



Solaris のシステム管理 (ネーミングとディレクトリサービス : NIS+ 編)

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819-0384-10
2005 年 1 月

Copyright 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

本製品およびそれに関連する文書は著作権法により保護されており、その使用、複製、頒布および逆コンパイルを制限するライセンスのもとにおいて頒布されます。サン・マイクロシステムズ株式会社の書面による事前の許可なく、本製品および関連する文書のいかなる部分も、いかなる方法によっても複製することが禁じられます。

本製品の一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。UNIX は、X/Open Company, Ltd. が独占的にライセンスしている米国ならびに他の国における登録商標です。フォント技術を含む第三者のソフトウェアは、著作権により保護されており、提供者からライセンスを受けているものです。

U.S. Government Rights Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

本製品に含まれる HG-MinchoL, HG-MinchoL-Sun, HG-PMinchoL-Sun, HG-GothicB, HG-GothicB-Sun, および HG-PGothicB-Sun は、株式会社リコーがリョービマジクス株式会社からライセンス供与されたタイプフェースマスタをもとに作成されたものです。HeiseiMin-W3H は、株式会社リコーが財団法人日本規格協会からライセンス供与されたタイプフェースマスタをもとに作成されたものです。フォントとして無断複製することは禁止されています。

Sun, Sun Microsystems, docs.sun.com, AnswerBook, AnswerBook2, SunOS, iPlanet, Sun ONE, および Sun Java System Directory Server は、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標、登録商標もしくは、サービスマークです。

サンのロゴマークおよび Solaris は、米国 Sun Microsystems 社の登録商標です。

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャに基づくものです。

OPENLOOK, OpenBoot, JLE は、サン・マイクロシステムズ株式会社の登録商標です。

Wnn は、京都大学、株式会社アステック、オムロン株式会社で共同開発されたソフトウェアです。

Wnn6 は、オムロン株式会社、オムロンソフトウェア株式会社で共同開発されたソフトウェアです。© Copyright OMRON Co., Ltd. 1995-2000. All Rights Reserved. © Copyright OMRON SOFTWARE Co., Ltd. 1995-2002 All Rights Reserved.

「ATOK」は、株式会社ジャストシステムの登録商標です。

「ATOK Server/ATOK12」は、株式会社ジャストシステムの著作物であり、「ATOK Server/ATOK12」にかかる著作権その他の権利は、株式会社ジャストシステムおよび各権利者に帰属します。

「ATOK Server/ATOK12」に含まれる郵便番号辞書 (7 桁/5 桁) は日本郵政公社が公開したデータを元に制作された物です (一部データの加工を行っています)。

「ATOK Server/ATOK12」に含まれるフェイスマーク辞書は、株式会社ビレッジセンターの許諾のもと、同社が発行する『インターネット・パソコン通信フェイスマークガイド』に添付のものを使用しています。

Unicode は、Unicode, Inc. の商標です。

本書で参照されている製品やサービスに関しては、該当する会社または組織に直接お問い合わせください。

OPEN LOOK および Sun Graphical User Interface は、米国 Sun Microsystems 社が自社のユーザおよびライセンス実施権者向けに開発しました。米国 Sun Microsystems 社は、コンピュータ産業用のビジュアルまたはグラフィカル・ユーザインタフェースの概念の研究開発における米国 Xerox 社の先駆者としての成果を認めるものです。米国 Sun Microsystems 社は米国 Xerox 社から Xerox Graphical User Interface の非独占的ライセンスを取得しており、このライセンスは米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、明示的であるか黙示的であるかを問わない、なんらの保証も行われぬものとします。

本製品が、外国為替および外国貿易管理法 (外為法) に定められる戦略物資等 (貨物または役務) に該当する場合、本製品を輸出または日本国外へ持ち出す際には、サン・マイクロシステムズ株式会社の事前の書面による承諾を得ることのほか、外為法および関連法規に基づく輸出手続き、また場合によっては、米国商務省または米国所轄官庁の許可を得ることが必要です。

原典: *System Administration Guide: Naming and Directory Services (NIS+)*

Part No: 816-4558-10

Revision A



041228@10536



目次

はじめに 21

パート I ネームサービスおよびディレクトリサービスの紹介 27

- 1 ネームサービススイッチ 29
 - ネームサービススイッチについて 29
 - nsswitch.conf ファイルのフォーマット 30
 - nsswitch.conf ファイル中のコメント 34
 - キーサーバーとスイッチファイルの publickey エントリ 34
 - nsswitch.conf テンプレートファイル 34
 - デフォルトスイッチテンプレートファイル 35
 - nsswitch.conf ファイル 37
 - 構成ファイルの変更 38
 - ▼ ネームサービススイッチの変更 38
 - NIS+ クライアントで IPv6 を使用できるようにする方法 39
 - +/- 構文との互換性を確保する 39
 - スイッチファイルとパスワード情報 40

パート II NIS+ の設定と構成 43

- 2 NIS+ の紹介 45
 - NIS+ について 45
 - NIS+ のメリット 46
 - NIS+ と NIS の違い 47
 - NIS+ のセキュリティ 50

Solaris 1.x と NIS 互換モード	51
NIS+ の管理コマンド	52
NIS+ の API	54
設定および構成の前に	54
NIS と NIS+	55
NIS+ のファイルとディレクトリ	56
NIS+ 名前空間の構造	57
NIS+ 名前空間のディレクトリ	59
NIS+ のドメイン	60
NIS+ サーバー	61
サーバーが変更を伝達する方法	63
NIS+ 主体 (クライアント)	65
NIS+ 主体	65
NIS+ クライアント	65
コールドスタートファイルとディレクトリキャッシュ	67
NIS+ サーバーはクライアントでもある	71
命名規則	72
NIS+ ドメイン名	74
ディレクトリオブジェクト名	75
テーブルおよびグループ名	75
テーブルエントリ名	76
ホスト名	76
NIS+ の主体名	77
使用できる記号	77
NIS+ の名前展開	77
環境変数 NIS_PATH	78
名前空間がすでに存在する場合の設定	79
2 通りの構成方法	80
3 NIS+ 設定スクリプト	83
NIS+ スクリプトについて	83
NIS+ スクリプトで実行すること	84
NIS+ スクリプトでは実行しないこと	84
4 スクリプトを使用した NIS+ の設定	87
NIS+ 設定の概要	87
NIS+ とサービス管理機能	89

NIS+ 名前空間サンプルの作成	92
NIS+ スクリプトのコマンド行の要約	93
NIS+ ルートサーバーの設定	95
nisserver を実行してルートサーバーを設定するための前提条件	95
▼ ルートマスターサーバーを作成する方法	96
▼ 誤った情報を変更する方法	98
▼ Multihomed NIS+ ルートマスターサーバーの設定方法	99
NIS+ テーブルの生成 (populate)	100
nispopulate を実行してルートサーバーテーブルを生成するための前提条件	100
▼ ルートマスターサーバーのテーブルを生成する方法	102
NIS+ クライアントマシンの設定	106
nisclient を実行してクライアントマシンを設定するための前提条件	107
▼ 新しいクライアントマシンを初期設定する方法	107
クライアントマシンの追加作成	109
NIS+ クライアントユーザーの初期設定	109
nisclient を実行してクライアントユーザーを初期設定するための前提条件	109
▼ NIS+ ユーザーを初期設定する方法	110
NIS+ サーバーの設定	111
NIS+ サーバーを設定するための前提条件	111
クライアントを NIS+ サーバーとして構成する方法	112
サーバーの追加作成	113
ルート複製サーバーの作成	113
nisserver を実行してルート複製サーバーを作成するための前提条件	114
▼ ルート複製サーバーを作成する方法	115
▼ Multihomed NIS+ 複製サーバーの設定方法	117
サブドメインの作成	118
nisserver を実行してサブドメインを作成するための前提条件	119
▼ 新しいルート以外のドメインを作成する方法	120
ドメインの追加作成	121
新しいサブドメインのテーブルの生成	121
nispopulate を実行してサブドメインのテーブルを生成するための前提条件	121
マスターサーバーテーブルを生成する方法	123
サブドメイン複製サーバーの作成	124
nisserver を実行してサブドメイン複製サーバーを作成するための前提条件	124
▼ 複製サーバーを作成する方法	125

サブドメインの NIS+ クライアントマシンの初期設定	125
nisclient を実行してクライアントマシンを初期設定するための前提条件	126
▼ 新しいサブドメインクライアントマシンを初期設定する方法	126
サブドメインの NIS+ クライアントユーザーの初期設定	127
nisclient を実行してサブドメインのクライアントユーザーを初期設定するための前提条件	127
▼ NIS+ サブドメインユーザーを初期設定する方法	128
NIS+ 名前空間サンプルで使ったコマンドのまとめ	128
5 ルートドメインの設定	131
ルートドメインの設定方法の概要	131
標準構成と NIS 互換構成の手順の相違	132
ルートドメインを確立する	132
手順の要約	133
ルートドメインを確立する — 作業マップ	134
セキュリティについて	134
ルートドメインを確立するための前提条件	134
必要な情報	134
▼ ルートドメインを構成する方法	135
ルートドメイン構成の要覧	145
6 NIS+ クライアントの構成	149
NIS+ クライアントの設定の概要	149
クライアントの設定	150
クライアント構成時のセキュリティについて	151
クライアントの資格を構成するための前提条件	151
必要な情報	152
クライアントの設定 — 作業マップ	152
▼ NIS+ クライアントを設定する方法	152
DNS 転送の設定	154
マシンのドメイン名を変更する	155
マシンのドメイン名変更時のセキュリティについて	155
必要な情報	155
マシンのドメインの変更 - 作業マップ	155
▼ クライアントのドメイン名を変更する方法	155
NIS+ クライアントを初期設定する	156
ブロードキャストにより初期設定する	157
ブロードキャストによりクライアントを初期設定する方法	157

ホスト名により NIS+ クライアントを初期設定する	158
コールドスタートファイルを使用してクライアントを初期設定する	159
NIS+ クライアント構成の要覧	161

7 NIS+ サーバーの構成 163

NIS+ サーバーを設定する	163
標準構成と NIS 互換構成の相違 (NIS+ サーバー)	164
NIS+ サーバー構成時のセキュリティについて	164
NIS+ サーバーを構成するための前提条件	165
必要な情報	165
▼ NIS+ サーバーを構成する方法	165
複製サーバーを既存のドメインに追加する	166
NIS+ コマンドを使って複製サーバーを構成する	168
NIS+ コマンドを使って複製サーバーを構成する— 作業マップ	168
▼ NIS+ コマンドを使って複製サーバーを構成する方法	168
nisrestore を使ってデータを複製サーバーにロードする	169
nisrestore を使ってデータを複製サーバーにロードする — 作業マップ	170
▼ 名前空間データをロードする方法—nisrestore による方法	170
nisping を使ってデータを複製サーバーにロードする	171
nisping を使ってデータを複製サーバーにロードする — 作業マップ	172
▼ 名前空間データをロードする方法—nisping による方法	172
サーバー構成の要覧	173

8 ルート以外のドメインの構成 175

ルート以外のドメインを設定する	175
標準構成と NIS 互換構成の相違 (ルート以外のドメイン)	176
ルート以外のドメイン設定時のセキュリティについて	176
ルート以外のドメインを設定するための前提条件	177
必要な情報	177
ルート以外のドメインを設定する — 作業マップ	177
▼ ルート以外のドメインの設定方法	177
サブドメイン構成の要覧	182

9 NIS+ テーブルの設定 185

テーブルの設定	185
テーブルの生成の方法	186
NIS+ テーブルをファイルから生成する	187

ファイルからテーブルを生成する場合のセキュリティについて	187
ファイルからテーブルを生成するための前提条件	188
必要な情報	188
NIS+ テーブルをファイルから生成する — 作業マップ	188
▼ NIS+ テーブルをファイルから生成する方法	188
NIS+ テーブルを NIS マップから生成する	192
NIS マップからテーブルを生成する際のセキュリティについて	193
NIS マップからテーブルを生成するための前提条件	193
必要な情報	193
NIS+ テーブルを NIS マップから生成する — 作業マップ	194
▼ NIS+ テーブルをマップから生成する方法	194
NIS+ から NIS に情報を転送する	197
NIS から NIS+ に情報を転送する際のセキュリティについて	197
NIS から NIS+ に情報を転送するための前提条件	197
NIS+ から NIS に情報を転送する — 作業マップ	197
▼ NIS+ から NIS へ情報を転送する方法	197
所有者および管理者に対する Passwd へのアクセス制限	198
Passwd 列へのアクセスを制限する際のセキュリティについて	198
Passwd 列へのアクセスを制限するための前提条件	198
必要な情報	199
所有者および管理者に対する Passwd 列へのアクセス制限 — 作業マップ	199
▼ パスワード列へのアクセスを制限する方法	199
テーブルの生成のまとめ	200

パート III NIS+ の管理 203

10	NIS+ のテーブルと情報	205
	NIS+ テーブル構造	205
	列とエントリ	207
	検索パス	208
	テーブルの設定方法	210
	テーブルの更新方法	211
 11	 NIS+ のセキュリティの概要	 213
	Solaris のセキュリティ - 概要	213
	NIS+ のセキュリティ - 概要	215
	NIS+ の主体について	217

	NIS+ セキュリティレベル	217
	セキュリティレベルとパスワードコマンド	218
	NIS+ の認証と資格 - 紹介	218
	ユーザーおよびマシンの資格	218
	DES と LOCAL 資格	218
	ユーザー種類と資格種類	220
	NIS+ の承認とアクセス - 紹介	220
	承認クラス	221
	NIS+ アクセス権について	224
	NIS+ 管理者	225
	NIS+ のパスワード、資格、およびコマンド	226
12	NIS+ 資格の管理	227
	NIS+ 資格について	227
	資格の機能	228
	資格と資格情報	228
	認証	228
	主体の認証方法	229
	DES 資格の詳細	232
	DES 資格の Secure RPC ネット名	232
	DES 資格確認フィールド	233
	DES 資格の作成方法	233
	Secure RPC パスワードとログインパスワードの問題	235
	キャッシュに保存された非公開鍵の問題	236
	資格に関連する情報が格納されている場所	236
	cred テーブルの詳細	238
	資格情報の作成	239
	nisaddcred コマンド	239
	資格関連のコマンド	240
	nisaddcred コマンドを使って資格情報を作成する方法	241
	Secure RPC ネット名と NIS+ 主体名	242
	管理者のために資格情報を作成する方法	243
	NIS+ 主体の資格情報を作成する方法	243
	NIS+ 資格情報の管理	247
	自分の資格情報を更新する方法	247
	資格情報を削除する方法	247

13	NIS+ 鍵の管理	249
	NIS+ 鍵について	249
	NIS+ を使用したキーログイン	250
	NIS+ 主体の鍵の変更	251
	鍵の変更	252
	ルートからルート鍵を変更する	252
	別のマシンからルート鍵を変更する手順	253
	複製からルート複製の鍵を変更する手順	254
	ルート以外のサーバーの鍵の変更手順	255
	公開鍵の更新	255
	nisupdkeys コマンド	255
	公開鍵の引数の更新と例	256
	IP アドレスの更新	257
	クライアントの鍵情報を更新する	258
	クライアントの鍵情報を一括で更新する	258
14	拡張セキュリティ資格の管理	259
	Diffie-Hellman 拡張鍵	259
	新しい公開鍵ベースのセキュリティメカニズムへの移行	260
	NIS+ セキュリティメカニズムの構成	260
	新しいセキュリティメカニズム資格の作成	261
	新しいセキュリティメカニズム資格 - 例	261
	NIS+ ディレクトリオブジェクトへの新しい鍵の追加	261
	NIS+ ディレクトリオブジェクトへの新しい公開鍵の追加 - 例	262
	新しいセキュリティメカニズム資格を受け入れるようにする NIS+ サーバーの構成	262
	新しいセキュリティメカニズム資格を受け入れるようにする NIS+ サーバーの構成 - 例	262
	新しいセキュリティメカニズム資格を使用するようにするマシンの構成	263
	新しいセキュリティメカニズム資格を使用するようにするマシンの構成 - 例	263
	新しい資格を保護するパスワードの変更	264
	新しい資格を保護するパスワードの変更 - 例	264
	新しいセキュリティメカニズム資格だけを受け入れるようにするサーバーの構成	264
	新しいセキュリティメカニズム資格だけを受け入れるようにするサーバーの構成 - 例	264
	cred テーブルからの古い資格の削除	265
	cred テーブルからの古い資格の削除 - 例	265

15	NIS+ のアクセス権の管理	267
	NIS+ のアクセス権について	267
	承認およびアクセス権について	268
	承認クラス - 復習	268
	アクセス権 - 復習	268
	アクセス権の連鎖	269
	アクセス権の割り当てと変更方法	269
	テーブル、列、およびエントリのセキュリティ	270
	アクセス権の格納場所	274
	NIS+ オブジェクトのアクセス権の読み取り	274
	デフォルトのアクセス権	275
	テーブルに対するアクセス権をサーバーが割り当てる方法	276
	コマンドによるアクセス権の指定	277
	アクセス権の構文	277
	NIS+ デフォルトの表示 - nisdefaults コマンド	280
	デフォルトセキュリティ値の設定	281
	NIS_DEFAULTS の値を表示する	281
	デフォルトを変更する	282
	NIS_DEFAULTS の値を再設定する	282
	デフォルトを無効にする	283
	オブジェクトとエントリのアクセス権を変更する	283
	nischmod コマンドを使用して権利を追加する	283
	nischmod を使用して権利を削除する	284
	列アクセス権を指定する	284
	テーブル作成時の列権利設定	285
	既存のテーブル列に権利を追加する	286
	テーブル列から権利を削除する	286
	オブジェクトとエントリの所有権の変更	286
	nischown コマンドを使用してオブジェクトの所有者を変更する	287
	nischown コマンドを使用してテーブルエントリの所有者を変更する	287
	オブジェクトまたはエントリグループの変更	288
	nischgrp コマンドを使用してオブジェクトのグループを変更する	288
	nischgrp コマンドを使用してテーブルエントリのグループを変更する	288
16	パスワードの管理	291
	パスワードの使用	291
	ログイン	291
	パスワードの変更	293

パスワードの選択	294
パスワードの管理	296
nsswitch.conf ファイルの必要条件	296
nispasswd コマンド	296
yppasswd コマンド	297
passwd コマンド	297
nistbladm コマンド	300
パスワード関連のコマンド	303
パスワード情報の表示	304
パスワードの変更	305
root のパスワードの変更	306
パスワードのロック	306
パスワードの使用期間に関する設定	307
パスワードの使用規則の設定 (およびそのデフォルト)	315
17 NIS+ グループの管理	319
Solaris グループ	319
NIS+ グループ	320
関連する管理コマンド	320
NIS+ グループメンバーのタイプ	321
メンバーのタイプ	321
メンバー以外の主体のタイプ	322
グループの構文	322
NIS+ グループについて niscat を使用する	323
グループのオブジェクト属性を表示する方法	323
nisgrpadm コマンド	324
NIS+ グループを作成する	324
NIS+ グループを削除する	325
NIS+ グループにメンバーを追加する	326
NIS+ グループのメンバーを表示する	326
NIS+ グループからメンバーを削除する	327
NIS+ グループのメンバーかどうかを調べる	327
18 NIS+ ディレクトリの管理	329
NIS+ ディレクトリ	329
ディレクトリについて niscat コマンドを使用する	329
ディレクトリのオブジェクト属性を表示する	330

nislsl コマンドでディレクトリを表示する	330
ディレクトリの内容を表示する (簡潔形式)	331
ディレクトリの内容を表示する (詳細形式)	332
nismkdir コマンド	332
ディレクトリを作成する	333
複製サーバーを既存のディレクトリに追加する	334
nisrmdir コマンド	336
ディレクトリを削除する	336
複製サーバーをディレクトリから切り離す	336
nisrm コマンド	337
ディレクトリ以外のオブジェクトを削除する	337
rpc.nisd デーモン	338
rpc.nisd デーモンを起動する	338
rpc.nisd デーモンを停止する	338
rpc.nisd 構文オプションを変更する	339
nisinit コマンド	339
クライアントを初期設定する	339
ルートマスターサーバーを初期設定する	340
nis_cachemgr デーモン	341
キャッシュマネージャの起動と停止	341
nisshowcache コマンド	342
NIS+ キャッシュの内容を表示する	342
ping とチェックポイントを実行する	342
nisping コマンド	343
最新更新時間を表示する	343
ping を強制的に実行する	344
ディレクトリにチェックポイントを実行する	344
nislog コマンド	346
トランザクションログの内容を表示する	346
nischttl コマンド	347
オブジェクトの生存期間を変更する	348
テーブルエントリの生存期間を変更する	349
19 NIS+ テーブルの管理	351
NIS+ のテーブル	351
テーブルでの nistbladm コマンドの使用	352
nistbladm 構文	352
nistbladm と列の値	353

nistbladm、検索可能列、キー、列の値	354
nistbladm とインデックス名	355
nistbladm とグループ	356
テーブルを作成する	356
テーブルの列を指定する	357
テーブルを削除する	359
エントリをテーブルに追加する	359
-a オプションを指定してエントリを追加する	360
-A オプションを指定してエントリを追加する	361
エントリを修正する	362
-e オプションを指定してエントリを編集する	362
-E オプションを指定してエントリを編集する	364
テーブルからエントリを削除する	364
1 つのエントリを削除する	364
複数のエントリを削除する	365
niscat コマンド	366
niscat コマンドの構文	366
テーブルの内容を表示する	367
テーブルまたはエントリのオブジェクト属性を表示する方法	367
nismatch と nisgrep コマンド	368
正規表現について	369
nismatch と nisgrep コマンドの構文	370
最初の列を検索する	371
特定の列を検索する	371
複数の列を検索する	371
nisln コマンド	372
nisln コマンドの構文	372
リンクを作成する	372
nissetup コマンド	373
ディレクトリを NIS+ ドメインに展開する	373
ディレクトリを NIS 互換ドメインに展開する方法	374
nisaddent コマンド	374
nisaddent コマンドの構文	375
ファイルから情報をロードする	375
NIS マップからデータをロードする	376
NIS+ テーブルの内容をファイルにダンプする	378

20	サーバー使用のカスタマイズ	379
	NIS+ サーバーと NIS+ クライアント	379
	デフォルトでのクライアントの検索動作	379
	優先サーバーを指定する	380
	広域ネットワーク (WAN) 上での NIS+	380
	サーバー使用の最適化 - 概要	381
	nis_cachemgr が必要	381
	グローバルテーブルまたはローカルファイル	381
	優先順位の番号の指定	382
	優先サーバー限定と全サーバー	383
	優先順位の表示	384
	サーバー名とクライアント名	384
	サーバーの優先順位	384
	サーバーの優先順位が有効になるタイミング	385
	nisprefadm コマンドの使用方法	385
	現行のサーバー優先順位の表示	387
	マシンに指定された優先順位の表示方法	387
	1 台のマシンに設定されたグローバルな優先順位の表示方法	387
	サブネットに設定されたグローバル優先順位の表示方法	388
	優先順位格付け番号の指定方法	388
	グローバルサーバー優先順位を指定する	388
	サブネットにグローバル優先順位を設定する方法	389
	個別のマシンにグローバル優先順位を設定する方法	389
	遠隔ドメインにグローバル優先順位を設定する方法	390
	ローカルサーバー優先順位を指定する	390
	ローカルマシン上に優先順位を設定する方法	391
	サーバーの優先順位を変更する	391
	サーバーの優先順位番号を変更する方法	392
	優先順位リスト内で 1 つのサーバーを別のサーバーに置換する方法	392
	優先順位リストからサーバーを削除する方法	393
	優先サーバーリスト全体を置換する方法	394
	優先サーバー限定指定	394
	優先サーバーだけを指定する方法	394
	優先サーバー以外のサーバーを使用する方法	395
	サーバーの優先順位の使用の終了	395
	グローバルサーバー優先順位を削除する方法	396
	ローカルサーバー優先順位を削除する方法	396
	ローカルからグローバルサブネットの優先順位に置換する方法	396

	ローカルからマシン固有のグローバル優先順位に置換する方法	397
	マシンにサーバーの優先順位の使用を中止させる方法	397
	サーバーの優先順位の有効化	397
	優先順位の変更内容をただちに実現する方法	398
21	NIS+ のバックアップと復元	399
	nisbackup を使用して名前空間をバックアップする	399
	nisbackup の構文	400
	nisbackup によるバックアップの対象	401
	バックアップ転送先ディレクトリ	402
	NIS+ のバックアップを日付順に保存する	402
	特定の NIS ディレクトリをバックアップする	402
	すべての NIS+ 名前空間をバックアップする	403
	バックアップディレクトリの構造	403
	バックアップファイル	404
	nisrestore を使用して NIS+ 名前空間を復元する	405
	nisrestore を実行するための前提条件	405
	nisrestore の構文	406
	nisrestore を使用する	406
	バックアップと復元を使用して複製サーバーを設定する	407
	サーバーマシンを置換する	408
	マシンを置換する場合の必要条件	408
	サーバーマシンの置換方法	408
22	NIS+ の削除	411
	クライアントマシンから NIS+ を削除する	412
	nisclient を使用してインストールした NIS+ を削除する	412
	NIS+ コマンドでインストールした NIS+ を削除する	412
	サーバーから NIS+ を削除する	413
	NIS+ 名前空間を削除する	414
23	NIS+ テーブルの情報	417
	NIS+ のテーブル	418
	NIS+ テーブルとほかのネームサービス	418
	NIS+ テーブル入力ファイルフォーマット	418
	auto_home テーブル	419
	auto_master テーブル	419

bootparams テーブル	420
client_info テーブル	422
cred テーブル	422
ethers テーブル	423
group テーブル	423
hosts テーブル	424
mail_aliases テーブル	424
netgroup テーブル	425
netmasks テーブル	426
networks テーブル	427
passwd テーブル	427
protocols テーブル	429
rpc テーブル	430
services テーブル	430
timezone テーブル	431
その他のデフォルトのテーブル	431

24 NIS+ の問題解決 433

NIS+ のデバッグオプション	433
NIS+ の管理上の問題	434
NIS+ データベースの問題	438
NIS+ と NIS の互換性の問題	439
NIS+ オブジェクトが見つからない問題	441
NIS+ の所有権とアクセス権の問題	444
NIS+ のセキュリティの問題	447
NIS+ の性能の低下とシステムのハングアップの問題	456
NIS+ のシステムリソースの問題	460
NIS+ のユーザーの問題	461
NIS+ に関するその他の問題	463

A エラーメッセージ 467

エラーメッセージについて	467
エラーメッセージの内容	467
状況によって異なる意味	468
エラーメッセージのアルファベット順ソート規則	468
エラーメッセージ内の番号	469
NIS+、FNS に共通するエラーメッセージ	469

B	本書『Solaris のシステム管理 (ネーミングとディレクトリサービス : NIS+ 編)』の更新情報	511
	Solaris 10	511
	FNS	511
	サービス管理機能 (SMF)	511
	用語集	513
	索引	521

図目次

図 2-1	階層型ドメインの例	48
図 2-2	NIS+ ディレクトリ構造の例	60
図 2-3	NIS+ ドメインの例	61
図 2-4	名前空間の構成要素の完全指定名	73
図 11-1	Solaris のセキュリティゲートとフィルタ	213
図 11-2	NIS+ セキュリティプロセスの概要	216
図 11-3	資格とドメイン	219
図 11-4	承認クラス	221
図 11-5	NIS+ のディレクトリ構造	223
図 12-1	keylogin コマンドで主体の非公開鍵を作成	234
図 12-2	DES 資格の作成	234
図 12-3	nisaddcred コマンドを使って主体の鍵を作成する方法	241
図 15-1	アクセス権の表示	275
図 21-1	nisbackup によって作成されたディレクトリの例	403
図 24-1	ディレクトリオブジェクトへの公開鍵の伝播	450

はじめに

『Solaris のシステム管理 (ネーミングとディレクトリサービス : NIS+ 編)』では、Solaris™ 10 オペレーティングシステム (Solaris OS) で NIS+ ネットワークのネームサービスを 設定、構成、および管理する方法について説明します。このマニュアルは、Solaris 10 System Administrator Collection に含まれます。

注 – NIS+ は、将来のリリースではサポートされなくなる可能性があります。NIS+ から LDAP への移行支援ツールは、Solaris 9 リリース以降で使用できます。詳細については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』および <http://www.sun.com/directory/nisplus/transition.html> を参照してください。

注 – このリリースでは、SPARC® および x86 系列のプロセッサアーキテクチャ (UltraSPARC®, SPARC64、AMD64、Pentium、Xeon EM64T) を使用するシステムをサポートします。サポートされるシステムについては、*Solaris 10 Hardware Compatibility List* (<http://www.sun.com/bigadmin/hcl1>) を参照してください。本書では、プラットフォームにより実装が異なる場合は、それを特記します。

本書では、「x86」という用語は AMD64 あるいは Intel Xeon/Pentium 製品系列と互換性のあるプロセッサを使用して製造された 32 ビットおよび 64 ビットシステムを意味します。サポートされるシステムについては、*Solaris 10 Hardware Compatibility List* を参照してください。

対象読者

このマニュアルは、経験を積んだシステムおよびネットワーク管理者を対象としています。

このマニュアルでは、Solaris のネームサービスおよびディレクトリサービスに関連するネットワーク概念について説明します。ただし、ネットワークの基礎および Solaris OS の管理ツールについては説明しません。

このマニュアルの構成

このマニュアルは、各ネームサービスおよびディレクトリサービスに対応して複数の部分に分かれています。

ネームサービスとディレクトリサービス : [パート I 「ネームサービスおよびディレクトリサービスの紹介」](#)

NIS+ の設定と構成 : [パート II 「NIS+ の設定と構成」](#)

NIS+ の管理 : [パート III 「NIS+ の管理」](#)

NIS+ エラーメッセージ : [付録 A](#)

Solaris システム管理マニュアルセットの構成

システム管理マニュアルセットに含まれる各マニュアルとその内容は、次のとおりです。

マニュアル名	内容
『Solaris のシステム管理 (基本編)』	ユーザーアカウントとグループ、サーバーとクライアントのサポート、システムのシャットダウンとブート、サービスの管理、およびソフトウェア (パッケージとパッチ) の管理

マニュアル名	内容
『Solaris のシステム管理 (上級編)』	印刷サービスの管理、端末とモデムの設定、システム資源の管理 (ディスク割り当て、アカウントティング、および crontab ファイルの管理)、システムプロセスの管理、 および Solaris ソフトウェアの障害追跡
『Solaris のシステム管理 (デバイスとファイルシステム)』	リムーバブルメディア、ディスクとデバイス、ファイルシステム、 およびデータのバックアップと復元
『Solaris のシステム管理 (IP サービス)』	TCP/IP ネットワークの管理、IPv4 と IPv6 のアドレスの管理、DHCP、IPsec、IKE、Solaris IP フィルタ、モバイル IP、IP ネットワークマルチパス (IPMP)、および IPQoS
『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS, NIS, LDAP 編)』	DNS、NIS、LDAP のネームサービスおよびディレクトリサービス (NIS から LDAP への移行および NIS+ から LDAP への移行を含む)
『Solaris のシステム管理 (ネーミングとディレクトリサービス : NIS+ 編)』	NIS+ のネーミングとディレクトリサービスの管理
『Solaris のシステム管理 (ネットワークサービス)』	Web キャッシュサーバー、時間関連サービス、ネットワークファイルシステム (NFS と Autofs)、メール、SLP、および PPP
『Solaris のシステム管理 (セキュリティサービス)』	監査、デバイス管理、ファイルセキュリティ、BART、Kerberos サービス、PAM、Solaris 暗号化フレームワーク、特権、RBAC、SASL、Solaris Secure Shell
『Solaris のシステム管理 (Solaris コンテナ: 資源管理と Solaris ゾーン)』	資源管理に関連するプロジェクトと手順、拡張アカウントティング、資源制御、公平配分スケジューラ (FSS)、資源上限デーモン (rcapd) を使用した物理メモリー制御、動的資源プール (Solaris ゾーンソフトウェア区分技術を使用した仮想化)

関連マニュアル

- 『DNS and Bind』 - Cricket Liu & Paul Albitz 共著、浅羽登志也／上水流由香 監訳、アスキー出版社、1995年

Sun のオンラインマニュアル

docs.sun.com では、Sun が提供しているオンラインマニュアルを参照することができます。マニュアルのタイトルや特定の主題などをキーワードとして、検索を行うこともできます。URL は、<http://docs.sun.com> です。

表記上の規則

このマニュアルでは、次のような字体や記号を特別な意味を持つものとして使用します。

表 P-1 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 system%
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	system% su password:
<i>AaBbCc123</i>	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm <i>filename</i> と入力します。
『 』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。
「 」	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第 5 章「衝突の回避」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	sun% grep `^#define \ XV_VERSION_STRING`

コード例は次のように表示されます。

■ C シェル

```
machine_name% command y|n [filename]
```


- C シェルのスーパーユーザー

```
machine_name# command y|n [filename]
```

- Bourne シェルおよび Korn シェル

```
$ command y|n [filename]
```

- Bourne シェルおよび Korn シェルのスーパーユーザー

```
# command y|n [filename]
```

[] は省略可能な項目を示します。上記の例は、*filename* は省略してもよいことを示しています。

| は区切り文字 (セパレータ) です。この文字で分割されている引数のうち 1 つだけを指定します。

キーボードのキー名は英文で、頭文字を大文字で示します (例: Shift キーを押します)。ただし、キーボードによっては Enter キーが Return キーの動作をします。

ダッシュ (-) は 2 つのキーを同時に押すことを示します。たとえば、Ctrl-D は Control キーを押したまま D キーを押すことを意味します。

パート I

ネームサービスおよびディレクトリ サービスの紹介

次の章では、`nsswitch.conf` ファイルについて説明します。`nsswitch.conf` ファイルは、異なるサービスの使用方法を調整するために利用します。

第 1 章

ネームサービススイッチ

この章では、ネームサービススイッチの機能と、これを使用してクライアントが 1 つまたは複数のソースからネーミング情報を入手する方法について説明します。ネームサービススイッチは、異なるネームサービスの使用方法を調整するために使います。Solaris ネームサービスとディレクトリサービスの DNS、NIS、および LDAP の概要については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「ネームサービスとディレクトリサービス (概要)」を参照してください。

ネームサービススイッチについて

ネームサービススイッチは `nsswitch.conf` という名前のファイルで、クライアントのマシンやアプリケーションがネットワーク情報を得る方法を管理します。これは、クライアントマシンまたはクライアントアプリケーションがネットワーク情報を取得する方法を制御します。ネームサービススイッチは、次のような `getXbyY()` インタフェースのいずれかを呼び出すクライアントアプリケーションによって使用されます。

- `gethostbyname()`
- `getpwnuid()`
- `getpwnam()`
- `getipnodebyname()`

各マシンの `/etc` ディレクトリには、スイッチファイルがあります。ファイルの各行は、`host`、`passwd`、`group` などの特定タイプのネットワーク情報を識別します。その後には、クライアントがネットワーク情報を探するための 1 つまたは複数のソースが続きます。

クライアントは、1 つまたは複数のスイッチのソースからネーミング情報を入手できます。たとえば、NIS+ のクライアントは、NIS+ テーブルからホスト情報を、ローカルの `/etc` ファイルからパスワード情報をそれぞれ入手できます。さらに、スイッチが各ソースを使用する条件を指定することもできます。表 1-1 を参照してください。

Solaris ソフトウェアは、インストールの過程において各マシンの /etc ディレクトリに nsswitch.conf ファイルを自動的に配置します。LDAP、NIS、NIS+、またはファイル用にスイッチファイルの 4 つの代替 (テンプレート) バージョンも /etc ディレクトリにロードされます。34 ページの「nsswitch.conf テンプレート ファイル」を参照してください。

これら 4 つのファイルは、代替デフォルトスイッチファイルです。各ファイルはそれぞれ/etc ファイル、NIS、NIS+、LDAP という異なる主要なネームサービス用に設計されています。Solaris ソフトウェアをマシンに最初にインストールする時に、インストール担当者はマシンのデフォルトのネームサービス (NIS+、NIS、ローカルファイル、または LDAP) を選択します。インストール中に、対応するテンプレートファイルが nsswitch.conf ファイルにコピーされます。たとえば、LDAP を使用しているクライアントマシンでは、インストールの過程で nsswitch ldap が nsswitch.conf にコピーされます。特殊な名前空間を持っている場合を除き、通常の操作には nsswitch.conf にコピーされるデフォルトのテンプレートファイルを使用します。

マシンの主要なネームサービスを後から変更する場合は、該当する代替スイッチファイルを nsswitch.conf にコピーします (34 ページの「nsswitch.conf テンプレートファイル」を参照)。また、/etc/nsswitch.conf ファイルの該当行を編集することで、クライアントが使用するネットワーク情報のソースを変更することもできます。この構文については以下で説明します。詳細については、38 ページの「ネームサービススイッチの変更」を参照してください。

nsswitch.conf ファイルのフォーマット

nsswitch.conf ファイルは、基本的には 16 種類の情報とそのソース (getXXbyYY () 関数の情報検索先) のリストです。順序は必ずしも以下のとおりではありません。

- aliases
- bootparams
- ethers
- group
- hosts
- ipnodes
- netgroup
- netmasks
- networks
- passwd (シャドウ情報含む)
- protocols
- publickey
- rpc
- services
- automount
- sendmailvars

次の表に、スイッチファイル内で上記の各情報タイプ用に指定できるソースの種類の詳細を示します。

表 1-1 スイッチソースの例

ソース	目的
files	クライアントの /etc ディレクトリに格納されているローカルファイル/etc/passwd など
nisplus	NIS+ テーブル。hosts テーブルなど
nis	NIS マップ。hosts マップなど
compat	パスワードとグループ情報を対象に、/etc/passwd、/etc/shadow、/etc/group ファイルで旧書式の「+」または「-」構文をサポートする
dns	ホスト情報を DNS から入手するように指定する
ldap	エントリを LDAP ディレクトリから入手するように指定する

検索規準

「単一ソース」。nisplus のような情報のソースが 1 つだけの場合、スイッチを使用している関数は、そのソースだけで情報を検索します。情報が見つかった場合、success という状態メッセージを返します。情報が見つからない場合は、検索が停止され、success 以外の状態メッセージを返します。状態メッセージに基づいて何をするかは、関数によって異なります。

「複数ソース」。テーブルに複数のソースがある場合、スイッチは最初のソースから情報検索を始めるように関数に指示します。情報が見つかった場合、success という状態メッセージを返します。情報が見つからない場合は、次のソースが検索されます。関数は必要な情報が見つかるか、return 処理によって中止されるまで全ソースの検索を続けます。必要な情報がどのソースにもなかったとき、関数は検索を停止し、non-success という状態メッセージを返します。

スイッチ状態メッセージ

関数は情報を見つけると、success という状態メッセージを返します。情報が見つからない場合は、その理由に応じて 3 種類のメッセージのうちの 1 つを返します。表示される状態メッセージを次の表に示します。

表 1-2 スイッチ状態メッセージ

状態メッセージ	意味
SUCCESS	要求されたエントリがソース内で発見された
UNAVAIL	ソースが応答しない、または使用不可。つまり、NIS+ テーブル、NIS マップ、または /etc ディレクトリのファイルが見つからなかった (アクセスできなかった)

表 1-2 スイッチ状態メッセージ (続き)

状態メッセージ	意味
NOTFOUND	エントリなし。テーブル、マップ、ファイルにアクセスしたが、必要な情報は見つからなかった
TRYAGAIN	ソース使用中のため再検索の必要あり。テーブル、マップ、ファイルは見つかったが、照会に対して応答しなかった

スイッチの動作に関するオプション

次の表に示すように、状態メッセージに対して次の 2 つの動作のどちらかで応答するようにスイッチに指示できます。

表 1-3 スイッチ状態メッセージへの応答

動作	意味
return	情報の検索を中止する
continue	次のソースがあれば、それを検索する

デフォルト検索基準

nsswitch.conf ファイルの状態メッセージと動作オプションの組み合わせによって、関数の各ステップでの動作が決まります。この状態と動作の組み合わせのことを、「検索基準」と呼びます。

スイッチのデフォルト検索基準は、どのソースについても同じです。これらを上記の状態メッセージに基づいて説明すれば次のようになります。

- SUCCESS=return。情報の検索を中止し、見つかった情報を使用して処理を続行する
- UNAVAIL=continue。次のソース (nsswitch.conf ファイルに指定されたもの) を使用して検索を続行する。次のソースがなければ、NOTFOUND という状態メッセージを返す
- NOTFOUND=continue。次のソース (nsswitch.conf ファイルに指定されたもの) を使用して検索を続行する。次のソースがなければ、NOTFOUND という状態メッセージを返す
- TRYAGAIN=continue。次のソース (nsswitch.conf ファイルに指定されたもの) を使用して検索を続行する。次のソースがなければ、NOTFOUND という状態メッセージを返す

これらはデフォルトの検索基準であるため、暗黙的に使用されます。つまり、スイッチファイルで、はっきりと指定する必要はありません。ほかの検索基準を明示してデフォルトの検索基準を変更するには、上記の *STATUS=action* という構文を使用します。たとえば、NOTFOUND 状態に対し、デフォルトの動作では次のソースに移って検索を続行します。networks など、特定の情報を設定して検索すると、検索は NOTFOUND で中止します。スイッチファイルの networks の行を、以下のように編集してください。


```
networks: nis [NOTFOUND=return] files
```

`networks: nis [NOTFOUND=return] files` は、NOTFOUND に関してデフォルトでない検索基準を設定するものです。デフォルト以外の設定をするときは `[]` を使用します。

この例では、検索関数は以下のような働きをします。

- `networks` マップが見つかり、必要な情報があった場合、関数は `SUCCESS` という状態メッセージを返します。
- `networks` マップが見つからなかった場合、関数は `UNAVAIL` という状態メッセージを返し、デフォルトにより適当な `/etc` ファイルの検索を続行します。
- `networks` マップは見つかったが必要な情報がなかった場合、関数は `NOTFOUND` という状態メッセージを返します。そして `/etc` ファイルの検索を続行する (デフォルトの動作) 代わりに検索を停止します。
- `networks` マップが使用中の場合、関数は `TRYAGAIN` という状態メッセージを返し、デフォルトで適当な `/etc` ファイルの検索を続行します。

注 - `nsswitch.conf` ファイル内の検索は、項目の記載順に実行されます。ただし、`passwd -r repository` コマンドで指定されていない限り、パスワードの更新は逆順で実行されます。詳細については、[40 ページの「スイッチファイルとパスワード情報」](#)を参照してください。

構文が正しくない場合の処理

クライアントのライブラリ関数には、`nsswitch.conf` ファイルにおいて「必要なエントリがない」、「エントリの構文が誤っている」といった場合に使用されるコンパイル時に組み込まれるデフォルトエントリが含まれています。これらのエントリは `nsswitch.conf` ファイルのデフォルトエントリと同じものです。

ネームサービススイッチは、テーブル名やソース名のスペルが正しいものとして処理をします。スペルが正しくない場合は、デフォルト値が使用されます。

auto_home と auto_master

`auto_home` テーブル、`auto_master` テーブルとマップのスイッチ検索基準は、`automount` と呼ばれる 1 つのカテゴリに統合されます。

Timezone とスイッチファイル

`timezone` テーブルはスイッチを使用しないため、スイッチファイルのリストには含まれていません。

nsswitch.conf ファイル中のコメント

nsswitch.conf ファイル中の行のうち、「#」で始まっているものはコメント行として解釈され、ファイルを検索する関数では無視されます。

行の途中にコメント文字 (#) が含まれる場合、コメント文字の前の文字列は nsswitch.conf ファイルを検索するルーチンによって解釈されます。コメント文字よりあとの文字列は、コメントとして解釈され、無視されます。

表 1-4 スイッチファイルのコメント例

行の種類	コメント例
コメント行 (無視される)	# hosts: nisplus [NOTFOUND=return] files
完全に解釈される行	hosts: nisplus [NOTFOUND=return] file
部分的に解釈される行 (「files」の部分は解釈されない)	hosts: nisplus [NOTFOUND=return] # files

キーサーバーとスイッチファイルの publickey エントリ



注意 - nsswitch.conf に変更を加えた後は、キーサーバーを再起動する必要があります。

キーサーバーは、起動時にだけネームサービススイッチ構成ファイルの publickey エントリを参照します。つまり、スイッチ構成ファイルを更新しても、再起動しない限りキーサーバーは publickey エントリへの変更を認識しません。

nsswitch.conf テンプレートファイル

Solaris ソフトウェアでは、4 つの nsswitch.conf テンプレートファイルを用意して、さまざまなネームサービスに対応しています。デフォルトの情報ソース (一次ソース、およびそれに続くもの) としては、それぞれ異なったものが指定されています。

4 つのテンプレートファイルについて説明します。

- 「NIS+ テンプレートファイル」(nsswitch.nisplus ファイル)。このテンプレートファイルで passwd、group、automount、aliases を除くすべての情報の一次ソースとして指定されているのは、NIS+ です。passwd、group、

automount、aliases の一次ソースとして指定されているのは /etc ディレクトリのファイルで、二次ソースとして指定されているのは NIS+ テーブルです。
[NOTFOUND=return] という検索基準は、「No such entry」というメッセージを受け取ったら NIS+ テーブルの検索を停止する」という意味です。また、ローカルファイルを検索するのは NIS+ サーバーを使用できない場合だけです。

- 「NIS テンプレートファイル」(nsswitch.nis ファイル)。NIS+ テーブルではなく NIS マップを使用するという点を除けば NIS+ テンプレートファイルとほぼ同じです。passwd、group の情報に関しては files nis という順序で検索するよう指定されているため、/etc/passwd と /etc/group に + エントリを指定する必要はありません。
- 「Files テンプレートファイル」(nsswitch.files ファイル)。このテンプレートファイルでは、ローカルの /etc ディレクトリのファイルだけがマシンの情報ソースとして指定されています。netgroup に関する files のソースは存在しないため、クライアントがスイッチファイルでこのエントリを使用することはありません。

要求するものにもっとも近いテンプレートファイルを nsswitch.conf にコピーし、必要に応じて修正します。

たとえば、LDAP テンプレートファイルを使用するには、以下のコマンドを入力します。

```
mymachine# cp nsswitch.ldap nsswitch.conf
```

デフォルトスイッチテンプレートファイル

Solaris ソフトウェアの 4 つのスイッチファイルを、以下に示します。

例 1-1 NIS+ スイッチファイルテンプレート (nsswitch.nisplus)

```
#
#
# /etc/nsswitch.nisplus:
#
#
# An example file that could be copied over to /etc/nsswitch.conf;
# it uses NIS+ (NIS Version 3) in conjunction with files.
#
# "hosts:" and "services:" in this file are used only if the
# /etc/netconfig file has a "-" for nametoaddr_libs of "inet"
# transports.

# the following two lines obviate the "+" entry in /etc/passwd
# and /etc/group.
passwd: files nisplus
group: files nisplus
# consult /etc "files" only if nisplus is down.
hosts: nisplus [NOTFOUND=return] files
# Uncomment the following line, and comment out the above, to use
```

例 1-1 NIS+ スイッチファイルテンプレート (nsswitch.nisplus) (続き)

```
# both DNS and NIS+. You must also set up the /etc/resolv.conf
# file for DNS name server lookup. See resolv.conf(4).
# hosts: nisplus dns [NOTFOUND=return] files
# services: nisplus [NOTFOUND=return] files
# networks: nisplus [NOTFOUND=return] files
# protocols: nisplus [NOTFOUND=return] files
# rpc: nisplus [NOTFOUND=return] files
# ethers: nisplus [NOTFOUND=return] files
# netmasks: nisplus [NOTFOUND=return] files
# bootparams: nisplus [NOTFOUND=return] files
# publickey: nisplus
# netgroup: nisplus
# automount: files nisplus
# aliases: files nisplus
# sendmailvars: files nisplus
```

例 1-2 NIS スイッチファイルテンプレート

```
#
# /etc/nsswitch.nis:
#
# An example file that could be copied over to /etc/nsswitch.conf;
# it uses NIS (YP) in conjunction with files.
#
# "hosts:" and "services:" in this file are used only if the
# /etc/netconfig file has a "-" for nametoaddr_libs of "inet"
# transports.
#
# the following two lines obviate the "+" entry in /etc/passwd
# and /etc/group.
passwd: files nis
group: files nis
# consult /etc "files" only if nis is down.
hosts: nis [NOTFOUND=return] files
networks: nis [NOTFOUND=return] files
protocols: nis [NOTFOUND=return] files
rpc: nis [NOTFOUND=return] files
ethers: nis [NOTFOUND=return] files
netmasks: nis [NOTFOUND=return] files
bootparams: nis [NOTFOUND=return] files
publickey: nis [NOTFOUND=return] files
netgroup: nis
automount: files nis
aliases: files nis
# for efficient getservbyname() avoid nis
services: files nis
sendmailvars: files
```

例 1-3 Files スイッチファイルテンプレート

```
#
# /etc/nsswitch.files:
#
```

例 1-3 Files スイッチファイルテンプレート (続き)

```
# An example file that could be copied over to /etc/nsswitch.conf;
# it does not use any naming service.
#
# "hosts:" and "services:" in this file are used only if the
# /etc/netconfig file has a "-" for nametoaddr_libs of "inet"
# transports.
passwd: files
group: files
hosts: files
networks: files
protocols: files
rpc: files
ethers: files
netmasks: files
bootparams: files
publickey: files
# At present there isn't a 'files' backend for netgroup;
# the system will figure it out pretty quickly, and will not use
# netgroups at all.
netgroup: files
automount: files
aliases: files
services: files
sendmailvars: files
```

nsswitch.conf ファイル

Solaris ソフトウェアをはじめてインストールするときのデフォルトの nsswitch.conf ファイルは、Solaris ソフトウェアをインストールする際に選択した ネームサービスで決まります。このファイルの各行は、ネットワーク情報の種類 (ホスト、パスワード、グループなど) と、それに対する 1 つ以上の情報源 (NIS+ テーブル、NIS マップ、DNS ホストテーブル、同一ワークステーション上の /etc など) を対応させています。クライアントは、この情報源から各情報を検索します。ネームサービスが選択されると、そのサービスのスイッチテンプレートファイルがコピーされ、新しい nsswitch.conf ファイルが作成されます。たとえば、NIS+ が選択された場合は、nsswitch.nisplus ファイルがコピーされて nsswitch.conf ファイルが作成されます。

