにあるデータのコード・ページを決定するには、省略時のジョブの CCSID が使用されます。

例 4: オプションを指定して NetWare ファイル・システムをマウントする

```
ADDMFS TYPE(*NETWARE)
      MFS('TULAB2/NET:WORK/PROJONE')
      MNTOVRDIR('/temp1')
      OPTIONS('ro,agregmax=120')
```

このコマンドは、サーバー TULAB2 に存在するボリューム NET 内の NetWare ディレクトリー WORK/PROJONE を、ディレクトリー /temp1 上にマウントします。さらに、読み取り専用としてマウントし、ファイル属性をローカルに保管する最大時間を 120 秒に設定することを指定します。

例 5: NetWare Directory Services** のコンテキストを使用してマウントする

ここでは、NetWare Directory Services (NDS**) のコンテキストを使用して NetWare ファイル・システムをマウントするいくつかの例について説明します。

```
ADDMFS TYPE(*NETWARE) MFS('.COMP.TULAB.UNIVER')
      MNTOVRDIR('/temp1')
```

このコマンドは、NDS ボリューム COMP を、区別されたコンテキストを使用して、ディレクトリー /temp1 上にマウントします。

```
ADDMFS TYPE(*NETWARE)
      MFS('CN=NET_VOL.OU=TULAB2:WORK/PROJONE')
      MNTOVRDIR('/temp1')
```

このコマンドは、NDS ボリューム NET にあるパス WORK/PROJONE を、相対パスと完全修飾名を使用して、ディレクトリー /temp1 上にマウントします。

```
ADDMFS TYPE(*NETWARE)
      MFS('.CN=NETMAP.OU=COMP.O=TULAB')
      MNTOVRDIR('/temp1')
```

このコマンドは、区別されたコンテキストと完全修飾名を使用して、ディレクトリー・マップ・オブジェクトをディレクトリー /temp1 にマウントします。

RMVMFS (マウント済みファイル・システムの除去) コマンド

目的

「マウント済みファイル・システムの除去 (RMVMFS)」コマンドは、以前にマウントしたファイル・システムを、統合ファイル・システムの名前空間内でアクセスできないようにします。アクセス不能にできるファイル・システムは、以下のいずれかです。

1. ローカル・サーバー上のユーザー定義ファイル・システム (*UDFS)
2. ネットワーク・ファイル・システム・サーバーを経由してアクセスされるリモート・ファイル・システム (*NFS)
3. ローカルまたはリモートの NetWare ファイル・システム (*NETWARE)

ファイル・システム内のいずれかのファイルが使用中である場合は、ユーザーにエラー・メッセージが戻されます。ファイル・システムそのもののいずれかの部分がマウントされて覆い隠されている場合は、その覆い隠しを解除してからでないと、そのファイル・システムをマウント解除することはできません。同じマウント・ポイント上に複数のファイル・システムがマウントされている場合は、最後にマウントされたファイル・システムが最初に除去されます。

このコマンドは、次の代替コマンド名を使用して発行することもできます。

- UNMOUNT

制約事項

1. このコマンドを使用するには、*IOSYSCFG 特殊権限が必要です。

RMVMFS および UNMOUNT コマンドとその関連パラメーターおよびオプションについて詳しくは、iSeries Information Center のトピック「CL」を参照してください。

RMVMFS/UNMOUNT 画面

マウント済み FS の除去 (RMVMFS)

選択項目を入力して、実行キーを押してください。

ファイル・システムのタイプ > *NFS *NFS, *UDFS, *NETWARE, *ALL
マウント済みディレクトリー > '/USER/WORK'
マウント済みファイル・システム > '/CLASSES/CLASS1'

終り

F3= 終了 F4= プロンプト F5= 最新表示 F12= 取り消し
F13= この画面の使用法 F24= キーの続き

図 42. 「マウント済み FS の除去 (RMVMFS)」画面

RMVMFS または UNMOUNT コマンドを使用するときには、次のような多くのパラメーターおよびオプションを指定することができます。

- RMVMFS コマンドの TYPE パラメーターでは、マウント解除するファイル・システムのタイプを指定します。
- RMVMFS コマンドの MNTOVRDIR パラメーターでは、以前の ADDMFS コマンドのマウント先になった（「覆い隠された」）ディレクトリーのパス名を指定します。
- RMVMFS コマンドの MFS パラメーターでは、マウント解除するファイル・システムを指定します。

注: このパラメーターは、RMVMFS コマンドの TYPE パラメーターに *UDFS 値を指定した場合に、ブロック特殊ファイル (*BLKSF オブジェクト) をマウント解除するときだけに使用します。

例

例 1: ディレクトリーをマウント解除する

```
RMVMFS TYPE (*NFS) MNTOVRDIR('/tools')
```

このコマンドは、ディレクトリー */tools* 上でアクセスできるネットワーク・ファイル・システムをマウント解除します。

例 2: ユーザー定義ファイル・システムをマウント解除する

```
RMVMFS TYPE(*UDFS) MFS('/DEV/QASP02/A.udfs')
```

このコマンドは、ユーザー定義ファイル・システム */DEV/QASP02/A.udfs* をマウント解除します。

例 3: クライアントにマウントされているすべてのファイル・システムをマウント解除する

```
RMVMFS TYPE(*ALL) MNTOVRDIR(*ALL)
```

このコマンドは、クライアントにマウントしたすべてのファイル・システムをマウント解除します。

例 4: 特定のクライアント・ディレクトリー上にマウントされたすべてのファイル・システムをマウント解除する

```
RMVMFS TYPE(*ALL) MNTOVRDIR('/napa')
```

このコマンドは、クライアントが */napa* 上にマウントしたすべてのファイル・システムをマウント解除します。

DSPMFSINF (マウント済みファイル・システム情報の表示) コマンド

目的

「マウント済みファイル・システム情報の表示 (DSPMFSINF)」コマンドは、マウントされたファイル・システムについての情報を提供します。

このコマンドは、次の代替コマンド名を使用して発行することもできます。

- STATFS

DSPMFSINF コマンドとその関連パラメーターおよびオプションについては詳しくは、iSeries Information Center のトピック「CL」を参照してください。

DSPMFSINF/STATFS 画面

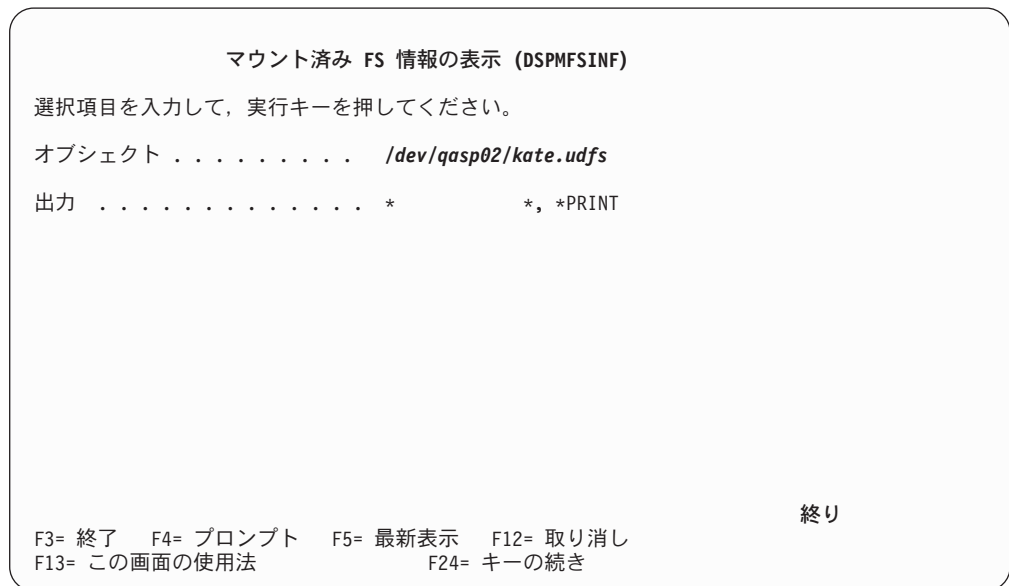


図43. 「マウント済み FS 情報の表示 (DSPMFSINF)」画面

DSPMFSINF コマンドを使用するときに指定する必要があるパラメーターは 1 つだけです。

- DSPMFSINF コマンドの OBJ パラメーター (必須) では、マウントされたファイル・システム内にあるオブジェクトで、その統計情報を表示したいもののパス名を指定します。マウントされたファイル・システム内にある任意のオブジェクトを指定できます。たとえば、ディレクトリー (*DIR) やストリーム・ファイル (*STMF) を指定することができます。

DSPMFSINF または STATFS コマンドを使用すると、次の 2 つのような一連の画面が表示されます。


```

マウント済み FS 情報の表示

オブジェクト . . . . . : /DEV/QASP02/KATE.UDFS
ファイル・システムのタイプ . . . : /dev/QASP02

ブロック・サイズ . . . . . : 4096
合計ブロック数 . . . . . : 3488768
空きブロック数 . . . . . : 1645646
オブジェクト・リンク最大 . . . : 32767
ディレクトリー・リンク最大 . . . : 32767
パス名構成要素最大 . . . . . : 640
パス名最大 . . . . . : 最大なし
所有者変更制約付き . . . . . : YES
切り捨てなし . . . . . : YES
大文字と小文字を区別 . . . . . : YES

                                                                    続く ...

続行するには、実行キーを押してください。

F3= 終了   F12= 取り消し

```

図 44. 「マウント済み FS 情報の表示 (DSPMFSINF)」出力画面 (1/2)

この最初の画面には、マウントされたファイル・システムについての基本的な情報が表示されます。

```

マウント済み FS 情報の表示

取り付けられたファイル・システム
のパス . . . . . : /DEV/QASP02/KATE.UDFS

上方取り付けパス . . . . . : /HOME/STUDENTS/ANN

保護 . . . . . : 読み取り書き込み
SETUID の実行 . . . . . : サポートされていない
取り付けタイプ . . . . . : サポートされていない
読み取りバッファ・サイズ . . . : サポートされていない
書き込みバッファ・サイズ . . . : サポートされていない
タイムアウト . . . . . : サポートされていない
再試行回数 . . . . . : サポートされていない
再送信回数 . . . . . : サポートされていない
通常のファイル属性最小時間 . . : サポートされていない
通常のファイル属性最大時間 . . : サポートされていない
ディレクトリー属性最小時間 . . : サポートされていない
ディレクトリー属性最大時間 . . : サポートされていない

                                                                    続く ...

続行するには、実行キーを押してください。

F3= 終了   F12= 取り消し

```

図 45. 「マウント済み FS 情報の表示 (DSPMFSINF)」出力画面 (2/2)

この画面に表示される拡張情報は、ユーザー定義ファイル・システムの場合にはサポートされません。

例

例 1: マウントされたファイル・システムの統計を表示する

```
DSPMFSINF OBJ('/home/students/ann')
```

このコマンドは、/home/students/ann が入っているマウント済みファイル・システムの統計情報を表示します。

例 2: '/QSYS.LIB' ファイル・システム統計の表示

```
DSPMFSINF OBJ('/QSYS.LIB/MYLIB.LIB/MYFILE.FILE')
```

このコマンドは、ライブラリー MYLIB に *FILE オブジェクト MYFILE が入っている /QSYS.LIB ファイル・システムの統計情報を表示します。

第 6 章 ネットワーク・ファイル・システムを iSeries のファイル・システムと一緒に使用する

各種のクライアントで iSeries のファイル・システムと一緒に NFS を使用する際には、いくつかの例外事項があります。これは、iSeries NFS サーバーには複数の異なるファイル・システムをエクスポートできるからです。各ファイル・システムごとに独自の要件や NFS を使用する場合と通常の機能との違いがあります。この章の目的は、NFS を使用してアクセスする特定のファイル・システムについて、このような違いを認識することです。

OS/400 バージョン 4 リリース 4 (V4R4) では、以下のファイル・システムには 2 ギガバイトを超えるストリーム・ファイルをサポートするための拡張機能が与えられていました。

- ライブラリー・ファイル・システム (QSYS.LIB)
- オープン・システム・ファイル・システム (QOpenSys)
- 「ルート」 (I)
- ユーザー定義・ファイル・システム (UDFS)

iSeries 上の NFS も、V4R4 のこれらのラージ・ファイルをサポートします。ただし、バイト範囲のロックが行われている場合はサポートしません。fcntl() API は、バイト範囲のロック・サイズおよびオフセットが 2 GB 限界より小さくなると、NFS クライアントからのファイルについてのみ機能します。

NFS がサポートするファイル・システムおよびラージ・ファイル・サポートについて詳しくは、iSeries Information Center のトピック「**ファイル・システムおよび管理**」の中の「**統合ファイル・システム API のラージ・ファイル・サポート**」を参照してください。

「ルート」ファイル・システム (I)

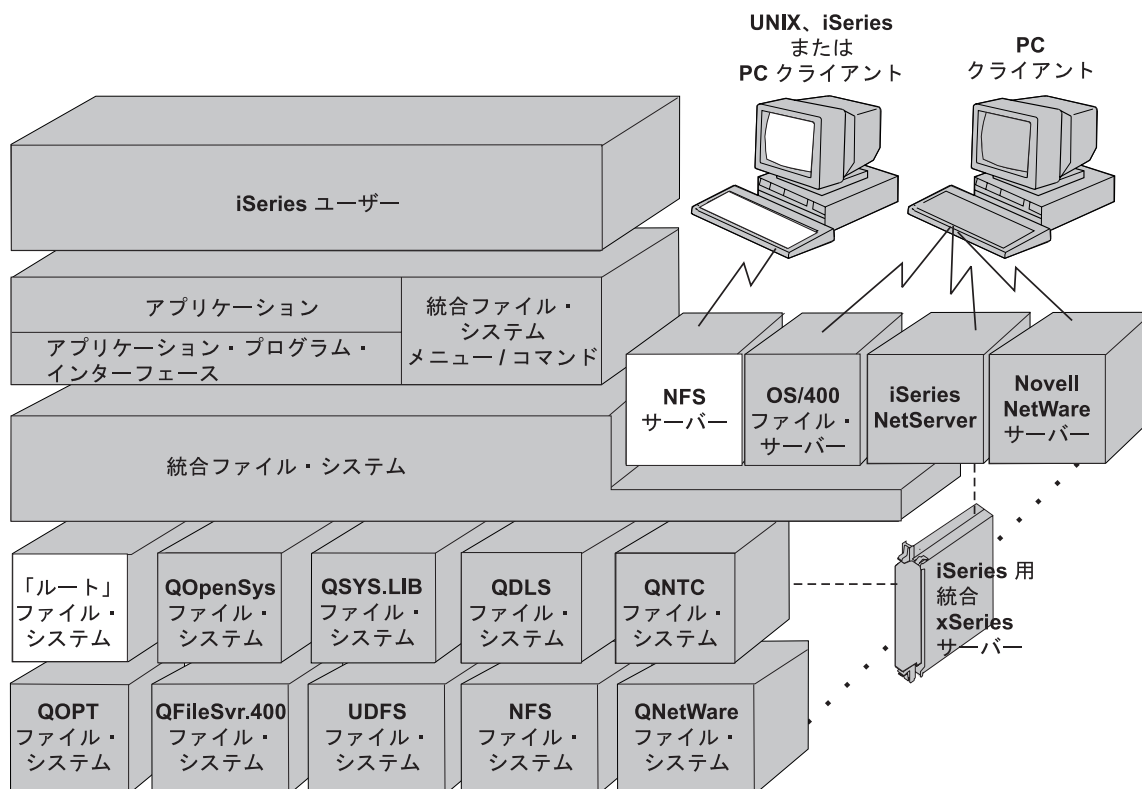


図 46. NFS サーバー経由でアクセスされる「ルート」(I) ファイル・システム

ネットワーク・ファイル・システムでの相違

大文字 / 小文字の区別

サーバーが「ルート」(I) ファイル・システムからエクスポートしたオブジェクトをリモート UNIX クライアントがマウントした場合、そのオブジェクトではいつも大文字 / 小文字は区別されません。

読み取り / 書き込みオプション

クライアントで MOUNT コマンドにどんなオプションが指定されたかにかかわらず、「ルート」(I) からのサーバー・ファイル・システムの一部は読み取り / 書き込みとしてのみ存在します。しかし、ファイル・システムをマウントするクライアントによって、そのファイル・システムがどのように扱われるか、およびそれがクライアント上でどのように機能するかが決まります。

オープン・システム・ファイル・システム (QOpenSys)