/etc/nsswitch.conf ファイルは、Solaris 10 ソフトウェアにより各マシンの /etc ディレクトリに自動的に配置されます。また次の代替テンプレートファイルも作成されます。

- /etc/nsswitch.nisplus
- /etc/nsswitch.nis
- /etc/nsswitch.files

これらの代替テンプレートファイルには、NIS+ と NIS サービス、ローカルファイル、および LDAP で使用されるデフォルトのスイッチ構成が入っています。Solaris ソフトウェアをマシンに最初にインストールする時に、インストール担当者はマシンの

デフォルトのネームサービス (NIS+, NIS, ローカルファイル、または LDAP) を選択します。インストール時には、対応するテンプレートファイルが `/etc/nsswitch.conf` にコピーされます。たとえば、クライアントが NIS+ を使用しているマシンでは、インストールの過程で `nsswitch.nisplus` ファイルが `nsswitch.conf` にコピーされます。

特殊な名前空間を持っている場合を除き、通常の操作には `nsswitch.conf` にコピーされるデフォルトのテンプレートファイルを使用します。

構成ファイルの変更

マシンのネームサービスを変更するときは、そのマシンのスイッチファイルを新しいネームサービスに対応させて変更する必要があります。たとえば、マシンのネームサービスを NIS から NIS+ に変更する場合は、スイッチファイルを NIS+ に対応したものに変更する必要があります。スイッチファイルを変更するには、対応するテンプレートファイルを `nsswitch.conf` にコピーします。

NIS+ インストールスクリプトを使って NIS+ をマシンにインストールすると、NIS+ テンプレートファイルが自動的に `nsswitch.conf` にコピーされます。この場合、特にスイッチファイルをカスタマイズしたいというのであれば、スイッチファイルを明示的に変更する必要はありません。

スイッチファイルを変更する前に、ファイルに列挙されている情報源が正しく設定されていることを確認してください。たとえば、NIS+ 用スイッチファイルに変更するのであれば、ワークステーションには NIS+ サービスへのアクセス権が必要になり、ローカルファイル用スイッチファイルに変更するのであれば、それらのローカルファイルがワークステーション上に正しく設定されている必要があります。

▼ ネームサービススイッチの変更

スイッチファイルを変更する場合は、次の手順に従います。

1. クライアントにスーパーユーザーとしてログインします。
2. 使用するネームサービス用のテンプレートファイルを **`nsswitch.conf`** にコピーします。

「NIS+ 用」(NIS+ スクリプトにより自動的にコピーされる)

```
client1# cd /etc
client1# cp nsswitch.nisplus nsswitch.conf
```

「NIS 用」

```
client1# cd /etc
client1# cp nsswitch.nis nsswitch.conf
```

「ローカルの /etc ファイル用」

```
client1# cd /etc
client1# cp nsswitch.files nsswitch.conf
```

3. マシンを再起動します。

nscd ネームサービスキャッシュデーモンによってスイッチ情報がキャッシュに書き込まれます。ライブラリ関数の中には、nsswitch.conf ファイルが変更されたかどうかを定期的にチェックしないものがあります。このため、マシンを再起動して、nscd デーモンとそれらのライブラリ関数が確実に最新のスイッチ情報を持つようにする必要があります。

NIS+ クライアントで IPv6 を使用できるようにする方法

1. スーパーユーザーとしてログインします。

2. /etc/nsswitch.conf ファイルを編集します。

3. 新しい ipnodes ソースを追加して、ネームサービス (ldap など) を指定します。

```
ipnodes: ldap [NOTFOUND=return] files
```

ipnodes は、デフォルトでは files です。IPv4 から IPv6 への変更中には、すべてのネームサービスが IPv6 のアドレスを認識できるわけではないので、デフォルトの files を使用する必要があります。デフォルトを使用しない場合は、アドレスの解決中に不必要な遅延が生じることがあります。

4. ファイルを保存して、マシンを再起動します。

nscd デーモンはこの情報をキャッシュに保存して起動時にこの情報を読み取るので、ここでマシンを再起動する必要があります。

+/- 構文との互換性を確保する

/etc/passwd、/etc/shadow、および /etc/group ファイルに +/- が使用される場合は、互換性を確保するために、nsswitch.conf ファイルを変更する必要があります。

- 「NIS+」。NIS+ で +/- 構文と同じ効果を得るには、passwd ソースと group ソースを compat に変更し、nsswitch.conf ファイルの passwd エントリまたは group エントリの後に passwd_compat: nisplus というエントリを追加します (下記参照)。

```
passwd: compat
passwd_compat: nisplus
group: compat
group_compat: nisplus
```

この指定により、クライアント関数のネットワーク情報獲得先は、ファイル中に +/- エントリを指定したのと同じように、/etc ディレクトリのファイルと NIS+ テーブルになります。

- 「NIS」。SunOS™ 4 と同じ構文を使用するには、passwd ソースと group ソースを compat に変更します。

```
passwd: compat
group: compat
```

この指定により、クライアント関数のネットワーク情報獲得先は、ファイル中に +/- エントリを指定したのと同じように、/etc ファイルと NIS マップになります。

注 - NIS+ サーバーが NIS 互換モードで動作している場合、クライアントマシンでは netgroup テーブルに対して ypcat を実行できません。実行すると、エントリの有無に関わらず「テーブルが空である」という結果が出ます。

スイッチファイルとパスワード情報

files や nisplus など複数のリポジトリ内のパスワード情報全体にアクセスすることが可能です。nsswitch.conf ファイルを使って、この情報の検索順序を設定できます。



注意 - nsswitch.conf ファイルでは、files を passwd 情報の 1 番目のソースにしてください。

NIS+ 環境では、nsswitch.conf ファイルの passwd 行に、以下の順番でリポジトリを指定する必要があります。

```
passwd: files nisplus
```

NIS 環境では、nsswitch.conf ファイルの passwd 行に、以下の順番でリポジトリを指定する必要があります。

```
passwd: files nis
```

ヒント - `files` を先頭に指定することで、大半の状況では (システムでネットワークまたはネームサービスに関連した問題が発生している場合でも) `root` でのログインが許可されます。

同一ユーザーを複数のリポジトリで管理することは、推奨されていません。ユーザーごとに単一のリポジトリ内でパスワードを集中管理することで、混乱やエラーが発生する可能性を低減できます。ユーザーごとに複数のリポジトリを管理する場合は、`passwd -r` コマンドを使用して、パスワード情報を更新してください。

`passwd -r repository`

`-r` オプションとともにリポジトリが指定されていない場合、`passwd` は `nsswitch.conf` に記載されているリポジトリを逆順に更新します。

パート II NIS+ の設定と構成

このパートでは、Solaris OS 上で NIS+ ネームサービスを設定および構成する方法について説明します。

第 2 章

NIS+ の紹介

この章では、NIS+ の概要を述べます。

注 - NIS+ は、将来のリリースではサポートされなくなる可能性があります。NIS+ から LDAP への移行支援ツールは、Solaris 9 リリース以降で使用できます (『Solaris のシステム管理 (ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』を参照)。詳細については、<http://www.sun.com/directory/nisplus/transition.html> を参照してください。

NIS+ について

NIS+ は NIS によく似たネットワークネームサービスですが、より多くの機能を備えています。NIS+ は、NIS の拡張機能ではなく、新しいソフトウェアプログラムです。

NIS+ ネームサービスは、ネットワークがどのような構造であっても、その周囲を取り巻くことにより、サービスを設置した組織の形態に適合するように設計されています。

NIS+ はマシンのアドレス、セキュリティ情報、メール情報、Ethernet インタフェース、ネットワークサービスなどの情報を集中的に格納して、ネットワーク上のすべてのマシンからアクセスできるようにします。このように構成されたネットワーク情報を、NIS+ 「名前空間」と呼びます。

NIS+ 名前空間は階層構造となっていて、UNIX® のディレクトリファイルシステムによく似ています。階層構造になっていることから、NIS+ 名前空間を企業組織の階層に合わせて構成できます。NIS+ 名前空間は、独立して管理できる複数のドメインに分割できます。クライアントは、適切なアクセス権があれば、自分のドメインだけではなく、ほかのドメインの情報にもアクセスできます。

NIS+ はクライアントサーバーモデルを使用して、NIS+ 名前空間に情報を格納し、またその情報にアクセスできます。各ドメインは複数のサーバーによってサポートされます。もっとも重要なサーバーは「マスター」サーバーと呼ばれ、バックアップサーバーは「複製」サーバーと呼ばれます。ネットワーク情報は、内部 NIS+ データベース内にある 16 個の標準 NIS+ テーブルに格納されています。マスターサーバーと複製サーバーは、両方とも NIS+ のサーバーソフトウェアを実行し、NIS+ テーブルのコピーを保持します。マスターサーバー上の NIS+ データの変更は、複製サーバーにも自動的に伝達されます。

NIS+ には、名前空間の構造とその情報を保護するために、高度なセキュリティシステムが組み込まれています。NIS+ は認証 (authentication) と承認 (authorization) を使用して、クライアントの情報要求に応えるべきかどうかを検証します。「認証」とは、情報の要求者がネットワークの正当なユーザーであるかどうかを判定することです。「承認」とは、要求された情報に関して特定のユーザーが入手または変更を許可されているかどうかを判定することです。

Solaris のクライアントは、ネームサービススイッチ (/etc/nsswitch.conf ファイル) を使用して、マシンがどこからネットワーク情報を取り出すかを決定します。この種の情報はローカル側の /etc ファイルや、NIS、DNS、NIS+ に格納されます。ネームサービススイッチでは、情報の種類ごとに異なるソースを指定できます。このスイッチの詳細は、第 1 章を参照してください。

NIS+ のメリット

NIS と比較すると、NIS+ には次のようなメリットがあります。

- 安全性の高いデータアクセス
- 階層構造を使用した分散型のネットワーク管理
- 非常に大きな名前空間の管理
- 異なるドメインにあるリソースへのアクセス
- 増分 (incremental) 更新

50 ページの「NIS+ のセキュリティ」で説明したセキュリティシステムを使用すれば、特定のテーブルの個々のエントリに対する特定のユーザーのアクセスを制御できます。このようなセキュリティ方式では、システムを安全に維持でき、しかも NIS+ 名前空間全体またはテーブル全体が損傷する危険性がなく、管理業務を広く分散させることができます。

NIS+ の階層構造により、使用する名前空間に複数のドメインを置くことができます。ドメインに分割することにより、管理が容易になります。個々のドメインは完全に独立して管理できるため、システム管理者の負担が軽減され、非常に大きな名前空間の管理責任から解放されます。このように、セキュリティシステムを分散型のネットワーク管理と組み合わせることによって、管理作業の負担を分担することができます。

ドメインが個別に管理されているとしても、すべてのクライアントに名前空間内のすべてのドメインの情報を参照するアクセス権を与えることができます。クライアントは自分のドメイン内のテーブルしか見ることができないため、クライアントがほかのドメイン内のテーブルにアクセスするには、そのテーブルについて明示的にアドレスを指定しなければなりません。

増分更新とは、名前空間内での情報の高速の更新を意味します。ドメインは独立して管理されるため、マスターサーバーテーブルへの変更は、名前空間全体ではなく、その複製サーバーだけに伝達します。伝達が行われると、これらの変更はすぐに名前空間全体で有効になります。

NIS+ と NIS の違い

ネットワーク情報サービスプラス (NIS+) は、ネットワーク情報サービス (NIS) といくつかの点で異なります。NIS+ には、多くの新しい機能が追加されています。また、NIS と似た概念に対して、使用する用語が異なることがあります。わからない用語がある場合は、用語集を参照してください。次の表に、NIS と NIS+ の主な相違点を示します。

表 2-1 NIS と NIS+ の違い

NIS	NIS+
フラットなドメイン - 階層なし	階層構造 - データを名前空間内の異なるレベルに格納
データを 2 列のマップに格納	データを複数の列のテーブルに格納
認証を使用しない	DES 認証を使用
ネットワーク情報源は 1 つ	ネームサービススイッチ - クライアントは NIS、NIS+、DNS、またはローカル側の /etc ファイルから情報源を選択できる
パッチ伝達のため更新は遅い	すぐに伝達される増分更新

NIS+ は NIS に代わるものとして設計されました。NIS は、1980 年代に普及していたクライアントサーバーコンピューティングネットワークの管理要件に応えるものです。その当時のクライアントサーバーネットワークは、通常、クライアント数が数百を超えることはなく、多目的サーバーの数もわずかでした。これらのネットワークは数カ所のリモートサイトを結んでいるだけであり、しかもユーザーに専門知識があり信頼できたため、セキュリティを必要としませんでした。

しかしクライアントサーバーネットワークは 1980 年代半ば頃から急速な成長を遂げました。現在では、世界中のサイトに配置された 10~100 台の専用サーバーにサポートされた 100~10,000 台のマルチベンダークライアントが存在し、複数の公衆網に接

続されています。さらに、ネットワークが格納する情報は、NIS の時代よりもはるかに急速に変化しています。このようなネットワークの規模と複雑性に対処するため、新しい管理方式が必要になりました。NIS+ はこれらの必要性に焦点をあてて設計されました。

NIS の名前空間は、フラットな状態で管理機能を集中管理しています。1990 年代に入りネットワークはスケーラビリティと管理の分散化を求めるようになったため、NIS+ の名前空間は DNS の場合のように階層ドメインをベースに設計されました。

たとえば図 2-1 は doc という名前の親ドメインと、sales と manf という 2 つのサブドメインを持つ会社の例を示しています。

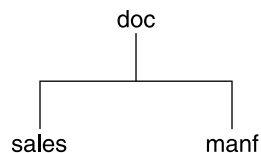


図 2-1 階層型ドメインの例

これによって NIS+ は、小規模から大規模まで、広い範囲のネットワークで使用できます。また、NIS+ のサービスを組織の成長に適合させることもできます。たとえば、ある会社が 2 つの部門に分かれた場合、それに対する NIS+ の名前空間を 2 つのドメインに分割し、これらを自律的に管理できます。インターネットがドメインの管理を下のレベルに委譲するように、NIS+ のドメインも程度の差はあっても互いに独立して管理できます。

NIS+ は DNS と似たドメイン階層を使用しますが、NIS+ のドメインは DNS のドメインよりずっと多くの情報を持っています。DNS のドメインは、そのクライアントの名前とアドレスの情報を格納するだけです。一方 NIS+ のドメインには、組織の一部の中でのマシン、ユーザー、およびネットワークサービスについての「情報」が集められています。

ドメインをこのように分割することで、管理はより自律的になり、規模が拡大した場合にもうまく対処できますが、情報へのアクセスが以前より困難になることもあります。クライアントはほかのドメインの情報にも、同じドメインと同じようにアクセスできます。あるドメインを別のドメインの内部から管理することもできます。

主 NIS+ サーバーは「マスター」サーバーと呼ばれ、バックアップサーバーは「複製」サーバーと呼ばれます。マスターサーバーと複製サーバーは、両方とも NIS+ のサーバーソフトウェアを実行し、NIS+ テーブルのコピーを保持します。NIS の情報はマップに格納されますが、NIS+ の情報はテーブルに格納されます。主サーバーはオリジナルのテーブルを、バックアップサーバーはコピーを、それぞれ格納します。

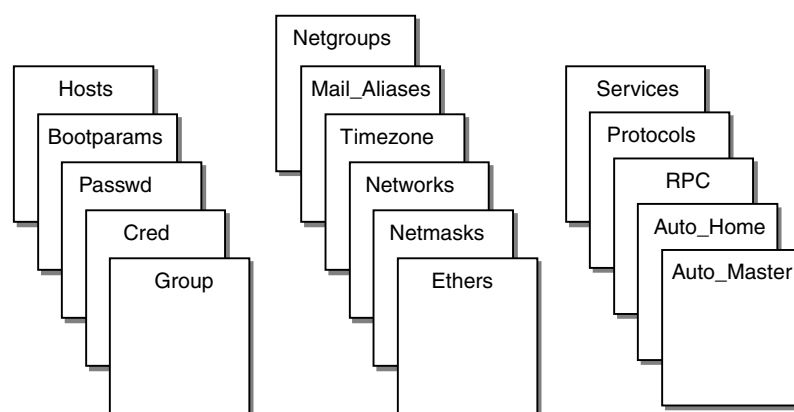
しかし NIS+ は、NIS とはまったく異なる更新方式を使用します。NIS が開発された当時は、格納される情報の型はめったに変化しなかったため、NIS は安定性に重点を置いた更新方式によって開発されました。NIS テーブルの更新は手作業で処理され、大規模な組織では、すべての複製サーバーへの伝達に 1 日以上かかることもあります。この原因の一つは、マップ内の情報が変化するたびにマップ全体を再作成して伝達しなければならなかったからです。

しかし NIS+ は、複製サーバーに対する「増分 (incremental)」更新が可能です。マスターサーバー上での変更は依然として必要ですが、いったん変更すれば、複製サーバーに自動的に伝達され、すぐに名前空間全体から使用できるようになります。マップを「作成」したり、伝達を待つ必要はありません。

NIS+ のドメイン構造、サーバー、およびクライアントの詳細については、それぞれ 60 ページの「NIS+ のドメイン」、61 ページの「NIS+ サーバー」、および 65 ページの「NIS+ 主体 (クライアント)」を参照してください。

NIS+ のドメインは、ネームサービススイッチを使用して、その NIS+ クライアントを経由してインターネットに接続できます (例 1-1 を参照)。このクライアントが DNS のクライアントでもある場合、自分のスイッチ構成ファイルを設定して、NIS+ テーブルだけでなく、DNS ゾーンファイルまたは NIS マップ内の情報を検索できます。

NIS+ は、マップやゾーンファイルではなく、「テーブル」に情報を格納します。NIS+ は 16 種類のあらかじめ定義したテーブル (つまり「システム」テーブル) を提供します。



各テーブルにはそれぞれ異なる情報が格納されています。たとえば、hosts テーブルにはマシンアドレスについての情報が、passwd テーブルにはネットワークのユーザーについての情報が格納されています。

NIS+ のテーブルは、NIS で使用されるマップと比較して 2 つの点で大きく改良されています。第 1 に、NIS+ のテーブルは、最初の列 (キーと呼ばれることもある) だけでなく、任意の列ごとにアクセスできます。これによって、NIS で使用される hosts.byname や hosts.byaddr マップなどの二重マップが必要なくなります。第 2 に、NIS+ のテーブル内の情報は、3 つの細分化されたレベルでアクセスし、操作できます。テーブルレベル、エントリレベル、および列レベルです。NIS+ のテーブルとそこに格納されている情報については、第 10 章で説明します。

NIS 管理者は、以下に示す原則および条件下で NIS を NIS+ とともに使用できます。

- 同一ドメインに NIS サーバーと NIS+ サーバーの両方が存在する。NIS 管理者は、同一ドメインで NIS サーバーと NIS+ サーバーの両方を動作させることは可能ですが、このような動作を長時間行わせることは望ましくありません。一般に、同一ドメイン内での NIS サーバーと NIS+ サーバーを両方使用するのには、NIS から NIS+ への短い移行期間だけに制限する必要があります。
- サブドメイン。NIS 管理者のルートドメインのマスターサーバーで NIS+ が動作している場合は、NIS 管理者は、すべてのサーバーで NIS が動作しているサブドメインを設定できます。NIS 管理者のルートドメインのマスターサーバーで NIS が動作している場合は、NIS 管理者はサブドメインの設定はできません。
- 同一ドメインに複数のマシンが存在する。同一ドメイン内のサーバーで NIS+ が動作している場合は、NIS+、NIS、または /etc ファイルを使ってネームサービス情報を取得するように、ドメイン内の各マシンを設定できます。NIS+ サーバーが NIS クライアントのニーズを満たすには、この NIS+ サーバーが NIS 互換モードで動作する必要があります。

同一ドメイン内のサーバーで NIS が動作している場合は、NIS または /etc ファイルを使ってネームサービス情報を取得するように、このドメイン内の各マシンを設定できます (各マシンは NIS+ を使用することはできません)。

さまざまなネームサービス情報を取得するためにマシンが使用するサービスは、そのマシンの nsswitch.conf ファイルで制御されます。このファイルは、「スイッチ」ファイルと呼ばれています。詳細は、[第 1 章](#)を参照してください。

NIS+ のセキュリティ

NIS+ は、「承認と認証」という相補的なプロセスによって、名前空間の構造と格納する情報を保護します。

- 「承認 (authorization)」。名前空間の各構成要素は、「だれからのどのような操作を許可するか」を規定しています。これが承認です。
- 「認証 (authentication)」。名前空間に対するアクセス要求はすべて、NIS+ によって「認証」されます。アクセス要求は、NIS+ 主体から発行されます。NIS+ 主体には、プロセス、マシン、ルート、またはユーザーになることができます。正当な NIS+ 主体は、NIS+ 「資格」を持ちます。名前空間へのアクセスが行われると、必ずその要求者 (主体) の認証が行われます。

NIS+ が要求どおりの動作をするのは、「主体が認証されていて (正当な資格を持っていて)、要求された操作が、該当する構成要素において承認されている」という場合のみです。資格がない (または完全でない)、要求された操作が承認されていないという場合、NIS+ はアクセス要求を拒否します。NIS+ セキュリティシステム全体の概要については、[第 11 章](#)を参照してください。

Solaris 1.x と NIS 互換モード

Solaris 1.x または 2.x では、NIS+ を NIS が動作しているマシンで使用できます。つまり、NIS+ ドメイン内にあるマシンはそれぞれの `nsswitch.conf` ファイルを `nisplus` ではなく `nis` に設定できます。NIS を実行中のマシン上で NIS+ のサービスにアクセスする場合は、NIS+ のサーバーを「NIS 互換モード」で動作させる必要があります。

NIS 互換モードを使用すれば、Solaris ソフトウェアを実行している NIS+ サーバーは、NIS+ クライアントからの要求に応答しながら、NIS クライアントからの要求にも応答できます。NIS+ は 2 つのサービスインタフェースを提供することによってこれを実現します。1 つが NIS+ クライアントの要求に応答し、もう 1 つが NIS クライアントの要求に応答します。

このモードでは、NIS クライアントに対してさらに設定や変更を行う必要はありません。実際、NIS クライアントは、応答しているサーバーが NIS サーバーではないことを意識する必要はありません。ただし、NIS 互換モードで動作している NIS+ サーバーは `ypupdate` と `ypxfr` のプロトコルをサポートしないため、複製またはマスターの NIS サーバーとしては使用できません。NIS 互換モードの詳細については、『Solaris のシステム管理 (ネーミングとディレクトリサービス: NIS+ 編)』の「NIS+ サーバーの設定」を参照してください。

さらに 2 つの相違を指摘しておく必要があります。1 つは、NIS 互換モードでサーバーを設定する命令が標準の NIS+ サーバーの設定に使用される命令とは少し異なるということです。もう 1 つの相違としては、NIS 互換モードは、NIS+ の名前空間内のテーブルに対するセキュリティにも関係があります。NIS のクライアントソフトウェアは、NIS+ サーバーが NIS+ クライアントに与える資格 (credential) を提供する機能がないため、NIS クライアントからの要求はすべて「未認証」として分類されます。したがって、NIS クライアントが NIS+ テーブル内の情報にアクセスできるように、NIS+ の名前空間内のテーブルは未認証の要求に対してアクセス権を提供しなければなりません。これはパート II で説明するように、サーバーを NIS 互換モードで設定するために使用されるユーティリティによって自動的に処理されます。認証プロセスと NIS 互換モードについては、『Solaris のシステム管理 (ネーミングとディレクトリサービス: NIS+ 編)』の「NIS+ のセキュリティの概要」を参照してください。

再起動を繰り返しても NIS 互換モードが持続するように設定する場合は、[89 ページ](#)の「NIS+ とサービス管理機能」の説明に従って、`/lib/svc/method/nisplus` ファイルを変更する必要があります。

NIS+ の管理コマンド

NIS+ では、名前空間を管理するのに必要なコマンドがすべて提供されています。次の表に、これらのコマンドについてまとめます。

注 – NIS+ サービスに関連したコマンド行管理タスクのほとんどは、サービス管理機能 (Service Management Facility、SMF) によって管理されます。SMF を NIS+ で使用する方法については、89 ページの「NIS+ とサービス管理機能」を参照してください。SMF の概要については、『Solaris のシステム管理 (基本編)』の「サービスの管理 (概要)」を参照してください。詳細については、svcadm(1M) と svcs(1) の各マニュアルページも参照してください。

表 2-2 NIS+ の名前空間管理コマンド

コマンド	説明
<code>nisaddcred</code>	NIS+ 主体用の資格を作成し、これらを cred テーブルに格納する
<code>nisaddent</code>	/etc 内のファイルまたは NIS マップからの情報を NIS+ のテーブルに追加する
<code>nisauthconf</code>	オプションで Diffie - Hellman 鍵の長さを設定する
<code>nisbackup</code>	NIS ディレクトリのバックアップコピーを取る
<code>nis_cachemgr</code>	NIS+ クライアント上で NIS+ キャッシュマネージャを起動する
<code>niscat</code>	NIS+ テーブルの内容を表示する
<code>nis_checkpoint</code>	ログには入力されたが、ディスクにチェックポイントが実行されていないデータについて強制的にチェックポイントを実行する
<code>nischgrp</code>	NIS+ オブジェクトのグループ所有者を変更する
<code>nischmod</code>	オブジェクトのアクセス権を変更する
<code>nischown</code>	NIS+ オブジェクトの所有者を変更する
<code>nischttl</code>	NIS+ オブジェクトの生存期間の値を変更する
<code>nisclient</code>	NIS+ 主体を初期化する
<code>nisdefaults</code>	NIS+ オブジェクトのデフォルト値 (ドメイン名、グループ名、マシン名、NIS+ 主体名、アクセス権、ディレクトリ検索パス、および生存期間) を表示する
<code>nisgrep</code>	NIS+ テーブル内のエントリを検索する

表 2-2 NIS+ の名前空間管理コマンド (続き)

コマンド	説明
<code>nisgrpadm</code>	NIS+ グループの作成または削除、あるいはそのメンバーリストを表示する。また、グループへのメンバー追加、メンバー削除、またはグループメンバーかどうかのテストを行う
<code>nisinit</code>	NIS+ のクライアントまたはサーバーを初期設定する
<code>nisln</code>	2 つの NIS+ オブジェクト間でシンボリックリンクを作成する
<code>nislog</code>	NIS+ 処理用ログの内容を表示する
<code>nisls</code>	NIS+ ディレクトリの内容を表示する
<code>nismatch</code>	NIS+ テーブル内のエントリを検索する
<code>nismkdir</code>	NIS+ のディレクトリを作成し、そのマスターサーバーと複製サーバーを指定する
<code>nispasswd</code>	NIS+ の <code>passwd</code> テーブルに保管されているパスワード情報を変更する (<code>nispasswd</code> ではなく、 <code>passwd</code> または <code>passwd -r nisplus</code> を使用する)
<code>nis_ping</code>	複製サーバーのデータをマスターサーバーのデータに強制更新する
<code>nispopulate</code>	新しい NIS+ ドメインに NIS+ テーブルを生成する
<code>nisprefadm</code>	クライアントが NIS+ サーバーから NIS+ 情報を検索する順序を指定する
<code>nisrestore</code>	以前にバックアップを取った NIS+ ディレクトリを復元する。新しい NIS+ 複製サーバーを急速にオンラインにするときにも使用する
<code>nismrm</code>	名前空間から NIS+ のオブジェクト (ディレクトリを除く) を削除する
<code>nismrmdir</code>	名前空間から NIS+ のディレクトリと複製を削除する
<code>nisserver</code>	新しい NIS+ サーバーを設定するときに使うシェルスクリプト
<code>nissetup</code>	<code>org_dir</code> ディレクトリと <code>groups_dir</code> ディレクトリ、および NIS+ ドメイン用の完全セットの (未生成) NIS+ テーブルを作成する
<code>nisshowcache</code>	NIS+ キャッシュマネージャによって管理される NIS+ 共有キャッシュの内容を表示する
<code>nisstat</code>	NIS+ サーバーに関する統計やその他の情報を報告する
<code>nistbladm</code>	NIS+ テーブルの作成または削除と、NIS+ テーブル内のエントリの追加、修正、または削除を行う
<code>nistest</code>	NIS+ 名前空間の現在の状態を報告する

表 2-2 NIS+ の名前空間管理コマンド (続き)

コマンド	説明
<code>nissupdkeys</code>	NIS+ オブジェクトに格納されている公開鍵を更新する
<code>passwd</code>	NIS+ の <code>passwd</code> テーブルに保管されているパスワード情報を変更するまたパスワードの経過時間やその他のパスワード関連のパラメータを管理する

NIS+ の API

NIS+ の API (アプリケーションプログラミングインタフェース) とは、アプリケーションが NIS+ のオブジェクトへのアクセスと修正を行うために呼び出す関数群です。NIS+ の API には 54 個の関数があり、それらは次の 9 つのグループに分類されます。

- オブジェクト操作関数 (`nis_names()`)
- テーブルアクセス関数 (`nis_tables()`)
- ローカル名関数 (`nis_local_names()`)
- グループ操作関数 (`nis_groups()`)
- アプリケーションサブルーチン関数 (`nis_subr()`)
- その他の関数 (`nis_misc()`)
- データベースアクセス関数 (`nis_db()`)
- エラーメッセージ表示関数 (`nis_error()`)
- トランザクションログ関数 (`nis_admin()`)

設定および構成の前に

NIS+ 名前空間を構成する前に、次の作業を行う必要があります。

- NIS+ を使用するすべてのマシンの `nsswitch.conf` ファイルを正しく構成します。詳細は、[第 1 章](#)を参照してください。
- NIS+ 名前空間のレイアウトを設計します。レイアウトを設計する際、次の点を考慮してください。
 - 名前空間の設計。どんなドメイン名を付けるか。サブドメインを作るか。サブドメインを作るとしたら、それらをどのように編成するか。どのマシンをどのドメインに入れるか。ドメインを上位のドメインあるいはインターネットに接続するか。
 - サーバー要件の確認。各ドメインに複製サーバーをいくつ置くか。必要なサーバーの種類、処理速度、メモリー容量はサーバーのディスク容量はどれくらい必要か。

上記およびほかの設計上の考慮事項、および推奨ガイドラインの詳細については本書をよくお読みください。

- 名前空間がすでに存在している場合、NIS+ に移行するための準備をします。79 ページの「名前空間がすでに存在する場合の設定」を参照してください。
- ルートサーバーマシンを選択します。
- ルートマスターサーバーとして使用できるシステムが少なくとも 1 つは稼働していることを確認します。ルートマスターサーバーでは、`/etc/passwd` などのシステム情報ファイルに少なくとも 1 人のユーザー (root) が登録されていなければなりません。通常のマシンのシステム情報ファイルには root が存在しますので、これが問題となることはないはずです。

NIS と NIS+

NIS と NIS+ は、いくつかの同じ作業を行います。ただし NIS+ では、NIS で提供されない階層ドメイン、名前空間セキュリティなどの機能が使用できます。NIS と NIS+ の相違点の詳細は、「NIS+ と NIS の違い」を参照してください。

NIS 管理者は、以下に示す原則および条件下で NIS を NIS+ とともに使用できます。

- 同一ドメインに NIS サーバーと NIS+ サーバーの両方が存在する。NIS 管理者は、同一ドメインで NIS サーバーと NIS+ サーバーの両方を動作させることは可能ですが、このような動作を長時間行わせることは望ましくありません。一般に、同一ドメイン内での NIS サーバーと NIS+ サーバーを両方使用するのには、NIS から NIS+ への短い移行期間だけに制限するべきです。クライアントから NIS サービスの要求があった場合に、NIS 管理者は、NIS+ を NIS 互換モードで動作させることができます。詳細は、51 ページの「Solaris 1.x と NIS 互換モード」を参照してください。
- サブドメイン。NIS 管理者のルートドメインのマスターサーバーで NIS+ が動作している場合は、NIS 管理者は、すべてのサーバーで NIS が動作しているサブドメインを設定できます。NIS 管理者のルートドメインのマスターサーバーで NIS が動作している場合は、NIS 管理者はサブドメインの設定はできません。この操作は、NIS 管理者が NIS から NIS+ に移行する際に便利な場合があります。たとえば、互いに独立した複数の NIS ドメイン (ドメインは地理的に別々のサイトに置かれていることもある) を持った会社の中で、NIS 管理者がそれらのドメインをすべてリンクさせて NIS+ の下に 1 つの階層型マルチドメイン名前空間を構築するケースを考えてみます。この場合、NIS 管理者はまずルートドメインを NIS+ 下に設定し、次に従来の NIS ドメインをサブドメインとして指定できます。これらのサブドメインでは、NIS+ へ切り換える準備ができるまで NIS が継続して動作します。
- 同一ドメインに複数のマシンが存在する。
 - 同一ドメイン内のサーバーで NIS+ が動作している場合は、NIS+、NIS、または `/etc` ファイルを使ってネームサービス情報を取得するように、ドメイン内の各マシンを設定できます。NIS+ サーバーが NIS クライアントのニーズを満たすには、この NIS+ サーバーが NIS 互換モードで動作している必要があります。51 ページの「Solaris 1.x と NIS 互換モード」を参照してください。

- 同一ドメイン内のサーバーで NIS が動作している場合は、NIS または `/etc` ファイルを使ってネームサービス情報を取得するように、このドメイン内の各マシンを設定できます (各マシンは NIS+ を使用することはできません)。

NIS+ のファイルとディレクトリ

表 2-3 は、NIS+ ファイルが格納される UNIX ディレクトリの一覧です。

表 2-3 NIS+ ファイルが存在する場所

ディレクトリ	存在するマシン	内容
<code>/usr/bin</code>	すべて	NIS+ ユーザーコマンド
<code>/usr/lib/nis</code>	すべて	NIS+ 管理者コマンド
<code>/usr/sbin</code>	すべて	NIS+ デーモン
<code>/usr/lib/</code>	すべて	NIS+ 共有ライブラリ
<code>/var/nis/data</code>	NIS+ サーバー	サーバーの使用できるデータファイル
<code>/var/nis</code>	NIS+ サーバー	NIS+ ワーキングファイル
<code>/var/nis</code>	NIS+ クライアントマシン	NIS+ の使用するマシン固有のデータ



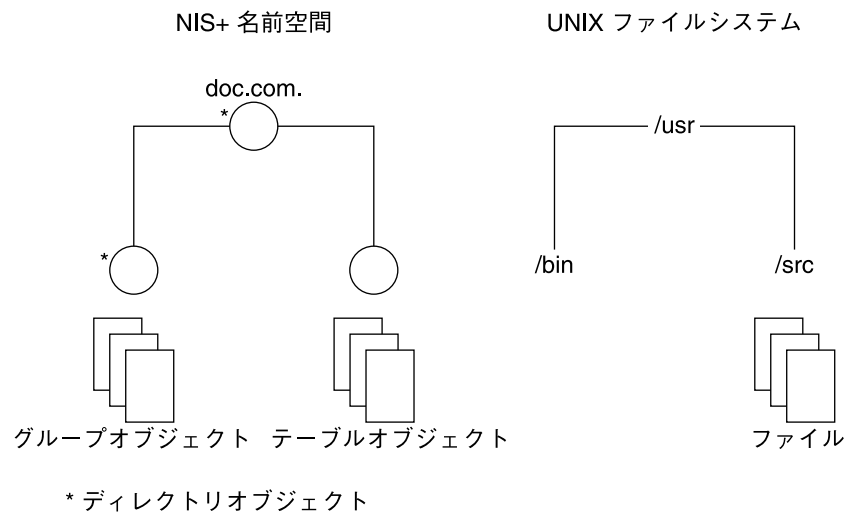
注意 - `nisinit` など NIS+ 設定プロシージャによって作成された `/var/nis`、`/var/nis/data` といったディレクトリ、およびその下のファイルは、名前を変更しないでください。Solaris 2.4 以前では、`/var/nis` ディレクトリに `hostname.dict`、`hostname.log` という 2 つのファイルが含まれていました。またサブディレクトリ `/var/nis/hostname` もありました。Solaris 2.5 を起動すると、2 つのファイル名は `trans.log`、`data.dict` となり、サブディレクトリ名は `/var/nis/data` となります。ファイルの「内容」も変更されており、Solaris 2.4 以前との互換性はなくなっています。したがって、これらのファイルやディレクトリを Solaris 2.4 での名前にしてしまうと、Solaris 2.4 と現行バージョンのどちらの `rpc.nisd` でも機能しなくなるので名前の変更をしないようにしてください。ディレクトリ名もファイル名も変更しないでください。

注 – Solaris プラットフォームでは、NIS+ データディレクトリ (/var/nis/data.dict) はマシンに依存しません。そのため、NIS+ 名を簡単に変更することができます。また、NIS+ のバックアップコピーを使用したり機能を復元したりして、NIS+ のデータをひとつのサーバーから別のサーバーに転送もできます。[第 21 章](#)を参照してください。

NIS+ 名前空間の構造

NIS+ の名前空間は、NIS+ によって格納された情報を配置したものです。この名前空間は、組織のニーズに合わせていろいろな方法で配置することができます。たとえば、3 つの部門を持つ組織の場合、その NIS+ の名前空間も各部門ごとに 1 つずつで 3 つの部分に分割されます。分割した各部分は、その部門のユーザー、マシン、およびネットワークサービスについての情報を格納しますが、互いに通信することも簡単です。このような配置により、ユーザーによる情報へのアクセスや、管理者による情報の管理が更に容易に行えます。

NIS+ の名前空間の配置はサイトごとに異なりますが、ディレクトリ、テーブル、グループという構成はすべてのサイトが使用します。これらの構成は NIS+ 「オブジェクト」と呼ばれます。NIS+ オブジェクトは、UNIX のファイルシステムに似た階層形式に配置できます。たとえば、次の図を参照してください。左側の名前空間は、3 つのディレクトリオブジェクト、3 つのグループオブジェクト、および 3 つのテーブルオブジェクトから構成されています。右側の UNIX ファイルシステムは、3 つのディレクトリと 3 つのファイルから構成されています。

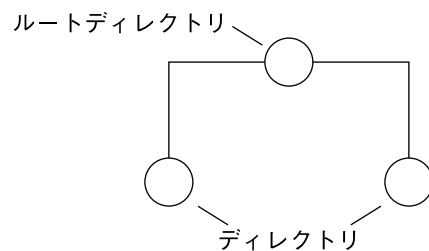


NIS+ の名前空間は UNIX ファイルシステムに似ていますが、重要な違いが5つあります。

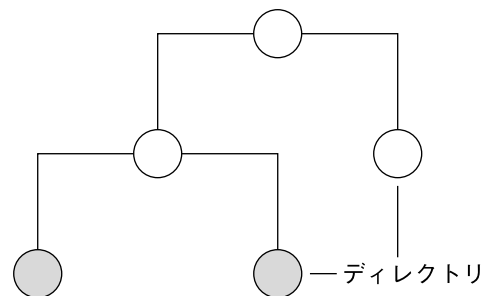
- 両方ともディレクトリを使用するが、NIS+ の名前空間でのオブジェクトはテーブルとグループであり、ファイルではない
- NIS+ の名前空間は、NIS+ の管理コマンド、または管理用に設計された Solaris 管理コンソール™ ツールなどのグラフィカルユーザーインターフェース (GUI) だけで管理され、UNIX 標準のファイルシステムコマンドや GUI では管理できない
- UNIX ファイルシステム構成要素の名前はスラッシュで区切られる (/usr/bin) が、NIS+ の名前空間オブジェクト名はドットで区切られる (doc.com.)
- UNIX ファイルシステムの「ルート」へは、ディレクトリを右から左にたどっていけば到達する (/usr/src/file1) が、NIS+ の名前空間のルートは左から右にたどって到達する (sales.doc.com.)
- NIS+ オブジェクト名は、左から右に構成されているので、完全指定名は常にドットで終わる。また最後に「.」がないものは完全な名前とはみなされず、相対的な名前 (まだ上の階層があるという意味) であるとみなされる

NIS+ 名前空間のディレクトリ

ディレクトリオブジェクトは、名前空間の骨格を形成します。ツリー構造に配置した場合、名前空間を別々に分けます。ディレクトリ階層を理解するには、木の根を一番上に置き、逆さまの木として見るとよく分かります。名前空間の一番上のディレクトリが「ルート」ディレクトリです。名前空間が1つの層だけの場合、ディレクトリは1つしかありませんが、そのディレクトリはルートディレクトリです。ルートディレクトリの下にあるディレクトリオブジェクトは、単に「ディレクトリ」と呼ばれます。



1つの名前空間には、複数の階層のディレクトリを割り当てることができます。



ディレクトリ間の関係を識別する場合には、下位ディレクトリは「子」ディレクトリと呼ばれ、上位ディレクトリは「親」ディレクトリと呼ばれます。

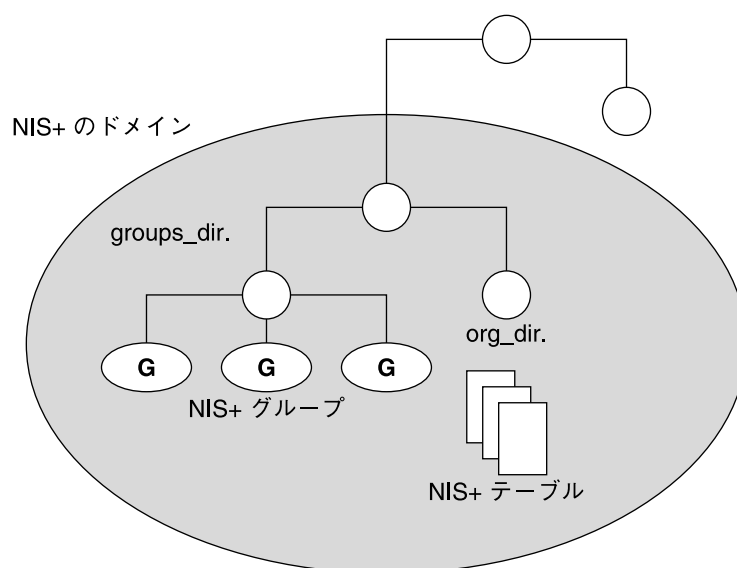
UNIX のディレクトリは UNIX のファイルを格納するように設計されていますが、NIS+ のディレクトリは NIS+ オブジェクト (ほかのディレクトリ、テーブル、およびグループ) を格納するように設計されています。各 NIS+ のドメインレベルのディレクトリには、以下のサブディレクトリが含まれています。

- NIS+ グループ情報を格納する
- `org_dir.` - NIS+ システムテーブルを格納する

技術的には、ユーザーはディレクトリ、テーブル、およびグループをどんな構造にでも配置することができます。しかし、名前空間内の NIS+ のディレクトリ、テーブル、およびグループは、「ドメイン」と呼ばれる構成に配置されるのが普通です。ドメインは、名前空間の別々の部分をサポートするよう設計されています。たとえば、あるドメインが会社の営業部門 (Sales Division) をサポートし、別のドメインが製造部門 (Manufacturing Division) をサポートできます。

NIS+ のドメイン

1 つの NIS+ のドメインは、ディレクトリオブジェクト、その `org_dir` ディレクトリ、その `groups_dir` ディレクトリ、および NIS+ テーブルのセットから構成されます。



NIS+ のドメインは、実際に存在する名前空間の構成要素ではありません。これらは単に、現実の組織をサポートするために使用される名前空間の「一部分」を示すための便利な方法にすぎません。

たとえば、DOC 社に営業部門と製造部門があるとします。これらの部門をサポートするため、DOC 社の NIS+ 名前空間は、以下の図のような構造によって、3 つの主要ディレクトリグループに配置されることになります。

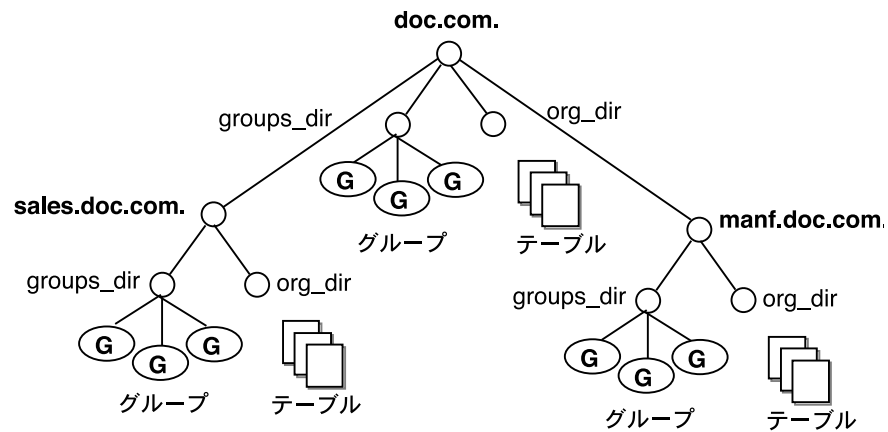


図 2-2 NIS+ ディレクトリ構造の例

このような構造を、3つのディレクトリ、6つのサブディレクトリ、およびいくつかの追加オブジェクトと表現するよりも、3つの NIS+ ドメインと表現する方が便利です。

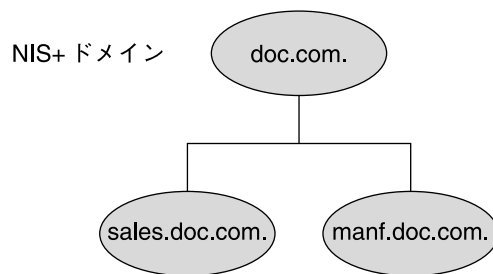
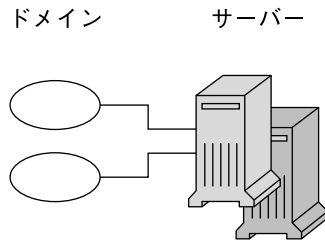


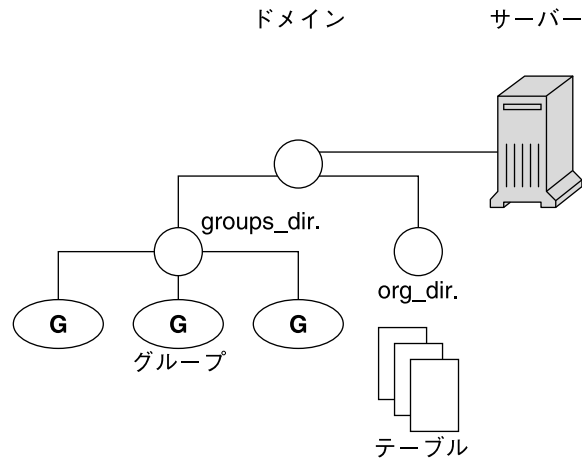
図 2-3 NIS+ ドメインの例

NIS+ サーバー

すべての NIS+ ドメインは複数の NIS+ 「サーバー」によってサポートされます。サーバーは、ドメインのディレクトリ、グループ、およびテーブルを格納し、ユーザー、管理者、およびアプリケーションからのアクセス要求に応答します。各ドメインは、1組の複数のサーバーだけによってサポートされます。ただし、この1組のサーバーは複数のドメインをサポートできます。



ドメインはオブジェクトではなく、オブジェクトの集合を意味するものであることを忘れないでください。したがって、ドメインをサポートするサーバーは、実際にはドメインにではなく、ドメインのメインディレクトリに関連付けられています。

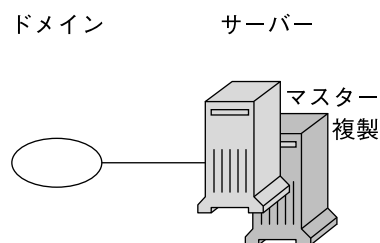


サーバーとディレクトリオブジェクトとこのような接続は、ドメインを設定するときに行われます。ここで、重要なことを1つ指摘しておく必要があります。それは、この接続が確立されたときに、ディレクトリオブジェクトはそのサーバー名とIPアドレスを格納するということです。後述しますが、クライアントはこの情報を使用してサービス要求を送信します。

Solaris プラットフォームベースのマシンは、どれも NIS+ サーバーとなることができます。このリリースには、NIS+ のサーバー用とクライアント用の両方のソフトウェアが入っています。したがって、Solaris 2.x がインストールされているマシンであれば、サーバーにもクライアントにも、あるいはその両方にもなることができます。クライアントとサーバーを区別するのは、その果たす役割です。マシンが NIS+ サービスを提供している場合、それは NIS+ サーバーとして機能しています。同じマシンが NIS+ サービスを要求している場合、これは NIS+ クライアントとして機能しています。

多数のクライアント要求にサービスを提供する必要があるため、NIS+ サーバーとして機能するマシンは、平均的なクライアントよりも高いコンピューティング能力と多くのメモリーを装備して構成されます。そして、NIS+ データを格納する必要があるため、より大型のディスクも装備することになります。しかし、性能を高めるためのハードウェアを除いては、サーバーと NIS+ クライアントとの本質的な違いはありません。

NIS+ ドメインをサポートするのは、マスターとその複製の 2 種類のサーバーです。



ルートドメインのマスターサーバーを「ルートマスター」サーバーと呼びます。1 つの名前空間には 1 つのルートマスターサーバーしか存在しません。ほかのドメインのマスターサーバーは、単にマスターサーバーと呼びます。同様に、ルート複製サーバーと単なる複製サーバーがあります。

マスターサーバーと複製サーバーは、両方とも NIS+ テーブルを格納し、クライアント要求に応答します。ただし、マスターサーバーはドメインのテーブルのマスターコピーを格納します。複製サーバーは複製だけを格納します。管理者は、情報をマスターサーバー内のテーブルにロードし、マスターサーバーはそれを複製サーバーに伝達します。

この配置には 2 つのメリットがあります。第 1 に、マスターテーブルが 1 組しか存在しないことから、テーブル間の不一致を避けることができます。複製サーバーに格納されたテーブルは、マスターのコピーにすぎません。第 2 に、これによって NIS+ サービスの「信頼性」が大幅に向上します。マスターまたはスレーブのどちらかがダウンした場合、ほかのサーバーがバックアップとして機能し、サービス要求に応えることができます。

サーバーが変更を伝達する方法

NIS+ のマスターサーバーは、そのオブジェクトをすぐに更新します。しかし、更新内容を複製サーバーに伝達する前に、複数の更新をまとめ (バッチ) ようと試みます。マスターサーバーが、ディレクトリ、グループ、リンク、またはテーブルといったオブジェクトへの更新を受信した場合、約 2 分間ほかの更新が到着しないかどうかを待機して確認します。待機時間が終了すると、これらの更新をディスクと「トランザクションログ」の 2 カ所に格納します (メモリーにはすでに更新を格納)。

トランザクションログ

NIS+ 主体 (クライアント)

NIS+ 主体とは、NIS+ サービスに要求をするエンティティ (クライアント) のことです。

NIS+ 主体

一般ユーザーであってもスーパーユーザー (root) であってもログインをすれば NIS+ 主体になります。実際の要求は、前者の場合はクライアントユーザーから、後者の場合はクライアントマシンから送られます。つまり NIS+ 主体は、クライアントユーザーである場合と、クライアントマシンである場合とがあります。

また NIS+ サーバーから NIS+ サービスを提供するエンティティも、NIS+ 主体になることができます。すべての NIS+ サーバーは NIS+ クライアントにもなるので、サーバーが NIS+ の主体となる場合もあります。

NIS+ クライアント

NIS+ クライアントとは、NIS+ サービスを要求するように設定されたマシンです。NIS+ クライアントの設定では、セキュリティ資格 (credential) を設定し、クライアントを適切な NIS+ グループのメンバーとし、そのホームドメインを確認し、スイッチ構成ファイルを確認し、最後に NIS+ 初期設定スクリプトを実行します (詳細は、パート II 「NIS+ の紹介と概要」を参照)。

NIS+ クライアントは、セキュリティ上の制約に従って、名前空間のどの部分にもアクセスできます。つまり、認証されており、しかも適切なアクセス権が与えられている場合、名前空間内のどのドメインの情報やオブジェクトにもアクセスできます。

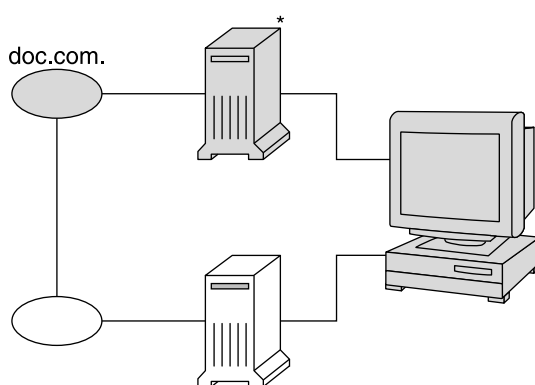
クライアントは名前空間全域にアクセスできますが、1 つのクライアントが所属するドメインは 1 つだけであり、これをその「ホーム」ドメインと呼びます。クライアントのホームドメインは、インストール時に指定されるのが普通ですが、インストール後でも変更または指定できます。クライアントの IP アドレスや資格など、クライアントについてのすべての情報は、そのホームドメインの NIS+ テーブルに格納されています。

NIS+ クライアントであることと、NIS+ テーブルに登録されていることは、微妙な違いがあります。あるマシンについての情報を NIS+ テーブルに入力しても、そのマシンが自動的に NIS+ クライアントになるわけではありません。これは単に、すべての NIS+ クライアントがそのマシンの情報を使用できるようにするだけです。実際に NIS+ クライアントとして設定しない限り、そのマシンから NIS+ サービスを要求することはできません。

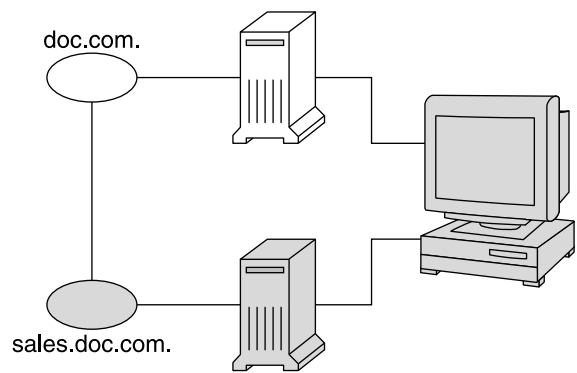
逆に、あるマシンを NIS+ クライアントにしても、そのマシンについての情報は NIS+ テーブルに入力されません。これは単に、そのマシンで NIS+ サービスを受信できるようにするだけです。管理者がそのマシンについての情報を明確に NIS+ テーブルに入力しないと、ほかの NIS+ クライアントはその情報を入手できません。

クライアントが名前空間へのアクセスを要求した場合、実際には、クライアントは名前空間内の特定ドメインへのアクセスを要求していることになります。したがって、クライアントは、アクセスしようとしているドメインをサポートするサーバーに自分の要求を送信します。簡単に表現すると次のようになります。

ドメイン サーバー クライアント



* doc.com ドメイン内のオブジェクトにアクセスする場合、クライアントはこのサーバーによってサポートされる



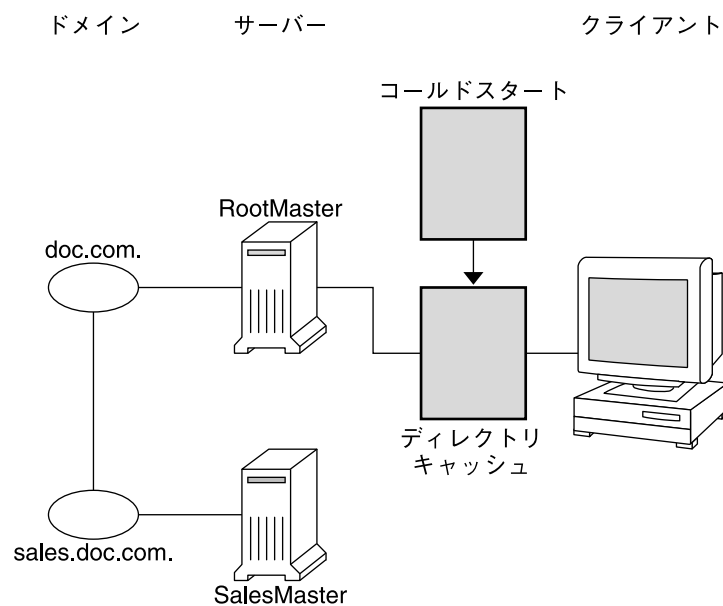
sales.doc.com ドメイン内のオブジェクトにアクセスする場合、クライアントはこのサーバーによってサポートされる

クライアントはこのサーバーを、試行錯誤によって認識します。クライアントは、自分のホームサーバーから始めて、正しいサーバーが見つかるまでサーバーを1台ずつ試していきます。サーバーがクライアントの要求に応じられない場合、サーバーは適切なサーバーの検索に役立つ情報をクライアントに送信します。やがて、クライアントは自分の情報キャッシュを構築し、適切なサーバーをより効率的に検索できるようになります。このプロセスの詳細を次に示します。

コールドスタートファイルとディレクトリ キャッシュ

クライアントが初期設定されるとき、クライアントには「コールドスタートファイル」が与えられます。コールドスタートファイルの目的は、名前空間内のサーバーと連絡をとるための起点として使用できるディレクトリオブジェクトのコピーをクライアントに提供することです。ディレクトリオブジェクトには、アドレス、公開鍵、およびそのディレクトリをサポートするマスターサーバーと複製サーバーについての情報が収められています。通常、コールドスタートファイルには、クライアントのホームドメインのディレクトリオブジェクトが収められています。

コールドスタートファイルは、クライアントのローカルの「ディレクトリキャッシュ」を初期設定するためにだけ使用されます。ディレクトリキャッシュは、「キャッシュマネージャ」と呼ばれる NIS+ 機能によって管理されます。キャッシュマネージャには、クライアントが自身の要求を適切なサーバーに送信できるようにするためのディレクトリオブジェクトが格納されます。クライアントのコールドスタートファイルから取得された情報は、`/var/nis` 内の `NIS_SHARED_DIRCACHE` というファイルにダウンロードされます。

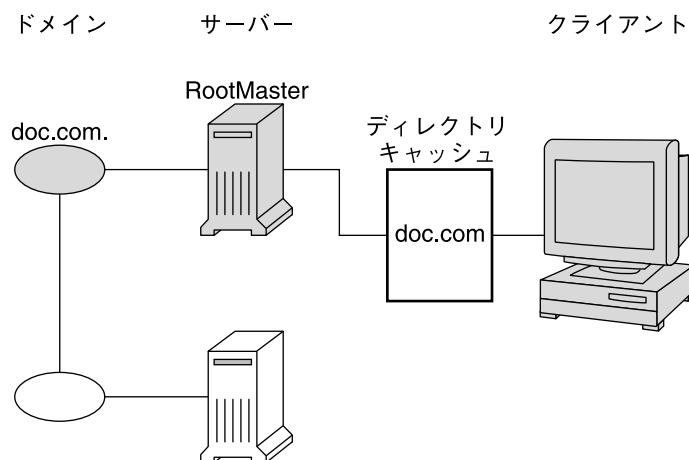


名前空間のディレクトリオブジェクトのコピーを自分のディレクトリキャッシュに格納することによって、クライアントは、どのサーバーがどのドメインをサポートするかを知ることができます。クライアントのキャッシュの内容を表示するには、`nisshowcache` コマンドを使用してください (342 ページの「`nisshowcache` コマンド」を参照)。簡単な例を次に示します。

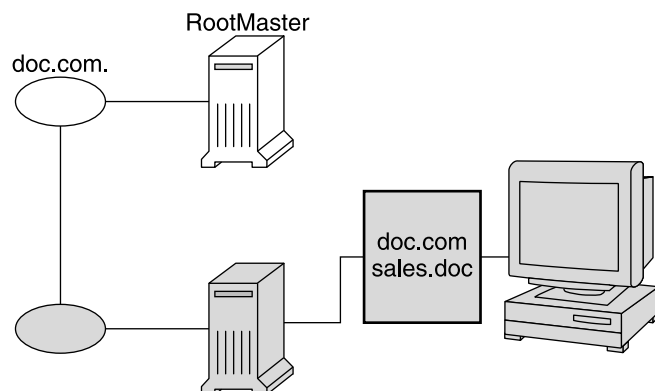
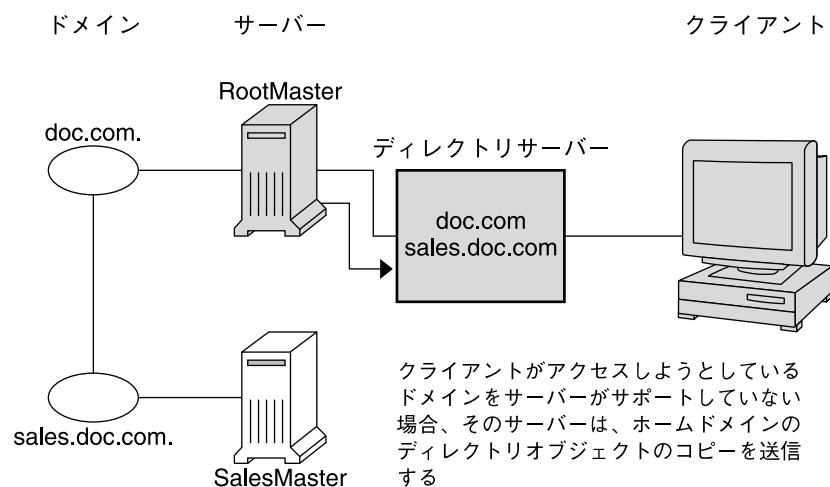
ドメイン名とディレクトリ名が同じ	サポートするサーバー	IP アドレス
doc.com.	rootmaster	172.29.6.77
sales.doc.com.	salesmaster	172.29.6.66
manf.doc.com.	manfmaster	172.29.6.37
int.sales.doc.com.	Intlsalesmaster	10.22.3.7

これらのコピーを最新の状態に保つため、各ディレクトリオブジェクトには「生存期間」フィールドがあります。このデフォルト値は 12 時間です。クライアントがディレクトリオブジェクトを求めてディレクトリキャッシュを調べ、最近の 12 時間以内にこれが更新されていないことを発見した場合、キャッシュマネージャはオブジェクトの新しいコピーを獲得します。347 ページの「`nischttl` コマンド」で説明するように、ディレクトリオブジェクトの生存期間の値は、`nischttl` コマンドで変更できます。ただし、生存期間を長くするほどオブジェクトのコピーが古くなる可能性が高くなり、生存期間を短くするほどネットワークトラフィックとサーバーのロードが増加することに注意してください。

ディレクトリキャッシュは、これらのディレクトリオブジェクトをどうやって蓄積するのでしょうか。前に説明したように、コールドスタートファイルはキャッシュ内の最初のエントリを提供します。したがって、クライアントが最初の要求を送信するとき、コールドスタートファイルによって指定されたサーバーに要求を送信します。その要求がそのサーバーによってサポートされるドメインへのアクセスである場合、そのサーバーは要求に応答します。



その要求が別のドメイン (たとえば、sales.doc.com.) へのアクセスである場合、サーバーは、クライアントが適切なサーバーを探すのを助けようとします。サーバーが、自分のディレクトリキャッシュ内に該当するドメイン用のエントリを保有している場合、そのドメインのディレクトリオブジェクトのコピーをクライアントに送信します。クライアントは、その情報を今後の参照用として自分のディレクトリキャッシュにロードし、そのサーバーに要求を送信します。



ほとんどありえないことですが、クライアントがアクセスしようとしているディレクトリオブジェクトのコピーを、サーバーが保有していない場合、そのサーバーは、自分のホームドメインのディレクトリオブジェクトのコピーをクライアントに送信します。ここには、そのサーバーの親のアドレスが登録されています。クライアントは、親サーバーを使ってこのプロセスを繰り返し、適切なサーバーを見つけるか、または名前空間内の全サーバーを試し終わるまで、これを繰り返します。クライアントがドメイン内の全サーバーを試し終わった後でどうするかは、そのネームサービスイッチ構成ファイルの指示によって決まります。

やがてクライアントは、名前空間内のすべてのディレクトリオブジェクトのコピーをキャッシュ内に蓄積します。したがって、これらディレクトリオブジェクトをサポートするサーバーの IP アドレスも蓄積することになります。クライアントがほかのドメインへのアクセス要求を送信する必要があるとき、通常は、自分のディレクトリキャッシュの中から自分のサーバー名を見つけ、そのサーバーにアクセス要求を直接送信できます。

NIS+ サーバーはクライアントでもある

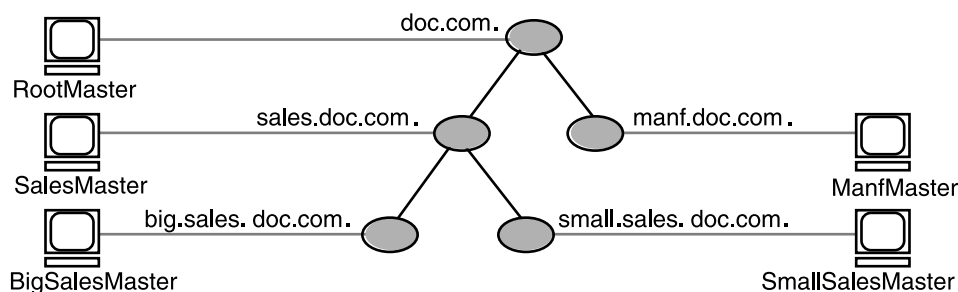
NIS+ サーバーは NIS+ クライアントでもあります。サーバーとして設定するマシンは、まずクライアントとして初期化する必要があります。唯一の例外はルートマスターサーバーであり、このサーバーには独自の設定を行う必要があります。

つまり、サーバーはドメインをサポートするだけでなく、ドメインにも「所属する」のです。つまり、クライアントになることによって、サーバーにはホームドメインがあるということです。サーバーのホスト情報は自分のホームドメインの `hosts` テーブルに格納され、その DES 資格は自分のホームドメインの `cred` テーブルに格納されます。ほかのクライアントと同様、サーバーは、自分のディレクトリキャッシュに記録されているサーバーに対して、サービス要求を送信します。

忘れてはならない重要な点は、ルートドメインを除いて、サーバーのホームドメインは、そのサーバーがサポートするドメインの「親」ドメインであるということです。

つまり、サーバーは 1 つのドメイン内のクライアントをサポートしますが、別のドメインの「クライアント」になるのです。ルートドメインを除いて、サーバーは自分のサポートするドメインのクライアントにはなれません。ルートドメインをサポートするサーバーには親ドメインがないため、これらはルートドメイン自体に所属します。

たとえば、次のような名前空間を考えてみます。



各サーバーがどのドメインをサポートし、どのドメインに所属するかを次に示します。

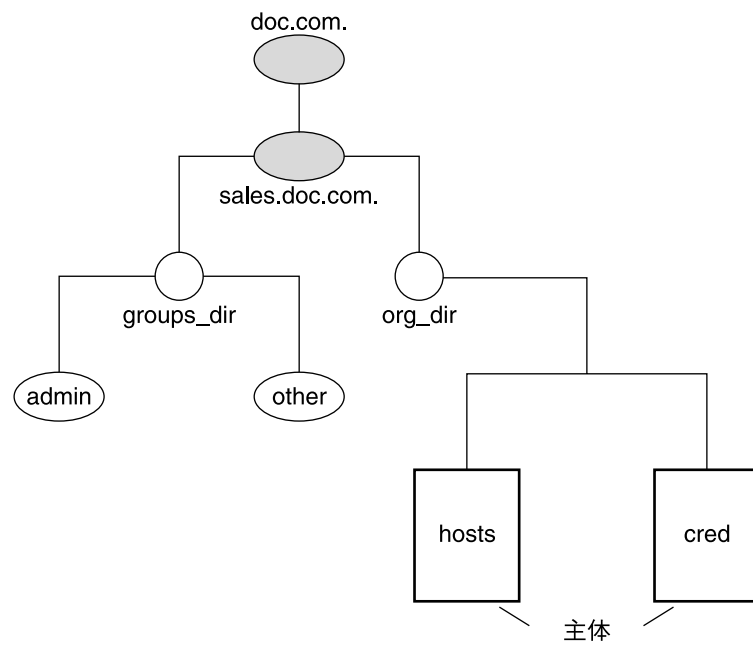
サーバー	サポートするドメイン	所属するドメイン
RootMaster	doc.com.	doc.com.
SalesMaster	sales.doc.com.	doc.com.
IntlSalesMaster	intl.sales.doc.com.	sales.doc.com.
ManfMaster	manf.doc.com.	doc.com.

命名規則

NIS+ 名前空間内のオブジェクトは、「部分指定」および「完全指定」の 2 種類の名前によって識別できます。部分指定名は、「単純」名とも呼ばれ、オブジェクト名だけまたは完全指定名の一部です。管理操作中にオブジェクトまたは主体の部分指定名を入力すると、NIS+ によってその名前は完全指定名に展開されます。詳細については、[72 ページの「命名規則」](#)を参照してください。

完全指定名とは、名前空間内でオブジェクトを探すために必要なすべての情報を含んだオブジェクトの完全な名前です。必要なすべての情報とはオブジェクトの親ディレクトリ (もしあれば)、最後のドットも含んだ完全なドメイン名などです。

これはオブジェクトの種類によって異なるため、各タイプと NIS+ の主体ごとの規則を別個に説明します。次の名前空間を例として使用します。



NIS+ の主体を含む、この名前空間内の全オブジェクトの完全指定名を次の図に要約します。

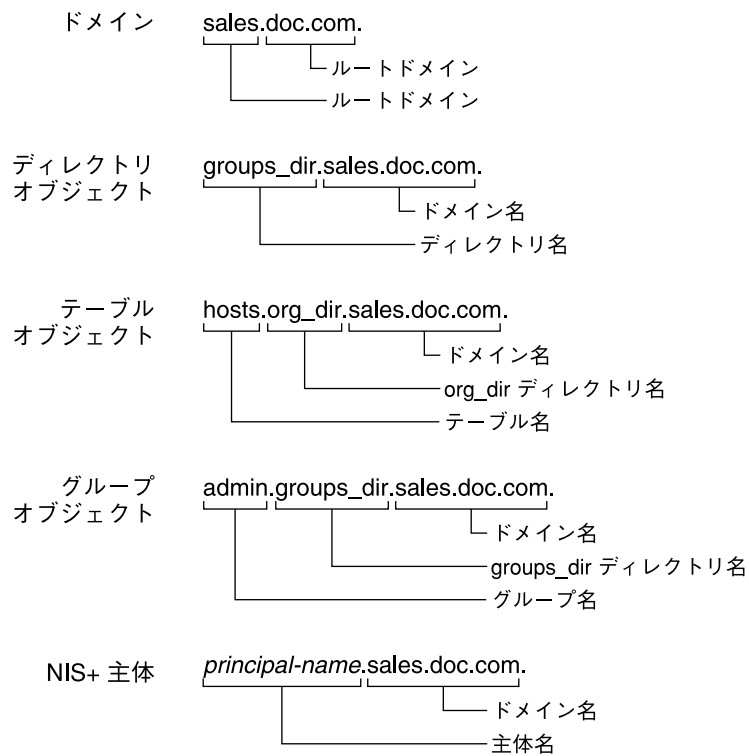


図 2-4 名前空間の構成要素の完全指定名

NIS+ ドメイン名

完全指定ドメイン名は左から右に作成され、ローカルドメインで始まってルートドメインで終わります。

doc.com. (ルートドメイン)

sales.doc.com. (サブドメイン)

intl.sales.doc.com. (第3レベルのサブドメイン)

最初の行はルートドメインの名前を示します。ルートドメインは、少なくとも2つのラベルを持ち、ドットで終わらなければなりません。最後(右端)のラベルは自由につけられますが、インターネット上の互換性を維持するためには、以下に示すインターネットの組織名、または2、3文字の地域識別子(日本であれば .jp)をつける必要があります。

表 2-4 インターネットの組織ドメイン

ドメイン	目的
com	営利団体
edu	教育機関
gov	行政機関
mil	軍事組織
net	主要ネットワークサポートセンター
org	非営利団体
int	国際組織

上の 2 行目と 3 行目は下位レベルドメインの名前です。

ディレクトリオブジェクト名

ディレクトリの単純名は、単にディレクトリオブジェクトの名前です。この完全指定名は、単純名にそのドメインの完全指定名を加えたものです (常に最後のドットを含む)。

groups_dir (単純名)

groups_dir.manf.doc.com. (完全指定名)

複数のディレクトリ層が 1 つのドメインを形成しないような変則的な階層を設定する場合、中間ディレクトリ名を含めるようにしてください。たとえば、以下のようになります。

lowest_dir.lower_dir.low_dir.mydomain.com.

通常単純名は、同じドメイン内から使用され、完全指定名は、リモートドメインから使用されます。ただし、ドメインの NIS_PATH 環境変数に検索パスを指定することによって、リモートドメインから単純名を使用できます (78 ページの「環境変数 NIS_PATH」を参照)。

テーブルおよびグループ名

完全指定されたテーブル名とグループ名は、オブジェクト名から始まり、ディレクトリ名をつけ、その後に完全指定されたドメイン名が続いて作られます。すべてのシステムテーブルオブジェクトは org_dir ディレクトリに格納されて、すべてのグループオブジェクトは groups_dir ディレクトリに格納されていることを忘れないでください。独自の NIS+ テーブルを作成した場合は、どこでも好きなところに格納できます。グループ名とテーブル名の例を次に示します。

```
admin.groups_dir.doc.com.  
admin.groups_dir.doc.com.  
admin.groups_dir.sales.doc.com.  
admin.groups_dir.sales.doc.com.  
hosts.org_dir.doc.com.  
hosts.org_dir.doc.com.  
hosts.org_dir.sales.doc.com.  
hosts.org_dir.sales.doc.com.
```

テーブルエントリ名

NIS+ テーブル内のエントリを識別するには、その中にあるエントリとテーブルオブジェクトを識別する必要があります。このような名前を「インデックス付き」名と呼びます。この構文を次に示します。

```
[column=value,column=value,...],tablename
```

Column はテーブル列の名前です。*Value* はその列の実際の値です。*Tablename* は、テーブルオブジェクトの完全指定名です。hosts テーブルのエントリ例を次に示します。

```
[addr=129.44.2.1,name=pine],hosts.org_dir.sales.doc.com.  
[addr=129.44.2.2,name=elm],hosts.org_dir.sales.doc.com.  
[addr=129.44.2.3,name=oak],hosts.org_dir.sales.doc.com.
```

括弧の中には、テーブルエントリを一意に識別するために必要な最小限の列の値のペアを使用できます。

一部の NIS+ 管理コマンドは、この構文のバリエーションを利用できます。詳細は、パート II 「NIS+ の紹介と概要」の nistbladm、nismatch、nisgrep の各コマンドの説明を参照してください。

ホスト名

ホスト名の長さは 24 文字までで、使用できるのは文字、数字、ダッシュ (-)、および下線 (_) です。大文字と小文字の区別はありません。また先頭は英文字とします。スペースは使用できません。

注 - ドット (.) はホスト名には使用できません。たとえば、ホスト名には doc.2 などを使用できません。また、引用符で囲んだドットも、ホスト名には使用できません。たとえば、'doc.2' は使用できません。ドメイン構成要素を識別するために、完全指定ホスト名の一部として使用する場合にのみ、ドットを使用します。たとえば、doc-2.sales.doc.com. は、正しい完全指定のホスト名です。

またドメイン名とホスト名が同じにならないようにします。たとえば、sales ドメインがある場合は、ホスト名に sales を使用することはできません。同様に、home というホスト名がある場合には、ドメイン名に home を使用できません。これは、サブドメインについても同様です。たとえば、すでにホスト名として west がある場合には、sales.west.doc.com というサブドメインを作成することはできません。

NIS+ の主体名

NIS+ の主体名は、Secure RPC のネット名とよく混同することがあります。念のためここで 1 つだけ違いを指摘しておきます。NIS+ の主体名は必ずドットで終わりますが、Secure RPC のネット名はドットで終わることは決してありません。以下に、例を示します。

NIS+ の主体名 olivia.sales.doc.com.

Secure RPC netname unix.olivia@sales.doc.com

また、たとえ主体の資格が cred テーブルに格納されていても、cred テーブル名も org_dir ディレクトリ名も主体名には含まれません。

使用できる記号

名前空間名は、ISO ラテン 1 セットの印刷可能文字を自由に使用して作成できます。ただし、名前の先頭には次に示す文字は使用できません。@ < > + [] - / = . , : ;

文字列を使用するには、二重引用符で囲みます。名前の中に引用符を使用するには、その記号も引用符で囲みます (たとえば、o'henry を使用するには o'"henry と入力します)。John Smith などのように空白スペースを組み込むには、次のように一重引用符の中で二重引用符を使用します。

`"John Smith"``

ホスト名に適用される制限については、[76 ページの「ホスト名」](#)を参照してください。

NIS+ の名前展開

NIS+ のコマンドで完全指定名を入力することは面倒です。この作業を簡単にするため、NIS+ は名前展開機能を提供します。部分指定名が入力されると、NIS+ は別のディレクトリのもとでこのオブジェクトを見つけようと試みます。最初は、デフォルトドメインから探します。これは、このコマンドを入力したクライアントのホームドメインです。デフォルトドメインでオブジェクトが見つからなかった場合、このオブジェクトが見つかるまで、NIS+ はデフォルトドメインの親ディレクトリを下から上

へ探していきます。2つのラベルしかない名前に到達すると、検索を停止します。次にその例を示します。この例では、`software.big.sales.doc.com.` ドメインに所属するクライアントにログインしたと仮定します。

```
mydir — 展開する → mydir.software.big.sales.doc.com.  
mydir.big.sales.doc.com.  
mydir.sales.doc.com.  
mydir.doc.com.  
  
hosts.org_dir — 展開する → hosts.org_dir.software.big.sales.doc.com.  
hosts.org_dir.big.sales.doc.com.  
hosts.org_dir.sales.doc.com.  
hosts.org_dir.doc.com.
```

環境変数 NIS_PATH

環境変数 `NIS_PATH` の値を変更することによって、NIS+ が検索するディレクトリのリストを変更または拡張できます。`NIS_PATH` には、コロンで区切って複数のディレクトリ名を設定できます。次に例を示します。

```
setenv NIS_PATH directory1: directory2: directory3 ...
```

または

```
NIS_PATH=directory1: directory2: directory3 ...;export NIS_PATH
```

NIS+ は、これらのディレクトリを左から右へ検索していきます。たとえば、以下のようになります。

```
NIS_PATH —————→ sales.doc.com.:manf.doc.com:doc.com.  
mydir — 展開する → mydir.sales.doc.com.  
mydir.manf.doc.com.  
mydir.doc.com.  
  
hosts.org_dir — 展開する → hosts.org_dir.sales.doc.com.  
hosts.org_dir.manf.doc.com.  
hosts.org_dir.doc.com.
```

\$PATH や \$MANPATH のように、変数 NIS_PATH には特殊記号 \$ を使用できます。\$ 記号は、ディレクトリ名に追加するか、または単独で追加できます。ディレクトリ名に追加した場合、NIS+ はその名前にデフォルトディレクトリを追加します。たとえば、以下のようになります。

NIS_PATH	\$:org_dir.\$:groups_dir.\$ mydir.big.sales.doc.
デフォルトディレクトリ が以下のような場合	NIS_PATH は 以下ようになる
sales.doc.com.	sales.doc.com.:org_dir.sales.doc.com.:groups_dir.sales.doc.com.
manf.doc.com.	manf.doc.com.:org_dir.manf.doc.com.:groups_dir.manf.doc.com.
doc.com.	doc.com.:org_dir.doc.com.:groups_dir.doc.com.

\$ 記号を単独で使用する (たとえば、org_dir.\$:\$) 場合、NIS+ は前述のような標準の名前展開を実行します。つまり、デフォルトディレクトリの検索から始め、親ディレクトリへと進めていきます。NIS_PATH のデフォルト値は \$ です。

注 - NIS_PATH に追加や変更を加えると、NIS+ がより多くの検索を行わなければならないので、性能が低下するという点に注意してください。

名前空間がすでに存在する場合の設定

NIS ドメインが NIS 名前空間にすでに存在する場合は、そのドメインを NIS+ 名前空間に同じフラット構造で移行できます。移行したドメインは、あとで階層構造に変更できます。NIS から NIS+ への移行を始める前に、計画および準備に関する重要な情報について、『Solaris のシステム管理 (ネーミングとディレクトリサービス : NIS+ 編)』の「スクリプトを使用した NIS+ の設定」を参照してください。NIS+ のスクリプトを使用すると、NIS マップのデータを利用して簡単に NIS+ を起動できます。第 4 章で、NIS+ スクリプトを使用してシステムファイルや NIS マップから NIS+ 名前空間を作成する方法を説明します。

ただし、名前空間がすでに存在している場合、スクリプトをスムーズに実行するため NIS+ への移行用の設定が必要です。

準備に関する主な注意事項は次のとおりです。

- 「ドメイン名とホスト名」。ドメイン名とホスト名は、同じにならないようにする必要があります。たとえば、sales ドメインがある場合は、ホスト名に sales を使用することはできません。同様に、home というホスト名がある場合には、ドメイン名に home を使用できません。これはサブドメインの場合も同様です。たとえば、マシンに west という名前を付けている場合、sales.west.doc.com というサブディレクトリを作成しないでください。
- 「ホスト名にはドットを使用しない」。NIS+ では、マシン名とドメイン名の間や、親とサブドメインの間の区切りにドット (ピリオド) を使用します。このため、マシン名の中ではドットを使用できません。使用している場合は、NIS+ に移行する前に (スクリプトを実行する前に) 必ず変更してください。ホスト名中のドットは、ハイフンに置き換えます。たとえば、sales.alpha というマシン名を付けることはできません。この名前は、sales-alpha に置き換えることができます (使用可能なホスト名の詳細については、hosts のマニュアルページを参照してください)。
- 「ルートサーバーが動作していることを確認する」。ルートサーバーになるマシンが起動されていることを確認します。また、スーパーユーザーとしてアクセスできることも確認します。
- データのロード元のローカル /etc ファイル、または NIS マップのエントリを確認する。不正なエントリがないことを確認します。正しいデータが、所定の場所に正しい書式で記録されていることを確認します。エントリのうち、古いもの、無効なもの、破損しているものは削除します。また、不完全なエントリや一部のみのエントリも削除します。構成完了後は、いつでもエントリを追加できます。必要なエントリをあとから追加する方が、不完全なエントリや破損しているエントリを読み込もうとするよりも簡単です。



注意 – Solaris 2.4 以前では、/var/nis ディレクトリに *hostname.dict*、*hostname.log* という 2 つのファイルが含まれていました。また、/var/nis/*hostname* というサブディレクトリも含まれていました。Solaris 2.5 で NIS+ をインストールした場合、2 つのファイル名は *trans.log*、*data.dict*、サブディレクトリ名は /var/nis/*data* となります。Solaris 2.5 ではこれらのファイルの内容も変更されており、Solaris 2.4 以前との互換性はなくなっています。したがって、これらのファイルやディレクトリを Solaris 2.4 での名前にしてしまうと、Solaris 2.4、2.5 双方の *rpc.nisd* で機能しなくなります。ディレクトリ名もファイル名も変更しないでください。

2 通りの構成方法

ここでは、NIS+ 名前空間の 2 通りの構成方法を紹介します。

- 「設定 (構成) 用のスクリプトを使う方法」。nissserver、nispopulate、およびnisclient という 3 つの NIS+ スクリプトを使用して NIS+ を構成する方法については、この章と第 4 章で説明しています。簡単に済ませることができるため、こちらの方法をお勧めします。
- 「NIS+ コマンドを使う方法」。NIS+ コマンドを使用して NIS+ を構成する方法については、第 6 章、第 7 章、および第 8 章で説明します。NIS+ コマンドを使用すれば、NIS+ スクリプトより柔軟に構成できますが、複雑になります。NIS+ コマンドは、NIS+ スクリプトによって生成される名前空間と特性が大きく異なる名前空間を構成する必要があるときに、経験豊富な NIS+ 管理者だけが使用してください。

注 - NIS+ コマンドセットを使用する場合は、ネームサービスに NIS+ を使用する各マシンの /etc ディレクトリに正しい nsswitch.conf ファイルが存在することも確認してください (第 1 章を参照)。この作業は、NIS+ 構成スクリプトを使った場合は自動的に行われます。

NIS+ ディレクトリまたはドメイン、NIS+ サーバー、あるいは NIS+ 名前空間の削除方法については第 22 章を参照してください。

第 3 章

NIS+ 設定スクリプト

この章では、NIS+ スクリプトとその機能について説明します。

注 - NIS+ は、将来のリリースではサポートされなくなる可能性があります。NIS+ から LDAP への移行支援ツールは、Solaris 9 リリース以降で使用できます (『Solaris のシステム管理 (ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』を参照)。詳細については、<http://www.sun.com/directory/nisplus/transition.html> を参照してください。

NIS+ スクリプトについて



注意 - 必ず「NIS+ 設定の概要」の手順を実行してから、NIS+ スクリプトを実行してください。

3 つの NIS+ スクリプト `nisservice`、`nispopulate`、`niscclient` を使用すれば、NIS+ 名前空間を設定できます。NIS+ スクリプトは NIS+ コマンド群を実行する Bourne シェルスクリプトであり、NIS+ コマンドを個別に入力する必要はありません。次の表に、各スクリプトの機能を示します。

表 3-1 NIS+ スクリプト

NIS+ スクリプト	機能
nisserver	ルートマスターサーバー、ルート以外のマスターサーバー、複製サーバーをレベル 2 のセキュリティ (DES) で設定する
nispopulate	NIS+ テーブルを、対応するシステムファイルまたは NIS マップから指定されたドメイン内に生成する
nisclient	ホストとユーザーの NIS+ 資格を作成し、NIS+ のホストとユーザーを初期設定する

NIS+ スクリプトで実行すること

NIS+ スクリプトといくつかの NIS+ コマンドと組み合わせることで、NIS+ 名前空間の設定に必要な作業を実行できます。NIS+ スクリプトとそのオプションについての詳しい説明は、`nisserver(1M)`、`nispopulate(1M)`、`nisclient(1M)` のマニュアルページを参照してください。第 4 章では、NIS+ スクリプトを使用した NIS+ 名前空間の設定方法について説明しています。

-x オプションを使用して、コマンドを実際に行わずに各スクリプトを実行できます。このオプションを使用すると、スクリプトが呼び出すコマンドとおおよその結果をシステムを変更せずに確認できます。-x オプションを使用してスクリプトを実行することで、予想外の結果を最小限にできます。

NIS+ スクリプトでは実行しないこと

NIS+ スクリプトを使用すれば NIS+ 名前空間の作成に必要な作業が軽減されるとはいえ、スクリプトですべての NIS+ コマンドを代行できるわけではありません。スクリプトは、NIS+ の一部の機能を提供するだけです。NIS+ にまだ慣れていない場合、NIS+ 名前空間のサンプルを作成してからこの節を読むこともできます。

`nisserver` スクリプトは、標準のテーブルとアクセス権 (承認) で NIS+ サーバーを設定するだけです。このスクリプトでは次のことは実行しません。

- テーブルとディレクトリに特別なアクセス権を設定する
- NIS+ 管理グループへ NIS+ 主体を追加する

NIS+ スクリプトではなく `nisgrpadm` コマンドを使用して、NIS+ 管理グループに特別な NIS+ 主体を追加する方法については、第 4 章を参照してください。
- 独自のテーブルを作成する

- レベル 2 以外の任意のセキュリティレベルで NIS+ サーバーを実行する
- リモートサーバーで `rpc.nisd` デーモンを起動する。この作業は、サーバーのインストールを完了するために必要

NIS+ スクリプトではなく `rpc.nisd` コマンドを使用して、NIS+ クライアントマシンをルート以外のサーバーに変更する方法については、[第 4 章](#)を参照してください。

`nisclient` スクリプトは、DNS を使用してホスト名を検索するように NIS+ クライアントを設定することはありません。この設定を必要とするクライアントの場合、DNS を明確に設定しなければなりません。

第 4 章

スクリプトを使用した NIS+ の設定

この章では、`nissserver`、`nispopulate`、`nisclient` の各スクリプトと NIS+ コマンドを組み合わせ、基本的な NIS+ 名前空間を構成する方法について説明します。

注 - NIS+ は、将来のリリースではサポートされなくなる可能性があります。NIS+ から LDAP への移行支援ツールは、Solaris 9 リリース以降で使用できます (『Solaris のシステム管理 (ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』を参照)。詳細については、<http://www.sun.com/directory/nisplus/transition.html> を参照してください。

NIS+ 設定の概要

NIS+ 名前空間の設定と構成には、スクリプトを使用する方法をお勧めします。これらのスクリプトを使用すると、NIS+ コマンドセットを使用する場合よりも簡単に NIS+ 名前空間を設定できます。第 6 章、第 7 章、および第 8 章を参照してください。

スクリプトの詳細は、`nissserver(1M)`、`nispopulate(1M)`、`nisclient(1M)` の各マニュアルページを参照してください。用語や略語の定義については、用語集を参照してください。

このチュートリアルで説明するサンプルの小さな NIS+ 名前空間を土台にして実際の NIS+ 名前空間を作ることはしないでください。名前空間についてひととおり理解できたら、サンプルの名前空間は削除してください。サンプルの名前空間に実際の名前空間を追加しないでください。あらためて、初めから NIS+ 階層の計画を注意深く立ててから、実際の名前空間を作成してください。

一般に推奨される設定手順を表 4-1 に要約します。左端の列には、ルートドメインの構成やクライアントの作成などの主な設定作業を示します。中央の列は、作業の説明です。3 番目の列は、各手順に必要なスクリプトまたはコマンドです。

表 4-1 NIS+ の推奨構成手順の概要

作業	目的	スクリプトまたはコマンド
NIS+ 名前空間の計画を立てる	NIS+ 名前空間の計画を立てる。計画の要件と手順の詳細は、『Solaris のシステム管理 (ネーミングとディレクトリサービス : NIS+ 編)』の「NIS+ の紹介」を参照してください。(試験的なネットワークで NIS+ チュートリアルを使用している場合、この手順は不要)	
既存の名前空間を設定する	スクリプトを正常に実行するためには、既存の名前空間を適切に設定する必要がある。必要な準備については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : NIS+ 編)』の「名前空間がすでに存在する場合の設定」を参照してください。(試験的なネットワークで NIS+ チュートリアルを使用している場合、この手順は不要)	
Diffie-Hellman キー長を設定する	DES 認証を使用する場合には、デフォルトの 192 ビットよりも長い Diffie-Hellman キーの使用を考慮する。キーを長くする場合、その長さはドメイン内のすべてのマシンで同一である必要がある。各初期設定スクリプトの実行する前に、目的のキー長を指定する	nisauthconf
ルートドメインの構成	ルートドメインを作成する。ルートマスターサーバーの構成と初期設定を行う。ルートドメイン管理グループを作成する	nissserver
テーブルの生成	テキストファイルまたは NIS マップからルートドメインの NIS+ テーブルを生成 (populate) する。ルートドメインクライアントの資格 (credential) を作成する。管理者の資格を作成する	nispopulate nisgrpadm nisping
ルートドメインクライアントの構成	クライアントマシンを構成する (そのうちの何台かは後にサーバーになる)。ユーザーを NIS+ クライアントとして初期設定する	nisclient
サーバーを使用可能にする	ルートドメインの一部のクライアントをサーバーとして使用可能にする。サーバーの一部は後でルート複製サーバーとなり、そのほかは下位レベルのドメインをサポートする	svcadm enable
ルート複製サーバーの構成	構成したばかりのサーバーのうち、1 台または複数をルートドメインの複製として指定する	nissserver svcadm
ルート以外のドメインの構成	新しいドメインを作成します以前の手順で使用可能にしたサーバーを、そのマスターとして指定する。その管理グループと管理資格を作成する	nissserver

表 4-1 NIS+ の推奨構成手順の概要 (続き)

作業	目的	スクリプトまたはコマンド
テーブルの生成	新しいドメインのクライアントの資格を作成する。テキストファイルまたは NIS マップから、新しいドメインの NIS+ テーブルを生成する	<code>nispopulate</code>
ルート以外のドメインのクライアントを構成する	新しいドメインのクライアントを構成する (一部のクライアントは、後に下位レベルのドメインのサーバーとなる場合がある)。ユーザーを NIS+ クライアントとして初期設定する	<code>nisclient</code>

NIS+ スクリプトを使用することによって、上の作業で示される個々の手順の大部分を省略できます。

NIS+ とサービス管理機能

NIS+ サービスに関連したコマンド行管理タスクのほとんどは、サービス管理機能 (SMF) によって管理されます。SMF の概要については、『Solaris のシステム管理 (基本編)』の「サービスの管理 (概要)」を参照してください。詳細については、`svcadm (1M)` と `svcs (1)` の各マニュアルページも参照してください。

- NIS+ サービスに対する有効化、無効化、再起動などの管理操作を実行するには、`svcadm` コマンドを使用します。サービスを起動または停止すると、依存するプロセスもすべて起動または停止します。

ヒント - `-t` オプションを使ってサービスを一時的に無効にすると、サービス構成の一部を保護できます。`-t` オプションを使ってサービスを無効にした場合は、再起動後にサービスの元の設定が復元されます。`-t` オプションを使わずにサービスを無効にした場合は、再起動後もサービスは引き続き無効になります。

- SMF では、`/var/nis/NIS_COLD_START` ファイルが検出されると、NIS+ サービスを有効にしたときに `nis_cachemgr` が自動的に起動されます。
- NIS+ の FMRI (Fault Managed Resource Identifier) は `svc:/network/rpc/nisplus:<instance>` です。
- `keyserv` の FMRI は `svc:/network/rpc/keyserv:<instance>` です。
- NIS+ の状態を問い合わせるには、`svcs` コマンドを使用します。
 - `svcs` コマンドと出力の例を次に示します。

```
# svcs \*nisplus\*
STATE      STIME      FMRI
disabled   Sep_01     svc:/network/rpc/nisplus:default
```

- `svcs -l` コマンドと出力の例を次に示します。

```
# svcs -l network/rpc/nisplus
fmri      svc:/network/rpc/nisplus:default
```

```

enabled      false
state        disabled
next_state   none
restarter    svc:/system/svc/restarter:default
dependency   require_all/none svc:/network/rpc/keyserv (online)

```

また、svccfg ユーティリティを使用すれば、サービスに関する詳細な情報を取得できます。svccfg (1M) のマニュアルページを参照してください。

- デーモンが存在するかどうかを確認するには、ps コマンドを使用します。

```
# ps -e | grep rpc.nisd
```

注 -f オプションは ps とともに使わないでください。このオプションはユーザー ID を名前に変換しようとするため、より多くのネームサービスの検索が失敗する可能性があります。

svcadm を rpc.nisd -x とともに使用する

一般に、/usr/sbin/rpc.nisd デーモンは svcadm コマンドを使って管理します。ただし、rpc.nisd が -x nisplusLDAPinitialUpdateOnly=yes を使って呼び出された場合、rpc.nisd は指定された操作を実行して、終了します。つまり、rpc.nisd はデーモンとして動作しません。SMF は -x nisplusLDAPinitialUpdateOnly=yes とともに使用しません。SMF は、それ以外で rpc.nisd デーモンを起動、停止、または再起動する場合にいつでも使用できます。

rpc.nisd を -x nisplusLDAPinitialUpdateOnly=yes で使用する例を次に示します。

```
# /usr/sbin/rpc.nisd -m mappingfile \
-x nisplusLDAPinitialUpdateAction=from_ldap \
-x nisplusLDAPinitialUpdateOnly=yes
```

/lib/svc/method/nisplus ファイルの変更

SMF を使って rpc.nisd デーモンを呼び出すときに特定のオプションを指定する場合は、そのオプションを /lib/svc/method/nisplus ファイルに追加します。次に、よく使われるオプションをいくつか示します。

-s 0	サーバーのセキュリティレベルを 0 に設定します。現時点ではブートストラップに必要です。
	cred テーブルがまだ存在しないので、NIS+ 主体は資格を持つことができません。このため、高いセキュリティレベルを使用すると、サーバーからロックアウトされます。
-B	DNS 転送をサポートします。
-Y	NIS 互換モードで NIS+ デーモンを起動します。

▼ */lib/svc/method/nisplus* ファイルの変更方法

1. スーパーユーザーになるか、同等の役割になります。
役割には、承認と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「役割に基づくアクセス制御の使用 (作業)」を参照してください。

2. NIS+ サービスを停止します。

```
# svcadm disable network/rpc/nisplus:default
```

3. */lib/svc/method/nisplus* ファイルを開きます。
適切なテキストエディタを使用します。

4. ファイルを編集して必要なオプションを追加します。

例 —

変更前:

```
/usr/sbin/rpc.nisd $nisd_flags || exit $?
```

変更後:

```
/usr/sbin/rpc.nisd $nisd_flags -Y -B || exit $?
```

この例では、-Y および -B オプションが *rpc.nisd* に追加され、起動時に自動的に実装されます。

5. 保存して終了します。

6. NIS+ サービスを開始します。

```
# svcadm enable network/rpc/nisplus:default
```

NIS+ 名前空間サンプルの作成

この章では、サンプルの NIS+ 名前空間を作成する方法を説明します。NIS+ 名前空間のサンプルは、`/etc` 内のファイルと NIS マップから作成されます。このサンプルでは、サイトで NIS を実行している場合と実行していない場合の両方について、スクリプトの使用法を説明します。サーバーが NIS クライアントにサービスを提供する場合、サーバーを NIS 互換モードに設定できます。NIS 互換モードの詳細については、[51 ページの「Solaris 1.x と NIS 互換モード」](#)を参照してください。

注 - サイトの実際の NIS+ 名前空間とそのドメイン階層は、名前空間サンプルのものとはおそらく異なり、サーバー、クライアント、およびドメインの数も異なります。最終的なドメイン構成と階層は、このサンプルと異なるものと考えてください。この名前空間サンプルは、NIS+ スクリプトの使用法を説明するためだけのものです。この名前空間サンプルを作成すると、自分のサイトでのドメイン、サーバー、およびクライアントの作成方法も理解できるはずです。

名前空間サンプルには次の構成要素があります。

- ルートマスターサーバー 1 台 (名称 `master`、`doc.com`. ドメイン用)
- ルートドメイン (`doc.com.`) のクライアント 4 台
 - `client1` は、ルート複製サーバーとなる (`doc.com`. ドメイン用)
 - `client2` は、新しいサブドメインのマスターサーバーとなる (`sub.doc.com`. ドメイン用)
 - `client3` は、新しいサブドメインのルート以外の複製サーバーとなる (`sub.doc.com`. ドメイン用)
 - `client4` は、ルートドメインのクライアントとして残る (`doc.com`. ドメイン用)
- サブドメインの 2 台のクライアント (名称 `subclient1` および `subclient2`、`sub.doc.com`. ドメイン用)

ここでは、`/etc/hosts` などのシステム情報ファイルと NIS マップの両方を使用してネットワークサービス情報を格納するサイトで、NIS+ の設定に使用されるスクリプトを説明します。NIS+ 名前空間サンプルでこのような混合型のサイトを使用するのは、単にサンプルとして示すことが目的です。