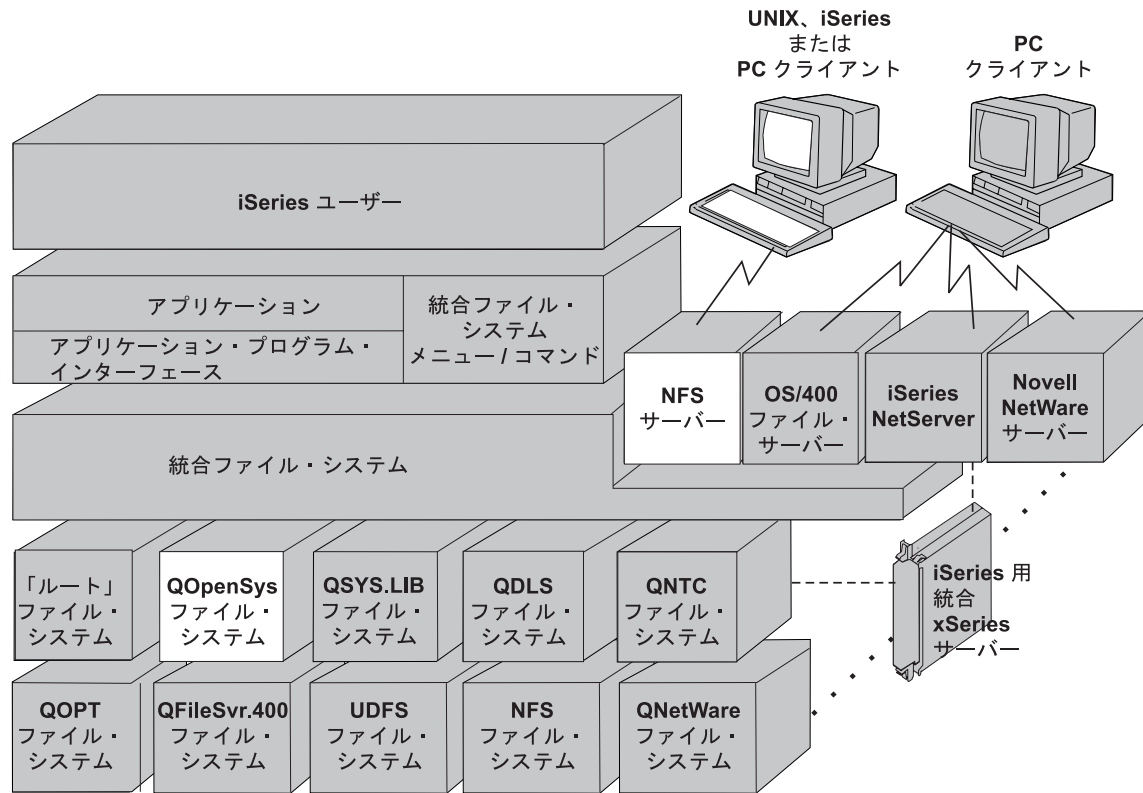


図 47. NFS サーバー経由でアクセスされる QOpenSys ファイル・システム

ネットワーク・ファイル・システムでの相違

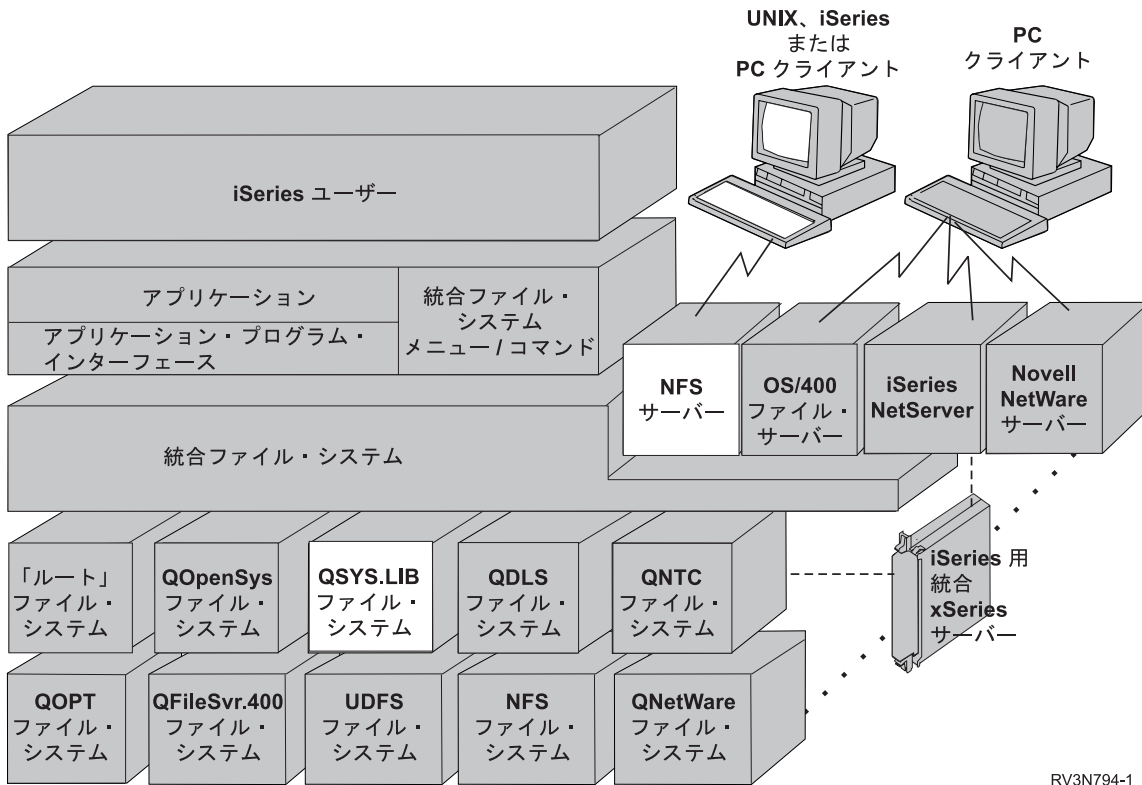
大文字 / 小文字の区別

サーバーが QOpenSys ファイル・システムからエクスポートしたオブジェクトをリモート UNIX クライアントがマウントした場合、そのオブジェクトではいつも大文字 / 小文字が区別されます。

読み取り / 書き込みオプション

クライアントで MOUNT コマンドにどのようなオプションが指定されたかにかかわらず、QOpenSys からのサーバー・ファイル・システムの一部は読み取り専用または読み取り / 書き込みとして存在します。しかし、ファイル・システムをマウントするクライアントによって、そのファイル・システムがどのように扱われるか、およびそれがクライアント上でどのように機能するかが決まります。

ライブラリー・ファイル・システム (QSYS.LIB)



RV3N794-1

図 48. NFS サーバー経由でアクセスされる QSYS.LIB ファイル・システム

ネットワーク・ファイル・システムでの相違

エクスポートと QSYS.LIB

エクスポートできるのは、一部の .LIB および .FILE オブジェクトです。SAVE ファイルをエクスポートし、クライアントがそれをマウントした場合、ファイルを開こうとしても必ず失敗します。一般に、クライアントがアクセスする必要のあるものだけをエクスポートする必要があります。

QSYS.LIB ファイル・システム内のすべてのオブジェクト・タイプは、エラーなしで正常にエクスポートおよびマウントすることができます。エクスポートされてマウントされたファイル・システムに対するすべての操作は、そのオブジェクトがサーバーにローカルに存在する場合と同じように機能します。たとえば、QSYS.LIB ファイル・システム内のオブジェクト・タイプで、すべての入出力 (I/O) 操作をサポートしているのは、データベース・メンバー (.MBR) とユーザー・スペース (.USRSPC) だけです。

NFS は常にデータ・ファイルをバイナリー・モードでオープンします。ソース物理ファイルのメンバーを読み取っている場合は、改行または復帰が行われなことがあります。ファイル・メンバーを改行および復帰して見るためには、まず、CPYTOIMPF コマンドを使用してください。

QPWFSEVER 権限リスト

QPWFSEVER は、リモート・クライアントからアクセスされる QSYS.LIB ファイル・システム内のすべてのオブジェクトについて追加のアクセス要件を提供する権限リスト (オブジェクト・タイプ *AUTL) です。この権限リストに指定された権限は、QSYS.LIB ファイル・システム内のすべてのオブジェクトに適用されます。

このオブジェクトに対する省略時の権限は、PUBLIC *USE 権限です。管理者は、「EDTAUTL (権限リストの編集)」コマンドまたは「WRKAUTL (権限リストの処理)」コマンドを使用して、この権限の値を変更することができます。権限リストに PUBLIC *EXCLUDE 権限を割り当てれば、一般ユーザーがリモート・クライアントから QSYS.LIB オブジェクトにアクセスすることを禁止できます。

マウントと QSYS.LIB

ユーザーは、QSYS.LIB ファイル・システムをクライアントにマウントすることができますが、QSYS.LIB ファイル・システム上にマウントを行うことはできません。ユーザーは、QSYS.LIB をクライアントに直接マウントするのではなく、QSYS.LIB のサブライブラリーをエクスポートしてマウントするべきです。QSYS.LIB には数百のオブジェクトが含まれているからです。すべてのオブジェクトを表示または処理しようとする、クライアントのパフォーマンスに影響するおそれがあります。

ユーザー・スペースのサポート

NFS はユーザー・スペースのエクスポートとマウントをサポートしていますが、以下の例外事項があります。

- ユーザー・スペースは 16 メガバイトを超えることができません。
- 省略時には、ユーザー・スペースは CCSID でタグ付けされるのではなく、コード・ページ用にタグ付けされます。CCSID が要求されると、ユーザーが CCSID を指定した場合に、NFS はデータを変換します。
- ユーザー・スペース・ファイル (*USRSPC オブジェクト・タイプ) は NFS を使用して結合することができますが、データがどのように書き込まれるかに関しては結果を予測できないことがあります。

データベース・メンバーのファイル・モード

データベース・メンバーのファイル・モードはすべて、親ファイルのファイル・モードと同じである必要があります。ユーザーが何を指定したかにはかかわりなく、NFS は新しいデータベース・メンバーを常に親ファイルのファイル・モードで作成します。親ファイルと異なるファイル・モードをユーザーが指定しても、ユーザーはエラー戻りコードを受け取りません。NFS は、ユーザーがどんな指定をしたとしても、新しいメンバーを親ファイルのファイル・モードで作成します。

.FILE オブジェクトのパス名

データベース・メンバーを編集するユーザーは、データベース・レコードの最大長に気を付ける必要があります。レコード長は、物理ファイルを作成する時に指定します。次のいずれかの方法で QSYS.LIB 内に作成する .FILE オブジェクトの省略時のレコード長は、92 バイトです。

1. mkdir() API
2. MKDIR (ディレクトリーの作成) コマンド
3. MD (ディレクトリーの作成) コマンド

4. CRTDIR (ディレクトリーの作成) コマンド

ソース物理ファイルには日付スタンプ (6 バイト) と順序番号 (6 バイト) が含まれており、合計で 12 バイトを使用します。このことを考慮に入れて 92 から 12 バイトを差し引くと、ソース物理ファイルの 1 レコード当たり、省略時には 80 バイトが残ります。他のレコード長を指定した場合にも、ソース物理ファイルの 1 レコード当たりの実際のバイト数は、指定したバイト数から 12 バイトを差し引いた値になります。

バイト範囲ロック

QSYS.LIB は、バイト範囲ロックをサポートしません。クライアントが `fcntl()` API を使用すると、エラー条件 `ENOSYS` が出て失敗します。

大文字/ 小文字の区別

QSYS.LIB は、大文字 / 小文字を区別しません。UNIX クライアントは、一般に大文字 / 小文字を区別します。ディレクトリーがどのように読み取られ、パターンがどのように照合されるかに応じて、UNIX クライアントから QSYS.LIB を表示する際にシステムによってどのファイルが表示されるかが決まります。

たとえば、ある特定のディレクトリーに 1 つのファイル `AMY.FILE` があるとしめます。UNIX の `LS` (リスト) コマンドは、ディレクトリーのすべての内容を表示します。ユーザーが次のコマンドを発行した場合、

```
ls a*
```

システムはファイルやオブジェクトを 1 つも表示しません。

しかし、ユーザーが次のコマンドを発行した場合、

```
ls A*
```

システムは `AMY.FILE` を表示します。

パターン照合はサーバーではなくクライアントで実行され、すべての項目は QSYS から取られます。なぜなら、NFS はディレクトリーを読み取るだけで、パターン照合は行わないからです。 `ls annie.file` というコマンドは、パターン照合を必要としないため、正しく機能します。引用符で囲まれた (拡張された) 名前は、大文字 / 小文字が区別されるので、格納されているとおりに戻されます。

文書ライブラリー・サービス・ファイル・システム (QDLS)

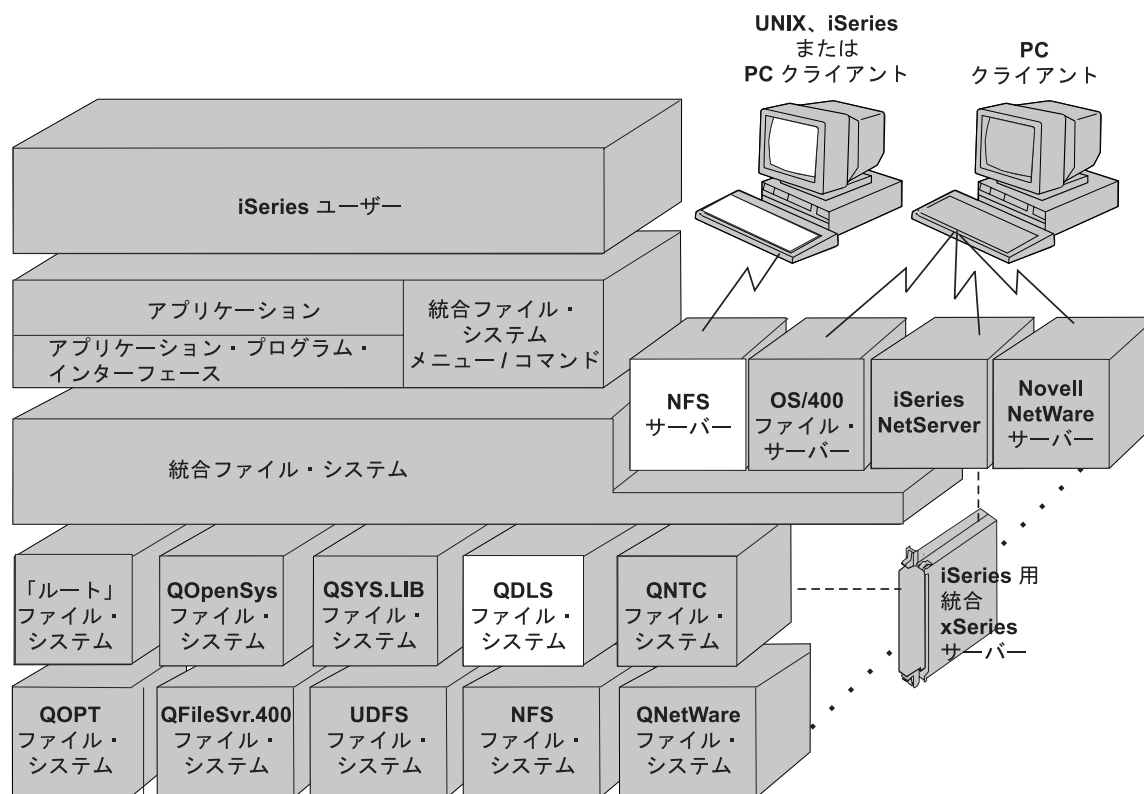


図 49. NFS サーバー経由でアクセスされる QDLS ファイル・システム

ネットワーク・ファイル・システムでの相違

マウントと QDLS

ユーザーは、QDLS ファイル・システムをクライアントにマウントすることができますが、QDLS ファイル・システム上にマウントを行うことはできません。

ファイルの作成

ユーザーは、最上位の /QDLS ディレクトリーに通常のファイルを作成することはできません。ユーザーがファイルを作成できるのは、/QDLS のサブディレクトリー内だけです。

パス名の長さ

QDLS 構成要素の名前は最大 8 文字まで、拡張子 (指定する場合) は最大 3 文字までです。パス名の最大長は、絶対パス名が /QDLS で始まる場合、82 文字です。QDLS ファイル・システムをクライアントにマウントする場合、パス名全体は 87 文字を超えてはなりません (/QDLS を含む)。

無名ユーザー

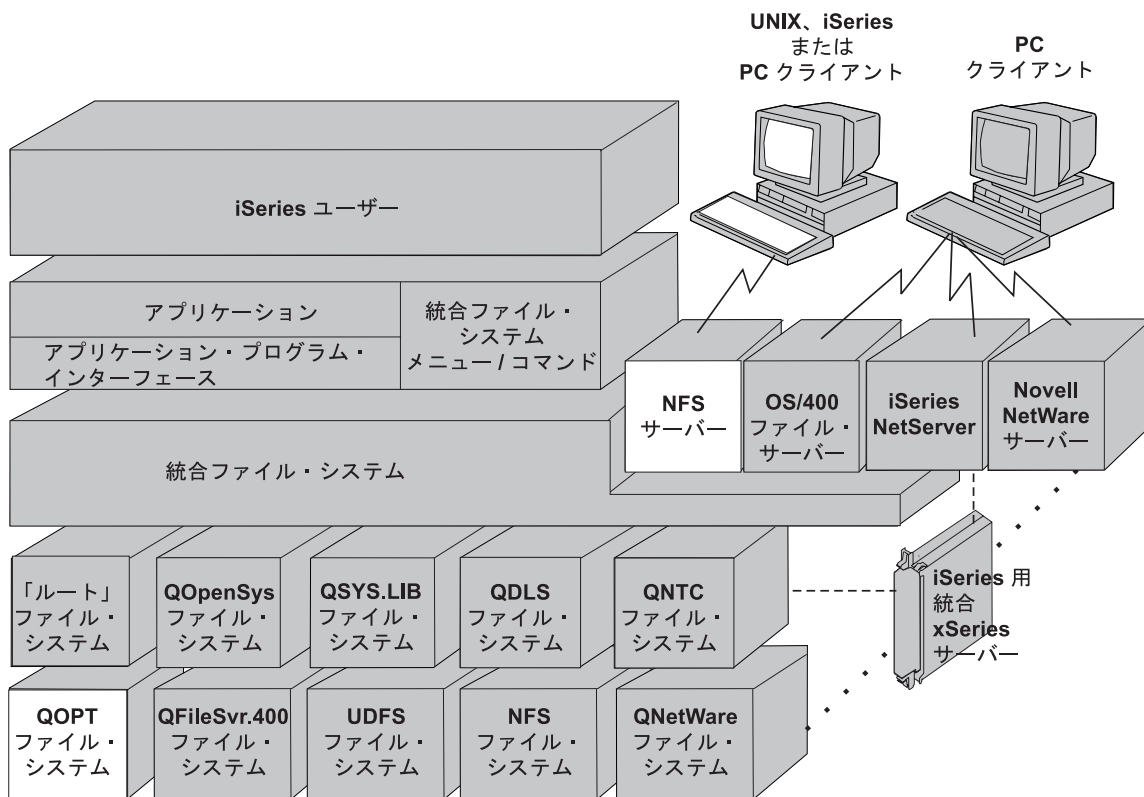
クライアントは、NFS サーバー経由で QDLS ファイル・システムをマウントすることができます。しかし、無名 (匿名) のクライアントから QDLS 内のオブジェク

トを使用しようとする場合は、まず最初に登録操作によって文書ライブラリー・サービスに登録しなければなりません。管理者は、「ADDDIRE (ディレクトリー項目の追加)」コマンドを使用して、QNFSANON または他のユーザーを QDLS フォルダー管理サービス (FMS) に登録できます。QNFSANON ユーザー・プロファイルを FMS に登録しない場合、QNFSANON にマップされるすべての無名クライアント要求は、サーバーにおいて失敗します。

QDLS ファイル・システムの詳細については、以下の資料を参照してください。

- iSeries Information Center のトピック「文書ライブラリー・サービス・ファイル・システム (QDLS)」

光ディスク・ファイル・システム (QOPT)



RV3N778-1

図 50. NFS サーバー経由でアクセスされる QOPT ファイル・システム

ネットワーク・ファイル・システムでの相違

マウントと QOPT

ユーザーは、QOPT ファイル・システムをエクスポートして、それをクライアントにマウントすることができます。QOPT ファイル・システム上にマウントを行うことはできません。状態がないという NFS の特性や、光ディスク記憶域は光ディスク・ボリューム全体を再初期設定しないと再利用できないという事実のため、QOPT ファイル・システムは、いったんエクスポートされてクライアントにマウントされ

た後は読み取り専用のファイル・システムとして取り扱われます。サーバー上のネイティブ・ユーザーは、引き続き QOPT ファイル・システムに対する書き込みを実行できます。

注: QOPT ファイル・システム内のどのパスをエクスポートする場合も、読み取り専用 (RO) オプションを指定する必要があります。これを指定しないと、エクスポート要求は失敗します。

大文字 / 小文字の区別

QOPT は、大文字 / 小文字を区別しません。オブジェクト名の中で英小文字が使用されていると、QOPT はそれを大文字に変換します。したがって、/QOPT/volume/dir/file というパス名は、/QOPT/VOLUME/DIR/FILE と同じパスを表します。