NIS+ スクリプトのコマンド行の要約

表 4-2 に、NIS+ ドメインのサンプルを作成するときの、NIS+ スクリプトとコマンドの汎用的な手順を示します。このあとの節ではこれらのコマンド行について詳しく説明します。NIS+ のドメイン、サーバー、およびクライアントの作成に必要な作業に習熟した後、表 4-2 はコマンド行のクイックリファレンスガイドとして使用してください。表 4-2 は、NIS+ 名前空間サンプルを作成するために実際に入力するコマンドと変数をまとめたものです。

表 4-2 NIS+ ドメインの構成に使うコマンド行のまとめ

動作	マシン	コマンド
/usr/lib/nis をルートのパスに追加する。C シェルまたは Bourne シェルを使用	ルートマスターサーバーとクライアントマシン。スーパーユーザーとして行う	setenv PATH \$PATH:/usr/lib/nis または PATH=\$PATH:/usr/lib/nis; export PATH
オプションで DES 認証を使用する場合には、Diffie-Hellman キー長を選択する	サーバーとクライアントマシン。スーパーユーザーとして行う	nisauthconf -dhkey-length-alg-type des
NIS (YP) との互換性があるか、または互換性がないルートマスターサーバーを作成する	ルートマスターサーバー。スーパーユーザーとして行う	nisserver -r-d newdomain. または nisserver -Y-r-d newdomain.
ファイルまたは NIS マップからルートマスターサーバーテーブルを生成する	ルートマスターサーバー。スーパーユーザーとして行う	nispopulate -F-p /files -d newdomain. または nispopulate -Y-d newdomain. -h NISservername\ -a NIS_server_ipaddress -y NIS_domain
NIS+ 管理グループにユーザーを追加する	ルートマスターサーバー。スーパーユーザーとして行う	nisgrpadm -a admin.domain.name.domain.
NIS+ データベースのチェックポイントを実行する	ルートマスターサーバー。スーパーユーザーとして行う	nisping -C domain.
新しいクライアントマシンを初期設定する	クライアントマシン。スーパーユーザーとして行う	nisclient -i-d domain. -h master1
ユーザーを NIS+ クライアントとして初期設定する	クライアントマシン。ユーザーとして行う	nisclient -u

表 4-2 NIS+ ドメインの構成に使うコマンド行のまとめ (続き)

動作	マシン	コマンド
NIS+ サービス (rpc.nisd デモン) を起動する — クライアントを NIS との互換性 (および DNS 転送機能) を備えた (または備えていない) サーバーに変換するのに必要となる	クライアントマシン。スーパーユーザーとして行う	必要に応じて -Y または -B オプションを /lib/svc/method/nisplus ファイルに追加してから、NIS+ サービスを有効にする svcadm enable /network/rpc/nisplus
サーバーをルート複製サーバーにする	ルートマスターサーバー。スーパーユーザーとして行う	nisserver-R-d <i>domain</i> . -h <i>clientname</i>
サーバーをルート以外のマスターサーバーにする	ルートマスターサーバー。スーパーユーザーとして行う	nisserver -M-d <i>newsubdomain.domain</i> . -h\ <i>clientmachine</i>
ファイルまたは NIS マップから新しいマスターサーバーテーブルを生成する	新しいサブドメインマスターサーバー。スーパーユーザーとして行う	nispopulate -F-p/ <i>subdomain</i> directory -d \ <i>newsubdomain.domain</i> . または nispopulate-Y-d <i>newsubdomain.domain</i> . -h <i>NISservername</i> -a <i>NIS_server_ipaddress</i> -y <i>NIS_domain</i>
クライアントをマスターサーバーの複製にする	サブドメインマスターサーバー。スーパーユーザーとして行う	nisserver-R-d <i>subdomain.domain</i> パターンを使用して名前を付けることができます。-h <i>clientname</i>
サブドメインの新しいクライアントを初期設定する。クライアントは、サブドメインの複製またはそのほかのサーバーにできる	新しいサブドメインクライアントマシン。スーパーユーザーとして行う	nisclient -i -d <i>newsubdomain.domain</i> パターンを使用して名前を付けることができます。-h \ <i>subdomainmaster</i>
ユーザーを NIS+ クライアントとして初期設定する	クライアントマシン。ユーザーとして行う	nisclient -u

注 - 実際にコマンドを実行させないで、しかも NIS+ スクリプトがどんなコマンドを呼び出すかを表示するには、-x オプションを使います。-x オプションによって、あたかも実際にスクリプトを実行しているかのように、それらのコマンド名とそれぞれのおおよその出力を画面に表示できます。最初に -x を指定してスクリプトを実行すれば、結果を先に見ることができます。詳細は各スクリプトのマニュアルページを参照してください。

NIS+ ルートサーバーの設定

NIS+ ドメインを設定するための最初の作業が、ルートマスターサーバーの設定です。この節では、`nisserver` スクリプトを使用してデフォルト設定でルートマスターサーバーを構成する方法を説明します。ルートマスターサーバーは、次のデフォルトを使用します。

- セキュリティレベル 2 (DES) - NIS+ セキュリティの最高レベル
- NIS 互換を OFF に設定 (NIS 互換に設定する方法も記述)
- ネームサービス情報の情報源としてシステム情報ファイル (`/etc`) または NIS マップ
- NIS+ グループとして `admin.domainname`

注 - `nisserver` スクリプトは、ルートマスターサーバーを設定するときに、ネームサービススイッチファイルを NIS+ 用に変更します。`/etc/nsswitch.conf` ファイルは後で変更できます。ネームサービススイッチについては、[第 1 章](#)を参照してください。

`nisserver` を実行してルートサーバーを設定するための前提条件

ルートマスターサーバーにしたいマシンの `/etc/passwd` ファイルに `root` のエントリがあるかどうかを確認します。

`nisserver` を実行する前に、次の情報が必要になります。

- マシン (ルートマスターサーバーにする) のスーパーユーザーのパスワード
- 新しいルートドメイン名。ルートドメイン名は少なくとも 2 つの要素 (ラベル) で構成され、その末尾にドット (ピリオド) が打たれていなければならない (例: `something.com`)。最後 (右端) のラベルは自由につけられますが、インターネット上の互換性を維持するためには、[表 4-3](#) に示されるようなインターネットの組織名、または 2、3 文字の地域識別子 (日本であれば `.jp`) をつける必要があります。

表 4-3 インターネットの組織ドメイン

ドメイン	目的
com	営利団体
edu	教育機関
gov	行政機関

表 4-3 インターネットの組織ドメイン (続き)

ドメイン	目的
mil	軍事組織
net	主要ネットワークサポートセンター
org	非営利団体
int	国際組織

次の例では、ルートマスターサーバーに指定するマシンは master1 で、doc.com. が新しいルートドメインになります。

注 - またドメイン名とホスト名が同じにならないようにします。たとえば、ルートドメインに doc.com. という名前を付けている場合、ドメイン内で使用するマシンには doc という名前は付けしないでください。同様に、home というホスト名がある場合には、ドメイン名に home を使用できません。これは、サブドメインについても同様です。たとえば、すでにホスト名として west がある場合には、sales.west.doc.com というサブドメインを作成することはできません。

▼ ルートマスターサーバーを作成する方法

1. スーパーユーザーの **PATH** 変数に **/usr/lib/nis** を追加します。
このパスを root の .cshrc または .profile ファイルに追加するか、または変数を直接設定します。

2. **DES** 認証を使用する場合には、オプションで、**Diffie-Hellman** キー長を指定します。

640 ビットの Diffie-Hellman キーとデフォルトの 192 ビットのキーを使用するには、以下を入力します。

```
nisauthconf dh640-0 des
```

640 ビットのキーだけを許可し、192 ビットのキーを拒否するには、以下を入力します。

```
nisauthconf dh640-0
```

3. 次のコマンドをスーパーユーザー (**root**) として入力し、ルートマスターサーバーを構成します。

-r オプションはルートマスターサーバーを構成することを示します。-d オプションは NIS+ ドメイン名を指定します。

```
master1# nisserver -r -d doc.com.
This script sets up this machine "master1" as a NIS+ root master
server for domain doc.com.
Domain name : doc.com.
```



```
NIS+ group : admin.doc.com.  
NIS (YP) compatibility : OFF  
Security level : 2=DES
```

Is this information correct? (type 'y' to accept, 'n' to change)

「NIS+ グループ」は、doc.com. ドメイン (ドメイン名は常にピリオドで終了する) の情報を変更する権限を持つユーザーのグループです。NIS+ グループは、ドメインを削除することもできます。NIS+ グループのデフォルト名は、*admin.domainname* です。この名前の変更方法については、98 ページの「[誤った情報を変更する方法](#)」を参照してください。

「NIS 互換」とは、NIS+ サーバーが NIS クライアントからの情報要求を受け付けるかどうかを意味します。デフォルト設定の OFF に設定されている場合、NIS+ サーバーは NIS クライアントからの要求を処理しません。ON に設定されている場合、NIS+ サーバーはこの要求を処理します。このスクリプトでは、NIS 互換設定を変更できます。98 ページの「[誤った情報を変更する方法](#)」を参照してください。

注 - このスクリプトは、マシンを NIS+ セキュリティの最高レベルであるセキュリティレベル 2 で設定します。このスクリプトを使用する場合、セキュリティレベルを変更できません。スクリプトが終了した後、適切な NIS+ コマンドでセキュリティレベルを変更できます。セキュリティレベルの変更の詳細については、*rpc.nisd* のマニュアルページを参照してください。

4. **y** を入力します (画面の情報が正しい場合)。

n を入力すると、スクリプトは正しい情報の入力を要求します。n を入力した場合の操作については、98 ページの「[誤った情報を変更する方法](#)」を参照してください。

```
Is this information correct? (type 'y' to accept, 'n' to change)  
y  
This script will set up your machine as a root master server for  
domain doc.com. without NIS compatibility at security level 2.  
Use "nisclient -r" to restore your current network service environment.  
Do you want to continue? (type 'y' to continue, 'n' to exit the script)
```

5. **y** を入力して、NIS+ の構成を続けます。

n を入力するとスクリプトは問題なく終了します。y を選んだ後でスクリプトの実行中にスクリプトを中断した場合、スクリプトは動作を停止し、それまでに作成された構成内容はそのまま残されます。スクリプトは自動回復や後始末は行いません。このスクリプトはいつでも再実行できます。

```
Do you want to continue? (type 'y' to continue, 'n' to exit the script)  
y  
setting up domain information "doc.com." ...  
setting up switch information ...  
running nisinit ...  
This machine is in the doc.com. NIS+ domain.  
Setting up root server ...  
All done.
```

```
starting root server at security level 0 to create credentials...
running nissetup ...
(creating standard directories & tables)
org_dir.doc.com. created
Enter login password:
```

nissetup(1M) コマンドは、各 NIS+ テーブルのディレクトリを作成します。

6. プロンプトに対してマシンの **root** パスワードを入力し、**Return** キーを押します。
この例の場合、ユーザーは master1 マシンの root パスワードを入力しています。

```
Wrote secret key into /etc/.rootkey
setting NIS+ group to admin.doc.com. ...
restarting root server at security level 2 ...
This system is now configured as a root server for domain doc.com.
You can now populate the standard NIS+ tables by using the
nispopulate or /usr/lib/nis/nisaddent commands.
```

これでルートマスターサーバーが構成され、NIS+ の標準テーブルを生成する用意が整いました。続けてテーブルを生成する場合は、[100 ページの「NIS+ テーブルの生成 \(populate\)」](#)に進みます。

▼ 誤った情報を変更する方法

上記の[手順 4](#)で返された情報の一部あるいは全部が誤っていたために n を入力した場合、次のように表示されます。

```
Is this information correct? (type 'y' to accept, 'n' to change)
n
Domain name: [doc.com.]
```

1. ドメイン名が正しい場合は **Return** キーを押します。誤っている場合は、正しいドメイン名を入力して **Return** キーを押します。

この例では、**Return** キーを押して、目的のドメイン名が doc.com. であることを確定しました。次に、このスクリプトは NIS+ グループ名の確認を要求します。

```
Is this information correct? (type 'y' to accept, 'n' to change)
n
Domain name: [doc.com.]
NIS+ group: [admin.doc.com.]
```

2. **NIS+ グループ**が正しい場合は **Return** キーを押します。間違っている場合、正しい **NIS+ グループ名**を入力して **Return** キーを押します。

この例では、名前を変更しました。次に、スクリプトは NIS 互換性の入力を要求します。

```
NIS+ group: [admin.doc.com.] netadmin.doc.com.
NIS (YP) compatibility (0=off, 1=on): [0]
```

3. **NIS 互換性**を必要としない場合は **Return** キーを押します。互換性が必要な場合は **1** を入力して **Return** キーを押します。

この例では、Return キーを押して、NIS 互換の状態が正しいことを確認しました。スクリプトは、情報が正しいかどうかを再度尋ねてきます。

注 – このサーバーを NIS 互換に選択した場合、この選択を有効にするには、ファイルを編集して、rpc.nisd デーモンを再起動する必要があります。詳細については、112 ページの「クライアントを NIS+ サーバーとして構成する方法」を参照してください。

```
NIS (YP) compatibility (0=off, 1=on): [0]
Domain name : doc.com.
NIS+ group : netadmin.doc.com.
NIS (YP) compatibility : OFF
Security level : 2=DES
Is this information correct? (type 'y' to accept, 'n' to change)
```

情報が正しい場合、96 ページの「ルートマスターサーバーを作成する方法」の手順 3 に移ります。正しい情報となるまで、n を繰り返し選択できます。

▼ Multihomed NIS+ ルートマスターサーバーの設定方法

Multihomed NIS+ サーバーを設定する手順は、単一インタフェースサーバーの設定手順と同じです。唯一の相違点は、ローカルの /etc/hosts ファイル、/etc/inet/ipnodes ファイルと NIS+ の hosts テーブルおよび ipnodes テーブル内に定義する必要があるインタフェースが、単一インタフェースサーバーよりも多いという点です。ホスト情報を定義したら、nisclient と nisserver スクリプトを使用して Multihomed NIS+ サーバーを設定します。Multihomed 複製サーバーの設定の詳細は、117 ページの「Multihomed NIS+ 複製サーバーの設定方法」を参照してください。



注意 – Multihomed NIS+ サーバーを設定する場合は、サーバーの主体名はシステムのノード名と同じにする必要があります。これは、Secured RPC と nisclient を同時に使用する際の必要条件です。

- Secured RPC は、ノード名を使用して認証に必要なネット名を作成する
- nisclient は、主体名を使用してクライアントの資格を作成する

ノード名と主体名が異なる場合、Secured RPC 認証は適正に動作できず、その結果 NIS+ で問題が発生します。

次に、NIS+ ルートマスターサーバーの設定手順を示します。

1. ルートマスター上で、サーバーのホスト情報を /etc/hosts ファイルまたは /etc/inet/ipnodes ファイル内に追加します。

たとえば、3つの Ethernet インタフェースを装備した *hostA* システムの場合、*/etc/hosts* ファイルには次のように入力します。

```
127.0.0.1 localhost loghost
192.168.10.x hostA hostA-10 hostA-eri0
192.168.11.y hostA hostA-11 hostA-eri1
192.168.12.z hostA hostA-12
```

2. **nisserver** を使用して、サーバーを **multihomed NIS+** ルートサーバーとして設定します。

```
hostA# nisserver -r -d sun.com
```

上記の例では、*sun.com* はルートドメイン名を表しています。ルートドメイン名を指定して **nisserver** コマンドを実行してください。

multihomed NIS+ ルートサーバーの設定が完了したら、このあとの設定手順は、単一インタフェースサーバーの設定手順とまったく同じです。

NIS+ テーブルの生成 (populate)

ルートマスターサーバーの構成が終了すると、ほかのネットワークサービスの情報から標準の NIS+ テーブルを生成できます。この節では、**nispopulate(1M)** スクリプトをデフォルトの設定で使用して、ファイルまたは NIS マップのデータからルートマスターサーバーのテーブルを生成する方法を説明します。このスクリプトは次のものを使用します。

- 前の例で作成されたドメイン (*doc.com.*)
- ネットワークサービスのソースとしてシステム情報ファイルまたは NIS マップ
- 標準の NIS+ テーブル: *auto_master*、*auto_home*、*ethers*、*group*、*hosts*、*networks*、*passwd*、*protocols*、*services*、*rpc*、*netmasks*、*bootparams*、*netgroup* および *aliases*

注 - テーブルの情報源がファイルである場合、*shadow* ファイルの内容が *passwd* ファイルの内容とマージされて *passwd* テーブルが作成されます。*shadow* テーブルは作成されません。

nispopulate を実行してルートサーバーテーブルを生成するための前提条件

nispopulate スクリプトを実行するには、次の条件を満たしている必要があります。

- データを読み込むローカルの /etc 内の各ファイルまたは NIS マップを表示します。不正なエントリがないことを確認します。正しいデータが、所定の場所に正しい書式で記録されていることを確認します。エントリのうち、古いもの、無効なもの、破損しているものは削除します。また、不完全なエントリや一部のみのエントリも削除します。構成完了後は、いつでもエントリを追加できます。必要なエントリをあとから追加する方が、不完全なエントリや破損しているエントリを読み込むとより簡単です。
- ファイル内の情報は、ロード先のテーブルに合った書式で書かれていなければなりません。対応する NIS+ テーブルに変換するときに使用するテキストファイルの書式については、第 9 章を参照してください。
- ドメイン名とホスト名が同一でないことを確認します。ドメインとホストで同じ名前は使用しないでください。たとえば、sales ドメインがある場合は、ホスト名に sales を使用することはできません。同様に、home というホスト名がある場合には、ドメイン名に home を使用できません。これは、サブドメインについても同様です。たとえば、マシンに west という名前を付けている場合、sales.west.doc.com というサブドメインを作成しないでください。
- ホスト名のドットと下線はすべて削除します。NIS+ では、ドット (ピリオド) をマシン名とドメインの間、および親ドメインとサブドメインの間の区切りに使用するため、ドットをマシン名に使うことはできません。DNS ではホスト名に下線を使うことを認めていませんので、ホスト名に下線を使うことはできません。nispopulate スクリプトを実行する前に、ホスト名で使用されているドットを必ず削除してください。ドットをハイフンに置き換えるのも 1 つの手です。たとえば、sales.alpha というマシン名を付けることはできません。この名前は、sales-alpha に置き換えることができます (使用可能なホスト名の詳細については、hosts のマニュアルページを参照してください)。
- ネットワークをはじめて設定する場合、多くのネットワーク情報をすでにどこかに格納しているということはおそらくないでしょう。このような場合、まずネットワーク情報を集め、「入力ファイル」に手入力する必要があります。このファイルは、基本的に /etc 内のファイルに対応するものです。
- 安全のため、/etc 内のファイルのコピーを作成します。実際のファイルは使用せずに、作成したコピーを使用してテーブルを生成してください。たとえば、この例では /nisplusfiles というディレクトリ内のファイルを使用します。
- コピーした NIS テーブルファイルのうち、passwd、shadow、aliases、hosts の各ファイルには、名前空間全体に分散させるとセキュリティ上問題がある項目があるので、それらを編集します。たとえば、次に示す行をローカル側の passwd ファイルのコピーから削除して、名前空間からはそれらの情報にアクセスできないようにします。

```
root:x:0:1:0000-Admin(0000):/:/sbin/sh
daemon:x:1:3:0000-Admin(0000):/:
```

```
bin:x:3:5:0000-Admin(0000):/usr/bin:
sys:x:3:3:0000-Admin(0000):/:
```

```
adm:x:4:4:0000-Admin(0000):/var/adm:
lp:x:78:9:0000-lp(0000):/usr/spool/lp:
smtp:x:0:0:mail daemon user:/:
```

```
uucp:x:5:5:0000-uucp(0000):/usr/lib/uucp:
nuucp:x:7:8:0000-uucp (0000):/var/spool/uucppublic:/usr/lib/uucp/uucico
```

```
listen:x:22:6:Network Admin:/usr/net/nls
nobody:x:60000:60000:uid no body:/:
noaccess:x:60002:60002:uid no access:/:
```

- ドメインがすでに構成され、そのマスターサーバーが実行されている
- ドメインのサーバーには、新しいテーブル情報を収容できるだけの十分なディスク空き領域が必要です。
- NIS+ 主体 (適切な資格を持つクライアント) としてログインし、指定されたドメイン内の NIS+ テーブルに対する書き込み権を保持する必要があります。この例では、マシン master1 上の root ユーザーでなければなりません。

必要な情報

ファイルから生成する場合、次の情報が必要です。

- 新しい NIS+ ドメイン名
- データが転送される、適切に編集されたテキストファイルのパス
- ルートパスワード

NIS マップから生成する場合、次の情報が必要です。

- 新しい NIS+ ドメイン名
- NIS ドメイン名
- NIS サーバー名
- NIS サーバーの IP アドレス
- ルートパスワード

注 - NIS ドメインの名前は大文字と小文字を区別しますが、NIS+ ドメインの名前は区別しません。

▼ ルートマスターサーバーのテーブルを生成する方法

1. 「a」または「b」を実行してルートマスターサーバーテーブルを生成し、[手順 2](#)に進みます。

「a」では、ファイルからテーブルを生成する方法を示します。「b」では、NIS マップからテーブルを生成する方法を示します。これらのコマンドはスクロールできるウィンドウ内で実行します。そうしないと、スクリプトの出力がスクロールされず読めないことがあります。

注 - システム上の /tmp 領域が不足する場合、nispopulate スクリプトは異常終了することがあります。これを防止するには、環境変数 TMPDIR に別のディレクトリを設定します。TMPDIR に有効なディレクトリが設定されていない場合、スクリプトは /tmp ディレクトリを使用します。

- a. 次のように入力して、ファイルからテーブルを生成します。

```
master1# nispopulate -F -p /nis+files -d doc.com.
NIS+ domain name : doc.com.
Directory Path : /nis+files
Is this information correct? (type 'y' to accept, 'n' to change)
-F オプションは、テーブルがデータをファイルから取り出すことを示します。
-p オプションには、ソースファイルのディレクトリ検索パスを指定します
(この例では、/nis+files を指定します)。-d オプションは NIS+ ドメイン名
を指定します (この例では、doc.com. を指定します)。

NIS+ 主体のユーザーは root です。この例では、はじめてルートマスターサー
バーのテーブルを生成することになるため、この作業はスーパーユーザーとし
て実行しなければなりません。nispopulate スクリプトは、NIS+ 管理グルー
プのすべてのメンバーの資格を追加します。
```

- b. 次のように入力して、NIS マップからテーブルを生成します。

```
master1# nispopulate -Y -d doc.com. -h salesmaster -a 130.48.58.111
-y sales.doc.com.
NIS+ domain name : doc.com.
NIS (YP) domain : sales.doc.com.
NIS (YP) server hostname : salesmaster
Is this information correct? (type 'y' to accept, 'n' to change)
-Y オプションは、テーブルがデータを NIS マップから取り出すことを示しま
す。-d オプションは NIS+ ドメイン名を指定します。-h オプションは、NIS
サーバーのマシン名を指定します。salesmaster は NIS サーバーの名前の例
です。サンプルドメインを作成する際は、salesmaster の代わりに、サイトでの
実際の NIS サーバー名を指定してください。-a オプションは、NIS サーバー
の IP アドレスを指定します。130.48.58.111 はアドレスの例です。サンプル
ドメインを作成する際は、サイトでの実際の NIS サーバーの IP アドレスを指
定してください。-y オプションは、NIS ドメイン名を指定します。この例で
は、ドメイン名は sales.doc.com. です。サンプルドメインを作成する際
は、sales.doc.com. の代わりに、サイトでの実際の NIS ドメインの NIS ドメイ
ン名を指定してください。

NIS+ 主体のユーザーは root です。この例では、はじめてルートマスターサー
バーのテーブルを生成することになるため、この作業はスーパーユーザーとし
て実行しなければなりません。nispopulate(1M) スクリプトは、NIS+ 管理
グループのすべてのメンバーの資格を追加します。
```

2. **y** を入力します (画面に表示された情報が正しい場合)。

n を入力すると、スクリプトは正しい情報の入力进行を要求します。情報が誤っている場合の操作は、[98 ページの「誤った情報を変更する方法」](#)を参照してください。

- 手順1の「a」を実行すると、次のように表示されます。

```
Is this information correct?  
(type 'y' to accept, 'n' to change)  
y
```

```
This script will populate the following NIS+ tables for domain doc.com. from  
the files in /nis+files: auto_master auto_home ethers group hosts networks  
passwd protocols services rpc netmasks bootparams netgroup aliases shadow  
**WARNING: Interrupting this script after choosing to continue may leave  
the tables only partially populated. This script does not do any automatic  
recovery or cleanup.  
Do you want to continue? (type 'y' to continue, 'n' to exit this script)
```

- 手順1の「b」を実行すると、次のように表示されます。

```
Is this information correct? (type 'y' to accept, 'n' to change)  
y  
This script will populate the following NIS+ tables for domain doc.com. from the  
NIS (YP) maps in domain sales: auto_master auto_home ethers group hosts networks  
passwd protocols services rpc netmasks bootparams netgroup aliases  
**WARNING: Interrupting this script after choosing to continue may leave the  
tables only partially populated. This script does not do any automatic recovery  
or cleanup.  
Do you want to continue? (type 'y' to continue, 'n' to exit this script)
```

3. y を入力して、テーブルの生成を続けます。

n を入力するとスクリプトは問題なく終了します。y を選択した後で、スクリプトの実行中にスクリプトを中断した場合、スクリプトは動作を停止し、テーブルは一部だけが生成された状態になることがあります。スクリプトは自動回復や後始末は行いません。このスクリプトは安全に再実行できますが、テーブルは最新の情報で上書きされます。

- ファイルからテーブルを生成する場合、スクリプトが hosts 情報と passwd 情報に基づいてホストとユーザーの資格を作成することを示す、次のようなメッセージが表示されます。

```
Do you want to continue? (type 'y' to continue, 'n' to exit this script)  
y  
populating auto_master table from file /nis+files/auto_master  
... auto_master table done.  
populating auto_home table from file /nis+files/auto_home  
... auto_home table done.  
Credentials have been added for the entries in the hosts and passwd table(s).  
Each entry was given a default network password (also known as a Secure-  
RPC password). This password is: nisplus  
Use this password when the nisclient script requests the network password.  
Done!
```

Secure RPC パスワード (上の例では nisplus) は必ず控えておき、忘れないようにしてください。ネットワークパスワードまたは Secure RPC パスワードを求められたら、ここで控えておいたパスワードを入力します。

スクリプトは、必要なすべてのファイルを検索し、使用できるファイルからすべてのテーブルを生成します。

- NIS マップからテーブルを生成する場合、スクリプトが `hosts` と `passwd` の情報に基づいてホストとユーザーの資格を作成する際に、次の内容が表示されます。

```
Do you want to continue? (type 'y' to continue, 'n' to exit this script)
y
populating auto_master table from sales.doc.com. NIS(YP) domain...
auto_master table done.
populating auto_home table from file sales.doc.com. NIS(YP) domain...
auto_home table done.
....
Credentials have been added for the entries in the hosts and passwd table(s).
Each entry was given a default network password (also known as a Secure-RPC password).
This password is: nisplus
Use this password when the nisclient script requests the network password.
Done!
```

Secure RPC パスワード (上の例では `nisplus`) は必ず控えておき、忘れないようにしてください。ネットワークパスワードまたは Secure RPC パスワードを求められたら、ここで控えておいたパスワードを入力します。

これですべてのテーブルが生成されました。 `parse error` という警告が表示されるかもしれませんが無視してかまいません。これらのエラーは、NIS+ が特定の NIS マップのフィールドで空の値または予期せぬ値を見つけたことを示します。必要に応じて、スクリプト終了後、データを検証してください。

4. ルートドメインの管理グループに自分自身とほかの管理者を追加します (必要に応じて)。

たとえば、自分自身のログイン ID が `topadm` で、ほかの管理者の ID が `secondadmin` の場合、次のように入力します。

```
master1# nisgrpadm -a admin.doc.com. topadm.doc.com. secondadm.doc.com.
Added "topadm.doc.com." to group "admin.doc.com.".
Added "secondadm.doc.com." to group "admin.doc.com.".
```

上記の `nisgrpadm -a` コマンドの引数 `admin.doc.com.` は、グループ名を表し、最初に記述しなければなりません。残りの 2 つの引数は管理者名です。

注 - この時点で管理グループにユーザーを追加したい場合にだけ、この手順が必要です。ルートサーバーに管理者を追加するには、この時点で行います。NIS+ を構成した後でも、管理グループにユーザーを追加できます。

この手順を実行するには、ほかの管理者がデフォルトパスワードを変更するまで待つ必要はありません。ただし、管理グループにほかの管理者を追加する場合は、事前に `passwd` テーブルに管理者が登録されている必要があります。管理グループのメンバーは、ドメインに自分自身を追加しなければ、NIS+ 主体として行動できません。ユーザーの初期設定については、[110 ページの「NIS+ ユーザーを初期設定する方法」](#)を参照してください。また、新規にメンバーが登録されても、管理グループに対する古いキャッシュが破棄された後でなければ有効になりません。

5. 次のコマンドを入力して、ドメインのチェックポイントを実行します。

```
master1# nisping -C doc.com.  
Checkpointing replicas serving directory doc.com.  
Master server is master1.doc.com.  
Last update occurred at date  
Master server is master1.doc.com.  
checkpoint scheduled on master1.doc.com.
```

この手順によって、そのドメインをサポートする全サーバーは、初期設定 (.log) ファイルから新しい情報をテーブルのディスク上のコピーに転送します。ルートドメインの設定が終了したばかりであるため、このルートドメインにはまだ複製が存在せず、この手順はルートマスターサーバーにだけ影響を与えます。



注意 - スワップ領域またはディスク領域が不足している場合、サーバーは適切にチェックポイントを実行できませんが、その旨の通知は行われません。スワップ領域またはディスク領域が不足していないか確認するには、niscat コマンドでテーブルの内容を表示します。たとえば、rpc テーブルの内容をチェックするには、次のように入力します。

```
master1# niscat rpc.org_dir  
rpcbind rpcbind 100000  
rpcbind portmap 100000  
rpcbind sunrpc 100000
```

スワップ領域が不足していると「チェックポイントを実行できない」という旨のメッセージではなく、次のエラーメッセージが表示されます。

```
can't list table: Server busy, Try Again.
```

このメッセージには明示されていませんが、これはスワップ領域の不足を示します。スワップ空間を増やし、このドメインに再びチェックポイントを実行します。

NIS+ クライアントマシンの設定

ルートマスターサーバーのテーブルがファイルまたは NIS マップから生成されると、NIS+ クライアントマシンを初期設定できます。ルートマスターサーバーは自分のドメインの NIS+ クライアントであるため、ルートマスターサーバーをこれ以上初期設定する必要はありません。この節では、デフォルト設定の nisclient スクリプトを使用して、NIS+ クライアントを初期設定する方法を説明します。次のスクリプトを使用します。

- 前の例で使用したドメイン doc.com.
- 前の例で、nispopulate スクリプトによって作成された Secure RPC パスワード (ネットワークパスワードとも呼ばれる。デフォルトは nisplus)

注 - 107 ページの「新しいクライアントマシンを初期設定する方法」で使用する `-i` オプションでは、DNS を使用してホスト名を検索するような NIS+ クライアントを構成しません。DNS を使用する場合には、ネームサービススイッチファイルの中でクライアント用に、DNS を明確に指定しなければなりません。

`nisclient` を実行してクライアントマシンを設定するための前提条件

`nisclient` スクリプトを実行するには、次の条件を満たしている必要があります。

- ドメインがすでに構成され、そのマスターサーバーが実行されている
- ドメインのマスターサーバーのテーブルが生成されている (少なくとも、`hosts` テーブルまたは `ipnodes` テーブルには新しいクライアントマシンのエントリが必要)
- NIS+ クライアントとなるマシンにスーパーユーザーとしてログインしている。この例では、新しいクライアントマシン名は `client1`

必要な情報

`nisclient` を実行するには、次の情報が必要です。

- ドメイン名
- デフォルトの Secure RPC パスワード (`nisplus`)
- クライアントとなるマシンの `root` パスワード
- クライアントのホームドメインにある NIS+ サーバーの IP アドレス
- DES 認証を使用する場合には、マスターサーバー上で、Diffie-Hellman キー長が使用される。`nisauthconf` を使用して、マスターサーバー上で Diffie-Hellman キー長を確認する

▼ 新しいクライアントマシンを初期設定する方法

1. DES 認証を使用する場合には、オプションで、**Diffie-Hellman** キー長を指定します。

マスターサーバー上で以下を入力します。

```
nisauthconf
```

クライアントマシン上で `nisauthconf` コマンドを実行するときは、上記コマンドの出力を引数として使用します。たとえば、マスターサーバー上で `nisauthconf` を実行すると

```
dh640dh-0 des
```

と出力される場合には、クライアントマシン上で次のコマンドを入力します。

```
nisauthconf dh640dh-0 des
```

2. 次のコマンドを入力して、新しいクライアントマシン上で新しいクライアントを初期設定します。

-i オプションはクライアントを初期設定します。-d オプションは新しい NIS+ ドメイン名を指定します。ドメイン名が指定されていない場合、デフォルトは現在のドメイン名です。-h オプションは NIS+ サーバーのホスト名を指定します。

```
client1# nisclient -i -d doc.com. -h master1
Initializing client client1 for domain "doc.com.".
Once initialization is done, you will need to reboot your machine.
Do you want to continue? (type 'y' to continue, 'n' to exit this script)
```

3. **y** を入力します。

n を入力するとスクリプトが終了します。クライアントの /etc/hosts ファイルまたは /etc/inet/ipnodes ファイルにルートサーバーのエントリがない場合、スクリプトはルートサーバーの IP アドレスの入力を指示します。

```
Do you want to continue? (type 'y' to continue, 'n' to exit this script)
y
Type server master1's IP address:
```

4. 正しい IP アドレスを入力して、**Return** キーを押します。

この例では、アドレス 123.123.123.123 を使用します。

```
Type server master1's IP address: 123.123.123.123
setting up the domain information...
setting up the name service switch information...
At the prompt below, type the network password (also known as the
Secure-RPC password) that you obtained either from your administrator or
from running the nispopulate script.
Please enter the Secure-RPC password for root:
```

5. **Secure RPC** パスワード (ネットワークパスワード) がルートのログインパスワードと異なる場合のみ、**Secure RPC** パスワードを入力します。

この例では、デフォルトの nisplus を使用します。

パスワードは画面に表示されません。入力を誤った場合、正しいパスワードを入力するよう要求されます。2 回目も入力を誤った場合、スクリプトが終了し、前回のネットワークサービスが復元されます。この場合、スクリプトを再度実行してください。

```
Please enter the login password for root:
```

6. このクライアントマシンの **root** パスワードを入力します。

パスワードは画面に表示されません。ネットワークパスワードと root のログインパスワードが同じ場合、root のログインパスワードの入力要求はありません。

root パスワードを入力すると、このマシンの資格が変更されます。ここで、このマシンでは RPC パスワードと root パスワードが同じになります。

```
Please enter the login password for root:
Wrote secret key into /etc/.rootkey
```

```
Your network password has been changed to your login one.  
Your network and login passwords are now the same.  
Client initialization completed!!  
Please reboot your machine for changes to take effect.
```

7. クライアントマシンを再起動します。

ここで行なった変更は、マシンを再起動すると有効になります。

以上の手順により、この NIS+ クライアントマシンのユーザーを NIS+ ドメインに追加できるようになりました。

クライアントマシンの追加作成

これまでに説明したクライアントの初期設定手順を、必要な各マシンで繰り返します。ほかのドメインのクライアントを初期設定するには、ドメイン名とマスターサーバー名を適宜変更して、上記の手順を繰り返します。

この章で説明する NIS+ ドメインのサンプルでは、ドメイン `doc.com.` 内の 4 台のクライアントを初期設定すると想定しています。そして、このうち 2 台のクライアントをルート以外の NIS+ サーバーとして構成し、3 番目のクライアントを `doc.com.` ドメインのルートマスターサーバーのルート複製サーバーとして構成します。

注 - システムをサーバーにするには、システムをどの種類のサーバーにするにしても、事前にシステムを親ドメインのクライアントにしておく必要があります。

NIS+ クライアントユーザーの初期設定

マシンが NIS+ クライアントになったら、そのマシンのユーザーは自分自身を NIS+ ドメインに追加しなければなりません。ユーザーをドメインに追加することは、Secure RPC パスワードをそのユーザーのログインパスワードに変更することを意味します。実際には、ユーザーのパスワードと Secure RPC パスワードが同じになります。この手順では `nisclient(1M)` スクリプトを使用します。

`nisclient` を実行してクライアントユーザーを初期設定するための前提条件

`nisclient` スクリプトを使用してユーザーを初期設定するには、次の条件を満たしている必要があります。

- ドメインがすでに構成され、そのマスターサーバーが実行されている

- ドメインのマスタースerverのテーブルが生成されている。少なくとも、`hosts` テーブルには新しいクライアントマシンのエントリが必要
- ドメイン内のクライアントマシンが初期設定されている
- クライアントマシンに「ユーザー」としてログインしている (この例では、ユーザー名は `user1`)
- DES 認証を使用する場合には、クライアントマシンでマスタースerverと同じ Diffie-Hellman キー設定を使用している

必要な情報

`nisclient` を実行するには、次の情報が必要です。

- ユーザーのログイン名 (この例では `user1`)
- デフォルトの Secure RPC パスワード (この例では `nisplus`)
- NIS+ クライアントになるユーザーのログインパスワード

▼ NIS+ ユーザーを初期設定する方法

1. NIS+ クライアントになるユーザーとしてログインして、次のように `nisclient` コマンドを入力します。

```
user1prompt% nisclient -u
At the prompt below, type the network password (also known as the
Secure-RPC password) that you obtained either from your administrator
or from running the nispopulate script.
Please enter the Secure-RPC password for user1:
```

2. Secure RPC パスワード (この例では **`nisplus`**) を入力します。
パスワードは画面に表示されません。

```
Please enter the login password for user1:
```

3. ユーザーのログインパスワードを入力し、**Return** キーを押します。
パスワードは画面に表示されません。

```
Your network password has been changed to your login one.
Your network and login passwords are now the same
```

これで、このユーザーは NIS+ クライアントになりました。全ユーザーを NIS+ クライアントにする必要があります。

NIS+ サーバーの設定

初期設定が終了したら、どのクライアントマシンも次の NIS+ サーバーに変更できます。

- ルート複製サーバー (ルートマスターサーバー上にある NIS+ テーブルのコピーを格納する)
- ルートドメインのサブドメインのマスターサーバー
- ルートドメインのサブドメインのマスターサーバーの複製サーバー

注 - NIS+ マスタールートサーバーは 1 つしか持てません。ルート NIS+ サーバーは特殊な NIS+ サーバーです。この節ではルートマスターサーバーの構成方法は説明しません。詳細は、[95 ページの「NIS+ ルートサーバーの設定」](#)を参照してください。

サーバーを構成する場合、次の 3 つの設定があります。

- NIS と互換性なし
- NIS と互換性あり
- NIS との互換性と DNS 転送あり - NIS+ 名前空間に Sun OS 4.x クライアントを配置する場合は、DNS 転送を設定する必要があります。

サーバーとその複製は、NIS 互換についての設定が同じでなければなりません。設定が異なる場合、ネットワーク情報を受信するために NIS 互換の設定を必要とするクライアントは、必要なサーバーまたは複製サーバーを使用できなければ、この情報を受信できません。

NIS+ サーバーを設定するための前提条件

svcadm を使用してコマンド行から NIS+ サービスを起動するには、次の条件が満たされていることを確認してください。

- ドメインがすでに構成され、そのマスターサーバーが実行されている
- ドメインのマスターサーバーのテーブルが生成されている。少なくとも、hosts テーブルには新しいクライアントマシンのエントリが必要
- ドメイン内のクライアントマシンが初期設定されている
- クライアントマシンにスーパーユーザーとしてログインしている (この例では、クライアントマシン名は client1)
- DES 認証を使用する場合には、クライアントマシンでマスターサーバーと同じ Diffie-Hellman キー設定を使用している

必要な情報

svcadm コマンドを使用して NIS+ サービスを起動するには、サーバーに変換するクライアントのスーパーユーザーパスワードが必要です。

クライアントを NIS+ サーバーとして構成する方法

クライアントをサーバーとして構成するには、次の手順のどれかを実行します。この手順では、サーバーと同じ名前のディレクトリを作成し、サーバーの初期設定ファイルを作成します。これらは /var/nis に置かれます。

注 - 同じドメイン内のすべてのサーバーは、NIS 互換の設定が同じでなければなりません。たとえば、マスターサーバーが NIS 互換である場合、その複製も NIS 互換でなければなりません。

▼ NIS と互換性のないサーバーを構成する方法

この手順を実行するには、スーパーユーザーになるか、同等の役割になる必要があります。

1. **/lib/svc/method/nisplus** ファイルを表示して、**-y** オプションが含まれていないことを確認します。

詳細については、89 ページの「NIS+ とサービス管理機能」を参照してください。

2. **NIS+ サービスを開始します。**

この手順を実行するには、スーパーユーザーになるか、同等の役割になる必要があります。

```
client1# svcadm enable /network/rpc/nisplus:default
```

これで、このサーバーはドメインのマスターまたは複製として指定できます。

▼ NIS と互換性のあるサーバーを構成する方法

この手順を実行するには、スーパーユーザーになるか、同等の役割になる必要があります。

1. サーバー上で **/lib/svc/method/nisplus** ファイルを編集して **-y** オプションを追加します。

詳細については、89 ページの「NIS+ とサービス管理機能」を参照してください。

2. **NIS+ サービスを開始します。**

この手順を実行するには、スーパーユーザーになるか、同等の役割になる必要があります。

```
client1# svcadm enable /network/rpc/nisplus
```


これで、このサーバーはドメインのマスターまたは複製として指定できます。

▼ DNS 転送機能と NIS との互換性を備えたサーバーを構成する方法

次の手順を実行して、DNS 転送と NIS との互換性の両方の機能を備えた NIS+ サーバーを構成します。SunOS 4.x クライアントをサポートするには、両方の機能が必要です。

この手順を実行するには、スーパーユーザーになるか、同等の役割になる必要があります。

1. サーバー上で **/lib/svc/method/nisplus** ファイルを編集して **-Y** オプションと **-B** オプションを追加します。
詳細については、[89 ページの「NIS+ とサービス管理機能」](#)を参照してください。

2. **NIS+ サービスを開始します。**
この手順を実行するには、スーパーユーザーになるか、同等の役割になる必要があります。

```
client1# svcadm enable /network/rpc/nisplus:default
```

これで、このサーバーはドメインのマスターまたは複製として指定できます。

サーバーの追加作成

これまでのクライアントをサーバーに変更する手順を、必要なクライアントマシンごとに繰り返します。

この章で説明する NIS+ サンプルドメインは、3 台のクライアントをサーバーに変更すると想定しています。そして、サーバーの 1 台をルート複製サーバーに、もう 1 台を新しいサブドメインのマスターに、そして 3 台目を新しいサブドメインのマスターの複製に構成します。

ルート複製サーバーの作成

NIS+ サービスを常に利用可能な状態にしておくため、ルート複製サーバーを少なくとも 1 つは作成しておくことをお勧めします。複製サーバーを作成すると複数のサーバーが存在することになり、要求の処理を分散させることができるため、ネットワーク要求の処理も高速化されます。

パフォーマンス上の理由から、1つのドメインに多くの複製サーバーを置くことはお勧めできません。ネットワークが複数のサブネット構成されている場合、あるいは広域ネットワーク (WAN) でリモートサイトに接続されている場合には、複製サーバーを追加すると良い場合があります。

- 「サブネット」。複数のサブネット構成されているドメインの場合、各サブネットに複製サーバーを少なくとも1つは作成することをお勧めします。そうしておけば、ネットワーク間の通信が一時的に途絶しても、接続が回復するまでの間、サブネットレベルの機能は維持されます。
- 「リモートサイト」。WANによりリモートサイトに接続されているドメインの場合、WAN接続の両側に複製サーバーを少なくとも1つは作成することをお勧めします。組織論的な見地からしても、同一のNIS+ドメインに物理的に離れた2つのサイトがあるのは意味のあることです。たとえば、ドメイン内のマスターサーバーとその複製サーバーがすべて一方のサイトに置かれている場合、そのサイトともう一方のサイトとの間のNIS+ネットワークトラフィックが増大するのは目に見えています。もう一方のサイトにも複製サーバーを置いておけば、ネットワークトラフィックが減るはずで

最適な複製サーバー数の決定方法については、[113 ページの「ルート複製サーバーの作成」](#)を参照してください。

[115 ページの「ルート複製サーバーを作成する方法」](#)では、マシン `client1` を `doc.com` ドメインのルート複製サーバーとして構成します。この手順では、NIS+ スクリプトの `nissserver` を使用します。[168 ページの「NIS+ コマンドを使って複製サーバーを構成する」](#)で説明しているように、NIS+ コマンドを使用して複製サーバーを構成することもできます。

nissserver を実行してルート複製サーバーを作成するための前提条件

`nissserver` を実行して複製サーバーを作成するには、次の条件を満たしている必要があります。

- ドメインがすでに構成され、マスターサーバーが稼働している
- マスターサーバーのテーブルが生成されている。少なくとも、`hosts` テーブルには新しいクライアントマシンのエントリが必要
- 新たに作成したサーバーをそのドメインのクライアントマシンとして初期設定している (具体的な手順については、[106 ページの「NIS+ クライアントマシンの設定」](#)を参照)
- 新たに作成した複製サーバー上でNIS+ サービス (`rpc.nisd`) を起動している (具体的な手順については、[111 ページの「NIS+ サーバーの設定」](#)を参照)
- ルートマスターサーバーにスーパーユーザーとしてログインしている。この例では、ルートマスターマシン名は `master1`

必要な情報

nisserver を実行するには、次の情報が必要です。

- ドメイン名
- クライアントマシン名 (この例では、client1)
- ルートマスターサーバーのスーパーユーザーのパスワード

▼ ルート複製サーバーを作成する方法

1. **NIS+** ドメインのルートマスターサーバー上でスーパーユーザー (**root**) として次のように入力して、ルート複製サーバーを作成します。

```
master1# nisserver -R -d doc.com. -h client1
This script sets up a NIS+ replica server for domain doc.com.
Domain name: :doc.com.
NIS+ server      : :client1
Is this information correct? (type 'y' to accept, 'n' to change)
-R オプションは、複製サーバーを構成することを示します。-d オプションは
NIS+ ドメイン名 (このサンプルでは doc.com.) を指定します。-h オプションは
ルート複製サーバーとなるクライアントマシン (この例では client1) を指定しま
す。
```

2. **y** を入力して操作を続けます。

n を入力すると、スクリプトは正しい情報の入力を要求します。n を入力した場合の操作については、[98 ページの「誤った情報を変更する方法」](#)を参照してください。

```
Is this information correct? (type 'y' to accept, 'n' to change)
y
This script will set up machine "client1" as an NIS+ replica server for domain
doc.com. without NIS compatibility. The NIS+ server daemon, rpc.nisd, must
be running on client1 with the proper options to serve this domain.
Do you want to continue? (type 'y' to continue, 'n' to exit this script)
```

3. **y** を入力して操作を続けます。

n を入力するとスクリプトは問題なく終了します。クライアントマシン上で rpc.nisd が動作していない場合、スクリプトは自動的に終了します。

```
Is this information correct? (type 'y' to continue, 'n' to exit this script)
y
The system client1 is now configured as a replica server for domain doc.com..
The NIS+ server daemon, rpc.nisd, must be running on client1 with the proper
options to serve this domain. ...
```

注 - この複製サーバーを NIS (YP) 互換モードで実行する場合は、`/lib/svc/method/nisplus` ファイルに `-Y` オプションを追加します。このファイルの変更が必要となるのは、ルート複製サーバーで NIS クライアント要求を実行する場合で、このサーバーが NIS 互換サーバーとして構成されていない場合に限られます。NIS 互換サーバーの作成方法については、112 ページの「クライアントを NIS+ サーバーとして構成する方法」を参照してください。NIS+ でのサービス管理機能コマンドの使用方法については、89 ページの「NIS+ とサービス管理機能」を参照してください。

4. (省略可能) 複製サーバーを NIS (YP) 互換モードで稼働させることができるように構成します。

複製サーバーを NIS 互換モードで稼働させるには、次の手順に従います。

- a. NIS+ サービスを停止します。

```
# svcadm disable /network/rpc/nisplus:<instance>
```

- b. サーバーの `/lib/svc/method/nisplus` ファイルを編集して `-Y` オプションを追加します。

- c. NIS+ サービスを再起動します。

```
# svcadm enable /network/rpc/nisplus:<instance>
```

5. 新たに作成した複製サーバーに名前空間データをロードします。

これには、次の 2 つの方法があります。

- NIS+ の保存と復元の機能を使用して、マスターサーバーのデータを保存し、作成した複製サーバーに復元します (こちらの方法をお勧めします)。具体的な手順については、170 ページの「名前空間データをロードする方法—`nisrestore` による方法」を参照してください。
- `nisping` を実行します。`nisping` を実行すると、マスターサーバーのすべての NIS+ データを複製サーバーに完全に再同期させることができます。しかし、大きな名前空間の場合、処理に長時間かかることがあります。その間、マスターサーバーはビジーになって応答が遅く、また作成した複製サーバーはまだ NIS+ 要求に応答できない状態になることがあります。具体的な手順については、172 ページの「名前空間データをロードする方法—`nisping` による方法」を参照してください。

名前空間データのロードが終了すれば、マシン `client1` は NIS+ ルート複製サーバーとなります。この新しいルート複製サーバーは、ルートドメインのクライアントからの要求を処理できます。これでドメインで使用可能なサーバーが 2 台となるため、要求を速く処理できます。