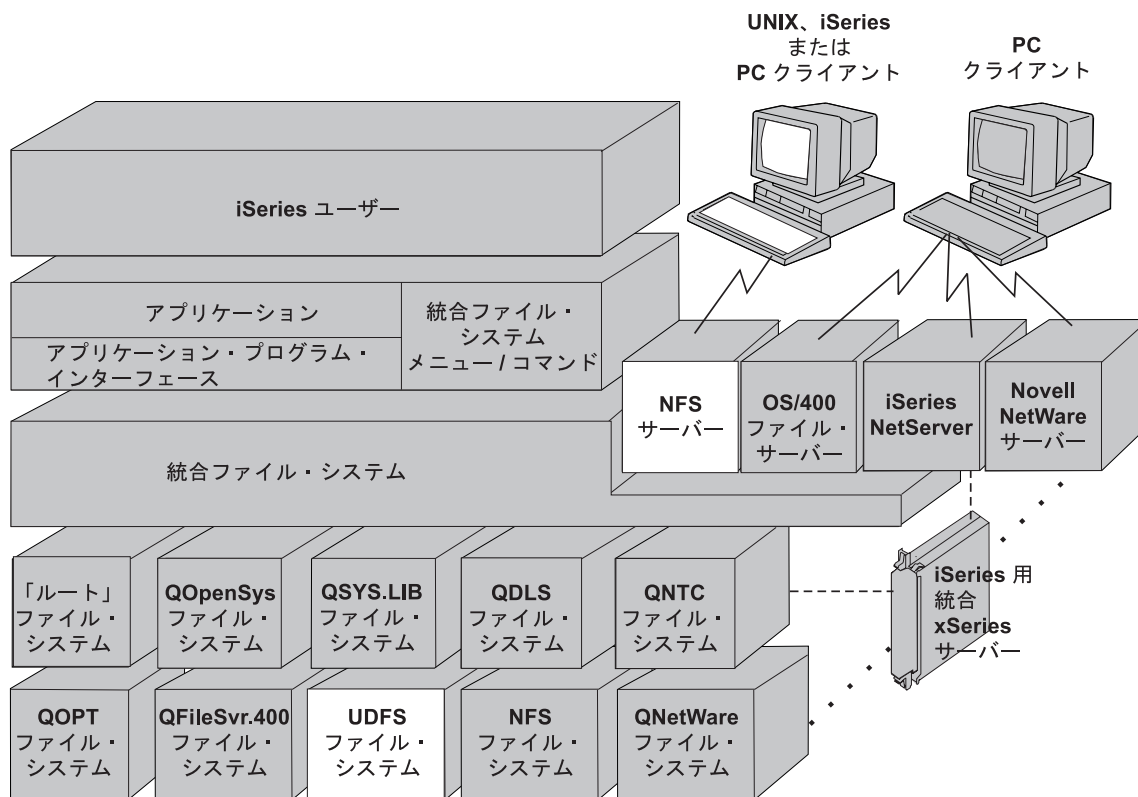
機密保護と許可

QOPT ファイル・システムは、ファイルまたはディレクトリー・レベルの機密保護ではなく、**ボリューム・レベルの機密保護**を提供します。それぞれの光ディスク・ボリュームは、権限リストによって機密保護が実施されます。あるボリューム上のディレクトリーまたはファイルにアクセスする必要があるユーザーは、その光ディスク・ボリュームにアクセスする必要があります。システム管理者、または権限リスト管理権限を持つユーザーは、次のようにしてアクセスを許可することができます。

1. 「光ディスク・ボリュームの処理 (WRKOPTVOL)」コマンドを使用して、そのボリュームの機密保護を実施している権限リストを見つけます。
2. 「権限リストの編集 (EDTAUTL)」コマンドを使用して、権限リストにユーザーを追加します。

光ディスクの機密保護については、光ディスクのサポートに関するブックを参照してください。

ユーザー定義ファイル・システム (UDFS)



RV3N779-1

図 51. NFS サーバー経由でアクセスされる UDFS ファイル・システム

ネットワーク・ファイル・システムでの相違

大文字 / 小文字の区別

サーバーが UDFS からエクスポートしたオブジェクトをリモート UNIX クライアントがマウントするとき、大文字 / 小文字が区別されるかどうかは、ユーザーが UDFS を作成した方法に応じて異なります。UNIX クライアントにマウントされた UDFS は、大文字 / 小文字を区別するかどうかをディレクトリー・ツリーの途中で変更することができます。

システムおよびユーザー補助記憶域プール

UDFS の内容はユーザー補助記憶域プール (ASP) に入っていますが、UDFS のブロック特殊ファイルそのものはシステム ASP に入っています。

UNIX クライアントの管理者

ネットワーク・ファイル・システムでの相違

ディレクトリー権限

UNIX クライアントには、ディレクトリー内にファイルを作成するために、*X (実行) 権限が必要です。このことは、マウントされた iSeries ファイル・システムを処理している場合に当てはまります。省略時には、UNIX クライアントは、*X (実行) 権限を付加してディレクトリーを作成することはありません。UNIX クライアントは、ディレクトリーの作成時に、UMASK モード・ビットをモード・ビットを変更して実行ビットを ALL にしなければならない場合があります。こうすれば、そのディレクトリー内にオブジェクトを作成できるようになります。

第 7 章 NFS の開始、遮断、および回復

NFS の開始は、各マシンごとに独立して別個に実行されます。1 つのサーバーで NFS 構成要素が開始することにより、別のサーバーの NFS 構成要素の開始がトリガーされることはありません。たとえば、ネットワーク・ロック管理プログラム (NLM) をクライアントで開始しても、サーバーの NLM が自動的に開始されることはありません。NFS を正しく機能させるためには、各種のデーモンを特定のサーバーまたはネットワークにおいて開始する際を守るべき暗黙の順序があります。

たとえば、特定のマシン上で NFS サーバー・デーモンが操作可能になっていなければ、別のサーバー上の NFS クライアントからそのサーバーに要求を発行しても正しく実行されません。同様に、特定のサーバーのマウント・デーモンが操作可能になっていないと、別のマシンからそのデーモンにマウント要求を発行しても正しく実行されません。

管理者は、NFS サーバーおよび NFS クライアントを、任意のときに任意の順序で開始させたり終了させたりすることができます。この処理によってエラー・メッセージが出されることはありません。とはいえ、快適に操作するためには、NFS の開始および終了を、この後で説明する順序に従って行うことをお勧めします。

注：ファイル・サーバー・サポート/400 (FSS/400) と NFS は、同時に開始しないようにしてください。iSeries 上で一度に稼働させることができるのは、これらのサーバー・アプリケーションのどちらか一方だけです。もし FSS OS/400用 が NFS の開始時に稼働していると、RPC バインダー・デーモン (ポート・マップ) はポート 111 に接続できません。さらに、RPC デーモンはサーバー・プロセスにポートを割り当てることができません。NFS の使用をお勧めします。

TCP/IP の構成

NFS サポートを開始するためには、まず、TCP/IP をインストールし、適切に構成しておく必要があります。NFS は、システム構成によって、ローカル・サーバーの名前を決定します。ドメイン名 (ドメイン・ネーム) サーバーを使用していないと、ホスト・テーブルには、「ロング」ネームと「ショート」ネーム両方に対して 1 つの項目が含まれなければなりません。TCP/IP とそのオプションを正しく構成するには、以下のステップに従ってください。

1. 「TCP/IP 構成」メニュー (CFGTCP) に進みます。
2. オプション 12 「TCP/IP ドメイン変更 (CHGTCPDMN) (Change TCP/IP Domain (CHGTCPDMN))」を選択します。

「ホスト名」パラメーターにショート・ネームが入っています。ロング・ネームはショート・ネームに「.」と「ドメイン名」(次のパラメーター) を付加したものです。たとえば、ASHOST01 はショート・ネームで、ASHOST01.NETWORK.DOMAIN.COM はロング・ネームです。

3. 必要であれば情報を更新します。

ホスト名検索優先順位によって、ホスト名を解決する順序をシステムに示します。*LOCAL は、TCP/IP ホスト・テーブルを最初に検索することを意味し、

*REMOTE は、指定された IP アドレスの DNS を最初に検索するよう指示します。この画面の情報を更新する必要がなければ、PF12 を押します。

4. オプション **10** (「TCP/IP ホスト・テーブル項目の処理 (Work with TCP/IP Host Table Entries)」) を選択します。
5. ローカル・システムの IP アドレス に対してはロング・ネームとショート・ネーム両方のホスト名項目があることを検証します。このアドレスがわからない場合は、CFGTCP メニューからオプション 1 を選択します。さらに、次の 2 つのオプションのいずれかを選択できます。
 - a. あるアドレスに名前を追加するには、CFGTCP メニューからオプション 2 (「変更」) を選択します。
 - b. すべての新規アドレスと名前を追加するには、CFGTCP メニューからオプション 1 を選択します。
6. LOOPBACK と LOCALHOST という名前が、ホスト・テーブル内の IP アドレス 127.0.0.1 に関連付けられているかを確認します。
7. アクセスする必要のある各 NFS サーバーのロング・ネームとショート・ネームがホスト・テーブルに含まれているかを確認します。これが必要なのは、システムを NFS クライアントとして使用する予定がある場合だけです。ホスト・テーブルにこれらの名前が含まれておらず、DNS も使用されていない場合は、ロング・ネームとショート・ネームを追加する必要があります。

次に示すのは TCP/IP ホスト・テーブルの例です。

Opt	Internet Address	Host Name
	19.45.216.4	THISHOST THISHOST.COMPANY.DOM1.COM
	19.45.216.93	NFSSERV1 NFSSERV1.COMPANY.DOM1.COM
	127.0.0.1	LOOPBACK LOCALHOST

不適切な開始および遮断によって起こる不具合

ネットワーク・ファイル・システムのさまざまな構成要素は、複雑で相互に依存し合うシステムを作り上げています。つまり、それらの構成要素は、正常に、また効率的に機能するために互いにそれぞれを必要としています。それで、ユーザーが不適切な開始または遮断を行うと正常に作動しなくなる機能が多数あります。

- まず最初に RPC バインダー・デーモンを開始しておかないと、すべてのローカル・サーバー・ポートが正しくマップされません。クライアントからの要求が処理されません。
- サーバーがマウント・デーモンをロードする前にクライアントが開始すると、そのクライアントが行うマウント要求はすべて失敗します。さらに、NLM および NSM デーモンがロックを許可できるようにするには、クライアントとサーバーの両方で その 2 つのデーモンを開始しておかなければなりません。
- サーバーがエクスポート・コマンドを発行する前にクライアントが開始してマウント要求を行うと、すべてのマウント要求は失敗します。

- RPC バインダー・デーモン (ポート・マッパー) を最初に終了させてしまうと、その他のすべてのデーモンは RPC バインダー (ポート・マッパー) デーモンから登録を抹消できなくなります。
- すべてのデーモンを終了させなかった場合、次の開始時に、どのデーモンが作動しているどのデーモンが作動していないのか混乱が生じる恐れがあります。

適切な開始シナリオ

NFS サーバーでの開始

1. ユーザーが、RPC バインダー (ポート・マッパー) デーモン (QNFSRPCD) を開始します。このデーモンは、開始した後、既知のポート (#111) にサービスを登録するローカル RPC 要求が来るのを待機します。このデーモンは、さらに、ローカル・サービスを照会するリモート RPC 要求を待機します。
2. ユーザーが、エクスポート・コマンドを呼び出し、`/etc/exports` ファイルに入っている情報に基づいて、エクスポートされたディレクトリーのリストがエクスポート・テーブルに作成されます。
3. ユーザーが、1 つまたは複数の NFS サーバー・デーモン (QNFSNFSD) を開始します。このデーモンは、ローカル RPC バインダー・デーモンに登録します。ローカル RPC バインダー・デーモンは、NFS サーバーがどのポートで要求を待機しているか (標準は #2049) を知っています。すべてのサーバー・デーモンは、この同じポートを使用します。その後、NFS サーバー・デーモンは、ローカル・ファイルへのアクセスを指示する、NFS クライアントからの RPC 要求をそのポートで待機します。
4. ユーザーが、マウント・デーモン (QNFSMNTD) を開始します。このデーモンは、ローカル RPC バインダー・デーモンに登録します。その後、ローカル・ファイル・システムのマウントを指示する、NFS クライアントからの RPC 要求をそのポートで待機します。
5. ユーザーが、NSM デーモン (QNFSNSMD) を開始します。このデーモンは、ローカル RPC バインダー・デーモンに登録します。その後、システムをモニターするよう指示する RPC 要求を、割り当てられたポートで待機します。
6. ユーザーが、NLM デーモン (QNFSNLMD) を開始します。このデーモンは、ローカル RPC バインダー・デーモンに登録します。その後、ロックを管理するよう指示する RPC 要求を、割り当てられたポートで待機します。

一般的な開始では、「ネットワーク・ファイル・システム・サーバーの開始 (STRNFSSVR)」コマンドの `SERVER` パラメーターに `*ALL` を指定すると、すべてのデーモンが自動的に正しい順序で開始するようになります。

NFS クライアントでの開始

1. ユーザーが、RPC バインダー (ポート・マッパー) デーモンを開始します (まだ操作可能になっていない場合)。特定の 1 つのシステムごとに、1 つのポート・マッパーがクライアントとサーバーの両方のために使用されます。
2. ユーザーが、1 つまたは複数のブロック I/O デーモン (QNFSBIOD) を開始します。このデーモンは、サーバーから送信されたデータおよび属性のキャッシングを制御します。

3. ユーザーが、NSM デーモンを開始します (まだ操作可能になっていない場合)。特定の 1 つのシステムごとに、1 つの NSM がクライアントとサーバーの両方のために機能します。
4. ユーザーが、NLM デーモンを開始します (まだ操作可能になっていない場合)。特定の 1 つのシステムごとに、1 つの NLM がクライアントとサーバーの両方のために使用されます。

STRNFSSVR (ネットワーク・ファイル・システム・サーバーの開始) コマンド

目的

「ネットワーク・ファイル・システム・サーバーの開始 (STRNFSSVR)」コマンドは、ネットワーク・ファイル・システム (NFS) サーバー・デーモンの 1 つまたは全部を開始します。

SERVER(*ALL) オプションを使用する必要があります。そうすれば、各デーモンが次に示す順序で開始され、エクスポート・コマンドが呼び出されます。この順序は、ネットワーク・ファイル・システムを開始する際の推奨されている順序です。

- リモート・プロシージャ・コール (RPC) バインダー・デーモン
- ブロック I/O (BIO) デーモン
- エクスポート・コマンドの呼び出し
- サーバー (SVR) デーモン
- マウント (MNT) デーモン
- ネットワーク状況モニター (NSM) デーモン
- ネットワーク・ロック管理プログラム (NLM) デーモン

デーモンを 1 つだけ開始することにした場合は、NFS の各デーモンを開始するための適切な順序と、上記の順序以外の順序で各デーモンを開始した場合に起こり得る結果について、よく理解しておいてください。

すでに実行されているデーモンを開始しようとしても、そのコマンドが失敗することはなく、開始するように要求した他のデーモンの開始処理が続行されます。デーモンがすでに実行されている場合は、コマンドから診断メッセージ CPDA1BA が出されます。最善の結果を得るためには、すべての NFS デーモンを終了させてから、STRNFSSVR コマンドを実行してください。

NFS サーバー・デーモンの表示

NFS サーバーのデーモンを表示するには、「活動ジョブの処理 (WRKACTJOB)」コマンドを使用できます。そして、サブシステム QSYSWRK に以下のジョブがあるかどうかを調べます。

- QNFSRPCD、RPC バインダー・デーモン (RPCD)
- QNFSNFSD、NFS サーバー・デーモン (NFSD、このデーモンは複数のこともある)
- QNFSMNTD、マウント・デーモン (MNTD)
- QNFSNSMD、ネットワーク状況モニター・デーモン (NSMD)
- QNFSNLM、ネットワーク・ロック管理プログラム・デーモン (NLM)

追加のデーモンが開始されます。このパラメーターは、「SERVER(*SVR) および SERVER(*ALL)」を指定した場合にだけ、使用することができます。

- STRNFSSVR コマンドの NBRBIO パラメーターでは、実行させたい NFS ブロック I/O (*BIO) デーモン・ジョブの数を指定します。このパラメーターに指定した数が、サーバーで既に実行されているブロック I/O デーモンの数より大きい場合は、追加のデーモンが開始されます。このパラメーターは、「SERVER(*BIO) および SERVER(*ALL)」を指定した場合にだけ、使用することができます。
- STRNFSSVR コマンドの STRJOBTIMO パラメーターでは、各デーモンが正しく開始されるまで待機する秒数を指定します。このタイムアウト時間以内にデーモンが開始されない場合、そのコマンドは失敗します。

例

例 1: すべての NFS デーモンを開始する

```
STRNFSSVR SERVER(*ALL) STRJOBTIMO(*NOMAX)
```

このコマンドは、すべての NFS デーモンを開始し、各デーモンが開始するまで無制限に待機します。事前にデーモンが実行されていることがないようにする必要があります。

例 2: デーモンを 1 つだけ開始する

```
STRNFSSVR SERVER(*MNT)
```

このコマンドは、NFS マウント・デーモンを開始し、そのデーモンが開始するまで省略時値の最大 30 秒まで待機します。マウント・デーモンが事前に実行されていることがないようにしてください。その他のデーモンは適切な順序で開始させておく必要があります。

適切な遮断シナリオ

NFS サーバーを正しく遮断すれば、すべてのジョブを終了させ、すべての要求の実行を完了させることができます。一般に、サーバーを遮断するために必要な順序は、サーバーを開始するために必要な処理のちょうど反対です。

1. ユーザーが、NLM デーモン (QNFSNLMD) を終了させます。
2. ユーザーが、NSM デーモン (QNFSNSMD) を終了させます。リモート・クライアントによってローカル・ファイルに保持されているすべてのロックは解除されます。
3. ユーザーが、マウント・デーモン (QNFSMNTD) を終了させます。リモート・クライアントがマウントしているすべてのローカル・ファイル・システムは、マウント解除されます。
4. ユーザーが、1 つまたは複数の NFS サーバー・デーモン (QNFSNFSD) を終了させます。
5. ユーザーが、RPC バインダー (ポート・マッパー) デーモン (QNFSRPCD) を終了させます。

「ネットワーク・ファイル・システム・サーバーの終了 (ENDNFSSVR)」コマンドの SERVER パラメーターに *ALL を指定すると、自動的にすべてのデーモンが正しい順序で終了されます。

クライアントを遮断する際の順序は、開始プロセスの場合のほとんど反対のプロセスになります。

1. ユーザーが、NLM デーモンを終了させます (クライアント上にある場合)。1 つのサーバー NLM が、クライアントとサーバーの両方のために実行されます。
2. ユーザーが、NSM デーモンを終了させます (クライアント上にある場合)。1 つのサーバー NSM が、クライアントとサーバーの両方のために実行されます。
3. ユーザーが、1 つまたは複数のブロック I/O デーモン (QNFSBIOD) を終了させます。
4. ユーザーが、RPC バインダー (ポート・マッパー) デーモンを終了させます。

遮断に関する考慮事項

TCP/UDP タイムアウトの対立

NFS サーバーを終了するとき、ソケット・ポートがクローズします。NFS サーバーをその後すぐに再始動すると、サーバーがソケット・ポートに接続できないことがあります。基礎になっている iSeries の TCP/IP サポートが、このポートを短い期間だけアクセス不能にするからです。NFS サーバーを再始動する前に短い時間だけ待つようにすれば、ソケット・ポートに通常どおり接続できるはずです。

ENDNFSSVR (ネットワーク・ファイル・システム・サーバーの終了) コマンド

目的

「ネットワーク・ファイル・システム・サーバーの終了 (ENDNFSSVR)」コマンドは、ネットワーク・ファイル・システム (NFS) サーバー・デーモンの 1 つまたは全部を終了させます。

SERVER(*ALL) を使用するべきです。そうすれば、各デーモンが以下の順序で終了します。(これは、ネットワーク・ファイル・システムのデーモンを終了するための推奨されている順序です。)

- ネットワーク・ロック管理プログラム (NLM) デーモン
- ネットワーク状況モニター (NSM) デーモン
- マウント (MNT) デーモン
- サーバー (SVR) デーモン
- ブロック I/O (BIO) デーモン
- リモート・プロシージャー呼び出し (RPC) デーモン

デーモンを 1 つだけ終了させることにした場合は、NFS の各デーモンを終了させるための適切な順序と、上記の順序以外の順序で各デーモンを終了させた場合に起こり得る結果について、よく理解しておいてください。

実行されていないデーモンを終了させようとしても、そのコマンドが失敗することではなく、終了させるように要求した他のデーモンの終了処理が続行されます。

NFS クライアント・デーモンの表示

NFS クライアントのデーモンを表示するには、「活動ジョブの処理 (WRKACTJOB)」コマンドを使用できます。そして、サブシステム QSYSWRK に以下のジョブがあるかどうかを調べます。