これらの手順に従って、複製サーバーを必要なだけ作ることができます。同じ手順を使ってサブドメインに複製サーバーを作ることもできます。

▼ Multihomed NIS+ 複製サーバーの設定方法

Multihomed NIS+ サーバーを設定する手順は、単一インタフェースサーバーの設定手順と同じです。唯一の相違点は、ローカルの `/etc/hosts` ファイル、`/etc/inet/ipnodes` ファイルと NIS+ の `hosts` テーブルおよび `ipnodes` テーブル内に定義する必要があるインタフェースが、単一インタフェースサーバーよりも多いという点です。ホスト情報を定義したら、`nisclient` と `nisserver` スクリプトを使用して Multihomed NIS+ サーバーを設定します。



注意 – Multihomed NIS+ サーバーを設定する場合は、サーバーの主体名はシステムのノード名と同じにする必要があります。これは、Secured RPC と `nisclient` を同時に使用する際の必要条件です。

- Secured RPC は、ノード名を使用して認証に必要なネット名を作成する
- `nisclient` は、主体名を使用してクライアントの資格を作成する

ノード名と主体名が異なる場合、Secured RPC 認証は適正に動作できず、その結果 NIS+ で問題が発生します。

次に、NIS+ ルートマスター以外のサーバーの設定手順を示します。以下の例では、ルートドメインの複製を作成しています。multihomed ルートサーバーの設定方法については、99 ページの「Multihomed NIS+ ルートマスターサーバーの設定方法」を参照してください。

1. サーバーのホスト情報を **hosts** ファイルまたは **ipnodes** ファイル内に追加します。

たとえば、3 つのインタフェースを装備した hostB システムの場合は、次のように入力します。

```
192.168.11.y hostB hostB-11
192.168.12.x hostB hostB-12
192.168.14.z hostB hostB-14
```

2. ルートマスターサーバー上で、**nispopulate** または **nisaddent** のどちらかを使用して、新しいホスト情報を **hosts** テーブルまたは **ipnodes** テーブル内にロードします。

たとえば、以下のようになります。

```
hostA# nispopulate -F -d sun.com hosts
```

この例では、`sun.com` は NIS+ ルートドメイン名を表しています。自分の NIS+ ルートドメイン名を指定して `nispopulate` コマンドを実行してください。

3. ルートマスターサーバー上で、**nisclient** スクリプトを使用して新しいクライアントの資格を作成します。

たとえば、以下のようになります。

```
hostA# nisclient -c -d sun.com hostB
```

この例では、sun.com はルートドメイン名を表しています。自分のルートドメイン名を指定して `nisclient` コマンドを実行してください。

4. ルート以外のマスターサーバー上で、新しいサーバーがまだ稼働していない場合は、**`nisclient`** を使用して新しいサーバーを起動し、このマシンを NIS+ クライアントとして初期化します。

たとえば、以下のようになります。

```
hostB# nisclient -i -d sun.com
```

この例では、sun.com はルートドメイン名を表しています。自分のルートドメイン名を指定して `nisclient` コマンドを実行してください。

5. ルートマスター上で、**`nisserver`** を使用してルートマスター以外のサーバーを設定します。

たとえば、以下のようになります。

```
hostA# nisserver -M -d eng.sun.com -h hostB.sun.com.
```

この例では、eng.sun.com は NIS+ ドメイン名を表し、hostB.sun.com は NIS+ サーバーの完全指定ホスト名を表しています。`nisserver` コマンドは、自分の NIS+ ドメイン名と NIS+ サーバーの完全指定ホスト名を指定して実行してください。

6. ルートマスターサーバー上で、**`nisserver`** を使用して複製サーバーを設定します。

たとえば、以下のようになります。

```
hostA# nisserver -R -d sun.com -h hostB.sun.com.
```

この例では、sun.com は複製サーバーを表し、hostB.sun.com は NIS+ サーバーの完全指定ホスト名を表しています。`nisserver` コマンドは、自分の複製サーバー名と NIS+ サーバーの完全指定ホスト名を指定して実行してください。

multihomed NIS+ 複製サーバーの設定が完了したら、このあとの設定手順は、単一インタフェースのサーバーの設定手順とまったく同じです。

サブドメインの作成

この節では、ルート以外の新しいドメイン (サブドメイン) のマスターサーバーの作成方法を説明します。この新しいドメインは doc.com. ドメインのサブドメインとなります。NIS+ の階層構造によって、組織構造に合わせたドメイン構造を作成できます。

この例では、マシン client2 が新しい sub.doc.com. ドメインのマスターサーバーに変更されます。ここでは NIS+ スクリプトの `nisserver` を使用します。

nisserver を実行してサブドメインを作成するための前提条件

nisserver を実行して、新しいサブドメインのマスターサーバーを作成するには、次の条件を満たしている必要があります。

- 親ドメインがすでに構成され、そのマスターサーバーが実行されている
- 親ドメインのテーブルが生成されている。少なくとも、hosts テーブルには新しいクライアントマシンのエントリが必要
- 親ドメイン内の新しいクライアントマシンが初期設定されている
- クライアント上で NIS+ サービス (rpc.nisd) が起動されている
- 新しいドメインを追加するための適切なアクセス権がある。この例では、親マスターサーバーにスーパーユーザーとしてログインする。この例では、親マスターマシン名は master1

必要な情報

nisserver を実行してサブドメイン用のマスターサーバーを作成するには、次の情報が必要です。

- クライアントマシン名 (この例では、client2)
 - 親マスターサーバーのスーパーユーザーパスワード
 - 新しいサブドメインの名前 – 親ドメインの名前を含んだ新しいドメイン名。書式は、「newdomain.rootdomain」という構文に従うこと
- [120 ページの「新しいルート以外のドメインを作成する方法」](#)では、新しいサブドメインを sub.doc.com. とします。

Solaris リリース 2.6 およびこれ以前のリリースでは、どんな NIS+ クライアントでも、そのサービスの提供先となるドメインの上位ドメインに置かれている限り、NIS+ マスターサーバーに変更できます。たとえば、ドメイン sales.doc.com. の NIS+ クライアントは、west.sales.doc.com. や alameda.west.sales.doc.com. などの下位ドメインにサービスを提供することができます。ただし、このクライアントは、ドメイン doc.com. にサービスを提供することはできません。これは、doc.com. が sales.doc.com. の上位ドメインであるためです。ただし、ルート複製サーバーはこのルールの唯一の例外であり、サービス提供先のドメインのクライアントです。

Solaris リリース 7 およびリリース 8 では、ドメインの NIS+ サーバーを、それがサービスしている同じドメイン内に置くことができます。ルート以外のサーバーは、設定されたドメイン内でサーバーとして動作します。つまり、ルート以外のサーバー上のアプリケーションを構成して、上位ドメインから渡される情報を使用することができます。

ただし、ルート以外のサーバーの資格は、上位ドメインになればなりません。ルート以外のサーバーの構成方法については、[120 ページの「新しいルート以外のドメインを作成する方法」](#)を参照してください。サーバーを正しく構成したあとで、ドメイン名をサーバーとして機能するドメインの名前に変更できます。nisinit の -k オプションおよび nisserver の -d オプションを参照してください。

▼ 新しいルート以外のドメインを作成する方法

1. **NIS+** ドメインのルートマスターサーバー上のスーパーユーザー (**root**) として次のように入力し、新しいルート以外のドメインマスターサーバーを作成します。

-M オプションは、新しいルート以外のドメインのマスターサーバーを作成することを示します。-d オプションは、ドメイン名 (この例では、sales.doc.com.) を指定します。-h オプションは、新しいドメインのマスターサーバーになるクライアントマシン (この例では、client2) を指定します。

```
master1# nisserver -M -d sales.doc.com. -h client2
This script sets up a non-root NIS+ master server for domain sales.doc.com.
Domain name : sales.doc.com.
NIS+ server : client2
NIS+ group : admin.sales.doc.com.
NIS (YP) compatibility : OFF
Security level : 2=DES
Is this information correct? (type 'y' to accept, 'n' to change)
```

新しいルート以外のドメインのマスターサーバーは、ルートサーバーと同じデフォルト値によって作成されます。NIS+ グループ、NIS (YP) 互換性、およびセキュリティレベルの詳細は、[96 ページの「ルートマスターサーバーを作成する方法」](#)を参照してください。

2. **y** を入力して操作を続けます。

n を入力すると、スクリプトは正しい情報の入力を要求します。n を入力した場合の操作については、[98 ページの「誤った情報を変更する方法」](#)を参照してください。

```
Is this information correct?
(type 'y' to accept, 'n' to change) y
This script sets up machine "client2" as an NIS+ non-root master
server for domain sales.doc.com.
Do you want to continue? (type 'y' to continue, 'n' to exit this script)
```

3. **y** を入力して操作を続けます。

n を入力すると、スクリプトは安全に終了します。クライアントマシン上で rpc.nisd が動作していない場合、スクリプトは自動的に終了します。

```
Do you want to continue? (type 'y' to continue, 'n'
to exit this script)
y
running nissetup ...
org_dir.sales.doc.com. created
groups_dir.sales.doc.com. created
```



```
...
...
setting NIS+ group admin.sales.doc.com. ...
The system client2 is now configured as a non-root server for
domain sales.doc.com.
You can now populate the standard NIS+ tables by using the
nispopulate or /usr/lib/nis/nisaddent commands.
```

これで、マシン `client2` は `sales.doc.com.` ドメインのマスターサーバーになりました。`sales.doc.com.` ドメインは、`doc.com.` ドメインのサブドメインです。マシン `client2` は、依然としてルートドメイン `doc.com.` のクライアントであり、しかも `sales.doc.com.` ドメインのマスターサーバーでもあります。これで、`sales.doc.com.` ドメインの新しいマスターサーバーに標準の NIS+ テーブルを生成できます。

ドメインの追加作成

サーバーを新しいルート以外のドメインのマスターサーバーに変更する前述の手順を、必要な各サーバーマシンに対して繰り返します。新しいマスターサーバーはすべて新しいドメインです。ドメイン構造の計画を立ててから、NIS+ 名前空間の作成を開始します。NIS+ 階層の計画については、[57 ページの「NIS+ 名前空間の構造」](#)を参照してください。

新しいサブドメインのテーブルの生成

新しいドメインを作成したら、そのマスターサーバーの標準の NIS+ テーブルを生成しなければなりません。新しいマスターサーバーのテーブルを生成するには、ルートマスターサーバーのテーブルを生成するときと同じ手順を使用します。大きな違いは、`nispopulate(1M)` スクリプトが、ルートマスターサーバー上ではなく、新しいマスターサーバー上で実行されることです。また、ドメイン名、およびファイルパス、または NIS サーバー名も変わることがあります。

この例では、新しいドメイン (`sales.doc.com.`) のテーブルを生成します。

`nispopulate` を実行してサブドメインのテーブルを生成するための前提条件

`nispopulate` スクリプトを実行して新しいマスターサーバーのテーブルを生成するには、次の条件を満たしている必要があります。

- ファイル内の情報は、ロード先のテーブルに合った書式で書かれていなければなりません。

- スクリプトを実行する前に、データを読み込むローカルの /etc 内の各ファイルまたは NIS マップを調べます。不正なエントリがないことを確認します。正しいデータが、所定の場所に正しい書式で記録されていることを確認します。エントリのうち、古いもの、無効なもの、破損しているものは削除します。また、不完全なエントリや一部のみのエントリも削除します。構成完了後は、いつでもエントリを追加できます。必要なエントリをあとから追加する方が、不完全なエントリや破損しているエントリを読み込もうとするよりも簡単です。
- ネットワークをはじめて設定する場合、多くのネットワーク情報をすでにどこかに格納しているということはおそらくないでしょう。このような場合、まずネットワーク情報を集め、「入力ファイル」に手入力する必要があります。このファイルは、基本的に /etc 内のファイルに対応するものです。
- 安全のため、/etc 内のファイルのコピーを作成します。実際のファイルは使用せずに、作成したコピーを使用してテーブルを生成してください。たとえば、この例では /nis+files というディレクトリ内のファイルを使用します。
- コピーした NIS テーブルファイルのうち、passwd、shadow、aliases、hosts の各ファイルには、名前空間全体に分散させるとセキュリティ上問題がある項目があるので、それらを編集します。たとえば、次に示す行をローカル側の passwd ファイルのコピーから削除して、名前空間で配布されないようにします。

```

root:x:0:1:0000-Admin(0000):/:/sbin/sh
daemon:x:1:3:0000-Admin(0000):/:/:
bin:x:3:5:0000-Admin(0000):/usr/bin:/:
sys:x:3:3:0000-Admin(0000):/:/:
adm:x:4:4:0000-Admin(0000):/var/adm:/:
lp:x:78:9:0000-lp(0000):/usr/spool/lp:/:
smtp:x:0:0:mail daemon user:/:/:
uucp:x:5:5:0000-uucp(0000):/usr/lib/uucp:/:
nuucp:x:7:8:0000-
uucp (0000):/var/spool/uucppublic:/usr/lib/uucp/uucico
listen:x:22:6:Network Admin:/usr/net/nls:/:
nobody:x:60000:60000:uid no body:/:/:
noaccess:x:60002:60002:uid no access:/:/:

```
- ドメインがすでに構成され、そのマスターサーバーが実行されている
- ドメインのサーバーには、新しいテーブル情報を収容できるだけの十分なディスク空き領域が必要です。
- NIS+ 主体としてログインしていて、しかも指定されたドメイン内の NIS+ テーブルに対する書き込み権が必要です。この例では、マシン client2 上の root でなければなりません。

注 - システム上の /tmp 領域が不足する場合、nispopulate スクリプトは異常終了することがあります。これを防止するには、環境変数 TMPDIR に別のディレクトリを設定します。TMPDIR に有効なディレクトリが設定されていない場合、スクリプトは代わりに /tmp ディレクトリを使用します。

必要な情報

ファイルから生成するか、NIS マップから生成するかによって収集する情報が異なります。

ファイルから生成する場合、次の情報が必要です。

- 新しい NIS+ ドメイン名
- データが転送される、適切に編集されたテキストファイルのパス
- NIS+ マスターサーバーの root パスワード

NIS マップから生成する場合、次の情報が必要です。

- 新しい NIS+ ドメイン名
- NIS ドメイン名
- NIS サーバー名
- NIS サーバーの IP アドレス
- NIS+ マスターサーバーの root パスワード

注 - NIS ドメインの名前は大文字と小文字を区別しますが、NIS+ ドメインの名前は区別しません。

マスターサーバーテーブルを生成する方法

この手順は、[102 ページの「ルートマスターサーバーのテーブルを生成する方法」](#)の手順と実質的に同じなので、この例では新しい sales.doc.com. ドメインのテーブルを生成するための入力についてのみ説明します。この手順の詳細は、[102 ページの「ルートマスターサーバーのテーブルを生成する方法」](#)を参照してください。

注 - このスクリプトは、ルートマスターサーバーではなく、新しいドメインのマスターサーバー上で実行しなければなりません。

次のいずれかの方法で、新しいマスターサーバー上にマスターサーバーテーブルを生成します。

- ファイルからマスターサーバーテーブルを生成する
- NIS マップからマスターサーバーテーブルを生成する

実行結果が一度に表示しきれないことがあるので、どちらの方法で行うにしてもスクロール可能なウィンドウを使用してください。

テーブルをファイルから生成する方法

ファイルからマスターサーバーテーブルを生成するには、次のコマンドを入力します。

```
client2# nispopulate -F -p /nis+files -d sales.doc.com.  
NIS+ domain name : sales.doc.com.  
Directory Path : /nis+files  
Is this information correct? (type 'y' to accept, 'n' to change
```

テーブルを **NIS** マップから生成する方法

NIS マップからマスターサーバーテーブルを生成するには、次のコマンドを入力します。

```
client2# nispopulate -Y -d sales.doc.com. -h businessmachine -a  
IP_addr_of_NIS_server -y business.doc.com.  
NIS+ Domain name : sales.doc.com.  
NIS (YP) domain : business.doc.com.  
NIS (YP) server hostname : businessmachine  
Is this information correct? (type 'y' to accept, 'n' to change)
```

詳細は、[102 ページの「ルートマスターサーバーのテーブルを生成する方法」](#)を参照してください。

サブドメイン複製サーバーの作成

ルートドメイン複製サーバーに対して適用される規則は、サブドメイン複製サーバーに対しても適用されます ([113 ページの「ルート複製サーバーの作成」](#)を参照)。

サブドメインの複製サーバーの作成には、ルート複製サーバーの作成と同じ手順を使用します。このルート複製サーバーの作成とサブドメイン複製サーバーの作成との大きな違いは、サブドメイン複製サーバーに変更しようとしているマシンが、複製サーバーとしてサービスを提供することになるドメインより上位のドメインのクライアントのままであることです。この例では、新しいドメインの複製サーバーを作成するための入力だけを示します。スクリプトの残りの出力については、[115 ページの「ルート複製サーバーを作成する方法」](#)を参照してください。

nisserver を実行してサブドメイン複製サーバーを作成するための前提条件

nisserver を実行して複製サーバーを作成するには、次の条件を満たしている必要があります。

- ドメインがすでに構成され、そのマスターサーバーが実行されている
- ドメインのテーブルが生成されている。少なくとも、`hosts` テーブルには新しいクライアントマシンのエントリが必要
- 親ドメイン内のクライアントマシンが初期設定されている

- クライアント上で NIS+ サービス (rpc.nisd) が起動されている
- マスターサーバーにスーパーユーザーとしてログインしている (この例では、マスターマシン名は client2)

必要な情報

nisserver を実行するには、次の情報が必要です。

- ドメイン名
- クライアントマシン名 (この例では、client3)
- ルートマスターサーバーのスーパーユーザーのパスワード

▼ 複製サーバーを作成する方法

- 複製サーバーを作成するには、NIS+ ドメインのマスターサーバー上でスーパーユーザー (**root**) として、**nisserver -R** コマンドを実行します。

```
client2# nisserver -R -d sales.doc.com. -h client3
This script sets up a NIS+ replica server for domain sales.doc.com.
Domain name      ::sales.doc.com.
NIS+ server :client
Is this information correct? (type 'y' to accept, 'n' to change)
```

この例では、client2 がマスターサーバーです。-R オプションは、複製サーバーを構成することを示します。-d オプションは NIS+ ドメイン名 (ここでは、sales.doc.com.) を指定します。-h オプションは複製サーバーとなるクライアントマシン (ここでは client3) を指定します。なお、このマシンは依然として doc.com. ドメインのクライアントであり、sales.doc.com. ドメインのクライアントではありません。

このスクリプトの残りの出力については、115 ページの「ルート複製サーバーを作成する方法」を参照してください。

サブドメインの NIS+ クライアントマシンの初期設定

マスターサーバーのテーブルをファイルまたは NIS マップから生成すれば、NIS+ クライアントマシンを初期設定できます。この節では、デフォルト設定の nisclient スクリプトを使用して、新しいドメイン内の NIS+ クライアントを初期設定する方法を説明します。NIS+ クライアントマシンは、NIS+ マスターサーバーとは別のマシンです。

注 - 126 ページの「新しいサブドメインクライアントマシンを初期設定する方法」で使用される `-i` オプションは、NIS+ クライアントを DNS を必要とするホスト名の解決ができるようには構成しません。DNS を使用する場合には、ネームサービススイッチファイルの中でクライアント用に、DNS を明確に指定しなければなりません。

新しいドメイン内のクライアントを初期設定するには、ルートドメイン内のクライアントの初期設定と同じ手順を使用します。この例では、新しいドメインのクライアントを初期設定するための入力だけを示します。スクリプトの残りの出力については、107 ページの「新しいクライアントマシンを初期設定する方法」を参照してください。

nisclient を実行してクライアントマシンを初期設定するための前提条件

nisclient スクリプトを使用してクライアントマシンを初期設定するには、次の条件を満たしている必要があります。

- ドメインがすでに構成され、そのマスターサーバーが実行されている
- ドメインのマスターサーバーのテーブルが生成されている。少なくとも、hosts テーブルには新しいクライアントマシンのエントリが必要
- ドメイン内のクライアントマシンが初期設定されている
- クライアントマシンに「ユーザー」としてログインしている (この例では、ユーザー名は user1)

必要な情報

nisclient を実行するには、次の情報が必要です。

- ドメイン名 (この例では、sales.doc.com.)
- デフォルトのネットワークパスワード (この例では、nisplus)
- クライアントとなるマシンの root のパスワード
- クライアントのホームドメイン内の NIS+ サーバーの IP アドレス (この例では、マスターサーバー client2 のアドレス)

▼ 新しいサブドメインクライアントマシンを初期設定する方法

- 新しいクライアントマシン上で新しいクライアントを初期設定するには、スーパーユーザーとして次のコマンドを入力します。

```
subclient1# nisclient -i -d sales.doc.com. -h client2
Initializing client subclient1 for domain "sales.doc.com.".
Once initialization is done, you will need to reboot your machine.
Do you want to continue? (type 'Y' to continue, 'N' to exit this script)
```

-i オプションはクライアントを初期設定します。-d オプションは新しい NIS+ ドメイン名を指定します。ドメイン名が指定されない場合、デフォルトは現在のドメイン名になります。-h オプションは NIS+ サーバーのホスト名を指定します。

このスクリプトの残りの出力については、[107 ページの「新しいクライアントマシンを初期設定する方法」](#)を参照してください。

サブドメインの NIS+ クライアントユーザーの初期設定

新しいドメイン内のユーザーを初期設定するには、ルートドメイン内のユーザーを初期設定する場合と同じ手順 (nisclient) を使用します。全ユーザーが NIS+ クライアントにならなければなりません。この例では、新しいドメインのユーザーを初期設定するための入力だけを示します。スクリプトの残りの出力については、[110 ページの「NIS+ ユーザーを初期設定する方法」](#)を参照してください。

nisclient を実行してサブドメインのクライアントユーザーを初期設定するための前提条件

nisclient スクリプトを使用してユーザーを初期設定するには、次の条件を満たしている必要があります。

- ドメインがすでに構成され、そのマスターサーバーが実行されている
- ドメインのマスターサーバーのテーブルが生成されている。少なくとも、hosts テーブルには新しいクライアントマシンのエントリが必要
- ドメイン内のクライアントマシンが初期設定されている
- クライアントマシンに「ユーザー」としてログインしている (この例では、ユーザー名は user2)

必要な情報

nisclient を実行するには、次の情報が必要です。

- ユーザーのログイン名 (この例では、user2)
- デフォルトのネットワークパスワード (この例では、nisplus)

- NIS+ クライアントになるユーザーのログインパスワード

▼ NIS+ サブドメインユーザーを初期設定する方法

- NIS+ クライアントとなるには、ユーザーとしてログインして、次のコマンドを実行します。

```
user2prompt% nisclient -u
At the prompt below, type the network password (also known as the
Secure-RPC password) that you obtained either from your administrator
or from running the nispopulate script.
Please enter the Secure-RPC password for user2:
```

このスクリプトの残りの出力については、110 ページの「NIS+ ユーザーを初期設定する方法」を参照してください。

NIS+ 名前空間サンプルで使ったコマンドのまとめ

名前空間サンプルの作成で使ったコマンドを表 4-4 にまとめます。各コマンドの前にあるプロンプトは、そのコマンドをどのマシンで入力するかを示します。

表 4-4 NIS+ 名前空間サンプル – コマンドのまとめ

作業	コマンド
/usr/lib/nis を含むように環境パスを設定 (C シェルまたは Bourne シェル)	# setenv PATH \$PATH:/usr/lib/nis または # PATH=\$PATH:/usr/lib/nis; export PATH
オプションで Diffie - Hellman 鍵の長さを設定する	master1# nisauthconf dh640-0 des
doc.com. ドメインのルートマスターサーバーを作成	master1# nisserver -r -d doc.com.
ルートマスターサーバーの NIS+ テーブルを、ファイルまたは NIS マップから生成	master1# nispopulate -F -p /nis+files -d doc.com. または master1# nispopulate -Y -d doc.com. -h salesmaster -a \ 172.31.58.111 -y sales.doc.com.

表 4-4 NIS+ 名前空間サンプル – コマンドのまとめ (続き)

作業	コマンド
管理グループ (2) にメンバーを追加する	master1# nisgrpadm -a admin.doc.com. topadmin.doc.com. \secondadmin.doc.com.
NIS+ データベースのチェックポイントを実行する	master1# nisping -C org_dir.doc.com.
オプションで Diffie - Hellman 鍵の長さを設定する	client1# nisauthconf dh640-0 des
doc.com. ドメイン内の NIS+ クライアントマシンを初期設定する	client1# nisclient -i -d doc.com. -h master1
ユーザーを NIS+ クライアントとして初期設定する	client1user1prompt% nisclient -u
NIS+ クライアントを、NIS 互換性あり、またはなし、あるいは NIS と DNS ありで NIS+ サーバーに変更	client1# rpc.nisd または client1# rpc.nisd -Y または client1# rpc.nisd -Y -B
ルート複製サーバーを作成	master1# nisserver -R -d doc.com. -h client1
サーバーを sales.doc.com. ドメインのルート以外のマスターサーバーに変更	master1# nisserver -M -d sales.doc.com. -h client2
新しいマスターサーバーの NIS+ テーブルを、ファイルまたは NIS マップから生成	client2# nispopulate -F -p /nis+files -d sales.doc.com. または client2# nispopulate -Y -d sales.doc.com. -h \businessmachine -a 172.31.58.242 -y business.doc.com.
マスターサーバーの複製を作成	client2# nisserver -R -d sales.doc.com. -h client3
sales.doc.com. ドメイン内の NIS+ クライアントを初期設定する	subclient1# nisclient -i -d sales.doc.com. -h client2
ユーザーを NIS+ クライアントとして初期設定する	subclient1user2prompt% nisclient -u

第 5 章

ルートドメインの設定

この章では、NIS+ コマンド群によるルートドメインと DES 認証の設定の手順を説明します。

注 - NIS+ は、将来のリリースではサポートされなくなる可能性があります。NIS+ から LDAP への移行支援ツールは、Solaris 9 リリース以降で使用できます (『Solaris のシステム管理(ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』を参照)。詳細については、
<http://www.sun.com/directory/nisplus/transition.html> を参照してください。

ルートドメインの設定方法の概要

ここでは、ルートマスターサーバーをセキュリティレベル 2 で稼働させるルートドメインの構成方法について説明します。

注 - ルートドメインの構成は、この章で説明する NIS+ コマンドより、NIS+ スクリプトを使用する方が簡単です。この章で説明する方法は、NIS+ に精通した管理者や、設定スクリプトでは提供されない標準以外の機能や構成を必要とする管理者だけが使用してください。

ルートドメインの設定には、次の 3 つの主な作業があります。

- ルートマスターサーバーの準備
- ルートドメインの作成
- ルートドメインの資格の作成

ただし、ルートドメインの設定はこれらの3つの作業を単にこの順序で行うといった簡単なものではありません。3つの作業はそれぞれ関連し合っています。たとえば、ルートディレクトリを作成する前にセキュリティパラメータの一部を指定しなければなりません。ほかのパラメータはその後で指定します。ルートドメインの構成を容易にするために、この節ではこれらの作業を個別の手順に分けて、一番効率の良い順番に並べています。

標準構成と NIS 互換構成の手順の相違

この章で説明する手順は、標準の NIS+ ルートドメインと NIS 互換のルートドメインの両方に適用できます。ただし、いくつかの重要な相違があります。NIS 互換ドメイン用の NIS+ デモンは `-y` オプションを使用して起動する必要があります。これによりルートマスターサーバーは、NIS クライアントからの要求に応えることができます。詳細については、[89 ページの「NIS+ とサービス管理機能」](#)を参照してください。

また、NIS 互換ドメインでは、`passwd` テーブルは、未認証クラスに対する読み取り権が必要です。これにより NIS クライアントはテーブルのパスワード列にある情報にアクセスすることができます。これは[手順 14](#)で `nissetup` コマンドに `-y` オプションを使用して行います。標準の NIS+ ドメインは同じコマンドを使用しますが、`-y` オプションは付けません。

注 - NIS+ サービスは、サービス管理機能 (SMF) によって管理されます。このサービスに対する有効化、無効化、再起動などの管理操作を実行するには、`svcadm` コマンドを使用します。SMF を NIS+ で使用する方法については、[89 ページの「NIS+ とサービス管理機能」](#)を参照してください。SMF の概要については、『Solaris のシステム管理 (基本編)』の「サービスの管理 (概要)」を参照してください。詳細については、`svcadm(1M)` と `svcs(1)` の各マニュアルページも参照してください。

ルートドメインを確立する

各手順では詳細を説明し、また関連する情報についても説明します。詳細な説明が必要でない場合は、[145 ページの「ルートドメイン構成の要覧」](#)に必要なコマンドをまとめたリストがありますので参照してください。

手順の要約

設定作業の手順を次にまとめます。

1. ルートマスターサーバーにスーパーユーザーとしてログインします。
2. ルートマスターサーバーのドメイン名をチェックします。
3. ルートマスターサーバーのスイッチ構成ファイルをチェックします。
4. オプションで Diffie-Hellman キー長を設定します。
5. `nsswitch.conf` ファイルに少しでも変更を加えた場合は、ネームサービス キャッシュ (`nscd`) を再起動します。
6. `/etc/.rootkey` を削除し、`keyserv` を再起動します。
7. NIS+ サービスを停止します。
8. NIS+ のファイルを削除しプロセスを終了します。
9. ルートドメインの管理グループを指定します。
10. ルートディレクトリを作成し、ルートマスターサーバーを初期設定します。
11. (省略可能) `/lib/svc/method/nisplus` ファイルを編集して必要なオプションを追加します。
12. NIS+ デーモンを起動します。
13. ルートオブジェクトが作成されているかどうか確認します。
14. ルートドメインのサブディレクトリとテーブルを作成します。
15. ルートマスターサーバーの DES 資格を作成します。
16. ルートドメインの管理グループを作成します。
17. ルートドメインの管理 (`admin`) グループにルートマスターを追加します。
18. ルートドメインの公開鍵を更新します。
19. NIS+ サービスを起動します (これにより、キャッシュマネージャと NIS+ デーモンがセキュリティレベル 2 で起動します)。
20. 自分の LOCAL 資格をルートドメインに追加します。
21. 自分の DES 資格をルートドメインに追加します。
22. ほかの管理者の資格を追加します。
23. ルートドメインの管理グループに自分とほかのシステム管理者を追加します。
24. 十分なスワップ空間を割り当てます。

ルートドメインを確立する — 作業マップ

表 5-1 ルートドメインを確立する

作業	目的	参照先
ルートドメインを確立する	domainname コマンドを使用して、ルートドメインを設定する。オプションで Diffie-Hellman キー長を長くする。ncsd デーモンの停止と起動を行う。keyserv を無効にしてから、再起動する。不要な NIS+ 情報を消去する	135 ページの「ルートドメインを構成する方法」

セキュリティについて

NIS+ はルートドメインに対しデフォルトのセキュリティを提供します。デフォルトのセキュリティレベルは、レベル 2 です。実際にユーザーが使用する本稼働のネットワークは、常にセキュリティレベル 2 で運用してください。セキュリティレベル 0 およびレベル 1 は、構成およびテスト目的だけに使用します。本番環境のネットワークは、レベル 0 または 1 で稼働しないでください。

注 - NIS+ のセキュリティシステムは複雑です。NIS+ セキュリティを使い慣れていない場合は、第 11 章を参照してから NIS+ 環境を構成することをお勧めします。

ルートドメインを確立するための前提条件

作業を開始する前に、以下の条件を満たしていることを確認します。

- ルートマスターサーバー上の /etc/passwd ファイルには、この構成作業でルートドメインに資格を追加するシステム管理者である自分と、ほかのすべてのシステム管理者のエントリが含まれなければなりません。
- サーバーが NIS 互換モードで動作し、Solaris 1.x のクライアントで DNS 転送を使用する場合は、/etc/resolv.conf ファイルを正しく構成しておく必要があります。
- サーバーには、必ずほかのユーザー ID と異なる、固有のマシン名を付けてください。
- サーバーに付けるマシン名には、ドットを使用しないでください。たとえば、マシン名に sales.alpha を使用できません。sales-alpha というマシン名は、使用できます。

必要な情報

ルートドメインを作成するには、次の内容について調べておく必要があります。

- マシン (ルートマスターサーバーにする) のスーパーユーザーのパスワード

- ルートドメイン名
- ルートドメインの管理グループ名
- 自分のユーザー ID とパスワード
- ルートドメインに資格を追加する予定のほかのシステム管理者のユーザー ID

▼ ルートドメインを構成する方法

1. ルートマスターサーバーとするマシン上にスーパーユーザーとしてログインします。
次の例では、rootmaster をルートマスターサーバーとして、doc.com. をルートドメインとして、それぞれ使用します。
2. ルートマスターサーバーのドメイン名をチェックします。
domainname コマンドを使用して、ルートマスターサーバーが正しいドメイン名を使用していることを確認します。domainname コマンドは、マシンの現在のドメイン名を返します。



注意 – またドメイン名とホスト名が同じにならないようにします。たとえば、sales ドメインがある場合は、ホスト名に sales を使用することはできません。同様に、home という名前のマシンを使用する場合は、home という名前のドメインを作成しないでください。これは、サブドメインについても同様です。たとえば、すでにホスト名として west がある場合には、sales.west.doc.com というサブディレクトリを作成することはできません。

ドメイン名が正しくない場合は、変更してください。

```
rootmaster# domainname
strange.domain
rootmaster# domainname doc.com
rootmaster# domainname
rootmaster# doc.com
rootmaster# rm -f /etc/defaultdomain
rootmaster# domainname > /etc/defaultdomain
```

domainname コマンドの最後にドットを付けしないでください。domainname コマンドは、NIS+ コマンドではないため、ドメイン名の最後にドットを付ける NIS+ の表記規則には従いません。

上記の例では、ルートマスターサーバーのドメイン名を strange.domain から doc.com に変更しています。ドメイン名を変更したり設定したりする場合は、doc ではなく doc.com のように、少なくとも 2 つのラベルを付けてください。末尾の要素はインターネットの組織コード (.com など) または地域識別子 (.jp、.uk など) にしてください。

3. ルートマスターサーバーのスイッチ構成ファイルをチェックします。
ルートマスターサーバーが、NIS 互換モードで動作する場合であっても、NIS+ 用の nsswitch.conf ファイルを使用していることを確認してください。この手順

で、ルートマスター用の情報の主情報源が確実に NIS+ テーブルとなります。

```
rootmaster# more /etc/nsswitch.conf
```

上のコマンドを実行すると、現在の `nsswitch.conf` ファイルの内容が表示されます。このファイルによって参照される主ネームサービスは `nisplus` です。ルートマスターサーバーの構成ファイルが主ネームサービスとして `nisplus` を使用していない場合は、38 ページの「構成ファイルの変更」を参照して、`nisplus` を使用する構成ファイルに置き換えてください。

4. オプションで **Diffie-Hellman** キー長を設定します。

DES 認証を使用する場合には、Diffie-Hellman キー長をデフォルトの 192 ビットより長くすることができます。たとえば、640 ビットと 192 ビットの両方を許可する場合には、以下を入力します。

```
rootmaster# nisauthconf dh640-0 des
```

5. **nsswitch.conf** ファイルに少しでも変更を加えた場合は、ネームサービス キャッシュ (**nscd**) デーモンを再起動します。

```
rootmaster# svcs \*name\*
online    Jan_12    svc:/system/name-service-cache:default
rootmaster# svcadm restart system/name-service-cache
```

`nscd` が `nsswitch.conf` ファイルの内容をキャッシュに格納しているため、ファイルの内容を変更したあとは、`nscd` を再起動する必要があります。

詳細な手順については、第 1 章を参照してください。

6. **/etc/.rootkey** ファイルを削除し、**keyserv** を再起動します。

```
rootmaster# cp /etc/nsswitch.nisplus /etc/nsswitch.conf
rootmaster# svcs \*keyserv\*
online    Jan_12    svc:/network/rpc/keyserv:default
rootmaster# svcadm disable network/rpc/keyserv
rootmaster# rm -f /etc/.rootkey
rootmaster# svcadm enable network/rpc/keyserv
```

7. **NIS+** サービスを停止します。

```
rootmaster# svcs \*nisplus\*
online    Jan_12    svc:/network/rpc/nisplus:default
rootmaster# svcadm disable network/rpc/nisplus:default
rootmaster# svcs \*nisplus\*
disabled  Jan_12    svc:/network/rpc/nisplus:default
```

8. **NIS+** のファイルを削除しプロセスを終了します。

現在作業しているマシンが、以前は NIS+ のサーバーまたはクライアントとして使用したものである場合、`/var/nis` 内にファイルがあればすべて削除します。この例では、`/var/nis` 内にはコールドスタートファイルとディレクトリ キャッシュファイルがまだ存在します。

```
rootmaster# ls /var/nis
NIS_COLD_START NIS_SHARED_CACHE
rootmaster# rm -rf /var/nis/*
```


この手順によって、/var/nis 内に残されたファイル、またはキャッシュマネージャによって格納されたディレクトリオブジェクトは完全に消去されるため、これらがこの設定作業で生成される新しい情報と重複することはありません。
/var/nis 内に管理スクリプトを格納していた場合、ルートドメインの設定が終わるまでは、これらを一時的にほかのどこかに格納しておくことができます。

9. ルートドメインの管理グループを指定します。

手順 16 までは、実際には管理グループを作成しませんが、ここで管理グループの名前を指定する必要があります。手順 13 で管理グループの名前を指定することによって、ルートドメインの org_dir ディレクトリオブジェクト、groups_dir ディレクトリオブジェクト、およびその全テーブルオブジェクトが手順 14 で作成されたときに、適切なデフォルトグループが割り当てられます。

管理グループを指定するには、環境変数 NIS_GROUP の値に、ルートドメインの管理グループの名前を設定します。次に、csh ユーザー用と、sh/ksh ユーザー用の 2 つの例を示します。どちらも、NIS_GROUP を admin.doc.com. に設定します。

C シェルの場合

```
rootmaster# setenv NIS_GROUP admin.doc.com.
```

Bourne シェルまたは Korn シェルの場合

```
rootmaster# NIS_GROUP=admin.doc.com.
```

```
rootmaster# export NIS_GROUP
```

10. ルートディレクトリを作成し、ルートマスターサーバーを初期設定します。

この手順では、名前空間内の最初のオブジェクトであるルートディレクトリを作成し、マシンをルートマスターサーバーに変換します。次に示すように、nisinit -r コマンドを使用します。この場合に限り、ドメインのディレクトリオブジェクトの作成と、そのマスターサーバーの初期設定が 1 回の操作で行われます。実際には nisinit -r が nismkdir を実行し、自動的にルートディレクトリを作成します。いずれにしても、ルートマスターの場合を除くと、これら 2 つのプロセスは別々の作業として実行されます。

```
rootmaster# nisinit -r
This machine is in the doc.com. NIS+ domain
Setting up root server ...
All done.
```

/var/nis/data という名前の UNIX ディレクトリが生成されます。

/var/nis ディレクトリ内には、root.object というファイルがあります。

```
rootmaster# ls -l /var/nis/data
-rw-rw-rw- 1 root other 384 date root.object
```

これはルートディレクトリオブジェクトではありません。実際には、NIS+ が内部目的で名前空間のルートを記述するために使用するファイルです。NIS+ のルートディレクトリオブジェクトはあとで作成します。

以降の手順では、この手順で作成されるディレクトリの下に、ほかのファイルが追加されます。UNIX ディレクトリを直接調べることによって、これらのファイルの存在を検証できますが、NIS+ にはもっと便利なコマンドがあります。以下の手順では、必要に応じてこれらのコマンドを実行します。



注意 -nisinit など NIS+ の構成手順によって作成された /var/nis、/var/nis/data といったディレクトリ、およびその下のファイルは、名前を変更しないでください。Solaris 2.4 以前では、/var/nis ディレクトリに *hostname.dict*、*hostname.log* という 2 つのファイルが存在していました。またサブディレクトリ /var/nis/*hostname* もありました。Solaris 2.5 では、これらの 2 つのファイル名は *trans.log*、*data.dict* とし、サブディレクトリ名は /var/nis/*data* となります。Solaris リリース 2.5 ではこれらのファイルの内容も変更されており、Solaris リリース 2.4 以前との互換性はなくなっています。したがって、これらのファイルやディレクトリを Solaris 2.4 での名前にしてしまうと、Solaris 2.4、2.5 双方の *rpc.nisd* で機能しなくなるので名前の変更をしないようにしてください。ディレクトリ名もファイル名も変更しないでください。

11. (省略可能) /lib/svc/method/nisplus ファイルを編集して必要なオプションを追加します。

-s 0 オプションを追加する必要があります。適切なテキストエディタを使用します。

/lib/svc/method/nisplus ファイルの編集方法については、[89 ページ](#)の「NIS+ とサービス管理機能」を参照してください。

-s 0

サーバーのセキュリティレベルを 0 に設定します。現時点ではブートストラップに必要です。

cred テーブルがまだ存在しないので、NIS+ 主体は資格を持つことができません。このため、高いセキュリティレベルを使用すると、サーバーからロックアウトされます。

-B

DNS 転送をサポートします。

-Y

NIS 互換モードで NIS+ デーモンを起動します。

12. NIS+ サービスデーモンを起動します。

```
rootmaster# svcadm enable network/rpc/nisplus:default
```

13. ルートオブジェクトが正しく作成されているかどうか確認します。

名前空間には次のものが作成される必要があります。

- ルートディレクトリオブジェクト (*root.dir*)
- NIS+ サービスを実行するルートマスターサーバー (*rootmaster*)
- マスターサーバー用のコールドスタートファイル (*NIS_COLD_START*)
- トランザクションログファイル (*trans.log*)
- テーブル辞書ファイル (*data.dict*)

ルートディレクトリオブジェクトは、[手順 10](#) で作成したディレクトリに格納されます。ディレクトリの内容を確認してください。

```
rootmaster# ls -l /var/nis/data
-rw-rw-rw- 1 root other 384 date root.object
-rw-rw-rw- 1 root other 124 date root.dir
```

この時点で、ルートディレクトリは空です。つまり、サブディレクトリはありません。このことを確認するには、`nislsl` コマンドを使用します。

```
rootmaster# nislsl -l doc.com.
doc.com.:
```

ただし、いくつかの「オブジェクト」属性は存在し、`niscat -o` を使用すればこれを調べることができます。

```
rootmaster# niscat -o doc.com.
Object Name : doc
Owner : rootmaster.doc.com.
Group : admin.doc.com.
Domain : Com.
Access Rights : r---rmcdrmcdr---
```

なお、ルートディレクトリオブジェクトは、「所有者」と「グループ」の両方には完全な (読み取り、変更、作成、削除) 権利を与え、「その他」と「未認証」には読み取り権だけを与えます。ディレクトリオブジェクトがこれらの権利を提供しない場合、`nischmod` コマンドを使用して変更できます。

NIS+ デモンが動作していることを確認するには、`ps` コマンドを使用します。

```
rootmaster# ps -e | grep rpc.nisd
root  1081  1  61  16:43.33  ?  0:01  rpc.nisd  -S 0
root  1087 1004 11  16:44:09 pts/1  0:00  grep  rpc.nisd
```

ルートマスターサーバーのインターネットアドレスが (最終的には公開鍵も) 収められているルートドメインの `NIS_COLD_START` ファイルは、`/var/nis` 内に置かれます。この内容を調べるための NIS+ コマンドはありませんが、この内容はサーバーのディレクトリキャッシュ (`NIS_SHARED_DIRCACHE`) に保存されます。これらの内容は、`/usr/lib/nis/nisshowcache` コマンドで調べることができます。

また、トランザクションログファイル (`trans.log`) と辞書ファイル (`data.dict`) も作成されます。マスターサーバーのトランザクションログは、前回の更新以降マスターサーバーとそのすべての複製サーバーによって実行されたすべてのトランザクションを格納しています。この内容を調べるには、`nislog` コマンドを使用します。辞書ファイルは、NIS+ が内部目的に使用するものであり、システム管理者には関係ありません。

14. ルートドメインのサブディレクトリとテーブルを作成します。

この手順では、ルートディレクトリオブジェクトの下に、`org_dir` ディレクトリと `groups_dir` ディレクトリ、および NIS+ テーブルを追加します。`nissetup` ユーティリティを使用してください。NIS 互換ドメインの場合、必ず `-y` フラグを付けてください。両方の場合の例を次に示します。

標準 NIS+ のみの場合

```
rootmaster# /usr/lib/nis/nissetup
```

NIS 互換のみの場合

```
rootmaster# /usr/lib/nis/nissetup -Y
```

このユーティリティによって追加されたオブジェクトを表示すると、次のようになります。

```
rootmaster# /usr/lib/nis/nissetup
org_dir.doc.com. created
groups_dir.doc.com. created
auto_master.org_dir.doc.com. created
auto_home.org_dir.doc.com. created
bootparams.org_dir.doc.com. created
cred.org_dir.doc.com. created
ethers.org_dir.doc.com. created
group.org_dir.doc.com. created
hosts.org_dir.doc.com. created
mail_aliases.org_dir.doc.com. created
sendmailvars.org_dir.doc.com. created
netmasks.org_dir.doc.com. created
netgroup.org_dir.doc.com. created
networks.org_dir.doc.com. created
passwd.org_dir.doc.com. created
protocols.org_dir.doc.com. created
rpc.org_dir.doc.com. created
services.org_dir.doc.com. created
timezone.org_dir.doc.com. created
```

-Y オプションは、標準の NIS+ ドメインと同じテーブルとサブディレクトリを作成します。しかし、NIS クライアントからの未認証要求が passwd テーブルに含まれる暗号化されたパスワードにアクセスできるようにするため、passwd テーブルへの読み取り権を未認証クラスに割り当てます。

現在、ルートディレクトリには 2 つのサブディレクトリがあります。

```
rootmaster# nisl doc.com.
doc.com.:
org_dir
groups_dir
```

サブディレクトリとテーブルのオブジェクト属性を調べるには、niscat -o コマンドを使用してください。また、フラグなしで niscat オプションを使用して、このテーブル内の情報を調べることもできますが、この時点では空です。

15. ルートマスターサーバーの DES 資格を作成します。

ルートマスターサーバーは、自分の要求が認証されるために、DES 資格を必要とします。これらの資格を作成するには、次に示すように nisaddcred コマンドを使用します。プロンプトに対して、サーバーの root パスワードを入力します。

```
rootmaster# nisaddcred des
DES principal name: unix.rootmaster@doc.com
Adding key pair for unix.rootmaster@doc.com
(rootmaster.doc.com.).
Enter login password:
Wrote secret key into /etc/.rootkey
```

サーバーの root パスワードと異なるパスワードを入力した場合、警告メッセージとパスワードを再入力するプロンプトが表示されます。

```
Enter login password:
nisaddcred: WARNING: password differs from login password.
Retype password:
```

それでも同じパスワードを再び入力すると、NIS+ は資格を作成します。この新しいパスワードは /etc/.rootkey に格納され、キーサーバーが起動時に使用します。キーサーバーに新しいパスワードをすぐに登録するには、keylogin -r を実行します。第 12 章を参照してください。

最終的に自分のログインパスワードを使用する場合は、Control-C を押して、この手順をやり直します。Control-C を押さずにメッセージに従って自分のログインパスワードを入力すると、別の目的のエラーメッセージが表示され、混乱を招くことがあります。

```
nisaddcred: WARNING: password differs from login password.
Retype password:
nisaddcred: password incorrect.
nisaddcred: unable to create credential.
```

この手順の結果として、ルートサーバーの非公開鍵と公開鍵がルートドメインの cred テーブル (cred.org_dir.doc.com.) に格納され、その非公開鍵は /etc/.rootkey. に格納されます。cred テーブル内の資格を確認するには、niscat コマンドを使用します。デフォルトのドメイン名は doc.com. であるため、cred テーブルの完全指定名を入力する必要はありません。接尾辞の org_dir で十分です。ルートマスターの資格を探すためには、その Secure RPC ネット名を探してください。

16. ルートドメインの管理グループを作成します。

この手順では、手順 9 で指定された管理グループを作成します。nisgrpadm コマンドに -c オプションを付けて実行してください。次の例では admin.doc.com. グループを作成します。

```
rootmaster# nisgrpadm -c admin.doc.com.
Group admin.doc.com. created.
```

この手順ではグループを作成するだけであり、そのメンバー名の指定は行いません。これについては、手順 17 で行います。グループのオブジェクト属性を見るには、niscat -o を使用します。しかし、グループ名では必ず groups_dir を使用します。たとえば、次のようになります。

```
doc.com.
Object Name : admin
Directory : groups_dir.doc.com
Owner : rootmaster.doc.com.
Group : admin.doc.com.
Domain : groups_dir.doc.com.
Access Rights : ----rmdr---r---
Time to Live : 1:0:0
Object Type : GROUP
Group Flags :
Group Members :
```

17. ルートドメインの管理 (admin) グループにルートマスターを追加します。

この時点では、このルートマスターサーバーは DES 資格をもつ唯一の NIS+ 主体であるため、管理グループに追加すべき唯一のメンバーです。今度は、`nisgrpadm` コマンドに `-a` オプションを付けて実行します。第 1 引数はグループ名、第 2 引数はルートマスターサーバー名です。この例では `doc.com` ドメインに `rootmaster.doc.com.` を追加します。

```
rootmaster# nisgrpadm -a admin.doc.com. rootmaster.doc.com.
```

Added rootmaster.doc.com. to group admin.doc.com.

ルートマスターがこの管理グループに属していることを確認するには、`nisgrpadm` コマンドを、`-l` オプションを指定して実行します。第 17 章を参照してください。

注 `-nisgrpadm` などのグループ関連コマンドでは、名前に `groups_dir` サブディレクトリを含める必要はありません。`niscat` などのコマンドの場合、NIS+ オブジェクト一般に対して動作するように設計されているため、このディレクトリを含む必要があります。グループ関連コマンドは、`groups_dir` サブディレクトリを対象としています。

```
rootmaster# nisgrpadm -l admin.doc.com.
```

Group entry for admin.doc.com. group:

Explicit members:

rootmaster.doc.com.