- QNFSRPCD、RPC バインダー・デーモン (RPCD)
- QNFSMNTD、マウント・デーモン (MNTD)
- QNFSNSMD、ネットワーク状況モニター・デーモン (NSMD)
- QNFSNLMD、ネットワーク・ロック管理プログラム・デーモン (NLMD)
- QNFSBIOD、ブロック I/O デーモン (BIOD、このデーモンは複数のこともある)

制約事項

1. このコマンドを使用するには、*IOSYSCFG 特殊権限が必要です。

ENDNFSSVR コマンドとそのパラメーターおよびオプションについては、iSeries Information Center のトピック「CL」を参照してください。

ENDNFSSVR 画面

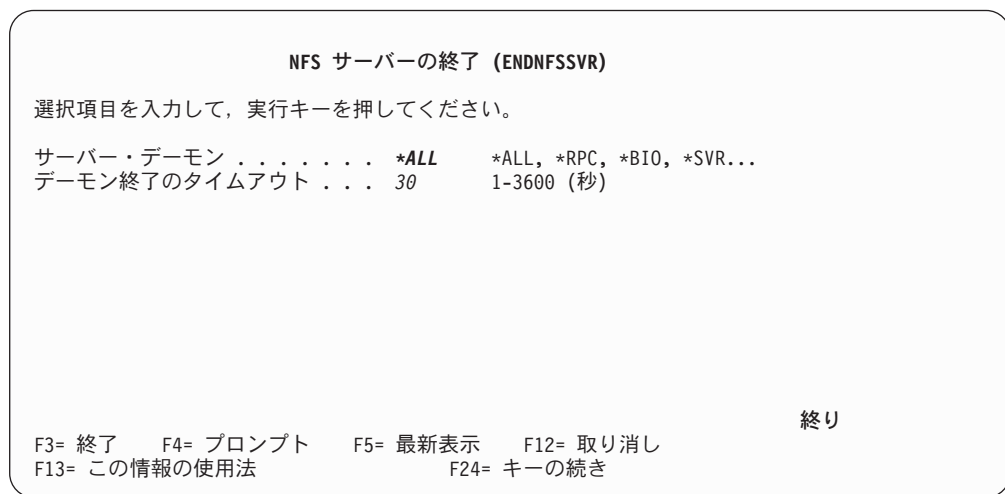


図 53. 「NFS サーバーの終了 (ENDNFSSVR)」画面の使用法

ENDNFSSVR コマンドを使用するときには、次のような多くのパラメーターを指定することができます。

- ENDNFSSVR コマンドの SERVER パラメーター (必須) では、終了させるネットワーク・ファイル・システム・デーモンのジョブを指定します。
- ENDNFSSVR コマンドの ENDJOBTIMO パラメーターでは、各デーモンが正しく終了するまで待機する秒数を指定します。このタイムアウト時間以内にデーモンが終了しない場合、そのコマンドは失敗します。

例

例 1: すべてのデーモンを終了させる

```
ENDNFSSVR SERVER(*ALL)
```

このコマンドは、実行されているすべての NFS デーモンのジョブを終了させます。

例 2: デーモンを 1 つだけ終了させる

```
ENDNFSSVR SERVER(*MNT) ENDJOBTIMO(*NOMAX)
```

このコマンドは、NFS マウント・デーモンを終了させ、それが終了するまで無制限に待ちます。このコマンドを実行する時点で、マウント・デーモンは実行されており、その他のデーモンは適切な順序で終了させてあります。

iSeries ナビゲーターでの NFS の開始または停止

NFS サーバーは、iSeries ナビゲーターによって管理することもできます。NFS サーバーのデーモンを開始または停止させるには、次のステップを実行してください。

1. 「**iSeries ナビゲーター**」で、ご使用のサーバーを展開します。
2. 「**ネットワーク**」を展開します。
3. 「**サーバー**」を展開します。
4. 「**TCP/IP**」をクリックします。右側のパネルに、TCP/IP サーバーのリストが表示されます。
5. 「**NFS**」を右マウス・ボタン・クリックして、実行したいタスクに応じて、「**開始**」または「**停止**」を選択します。すべての NFS サーバー・デーモンを開始または停止できますが、それらを個別に開始または停止することもできます。

次の画面図は、これらのステップを示しています。この図で、NFS は停止状況になっています。

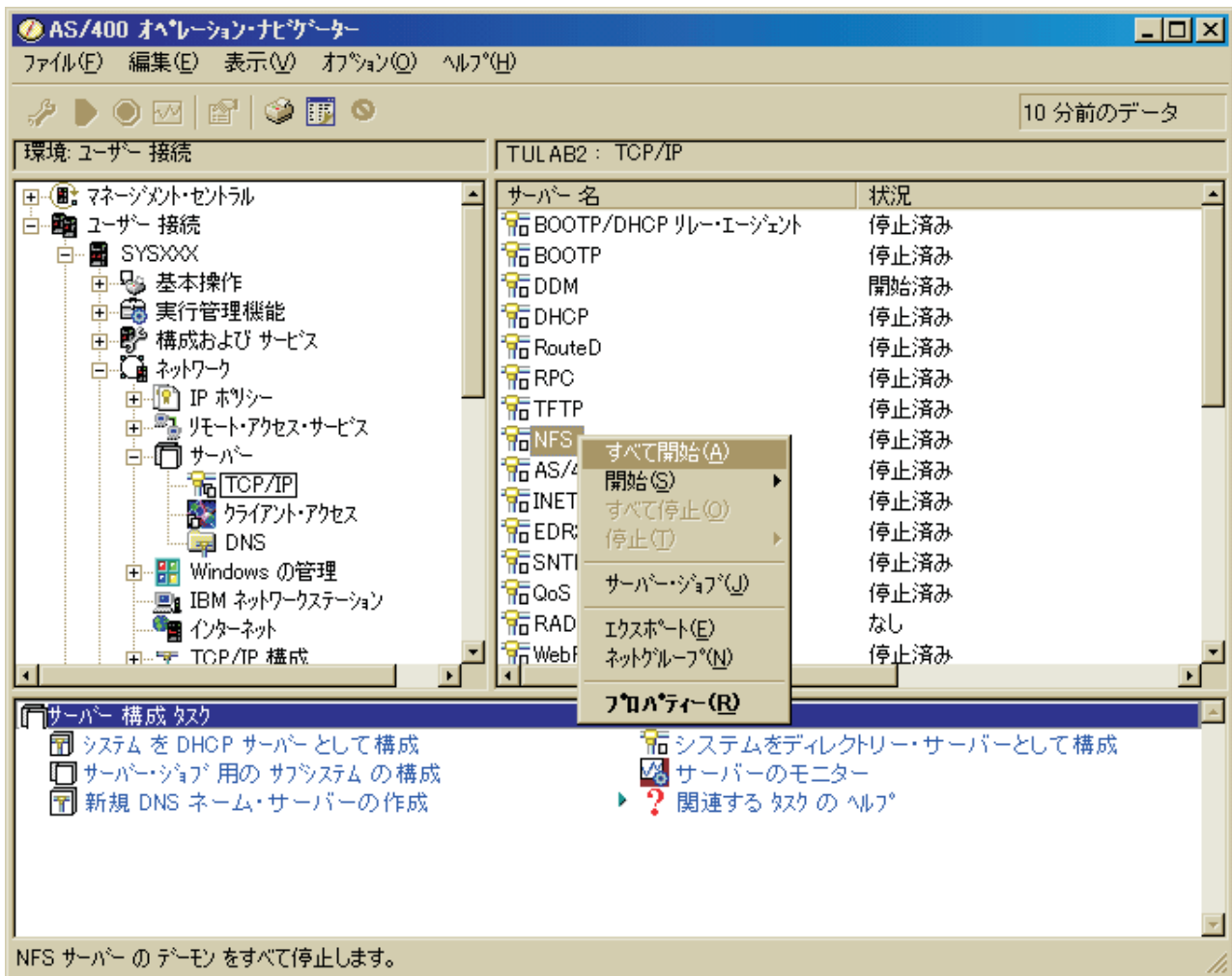


図 54. NFS サーバー・デーモンの開始または停止

「NFS」を右マウス・ボタン・クリックしてから、「プロパティ」を選択すれば、個々のデーモンの状況を表示することもできます。この選択によって次のダイアログ・ボックスが表示されます。

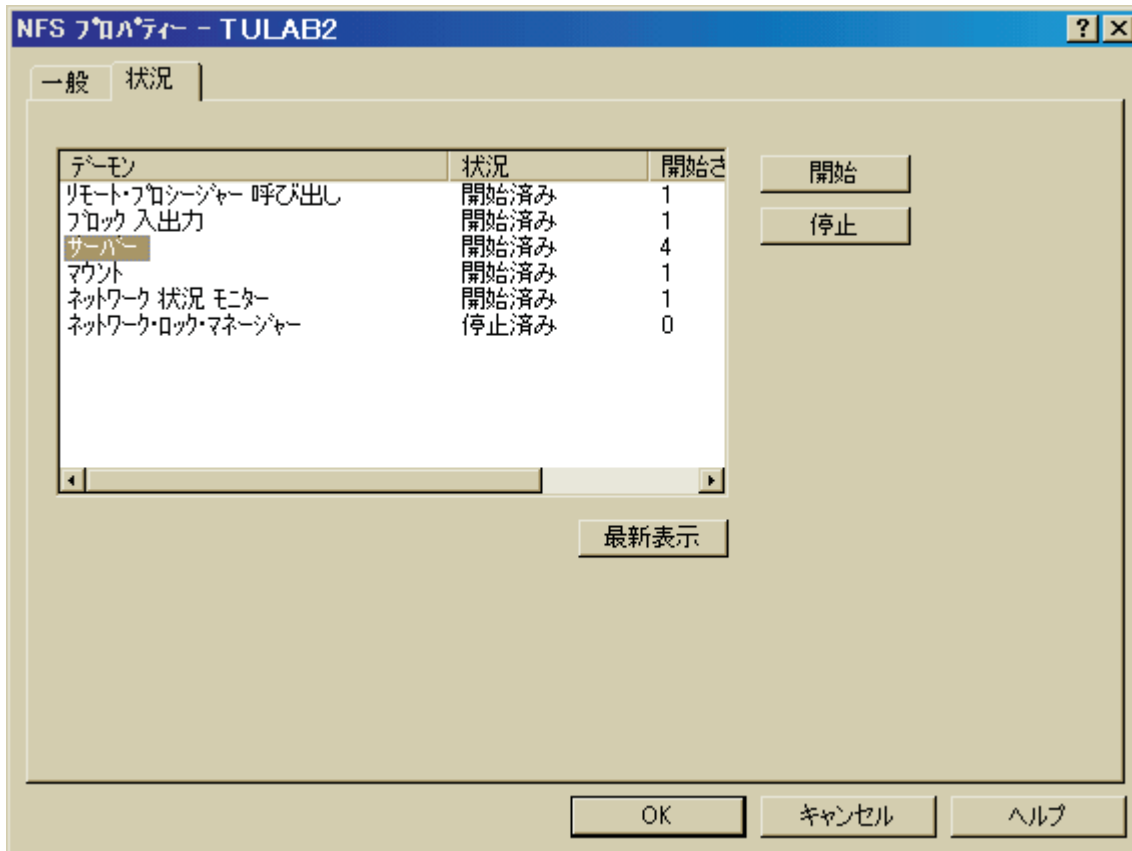


図 55. 「NFS プロパティ」 ダイアログ・ボックス

たとえば、Chris Admin は、スループットを向上させるために、サーバー・タイプのデーモンを 4 つ開始することにしました。このようなデーモンは、前のダイアログ・ボックスの「一般」タブから最大 20 個まで開始できます。ネットワーク・ロック管理プログラム・デーモンが停止していることに注意してください。これは、開始しようとしたときに問題が生じたことを示している場合があります。あるいは、バイト範囲ロックが必要ないので、管理者が意図的に終了を選択したことを示していることもあります。

NFS と RPC は両方とも同じリモート・プロシージャ呼び出しのバインダー・プログラム・デーモンを共有します。NFS の開始または停止によって、RPC が開始または停止しますし、その逆も行われます。たとえば、RPC サーバーを停止すると、リモート・プロシージャ呼び出しの NFS デーモンが停止し、それによって、NFS が正しい機能を停止する場合があります。

ロックと回復

クライアントは、マウントされたサーバー・ファイル・システムにもロックを設定することができます。ロックを利用すると、ユーザーはファイルの全部または一部への共用アクセスや排他アクセスを行うことができます。ユーザーがあるファイルをロックした場合、そのファイルにアクセスする必要のあるプロセスは、ロックが解除されるのを待ってからでないと処理を続行できません。

クライアントがファイルの全部または一部に設定できるロックには、**排他**と**共用**の2種類があります。あるクライアントが排他ロックを獲得した場合、他のプロセスはそのファイルのその部分に対してロックを取得することができません。あるクライアントが共用ロックを獲得した場合、他のプロセスはそのファイルの同じ部分に対して共用ロックを取得することができます。

なぜファイルをロックする必要があるか

ユーザーまたはアプリケーションは、NFS を介してバイト範囲ロックを使用することにより、NFS のパフォーマンスを改善できます。NFS プロトコルは状態なしのプロトコルなので、個々の特定のクライアントはサーバー上でなされた変更を感知することができません (クライアント側のキャッシュのため)。ロック機能は、サーバーがファイルを更新するクリティカルな期間内にこの問題が発生しないようにします。

ファイルをロックする方法

クライアント側のアプリケーションは、ユーザーの制御のもとで、NFS を使用してマウントされたリモート・ファイルに対してロック要求を開始することができます。次に、クライアントは、その命令を RPC 要求を介して自分のローカル・ネットワーク・ロック管理プログラム (NLM) デーモンに送ります。クライアント側の NLM デーモンは、その要求を RPC を介して対応するサーバー側の NLM に送ります。

ユーザーは、`fcntl()` API を呼び出すことにより、ファイルの各部分をロックしたり、ロック・パラメーターを指定したりできます。`fcntl()` API の完全な解説は、iSeries Information Center のトピック「API」を参照してください。

その後、サーバー側の NLM デーモンが、対応するファイルに対してロック操作を実行します。サーバーの NLM デーモンは応答を生成し、クライアントの NLM デーモンはそれを受け取ります。クライアントの NLM デーモンは、NFS クライアントに対して、ロック要求の結果を示す応答を生成します。最後に、NFS クライアントは、その結果をアプリケーションを経由してユーザーに戻します。

状態なしのシステムと状態ありの操作

状態がないというネットワーク・ファイル・システムの特徴は、状態があるというファイル・ロックの特性とうまくなじみません。クライアントかサーバーのデーモンがロックを保持している間にそのどちらかに障害が起きると、ロックの解除に関して問題が起きます。

もしロック要求を認可されたクライアントに障害が起きた場合は、起動時に以下の一連の操作が実行されて、ロックが回復されます。

1. ユーザーがシステムで NSM デーモンを再起動すると、そのデーモンは状態変更 RPC をネットワーク内の他の NSM デーモンに送信します。このメッセージは、障害の起きたシステムがネットワーク内で感知している他の NSM デーモンにだけ送信されます。
2. リモート・システムからの状態変更を受け取った NSM は、RPC を使用してローカル NLM と通信します。NSM は NLM に対して、システムに障害が起きた時点でロックが保持されていたことを通知します。

3. 障害が起きたのがサーバーであれば、回復の後、クライアントのプロセスが保持していたと思われるファイル・ロックをローカル・サーバー NLM が再設定しようとしています。どのリモート・クライアントがサーバー上にロックを保持しているかに関する情報は、ローカル・ファイル /etc/statd に格納されます。ユーザーは、このファイルを決して編集したり変更したりしないでください。このファイルを不適切に変更すると、サーバー障害からの回復に影響が出ます。

ローカル・システムがサーバーで、クライアントがまったく再起動しないとする
と、ロックされたファイルに関する要求が無限に蓄積されてしまいます。このファ
イルはロック状態から絶対に解除されなくなります。この問題は、「統合ファイ
ル・システム・ロックの解除 (RLSIFSLCK)」コマンドを使用すれば避けることがで
きます。

RLSIFSLCK (統合ファイル・システム・ロックの解除) コマンド

目的

「統合ファイル・システム・ロックの解除 (RLSIFSLCK)」コマンドを使用すると、
ネットワーク・ファイル・システム (NFS) のすべてのバイト範囲ロックを解除する
ことができます。解除できるロックは、指定した NFS クライアントが保持してい
るロック、または指定した特定のオブジェクトに対するロックです。このコマンド
は、通常の方法では解除できなくなった資源を解除するためにだけ使用してくださ
い。このコマンドを使用してロックが解除されると、正常終了の戻りコードがロッ
クを設定したアプリケーションに送られます。

バイト範囲ロックの詳細および `fcntl()` API の完全な解説は iSeries Information
Center のトピック「API」を参照してください。

制約事項

1. このコマンドを使用するには、*IOSYSCFG 特殊権限が必要です。

RLSIFSLCK とその関連パラメーターおよびオプションの詳細については、iSeries
Information Center のトピック「CL」を参照してください。

RLSIFSLCK 画面

ファイル・システム・ロックの解除 (RLSIFSLCK)

選択項目を入力して、実行キーを押してください。

遠隔ロケーション TULAB2

オブジェクト

終り

F3= 終了 F4= プロンプト F5= 最新表示 F12= 取り消し
F13= この画面の使用法 F24= キーの続き

図 56. 「ファイル・システム・ロックの解除 (RLSIFSLCK)」画面の使用法

RLSIFSLCK コマンドを使用するときには、次のような複数のパラメーターを指定することができます。

- RLSIFSLCK コマンドの RMTLOCNAME パラメーターでは、ローカル・ファイルに対する NFS 関連のロックを解除したいリモート・システムのホスト名またはインターネット・アドレスを指定します。
- RLSIFSLCK コマンドの OBJ パラメーターでは、すべてのバイト範囲ロックを解除したいオブジェクトのパス名を指定します。これを指定すると、そのオブジェクトに対するロックが、ロックのタイプやそのロックを保持しているプロセスのタイプにはかかわらず、すべて解除されます。

例

例 1: リモート・クライアントのロックを解除する

```
RLSIFSLCK RMTLOCNAME('TULAB2')
```

このコマンドは、システム TULAB2 によってローカル・ファイルに対して保持されている NFS 関連ロックを解除します。

例 2: ローカル・オブジェクトのロックを解除する

```
RLSIFSLCK OBJ('/classes/class2/proj3')
```

このコマンドは、オブジェクト /classes/class2/proj3 に対して保持されているすべてのバイト範囲ロックを解除します。

第 8 章 統合ファイル・システムの API とネットワーク・ファイル・システム

エラー状態

統合ファイル・システム API (アプリケーション・プログラミング・インターフェース) を使用してネットワーク・ファイル・システムを処理しているときによく発生するエラー状態で、特別に考慮する必要のある状態が 2 つあります。それは、ESTALE および EACCES エラー状態です。

ESTALE エラー状態

このエラー番号が戻された場合は、サーバーがファイルまたはオブジェクト・ハンドルを拒否したことを意味しています。

ネットワーク・ファイル・システム経由でリモート・ファイル・システムにアクセスしている場合は、そのファイルがサーバー側で削除された可能性があります。あるいは、そのオブジェクトが入っているファイル・システムにアクセスできなくなったことも考えられます。

EACCES エラー状態

このエラー番号が戻された場合は、サーバーがオブジェクトに対する許可を拒否したことを意味しています。

あるオブジェクトにアクセスしようとしたプロセスまたはジョブが、そのオブジェクトのアクセス許可では禁止されている方法でアクセスしようとしていました。

プロセスまたはジョブが、指定したファイル、ディレクトリー、構成要素、またはパスに対するアクセス権限を持っていません。

NFS 経由でリモート・ファイルにアクセスしている場合、クライアントは、ローカル・データの更新があるまではクライアントでのファイル許可を反映しません。

「マウント済みファイル・システムの追加 (ADDMFS)」および「MOUNT」コマンドに、ローカル・データを最新のものに更新する時間間隔を決定するいくつかのオプションがあります。リモート・ファイルへのアクセスが失敗した別の原因として、ローカル・システムとリモート・システムで UID または GID のマッピングが違っていることも考えられます。

API に関する考慮事項

ファイルを作成または除去する API を使用する際には、NFS サーバーのスループットに関連するいくつかの考慮事項があります。サーバーがビジー状態の場合、要求を出したクライアントがタイムアウトになり、同じ要求をサーバーに複数回送信してしまうことがあります。この理由で、API を使用してファイルを作成または除

去する際に、クライアントは EEXIST または ENOENT エラー状態を受け取る可能性があります。これらのエラー状態は、操作がまったく正常に完了した場合でも発生することがあります。

ユーザー・データグラム・プロトコル (UDP) に関する考慮事項

ファイルを作成または除去するために、クライアントはサーバーに要求を送ります。サーバーがビジー状態の場合、元々の要求をサーバーが完了するまでの間に、クライアントが何回もタイムアウトを起こすことがあります。NFS プロトコルの基礎になっている UDP サポートは、このような場合の多重要求を正しく処理できないことがあります。

UDP は、クライアントに戻されるデータの送達 または順序 は保証しません。操作が正常に完了した場合、クライアントは次のいずれかの戻りコードを受け取ります。

1. 戻りコード=0 (RC=0)。操作は正しく完了しました。
2. EEXIST。操作は正しく完了しました。戻りコード 0 (RC=0) が失われたか間違った順序で受信されたため、このエラー状態が戻されました。
3. ENOENT。操作は正しく完了しました。戻りコード 0 (RC=0) が失われたか間違った順序で受信されたため、このエラー状態が戻されました。

クライアント・タイムアウトの解決

ファイルの作成や除去に関して予期しない戻りコードやエラー状態を受け取るという混乱を減らすために、ユーザーは以下の作業を行うことができます。

1. クライアント・タイムアウトの時間を長くして、サーバーに要求を出す際にクライアントがタイムアウトを起こす頻度を減らす。これにより、同じ要求がサーバーに繰り返し送信されることが少なくなります。
2. 以下の方法でサーバーのスループットを上げる。
 - クライアントの要求を処理する NFS サーバー・デーモン (NFSD) の数を増やす。
 - サーバーへの要求の負荷を減らす。
 - サーバーにアクセスして要求を出すクライアントの数を減らす。

ネットワーク・ファイル・システムでの相違

以下の API は、ネットワーク・ファイル・システムと一緒に使用する場合の使用上の注意が更新されています。

- open()
- create()
- mkdir()
- access()
- fstat()
- lseek()
- lstat()
- stat()
- read()
- write()
- fcntl()

一般に、これらの API を使用する際には、ネットワーク・ファイル・システム経由でリモート・ファイルをローカル・アクセスすると、予期しない結果になることがあるという点を覚えておいてください。サーバー側での条件によって、リモート・ファイルへのクライアント・アクセスの特性が決まります。ローカルに格納された許可やその他の属性がサーバー側の許可や属性より制限の厳しいものであると、ファイルまたはディレクトリーの作成が失敗することがあります。ローカル・データを最新データに更新したり、ファイル・システムのマウントを再実行したりした後は、ファイルの作成操作が成功することもあります。「マウント済みファイル・システムの追加 (ADDMFS)」コマンドのいくつかのオプションは、ローカル・データを最新のものに更新する時間間隔を決定します。

ファイルまたはディレクトリーをオープンした後で、ファイルまたはディレクトリーに対して操作を実行する要求が失敗することがあります。これは、各要求ごとに属性がサーバー側でチェックされるからです。オブジェクトに対する許可がサーバー側で制限の厳しいものになっていると、NFS が更新を受け取った時点で、オープンしたファイル記述子に対する操作が失敗します。サーバーがオブジェクトをリンク解除するか利用不能にすると、NFS が更新を受け取った時点で、オープンしたファイル記述子に対する操作が失敗します。ネットワーク・ファイル・システムは、ファイルまたはディレクトリーの属性を取得する操作にも影響を与えます。サーバーでの最新の変更をクライアント側でまだ利用できないために、ある操作に対してサーバーが古い値を戻すことがあります。

「統合ファイル・システム API」の使用上の注意については、iSeries Information Center 内のトピック「API」を参照してください。

open()、create()、および mkdir() API

最近削除されたファイルまたはディレクトリーを再作成しようとする、その要求は失敗することがあります。これは、NFS が記憶しているローカル・データに、そのファイルまたはディレクトリーの存在がまだ記録されている場合です。ローカル・データを更新した後なら、ファイルまたはディレクトリーの作成は成功します。

fcntl() API

ネットワーク・ファイル・システムを使用してファイルの読み書きを行う際のデータ保全性の確保は、バイト範囲ロックに依存しています。データの不整合を防止するには、「fcntl() API」を使用してロックを取得したり解除したりしてください。

fcntl() API について詳しくは、iSeries Information Center のトピック「API」を参照してください。

未変更の API

ここに挙げる API は、ファイル記述子、ディレクトリー・ポインター、またはパス名を取らず、作成を行う関数ではありません。したがって、NFS はこれらの API に関しては EACCES および ESTALE エラー番号を戻しません。その他の API については、EACCESS および ESTALE エラー番号を受け取ることがあります。

- getegid()
- geteuid()

- `getgid()`
- `getgrgid()`
- `getgrnam()`
- `getgroups()`
- `getpwnam()`
- `getpwuid()`
- `getuid()`
- `sysconf()`
- `umask()`

第 9 章 ネットワーク・ファイル・システムの機密保護に関する考慮事項

ネットワーク・ファイル・システム (NFS) を使用すると、すべてのユーザーが適切な情報にいつでもアクセスできる、シームレスで透過的な名前空間を作成することができます。とはいえ、NFS には、機密保護に関して特別に考慮すべき点があります。その考慮点は、主として、ユーザー ID、グループ ID、および補足的なユーザー ID に関係しています。この章では、これらの考慮点について、CHGNFSEXP コマンドの特定のパラメーターおよびオプションと関連付けながら説明します。

このセクションでは、NFS の機密保護に関する数々の問題について取り上げ、機密性の高い名前空間を維持すると同時に、機密保護上の問題点や抜け穴を回避するための最善の方法について説明します。

OS/400 の機密保護の詳細については、以下の資料を参照してください。

- iSeries Information Center のトピックにあるセキュリティー
- iSeries Information Center に載っているブック「iSeries 機密保護解説書」

信頼できる共同体

信頼できる共同体は、「承認された」NFS サーバーおよびクライアントだけで構成されるグループで、信頼できるユーザー・ネットワークを表します。このグループ内では、名前空間の機密性を外部の信頼できないユーザーから保護することを個々の人の責任で行うというシステムに基づいて、ファイル・システムをエクスポートしたりマウントしたりします。

信頼できる共同体を定義付けるもう一つの特徴は、クライアント / サーバーの関係で、どのような種類の特別な暗号化も行われれないということです。NFS クライアントとサーバー間の伝送はコード化されません。クライアント上で実行されるアプリケーションだけが、最低限の暗号化を行い、クライアントとサーバーとの間でデータを送信します。このような理由で、NFS サーバーからどのようにファイルをエクスポートするかには注意を払うことが重要になります。クライアントとサーバー間の伝送が暗号化されていない場合に、「世界中に」エクスポートを行うと、エクスポートされたファイル・システムは誰でもアクセスできることになります。安全なエクスポートに関して詳しくは、101 ページの『ファイル・システムを安全にエクスポートする』を参照してください。

エクスポート・オプションの詳細については、102 ページの『エクスポート・オプション』を参照してください。

ネットワーク・データの暗号化

エクスポートと信頼できる共同体

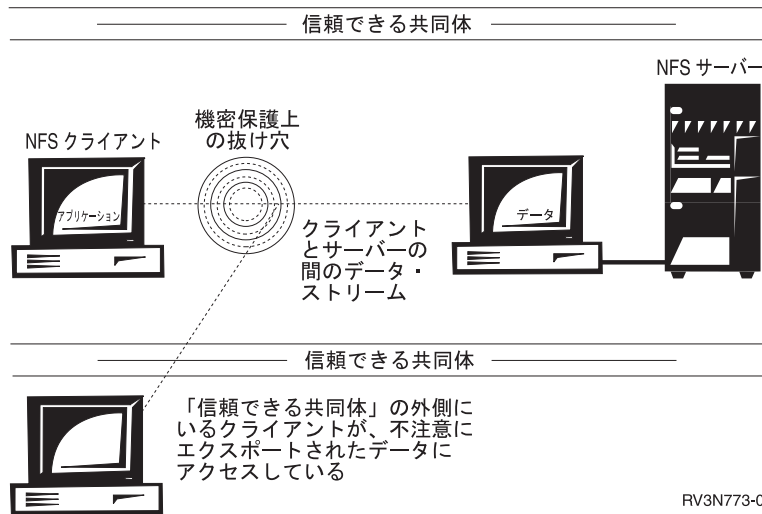


図 57. 信頼できる共同体の外側にいるクライアントが機密保護上の抜け穴を作る

信頼できる共同体の外側にいるクライアントが、その共同体の存在に気付くことがあります。さらに、悪意のあるクライアントが、クライアント / サーバー間で転送されているデータを読み取ったり改変したりできる「スパイ」プログラムを忍び込ませることもあります。そのプログラムは、データ・フローを傍受し、接触したデータを改変します。システムは、暗号化されないデータを、クライアント / サーバー間で平文でやり取りします。したがって、そのようなデータを「スパイ」しているシステムは、トークンリング、イーサネット、あるいは他のネットワーク・インターフェースと対話することにより、データにアクセスして読み取ることができます。

共同体に所属する個々のサーバーおよびクライアントを外部の侵入者から保護する方法はいろいろあります。また、クライアントからサーバーに送られたり、その逆の方向に戻されたりするデータを保護するための方法もいくつかあります。

1. 暗号化エンジンをクライアントとサーバーの両方に配置すれば、データをコード化することができます。暗号化エンジンは、「送り手」のクライアントまたはサーバーから発信された情報をコード化し、「受け手」のクライアントまたはサーバーに到着した情報をデコードします。
2. ユーザーの識別情報は、次のような方法を用いて認証することができます。
 - a. AUTH_UNIX。オブジェクトに対する許可は、ユーザー ID (UID) だけによって管理されます。暗号化は一切行われません。iSeries は、このタイプのオブジェクト許可を自動的に実行します。
 - b. AUTH_DES。これは、データ暗号化規格 (DES) です。このタイプの暗号化を使用すると、データを保護できます。
 - c. AUTH_KERBEROS。このタイプの暗号化は、信頼できるトークン管理プログラムに基づいて権限トークンを管理する第三者の管理者によってユーザーを保護します。Kerberos 機密保護では、信頼できる共同体を実現できます。Kerberos は、秘密鍵による許可を実施するために使用される認証プロトコル

です。Kerberos は、信頼できる共同体内でのユーザーおよび権限の完全で総合的な認証に依存しており、他の何者もデータにアクセスすることができません。

ユーザー権限

ユーザーが NFS クライアントおよびサーバーにログオンすると、各ユーザーのユーザー権限に応じて、そのユーザーが何を実行できて何を実行できないかが決まります。ユーザー権限は、管理者によって各ユーザーに割り当てられるもので、通常、特定のユーザーに対するユーザー ID (UID)、ユーザー・グループに割り当てられるグループ ID (GID)、およびあるユーザーが所属する各種のグループ ID を列挙した補足 GID がその実体です。

各ユーザーおよびシステムにユーザー権限を適切に割り当てることにより、機密性が高く、効率的な名前空間を構築できます。ユーザーは、いつでも必要な時に適切なデータをアクセスできるようになります。ユーザーおよびシステムへの権限の割り当てが不適切であると、名前空間の機密性が下がり、内部および外部に機密保護上の大きな抜け穴ができてしまいます。

ユーザー ID (UID)

UID は、ユーザー ID です。UID は、特定のシステムにおけるユーザーを表す 1 つの数値で構成されます。UID には、ユーザーの名前や、そのユーザーに関するその他の情報は一切、含まれていません。UID は、単にシステムがユーザーを認識するための手段であり、そのユーザーのユーザー・プロファイルを活動化して権限検査に利用できるようにします。各ユーザーは、いくつものシステムにアクセスできる場合があり、多数の異なった UID と、それに対応する各システムに対する権限を持っていることがあります。ユーザーは、本当に必要なアクセス権限だけを持ち、それ以上の権限は持たないようにするべきです。

たとえば、1 つの iSeries において、Bill は 136 という UID を持っています。別の iSeries では、Bill の UID は 142 かもしれませんし、さらに別のシステムでは 2700 という UID を持っていることもあります。これらの UID はいずれも、各ネットワーク・システムに対する数々の異なった権限を持っている可能性があります。Bill の場合、1 つのシステムでは *USER 権限を持ち、別のシステムでは *QSECOFR を持ち、さらに別のシステムでは別の権限を持っているかもしれません。このようなさまざまな UID を適切に管理しないと、NFS の機密性が危険にさらされます。

グループ ID (GID)

GID は、グループ ID です。これは、特定のシステムにおけるユーザーのアクセス権限を識別するという点では、UID と同様の働きをします。しかし、GID はさらに一歩進んで、ユーザーまたはマシンのグループに対する汎用権限を持つ働きをします。GID は UID にはまったく依存せず、いずれかの UID に対応していることもあります。各ユーザー・プロファイルには、GID が関連付けられている場合もあれば、関連付けられていない場合もあります。

したがって、GID は、会社における部や課、あるいはコンピューター室における 1 つのワークステーション・グループの機能を果たします。GID は、特定の UID が

ファイルまたはオブジェクトに対するアクセス権限を持たない場合、そのファイルまたはオブジェクトにアクセスするための第 2 の方法になります。GID は、それが関連付けられたユーザー・プロファイルから何かの権限を取り去ることはありません。GID は、何らかの権限を付け加える ことができるだけです。クライアントの要求と一致する GID がサーバー上に存在すると、それに対応する権限がその要求に付け加えられます。GID がサーバー上に存在しないと、その GID は無視されます。

あるオブジェクトに対する第 3、第 4、または第 n 番目のアクセス手段になり得る、補足 GID というものもあります。GID は、そのユーザーがどのグループに属しているかに基づいてユーザーに割り当てられます。UNIX サーバーおよびクライアントは、補足 GID を使用してユーザーの権限を判別することができます。iSeries 上の NFS は、補足 GID をサポートします。クライアント・システムのユーザーが複数のグループのメンバーである場合、システムは、それらのグループの GID をサーバーに送信します。サーバーは、補足 GID リストとして既存のグループ・プロファイルを持つ GID をそのサーバー上にマップされた NFS ユーザーに対して使用します。GID および補足 GID はユーザーに権限を付け加えるものなので、GID は、プロファイルによれば通常は禁止されているはずのオブジェクトにユーザーがアクセスできるようにすることがあります。どのグループに属するどのユーザーが自分のデータにアクセスできるかについては、常に注意を払っておくことが大切です。GID は、強力なグループからのユーザーが機密データに無許可アクセスするのを支援することがあります。

1 人のユーザーがさまざまな ID を持ち、多種の権限を付加されると、NFS の機密性が危険にさらされます。ファイル・システムをクライアントに利用させるために CHGNFSEXP コマンドのオプションを設定する際には、この危険が特に重大になります。安全なエクスポートに関して詳しくは、101 ページの『ファイル・システムを安全にエクスポートする』を参照してください。

ユーザー ID のマッピング

ユーザーがサーバーに正しくログオンすると、サーバー上にあるユーザー・プロファイルに基づいてすぐに権限が自動的に付与されます。

ユーザーがローカル・クライアント・システムからリモート・サーバー・ファイルにアクセスすると、その要求はリモート・サーバーに送られます。サーバーは、ユーザーの UID と権限を各要求ごとに検査します (NFS は状態なしのプロトコルなので)。ユーザーがリモート・サーバーにアクセスするとき、要求には UID だけが添付され、ユーザー・プロファイル名 (およびパスワード) は添付されません。次いで、サーバーは、その UID を対応する権限にマッピングしますが、その際、その UID が実際に持っているユーザー・プロファイル名は問題にしません。そのため、別々のシステムに属する 2 つの UID が互いに一致してしまい、しかもそれぞれの UID は別々のユーザーが所有しているという事態が発生することがあります。UID の不適切なマッピングについての詳細は、97 ページの『UID のマッピング例』を参照してください。

ユーザー ID のマッピングで起こり得るシナリオ

分散ネットワークにおける UID のマッピングでは、以下のような状況が起こり得ます。

1. ユーザーの UID がクライアントとサーバーで同じユーザー・プロファイルにマッピングされる。この場合は、何も矛盾はありません。
2. ユーザーの UID がクライアントとサーバーの両方に存在するが、別々のプロファイルにマッピングされる。この場合は、ユーザーが必要とする権限に対して過不足のあるプロファイルがマッピングされる可能性があり、機密保護上の矛盾が発生する恐れがあります。
3. ユーザーの UID がクライアントに存在するが、サーバーには存在しない。この場合は、エクスポート項目の ANON パラメーターが検査されます。無名ユーザーに対応するプロファイルが見つければ、このユーザーの UID はそのプロファイルにマッピングされます。サーバーが無名ユーザーのマッピングを許可していなければ、この要求を出したユーザーは EACCES エラー状態を受け取ります。ANON パラメーターの詳細と、無名ユーザーがエクスポートをアクセスできるようにする方法については、102 ページの『無名ユーザー』を参照してください。

ネットワーク経由での UID マッピングの違いのために、ユーザーがリモート・システム上のファイルを操作する際に問題が起きることがあります。この問題は、ユーザーがリモート・システムでローカル・システムと同じ権限を持っていないときに起こります。

ユーザー ID の管理

NFS 名前空間の管理者は、以下のことを行わなければなりません。

1. 可能な限り、ユーザーには一致した権限をセットアップする。これにより、マウント・ポイントを横切ってローカル・システムとリモート・システムの両方を操作する場合に、ユーザーが混乱しないようにできます。これには、UID および GID をネットワーク全体で適切にマッピングすることが含まれます。
2. システムとユーザーの両方に合わせて調整した個々の適切な権限を作成する。これらの権限は一致している必要はありませんが、該当するユーザーとシステムの両方に適している必要があります。1 つのシステムでユーザーに *SECOFR 権限が必要だからといって、同じ権限がすべてのマシンで必要であるとは限りません。

UID のマッピング例

TULAB において、工学系の学部卒業生 Bill が、TULAB1 において 136 という UID、iSeries TULAB2 において 142 という UID を持っているとします。Bill が、TULAB1 にあるファイルをマウントしたり他の操作をしたりしようすると、136 という UID がサーバーに送信されます。TULAB1 は、136 としての彼の権限を検査してから、彼が要求したオブジェクトへのアクセスを許可します。もし Bill が TULAB2 の管理者であったとしても、彼の TULAB1 での UID がマップされたプロファイルに適切な権限が含まれていなければ、彼の要求したオブジェクトにアクセスできない こともあります。