No implicit members

No recursive members

No explicit nonmembers

No implicit nonmembers

No recursive nonmembers

18. ルートドメインの公開鍵を更新します。

ディレクトリオブジェクトは、すでに DES 資格をもつ NIS+ 主体によって作成されるのが普通です。しかしこの場合、`cred` テーブルを作成するまでは、自分の資格を格納する親ドメインがないため、ルートマスターサーバーは DES 資格を獲得できません。その結果、`root`、`org_dir`、および `groups_dir` という 3 つのディレクトリオブジェクトには、ルートマスターサーバーの公開鍵のコピーがありません。これについては、`niscat -o` コマンドにどれかのディレクトリオブジェクトを指定して実行することによって確認できます。「Public Key:」フィールドを探してください。手順については、第 18 章を参照してください。

ルートマスターサーバーの公開鍵を、ルートドメインの `cred` テーブルからこれら 3 つのディレクトリオブジェクトに伝達するには、次に示すように、各ディレクトリオブジェクトに対して `/usr/lib/nis/nisupdkeys` ユーティリティを使用します。

```
rootmaster# /usr/lib/nis/nisupdkeys doc.com.
```

```
rootmaster# /usr/lib/nis/nisupdkeys org_dir.doc.com.
```

```
rootmaster# /usr/lib/nis/nisupdkeys groups_dir.doc.com.
```

コマンドを実行するたびに、次のような確認メッセージが表示されます。

```
Fetch Public key for server rootmaster.doc.com.  
netname = 'unix.rootmaster@doc.com.'  
Updating rootmaster.doc.com.'s public key.  
Public key:
```

これらのディレクトリを (niscat -o を使用して) 表示すると、「Public key:」フィールドに 1 つまたは複数のエントリが見つかります。

```
Public key: Diffie-Hellman (192 bits)
```

19. NIS+ サービスを再起動します。

これで、ルートマスターサーバーには DES 資格があり、ルートディレクトリオブジェクトにはルートマスターの公開鍵のコピーがあるため、ルートマスターを再起動すると、自動的にセキュリティレベル 2 で起動されます。

サービス管理機能では、/var/nis/NIS_COLD_START ファイルが検出されると、NIS+ サービスを有効にしたときに nis_cachemgr を自動的に起動します。

NIS と互換性のあるルートドメインの場合は、必ず /lib/svc/method/nisplus ファイルを編集して -Y フラグを追加してください。詳細については、[89 ページ](#)の「NIS+ とサービス管理機能」を参照してください。

```
rootmaster# svcs /*nisplus/*  
online Jan_12 svc:/network/rpc/nisplus:default  
rootmaster# svcadm restart network/rpc/nisplus:default
```

セキュリティレベル 2 はデフォルトであるため、-s 2 フラグは不要です。

注 - 実際にユーザーが存在するネットワークは、常にセキュリティレベル 2 で運用してください。セキュリティレベル 0 およびレベル 1 は、設定およびテスト目的だけに使用します。本番環境のネットワークは、レベル 0 または 1 で稼働しないでください。

20. 自分の LOCAL 資格をルートドメインに追加します。

ユーザーはルートドメインの cred テーブルに対するアクセス権がないため、この動作はスーパーユーザーとして実行しなければなりません。さらに、「前提条件」で説明したように、ルートマスターの /etc/passwd ファイルには自分用のエントリが必要です。nisaddcred コマンドに -p と -P フラグを付けて使用します。その構文と例を次に示します。

```
nisaddcred -p uid -P principal-name local
```

principal-name はシステム管理者のログイン名とドメイン名で構成されます。この例では、UID が 11177 で NIS+ 主体名が topadmin.doc.com. のシステム管理者用の LOCAL 資格を追加します。

```
rootmaster# nisaddcred -p 11177 -P topadmin.doc.com. local
```

nisaddcred コマンドの詳細については、[第 12 章](#)を参照してください。

21. 自分の DES 資格をルートドメインに追加します。

ここでも `nisaddcred` コマンドを使用しますが、次のような構文となります。

```
nisaddcred -p secure-RPC-netname -P principal-name des
```

`secure-RPC-netname` は、接頭辞 `unix` に自分のユーザー ID、`@` 記号、およびドメイン名を付けて構成しますが、最後にドットは付けません。`principal-name` は LOCAL 資格の場合と同じで、ログイン名にドメイン名を付け、さらに最後にドットを付けます。

```
rootmaster# nisaddcred -p unix.11177@doc.com -P topadmin.doc.com. des
Adding key pair for unix.11177@doc.com (topadmin.doc.com.).
Enter login password:
```

ログインパスワードの入力後に「password that differs from the login password」という警告が表示され、入力したパスワードが正しいログインパスワードである場合は、このエラーメッセージを無視してください。NIS+ がパスワードの格納されている、保護された `/etc/shadow` ファイルを期待どおりに読み込めないため、このメッセージが表示されます。ユーザーパスワード情報を `/etc/passwd` ファイルに格納していない場合、このメッセージは表示されません。

22. ほかのシステム管理者の資格を追加します。

そのルートドメインで作業するほかのシステム管理者の LOCAL 資格と DES 資格を追加します。これには 以下の方法があります。

- ほかの管理者に一時的な資格を作成する場合は、NIS+ モードで実行されている Solaris 管理コンソール (使用できる場合) を使用すると簡単です。
- 2 つめは、ほかの管理者にその管理者自身の資格を追加するよう要求する方法です。この方法は、スーパーユーザーとして実行する必要があります。ユーザー ID が 33355 で主体名が `miyoko.doc.com.` の管理者の資格を追加する例を次に示します。

```
rootmaster# nisaddcred -p 33355 -P miyoko.doc.com. local
rootmaster# nisaddcred -p unix.33355@doc.com -P miyoko.doc.com. des
Adding key pair for unix.33355@doc.com (miyoko.doc.com.).
Enter login password:
```

- 3 つめは、ダミーのパスワードを使用して、ほかの管理者に一時的な資格を作成する方法です。ほかの管理者、この例では `miyoko` は、NIS+ `passwd` テーブルにエントリがなければなりません。エントリがない場合は、まず `nistbladm` を使用して、必ずエントリを作成してください。次の例に、その手順を示します。

```
rootmaster# nistbladm -D owner=miyoko.doc.com. name=miyoko uid=33355 gcos=miyoko
home=/home/miyoko shell=/bin/tcsh passwd.org_dir
rootmaster# nisaddent -a -f /etc/passwd.xfr passwd
rootmaster# nisaddent -a -f /etc/shadow.xfr shadow
rootmaster# nisaddcred -p 33355 -P miyoko.doc.com. local
rootmaster# nisaddcred -p unix.33355@doc.com -P miyoko.doc.com. des
Adding key pair for unix.33355@doc.com (miyoko.doc.com.).
Enter miyoko's login password:
nisaddcred: WARNING: password differs from login passwd.
Retype password:
rootmaster# nischown miyoko.doc.com. '[name=miyoko],passwd.org_dir'
```


この場合、最初の `nisaddent` で `passwd` テーブルにエントリが生成されます (`shadow` 列は除く)。次の `nisaddent` により、`shadow` 列が生成されます。各システム管理者は、`chkey` コマンドを使用することによって、自分のネットワークパスワードを後で変更できます。この手順については、[第 12 章](#)を参照してください。

23. ルートドメインの管理グループに自分とほかのシステム管理者を追加します。

この手順を実行するには、ほかのシステム管理者がダミーパスワードを変更するまで待つ必要はありません。-a オプションを付けて `nisgrpadm` コマンドを実行します。最初の引数はグループ名であり、残りの引数はシステム管理者の名前です。この例では、`topadmin` と `miyoko` の 2 人のシステム管理者を `admin.doc.com`. グループに追加します。

```
rootmaster# nisgrpadm -a admin.doc.com. topadmin.doc.com. miyoko.doc.com.
Added topadmin.doc.com. to group admin.doc.com.
Added miyoko.doc.com. to group admin.doc.com.
```

24. NIS+ テーブルを格納するための、十分なスワップ空間を割り当てます。

スワップ空間は、`rpc.nisd` の最大サイズの 2 倍にする必要があります。`rpc.nisd` が使用するメモリー量を調べるには、次のコマンドを実行してください。

```
rootmaster# /usr/lib/nis/nisstat
```

`rpc.nisd` は、特定の条件のもとでは、自らのコピーを作成してフォークします。メモリーが不足すると、`rpc.nisd` は正しく動作しません。

また、NIS+ テーブルに必要なメモリーとスワップ空間のサイズも計算できます。たとえば、NIS+ テーブル内に、180,000 人のユーザーと 180,000 台のホストがある場合、これらの 2 つのテーブルが占有するメモリーは、約 190M バイトです。180,000 人のユーザーと 180,000 台のホストに資格を追加する場合、`cred` テーブルには、540,000 のエントリ (ユーザーごとにローカルの資格と DES の資格、合わせて 2 つの資格、ホストごとに 1 つの資格) が入ります。そのため、`cred` テーブルが占有するメモリーは、約 285M バイトになります。この例では、`rpc.nisd` に必要なメモリー容量は、少なくとも、190M バイト + 285M バイト = 475M バイトになります。この結果、少なくとも 1G バイトのスワップ空間が必要になります。また、`rpc.nisd` 全体をすべてメモリー内に保持するには、少なくとも 500M バイトが必要です。

ルートドメイン構成の要覧

ルートドメインの構成に必要な手順を[表 5-2](#)にまとめます。このまとめは、簡単な場合を想定しています。このまとめを参考として使用する前に、作業全体の説明を十分に理解しておいてください。また、ここでは、各コマンドに対するサーバーの応答を示していません。

表 5-2 ルートドメインを設定する - 操作のまとめ

作業	操作
ルートマスターサーバーにスーパーユーザーとしてログインする	rootmaster% su Password:
ドメイン名をチェックする	# domainname
スイッチファイルをチェックする	# more /etc/nsswitch.conf
(省略可能) Diffie-Hellman キー長を設定する	# nisauthconf dh640-0 des
スイッチファイルを変更した場合は、nscd を再起動する	# svcadm restart /system/name-service-cache
/etc/.rootkey を削除し、keyserv を再起動する	# svcadm disable /network/rpc/keyserv # rm -f /etc/.rootkey # svcadm enable /network/rpc/keyserv
NIS+ サービスを停止する	# svcadm disable /network/rpc/nisplus
NIS+ データを削除し、プロセスを終了する	# rm -rf /var/nis*
管理グループを指定する	# NIS_GROUP=admin.doc.com.; export NIS_GROUP
ルートマスターサーバーを初期設定する	# nisinit -r
NIS 互換モードの場合のみ -s 0 および -Y オプションを使ってデーモンを起動する	/lib/svc/method/nisplus ファイルを編集して -s 0 および -Y オプションを追加してから、次のようにサービスを再起動する # svcadm restart network/rpc/nisplus
NIS+ の場合のみ -s 0 を使ってデーモンを起動する	/lib/svc/method/nisplus ファイルを編集して -s 0 オプションを追加してから、次のようにサービスを有効にする # svcadm enable network/rpc/nisplus
ルートオブジェクトが作成されているかどうか確認する	# ls -l /var/nis/data # niscat -o doc.com.
org_dir と、groups_dir テーブルを作成する	# /usr/lib/nis/nissetup [-Y]
ルートマスターサーバー用の DES 資格を作成する	# nisaddcred des Enter login password:
管理グループを作成する	# nisgrpadm -c admin.doc.com.
ルートディレクトリに完全なグループ権を割り当てる	# nischmod g+rmcd doc.com.
ルートマスターを管理グループに追加する	# nisgrpadm -a admin.doc.com. rootmaster.doc.com.
ルートディレクトリの鍵を更新する	# /usr/lib/nis/nisupdkeys doc.com.

表 5-2 ルートドメインを設定する - 操作のまとめ (続き)

作業	操作
org_dir の鍵を更新する	# /usr/lib/nis/nisupdkeys org_dir.doc.com.
groups_dir の鍵を更新する	# /usr/lib/nis/nisupdkeys groups_dir.doc.com.
NIS+ サービスを再起動する	# svcadm restart network/rpc/nisplus
自分の LOCAL 資格を追加する	# nisaddcred -p 11177 -P topadmin.doc.com. local
自分の DES 資格を追加する	# nisaddcred -p unix.11177@doc.com -P topadmin.doc.com. des Enter login password:
ほかのシステム管理者の資格を追加する	# nisaddcred ...
ほかのシステム管理者を管理グループに追加する	# nisgrpadm -a admin.doc.com members
スワップ空間を割り当てる	# /usr/lib/nis/nisstat

第 6 章

NIS+ クライアントの構成

この章では、NIS+ コマンド群を使用した 3 通りの初期設定方法で NIS+ クライアントを設定する手順を説明します。この章の内容は、NIS+ モード、NIS+ 互換モードを問わず、ルートドメイン、サブドメインに共通するものです。

注 - NIS+ は、将来のリリースではサポートされなくなる可能性があります。NIS+ から LDAP への移行支援ツールは、Solaris 9 リリース以降で使用できます (『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』を参照)。詳細については、<http://www.sun.com/directory/nisplus/transition.html> を参照してください。

NIS+ クライアントの設定の概要

この章では、標準の NIS+ ドメインと NIS 互換ドメイン内のクライアントを構成する方法を説明します。各手順では詳細を説明し、また関連する情報についても説明します。詳しい手順の説明が必要ない場合は、表 6-6 で必要なコマンドの一覧を参照してください。

注 - NIS+ クライアントを設定する作業は、この章で説明する NIS+ のコマンドセットを使用する方法よりも、パート I で説明した NIS+ 設定スクリプトを使用する方が簡単です。この章で説明する方法は、NIS+ に精通した管理者や、設定スクリプトでは提供されない標準以外の機能や構成を必要とする管理者だけが使用してください。可能な場合は Solaris 管理コンソールを使用すると、NIS+ クライアントマシンの追加や設定の作業が簡単にできます。

クライアントを設定する作業のうち、[手順 11](#) では、ブロードキャスト、ホスト名、コールドスタートファイルのどれを使用するかを選択してください。それぞれ実装方法が異なるため、各作業について別々に説明します。クライアントを初期設定したあとは、[手順 12](#) に戻ってクライアントの設定を続けてください。

この章の最後の作業では、マシンのドメイン名を変更する方法を取り上げています。

クライアントの設定

ここでは、ルートドメイン内であるかどうかとは関係なく、一般的な NIS+ クライアントの構成方法を説明します。ここでの説明は、通常の NIS+ クライアント、および後で NIS+ サーバーとなるクライアントに当てはまります。また、標準の NIS+ ドメイン内、および NIS 互換ドメイン内のクライアントにも当てはまります。



注意 - またドメイン名とホスト名が同じにならないようにします。たとえば、sales ドメインがある場合は、ホスト名に sales を使用することはできません。同様に、home というホスト名がある場合には、ドメイン名に home を使用できません。これは、サブドメインについても同様です。たとえば、すでにホスト名として west がある場合には、sales.west.doc.com というサブドメインを作成することはできません。

NIS+ クライアントの設定には、次の作業が必要です。

- クライアントの資格の作成
- マシンの準備
- マシンを NIS+ クライアントとして初期設定

ただし、ルートドメインの設定と同様、クライアントの設定も、これら 3 つの作業を順番に実行するような単純なものではありません。構成手続を実行しやすくするため、これらの作業を個々の手順に分割し、次に示すように、これらの手順をもっとも効率的な順に並べてあります。

1. ドメインのマスターサーバーにログインします。
2. 新しいクライアントマシン用の DES 資格を作成します。
3. マスターサーバーで使用されている Diffie-Hellman キー長を確認します。
4. クライアントにスーパーユーザーとしてログインします。
5. クライアントに新しいドメイン名を設定します。
6. nsd の停止と再起動を行います。
7. クライアントの nsswitch.conf ファイルの設定を確認します。
8. クライアントの Diffie-Hellman キーを設定します。
9. NIS+ のファイルを削除し、プロセスを終了します。
10. クライアントを初期設定します。
11. /etc/.rootkey ファイルを削除し、keyserv デーモンを再起動します。

12. `keylogin` を実行します。
13. クライアントを再起動します。

クライアント構成時のセキュリティについて

クライアントの設定には、セキュリティ上の主な必要要件が2つあります。つまり、システム管理者とクライアントの両方が、適切な資格とアクセス権を持つことです。そうでない場合、クライアントがセキュリティレベル2で実行しているドメインの資格を入手する唯一の方法は、クライアントのホームドメイン内での有効な DES 資格と `cred` テーブルに対する変更権とを持つシステム管理者が資格を作成することです。システム管理者は DES 資格を、クライアントのホームドメイン内、または自分のホームドメイン内に所持できます。

システム管理者がクライアントの資格を作成すると、そのクライアントは構成プロセスを終了できます。しかしクライアントは、ホームドメインのディレクトリオブジェクトに対する読み取り権を必要とします。第5章または第8章の手順に従ってクライアントのホームドメインを構成した場合、ディレクトリオブジェクトの作成に使用した NIS+ コマンド (`nisinit` と `nismkdir`) によって、読み取り権がその他のクラスに提供されています。

ディレクトリオブジェクトのアクセス権をチェックするには、`niscat -o` コマンドを使用します。このコマンドは、アクセス権などのディレクトリ属性を表示します。次にその例を示します。

```
rootmaster# niscat -o doc.com.
ObjectName : Doc
Owner : rootmaster.doc.com.
Group : admin.doc.com.
Domain : Com.
Access Rights : r---rmcdr---r---
```

ディレクトリオブジェクトのアクセス権は、オブジェクトに対する変更権を持っている場合は、`nischmod` コマンドを使用して変更できます。詳細については、第15章を参照してください。

クライアントの資格を構成するための前提条件

クライアントの資格を設定するシステム管理者は、次の条件をすべて満たしている必要があります。

- 有効な DES 資格を持っていること
- クライアントのホームドメインにある `cred` テーブルを修正する権利を持っていること

クライアントは次の条件をすべて満たしている必要があります。

- ホームドメインのディレクトリオブジェクトに対する読み取り権を持っていること

- クライアントのホームドメインがあらかじめ構成されており、NIS+ を実行していること
- マスターサーバーの /etc/hosts ファイル、または /etc/inet/ipnodes ファイル、あるいはドメインの hosts テーブルまたは ipnodes テーブルの中にエントリが存在すること
- マシン名が、どのユーザーの ID とも重複しておらず、固有であること
- マシン名にドットが含まれていないこと。たとえば、sales.alpha というマシン名は使用できませんが、sales-alpha なら使用できます。

必要な情報

- クライアントのホームドメイン名
- クライアントとなるマシンのスーパーユーザーのパスワード
- クライアントのホームドメイン内にある NIS+ サーバーの IP アドレス

クライアントの設定 — 作業マップ

注 - NIS+ サービスは、サービス管理機能 (SMF) によって管理されます。このサービスに対する有効化、無効化、再起動などの管理操作を実行するには、svcadm コマンドを使用します。SMF を NIS+ で使用する方法については、[89 ページの「NIS+ とサービス管理機能」](#)を参照してください。SMF の概要については、『Solaris のシステム管理 (基本編)』の「サービスの管理 (概要)」を参照してください。詳細については、svcadm (1M) と svcs (1) の各マニュアルページも参照してください。

表 6-1 クライアントの設定

作業	目的	参照先
クライアントの設定	クライアントの資格を作成する。クライアントマシンを準備して、NIS+ クライアントとして初期設定する	152 ページの「NIS+ クライアントを設定する方法」

▼ NIS+ クライアントを設定する方法

1. ドメインのマスターサーバーにログインする
スーパーユーザーとして、または自分自身のユーザー名でログインします。どちらでログインするかは、どちらの NIS+ 主体がドメインの cred テーブルに資格を追加するための適切なアクセス権を所有しているのかに依存します。
2. 新しいクライアントマシン用の DES 資格を作成します。

`nisaddcred` コマンドを引数 `-p` と `-P` を付けて実行します。構文は、次のとおりです。

```
nisaddcred -p secure-RPC-netname principal-name des [domain]
```

secure-RPC-netname は、接頭辞 `unix` に、クライアントのホスト名、`@` 記号、およびクライアントのドメイン名を付けて構成しますが、最後にドットは付けません。*principal-name* は、クライアントのホスト名とドメイン名によって構成され、最後にドットを付けます。このクライアントの所属するドメインが、コマンドを入力したサーバーとは異なる場合、2 番目の引数の後にクライアントのドメイン名を追加します。

この例では、`doc.com` ドメイン内の `client1` という名前のクライアントマシンに対する DES 資格を追加します。

```
rootmaster% nisaddcred -p unix.client1@doc.com -P client1.doc.com. des
Adding key pair for unix.client1@doc.com (client1.doc.com.).
Enter client1.doc.com.'s root login passwd:
Retype password:
```

`nisaddcred` コマンドの詳細については、[第 12 章](#)を参照してください。

3. マスターサーバーで使用される **Diffie-Hellman** キー長を確認します。
たとえば、以下のようになります。

```
rootmaster% nisauthconf dh640-0 des
```

4. クライアントにスーパーユーザーとしてログインします。
これでクライアントマシンに資格ができたため、ユーザーはマスターサーバーからログアウトし、クライアント自体から作業を開始できます。この作業はローカルでもリモートでも可能です。
5. クライアントに新しいドメイン名を設定します。
クライアントのドメイン名を設定する (変更する) 方法については、[155 ページ](#)の「[マシンのドメイン名を変更する](#)」を参照し、次の[手順 6](#)に戻ります。
6. クライアントの **nsswitch.conf** ファイルをチェックします。
クライアントが NIS+ バージョンの **nsswitch.conf** ファイルを使用していることを確認します。これによって、クライアント情報の 1 次ソースが NIS+ テーブルということが確認できます。NIS+ スイッチファイルの詳細については、[例 1-1](#)を参照してください。
7. **nsswitch.conf** ファイルに少しでも変更を加えた場合 (または新規にファイルにコピーした場合)、ここで **nsd** を再起動する必要があります。

```
client1# svcadm restart /system/name-service-cache
```

キーサーバーをこの時点で停止および再起動する必要はありません。[手順 12](#)を行います。

8. [手順 3](#) の情報を使用して、**Diffie-Hellman** キー長を設定します。
たとえば、次のようにします。

```
client# nisauthconf dh640-0 des
```

9. NIS+ サービスを停止します。

```
client1# svcadm disable network/rpc/nisplus:default
client1# svcs \*nisplus\*
disabled   Jan_12   svc:/network/rpc/nisplus:default
```

10. NIS+ のファイルを削除しプロセスを終了します。

現在作業しているマシンが、以前は NIS+ のサーバーまたはクライアントとして使用したものである場合、/var/nis 内にファイルがあればすべて削除します。この例では、/var/nis 内にコールドスタートファイルとディレクトリ キャッシュファイルがまだ存在します。

```
client1# ls /var/nis
NIS_COLD_START NIS_SHARED_CACHE
client1# rm -rf /var/nis/*
```

/var/nis 内に残されたファイル、またはキャッシュマネージャによって保存されたディレクトリオブジェクトは、この手順によって完全に消去されます。したがって、この構成プロセスで生成された新しい情報と重複することはありません。/var/nis 内に管理スクリプトを 1 つでも格納していた場合、ルートドメインの設定が終わるまでは、それらを一時的にほかのどこかに格納しておくことをお勧めします。

11. クライアントを初期設定します。

クライアントを初期設定するには、次の 3 つの方法があります。ホスト名、コールドスタートファイル、ブロードキャストによる方法です。3 つの方法のうち、いずれかを選択して実行します。クライアントの初期設定が終了したら、[手順 12](#)に進みます。

12. /etc/.rootkey ファイルを削除し、keyserv デーモンを再起動します。

この手順では、非公開鍵をキーサーバー上に格納します。

```
client1# cp /etc/nsswitch.nisplus /etc/nsswitch.conf
client1# svcs \*keyserv\*
online    Jan_12   svc:/network/rpc/keyserv:default
client1# svcadm disable network/rpc/keyserv
client1# rm -f /etc/.rootkey
client1# svcadm enable network/rpc/keyserv
```

DNS 転送の設定

▼ NIS+ クライアントの DNS 転送機能を有効にするには

1. スーパーユーザーとしてログインします。
2. /etc/resolve.conf ファイルの **hosts** 行を構成します。たとえば、**hosts:nisplus dns files** とします。

該当するサーバー上に /etc/resolve.conf ファイルが存在する場合、この NIS 実装では DNS へ要求を転送するために、ypstart によって ypserv デーモンが -d オプション付きで「自動的に」起動されます。

マシンのドメイン名を変更する

この作業では、マシンのドメイン名を変更します。マシンのドメイン名は通常インストール時に設定されるため、domainname コマンドを引数なしで実行して、マシンのドメイン名をチェックしてからこの作業を実行してください。

マシンのドメイン名変更時のセキュリティについて

この作業は、ドメイン名を変更するマシン上のスーパーユーザーとして実行しなければなりません。

必要な情報

- マシンのスーパーユーザーのパスワード
- 新しいドメイン名

マシンのドメインの変更 - 作業マップ

表 6-2 クライアントの設定

作業	目的	参照先
マシンのドメインの変更	nisrestore を使用して、クライアントマシンのドメインを変更する	155 ページの「クライアントのドメイン名を変更する方法」

▼ クライアントのドメイン名を変更する方法

1. マシンにログインして、スーパーユーザーになります。
この例では、マシンに client1 を、新しいドメイン名に doc.com. を使します。

```
client1% su
Password:
```

2. マシンのドメイン名を変更します。

`domainname` コマンドを使用して新しい名前を入力します。名前の最後にドットを入力しないでください。たとえば、マシンのドメインを `doc.com` に変更する場合、次のように入力します。

```
client1# domainname doc.com
```

マシンが NIS クライアントの場合は、NIS サービスを受けることはできません。

3. 結果を確認します。

今度は、引数を付けずに `domainname` コマンドを実行し、サーバーの現在のドメインを表示させます。

```
client1# domainname
doc.com
```

4. 新しいドメイン名を保存します。

`domainname` コマンドの出力を `/etc/defaultdomain` ファイルに書き込みます。

```
client1# domainname > /etc/defaultdomain
```

5. 適当な時に、マシンを再起動します。

`/etc/defaultdomain` ファイルに新しいドメイン名を入力した後でも、一部のプロセスは依然として古いドメイン名で動作している可能性があります。すべてのプロセスに新しいドメイン名を確実に使用させるため、マシンを再起動します。

この作業は、ほかのいくつかの作業の流れの中で行うものです。再起動は、マシンでのすべての作業が完了したことを確認してから行なってください。確認を怠ると、何度も再起動が必要になる可能性があります。

`mountd` などのデーモンを個別に再起動すれば、NFS の問題が解決されることがあります。ただし、構成の変更をデーモン間で同期化するために、マシンを再起動することを強くお勧めします。マシンを再起動することによって、構成の変更の非同期が原因で発生するアプリケーションの失敗を最小限に抑えることができます。

NIS+ クライアントを初期設定する

NIS+ クライアントを初期設定する方法には、以下の 3 つの種類があります。

- ブロードキャストを使用する方法 (157 ページの「ブロードキャストにより初期設定する」を参照)
- ホスト名を使用する方法 (158 ページの「ホスト名により NIS+ クライアントを初期設定する」を参照)
- コールドスタートファイルを使用する方法 (159 ページの「コールドスタートファイルを使用してクライアントを初期設定する」を参照)

ブロードキャストにより初期設定する

この方法では、クライアントの存在するサブネット上に IP ブロードキャストを送信して NIS+ クライアントを「初期化」します。

初期化の方法としてはこれがもっとも簡単ですが、もっとも安全性の低い方法でもあります。ブロードキャストに応答した NIS+ サーバーはクライアントが自分自身のコールドスタートファイルに格納する必要がある情報 (サーバーの公開鍵など) をすべて送信します。おそらくブロードキャストに応答するのは NIS+ サーバーですが、クライアントからは、ブロードキャストに応答したマシンが確かに信用できるサーバーなのかどうか確認できません。そのため、この方法は小規模で、セキュリティが確保されたサイトでだけ使用することをお勧めします。

クライアント初期設定時のセキュリティについて

この作業は、クライアント上のスーパーユーザーとして実行しなければなりません。

クライアントを初期設定するための前提条件

クライアントと同じサブネット上に、少なくとも 1 台の NIS+ サーバーが存在しなければなりません。クライアントは、マスターサーバーで使用するのと同じ Diffie-Hellman キー長を使用する必要があります。nisauthconf (1M) を参照してください。

必要な情報

クライアントのスーパーユーザーのパスワードが必要です。

NIS+ クライアントを初期設定する — 作業マップ

表 6-3 NIS+ クライアントを初期設定する

作業	目的	参照先
NIS+ クライアントを初期設定する	nisclient コマンドを使用して NIS+ クライアントを初期設定する	152 ページの「NIS+ クライアントを設定する方法」

ブロードキャストによりクライアントを初期設定する方法

- クライアントを初期設定します。

この手順では、クライアントを初期設定し、その /var/nis ディレクトリ内に NIS_COLD_START ファイルを作成します。nisinit コマンドに -c と -B のオプションを付けて実行します。

```
client1# nisinit -c -B
This machine is in the doc.com. NIS+ domain.
Setting up NIS+ client ...
All done.
```

同じサブネット上の NIS+ サーバーがブロードキャストに応答し、その位置情報をクライアントのコールドスタートファイルに追加します。

ホスト名により NIS+ クライアントを初期設定する

クライアントをホスト名によって初期化する場合、信頼できるサーバーの IP アドレスを明確に指摘します。そしてこのサーバー名、位置情報、公開鍵がクライアントのコールドスタートファイルに格納されます。

この方法は、クライアントがサーバーの IP アドレスを指定するので、自分で自分を識別してくるサーバーに応答するブロードキャストよりも安全です。しかし、クライアントと信頼できるサーバーの間にルーターが介在している場合、正しい IP アドレスへのメッセージを横取りし、不正なサーバーに送ることもあり得ます。

ホスト名によりクライアントを初期設定する場合のセキュリティについて

この作業は、クライアント上のスーパーユーザーとして実行しなければなりません。

ホスト名によりクライアントを初期設定するための前提条件

- NIS+ サービスはクライアントのドメインで実行されていなければなりません。
- クライアントは、/etc/hosts ファイル内に信頼できるサーバーのエントリを持っていなければなりません。
- クライアントは、マスターサーバーで使用するのと同じ Diffie-Hellman キー長を使用する必要があります。nisauthconf(1M) を参照してください。

必要な情報

信頼できるサーバー名と IP アドレスが必要です。

#

コールドスタートファイルを使用してクライアントを初期設定するための前提条件

COLD_START ファイルに指定されたサーバーは、すでに構成されており、NIS+ を実行していなければなりません。

クライアントは、マスターサーバーで使用するのと同じ Diffie-Hellman キー長を使用する必要があります。nisauthconf(1M) を参照してください。

必要な情報

コピーする COLD_START ファイルの名前と位置が必要です。

コールドスタートファイルを使用して NIS+ クライアントを初期設定する —作業マップ

表 6-5 コールドスタートファイルを使用して NIS+ クライアントを初期設定する

作業	目的	参照先
コールドスタートファイル経由でクライアントを初期設定する	nisinit コマンドを使って、NIS+ クライアントをコールドスタートファイル経由で初期設定する	160 ページの「COLD_START 経由で NIS+ クライアントを初期設定する方法」

▼ COLD_START 経由で NIS+ クライアントを初期設定する方法

1. ほかのクライアントの **COLD_START** ファイルをコピーします。

ほかのクライアントの COLD_START ファイルを、新しいクライアントのディレクトリにコピーします。これを行うには、クライアント上のスーパーユーザーとしてではなく、自分のユーザー名でログインしている間に行う方が簡単です。クライアントを初期設定する前に、必ずスーパーユーザーになってください。

ただし、NIS_COLD_START ファイルを /var/nis にコピーしないでください。初期設定中にこのファイルは上書きされます。次の例では、client1 のコールドスタートファイルを、初期設定されていない client2 の /tmp ディレクトリにコピーします。

```
client2# exit
client2% rcp client1:/var/nis/NIS_COLD_START /tmp
client2% su
```

2. **COLD_START** ファイルからクライアントを初期設定します。

次に示すように、nisinit コマンドに -c と -C のオプションを付けて実行します。

```
client2# nisinit -c -C /tmp/NIS_COLD_START
This machine is in the doc.com. NIS+ domain.
```



```
Setting up NIS+ client ...
All done.
```

NIS+ クライアント構成の要覧

クライアントの構成に必要な手順を表 6-6 にまとめます。クライアントは doc.com ドメインにある client1 とします。これはもっとも簡単なケースを想定しているため、このまとめを参考用として使用するには、その前に自分の実際の作業の詳細を理解する必要があります。簡略化のため、ここでは各コマンドに対するサーバーの応答を示していません。

表 6-6 クライアントを設定する - コマンドのまとめ

作業	コマンド
ドメインのマスターサーバーにログインする	rootmaster%
クライアントの DES 資格を作成する	rootmaster% nisaddcred -p unix.client1.doc.com -P client1.doc.com. des
Diffie-Hellman キー長を確認する	rootmaster% nisauthconf
クライアントにスーパーユーザーとしてログインする	client1% su Password:
クライアントのドメイン名を設定する	client1# domainname doc.com client1# domainname > /etc/defaultdomain
クライアントのスイッチ構成ファイルが正しく設定されていることをチェックする	client1# more /etc/nsswitch.conf
Diffie-Hellman キー長を設定する	client1# nisauthconf dh640-0 des
/var/nis の下のファイルを削除する	client1# rm -rf /var/nis/*
クライアントを初期設定する	client1# nisinit -c -H rootmaster

表 6-6 クライアントを設定する - コマンドのまとめ (続き)

作業	コマンド
/etc/.rootkey ファイルを削除し、キーサーバーを再起動する	<pre>client1# svcadm disable network/rpc/keyserv client1# rm -f /etc/.rootkey client1# svcadm enable network/rpc/keyserv</pre>
クライアント上でキーログインを実行する	<pre>client1# keylogin -r password:</pre>
クライアントを再起動する	<pre>client1# reboot</pre>

第 7 章

NIS+ サーバーの構成

この章では、NIS+ コマンドセットを使って NIS+ サーバーを設定する手順と、既存の NIS+ ドメインに複製サーバーを追加する手順を説明します。

注 - NIS+ は、将来のリリースではサポートされなくなる可能性があります。NIS+ から LDAP への移行支援ツールは、Solaris 9 リリース以降で使用できます (『Solaris のシステム管理 (ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』を参照)。詳細については、<http://www.sun.com/directory/nisplus/transition.html> を参照してください。

NIS+ サーバーを設定する

NIS+ サーバーの初期設定は、この章で説明する NIS+ コマンドよりも、NIS+ スクリプトを使用した方が簡単に行うことができます。この章で説明する方法は、NIS+ に精通した管理者や、設定スクリプトでは提供されない標準以外の機能や構成を必要とする管理者だけが使用してください。

注 - NIS+ サービスは、サービス管理機能 (SMF) によって管理されます。このサービスに対する有効化、無効化、再起動などの管理操作を実行するには、svcadm コマンドを使用します。SMF を NIS+ で使用する方法については、89 ページの「NIS+ とサービス管理機能」を参照してください。SMF の概要については、『Solaris のシステム管理 (基本編)』の「サービスの管理 (概要)」を参照してください。詳細については、svcadm(1M) と svcs(1) の各マニュアルページも参照してください。

標準構成と NIS 互換構成の相違 (NIS+ サーバー)

NIS 互換の NIS+ サーバーと標準の NIS+ サーバーの設定における違いは、ルートマスターサーバーの場合と同じです (132 ページの「標準構成と NIS 互換構成の手順の相違」を参照)。サーバーには、正しく構成された `/etc/resolv.conf` ファイルが必要です。また、NIS 互換サーバー用の NIS+ デーモンは `-Y` オプション (DNS 転送を使用する場合は、`-B` オプションも必要) を使用して起動しなければなりません。これによって、サーバーは NIS クライアントからの要求に応答できます。`-Y` オプションと `-B` オプションの実装方法については、89 ページの「NIS+ とサービス管理機能」を参照してください。

注 - `-Y` または `-B` のいずれかのオプションを使用して `rpc.nisd` を起動した場合、必ず `rpc.nisd_resolv` という副デーモンが生成され、名前の解決を行います。

設定作業の手順を次にまとめます。

1. 新しくサーバーにするワークステーションにスーパーユーザーとしてログインします。
2. NIS+ デーモンを `-Y` で起動します (NIS 互換のみの場合)。
3. NIS+ デーモンを起動します (標準の NIS+ のみの場合)。

NIS+ サーバー構成時のセキュリティについて

注 - NIS+ のセキュリティシステムは複雑です。NIS+ セキュリティを使い慣れていない場合は、第 11 章を参照してから NIS+ 環境を構成することをお勧めします。

この手順は、サーバー上のスーパーユーザーとして実行しなければなりません。起動したサーバーのセキュリティレベルによって、そのクライアントが備えるべき資格が決まります。たとえば、サーバーがセキュリティレベル 2 で構成された場合、サーバーがサポートするドメイン内のクライアントは、DES 資格を必要とします。このマニュアルの指示に従ってクライアントを構成した場合、そのクライアントは適切なドメインに DES 資格を持ち、セキュリティレベル 2 でサーバーを起動できます。

注 - セキュリティレベル 0 は、管理者による構成とテストの目的だけに使用します。セキュリティレベル 1 はサポートされていません。一般のユーザーが通常の業務を行う環境では、レベル 0 またはレベル 1 を使用せず、常にセキュリティレベル 2 を使用してください。

NIS+ サーバーを構成するための前提条件

- ルートドメインがあらかじめ構成されている (第 5 章を参照)
- サーバーにするには、NIS+ クライアントとして初期設定しておく (第 6 章を参照)
- サーバーを構成するには、そのマシンにスーパーユーザーとしてログインする必要がある
- サーバーを NIS 互換モードで実行し、DNS 転送をサポートするためには、正しく構成された `/etc/resolv.conf` ファイルが必要である

必要な情報

サーバーに変換するクライアントのスーパーユーザーパスワードが必要です。

▼ NIS+ サーバーを構成する方法

1 台のマスターサーバーまたは複製サーバーから複数のドメインにサービスを提供することは可能ですが、お勧めしません。

1. 新しくサーバーにするワークステーションにスーパーユーザーとしてログインします。

以下の手順では、150 ページの「クライアントの設定」に従って、マシンを NIS+ クライアントとして設定した後、マシンを再起動したことを前提としています。マシンを再起動すると、次の手順の推奨前提条件であるキャッシュマネージャが起動します。マシンを再起動しなかった場合は、ここで `svcadm` を使用して NIS+ サービスを再起動します。

2. (省略可能) `/lib/svc/method/nisplus` ファイルを編集して必要なオプションを追加します。

適切なテキストエディタを使用します。

`/lib/svc/method/nisplus` ファイルの編集方法については、89 ページの「NIS+ とサービス管理機能」を参照してください。

<code>-B</code>	DNS 転送をサポートします。
<code>-Y</code>	NIS 互換モードで NIS+ デーモンを起動します。

3. NIS+ デーモンを起動します。

```
server# svcadm enable network/rpc/nisplus:default
```

NIS+ サービスが実行されていることを確認するには、`svcs` コマンドを実行します。

```
server# svcs \*nisplus\*
STATE      STIME      FMRI
```

```
online      Jan_12   svc:/network/rpc/nisplus:default
```

この手順によって、/var/nis/data という名前のディレクトリが作成されます。また、trans.log というトランザクションログファイルが作成され、/var/nis というディレクトリに格納されます。

```
compatserver# ls -F /var/nis
NIS_COLD_START data/ trans.log data.dict
```

trans.log ファイルは、トランザクションログです。トランザクションログの内容を確認するには、nislog コマンドを使用します。使用方法については、[346 ページの「nislog コマンド」](#)を参照してください。



注意 - /var/nis ディレクトリと /var/nis/data ディレクトリは、移動または名前の変更をしないでください。また、/var/nis/trans.log ファイルと /var/nis/data.dict ファイルについても、移動または名前の変更をしないでください。Solaris 2.4 以前からアップグレードする場合、それまで使っていた /hostname サブディレクトリは自動的に /var/nis/data に変換され、関連するファイルも必要に応じて変換されます。この自動変換がなされた後で、新しい名前に変更することは絶対にしないでください。

これでこのサーバーは、[第 8 章](#)の説明に従って、ドメインのマスターまたは複製に指定できます。NIS+ サーバーの設定は、この手順で完了です。作業の要約については [173 ページの「サーバー構成の要覧」](#)を参照してください。

複製サーバーを既存のドメインに追加する

NIS+ サービスを常にご利用できる状態にしておきたいのであれば、ルート複製サーバーを少なくとも 1 つは作成しておくことをお勧めします。複製サーバーを作成すると複数のサーバーが存在することになり、要求の処理を分散させることができるため、ネットワーク要求の処理も高速化されます。

パフォーマンス上の理由から、1 つのドメインに多くの複製サーバーを置くことはお勧めできません。ネットワークが複数のサブネット構成されている場合、あるいは広域ネットワーク (WAN) でリモートサイトに接続されている場合にだけ、複製サーバーを置くようにしてください。

- 「サブネット」。複数のサブネット構成されているドメインの場合、各サブネットに複製サーバーを少なくとも 1 つは作成することをお勧めします。そうしておけば、ネットワーク間の通信が一時的に途絶していても、接続が回復するまでの間、サブネットレベルの機能は維持されるからです。
- 「リモートサイト」。WAN によりリモートサイトに接続されているドメインの場合、WAN 接続の両側に複製サーバーを少なくとも 1 つは作成することをお勧めします。組織論的な見地からしても、同一の NIS+ ドメインに物理的に離れた 2 つの

サイトがあるのは意味のあることです。たとえば、ドメイン内のマスターサーバーとその複製サーバーがすべて一方のサイトに置かれている場合、そのサイトともう一方のサイトとの間の NIS+ ネットワークトラフィックが増大するのは目に見えています。もう一方のサイトにも複製サーバーを置いておけば、ネットワークトラフィックが減るはずですが。

複製サーバーの分散および最適な複製サーバー数の決定方法については、『Solaris のシステム管理 (ネーミングとディレクトリサービス: NIS+ 編)』の「ルート複製サーバーの作成」を参照してください。既存のドメインに複製サーバーを追加するには、その複製サーバーを構成してから該当する名前空間の NIS+ データセットをロードします。

新しい複製サーバーを構成して NIS+ データセットをロードする方法には、次の 2 通りがあります。

- 「スクリプト」。nisserver スクリプトを実行するには、113 ページの「ルート複製サーバーの作成」の説明に従ってください。この方法では、NIS+ データセットが新しい複製サーバーにロードされて自動的に再同期がとられます。格段に簡単なので、こちらの方法をお勧めしますが、「NIS+ コマンドセット」と「バックアップと復元」を利用する方法に比べると、時間が長くなることがあります。
- 「NIS+ コマンドセット」。NIS+ コマンドを使うには、168 ページの「NIS+ コマンドを使って複製サーバーを構成する」の説明に従ってください。nisserver スクリプトを実行する方法に比べると、NIS+ に対する深い知識が必要です。この NIS+ コマンドを使う方法には、きめの細かい設定と監視が可能であるという利点があります。そして、もう 1 つ、ドメインディレクトリを手作業で作成して複製サーバーを生成し、nisbackup と nisrestore を使って NIS+ データをロードできるという利点もあります。NIS+ のバックアップおよび復元機能を使うと、nisserver を使うより短時間でデータをロードできます。

新たに構成した複製サーバーに NIS+ データセットをロードする方法には、次の 2 通りがあります。

- 「nisping」。nisserver スクリプトか NIS+ コマンドのどちらかを使って新しい複製サーバーを構成する場合、マスターサーバーは nisping を使用してネットワーク経由で該当する名前空間のデータセットを新しい複製サーバーに自動的にロードします。このとき、大きな名前空間では処理に長時間かかり、その間、名前管理情報の要求が遅れることがあります。詳細は、171 ページの「nisping を使ってデータを複製サーバーにロードする」を参照してください。
- 「バックアップと復元」。nisping によるデータ転送に割り込みをかけ、NIS+ のバックアップ機能と復元機能を使って、名前空間データを新たに構成した複製サーバーにロードできます (169 ページの「nisrestore を使ってデータを複製サーバーにロードする」を参照)。複製サーバーから複製サーバーにデータセットがロードされることになり、マスターサーバーから複製サーバーにネットワーク経由でデータセットをロードする場合に比べて格段に早く終わるので、こちらの方法をお勧めします。

NIS+ コマンドを使って複製サーバーを構成する

この節では、NIS+ コマンドを使って複製サーバーを既存のドメインに追加する方法について説明します。

複製サーバー構成時のセキュリティについて

この作業を実行する NIS+ 主体には、ドメインのディレクトリオブジェクトに対する変更権が必要です。

複製サーバーを構成するための前提条件

- ドメインをあらかじめ構成し、マスターサーバーを稼働させておく
- 新しい複製サーバーが NIS+ サーバーとして構成されている (163 ページの「NIS+ サーバーを設定する」を参照)

必要な情報

- サーバー名
- ドメイン名

NIS+ コマンドを使って複製サーバーを構成する—作業マップ

表 7-1 NIS+ コマンドを使って複製サーバーを構成する

作業	目的	参照先
NIS+ サーバーを設定する	NIS+ コマンドを使って複製サーバーを設定する	168 ページの「NIS+ コマンドを使って複製サーバーを構成する方法」

▼ NIS+ コマンドを使って複製サーバーを構成する方法

この例では、マスターサーバー名を master1、新しい複製サーバー名を replica2 とします。

1. ドメインのマスターサーバーにログインします。
2. NIS+ サービスが実行されていることを確認します。

```
master1# svcs -l network/rpc/nisplus:default
```


3. ドメインに複製サーバーを追加します。

nismkdir コマンドに `-s` オプションを付けて実行します。次の例では、`doc.com.` ドメインに `replica2` という名前の複製サーバーマシンを追加します。

```
master1# nismkdir -s replica2 doc.com.  
master1# nismkdir -s replica2 org_dir.doc.com.  
master1# nismkdir -s replica2 groups_dir.doc.com.
```

すでに存在するディレクトリオブジェクトに `nismkdir` コマンドを実行すると、ディレクトリは再作成されずに、与えられたフラグに基づいてディレクトリが変更されます。この場合、`-s` フラグはドメインに追加する複製サーバーを割り当てます。複製サーバーが追加されたことを確認するには、`niscat -o` コマンドを実行して、ディレクトリオブジェクトの定義を調べます。



注意 – 複製サーバー上で `nismkdir` を実行しないでください。複製サーバーで `nismkdir` を実行すると、マスターサーバーと複製サーバーの間で通信上の問題が発生します。

これで新しい複製サーバーの構成は完了です。次は、構成した複製サーバーに NIS+ データセットをロードします。これには、次の 2 つの方法があります。

- 「`nisping`」。何もしなければ、マスターサーバーによって `nisping` コマンドが実行され、該当する名前空間データが新たに構成された複製サーバーにロードされます。名前空間が大きい場合は、データのロードに時間がかかることがあります。データのロード中は、ネーミング情報の要求は遅延することがあります。詳細は、171 ページの「[nisping を使ってデータを複製サーバーにロードする](#)」を参照してください。
- 「バックアップと復元」。 `nisping` によるデータ転送に割り込みをかけ、NIS+ のバックアップ機能と復元機能を使って、名前空間データを新たに構成した複製サーバーにロードできます (169 ページの「[nisrestore を使ってデータを複製サーバーにロードする](#)」を参照)。ほかの方法に比べて格段に早く効率的なので、こちらの方法をお勧めします。

nisrestore を使ってデータを複製サーバーにロードする

この節では、NIS+ のバックアップ機能と復元機能を使って名前空間データを新しい複製サーバーにロードする方法について説明します。この方法を使ってデータを複製サーバーにロードすることをお勧めします。

nisrestore を使って複製サーバーにデータをロードする場合のセキュリティについて

この作業を実行する NIS+ 主体には、ドメインのディレクトリオブジェクトに対する変更権が必要です。

nisrestore を使ってデータをロードするための前提条件

- ドメインをあらかじめ構成し、マスターサーバーを稼働させておく
- 新しい複製サーバーが NIS+ サーバーとして構成されている (163 ページの「NIS+ サーバーを設定する」を参照)
- 新しい複製サーバーが複製サーバーとして構成されている (168 ページの「NIS+ コマンドを使って複製サーバーを構成する」を参照)

nisrestore を使ってデータを複製サーバーにロードする — 作業マップ

表 7-2 nisrestore を使ってデータを複製サーバーにロードする

作業	目的	参照先
nisrestore を使ってデータを複製サーバーにロードする	nisrestore コマンドを使ってデータを複製サーバーにロードする	170 ページの「名前空間データをロードする方法—nisrestore による方法」

▼ 名前空間データをロードする方法—nisrestore による方法

この例では、マスターサーバー名を master1、新しい複製サーバー名を replica2 とします。

1. 新しい複製サーバー上の NIS+ サービスを停止します。
- この処理は、nisping コマンドを使用した、マスターから複製への名前空間データの自動転送に割り込んで実行されます。

```
replica2# svcadm disable /network/rpc/nisplus:default
```

2. マスターサーバー上で NIS+ バックアップ機能を実行します。
- この手順の詳細については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』を参照してください。以下の例では、nisbackup コマンドを使って master1 サーバーを /var/master1_backup ディレクトリにバックアップします。

```
master1# nisbackup -a /var/master1_backup
```

nisrestore を使って新しい複製サーバーを構成するもっとも簡単な方法は、マスターサーバーのデータを NFS にマウントされた (複製サーバーからアクセス可能な) ディレクトリにバックアップするというものです。この例では、マスターサーバーと新しい複製サーバーの両方に、/var/master1_backup ディレクトリへのアクセス権が与えられているものと想定します。

このほかに、tar コマンドを使って /var/master1_backup ディレクトリからテープカートリッジなどの可搬記憶メディアにデータをコピーし、次に、その可搬記憶メディアから新しい複製サーバーにマウントされているディレクトリにデータをコピーし、そのディレクトリを nisrestore コマンドの情報源として使うという方法 (手順 3 参照) もあります。

3. **nisrestore** コマンドを使って、NIS+ データセットを新しい複製サーバーにロードします。

この手順の詳細については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』を参照してください。以下の例では、nisrestore コマンドを使って NIS+ データを /var/master1_backup ディレクトリから client2 にダウンロードします。

```
replica2# nisrestore -a /var/master1_backup
```

作成している複製サーバーがルートドメイン用の場合、あるいは nisrestore が必要なデータを検証または検索できないというエラーメッセージが表示される場合は、nisrestore に -f オプションを付けて実行してください。たとえば、以下のようになります。

```
replica2# nisrestore -f -a /var/master1_backup
```

4. 新しい複製サーバー上で NIS+ サービスを起動します。

詳細は、165 ページの「NIS+ サーバーを構成する方法」を参照してください。

nisping を使ってデータを複製サーバーにロードする

この節では、nisping コマンドを使って名前空間データを新しい複製サーバーにロードする方法について説明します。通常、このプロセスは自動的に実行されるため、nisping コマンドを実行する必要はまずありません。

nisping コマンドを使う方法の問題点は、マスターサーバーから複製サーバーへデータの再同期をとるために、NIS+ プロトコルを使ったネットワーク上のデータのやりとりが必要だということです。名前空間が大きい場合は、この処理に何時間もかかり、その間、名前管理情報の要求に対する応答が遅れることがあります。

nisping を使ってデータをロードする場合のセキュリティについて

この作業を実行する NIS+ 主体には、ドメインのディレクトリオブジェクトに対する変更権が必要です。

nisping を使ってデータをロードするための前提条件

- ドメインをあらかじめ構成し、マスターサーバーを稼働させておく

- 新しい複製サーバーが NIS+ サーバーとして構成されている (163 ページの「NIS+ サーバーを設定する」を参照)
- 新しい複製サーバーが複製サーバーとして構成されている (168 ページの「NIS+ コマンドを使って複製サーバーを構成する」を参照)

nisping を使ってデータを複製サーバーにロードする — 作業マップ

表 7-3 nisping を使ってデータを複製サーバーにロードする

作業	目的	参照先
nisping を使ってデータを複製サーバーにロードする	nisping を使ってデータを複製サーバーにロードする	172 ページの「名前空間データをロードする方法—nisping による方法」

▼ 名前空間データをロードする方法—nisping による方法

通常、名前空間データのロードは、マスターサーバーによって自動的に開始されます。マスターサーバーによるロードが行われなかった場合は、次の説明に従って nisping コマンドを実行してください。

- ディレクトリに対して **nisping** を実行します。

この手順では、新しい複製サーバーにメッセージ「ping」を送信して、マスターサーバーに対して更新を要求するように通知します。複製サーバーがルートドメインに所属していない場合、必ずドメイン名を指定してください。次の例では、ドメイン名は完全を期すためにだけ記述してあります。この作業で使用する例は、ルートドメインに複製サーバーを追加しているため、次の例にあるドメイン名 doc.com. は必要ありません。

```
master1# nisping doc.com.
master1# nisping org_dir.doc.com.
master1# nisping groups_dir.doc.com.
```

この結果は以下のようになります。

```
master1# nisping doc.com.
Pinging replicas serving directory doc.com. :
Master server is master1.doc.com.
  No last update time
Replica server is replica1.doc.com.
  Last update seen was Wed Nov 18 11:24:32 1992
Pinging ... replica2.doc.com.
```

大きな名前空間の場合、この処理に何時間もかかる場合があります。nisping の詳細については、第 18 章を参照してください。

サーバー構成の要覧

表 7-4、表 7-5 では、この章で説明した作業のまとめを示しています。この 2 つの表は、もっとも簡単な場合を想定しているため、参考用として使用するには、実際の自分の作業の詳細を理解している必要があります。また、ここでは、各コマンドに対するサーバーの応答を示していません。

表 7-4 複製サーバー replica2 を doc.com. に追加する - コマンドのまとめ

作業	コマンド
ドメインマスターサーバーにスーパーユーザーとしてログインする	master1% su
新しい複製サーバーを指定する	# nismkdir -s replica2 doc.com. # nismkdir -s replica2 org_dir.doc.com. # nismkdir -s replica2 groups_dir.doc.com.
複製サーバーに対して nisping を実行する	# /usr/lib/nis/nisping doc.com. # /usr/lib/nis/nisping org_dir.doc.com. # /usr/lib/nis/nisping groups_dir.doc.com.

注 - 上記の例で説明したように、新しい複製サーバーにデータをロードする場合は、nisping を使用するより NIS+ のバックアップと復元機能を使用した方が簡単です。詳細については、169 ページの「[nisrestore を使ってデータを複製サーバーにロードする](#)」を参照してください。

表 7-5 ルート以外のマスターサーバーを起動する - コマンドのまとめ

作業	コマンド
スーパーユーザーとしてサーバーにログインする	server% su
NIS 互換モードの場合のみ -y オプションを使用してデーモンを起動する (DNS 転送が必要な場合は -B オプションも使用する)	/lib/svc/method/nisplus ファイルを編集して必要なオプションを追加してから、次のようにサービスを再起動する # svcadm restart network/rpc/nisplus
NIS+ の場合のみ デーモンを起動する	# svcadm enable network/rpc/nisplus

第 8 章

ルート以外のドメインの構成

この章では、NIS+ コマンドセットを使ってサブドメイン (ルート以外のドメイン) を構成する方法 (マスターサーバーと複製サーバーを設定する方法を含む) を、手順を追って説明します。

注 - NIS+ は、将来のリリースではサポートされなくなる可能性があります。NIS+ から LDAP への移行支援ツールは、Solaris 9 リリース以降で使用できます (『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』を参照)。詳細については、<http://www.sun.com/directory/nisplus/transition.html> を参照してください。

ルート以外のドメインを設定する

注 - NIS+ クライアントを設定する作業は、この章で説明する NIS+ のコマンドセットを使用する方法よりも、パート I で説明した NIS+ 設定スクリプトを使用する方が簡単です。この章で説明する方法は、NIS+ に精通した管理者や、設定スクリプトでは提供されない標準以外の機能や構成を必要とする管理者だけが使用してください。

最初にルート以外のドメインのサーバーを構成してから、ルート以外のドメインを構成してください。

ルート以外のドメインを設定するには、次の作業を行います。

- ドメインのセキュリティの設定
- ドメインのディレクトリの作成

- ドメインのテーブルの作成
- ドメインのサーバーの指定

ルートドメインの構成と同様に、これらの作業は連続して実行できません。構成プロセスを簡単にするため、これらを個々の手順に分割して、もっとも効率的な順序に並べています。

標準構成と NIS 互換構成の相違 (ルート以外のドメイン)

サブドメインにおける NIS 互換の NIS+ サーバーと標準の NIS+ サーバーとの違いは、ルートドメインの場合と同じです (132 ページの「標準構成と NIS 互換構成の手順の相違」参照)。

NIS 互換ドメインの各サーバーの NIS+ デーモンは、第 7 章の説明に従って、`-y` オプションを使用して起動する必要があります。また、NIS 互換ドメインでは、ドメインのテーブルによって未認証クラスに読み取り権を提供する必要があります。これにより、NIS クライアントはテーブルに格納されている情報にアクセスできます。手順 4 で説明するとおり、`nissetup` コマンドに `-y` オプションを追加すると、テーブル内の情報にアクセスできます。標準の NIS+ ドメインでも同じコマンドを使用しますが、`-y` オプションは使用しません。これについても手順 4 で説明します。

設定作業の手順を次にまとめます。

1. ドメインのマスターサーバーにログインします。
2. ドメインの管理グループを指定します。
3. ドメインのディレクトリを作成し、そのサーバーを指定する
4. ドメインのサブディレクトリとテーブルを作成します。
5. ドメインの管理グループを作成します。
6. ディレクトリオブジェクトに完全なグループアクセス権を割り当てます。
7. ドメインの管理グループにサーバーを追加します。
8. ほかの管理者の資格を追加します。
9. ドメインの管理グループに管理者を追加します。

ルート以外のドメイン設定時のセキュリティについて

注 - NIS+ のセキュリティシステムは複雑です。NIS+ セキュリティを使い慣れていない場合は、第 17 章を参照してから NIS+ 環境を構成することをお勧めします。

多くのサイトでは、親ドメインのセキュリティを確保するために、その下にドメインを作成できるのは、親ドメインのマスターサーバー、または親ドメインの管理グループに所属するシステム管理者に限定しています。これは、管理方針であり NIS+ の必

要条件ではありませんが、この章の操作説明ではこの作業を行う管理者がこの方針に従っているものと仮定します。もちろん、親ドメインの管理グループには、親ディレクトリオブジェクトに対する作成権が必要です。これを確認するには、`niscat -o` コマンドを実行します。

```
rootmaster# niscat -o doc.com.
Object Name : Doc
Owner : rootmaster
Group : admin.doc.com.
Domain : Com.
Access Rights : r---rmcdrmcdr---
:
```

安全性よりも便宜性を重視する場合、新しいドメインのマスターサーバーをその親ドメインの管理グループのメンバーとし、そのサーバーからすべて手順を実行できます。この場合、第 17 章で説明する `nisgrpadm` コマンドを使用します。

ルート以外のドメインを設定するための前提条件

- 親ドメインを構成し、実行していなければなりません。
- このドメインのマスターとして指定されるサーバーは、すでに初期設定され、NIS+ を実行していなければなりません。
- 複製サーバーを指定する場合、マスターサーバーは、その `/etc/hosts` ファイル、`/etc/inet/ipnodes` ファイルまたはその NIS+ `hosts` テーブルを通じて、複製サーバーの IP アドレスを入手できなければなりません。

必要な情報

- 新しいドメインの名前 (手順 3)
- 新しいドメインのマスターサーバーと複製サーバー名
- 新しいドメインの管理グループ名 (手順 2)
- 新しいドメインの管理グループに所属する管理者のユーザー ID (UID) (手順 8)

ルート以外のドメインを設定する — 作業マップ

表 8-1 ルート以外のドメインを設定する

作業	目的	参照先
ルート以外のドメインを設定する	NIS+ コマンドを使ってルート以外のドメインを設定する	177 ページの「ルート以外のドメインの設定方法」

▼ ルート以外のドメインの設定方法

1. ドメインのマスターサーバーにログインします。

新しいドメインのマスターにする予定のサーバーにログインします。この作業の手順では `smaster` という名前のサーバーを使用します。このサーバーは `doc.com.` ドメインに所属し、`sales.doc.com.` サブドメインのマスターサーバーになります。この作業を実行する管理者は、`admin.doc.com.` グループのメンバーである `nisboss.doc.com.` です。このグループには、`doc.com.` ディレクトリオブジェクトに対する完全なアクセス権が割り当てられています。

2. ドメインの管理グループを指定します。

実際に管理グループを作成するのは、[手順 5](#) の時点ですが、ここで管理グループを指定する必要があります。これによって、次の手順で使用される `nismkdir` コマンドは、このグループに対する適切なアクセス権をもつディレクトリオブジェクトを作成できます。またこれは、[手順 4](#) で使用する `nissetup` ユーティリティに対しても同じことを行います。

環境変数 `NIS_GROUP` に、ドメインの管理グループ名を設定します。ここでは、`C` シェルユーザーの場合と `Bourne` シェルまたは `Korn` シェルユーザーの場合の 2 つの例を示します。どちらの場合も `NIS_GROUP` を `admin.sales.doc.com.` に設定します。

`C` シェルの場合

```
smaster# setenv NIS_GROUP admin.sales.doc.com.
```

`Bourne` シェルまたは `Korn` シェルの場合

```
smaster# NIS_GROUP=admin.sales.doc.com.
```

```
smaster# export NIS_GROUP
```

3. ドメインのディレクトリを作成し、そのサーバーを指定する

`nismkdir` コマンドは、新しいドメインのディレクトリ作成と、そのサポートサーバーの指定を 1 つの手順で行います。この構文を次に示します。

```
nismkdir -m master -s replica domain
```

`-m` フラグはマスターサーバーを指定し、`-s` フラグは複製サーバーを指定します。この例を次に示します。

```
smaster# nismkdir -m smaster -s salesreplica sales.doc.com.
```



注意 - `nismkdir` は必ず (複製サーバーではなく) マスターサーバー上で実行してください。複製サーバー上で `nismkdir` を実行しないでください。複製サーバーで実行すると、マスターサーバーと複製サーバーの間で通信上の問題が発生します。

ディレクトリオブジェクトは `/var/nis` にロードされます。内容を表示するには、`niscat -o` コマンドを実行します。`cat` または `more` は使用しないでください。

```
smaster# niscat -o sales.doc.com.
```

```
Object Name : Sales
```

```
Owner : nisboss.doc.com.
```

```
Group : admin.sales.doc.com.
```

```
Domain : doc.com.
```

```
Access Rights : ----rmcdr---r---
```

```
.
```

ルートディレクトリとは異なり、このディレクトリオブジェクトには適切なグループが割り当てられています。したがって、`nischgrp` を実行する必要はありません。

4. ドメインのサブディレクトリとテーブルを作成します。

この手順では、`org_dir` ディレクトリと `groups_dir` ディレクトリ、および NIS+ テーブルを新しいディレクトリオブジェクトの下に追加します。`nissetup` ユーティリティを使用しますが、新しいドメイン名の追加を忘れないでください。NIS 互換ドメインの場合、`-Y` フラグを指定します。

「NIS 互換の場合」

```
smaster# /usr/lib/nis/nissetup -Y sales.doc.com.
```

「標準 NIS+ の場合」

```
smaster# /usr/lib/nis/nissetup sales.doc.com.
```

このユーティリティによって追加されたオブジェクトを表示すると、次のようになります。

```
smaster# /usr/lib/nis/nissetup
org_dir.sales.doc.com. created
groups_dir.sales.doc.com. created
auto_master.org_dir.sales.doc.com. created
auto_home.org_dir.sales.doc.com. created
bootparams.org_dir.sales.doc.com. created
cred.org_dir.sales.doc.com. created
ethers.org_dir.sales.doc.com. created
group.org_dir.sales.doc.com. created
hosts.org_dir.sales.doc.com. created
mail_aliases.org_dir.sales.doc.com. created
sendmailvars.org_dir.sales.doc.com. created
netmasks.org_dir.sales.doc.com. created
netgroup.org_dir.sales.doc.com. created
networks.org_dir.sales.doc.com. created
passwd.org_dir.sales.doc.com. created
protocols.org_dir.sales.doc.com. created
rpc.org_dir.sales.doc.com. created
services.org_dir.sales.doc.com. created
timezone.org_dir.sales.doc.com. created
```

`-Y` オプションによって、標準の NIS+ ドメインの場合と同じテーブルとサブディレクトリが作成されますが、NIS クライアントからの要求が NIS+ テーブル内の情報にアクセスできるよう、未認証クラスに読み取り権が割り当てられます。

`/var/nis/salesmaster` に相当する自分のマスターを調べることによって、`org_dir` ディレクトリと `groups_dir` ディレクトリが存在することを確認できます。これらのディレクトリは、ルートオブジェクトおよびその他の NIS+ ファイルとともに登録されています。テーブルは `org_dir` ディレクトリに存在します。任意のテーブルの内容を調べるには、[第9章](#)で説明する `niscat` コマンドを実行します。ただしこの時点ではテーブルは空です。

5. ドメインの管理グループを作成します。

この手順では、[手順 2](#) で指定した管理グループを作成します。nisgrpadm コマンドに -c オプションを付けて実行してください。次の例では admin.sales.doc.com. グループを作成します。

```
smaster# nisgrpadm -c admin.sales.doc.com.  
Group admin.sales.doc.com. created.
```

この手順ではグループを作成するだけであり、そのメンバー名の指定は行いません。指定は[手順 9](#)で行います。

6. ディレクトリオブジェクトに完全なグループアクセス権を割り当てます。

デフォルトでは、ディレクトリオブジェクトはそのグループに読み取り権を与えるだけであり、これではその他のカテゴリと同様、グループも使うことができません。クライアントとサブドメインの構成を簡単にするため、ディレクトリオブジェクトがそのグループに与えるアクセス権を、読み取り権のみから読み取り権、変更権、作成権、削除権に変更します。次に示すように、nischmod コマンドを実行します。

```
smaster# nischmod g+rmcd sales.doc.com.
```

nischmod コマンドの使用方法の詳細については、[第 15 章](#)を参照してください。

7. ドメインの管理グループにサーバーを追加します。

この時点で、このドメインのグループにはメンバーがありません。-a オプションを付けて nisgrpadm コマンドを実行し、マスターサーバーと複製サーバーを追加します。最初の引数はグループ名であり、残りの引数は新しいメンバーの名前です。この例では、smaster.doc.com. と salesreplica.doc.com. を admin.sales.doc.com. グループに追加します。

```
smaster# nisgrpadm -a admin.sales.doc.com. smaster.doc.com.  
salesreplica.doc.com.  
Added smaster.doc.com. to group admin.sales.doc.com.  
Added salesreplica.doc.com. to group admin.sales.doc.com.
```

このサーバーが管理グループに属していることを確認するには、nisgrpadm コマンドを -l オプションで実行します。手順については、[第 17 章](#)を参照してください。

```
smaster# nisgrpadm -l admin.sales.doc.com.  
Group entry for admin.sales.doc.com. group:  
Explicit members:  
smaster.doc.com.  
salesreplica.doc.com.  
No implicit members  
No recursive members  
No explicit nonmembers  
No implicit nonmembers  
No recursive nonmembers
```

8. ほかの管理者の資格を追加します。

そのドメインで仕事をするほかの管理者の資格を追加します。

すでにもう 1 つのドメインで DES 資格をもつ管理者の場合、単に LOCAL 資格を追加します。このとき、-p フラグと -P フラグ付きの nisaddcred コマンドを実

行します。たとえば、次のようになります。

```
smaster# nisaddcred -p 33355 -P nisboss.doc.com. local
```

まだ資格をもたない管理者の場合、2つの方法があります。

- 管理者に対して、自分の資格を追加するよう要求するのが1つの方法です。この方法は、スーパーユーザーとして実行する必要があります。ユーザー ID が 22244 で、主体名が `juan.sales.doc.com.` の管理者が、`sales.doc.com.` ドメインに自分の資格を追加する例を次に示します。

```
smaster# nisaddcred -p 22244 -P juan.sales.doc.com. local
smaster# nisaddcred -p unix.22244@sales.doc.com -P \
juan.sales.doc.com. des
Adding key pair for unix.22244@sales.doc.com.
Enter login password:
```

- もう1つの方法では、ダミーパスワードを使用して、ほかの管理者用の一時的な資格を作成します。各管理者には NIS+ passwd テーブル内にエントリが必要です。

```
smaster# nisaddcred -p 22244 -P juan.sales.doc.com. local
smaster# nisaddcred -p unix.22244@sales.doc.com -P \
juan.sales.doc.com. des
Adding key pair for unix.22244@sales.doc.com.
Enter juan's login password:
nisaddcred: WARNING: password differs from login passwd.
Retype password:
```

各管理者は、後で `chkey` コマンドを実行して、自分のネットワークパスワードを変更できます。パスワードの変更方法については、[第12章](#)と[第13章](#)を参照してください。

注 - 上記の[手順8](#)の2つの例で、小文字の `-p` フラグに続くドメイン名の終わりにドットを付けないでください。また、大文字の `-P` フラグに続くドメイン名の終わりにはドットを必ず付けてください。

9. ドメインの管理グループに管理者を追加します。

この手順を実行するには、ほかのシステム管理者がダミーパスワードを変更するまで待つ必要はありません。`-a` オプションを付けて `nisgrpadm` コマンドを実行します。最初の引数はグループ名、残りの引数は管理者名です。この例では、管理者 `juan` を `admin.sales.doc.com.` グループに追加します。

```
smaster# nisgrpadm -a admin.sales.doc.com. juan.sales.doc.com.
Added juan.sales.doc.com. to group admin.sales.doc.com.
```

10. NIS+ テーブルを格納するための、十分なスワップ空間を割り当てます。

スワップ空間は、`rpc.nisd` の最大サイズの2倍にする必要があります。`rpc.nisd` が使用するメモリー量を調べるには、次のコマンドを実行してください。

```
rootmaster# /usr/lib/nis/nisstat
```

rpc.nisd は、特定の条件のもとでは、自らのコピーを作成してフォークします。メモリーが不足すると、rpc.nisd は正しく動作しません。

また、NIS+ テーブルに必要なメモリーとスワップ空間のサイズも計算できます。たとえば、NIS+ テーブル内に、180,000 人のユーザーと 180,000 台のホストがある場合、これらの 2 つのテーブルが占有するメモリーは、約 190M バイトです。180,000 人のユーザーと 180,000 台のホストに資格を追加する場合、cred テーブルには、540,000 のエントリ (ユーザーごとにローカルの資格と DES の資格、合わせて 2 つの資格、ホストごとに 1 つの資格) が入ります。そのため、cred テーブルが占有するメモリーは、約 285M バイトになります。この例では、rpc.nisd に必要なメモリー容量は、少なくとも、190M バイト + 285M バイト = 475M バイトになります。この結果、少なくとも 1G バイトのスワップ空間が必要になります。また、rpc.nisd 全体をすべてメモリー内に保持するには、少なくとも 500M バイトが必要です。

サブドメイン構成の要覧

表 8-2 は、サブドメインの構成に必要な手順のまとめです。これはもっとも簡単なケースを想定しているため、このまとめを参考用として使用するには、その前に自分の実際の作業の詳細を理解することが必要です。また、ここでは、各コマンドに対するサーバーの応答を示していません。

表 8-2 サブドメインを設定する - コマンドのまとめ

作業	コマンド
ドメインマスターサーバーにスーパーユーザーとしてログインする	smaster% su
ドメインの管理グループを指定する	# NIS_GROUP=admin.sales.doc.com. # export NIS_GROUP
ドメインのディレクトリを作成し、そのサーバーを指定する	# nismkdir -m smaster -s salesreplica sales.doc.com.
org_dir、groups_dirおよびテーブルを作成する。NIS 互換の場合、-y を使用する	# /usr/lib/nis/nissetup sales.doc.com.
管理グループを作成する	# nisgrpadm -c admin.sales.doc.com.
ドメインのディレクトリに対して、完全なグループ権を割り当てる	# nischmod g+rmcd sales.doc.com.

表 8-2 サブドメインを設定する - コマンドのまとめ (続き)

作業	コマンド
管理グループにサーバーを追加する	# nisgrpadm -a admin.sales.doc.com. smaster.doc.com. sreplica.doc.com.
ほかのシステム管理者の資格を追加する	# nisaddcred -p 22244 -P juan.sales.doc.com. local # nisaddcred -p unix.22244@sales.doc.com. juan.sales.doc.com. DES
ドメインの管理グループに管理者を追加する	# nisgrpadm -a admin.sales.doc.com. juan.sales.doc.com.

第 9 章

NIS+ テーブルの設定

この章では、NIS+ コマンドセットを使用して、/etc ディレクトリ内のファイルや NIS マップからマスターサーバー上に NIS+ テーブルを作成する方法について説明します。また、この章では NIS+ テーブルから NIS マップへ情報を戻す方法と、passwd テーブルのパスワード列へのアクセスを制限する方法を説明します。

注 - NIS+ は、将来のリリースではサポートされなくなる可能性があります。NIS+ から LDAP への移行支援ツールは、Solaris 9 リリース以降で使用できます (『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』を参照)。詳細については、<http://www.sun.com/directory/nisplus/transition.html> を参照してください。

テーブルの設定

注 - NIS+ テーブルを生成する作業は、この章で説明する NIS+ コマンドセットを使用する方法よりも、パート I で説明した NIS+ 設定スクリプトを使用した場合の方が簡単です。この章で説明する方法は、NIS+ に精通した管理者や、設定スクリプトでは提供されない標準以外の機能や構成を必要とする管理者だけが使用してください。さらに、Solaris 管理コンソール ツールをお持ちの場合は、NIS+ テーブルに関連する作業が簡単になります。

NIS+ テーブルを生成するには次の 4 種類の方法があります。

- ファイルから作成する (187 ページの「NIS+ テーブルをファイルから生成する」参照)

- NIS マップから作成する (192 ページの「NIS+ テーブルを NIS マップから生成する」参照)
- nispopulate スクリプトを使用して作成する (100 ページの「NIS+ テーブルの生成 (populate)」および 121 ページの「新しいサブドメインのテーブルの生成」参照)
- 可能な場合、Solaris 管理コンソール ツールを使用する

マップまたはファイルからテーブルを生成する場合、第 5 章および第 8 章で説明する手順に従って、ルートまたはサブドメインの設定作業の中で、あらかじめテーブルを作成しておく必要があります。ドメインテーブルは、テーブルの作成後いつでも生成できますが、ドメインの設定が完了したらすぐに生成しておくことをお勧めします。テーブルを生成すると、クライアントに関する必要な情報がすでにドメインテーブル内に格納されることになるため、クライアントの追加が簡単にできます。

テーブルの生成の方法

ファイルまたは NIS マップからテーブルを生成する場合、次の 3 つの方法を使用できます。

- 「置換」 - この方法を使用すると、NIS+ は、初めにテーブル内の既存のエントリをすべて削除してから、情報源からのエントリを追加します。サイズの大きいテーブルの場合は、マスターサーバーの `/var/nis/trans.log` ファイルにエントリの大きなセット (既存のエントリを削除するためのセットと新しいエントリを追加するためのセット) が追加されます。`/var/nis` 内のディスク空間を大幅に使用するため、複製サーバーへの転送時間はさらに長くなります。
- 「追加」 - この方法は、NIS+ テーブルに情報源からのエントリを追加します。既存のエントリが影響されることはありません。
- 「マージ」 - この方法は、置換オプションと結果は同じですが、内部的な処理が異なります。既存のエントリを削除後、置換するのではなく、更新します。このオプションでは、NIS+ は次の 3 種類のエントリを異なる方法で処理します。
 - ソース内にだけ存在するエントリは、テーブルに追加する
 - ソースとテーブルの両方に存在するエントリは、テーブル内で更新する
 - NIS+ テーブルにだけ存在するエントリは、テーブルから削除する

大きなテーブルを更新する場合で、内容の変更があまりないときは、マージオプションを使用するとサーバーは多くの動作を節約できます。ソース内で重複していないエントリだけを削除する (置換オプションは全エントリを無差別に削除する) ため、重複エントリごとに 1 回の削除と 1 回の追加動作を省略できます。このため、この方法をお勧めします。

NIS+ テーブルをファイルから生成する

これは `/etc/hosts` などの ASCII ファイルの内容を NIS+ テーブルに転送する作業です。

手順を次に示します。

1. データを転送する各ファイルの内容を確認します。
2. 各ファイルのコピーを作成します。作成したコピーを実際の転送に使用します。このマニュアルでは、`hosts.xfr` のように、転送されるファイルのコピーの最後に `.xfr` を付けます。
3. NIS+ クライアントにログインするテーブルを更新するための資格とアクセス権が必要です。以下の [187 ページの「ファイルからテーブルを生成する場合のセキュリティについて」](#) を参照してください。
4. このシェルのコマンド検索パスに `/usr/lib/nis` を追加します。
5. `nisaddent` を使用して、次の必要なファイルを 1 つずつ転送します。`aliases`, `bootparams`, `ethers`, `group`, `hosts`, `netgroup`, `netmasks`, `networks`, `passwd`, `protocols`, `rpc`, `services`, `shadow`、および `ipnodes`。

注 - 新しい `/etc/inet/ipnodes` ファイルには、IPv4 および IPv6 のアドレスが含まれています。`nisaddent` を実行して、`/etc/inet/ipnodes` ファイルを `ipnodes.org_dir` テーブルに変換してください。

6. `publickey` ファイルを転送する
7. オートマウント情報を転送します。
8. 複製サーバーに対して `nisping` を実行します。
9. テーブルに対しチェックポイントを実行します。

ファイルからテーブルを生成する場合のセキュリティについて

NIS+ テーブルは、NIS+ クライアントまたは NIS+ ルートマスターサーバーから生成できます。NIS+ テーブルを生成するには、スーパーユーザー (`root`) としてログインする必要はありませんが、一定の資格とアクセス権が必要です。テーブル内のエントリをテキストファイルのエントリによって置換またはマージする場合、そのテーブルへの作成権と削除権が必要です。新しいエントリを追加する場合、作成権だけが必要です。

注 - NIS+ のセキュリティシステムは複雑です。NIS+ セキュリティを使い慣れていない場合は、[第 11 章](#)を参照してから NIS+ 環境を構成することをお勧めします。

この処理が完了した後、テーブルエントリは、動作を実行した NIS+ 主体と、環境変数 NIS_GROUP によって指定されたグループによって所有されます。

ファイルからテーブルを生成するための前提条件

- ドメインをあらかじめ設定していて、そのマスターサーバーを実行している
- ドメインのサーバーには、新しいテーブル情報を収容できるだけの十分なスワップ領域が必要
- ファイル内の情報は、ロード先のテーブルに合った書式で書かれていなければならない。対応する NIS+ テーブルに転送するテキストファイルで要求される書式については、[100 ページの「nispopulate を実行してルートサーバーテーブルを生成するための前提条件」](#)を参照してください。ローカルの /etc 内のファイルは、正しい書式で書かれているのが普通ですが、いくつかのコメントを削除しなければならないこともあります。
- マシン名とユーザー名は重複していない。ユーザーとマシンは、すべて固有の名前を付ける必要があります。また、ユーザーと同じ名前の付いたマシンは使用できません。
- マシン名にはドット (ピリオド) および下線は使用できない。たとえば、マシン名に sales.alpha を使用できません。ドットや下線の代わりにハイフンを使うことはできます。たとえば、sales-alpha というマシン名は有効です。

必要な情報

転送されるテキストファイルの名前と位置が必要です。

NIS+ テーブルをファイルから生成する — 作業マップ

表 9-1 NIS+ テーブルをファイルから生成する

作業	目的	参照先
NIS+ テーブルをファイルから生成する	NIS+ テーブルをファイルから生成する	188 ページの「NIS+ テーブルをファイルから生成する方法」

▼ NIS+ テーブルをファイルから生成する方法

1. データを転送する各ファイルをチェックします。

不正なエントリがないことを確認します。正しいデータが、所定の場所に正しい書式で記録されていることを確認します。エントリのうち、古いもの、無効なもの、破損しているものは削除します。また、不完全なエントリや一部のみのエントリも削除します。不完全なエントリは、設定を完了した後追加する方が、不完全なエントリや破損したエントリを転送するよりも簡単です。

2. 転送する各ファイルの作業用コピーを作成します。

このセクションで説明する実際のファイル転送の手順では、作成した作業用コピーを使用します。各作業用コピーには、同じファイル名拡張子を付けます (たとえば、`.xfr`)。

```
rootmaster% cp /etc/hosts /etc/hosts.xfr
```

万が一に備えて、使用する予定のすべてのファイルを `/etc` 以外のディレクトリにコピーしておくのもよいでしょう。`nisaddent` コマンドと `nispopulate` コマンドでは、ファイルソースディレクトリを指定できます。

3. NIS+ クライアントにログインする

この作業はどの NIS+ クライアントからでも実行できます。ただし、そのクライアントは、情報の転送先となるテーブルと同じドメインに所属していなければなりません。ここで示す例では、ルートマスターサーバーを使用します。これらの例では、システム管理者はスーパーユーザーとしてログインしているため、この操作を実際に実行している (適切な資格とアクセス権を必要とする) NIS+ 主体は、ルートマスターサーバーです。

しかし、NIS+ テーブルを更新するだけであれば、あえてルートマスターサーバーにスーパーユーザー (`root`) としてログインする必要はありません。スーパーユーザーであれ、一般ユーザーであれ、一定の資格、権限、許可さえあれば、どのマシンからでも NIS+ テーブルを更新できます。

4. このシェルのコマンド検索パスに `/usr/lib/nis` を追加します。

テーブルごとに `/usr/lib/nis/nisaddent` コマンドを使用するため、検索パスに `/usr/lib/nis` を追加しておく、毎回これを入力する必要はありません。ここでは、C シェルユーザーの場合の例と Bourne シェルまたは Korn シェルのユーザーの場合の例を示します。

C シェルの場合

```
rootmaster# setenv PATH $PATH:/usr/lib/nis
```

Bourne シェルまたは Korn シェルの場合

```
rootmaster# PATH=$PATH:/usr/lib/nis
rootmaster# export PATH
```

5. `nisaddent` を使用して、次のファイルを 1 つずつ転送します。

```
aliases
bootparams
ethers
group
hosts
ipnodes
netgroup
```

```
netmasks
networks
protocols
rpc, services
```

publickey、automounter、passwd、および shadow の各ファイルは、実行する手順がそれぞれ少し異なります。publickey ファイルの場合は、[手順 6](#)に進んでください。automounter ファイルの場合は、[手順 7](#)に進んでください。passwd および shadow ファイルの場合は、[手順 8](#)に進んでください。

デフォルトでは、nisaddent はファイル情報をテーブル情報に追加します。置換またはマージを行うには、`-r` または `-m` オプションを使用します。

置換する場合、次のように入力します。

```
rootmaster# nisaddent -r -f filename table [domain]
```

追加する場合、次のように入力します。

```
rootmaster# nisaddent -a -f filename table [domain]
```

マージする場合、次のように入力します。

```
rootmaster# nisaddent -m -f filename table [domain]
```

はじめてテーブルを生成する場合は、デフォルトの `-a` オプションが最適です。NIS+ テーブルを NIS マップまたは `/etc` 内のファイルと同期させるための最適なオプションは `-m` (マージ) オプションです。

- *filename* はファイル名です。通常は、nisaddent によって作成されたことを示す `.xfr` をファイル名の最後につけます。
- *table* は NIS+ テーブル名です。*domain* 引数は省略可能で、テーブルを別のドメインに生成するときに使用します。ルートドメインのマスタースerverから入力されたいくつかの例を次に示します。これらのファイルは、`/etc` 内のファイルを編集したものです。

```
rootmaster# nisaddent -m -f /etc/hosts.xfr hosts
rootmaster# nisaddent -m -f /etc/groups.xfr groups
```

この作業をルート以外のサーバーから実行する場合、ルート以外のサーバーは、サポートするドメインの上のドメインに所属することに注意してください。つまり、ルート以外のサーバーはもう 1 つのドメインのクライアントです。たとえば、`sales.doc.com`。マスタースerverは `doc.com`。ドメインに所属します。そのマスタースerverからテーブルを `sales.doc.com`。ドメインに生成するには、nisaddent 文に `sales.doc.com`。ドメイン名を追加する必要があります。

```
salesmaster# nisaddent -f /etc/hosts.xfr hosts Sales.doc.com.
```

この作業を `sales.doc.com`。ドメインのクライアントとして実行した場合、このコマンドにドメイン名を指定する必要はありません。

エントリが NIS+ テーブルに転送されたことを確認するには、niscat コマンドを使用します。詳細については、[第 19 章](#)を参照してください。

```
rootmaster# niscat groups.org_dir
root::0:root
other::1::
bin::2:root,bin,daemon
.
```

障害追跡のヒント:niscat を実行しても表示された内容が反映されていない場合、変更がマスターサーバーから複製サーバーに送られていないためかもしれません。そのような場合は、5～10 分後にもう一度 niscat の実行を試みるか、niscat に -M オプションを付けて実行してください。-M オプションを付けると、マスターサーバーのデータが取得されます。

6. publickey ファイルを転送します。

ドメインの cred テーブルには、すでにいくつかの資格が格納されているため、この cred テーブルに転送する publickey テキストファイルの内容によって、これらの資格が上書きされないよう確認する必要があります。これを避けるには、publickey テキストファイルからこれらの資格を取り除きます。rootmaster で次のような行がある場合は、すべて削除してください。

```
unix.rootmaster@doc.com public-key:private-key [alg-type]
```

こうすれば、publickey ファイルの内容を cred テーブルに転送できます。nisaddent に -a (追加) オプションを付けて実行します。

```
rootmaster# nisaddent -a -f /etc/publickey.xfr -t cred.org_dir publickey [domain]
```

ただし、この操作は DES 資格を cred テーブルに転送するだけです。主体は、cred テーブルに対する自分の LOCAL 資格を作成する必要があります。

7. オートマウント情報を転送します。

nissetup ユーティリティは auto_master テーブルと auto_home テーブルを作成しますが、これらは標準の NIS+ テーブルとはみなされません。したがって、これらのテーブルに情報を転送するには、少し異なる構文が必要となります。特に、-t フラグを使用し、そのテーブルが key-value 形式であることを指定しなければなりません。

```
rootmaster# nisaddent -f auto.master.xfr -t auto_master.org_dir key-value
```

```
rootmaster# nisaddent -f auto.home.xfr -t auto_home.org_dir key-value
```

8. NIS+ passwd テーブルを作成します。

NIS+ の passwd テーブルは、/etc/passwd および /etc/shadow の両方のディレクトリのファイルから引き出されるデータで構成されます。このため、nisaddent を 2 回実行して passwd テーブルを作成しなければなりません。passwd テーブルを対象として /etc/passwd ファイルからデータを引き出す場合と、shadow テーブルを対象として /etc/shadow ファイルからデータを引き出す場合の 2 回実行します。shadow ファイルに対して nisaddent を実行する場合、shadow テーブルが存在しなくても、ターゲットテーブルに shadow を指定すること、そしてデータが実際に passwd テーブルの shadow 列に配置されることに注意してください。

```
rootmaster# nisaddent -m -f /etc/passwd.xfr passwd
```

```
rootmaster# nisaddent -m -f /etc/shadow.xfr shadow
```

9. nisping を実行して更新内容を複製サーバーに送ります。

nisping を実行すると、複製サーバーに変更が反映されます。

```
master1# nisping domain
```

```
master1# nisping org_dir.domaincom.
```

```
master1# nisping groups_dir.domain
```

10. テーブルに対しチェックポイントを実行します。

ここまでの手順で、マスターサーバーと複製サーバーの NIS+ データセットがメモリー内で更新されました。今度はそれをディスク上のテーブルファイルに書き込みます。この作業を「チェックポイント」といいます。チェックポイントは必ずしもここで実行しなければならないわけではありません。定期的に行えば、問題ありません。ただし、重要な変更を加えた場合には、できるだけ早めにディスクに書き込むことをお勧めします。

この手順により、ドメインをサポートしている全サーバーが、それらの .log ファイルからディスク上のテーブルのコピーに新しい情報を転送します。ルートドメインを設定したばかりの場合、そのルートドメインにはまだ複製サーバーがないため、この手順はルートマスターサーバーだけが対象となります。チェックポイントを実行するには nisping コマンドに -C (大文字) オプションを付けて実行します。

```
rootmaster# nisping -C org_dir
Checkpointing replicas serving directory org_dir.doc.com. :
Master server is rootmaster.doc.com.
Last update occurred at July 14, 1997
Master server is rootmaster.doc.com.
checkpoint succeeded.
```

スワップ空間が不足している場合、サーバーはチェックポイントを正常に実行できませんが、そのことを通知してくれません。スワップ空間が十分あることを確認する方法として、niscat コマンドを使ってテーブルの内容をリスト表示する方法があります。スワップ空間が不足している場合、次のエラーメッセージが表示されます。

```
can't list table: Server busy, Try Again.
```

このメッセージ内容からはわかりませんが、これはスワップ空間の不足を示しています。スワップ空間を増やし、このドメインに再びチェックポイントを実行します。

NIS+ テーブルを NIS マップから生成する

ここでは、NIS マップの内容を NIS+ テーブルに転送します。手順を次に示します。

1. データを転送する各 NIS マップの内容を確認します。
2. NIS+ クライアントにログインする
3. このシェルのコマンド検索パスに /usr/lib/nis を追加します。
4. nisaddent を使用して、次のマップを1度に1つずつ転送します。aliases, bootparams, ethers, group, hosts, netgroup, netmasks, networks, passwd, protocols, rpc, services。

5. publickey マップを転送します。
6. オートマウント情報を転送します。
7. nisping を実行して変更内容を複製サーバーに反映させます。
8. テーブルに対しチェックポイントを実行します。

NIS マップからテーブルを生成する際のセキュリティについて

この作業を行うシステム管理者 (またはクライアント上のスーパーユーザー) に適切な資格とアクセス権がある限り、この作業は、どの NIS+ クライアントからでも実行できます。テーブル内のエントリを NIS マップのエントリで置換またはマージする場合、そのテーブルへの作成権と削除権が必要です。新しいエントリを追加する場合、作成権だけが必要です。

この作業が完了した後、テーブルエントリは、作業を実行した NIS+ 主体 (システム管理者である自分、またはスーパーユーザーとしてログインした場合はそのクライアント) と、環境変数 NIS_GROUP によって指定されたグループによって所有されます。

NIS マップからテーブルを生成するための前提条件

- ドメインをあらかじめ設定していて、そのマスターサーバーを実行している
- NIS+ テーブルにロードしようとしている NIS マップ用の dbm ファイル (.pag および .dir ファイル) は、/var/yp のサブディレクトリ内にある
- マシン名とユーザー名は重複していない。ユーザーとマシンは、すべて固有の名前を付ける必要があります。また、ユーザーと同じ名前の付いたマシンは使用できません。
- マシン名にはドット (ピリオド) を使用できない。たとえば、マシン名に sales.alpha を使用できません。sales-alpha というマシン名は、使用できます。

必要な情報

NIS マップの名前と位置が必要です。

NIS+ テーブルを NIS マップから生成する — 作業マップ

表 9-2 NIS+ テーブルを NIS マップから生成する

作業	目的	参照先
NIS+ テーブルを NIS マップから生成する	NIS+ テーブルを NIS マップから生成する	194 ページの「NIS+ テーブルをマップから生成する方法」

▼ NIS+ テーブルをマップから生成する方法

1. データを転送する各 **NIS** マップをチェックします。
不正なエントリがないことを確認します。正しいデータが、所定の場所に正しい書式で記録されていることを確認します。エントリのうち、古いもの、無効なもの、破損しているものは削除します。また、不完全なエントリや一部のみのエントリも削除します。不完全なエントリは、設定を完了した後に追加する方が、不完全なエントリや破損したエントリを転送するよりも簡単です。
2. **NIS+** クライアントにログインする
この作業はどの **NIS+** クライアントからでも実行できます。ただし、そのクライアントは、情報の転送先となるテーブルと同じドメインに所属していなければなりません。ここで示す例では、ルートマスターサーバーを使用します。これらの例では、システム管理者はスーパーユーザーとしてログインしているため、この操作を実際に実行している (適切な資格とアクセス権を必要とする) **NIS+** 主体は、ルートマスターサーバーです。
3. このシェルのコマンド検索パスに **/usr/lib/nis** を追加します。
テーブルごとに **/usr/lib/nis/nisaddent** コマンドを使用するため、コマンド検索パスに **/usr/lib/nis** を追加しておく、毎回これを入力する必要があります。ここでは、**C** シェルユーザーの場合の例と **Bourne** シェルまたは **Korn** シェルのユーザーの場合の例を示します。
C シェルの場合

```
rootmaster# setenv PATH $PATH:/usr/lib/nis
```

Bourne シェルまたは Korn シェルの場合

```
rootmaster# PATH=$PATH:/usr/lib/nis
rootmaster# export PATH
```
4. **nisaddent** を使用して、次のマップを **1** 度に **1** つずつ転送します。
aliases, bootparams, ethers, group, hosts, netgroup, netmasks, networks, passwd, protocols, rpc, services.
publickey とオートマウントマップでは、少し手順が異なります。**publickey** ファイルの場合は**手順 6**に、オートマウントファイルの場合は**手順 7**に進んでください。

デフォルトでは、`nisaddent` はファイル情報をテーブル情報に追加します。置換またはマージを行うには、`-r` または `-m` のオプションを使用します。

置換する場合、次のように入力します。

```
rootmaster# nisaddent -r -y nisdomain table
```

追加する場合、次のように入力します。

```
rootmaster# nisaddent -a -y nisdomain table
```

マージする場合、次のように入力します。

```
rootmaster# nisaddent -m -y nisdomain table
```

はじめてテーブルを生成するときに最適なオプションは、デフォルトの `-a` オプションです。NIS+ テーブルを NIS マップまたは `/etc` 内のファイルと同期させるための最適なオプションは `-m` (マージ) オプションです。

`-y` (小文字) オプションは、テキストファイルではなく、NIS ドメインを示します。`nisdomain` 引数は、ユーザーが NIS+ テーブルに転送しようとしているマップを持つ NIS ドメインの名前です。実際のマップを指定する必要はありません。`nisaddent` ユーティリティは、`table` 引数に対応する NIS マップを自動的に選択します。次に例をいくつか示します。

```
rootmaster# nisaddent -m -y olddoc hosts
rootmaster# nisaddent -m -y olddoc passwd
rootmaster# nisaddent -m -y olddoc groups
```

最初の例では、`olddoc` (NIS) ドメイン内の `hosts.byname` マップと `hosts.byaddr` マップの内容を、ルートドメイン (NIS+) 内の NIS+ `hosts` テーブルに転送します。2 番目の例では、パスワード関連情報を格納している NIS マップを、NIS+ `Passwd` テーブルに転送します。3 番目の例では、グループ関連情報で同じことを実行します。`nisaddent` コマンドの詳細については、[第 19 章](#)を参照してください。

5. **publickey** マップを転送します。

ドメインの `cred` テーブルには、すでにいくつかの資格が格納されているため、`cred` テーブルに転送する `publickey` マップの内容によって、これらの資格が上書きされないように確認する必要があります。

- a. 初めに、**publickey** マップをファイルにダンプします。続いて、テキストエディタでそのファイルをオープンします。たとえば、次のように入力します。

```
rootmaster# makedbm -u /var/yp/olddoc/publickey.byname \
/etc/publickey.xfr
rootmaster# vi /tmp/publickey.tmp
```

- b. **publickey** マップから、ログインしているマシンの資格を削除します。

`rootmaster` に対しては、次のような行は、すべて削除してください。

```
unix.rootmaster@doc.com public-key:private-key [alg-type]
```

- c. これにより、マップではなく「ファイル」の内容を **cred** テーブルに転送できます。**nisaddent** に `-a` (追加) オプションを付けて実行します。

```
rootmaster# nisaddent -a -f /etc/publickey.xfr -t cred.org_dir Publickey
```

ただし、この操作は DES 資格を cred テーブルに転送するだけです。cred テーブルに対する自分の LOCAL 資格は自分で作成する必要があります。

6. オートマウント情報を転送します。

nissetup ユーティリティは auto_master テーブルと auto_home テーブルを作成しますが、これらは標準の NIS+ テーブルとはみなされません。したがって、これらのテーブルに情報を転送するには、少し異なる構文が必要となります。

```
rootmaster# nisaddent -y olddoc -Y auto.master -t auto_master.org_dir key-value
rootmaster# nisaddent -y olddoc -Y auto.home -t auto_home.org_dir key-value
```

NIS ドメイン名 (この例では olddoc) と同様、-m と -y のオプションが必要です。しかし、NIS マップ名 (auto.master など) の前には -Y (大文字) を付けなければなりません。次に、オートマウントの「テキストファイル」を転送するときに必要のように、標準の NIS+ テーブルであることを示す -t オプションを使用しなければなりません。この引数は、NIS+ ディレクトリオブジェクト (auto_master.org_dir) とテーブルの種類 (key-value) です。NIS+ テーブル名の後ろには必ず接尾辞 org_dir を追加してください。

7. nisping を実行して更新内容を複製サーバーに送ります。

nisping を実行すると、複製サーバーに変更が反映されます。

```
master1# nisping domain
master1# nisping org_dir.domaincom.
master1# nisping groups_dir.domain
```

8. テーブルに対しチェックポイントを実行します。

この手順により、ドメインをサポートしている全サーバーが、それらの .log ファイルからディスク上のテーブルのコピーに新しい情報を転送します。ルートドメインを設定したばかりの場合、そのルートドメインにはまだ複製サーバーがないため、この手順はルートマスターサーバーだけが対象となります。nisping コマンドに -c (大文字) オプションを付けて実行します。

```
rootmaster# nisping -C org_dir
Checkpointing replicas serving directory org_dir.doc.com. :
Master server is rootmaster.doc.com.
Last update occurred at July 14, 1994
Master server is rootmaster.doc.com.
checkpoint succeeded.
```

スワップ空間が不足している場合、サーバーはチェックポイントを正常に実行できませんが、そのことを通知しません。スワップ空間が十分あることを確認する方法として、niscat コマンドを使ってテーブルの内容をリスト表示する方法があります。スワップ空間が不足している場合、次のエラーメッセージが表示されます。

```
can't list table: Server busy, Try Again.
```

このメッセージ内容からはわかりませんが、これはスワップ空間の不足を示しています。スワップ空間を増やし、このドメインに再びチェックポイントを実行します。

NIS+ から NIS に情報を転送する

ここでは、NIS+ テーブルの内容を、Solaris 1.x の NIS マスターサーバー上の NIS マップに転送します。手順を次に示します。

- 1. NIS+ サーバーにログインします。
- 2. NIS+ テーブルを出力ファイルに転送します。
- 3. 出力ファイルの内容を NIS マップに転送します。

NIS から NIS+ に情報を転送する際のセキュリティについて

この作業を実行するには、内容を転送する各テーブルへの読み取り権が必要です。

NIS から NIS+ に情報を転送するための前提条件

マップを NIS サーバー上にあらかじめ作成しておかなければなりません。

NIS+ から NIS に情報を転送する — 作業マップ

表 9-3 NIS+ から NIS に情報を転送する

作業	目的	参照先
NIS+ から NIS に情報を転送する	NIS+ テーブルから Solaris 1.x の NIS のマスターサーバー上の NIS マップに情報を転送する	197 ページの「NIS+ から NIS へ情報を転送する方法」