したがって、すべてのユーザーが、適切にマッピングされた UID を持っているかどうかを確認することが大切です。この例では、アクセスする権限が自分にあると思っているデータにアクセスできなかったというケースですが、UID のマッピングが不適切であると、さらに悪い影響が出る場合もあります。

たとえば、工学系の学部新入生 Jennifer が、Bill と正反対の UID、つまり TULAB1 において 142、TULAB2 において 136 を持っているとします。Jennifer が TULAB1 から TULAB2 に要求を出すと、彼女の UID とそれに対応する権限がサーバーに送信されます。彼女は Bill と同じ UID を持っているため、iSeries TULAB2 は、彼女が Bill である と見なします。それで、TULAB2 は、いつも Bill がアクセスしているすべてのオブジェクトやアクションを、彼女がアクセスすることを許可します。

このケースでは、一方のサーバーでは権限をほとんど、あるいはまったく持っていないユーザーが、UID のマッピングが不適切であるというだけの理由で、もう一方のサーバーで意図せずに混乱を発生させてしまいます。UID とそれに対応する権限を割り当てる際には、名前空間の全体像に注意を払うことが最も重要です。本当に必要なアクセス権限だけをユーザーに割り当てるようにし、UID が決して重なり合わないよう確認します。そうすれば、一層機密性の高い名前空間を構築できます。

たとえば、Chris Admin は、この状況を正すために、Bill が両方のシステムで 136 という UID を持つように変更できます。また、Jennifer が両方のマシンで 142 という UID を持つようにします。さらに、もし Jennifer が一方のシステムにアクセスする必要がなく、そのシステムにユーザー・プロファイルが不要なのであれば、Chris Admin はそのシステムから 142 という UID を消去することができます。

これまで説明した例では、ユーザーが意図せずに別のユーザーの UID を誤って使用しました。しかし、時には、ユーザーが意図的に別のユーザーになりすまし、その UID を悪用するかもしれません。これまでは手の届かなかったファイル・システムやオブジェクトに対して自分には不相応な権限と許可を獲得しようとして、あるいはいたずらをしようとして、そのようなことを試みるユーザーがいるかもしれません。いずれにせよ、賢明なシステム管理者は、このような種類の事態を阻止できます。

たとえば、TULAB1 と TULAB2 に、Ray という名前のユーザーがいるとします。Ray は、TULAB1 で 2700 という UID を持ち、そのシステムでの *SECADM 権限を持っています。彼は、TULAB2 で 150 という UID も持っており、そのシステムでは一般ユーザーです。

Joe という名前のもう 1 人のユーザーが TULAB2 への一般ユーザーとしてのアクセス権を持っており、彼の UID は 170 です。問題が起きるのは、Ray が Joe のユーザー・プロファイルにアクセスして、Joe の UID を見付け出した場合です。Ray は、TULAB1 に FOOL=170 という UID をセットアップして、TULAB2 に要求を出すことができます。サーバーが Ray の FOOL という UID を検査すると、FOOL を Joe と同じであると見なします。すると、TULAB1 は、Joe のホーム・ディレクトリや Joe の持っているすべてのオブジェクトに Ray がアクセスするのを許可します。それで、Ray は、Joe のホーム・ディレクトリを完全に削除したり、システムに対する Joe のアクセス権限を除去したりなど、Joe が実行できるあらゆる操作を実行する権限を持つこととなります。

その他にも、ユーザーのシステムやユーザーが所有しているオブジェクトに侵入するための方法がいくつかあります。クライアントの管理者は、意図的にリモート・サーバーの UID を自分のものと偽って使用することができます。

たとえば、クライアントの管理者は、クライアントにアクセスしている TULAB2 のユーザー Mary の UID にログオンしてアクセスすることができます。Mary の UID が 123 であれば、クライアントの管理者はこの UID を偽って使用し、事実上、両方のシステムで Mary になります。ことができます。

TULAB2 に、/home/mary/index.html という名前のファイルがあります。ユーザーがこのファイルに何かの操作を実行するには、まずそのファイル・ハンドルを取得しなければなりません。適切な権限を持つユーザーは、親ディレクトリーに要求を出すことにより、このハンドルを取得できます。/home/mary/index.html の親ディレクトリーは /home/mary です。ファイル・ハンドルとは、そのオブジェクトについて記述した固有 ID で、いつまでも変更されません。ファイル・ハンドルは、IPL が何度実行されても、クラッシュが起きても、ずっと不変です。

クライアントの管理者が 123 という UID の Mary になります。できたとすると、その管理者はファイル・ハンドルを取得することができます。そうすると、その管理者は、Mary のファイルを読み取ったり、書き込んだり、変更したり、削除したりできるようになるわけです。

名前空間の管理者にとって、このようなケースに対処する解決策は、名前空間全体で適切な UID のマッピングの規則に従うことです。管理者は、さらに、ファイル・システムをどのようにエクスポートするかについても注意を払う必要があります。信頼できる共同体は、アクセスを要求してくる外部のクライアントや無名ユーザーに対して門戸を開け放つべきではありません。

適切な UID マッピング

管理者は、ユーザーが適切なサーバーにある適切な情報にアクセスできる、機密性の高い名前空間を提供するために、UID を正しくマッピングする必要があります。そのためには、一度に 1 ユーザーおよび 1 サーバーずつ実行する、ステップ単位のプロセスがあります。

たとえば、Joan という名前のユーザーが、TULAB1 において 27 という UID を持ち、TULAB2 においては 14 という UID を持っているとします。Chris Admin は、この 2 つの UID を 2 つのシステムで同一にすることにより、どちらのシステムでも彼が自分でセットアップした一致したプロファイルが提供されるようにすべきだと考えます。もし Joan が統合ファイル・システム内に何かのオブジェクトを所有していると、Chris Admin は TULAB2 における Joan の UID を変更することができません。そこで、Chris Admin は、次のような一連の操作を行います。

1. Chris Admin は、Joan の汎用ネットワーク・ファイル・システム UID として 1 つの UID 番号を選びます。この番号が、ネットワークの名前空間全体での Joan の UID になります。ここでは、Chris は JOAN=27 と決めます。
2. 彼は、TULAB2 に一時的な UID として TEMP=27 を作成します。この UID がまだ使用されておらず、正しく処理されれば、Chris Admin は次の操作に進みません。
3. 彼は、「ユーザー・プロファイルの変更 (CHGUSRPRF)」コマンドを使用して、TEMP の UID を任意の未使用の番号 (12345) に変更します。

- 次に、「ユーザー・プロファイルの削除 (DLTUSRPRF)」コマンドを使用して、Joan のディレクトリーとオブジェクトをすべて TULAB2 から削除します。ただし、このコマンドを完了する前に、これらすべてのオブジェクトの所有権を TEMP に移します。
- 次に、「ユーザー・プロファイルの作成 (CRTUSRPRF)」コマンドを使用して、TULAB2 に JOAN=27 を作成します。
- Chris Admin は、もう一度 DLTUSRPRF コマンドを使用して TEMP のユーザー・プロファイルを削除し、すべてのオブジェクトの所有権を Joan に戻します。このとき、Joan の汎用 NFS UID は 27 になっています。

各システムの各ユーザーごとに UID を変更していく方法の代わりに、管理者は、QSYCHGID API を使用することができます。この新しい API は、iSeries コマンド行、C プログラム、COBOL プログラム、およびその他のインターフェースから呼び出すことができます。この関数は、システム提供プロファイルと通常のユーザー・プロファイルの両方について、UID と GID を変更できます。このコマンドを使用するには、管理者は以下の項目を指定する必要があります。

- 10 文字のプロファイル名。これは、変更するユーザー・プロファイルまたはシステム・プロファイルの名前です。たとえば、BILL または QUSER などのプロファイル名を指定します。
- UID の値。これは、指定したユーザー・プロファイルまたはシステム・プロファイルの新しい UID 値です。'-1' という値を指定すると、UID は変更されません。
- GID の値。これは、指定したユーザー・プロファイルまたはシステム・プロファイルの新しい GID 値です。'-1' という値を指定すると、GID は変更されません。'0' という値を指定すると、GID がユーザー・プロファイルまたはシステム・プロファイルから除去されます。
- エラー・コード構造体へのポインター。このポインターは、コマンド行からは指定できませんが、プログラミング・インターフェースでは機能します。このパラメーターに '0' という値を指定すると、コマンドが正しく機能できなかったときにユーザーは例外を受け取ります。

QSYCHGID を実行するには、まず最初に、該当するシステムまたはユーザー・プロファイル名で現在実行されているすべてのジョブを終了させなければなりません。これがシステム提供のプロファイルである場合、iSeries は、「制限状態」に切り替える必要があるかもしれませんが、つまり、すべての対話式、バッチ、デーモン、および要求ジョブを終了させます。「制限状態」にするためには、「ENDSBS」コマンドを発行します。引き続き稼働させておくべきなのは、iSeries に直接接続されたシステム・コンソールです。

QSYCHGID は、ユーザーまたはシステム・プロファイルの UID 値を適切な値に更新し、そのユーザーまたはシステムが所有しているオブジェクトの UID 値も更新します。ユーザーまたはシステムが所有しているオブジェクトの更新に失敗した場合は、「記憶域の再利用 (RCLSTG)」コマンドを使用して、失われた可能性のあるオブジェクトを再利用してください。

注: 既存のユーザーまたはシステム・プロファイルにすでに割り振られている UID 番号を選ぶと、QSYCHGID は必ず失敗します。1 つの UID を複数のプロファイルに所属させることはできません。

注: UID と GID は互いに排他的ではありません。つまり、500 という UID 値と、500 という GID 値は、同じプロファイルを意味するわけではありません。

ファイル・システムを安全にエクスポートする

名前空間の管理者がファイル・システムをエクスポートするときに考慮すべき点は、以下のとおりです。

1. エクスポートしたファイル・システムをマウントしたりその他の操作を実行したりするためのアクセス権限を持つクライアントを指定する、クライアントの完全なリストを使用します。こうすれば、CHGNSFSEXP コマンドの OPTIONS パラメーターの ACCESS オプションにリストされていないクライアントが機密データを手に入れるのを防止できます。
2. エクスポートしたファイル・システムを無名ユーザーがアクセスできるようにする際は、細心の注意を払って QNFSANON を使用します。ROOT パラメーターと一緒に指定すると、未知の (無名) ユーザーが名前空間にアクセスできるようになってしまいます。
3. /etc/exports ファイルへのアクセスはいつも除外します。このファイルはエクスポートしないでください。ユーザーがこのファイルにアクセスする手段を発見して、安全に守られていたはずのデータを他のユーザーが利用できるように改変してしまう恐れがあります。

管理者がファイル・システムをエクスポートする際には、以下の事柄は決して行わないでください。

1. 「ルート」(/) ディレクトリーは絶対にエクスポートしてはなりません。ファイル・システムをエクスポートすると、そのパスより「下流」にあるすべてのディレクトリーおよびオブジェクトもエクスポートすることになるという点を忘れないでください。もし「ルート」(/) ディレクトリーをエクスポートすると、「ルート」(/) の下流にあるすべてのディレクトリーおよびオブジェクトも一緒にエクスポートされます。クライアントが必要とするものだけをエクスポートしてください。
2. ファイル・システムを「世界中」に向けてエクスポートし、不特定多数のクライアントが情報をアクセスできるようにすることは絶対にすべきではありません。そうではなく、エクスポートするファイル・システムそれぞれを、各クライアントの必要に合わせて調整するべきです。サーバーを「世界中」からのアクセスに対して開放と、未知のユーザーがファイルを自由にマウントして改変できるようになります。特定のクライアントに対してのみエクスポートするようにしてください。

エクスポートを制限することについての詳細は、102 ページの『「世界」に向けてエクスポートする』を参照してください。

3. どのようなファイル・システムの場合でも、QNFSANON 汎用アクセスを許可する際には、細心の注意が必要です。このようなアクセスを許可すると、未知 (無名) のユーザーがファイル・システムをマウントして改変できるようになります。そのファイル・システムを必要とするクライアントに対してだけエクスポートするようにしてください。

無名ユーザーによるアクセスの制限についての詳細は、102 ページの『無名ユーザー』を参照してください。

エクスポート・オプション

ネットワークの名前空間に機密性を保てない状態で ファイルをエクスポートしてしまう主な状況には、次の 2 つがあります。

1. エクスポートしたファイル・システムに対するアクセスを無名ユーザーに対して許可することにより、未知のユーザーがデータを読み取ったり変更したりできるようにしてしまう。
2. 「世界中」に向けてエクスポートしてしまう、つまりエクスポートしたデータを信頼できる共同体の内部および外部の両方のユーザーがアクセスできるようにしてしまう。

信頼できる共同体に属さない未知のユーザーまたはシステムにファイル・システムを配布する必要はありません。NFS サーバーの管理者は、いつも、適正なデータだけを適正なクライアントに向けてのみエクスポートするようにすべきです。

無名ユーザー

エクスポートしたファイル・システムへのアクセスを無名ユーザーが獲得してしまうような事態を避けるには、「CHGNFSEXP」コマンドの ANON オプションについて完全に理解しておくことが大切です。管理者は、無名ユーザー、つまり未知の UID を QNFSANON UID にマッピングするように指定することにより、iSeries に対する一定レベルのアクセス権限をそのようなユーザーに付与することができます。

このような処置が取られないようにするには、ANON=-1 を指定して、無名ユーザーがシステムの QNFSANON にマッピングされるのを禁止することができます。省略時のオプション値は QNFSANON であり、無名ユーザーに *USER 権限が付与されるようになっていることに注意してください。この場合、無名ユーザーはファイルを読み取ることができますが、ファイルに書き込むことはできません。管理者は、異なった省略時の権限を持つ UID をこのオプションに指定することもできます。

ユーザーが QNFSANON にマッピングされた場合、そのユーザーが作成したオブジェクトは、そのユーザーのユーザー・プロファイルではなく、QNFSANON プロファイルに所属します。たとえば、Cayce という名前のユーザーが 123 という UID を持っていて、サーバー・ファイル・システムをクライアントにマウントしたとします。その後、Cayce はファイルを作成します。このファイルの所有者は、Cayce ではなく、QNFSANON です。許可を表示してみれば分かるように、読み取り、書き込み、および実行権限は、所有者のプロファイル、つまり QNFSANON だけに属しています。Cayce は、いま自分が作成したファイルに対してどんなアクセス権限も持ちません。彼は所有者ではないからです。

この問題の解決策は、次のいずれかの手順に従うことです。

- そのファイルを、「世界中」に許可を公開して作成する。
- そのファイルを QNFSANON にマッピングしたままで、そのファイルの許可を「世界中」に向けたものに変更する。

「世界」に向けてエクスポートする

管理者は、エクスポートしたデータをすべての人がアクセスできるようにするのではなく、クライアントを選択する技法を採用することができます。そのためには、CHGNFSEXP コマンドの ACCESS オプションを使用します。このオプションは、UID、GID、および補足 GID を受け入れることもできます。ACCESS オプションを

使用すると、エクスポートしたファイルへのアクセスを、指定したクライアントだけに制限できます。サーバーは、指定された以外のすべてのクライアントからのアクセス要求を無視します。/etc/netgroup ファイルを使用すると、クライアントのグループに対応する名前を作成できます。これらのネットグループ名は、CHGNFSEXP コマンドの ACCESS オプションで使用することができます。エクスポートにおけるネットグループの使用について詳しくは、33 ページの『ファイル・システムをエクスポートする方法』 および 111 ページの『/etc/netgroup ファイル』を参照してください。

NFS エクスポートへのアクセスを制限する唯一の方法は、エクスポート・コマンドの ACCESS= オプションを使用する方法です。RW= および ROOT= オプションに値を追加したり変更したりした後でも、エクスポート・コマンドは、「世界中」に向けた読み取り専用アクセスを提供します。RW= オプションは、指定したクライアントに対して、マウントしたオブジェクトへの読み取り / 書き込みアクセスを許可します。ROOT= オプションを使用すると、指定しないホストに対して、ルート・アクセスを禁止できます。ACCESS= オプションを使用すると、指定したクライアントだけがエクスポートにアクセスできるようにできます。

CHGNFSEXP コマンドにオプションを何も指定しない場合、省略時の処置は、「世界中」に向けてファイル・システムをエクスポートすることです。つまり、すべてのクライアントがデータをアクセスできます。これは、機密保護上の大きな抜け穴になるおそれがあります。NFS 名前空間の信頼できる共同体の外部に、機密データかもしれない情報が引き渡される可能性があるからです。管理者は、そのデータを必要とするクライアントにだけデータをエクスポートするべきです。

root ユーザーのマッピング

Lori は、リモート TULAB2 において、165 という UID と *ALLOBJ 権限を持っています。さらに、ローカル XHOST1 において、165 という UID と UNIX クライアントを持っています。もし Chris Admin がファイル・システムを正しくエクスポートしているのであれば、Lori が *ALLOBJ 権限を持っているそのファイル・システムに対してアクセス要求を出すと、そのアクセス要求は失敗します。ただし、Lori が ROOT= クライアントのリストに含まれていれば、その要求は受け入れられます。

UNIX サーバーと iSeries サーバーでは、UID のマッピングが異なります。UNIX サーバーは、UID が 0 のユーザーを root ユーザーとして扱います。要求元のクライアントが、エクスポートのオプションで ROOT= リストに指定されていないと、そのユーザーは無名ユーザーにマッピングされます。iSeries では、root ユーザーはもっと広い意味で定義されています。受信した UID を、サーバー上の *ALLOBJ 権限を持ついずれかのユーザー・プロファイルにマップすると、そのユーザーは、無名ユーザー・プロファイルにマップされます。ただし、これは、要求が、ROOT= リストに指定されていないクライアントから出された場合のみです。この例では、Lori は UNIX クライアントの root ユーザーではありませんが、TULAB2 では依然として無名プロファイルにマップされています。これは、UID が 165 のプロファイルは *ALLOBJ 特殊権限を持つからです。それで、iSeries は彼女を root ユーザーと見なすのです。このように扱うことによって、この特殊権限がエクスポート要求で明確に指定されていない限り、リモート・ユーザーがこの権限を持っていると見なすことはできないようになります。これによって確実に、エクスポート・ファイル・システムの機密保護を堅固なものにします。

付録 A. よく使用されるコマンドの要約

表 I. ネットワーク・ファイル・システムのアプリケーションで使用するコマンド

コマンド	説明
ADDMFS	マウント済みファイル・システムの追加。リモート・サーバーからエクスポートされたファイル・システムをローカル・クライアントのディレクトリーに置きます。
CHGNFSEXP	ネットワーク・ファイル・システムのエクスポートの変更。NFS クライアントにエクスポートされているファイル・システムのエクスポート・テーブルにディレクトリー・ツリーを追加したり、そのテーブルからディレクトリー・ツリーを除去したりします。
CRTUDFS	UDFS の作成。ユーザー定義ファイル・システムを作成します。
DLTUDFS	UDFS の削除。ユーザー定義ファイル・システムを削除します。
DSPMFSINF	マウント済みファイル・システム情報の表示。マウントされたファイル・システムの情報を表示します。
DSPUDFS	UDFS の表示。ユーザー定義ファイル・システムを表示します。
EXPORTFS	ネットワーク・ファイル・システムのエクスポートの変更。NFS クライアントにエクスポートされているファイル・システムのエクスポート・テーブルにディレクトリー・ツリーを追加したり、そのテーブルからディレクトリー・ツリーを除去したりします。
ENDNFSSVR	ネットワーク・ファイル・システム・サーバーの終了。サーバーおよびクライアント上の NFS デーモンの 1 つまたはすべてを終了させます。
MOUNT	マウント済みファイル・システムの追加。リモート・サーバーからエクスポートされたファイル・システムをローカル・クライアントのディレクトリーに置きます。
RMVMFS	マウント済みファイル・システムの除去。リモート・サーバーからエクスポートされたファイル・システムをローカル・クライアント名前空間から除去します。
RLSIFSLCK	統合ファイル・システム・ロックの解除。あるクライアントが保持している、またはあるオブジェクトに対して設定されている NFS バイト範囲ロックを解除します。
STATFS	マウント済みファイル・システム情報の表示。マウントされたファイル・システムの情報を表示します。このコマンドは DSPMFSINF コマンドの別名です。

表1. ネットワーク・ファイル・システムのアプリケーションで使用するコマンド (続き)

コマンド	説明
STRNFSSVR	ネットワーク・ファイル・システム・サーバーの開始。サーバーおよびクライアント上の NFS デーモンの 1 つまたはすべてを開始します。
UNMOUNT	マウント済みファイル・システムの除去。リモート・サーバーからエクスポートされたファイル・システムをローカル・クライアント名前空間から除去します。

付録 B. /etc ファイルについて理解する

統合ファイル・システムの名前空間内に、/etc という名前のディレクトリーが存在します。このディレクトリーに入っている重要なシステム・ファイルには、経験豊富な NFS 管理者以外は、絶対に書き込みを行ったり、変更を加えたりしないでください。NFS は、これらのファイルを使用して、特定のシステム機能を実行します。/etc ディレクトリーには以下のファイルが入っています。

- /etc/exports
- /etc/netgroup
- /etc/rpcstab
- /etc/statd

エクスポート・コマンドを使用すると、/etc/exports および /etc/netgroup ファイルが検索されます。この 2 つのファイルが存在しないと、システムは、その省略時 CCSID である ASCII 文字を使用して、これらのファイルを作成します。システムがこれらのファイルを作成したときに、初歩的なヘルプ情報がファイルに含まれます。これらのファイルを更新する場合は必ず、エクスポート・コマンド ('-A' オプションを指定) を使用して、更新後のエクスポートを現行用にしてください。

/etc ディレクトリー内のファイルを編集する

/etc ディレクトリー内に保管されるファイルはストリーム・ファイルです。これらのファイルは、以下の方法で編集できます。

1. 「ファイルの編集 (EDTF)」コマンドの使用
2. PC の基本エディターの使用
3. NFS を介しての UNIX エディターの使用

ファイルの編集 (EDTF) コマンドを使用してストリーム・ファイルを編集する

「編集ファイル (EDTF)」コマンドを使用すると、ストリーム・ファイルまたはデータベース・ファイルを編集できます。省略時解釈では、このコマンドは基本オペレーティング・システムに含まれません。その場合は、プログラム一時修正 (PTF) でコマンドが提供されます。V3R7 用の PTF は SF38832 です。この PTF に入っているコマンドのインストール手順については、PTF のカバー・レターをお読みください。PTF のインストールが終了したら、絶対パス名または相対パス名を指定して、必要なファイルを編集できます。

絶対パス名と相対パス名の違いは、パス名の先頭に "/" が付いているかどうかということです。先頭文字が "/" である場合は、それが絶対パス名であることを表しています。絶対パス名は常に、ルート・ディレクトリーから開始します。相対パス名は、現行ディレクトリーから開始します。

絶対パス名

```
EDTF STMF('/etc/exports')
```

このコマンドは、ディレクトリー /etc 内のストリーム・ファイル exports を編集します。

```
EDTF FILE(PGMLIB/ACCOUNT) MBR(ACC1)
```

このコマンドは、物理ファイル PGMLIB/ACCOUNT 内に格納されているファイル・メンバー ACC1 を編集します。

相対パス名

```
EDTF STMF('exports')
```

このコマンドは、現行ディレクトリー内のファイル exports を編集します。

EDTF ユーティリティーで、F1 (ヘルプ) ファンクション・キーを押すと、適切なすべての行コマンドが検索されます。

注：WRKLNK コマンドのオプション 2 を使用することによってもファイルの編集を行うことができます。

PC 基本エディターを使用してストリーム・ファイルを編集する

ネットグループ・ファイルを編集する 2 番目の方法は PC の基本エディターを使用することです。iSeries 上の /etc ディレクトリーにアクセスするには、iSeries ナビゲーターまたはファイル転送プロトコル (FTP) を使用します。iSeries ナビゲーターを使用する場合は、/etc ディレクトリーにアクセスできなければなりません。

FTP を使用してストリーム・ファイルを編集するには、以下の手順に従ってください。

1. FTP を利用して、ファイルをご使用の PC に転送します。
2. 選択したエディターを使用して、必要な変更を行います。
3. FTP を使用して、/etc ディレクトリーにファイルを転送して戻します。

NFS から UNIX エディターを使用してストリーム・ファイルを編集する

iSeries サーバー上で、ストリーム・ファイルを編集する 3 番目の方法は、NFS を介して UNIX エディターを使用することです。このためには、次のステップに従ってください。

1. EXPORTFS コマンドを使用して、NFS サーバー上に、ディレクトリー /etc をエクスポートします。
2. MOUNT コマンドを使用して、ご使用の UNIX システムにディレクトリー /etc/ をマウントします。
3. 「ディレクトリーの変更 (CD)」コマンドを使用して、ディレクトリー /etc を現行ディレクトリーにします。
4. VI エディターまたはユーザーが選択したエディターを使用して、目的のファイルを編集します。

にエクスポートされマウントされたファイルのパス名コンポーネントに使用されます。HostName パラメーターで指定しなかったクライアントまたはネットグループについては、省略時のネットワークパス名 CCSID (ASCII) が使用されます。

DataFileCCSID

データ・ファイル CCSID は、指定された NFS クライアントまたはネットグループにエクスポートされマウントされたファイルのデータに使用されます。HostName パラメーターで指定しなかったホストについては、省略時のネットワーク・データ CCSID (バイナリー、変換なし) が使用されます。

NoWaitForWrites

NoWaitForWrites オプションは、NFS クライアントまたはネットグループ名に関して、書き込み要求を同期的に処理するか非同期的に処理するかを指定します。このオプションを指定しない場合は、データはディスクに即時に (同期的に) 書き込まれます。このフラグを使用しない場合は、データが即時にディスクに書き込まれる保証はありませんが、パフォーマンスを向上させるために利用できます (非同期)。

HOSTOPT (ホスト・オプション) パラメーターの形式に関するその他の規則

1. オプションは #HOSTOPT で開始しなければなりません。ホスト名は複数指定できますが、#HOSTOPT は 1 つだけしか指定できません。
2. 複数のホスト名およびそれらのオプションを指定する場合は、それらをコロンで区切ります。
3. オプションでは、大文字小文字が区別されます。
4. 指定されないオプションは、109 ページの『HOSTOPT (ホスト・オプション) パラメーターの形式』で説明されている省略時値として処理されます。

HOSTOPT パラメーターによる /etc/exports の形式設定例

例 1: ホストにエクスポートし、すべてのオプションを指定する

```
| /home/joe access=sammy          ¥  
| #HOSTOPT HostName=sammy,        ¥  
|           PathNameCCSID=367,    ¥  
|           DataFileCCSID=850,    ¥  
|           NoWaitForWrites
```

/home/joe をホスト sammy にエクスポートし、パス名とデータ・ファイルの両方について CCSID を指定します。書き込まれたデータが即時にディスクに書き込まれる (非同期) 保証はありません。長い項目を読みやすくするために、継続文字 '¥' を使用して、複数行に分割していることに注目してください。

例 2: オプションを指定して、複数のクライアントおよびネットグループにエクスポートする

```
| /home/selfe access=sammy:rainfall:dept-0A5 ¥  
| #HOSTOPT HostName=sammy,          ¥  
|           PathNameCCSID=367:      ¥  
|           hostname=rainfall,      ¥  
|           NoWaitForWrites
```

/home/selfe をホスト sammy、rainfall、およびネットグループ dept-0A5 にエクスポートします。ホスト sammy にはパス名の CCSID が指定されています。ホスト rainfall については、データが即時に (非同期) ディスクに書き込まれる保証はありません。

例 3: オプションを指定して、ディレクトリーを QSYS.LIB にエクスポートする

```
| /qsys.lib/jon.lib access=dept-0A5 ¥  
| #HOSTOPT HostName=wheatley, ¥  
| DataFileCCSID=367
```

```
| /qsys.lib/jon.lib をネットグループ dept-0A5 にエクスポートします。データ・  
| ファイルの CCSID を指定します。それにより、データは CCSID に変換されてか  
| ら、要求元のクライアントに送信されます。
```

/etc/netgroup ファイル

/etc/netgroup ファイルにはネットグループの定義が入っています。ネットグループとは、複数のシステムのリストを 1 つの名前に結び付けるメカニズムです。たとえば、次のように指定します。

```
department-0A5 (larrys-box,,),(safe-house,,)  
my-hosts (rainfall,,),(sammy,,),(department-0A5,,)
```

上の例で定義されるネットグループ my-hosts は、rainfall、sammy、larrys-box、および safe-house のリストを含みます。larrys-box と safe-house が含まれているのは department-0A5 自体がネットグループだからです。ネットグループは、他のネットグループ内にネストさせることができます。

慣例としてネットグループは次のように定義されます。

ネットグループ名

(ホスト名、ユーザー名、ドメイン名)

ホスト名

/etc/hosts ファイル内の任意のホストの名前。iSeries は、/etc/hosts ファイルを別個の統合ファイル・システムとしてはサポートしません。
/etc/hosts は、iSeries サーバーの TCP/IP サポートに組み込まれます。

ユーザー名

/etc/passwd ファイル内の任意のユーザーの名前。iSeries は、/etc/passwd ファイルの概念をサポートしません。

ドメイン名

ドメインの名前。

iSeries では、形式に関する次の規則がネットグループに適用されます。

1. すべての項目は行末で終了します。
2. 1 つの項目を複数行に拡張するために、行の最後の文字として継続文字 '¥' を使用できます。
3. ポンド記号 '#' の後から行末までの文字はすべてコメントと見なされます。
4. 3 つの指定 (ホスト名、ユーザー名、ドメイン名) の中で、ホスト名だけが認識されます。ユーザー名またはドメイン名に指定された値はすべて無視されます。

5. リスト内で各要素を区切るためにはコンマ ‘,’ を使用してください。
6. ホスト名を空白にしてもワイルドカードとは見なされません。ワイルドカードのサポートはありません。
7. ネットグループ名またはホスト名の途中でなければ、どこにでも空白を入れることができます。
8. 16 進文字を組み込むことはできません。
9. 制御文字は使用できません (これは ASCII の x020 未満のすべての文字が含まれます)。
10. 項目では、大文字 / 小文字の区別はありません。
11. 各トリプレット (x,y,z) (a,b,c) は、スペースまたはコンマだけで区切ることができます。

/etc/rpcbtap ファイル

/etc/rpcbtap ファイルは、RPC バインダー (ポート・マッパー) デーモンによって使用されます。すべてのデーモンに関する情報はこのファイルに保管され、RPC バインダー (ポート・マッパー) の回復時に、デーモンを登録するために使用されます。

注: /etc/rpcbtap ファイルには決して書き込んだり、変更を加えたりしないことをお勧めします。

/etc/statd ファイル

/etc/statd ファイルは、サーバーのネットワーク状況モニター・デーモンによって排他的に使用されます。このファイルには、オープンされているサーバーのファイルに対してバイト範囲ロックを行っているすべてのクライアントに関する情報が保管されています。破損の後で、NSM デーモンを開始すると、その NSM デーモンは、このファイル内にリストされているすべてのクライアントと連絡をとり、それらのクライアントがサーバーに対して保持しているロックをすべて解除できるようにします。

注: /etc/statd ファイルには決して書き込んだり、変更を加えたりしないことをお勧めします。

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あり、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。実施権の許諾については、下記の宛先に書面にてご照会ください。

〒106-0032 東京都港区六本木 3 丁目 2-31

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ます。

本書は定期的に見直され、必要な変更 (たとえば、技術的に不適切な表現や誤植など) は、本書の次版に組み込まれます。IBM は予告なしに、随時、本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。

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

3605 Highway 52 N

Rochester, MN 55901-7829

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、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。

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。したがって IBM は、このサンプル・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。お客様は、IBM 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。

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。

コードの特記事項情報

本書には、プログラミング・サンプルが含まれています。

IBM は、お客様に、このプログラムをサンプルとして使用することができる非独占的な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。

すべてのサンプル・コードは、例として示す目的でのみ、IBM により提供されます。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。したがって、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めめかしたり、保証することはできません。

ここに含まれるすべてのプログラムは、現存するままの状態を提供され、いかなる保証条件も適用されません。第三者の権利の不侵害、商品性、特定目的適合性に関する黙示の保証の適用も一切ありません。

プログラミング・インターフェース情報

本書は、OS/400 ネットワーク・ファイル・システム・サポートを使用して、分散ネットワークを構築する担当者を支援することを目的としています。本資料では、OS/400 ネットワーク・ファイル・システム・サポートが提供する汎用プログラミング・インターフェースおよび関連するガイダンス情報について解説しています。

汎用プログラミング・インターフェースにより、お客様は OS/400 ネットワーク・ファイル・システム・サポートのサービスを含むプログラムを書くことができます。

商標

以下は、IBM Corporation の商標です。

Application System/400
AS/400
e (ロゴ)
IBM
iSeries
Operating System/400
OS/400
400

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. 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。

C-bus は、Corollary Inc. の米国およびその他の国における商標です。

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。

PC Direct は、Ziff Communications Company の米国およびその他の国における商標であり、IBM Corporation がライセンスを得て使用しています。

UNIX は、The Open Group がライセンスしている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などはそれぞれ各社の商標または登録商標です。

参考文献

ここでは、本書で説明する内容の背景となる情報や、さらに詳細な情報を取り扱っている資料をリストしています。

以下のリストには、各資料の正式名称と資料番号を記載します。OS/400 ネットワーク・ファイル・サポートは、ここに記載されている資料に言及するとき、資料の短縮名を使用していることがあります。

- iSeries Information Center のトピック「CL」(以前は SC88-5499)

ここでは、iSeries 制御言語 (CL) とコマンドを説明しています。各コマンドの説明には、構文図、パラメーター、省略時値、キーワード、および実例が含まれています。

- 「Exploring the NFS on AS/400」 (SG24-2158)

このレッドブックは、iSeries サーバー上のネットワーク・ファイル・システムを探索します。これは、NFS の機能を使用して、複数システムに分散しているデータを共用するのに役立ちます。さらに、いくつかのトピックについて説明しています。それには、機密保護に関する考慮事項、各国語サポート、FSS/400 から NFS へのマイグレーション、および NFS を使用している際によく経験することのあるいくつかの問題やそれらの解決方法などが含まれます。

- iSeries Information Center のトピック「iSeries 上の NetWare」(これまでは SC88-5475)

iSeries サーバーに接続された統合 xSeries サーバー上に Novell NetWare をインストールし、管理する方法を説明しています。また、iSeries サーバーに拡張 Netware 機能をインストールし、管理する方法についても解説しています。

- iSeries Information Center のトピック「統合ファイル・システム」(これまでは SC88-5495)

iSeries サーバーのための統合ファイル・システムに関する概念と用語を説明しています。また、それぞれの iSeries ファイル・システムの特徴だけでなく、インターフェース、CL コマンド、および API についても解説しています。

- 「光ディスク・サポート」(SC88-4016)

このブックは、OS/400 での IBM 光ディスク・サポートのユーザーズ・ガイドおよび解説書として作成されています。この資料は、光ディスク・ライブラリー・データ・サーバーを管理するために必要な光ディスク・ライブラリー・データ・サーバーの概念を理解するのに役立ちます。

- iSeries Information Center のトピック「基本システム機密保護および計画」(これまでは SC88-5481)

OS/400 の機密保護と権限に関する一般情報を提供します。

- 「機密保護解説書」(SD88-5027)

OS/400 の機密保護に関する詳細な技術情報を提供します。

- iSeries Information Center のトピック「API」(これまでは SC88-5101)

各 OS/400 API について解説しています。統合ファイル・システム API も含まれています。

- iSeries Information Center のトピック「実行管理機能」

作業管理およびジョブ管理の概念について解説します。

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

アプリケーション・プログラミング・インターフェース (API) 89, 90, 91, 100
エラー状態
EACCES エラー状態 89
EEXIST エラー状態 90
ENOENT エラー状態 90
ESTALE エラー状態 89
更新された API
create() 91
fcntl() 91
mkdir() 91
open() 91
追加のエラー番号 89
ネットワーク・ファイル・システムの機能 89
QSYCHGID 100
必要な項目 100
プロセス 100
暗号化
紹介 94
タイプの定義 94
エクスポート、ファイル・システムの 32
無名ユーザー 102
エクスポートとは何か 29
エクスポートの規則 33
エクスポート・オプション 102
エクスポート・テーブル 29
概説 1
画面 36
下流メタフォー 33
簡単な図解 1
機密保護に関する考慮事項 101, 102, 103
シナリオ 39
紹介 29
制約事項 36
「世界中」に向けてエクスポートする 102
「世界中」に向けてエクスポートするシナリオ 103
説明 29
定義 1

エクスポート、ファイル・システムの (続き)
どのファイル・システムをエクスポートできるか 31
なぜエクスポートが必要なのか 30
ファイル・システムをエクスポートする方法 33
プロセス 29, 33
例 38
CHGNFSEXP コマンド 35
紹介 32
EXPORTFS コマンド 35
/etc/exports ファイル 29
/etc/netgroup 36
エラー状態
追加のエラー番号 89
EACCES エラー状態 89
EEXIST エラー状態 90
ENOENT エラー状態 90
ESTALE エラー状態 89
遠隔プロシージャ呼び出し (RPC)
クライアント / サーバーの通信 7
大文字 / 小文字の区別 19
パターン照合 68
ユーザー定義ファイル・システム (UDFS) 72
ルート (/) ファイル・システム 64
CASE パラメーター 19
CRTUDFS 画面 18
CRTUDFS コマンド 17, 19
QSYS.LIB ファイル・システム 68
UDFS サポート 17
UDFS の作成 17
大文字 / 小文字の区別 19
/etc/netgroup ファイル 112

[カ行]

開始 76, 78
紹介 75
不適切な開始
起こり得る不具合 76
NFS クライアント
STRNFSSVR コマンド 78
NFS クライアントのシナリオ 77
NFS サーバー
STRNFSSVR コマンド 78
NFS サーバーのシナリオ 77
画面
ADDMFS 画面 55
CHGNFSEXP 画面 36

画面 (続き)
CRTUDFS 画面 18
DLTUDFS 画面 21
DSPMFSINF 画面 59
DSPMFSINF 出力画面 (1/2) 60
DSPMFSINF 出力画面 (2/2) 61
DSPUDFS 画面 19
DSPUDFS の出力 (1/2) 19
DSPUDFS の出力 (2/2) 20
ENDNFSSVR 画面 82
EXPORTFS 画面 36
MOUNT 画面 55
RLSIFSLCK 画面 87
RMVMFS 画面 58
STATFS 画面 59
STATFS 出力画面 (1/2) 60
STATFS 出力画面 (2/2) 61
STRNFSSVR 画面 79
UNMOUNT 画面 58
関連情報 117
機密保護 95, 96, 97, 99, 102, 103
エクスポート、ファイル・システムの 101, 102, 103
エクスポート・オプション 102
考慮事項 93, 95, 101
紹介 93
信頼できる共同体 93
データの暗号化 94
ユーザー権限 95
補足 GID 96
GID 95
UID 95, 96, 97, 99
キャッシュ 14, 15, 16
紹介 14
データ・キャッシュ
機能 15
局所性 15
クライアント・タイムアウト 16
説明 15
定義 8
ディレクトリーおよびファイル属性キャッシュ
機能 14
説明 14
クライアント / サーバーの関係
クライアント 9, 90
クライアント / サーバーの通信 7
クライアント / サーバー・モデル 7
クライアント定義 7
サーバー 9
サーバーの定義 7

クライアント / サーバーの関係 (続き)

紹介 7
通信の設計 8

[サ行]

参考文献 117

遮断 76, 81

紹介 75

不適切な遮断

起こり得る不具合 76

NFS クライアント

ENDNFSSVR コマンド 81

NFS クライアントのシナリオ 81

NFS サーバー

ENDNFSSVR コマンド 81

NFS サーバーのシナリオ 80

状態 4

状態あり

定義 4

状態なし

定義 4

[タ行]

データの暗号化

紹介 94

タイプの定義 94

デーモン 12

クライアント・デーモン 13

定義 8

ネットワーク・ロック管理プログラ

ム・デーモン (NLMD) 12

ブロック I/O デーモン (BIOD) 14

マウント・デーモン (MNTD) 11

NFS クライアントのデーモンを表示す

る 81

NFS サーバーのデーモンを表示する

78

NFS サーバー・デーモン 11

NFS サーバー・デーモン (NFSD) 11

RPC バインダー・デーモン

(RPCD) 11

[ナ行]

名前空間

エクスポート、ファイル・システムの
30, 31, 33

エクスポートの規則 33

概念 5

概要 5

機密保護に関する考慮事項 93

シナリオ 5

信頼できる共同体 93

名前空間 (続き)

定義 5

バイト範囲ロック 86

補助記憶域プール (ASP) 26

マウント、ファイル・システムの 50

ユーザー定義ファイル・システム

(UDFS) 26

TULAB の名前空間 7, 26, 31, 33,

52, 97, 99

ネットワーク状況モニター・デーモン

(NSMD)

開始の順序 78

遮断の順序 81

説明 12

通知リスト 12

表示 78, 81

ネットワーク・ファイル・システム 8,

10, 11, 12, 13, 14, 15, 16, 75, 76, 77,

78, 80, 81, 85, 86, 87, 93

アプリケーション・プログラミング・
インターフェース (API) 89

エクスポート、ファイル・システムの
29

沿革 3

エンコード 9

開始

起こり得る不具合 76

紹介 75

NFS クライアント 78

NFS クライアントのシナリオ 77

NFS サーバー 78

NFS サーバーのシナリオ 77

階層について 9

機能の概要 1

機密保護に関する考慮事項

紹介 93

キャッシュ

定義 8

クライアント 9, 12, 90

キャッシュ 14

データ・キャッシュ 15, 16

デーモン 13

ディレクトリーおよびファイル属性

キャッシュ 14

内容 12

ブロック I/O デーモン (BIOD) 14

クライアント / サーバー 7

クライアント / サーバーの関係 8

クライアント / サーバーの通信 7

サーバー 9, 10

内容 10

ネットワーク状況モニター・デー
モン (NSMD) 12

ネットワーク・ロック管理プログラ
ム・デーモン (NLMD) 12

マウント・デーモン (MNTD) 11

ネットワーク・ファイル・システム (続
き)

サーバー (続き)

NFS サーバー・デーモン

(NFSD) 11

RPC バインダー・デーモン (ポー
ト・マップ) 11

遮断

起こり得る不具合 76

紹介 75

NFS クライアント 81

NFS クライアントのシナリオ 81

NFS サーバー 81

NFS サーバーのシナリオ 80

紹介 1

説明 3

デーモン

定義 8

伝送 9

統合ファイル・システム 89

バイト範囲ロック

回復 85

状態あり 86

状態なし 86

なぜファイルをロックする必要があ
るのか 86

ファイルをロックする方法 86

RLSIFSLCK コマンド 87

プロセスのレイアウト 9

プロトコル 9

マウント、ファイル・システムの 47

ユーザー定義ファイル・システム

(UDF) 17

API 89

iSeries のファイル・システムを使用す
る 63

NFS とは何か 1

ネットワーク・ロック管理プログラム・デ
ーモン (NLMD)

開始の順序 78

遮断の順序 81

説明 12

バイト範囲ロック 12, 86

表示 78, 81

[ハ行]

バイト範囲ロック

紹介 85

使用する理由 86

状態あり 86

状態なし 86

プロセス 86

モニターされない 12

モニターされる 12

RLSIFSLCK コマンド 87

- ファイル・システム 3, 17, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72
 - エクスポート 31
 - エクスポートできるもの 31
 - オープン・システム・ファイル・システム (QOpenSys)
 - 大文字 / 小文字の区別 65
 - ネットワーク・ファイル・システムでの相違 64
 - 読み取り / 書き込みオプション 65
 - どこにマウントできるか 51
 - 何をマウントできるか 54
 - ネットワーク・ファイル・システム
 - 説明 3
 - 光ディスク・ファイル・システム (QOPT)
 - 大文字 / 小文字の区別 71
 - 機密保護 71
 - 許可 71
 - ネットワーク・ファイル・システムでの相違 70
 - マウント 70
 - 文書ライブラリー・サービス・ファイル・システム (QDLS)
 - 無名ユーザー 69
 - ネットワーク・ファイル・システムでの相違 69
 - パス名の長さ 69
 - ファイルの作成 69
 - マウント 69
 - ユーザー定義ファイル・システム (UDFS)
 - 大文字 / 小文字の区別 72
 - ネットワーク・ファイル・システムでの相違 72
 - 補助記憶域プール (ASP) 72
 - ライブラリー・ファイル・システム (QSYS.LIB)
 - エクスポート 66
 - 大文字 / 小文字の区別 68
 - データベース・メンバー 67
 - ネットワーク・ファイル・システムでの相違 66
 - バイト範囲ロック 68
 - パス名 67
 - ファイル・モード 67
 - マウント 67
 - ユーザー・スペース 67
 - QPWFSEVER 権限リスト 67
 - .FILE オブジェクト 67
 - ルート (/) ファイル・システム
 - 大文字 / 小文字の区別 64
 - ネットワーク・ファイル・システムでの相違 63
- ファイル・システム (続き)
 - ルート (/) ファイル・システム (続き)
 - 読み取り / 書き込みオプション 64
 - 論理構造 3
 - iSeries のファイル・システム 63
 - ブロック I/O デーモン (BIOD)
 - 説明 14
 - 表示 81
 - ブロック特殊ファイル (*BLKSF オブジェクト)
 - 概念 17
 - 定義 17
 - プロトコル 10, 90
 - 状態なし 4
 - NFS 4, 9
 - 単一スレッド 10
 - NFS プロトコル 90
 - RPC 4, 9
 - TCP 4, 9, 81
 - UDP 4, 9, 81
 - 考慮事項 90
 - XDR 9
 - 補助記憶域プール (ASP) 18, 19
 - シナリオ 26
 - 定義 17
 - 導入 17
 - CRTUDFS 画面 18
 - CRTUDFS コマンド 17
 - 例 19
 - UDFS パラメーター 18
 - UDFS と共に使用する 26
 - 補足 GID
 - 紹介 96
 - 定義 96
- [マ行]**
 - マウント、ファイル・システムの 50, 51, 52, 57
 - アクセス可能性 49
 - 覆い隠されたオブジェクト 49
 - 概説 1
 - 画面 55, 58, 59
 - 簡単な図解 1
 - グラフィカル・ユーザー・インターフェイス (GUI) 56
 - 紹介 47
 - 制約事項 55, 58
 - 説明 47
 - 定義 1
 - どのようなファイル・システムをマウントできるか
 - ネットワーク・ファイル・システム 50
 - マウント、ファイル・システムの (続き)
 - どのようなファイル・システムをマウントできるか (続き)
 - ユーザー定義ファイル・システム 50
 - Novell NetWare ファイル・システム 50
 - なぜファイル・システムをマウントするのか 50
 - 何をマウントできるか 54
 - ファイル・システム
 - どこにマウントできるか 51
 - マウント先 51
 - マウントの上からマウントする 52
 - ファイル・システムをどこにマウントできるか 51
 - ファイル・システムをマウントする方法 54
 - プロセス 54
 - マウントとは何か 47
 - マウント・ポイント 3, 54
 - 例 25, 59, 61
 - ADDMFS コマンド 54
 - DSPMFSINF コマンド 59
 - MOUNT コマンド 54
 - QSTRVPPGM プログラム 48
 - RMVMFS コマンド
 - 説明 57
 - 目的 57
 - STATFS コマンド 59
 - /etc/fstab ファイル 48
 - マウント・デーモン (MNTD)
 - 開始の順序 78
 - 遮断の順序 81
 - 説明 11
 - 表示 78, 81
 - マウント・ポイント
 - 説明 54
 - モニターされないロック 12
 - 定義 12
 - モニターされるロック
 - 定義 12
- [ヤ行]**
 - ユーザー権限 95, 96, 97, 99
 - 紹介 95
 - 補足 GID
 - 紹介 96
 - 定義 96
 - GID
 - 紹介 95
 - 定義 95
 - UID
 - 管理者の責務 97
 - 紹介 95

ユーザー権限 (続き)

UID (続き)

定義 95

マッピング 96, 99

例 97

ユーザー定義ファイル・システム

(UDFS) 17, 18, 19, 20, 21, 22, 23, 26, 72

グラフィカル・ユーザー・インターフェース (GUI) 23

説明 17

定義 17

ネットワーク・ファイル・システムでの相違 72

大文字 / 小文字の区別 72

補助記憶域プール (ASP) 72

ネットワーク・ファイル・システムの機能 26

補助記憶域プール (ASP) 26

ファイル・システムの管理 17

復元 23

ブロック特殊ファイル 17, 26

保管 23

補助記憶域プール (ASP) 17

UDFS の削除 20

画面 21

制約事項 20

例 21

UDFS の作成 17

画面 18

制約事項 17

例 19

UDFS の表示 19

画面 19, 20

例 20

UDFS のマウント 21

画面 22

プロセス 22

例 22

UDFS のマウント解除 22

画面 23

*BLKSF オブジェクト 17, 26

[ラ行]

リモート・プロシージャー呼び出し

(RPC)

紹介 4

定義 4

ルート (/) ファイル・システム 64

ネットワーク・ファイル・システムでの相違 63

大文字 / 小文字の区別 64

読み取り / 書き込みオプション

64

A

ADDMFS コマンド 55

オプション 14

オプション・リスト 55

画面 55

グラフィカル・ユーザー・インターフェース (GUI) 56

制約事項 55

説明 54

目的 54

例 25

CODEPAGE パラメーター

データ・ファイルのコード・ページ 55

パス名のコード・ページ 55

MFS パラメーター 55

MNOTOVRDIR パラメーター 55

OPTIONS パラメーター 55

TYPE パラメーター 55

C

CHGNFSEXP コマンド 37

オプション・リスト 37

画面 36

紹介 32

制約事項 36

説明 35

ディレクトリー項目 37

目的 35

例 38

DIR パラメーター 37

HOSTOPT パラメーター

書き込みモード 37

ネットワーク・データ・ファイルのコード・ページ 37

ネットワーク・パス名のコード・ページ 37

ホスト名 37

OPTIONS パラメーター 37

CL コマンド

コマンド表 105

よく使用されるコマンド 105

ADDMFS (マウント済みファイル・システムの追加) 21, 54

CHGNFSEXP (ネットワーク・ファイル・システム・エクスポートの変更) 35

CRTUDFS (ユーザー定義ファイル・システムの作成) 17

DLTUDFS (ユーザー定義ファイル・システムの削除) 20

DSPMFSINF (マウント済みファイル・システム情報の表示) 59

CL コマンド (続き)

DSPUDFS (ユーザー定義ファイル・システムの作成) 19

ENDNFSSVR (ネットワーク・ファイル・システム・サーバーの終了) 81

EXPORTFS (ネットワーク・ファイル・システム・エクスポートの変更) 35

MOUNT (マウント済みファイル・システムの追加) 21, 54

RLSIFSLCK (統合ファイル・システム・ロックの解除) 87

RMVMFS (マウント済みファイル・システムの除去) 22, 57

RST (復元) 23

SAV (保管) 23

STATFS (マウント済みファイル・システム情報の表示) 59

STRNFSSVR (ネットワーク・ファイル・システム・サーバーの開始) 78

UNMOUNT (マウント済みファイル・システムの除去) 22, 57

CRTUDFS コマンド

CASE パラメーター 18

CRTOBJAUD パラメーター 18

DTAAUT パラメーター 18

OBJAUT パラメーター 18

TEXT パラメーター 18

UDFS パラメーター 18

D

DLTUDFS コマンド

UDFS パラメーター 21

DSPMFSINF コマンド

画面 59

説明 59

目的 59

例 61

OBJ パラメーター 60

DSPUDFS コマンド

UDFS パラメーター 19

E

ENDNFSSVR コマンド 82

画面 82

制約事項 82

目的 81

ENDJOBTIMO パラメーター 82

SERVER パラメーター 82

EXPORTFS

制約事項 36

EXPORTFS コマンド 37

EXPORTFS コマンド (続き)

オプション・リスト 37

画面 36

説明 35

ディレクトリー項目 37

目的 35

例 38

DIR パラメーター 37

HOSTOPT パラメーター

書き込みモード 37

ネットワーク・データ・ファイルの

コード・ページ 37

ネットワーク・パス名のコード・ペ

ージ 37

ホスト名 37

OPTIONS パラメーター 37

F

FSS/400

紹介 3

説明 3

G

GID

紹介 95

定義 95

K

Kerberos

暗号化 95

許可 95

定義 95

M

MOUNT コマンド 55

オプション 14

オプション・リスト 55

画面 55

グラフィカル・ユーザー・インターフ

ェース (GUI) 56

制約事項 55

説明 54

目的 54

例 25

CODEPAGE パラメーター

データ・ファイルのコード・ページ

55

パス名のコード・ページ 55

MFS パラメーター 55

MNOTOVRDIR パラメーター 55

OPTIONS パラメーター 55

MOUNT コマンド (続き)

TYPE パラメーター 55

N

NFS クライアント 13, 14, 15, 16

キャッシュ

紹介 14

データ・キャッシュ 15, 16

説明 12

属性キャッシュ 12

データ・キャッシュ 12

デーモン

説明 13

デーモンを表示する 81

ディレクトリーおよびファイル属性キ

ャッシュ

機能 14

説明 14

内容 12

ネットワーク状況モニター (NSM) 12

ネットワーク・ロック管理プログラム

(NLM) 12

ブロック I/O デーモン (BIOD) 12,

14

マウント解除コマンド 12

マウント・コマンド 12

RPC バインダー・デーモン

(RPCD) 12

NFS サーバー 11

エクスポート・コマンド 10

エクスポート・テーブル 10

説明 10

デーモン

説明 11

デーモンを表示する方法 78

内容 10

ネットワーク状況モニター・デーモン

(NSMD) 12

ネットワーク・ロック管理プログラ

ム・デーモン (NLMD) 12

ポート・マップ 11

マウント・デーモン (MNTD) 11

NFS サーバー・デーモン 10

NFS サーバー・デーモン (NFSD) 11

RPC バインダー・デーモン

(RPCD) 11

/etc/exports ファイル 10

NFS サーバー・デーモン (NFSD)

開始の順序 78

遮断の順序 81

説明 11

表示 78

Q

QDLS ファイル・システム 69

ネットワーク・ファイル・システムで

の相違 69

無名ユーザー 69

パス名の長さ 69

ファイルの作成 69

マウント 69

QOpenSys ファイル・システム 65

ネットワーク・ファイル・システムで

の相違 64

大文字 / 小文字の区別 65

読み取り / 書き込みオプション

65

QOPT ファイル・システム 70, 71

ネットワーク・ファイル・システムで

の相違 70

大文字 / 小文字の区別 71

機密保護 71

許可 71

マウント 70

QSYCHGID API

紹介 100

説明 100

必要な項目 100

プロセス 100

QSYS.LIB ファイル・システム 66, 67,

68

ネットワーク・ファイル・システムで

の相違 66

エクスポート 66

大文字 / 小文字の区別 68

データベース・メンバー 67

バイト範囲ロック 68

パス名 67

ファイル・モード 67

マウント 67

ユーザー・スペース 67

QPWFSEVER 権限リスト 66

.FILE オブジェクト 67

R

RLSIFSLCK コマンド

画面 87

制約事項 87

目的 87

例 88

OBJ パラメーター 88

RMTLOCNAME パラメーター 88

RMVMFS コマンド

画面 58

制約事項 58

説明 57

目的 57

RMVMSFS コマンド (続き)

例 59

RPC バインダー (ポート・マップ) デーモン (RPCD)

説明 11

RPC バインダー・デーモン (RPCD)

開始の順序 78

遮断の順序 81

表示 78, 81

S

STATFS コマンド

画面 59

説明 59

目的 59

例 61

OBJ パラメーター 60

STRNFSSVR コマンド

画面 79

制約事項 79

目的 78

例 80

NBRBIO パラメーター 79

NBSVR パラメーター 79

SERVER パラメーター 79

STRJOBTIMO パラメーター 79

Sun Microsystems, Inc.

沿革 3

NFS の最初のリリース 3

T

TCP/IP

タイムアウトの対立 81

TCP/IP ファイル・サーバー・サポート OS/400 用 (FSS/400用)

紹介 3

説明 3

U

UID 96, 97, 99, 100

紹介 95

定義 95

マッピング 96

可能性 96

管理者の責務 97

適切なマッピング 99

例 97

UID の変更 99, 100

UID の変更

プロセス 99

QSYCHGID API 100

UNIX 11, 48, 64, 68, 72, 73

UNIX (続き)

大文字 / 小文字の区別 68, 72

許可 94

権限

*X (実行) 権限 73

自動マウント

/etc/fstab ファイル 48

ディレクトリーの作成 73

ネットワーク・ファイル・システムでの相違 73

パターン照合 68

ポート・マップ・デーモン 11

補足グループ ID (GID) 96

マウント、ファイル・システムの 72

ユーザー ID (UID) のマッピング 103

リモート・クライアント 73

ユーザー定義ファイル・システム (UDFS) 72

ルート (/) ファイル・システム 64

QSYS.LIB ファイル・システム 68

LS (リスト) コマンド 68

NFS の開発 3

NFS 要求 103

RPC バインダー・デーモン (RPCD)

ポート・マップ・デーモン 11

/etc/fstab ファイル 48

UNMOUNT コマンド

画面 58

制約事項 58

説明 57

目的 57

例 59

/etc/exports ファイル 109, 110

項目の形式 109

HOSTOPT パラメーター 109

HOSTOPT パラメーターの形式の規則 110

説明 109

目的 109

/etc/netgroup ファイル

形式の規則 111

説明 111

目的 111

/etc/pmap ファイル

説明 112

目的 112

/etc/statd ファイル

説明 112

目的 112

[特殊文字]

/etc ファイル 107, 109, 110

エクスポート・コマンド

ファイルの作成 107

/etc/exports 107

/etc/netgroup 107

紹介 107

説明 107

/etc/exports ファイル 109

項目の形式 109, 110

説明 109

目的 109

/etc/netgroup ファイル 111

形式の規則 111

説明 111

目的 111

/etc/pmap ファイル 112

説明 112

目的 112

/etc/statd ファイル 112

説明 112

目的 112



Printed in Japan

SC88-4019-00



日本アイ・ビー・エム株式会社
〒106-8711 東京都港区六本木3-2-12