▼ NIS+ から NIS へ情報を転送する方法

- 1. NIS+ サーバーにログインします。
この例では、`dualserver` という名前のサーバーを使用します。
- 2. NIS+ テーブルを出力ファイルに転送します。
次に示すように、各テーブルごとに 1 回、`-d` オプションを付けた `nisaddent` コマンドを使用します。

```
dualserver% /usr/lib/nis/nisaddent -d -t table tabletype > filename
-d オプションは、table の内容を /etc 内の標準のファイル形式に変換して
filename に出力します。
```

3. 出力ファイルの内容を **NIS** マップに転送します。

NIS+ の出力ファイルは、NIS マップ用の入力ファイルとして使用できる ASCII ファイルです。これらを NIS マスターの /etc ディレクトリにコピーし、通常の方法で make を実行します。

```
dualserver# cd /var/yp
dualserver# make
```

所有者および管理者に対する Passwd へのアクセス制限

ここでは、passwd テーブルのパスワードに関係する列に対する読み取り権を所有者とテーブルの管理者だけに制限し、しかも passwd テーブル内の残りの列に対する認証主体 (アプリケーションを含む) の読み取り権に影響を与えない方法を説明します。

この作業では、次のアクセス権を設定します。

	Nobody	Owner	Group	World
Table Level Rights:	----	rmcd	rmcd	----
Passwd Column Rights:	----	rm--	rmcd	----
Shadow Column Rights:	----	rm--	rmcd	----

Passwd 列へのアクセスを制限する際のセキュリティについて

- ドメインを NIS 互換モードで動作させないでください。
- ドメインのすべての NIS+ クライアントには DES 資格が必要です。
- ドメインのすべてのクライアントで、Solaris リリース 2.3 以降のリリースが実行されている必要があります。
- ユーザーのネットワークパスワード (DES 資格の暗号化に使用) は、ログインパスワードと同じディレクトリでなければなりません。

Passwd 列へのアクセスを制限するための前提条件

- passwd テーブルをあらかじめ設定しておかなければなりません。ただし、情報が入っている必要はありません。
- この作業を実行する NIS+ 主体には、passwd テーブルへの変更権が必要です。

必要な情報

必要なのは passwd テーブルの名前だけです。

所有者および管理者に対する Passwd 列へのアクセス制限 — 作業マップ

表 9-4 所有者および管理者に対する Passwd 列へのアクセス制限

作業	目的	参照先
所有者および管理者に対する Passwd 列へのアクセス制限	NIS+ のコマンドを使って passwd.org_dir を変更し、所有者および管理者に対する passwd 列へのアクセスを制限する	199 ページの「パスワード列へのアクセスを制限する方法」

▼ パスワード列へのアクセスを制限する方法

- ドメインのマスターサーバーにログインします。
この例ではルートマスターサーバー rootmaster を使用します。
- 現在のテーブルと列のアクセス権を確認します。
niscat -o コマンドを実行します。

```
rootmaster# niscat -o passwd.org_dir
この作業では、次のような既存のアクセス権になっていることを前提としています。

Access Rights      : ----rmcdrmcdr---
Columns           :
[0]  Name           : name
     Access Rights  : r-----r---
[1]  Name           : passwd
     Access Rights  : ----m-----
[2]  Name           : uid
     Access Rights  : r-----r---
[3]  Name           : gid
     Access Rights  : r-----r---
[4]  Name           : gcos
     Access Rights  : r----m-----r---
[5]  Name           : home
     Access Rights  : r-----r---
[6]  Name           : shell
     Access Rights  : r-----r---
[7]  Name           : shadow
     Access Rights  : r-----r---
```

自分のアクセス権が異なる場合、別の構文を使用する必要があります。手順については、[第 15 章](#)を参照してください。

3. テーブルのアクセス権を変更します。
nischmod コマンドを実行して、テーブルのオブジェクトレベルのアクセス権を ----rmcdrmcd---- に変更します。

rootmaster# nischmod og=rmcd,nw= passwd.org_dir
4. 列のアクセス権を変更します。
nistbladm コマンドを -u オプションを付けて使用して、passwd 列 および shadow 列のアクセス権を次のように変更します。

passwd ---- rm-- ---- ----
shadow ---- r--- ---- ----
rootmaster# nistbladm -u passwd=o+r, shadow=o+r passwd.org_dir
5. 新しいアクセス権を確認します。
手順 2 の場合と同様、niscat -o コマンドを実行します。アクセス権は、[手順 2](#) の出力と同じでなければなりません。

テーブルの生成のまとめ

NIS+ テーブルの生成に必要な手順のまとめを次に示しています。これは、もっとも簡単な場合を想定しています。このため、まとめを参考として使用する前に、作業の詳細を十分に理解しておいてください。また、このまとめでは各コマンドに対するサーバーの応答は示していません。

表 9-5 NIS+ テーブルへのファイルの転送 -コマンドのまとめ

作業	コマンド
NIS+ クライアントにログインする	rootmaster%
転送するファイルの作業用コピーを作成する	% cp /etc/hosts /etc/hosts.xfr
検索パスに /usr/lib/nis を追加する	% PATH=\$PATH:/usr/lib/nis; export PATH
各ファイルを 1 つずつ転送する	% nisaddent -m -f /etc/hosts.xfr hosts
publickey ファイルから古いサーバー資格を削除する	% vi /etc/publickey.xfer
資格テーブルに publickey ファイルを転送する	% nisaddent -a -f /etc/publickey.xfr cred

表 9-5 NIS+ テーブルへのファイルの転送 -コマンドのまとめ (続き)

作業	コマンド
オートマウントファイルを転送する	% nisaddent -f auto.master.xfr -t auto_master.org_dir key-value % nisaddent -f auto.home.xfr -t auto_home.org_dir key-value
テーブルディレクトリに チェックポイントを実行する	% nisping -C org_dir

表 9-6 NIS+ テーブルへのマップの転送 -コマンドのまとめ

作業	コマンド
NIS+ クライアントにログインする	rootmaster%
検索パスに /usr/lib/nis を追加する	% PATH=\$PATH:/usr/lib/nis; export PATH
各マップを1つずつ転送する	% nisaddent -m -y olddoc hosts
publickey マップをファイルに ダンプする	% makedbm -u /var/yp/olddoc/publickey.byname > /etc/publickey.xfr
新しい資格を削除する	% vi /etc/publickey.xfr
publickey ファイルを転送する	% nisaddent -a -f /etc/publickey.xfr -t cred.ortg_dir publickey % nisaddent -y olddoc -Y auto.master -t auto_master.org_dir key-value % nisaddent -y olddoc -Y auto.home -t auto_home.org_dir key-value
オートマウントマップを転送する	
テーブルディレクトリにチェック ポイントを実行する	% nisping -C org_dir

表 9-7 NIS マップへの NIS+ テーブルの転送 -コマンドのまとめ

作業	コマンド
NIS+ サーバーにログインする	dualserver#
NIS+ テーブルをファイルに転送する	% /usr/lib/nis/nisaddent -d [-t table] tabletype > filename
ファイルを NIS マップに転送する	% makedbm flags output-file NIS-dbm-file

表 9-8 Passwd 列へのアクセスの制限 -コマンドのまとめ

作業	コマンド
ドメインのマスターサーバーに ログインする	rootmaster#
テーブルの既存の権利を チェックする	# niscat -o passwd.org_dir
テーブルに新しい権利を割り当 てる	# nischmod og=rmcd,nw= passwd.org_dir
列に新しい権利を割り当てる	# nistbladm -u passwd=o+r, shadow=n+r passwd.org_dir
新しい権利を確認する	# niscat -o passwd.org_dir

パート III NIS+ の管理

このパートでは、Solaris OS 上での NIS+ ネームサービスの管理および問題解決について説明します。

第 10 章

NIS+ のテーブルと情報

この章では、これらのテーブルの構造と設定方法の概要について説明します。

注 - NIS+ は、将来のリリースではサポートされなくなる可能性があります。NIS+ から LDAP への移行支援ツールは、Solaris 9 リリース以降で使用できます (『Solaris のシステム管理 (ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』を参照)。詳細については、<http://www.sun.com/directory/nisplus/transition.html> を参照してください。

NIS+ テーブル構造

NIS+ は、さまざまなネットワーク情報をテーブルに格納します。NIS+ テーブルには、単純なテキストファイルやマップにはない、いくつかの機能があります。これらのテーブルは、列エントリ構造を持ち、検索パスを受け付けます。また、これらのテーブルをリンクして、いくつかの異なる方法で構成することもできます。NIS+ には事前構成された 16 個のシステムテーブルがあり、独自のテーブルを作成することもできます。表 10-1 は、事前構成した NIS+ テーブルを示します。

注 - ネームサービスとして、NIS+ テーブルはオブジェクト自体ではなく、オブジェクトの参照情報を格納するように設計されています。そのため、NIS+ はエントリの大きなテーブルをサポートできません。テーブルに非常に大きなエントリがある場合、`rpc.nisd` は機能しません。

表 10-1 NIS+ のテーブル

テーブル	テーブル内の情報
hosts	ドメイン内の全マシンのネットワークアドレスとホスト名
bootparams	ドメイン内の全ディスクレスクライアントのルート、スワップ、およびダンプパーティションの位置
passwd	ドメイン内の全ユーザーに関するパスワード情報
cred	ドメインに所属する主体の資格
group	ドメイン内の全 UNIX グループのグループ名、グループパスワード、グループ id、およびメンバー
netgroup	ドメイン内のマシンやユーザーが所属できるネットグループ
mail_aliases	ドメイン内のユーザーのメール別名に関する情報
timezone	ドメイン内の全マシンの時間帯
networks	ドメイン内のネットワークとこれらの標準の名前
netmasks	ドメイン内のネットワークとこれらに関連付けられているネットマスク
ethers	ドメイン内の全マシンの Ethernet アドレス
services	ドメイン内で使用される IP サービス名とそれらのポート番号
protocols	ドメイン内で使用される IP プロトコルのリスト
RPC	ドメイン内で使用可能な RPC サービスの RPC プログラム番号
auto_home	ドメイン内の全ユーザーのホームディレクトリの位置
auto_master	自動マウンタのマップ情報

cred テーブルには NIS+ のセキュリティに関係する情報だけが入っています。これについては第 12 章で説明します。

これらのテーブルには、ユーザー名からインターネットサービスまでの幅広い情報が格納されています。この情報の多くは、設定時または構成の手順の中で生成されます。たとえば、password テーブル内のエントリは、ユーザーアカウントが設定されたときに作成されます。hosts テーブル内のエントリは、マシンがネットワークに追加されたときに作成されます。また、networks テーブル内のエントリは、新しいネットワークが設定されたときに作成されます。

この情報はこのような広範囲にわたる操作によって生成されるため、このマニュアルでは詳細を説明していません。ただし、使いやすさを考慮して、テーブルの各列に含まれる情報は、第 23 章に要約してあります。詳細な説明は、グループを NIS+ グループやネットグループと区別する場合など、理解しにくい内容についてのみ行います。テーブルの情報の詳細は、Solaris の「システム管理マニュアルセット」と「ネットワーク管理マニュアルセット」を参照してください。

注- ドメインにさらに多くのオートマウントマップを作成できますが、必ず NIS+ テーブルとして格納し、auto_master テーブルの中に入れてください。オートマウントマップを作成して、ユーザー用に作成された auto_master に追加するときは、列名にはキー (key) と値 (value) が含まれている必要があります。オートマウントの詳細は、オートマウントまたは NFS ファイルシステムの関連マニュアルを参照してください。

列とエントリ

NIS+ テーブルにはさまざまな種類の情報が格納されていますが、これらの基本構造はすべて同じで、行と列から構成されます。この列は、「エントリ」または「エントリオブジェクト」と呼ばれます。

ホスト名 列			
	nose		
	grass		
	violin		
	baseball		

クライアントは、キー、または検索可能な任意の列によりアクセスできます。たとえば、baseball という名前のマシンのネットワークアドレスを見つけるには、ホスト名の列を探して baseball を見つけます。

ホスト名 列			
	nose		
	grass		
	violin		
	baseball		

次に、baseball のエントリ行を移動して、そのネットワークアドレスを見つめます。

	アドレス 列	ホスト名 列		
		nose		
		grass		
		violin		
Baseball 行	129.44.1.2	baseball		

クライアントは、任意のレベルでテーブル情報にアクセスできる (つまり、テーブルに全体としてアクセスできる) ため、NIS+ は 3 つのレベルすべてにセキュリティ機構を提供しています。たとえば、管理者は、オブジェクトレベルで全員にテーブルの読み取り権を割り当て、列レベルで所有者に変更権を割り当て、エントリレベルでグループに変更権を割り当てることができます。テーブルのセキュリティの詳細は、[第 15 章](#)で説明しています。

検索パス

テーブルには、その「ローカル」ドメインについての情報だけが収められています。たとえば、doc.com. ドメイン内のテーブルには、doc.com. ドメインのユーザー、クライアント、およびサービスについての情報だけが収められています。また、sales.doc.com. ドメイン内のテーブルには、sales.doc.com. ドメインのユーザー、クライアント、およびサービスについての情報だけが収められます。

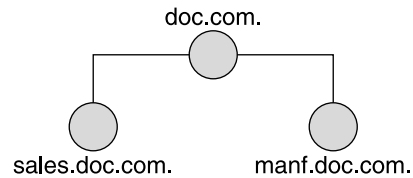
あるドメイン内のクライアントが、別のドメインに格納されている情報を見つけようとした場合、完全指定名を使用しなければなりません。[78 ページの「環境変数 NIS_PATH」](#)で説明したように、環境変数 NIS_PATH が正しく設定されていれば、NIS+ サービスによって自動的に処理されます。

情報を探すときにサーバーがたどる「検索パス」をどの NIS+ テーブルでも指定できます。この検索パスは、NIS+ テーブルをコロンで区切って順番に並べたものです。

```
>tabletable:table...
```

検索パス内のテーブル名は、完全指定する必要はありません。テーブル名は、コマンド行で入力された名前と同じように展開されます。サーバーが自分のローカルテーブル内に情報を検出できなかった場合、サーバーはクライアントにテーブルの検索パスを返します。クライアントはそのパスを使用して、その情報が見つかるかまたは最後まで、検索パスに名前があるすべてのテーブルを順番に探していきます。

検索パスのメリットを次に示します。次のようなドメイン階層があるとします。



下位にある 2 つのドメインの hosts テーブルの内容は次のようになります。

表 10-2 hosts テーブルの例

sales.doc.com.		manf.doc.com.	
10.0.0.1	localhost	172.18.0.1	localhost
10.22.3.22	luna	172.29.6.1	sirius
10.22.3.24	phoebus	172.29.6.112	rigel
10.22.3.25	europa	172.29.6.90	antares
10.22.3.27	ganymede	172.29.6.101	polaris
10.22.3.28	mailhost	172.29.6.79	mailhost

ここで、sales.doc.com. ドメイン内の luna にログインしたユーザーが、別のクライアントにリモートログインする場合を考えてみます。このユーザーが完全指定名を指定しない場合、そのユーザーがリモートログインできるのは localhost、phoebus、europa、ganymede、および mailhost の 5 台のマシンだけです。
localhost, phoebus, europa, ganymede, and the mailhost.

sales.doc.com. ドメインの hosts テーブルの検索パスに、manf.doc.com. ドメインの hosts テーブルが指定されていると仮定します。

hosts.org_dir.manf.doc.com.

これで sales.doc.com. ドメイン内のユーザーは、たとえば、rlogin sirius と入力することができ、NIS+ サーバーはこれを検出します。NIS+ サーバーはまずローカルドメインで sirius を探し、見つからなければ manf.doc.com. ドメイン内を探します。クライアントは manf.doc.com. ドメインをどのように認識するのでしょうか。第 2 章で説明したように、この情報はディレクトリキャッシュに格納されています。これがディレクトリキャッシュにない場合、クライアントは、第 2 章で説明したプロセスに従ってこの情報を入手します。

ただし、検索パスを指定する方法には、若干の欠点があります。ユーザーがたとえば、`rlogin luba` (`luna` でなく) などと間違った名前を入力した場合、サーバーは 1 つのテーブルではなく、3 つのテーブルを探さなければエラーメッセージを返せません。名前空間の全域に検索パスを設定した場合、1 つの操作は 2、3 個のドメインではなく、10 個のドメイン内のテーブルを検索してから終了します。もう 1 つの欠点は、多数のクライアントが NIS+ テーブルにアクセスする必要があるとき、複数のサーバーのセットと通信するために、パフォーマンスの低下を招くという点です。

また、`mailhost` は別名として使用されることが多いため、特定のメールホストについての情報を探したいときは、その完全指定名 (`mailhost.sales.doc.com.` など) を使用しなければなりません。そうしないと、NIS+ は検索したすべてのドメインで見つけたすべてのメールホストを返します。

テーブルの検索パスを指定するには、[352 ページの「テーブルでの `nistbladm` コマンドの使用](#)」で説明するように、`nistbladm` コマンドに `-p` オプションを付けて実行します。

テーブルの設定方法

NIS+ テーブルの設定には 3 つまたは 4 つの作業が伴います。

1. `org_dir` ディレクトリの作成
2. システムテーブルの作成
3. システムテーブル以外のテーブルの作成 (必要に応じて)
4. 情報を入力してテーブルを生成 (`populate`) する

[56 ページの「NIS+ のファイルとディレクトリ」](#)で説明したように、NIS+ のシステムテーブルは `org_dir` ディレクトリに格納されます。したがって、テーブルを作成するには、その前にテーブルを入れる `org_dir` ディレクトリを作成しなければなりません。これには 3 つの方法があります。

- `nissserver` スクリプトを使用。`nissserver` スクリプトは、ディレクトリとシステムテーブルのフルセットを作成します。`nissserver` スクリプトを実行することを推奨します。
- `nismkdir` コマンドを使用。`nismkdir` コマンドは単にディレクトリを作成するものです。
- `/usr/lib/nis/nissetup` ユーティリティを使用。`nissetup` ユーティリティは `org_dir` および `groups_dir` ディレクトリとシステムテーブルのフルセットを作成します。

`nissserver` スクリプトと、`nissetup` および `nismkdir` ユーティリティについては、[332 ページの「`nismkdir` コマンド](#)」を参照してください。`nismkdir` コマンドに関するその他の情報については、[332 ページの「`nismkdir` コマンド](#)」に説明があります。

`nissetup` ユーティリティのメリットは、サーバーが NIS 互換モードで動作しているドメインのテーブルに対して、適切なアクセス権を割り当てる機能です。`-y` フラグを付けて実行すると、このユーティリティは作成するオブジェクトの「未認証」カテゴリに読み取りアクセス権を割り当てます。その結果、未認証の NIS クライアントは、そのドメインの NIS+ テーブルから情報を得ることができます。

NIS+ の 16 個のシステムテーブルと、これらに格納される情報の種類については、[第 10 章](#)で説明します。これらを作成するには、前述の 3 つの方法を使用できます。`nistbladm` コマンドは NIS+ テーブルの作成と変更を行います。`nistbladm` コマンドで名前空間内のすべてのテーブルを作成できますが、入力量が多くなり、列の正確な名前やアクセス権を知っている必要があります。`nisserv` スクリプトを使う方が、はるかに簡単です。

システムテーブル以外のテーブル、つまり NIS+ によってまだ構成されていないテーブルを作成するには、`nistbladm` コマンドを使用します。オートマウントマップを追加する場合は、最初の列の名前が `key`、2 番目の列の名前が `value` になっている必要があります。

NIS+ テーブルを生成するには、NIS マップから行う方法、ASCII ファイル (`/etc` 内のファイルなど) から行う方法、および手作業の 3 つの方法があります。

NIS サービスからアップグレードしている場合、すでにネットワーク情報の大部分が NIS マップに格納されています。この情報を NIS+ テーブルに手作業で再入力する必要はありません。`nispopulate` スクリプトか `nisaddent` ユーティリティを使用すると自動的に転送できます。

ほかのネットワーク情報サービスを使用していない場合で、しかもネットワークデータを `/etc` 内のファイルで管理しているときも、この情報を再入力する必要はありません。`nispopulate` スクリプトか `nisaddent` ユーティリティを使用しても自動的に転送できます。

ネットワークをはじめて設定する場合、多くのネットワーク情報をすでにどこかに格納しているということはおそらくないでしょう。この場合、まず情報を入手して、次に NIS+ テーブルに手作業で入力する必要があります。これを行うには、`nistbladm` コマンドを使用できます。また、特定のテーブルの全情報を「入力ファイル」(これは本質的に `/etc` 内のファイルと同じです)に入力し、次に、`nispopulate` スクリプトか `nisaddent` ユーティリティを使用してファイルの内容を転送することによって同じことが行えます。

テーブルの更新方法

ドメインが設定されると、サーバーはそのドメインの NIS+ テーブルの最初のバージョンを受け取ります。これらのバージョンはディスクに格納されますが、サーバーは起動されると、これらをメモリーにロードします。サーバーがテーブルに対する更新を受信すると、サーバーはそのメモリーにあるテーブルをすぐに更新します。サーバーが情報の要求を受信すると、その応答のためにメモリーにあるコピーを使用します。

もちろん、サーバーはその更新をディスクにも格納する必要があります。ディスクにあるテーブルを更新するには時間がかかるため、NIS+ の全サーバーはテーブル用の「ログ」ファイルを持っています。このログファイルは、テーブルに対して行われた変更内容を、ディスク上で更新できるまで、一時的に格納するように設計されています。ログファイルは、テーブル名を接頭辞として使用し、これに `.log` を追加します。たとえば、以下のようになります。

```
hosts.org_dir.log
bootparams.org_dir.log
password.org_dir.log
ipnodes.org_dir.log
```

ログファイルが大きくなりすぎて大量のディスク領域を占有しないように、ディスクにあるテーブルのコピーは毎日更新しなければなりません。このプロセスは「チェックポイントを実行する」と呼ばれます。これには、`nisping -c` コマンドを使用します。

第 11 章

NIS+ のセキュリティの概要

この章では、NIS+ のセキュリティシステムと、システムが NIS+ 名前空間全体に与える影響について説明します。

注 - NIS+ は、将来のリリースではサポートされなくなる可能性があります。NIS+ から LDAP への移行支援ツールは、Solaris 9 リリース以降で使用できます (『Solaris のシステム管理 (ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』を参照)。詳細については、

<http://www.sun.com/directory/nisplus/transition.html> を参照してください。

Solaris のセキュリティ - 概要

Solaris のセキュリティは、ユーザーが Solaris システムに入るために通過しなければならないゲートと、内部でそのユーザーが何をできるのかを判定するアクセス権マトリックスの 2 つによって確保されています。このセキュリティシステムの概略については、[図 11-1](#) を参照してください。

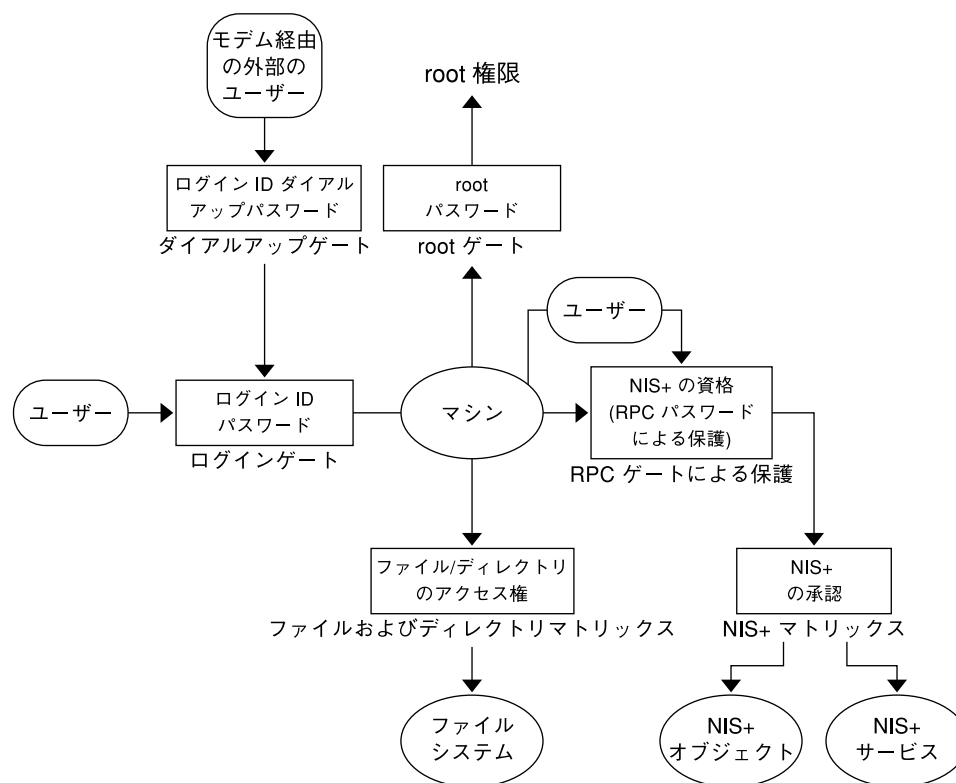


図 11-1 Solaris のセキュリティゲートとフィルタ

上の図に示すように、システム全体は 4 つのゲートと 2 つのアクセス権マトリックスで構成されています。

- 「ダイアルアップゲート」。モデムや電話回線により Solaris システムにアクセスするには、正しいログイン ID とダイアルアップパスワードが必要です。
- 「ログインゲート」。Solaris システムに入るには、正しいログイン ID とユーザーパスワードが必要です。
- 「ファイルおよびディレクトリマトリックス」。Solaris システムに入ってから、ファイルやディレクトリに関して、読み取り、実行、変更、作成、削除等の各種操作を行う権限はアクセス権マトリックスによってコントロールされます。
- 「root ゲート」。root 権限にアクセスするには、正しいスーパーユーザー (root) パスワードが必要です。
- 「RPC ゲートによる保護」。セキュリティレベル 2 (デフォルト値) の NIS+ 環境においては、NIS+ サービスを使って NIS+ オブジェクト (サーバー、ディレクトリ、テーブル、テーブルエントリなど) にアクセスしようとする場合、ユーザー ID は Secure RPC プロセスを使った NIS+ によって確認されます。

ダイヤルアップ、ログイン、ルートゲート、およびファイルとディレクトリのアクセス権マトリクスの詳細は、Solaris の各マニュアル及び資料を参照してください。

Secure RPC ゲートをパスするには、Secure RPC パスワードが必要です。「Secure RPC パスワード」は、「ネットワークパスワード」と呼ばれる場合もあります。Secure RPC パスワードとログインパスワードは通常同一なので、ゲートをパスした場合はパスワードを再入力する必要がありません(2つのパスワードが異なる場合についての説明は、235 ページの「Secure RPC パスワードとログインパスワードの問題」を参照)。

ユーザー要求が Secure RPC ゲートをパスするには、1 組の「資格」セットが使われます。ユーザーの資格を、作成、提示、判定するプロセスを、そのプロセスがユーザーがだれであり、正しい RPC パスワードを所持しているか否かを確認することから、「認証 (authentication)」と呼びます。この認証プロセスは、ユーザーが NIS+ サービスを要求するごとに自動的に実行されます。

NIS 互換モード (YP 互換モードとも呼ばれる) の NIS+ 環境では、ユーザーが正しい資格を所持しているか否かにかかわらず (認証プロセスによって、ID が確認され、Secure RPC パスワードがチェックされたか否かにかかわらず)、だれでもすべての NIS+ オブジェクトを読み取ることができ、またそれらエントリを変更できるので、Secure RPC ゲートによるガードは極めて弱くなります。だれもがすべての NIS+ オブジェクトを読み取りかつそれらエントリを変更できるので、互換モードの NIS+ ネットワークは通常モードのよりも安全性が低くなります。

NIS+ の認証と資格の作成および管理については、第 12 章を参照してください。

- 「NIS+ オブジェクトマトリックス」。一度正しく認証されると、ユーザーが NIS+ オブジェクトを読み取り、変更、作成、削除する権限はアクセス権マトリックスに管理されます。このプロセスを NIS+ の「承認 (authorization)」と呼びます。

NIS+ のアクセス権および承認の詳細は、第 15 章を参照してください。

NIS+ のセキュリティ - 概要

NIS+ のセキュリティは NIS+ の名前空間全体にかかわっています。セキュリティと名前空間を別に設定することはできません。このため、セキュリティの設定方法は名前空間の各構成要素を設定する手順と密接に関係することになります。一度 NIS+ のセキュリティ環境を設定すると、その後はユーザーの追加および削除、アクセス権の変更、グループメンバーの再割当やネットワーク展開を管理するために必要なその他すべての管理ルーチンタスクを実行できます。

NIS+ のセキュリティ機能は名前空間内の情報と、名前空間そのものの構造を不正アクセスから保護するものです。このセキュリティ機能がなければ、どんな NIS+ クライアントであっても名前空間に保存された情報を獲得することも変更することもできないだけでなく、ダメージを与える可能性があります。

NIS+ セキュリティには次の 2 つの機能があります。

- 「認証 (authentication)」。認証は NIS+ 主体を識別するのに使われます。主体 (ユーザーまたはマシン) が NIS+ オブジェクトにアクセスしようとするときはいつも、ユーザー ID と Secure RPC パスワードが確認されチェックされます。
- 「承認 (authorization)」。承認はアクセス権を識別するのに使われます。NIS+ 主体が NIS+ オブジェクトにアクセスしようとするときはいつも、所有者、グループ、その他、未認証の 4 つのクラスの 1 つに位置付けられています。NIS+ セキュリティシステムは NIS+ 管理機能により、各クラスに NIS+ オブジェクトに対する読み取り、変更、作成、削除の各権限を指定します。たとえば、あるクラスは passwd テーブルの特定列を変更できるが、その列を読み取ることはできないとか、別のクラスはいくつかのエントリを読み取ることはできるが、それ以外はできないというように設定できます。

NIS+ のセキュリティ機能は 2 段階のプロセスを用意しています。

1. 「認証 (authentication)」。NIS+ は資格 (credential) を使ってユーザーを確認します。
2. 「承認 (authorization)」。承認プロセスがユーザー ID を確認すると、NIS+ はクラスを判定します。NIS+ オブジェクトまたはサービスに対して何が行えるのかは、ユーザーがどのクラスに属しているかに依存します。これは UNIX の標準的なファイルおよびディレクトリのアクセス権システムと同様の考え方に基づいています (クラスの詳細は、221 ページの「承認クラス」を参照)。

このプロセスは、マシン A のルート権限により su コマンドを使って、第 2 の NIS+ アクセス権で NIS+ オブジェクトにアクセスすることを防ぐものです。

しかしながら NIS+ は、他人のユーザーログインパスワードを知っている者が、そのユーザーになりすましてユーザー ID と NIS+ アクセス権を使用することを妨げることはできません。またルート権限を持つユーザーが同一マシンからログインしている別のユーザーになりすますのを防ぐこともできません。

図 11-2 に、このプロセスの詳細を示します。

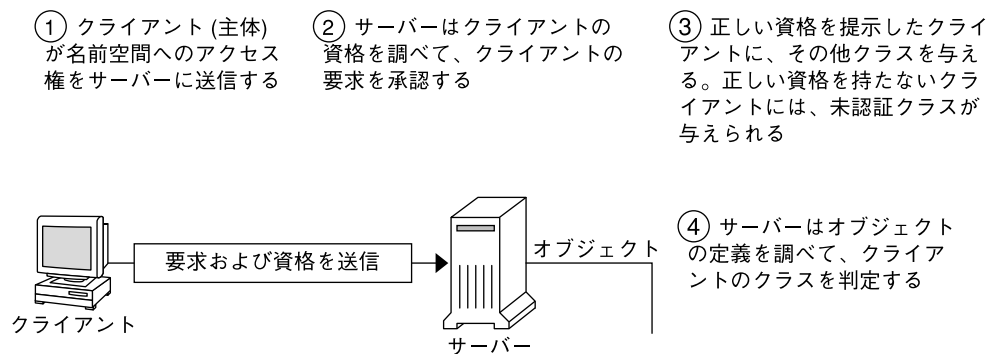


図 11-2 NIS+ セキュリティプロセスの概要

NIS+ の主体について

NIS+ 主体とは、NIS+ サービスに要求をするエンティティ (クライアント) のことです。NIS 主体には、クライアントマシンに一般ユーザーとしてログインしたユーザー、スーパーユーザーとしてログインしたユーザー、NIS+ クライアントマシンにおいてスーパーユーザー特権で動作しているプロセスを含みます。つまり NIS+ 主体は、クライアントユーザーである場合と、クライアントマシンである場合とがあります。

NIS+ の主体は、NIS+ サーバーから NIS+ サービスを提供する主体を指すこともあります。すべての NIS+ サーバーは NIS+ クライアントにもなるので、サーバーが NIS+ の主体となる場合もあります。

NIS+ セキュリティレベル

NIS+ サーバーは、2 つのセキュリティレベルのどちらかで動作します。セキュリティレベルにより、要求を送信して認証を取得しなければならない資格主体の種類が決まります。NIS+ は、最高セキュリティレベル (セキュリティレベル 2) で動作するように設計されています。レベル 0 は、テスト、設定、およびデバッグのときにだけ使用します。セキュリティレベルについては、表 11-1 を参照してください。

表 11-1 NIS+ セキュリティレベル

セキュリティレベル	目的
0	セキュリティレベル 0 は、NIS+ 名前空間の初期テストと初期設定を行うレベル。セキュリティレベル 0 で動作している NIS+ サーバーは、ドメイン内の全 NIS+ オブジェクトに対する完全なアクセス権をどの NIS+ 主体にも与える。レベル 0 はシステム管理者だけが設定管理のためにだけ使用する。レギュラーユーザーはネットワークにおける通常のオペレーションに使用できない
1	セキュリティレベル 1 は AUTH_SYS セキュリティを利用する。このレベルは NIS+ によってサポートされていないため、利用すべきではない
2	セキュリティレベル 2 はデフォルトで、NIS+ が現在提供している最高レベルのセキュリティ。DES 資格を使用する要求だけを認証する。資格を持たない要求には未認証クラスが与えられ、未認証クラスに割り当てられたアクセス権が与えられる。無効な DES 資格を使用する要求は再試行される。有効な DES 資格の獲得に繰り返し失敗すると、無効資格を持った要求として認証エラーになる(資格が無効になる場合、要求をした主体に対してそのマシンで keylogin が実行されていない、クロックの同期がとれていない、鍵が不一致であるなどの原因が考えられる)。

セキュリティレベルとパスワードコマンド

Solaris リリース 2.0 から 2.4 の環境では、`nispasswd` を使用してパスワードを変更します。ただし、資格がなければ `nispasswd` は機能しません (より上位のレベルでの資格が残っている場合を除き、セキュリティレベル 0 では機能しない)。Solaris リリース 2.5 以降の場合は、セキュリティレベルや資格ステータスにかかわらず、`passwd` コマンドを使ってパスワードを変更します。

NIS+ の認証と資格 - 紹介

NIS+ 資格の目的は、NIS+ サービスや NIS+ オブジェクトへのアクセスを要求する各主体の ID を「認証」(確認) することにあります。つまり、NIS+ 資格および承認プロセスは Secure RPC システムを実装することです。

資格および認証システムは他人になりすますのを防ぐシステムです。あるマシンのルート権限を持つユーザーが `su` コマンドを使って、ログインしていないもしくは別のマシンにログインしている第 2 のユーザーになりすまし、NIS+ アクセス権を使用して NIS+ オブジェクトにアクセスすることを妨げます。

サーバーが主体を認証すると、主体は 4 つの認証クラスの 1 つに分類されます。次に、サーバーは主体がアクセスしようとしている NIS+ オブジェクトをチェックして、主体のクラスが実行を認証される動作を確認します。承認の詳細は、[220 ページの「NIS+ の承認とアクセス - 紹介」](#)を参照してください。

ユーザーおよびマシンの資格

主体には、「ユーザー」と「マシン」の 2 つの基本的な種類があります。

- 「ユーザー (user) 資格」。一般ユーザーとして NIS+ クライアントにログインした場合、NIS+ サービス要求には当人の「ユーザー」資格が含まれます。
- 「マシン (machine) 資格」。スーパーユーザーとして NIS+ クライアントにログインした場合、サービス要求は「クライアントマシン」資格を使います。

DES と LOCAL 資格

NIS+ 主体には DES と LOCAL の 2 つの資格が用意されています。

DES 資格

注 - 認証を行う仕組みとしては DES 資格が唯一のものです。将来はほかの仕組みも利用できるようになるかもしれません。DES 資格と NIS+ 資格は同等のものではありません。

このマニュアルでは、DES 資格という用語は、鍵の長さにかかわらず Diffie - Hellman 鍵をベースにした認証を表すものとして一般的に使われています。鍵の長さはあらかじめ決められたセットから指定することができます。Diffie - Hellman 鍵の長さの設定や表示を行うには `nisauthconf (1M)` を使用します。

DES (データ暗号化規格) 資格は資格の 1 種であり、保護認証を提供するものです。このマニュアルで NIS+ 主体を認証するために資格をチェックすると記述している場合は、NIS+ がチェックしている DES 資格のことを意味します。

主体が NIS+ サービスを要求したり、NIS+ オブジェクトにアクセスするごとに、ソフトウェアは格納してある資格情報を使って主体の資格を作成します。DES 資格は NIS+ 管理機能によって各主体に作られた情報から作成されます。第 12 章を参照してください。

- 主体の DES 資格の正当性を NIS+ が確認すると、その主体は「認証」されます。
- 主体は所有者、グループ、その他のいずれかの承認クラスを得るために認証されなければなりません。つまり、これらクラスの 1 つを得るために、有効な DES 資格を持つ必要があるのです。有効な DES 資格を持たない主体は自動的に未認証クラスに割り当てられます。
- 当該主体がクライアントユーザーであるかクライアントマシンであるかにかかわらず、DES 資格情報は常に主体のホームドメインの `cred` テーブルに格納されます。

LOCAL 資格

LOCAL 資格は、ユーザー ID 番号とホームドメイン名を含む NIS+ 主体名とのマップです。ユーザーがログインすると、システムは DES 資格が格納されているホームドメインを特定する LOCAL 資格をチェックします。システムはその情報を使ってユーザーの DES 資格情報を獲得します。

ユーザーがリモートドメインにログインした場合、その要求はユーザーのホームドメインを示す LOCAL 資格を使います。NIS+ は次にユーザーの DES 資格情報を知るためにユーザーのホームドメインを問い合わせます。こうして、たとえユーザーの DES 資格情報がリモートドメインに格納されていなくても、リモートドメインで認証されます。

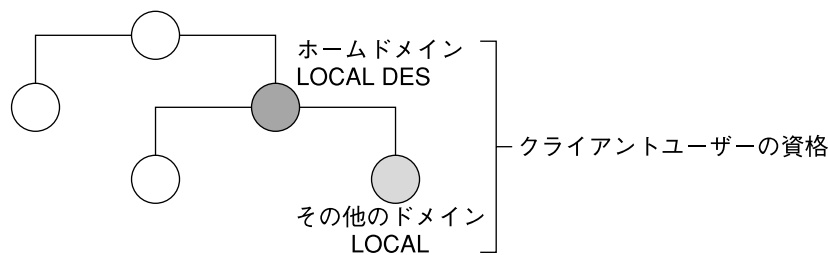


図 11-3 資格とドメイン

LOCAL 資格情報はどのドメインにも格納できます。実際、リモートドメインにログインし認証されるためには、クライアントユーザーはリモートドメインの cred テーブルに LOCAL 資格を持っている必要があります。ユーザーが、アクセスしようとするリモートドメインに LOCAL 資格を持っていなかった場合、NIS+ はユーザーの DES 資格を獲得するためにユーザーのホームドメインに入ることができなくなります。そのような場合、ユーザーは認証されず未認証クラスを与えられることとなります。

ユーザー種類と資格種類

ユーザーは両方の資格を持つことができますが、マシンは DES 資格しか持ってません。

すべてのマシンのルート UID は常に 0 であるため、ルートはほかのマシンに対して NIS+ アクセスできません。もしマシン A のルート (UID=0) がマシン B にアクセスしようとする、すでに存在しているマシン B のルート (UID=0) と衝突します。このため、LOCAL 資格は、クライアント「マシン」については意味を持たないため、クライアント「ユーザー」にだけ認められています。

表 11-2 資格のタイプ

資格のタイプ	クライアントユーザー	クライアントマシン
DES	有効	有効
LOCAL	有効	無効

NIS+ の承認とアクセス - 紹介

NIS+ の承認の基本的目的は、各 NIS+ 主体が各 NIS+ オブジェクトとサービスに対して持っているアクセス権を特定することにあります。

一度主体の NIS+ 要求が認証されると、NIS+ は承認クラスを与えます。NIS+ オブジェクトに対して行うことのできる動作を指定するアクセス権は各クラスに与えられます。クラスごとに別のアクセス権が与えられる場合、各承認クラスはそれぞれアクセス権を持つということです。

- 「承認クラス」。「承認クラス」には次の 4 つがあります。所有者、グループ、その他、未認証 (詳細は、次の [221 ページの「承認クラス」](#) を参照)。
- 「アクセス権」。「アクセス権」には、作成、削除、変更、読み取りの 4 つがあります (詳細は、[224 ページの「NIS+ アクセス権について」](#) を参照)。

承認クラス

NIS+ オブジェクトは NIS+ 主体に直接アクセス権を与えずに、主体の 4 つの「クラス」にアクセス権を与えます。

- 「所有者」。オブジェクトの所有者になった主体が所有者クラスの権限を与えられます。
- 「グループ」。各 NIS+ オブジェクトグループは関連した 1 つのグループを持ちます。オブジェクトグループのメンバーは NIS+ 管理機能が指定します。オブジェクトグループクラスに属している主体がグループクラスの権限を与えられます。この場合の「グループ」とは、UNIX のグループやネットグループのことではありません。
- 「その他」。その他クラスは、サーバーが認証できるすべての NIS+ 主体を包含するクラスです。所有者とグループクラスを除く、すべての認証されたものを意味します。
- 「未認証」。認証されていないもののすべてを意味します。

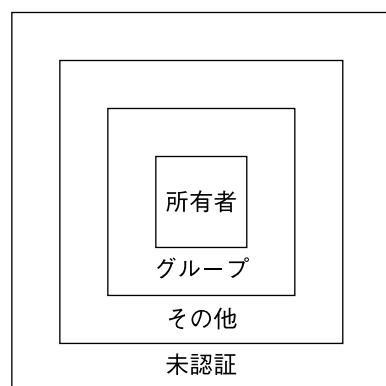


図 11-4 承認クラス

システムは、NIS+ 要求を受信すると、要求を出している主体が属しているクラスを判定します。主体は、判定されたクラスに属しているすべてのアクセス権を使用できます。

オブジェクトに対するアクセス権は、各クラスで任意の組み合わせが可能です。しかしながら、通常は上位クラスが下位クラスのすべての権限に追加権限を持つように割り当てられます。

たとえば、未認証クラスとその他クラスに読み取り権が与えられていれば、読み取り権と変更権の両方をグループクラスに、そして読み取り権、変更権、作成権、および削除権を所有者クラスに与えます。

4つのクラスについては、以下に詳述します。

所有者クラス

所有者は「単一」の NIS+ 主体です。

NIS+ オブジェクトへのアクセスを要求する主体は、所有者アクセス権を得る前に認証され (有効な DES 資格を提示し) なければなりません。

デフォルトではオブジェクトの所有者は、オブジェクトを作成した主体になります。しかしながら、オブジェクト所有者は2つの方法で別の主体に所有権を譲ることができます。

- オブジェクトが作成されたときに、主体が別の所有者を指定する方法 (277 ページの「コマンドによるアクセス権の指定」を参照)
- オブジェクトの作成後に、主体がオブジェクトの所有権を変更する方法 (286 ページの「オブジェクトとエントリの所有権の変更」を参照)

一度主体が所有権を放棄すると、その主体はオブジェクトに対する所有者のアクセス権すべてを放棄することになり、グループ、その他、未認証のいずれかの所有権を持つこととなります。

グループクラス

オブジェクトグループは1つの NIS+ グループです。この場合の「グループ」とは、UNIX のグループやネットグループのことではありません。

NIS+ オブジェクトにアクセスを要求する主体は、認証され (有効な DES 資格を提示し) かつグループアクセス権を与えられる前にそのグループに属していなければなりません。

1つの NIS+ グループは NIS+ 主体の集合であり、名前空間へのアクセス権を与えるのに都合の良いようにまとめられたものです。NIS+ グループに与えられるアクセス権はそのグループのメンバーである主体すべてに適用されます。オブジェクトの所有者が、そのオブジェクトのグループに属している必要はありません。

オブジェクトが作成された時、そのオブジェクトはデフォルトグループに割り当てられます。オブジェクトが作成された時でもその後も、そのオブジェクトをデフォルト以外のグループに割り当てることができます。オブジェクトグループはいつでも変更できます。

注 - NIS+ グループに関する情報は NIS+ group テーブルに格納されるのではないことに注意してください。NIS+ group テーブルには UNIX グループに関する情報が格納されます。NIS+ グループに関する情報は、適切な groups_dir ディレクトリのオブジェクトに格納されます。

NIS+ グループの情報は、各 NIS+ ドメインの groups_dir サブディレクトリにある、NIS+ グループ「オブジェクト」に格納されます。

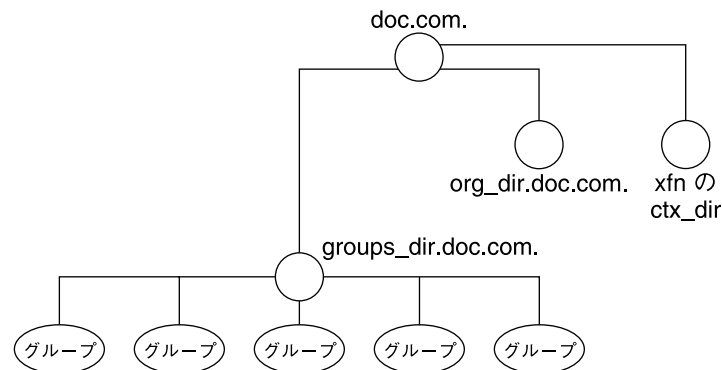


図 11-5 NIS+のディレクトリ構造

NIS+ グループの管理については、[第 17 章](#)を参照してください。

その他クラス

その他クラスは、NIS+ によって認証されたすべての NIS+ 主体のクラスです。すなわち、所有者およびグループクラスのすべてと有効な DES 資格を提示したものを加えたものです。

その他に与えられたアクセス権は、認証されたすべての主体に適用されます。

未認証クラス

未認証クラスは、正しく認証されなかったものすべてで構成されます。すなわち、有効な DES 資格を提示しなかったものです。

承認クラスと NIS+ オブジェクトの階層

NIS+ オブジェクトと承認クラスには各レベルに適用される階層があります。標準的な NIS+ ディレクトリ階層のデフォルトは次のようになります。

- 「ディレクトリレベル」。各 NIS+ ドメインには、groups_dir と org_dir という 2 つの NIS+ ディレクトリオブジェクトがあります。groups_dir ディレクトリオブジェクトはさまざまなグループで構成されます。各 org_dir ディレクトリオブジェクトには種々のテーブルが含まれます。
- 「グループレベルまたはテーブルレベル」。グループは個々のエントリやグループで構成されます。テーブルには列 と個々のエントリが含まれます。
- 「列レベル」。テーブルは 1 つ以上の列で構成されます。
- 「エントリ (行) レベル」。グループもしくはテーブルは 1 つ以上のエントリを持ちます。

各レベルには 4 つの承認クラスが適用されます。ディレクトリオブジェクトはそれ自身の所有者とグループを持ちます。ディレクトリ内の個々のテーブルは、ディレクトリの所有者およびグループと異なる所有者およびグループを持ちます。テーブル内では、1 エントリ (行) が個々の所有者もしくはグループを持ちます。この所有者もしくはグループは、テーブル全体の所有者やグループとは異なり、またオブジェクト全体の所有者やグループとも異なります。テーブル内の個々の列は、テーブル全体と同じ所有者やグループを持ちます。

NIS+ アクセス権について

NIS+ オブジェクトのオブジェクト定義には、オブジェクトのアクセス権を指定します。オブジェクト定義は、niscat -o コマンドで確認できます。

NIS+ オブジェクトは、UNIX ファイルが UNIX ユーザーに対するアクセス権を指定するのと同じ方法で、NIS+ 主体に対するアクセス権を指定します。アクセス権は、NIS+ 主体が NIS+ オブジェクトについて実行することが許されている動作の種類を表します。

NIS+ の動作はオブジェクトの種類によって異なりますが、読み取り、変更、作成、および削除の 4 つのアクセス権に分類されます。

- 「読み取り」。オブジェクトの読み取り権を持った主体はオブジェクトの内容を読み取ることができます。
- 「変更」。オブジェクトに対する変更権を持った主体はオブジェクトの内容を変更できます。
- 「削除」。オブジェクトの削除権を持った主体はオブジェクトの内容を削除できます。
- 「作成」。上位レベルのオブジェクトに対する作成権を持った主体が、そのレベルの下位に新規のオブジェクトを作成できます。すなわち、NIS+ ディレクトリオブジェクトの作成権を持っていれば、そのディレクトリ内に新しいテーブルを作成できます。NIS+ テーブルの作成権を持っていれば、そのテーブル内に新しい列とエントリを作成できます。

NIS+ クライアントから NIS+ サーバーへのすべての通信は、実際には特定の NIS+ オブジェクトに対してこれらの動作のうちの 1 つを実行してほしいという要求です。たとえば、NIS+ 主体がほかのマシンの IP アドレスを要求した場合、これは実際には、

この種の情報を格納している「hosts」テーブルオブジェクトへの読み取り権を要求しています。主体がNIS+ 名前空間にディレクトリを追加するようサーバーに要求した場合、これは実際には、そのディレクトリの親オブジェクトに対して変更権を要求しています。

これらの権限は論理的には、ディレクトリ、テーブル、列とエントリのように下位に展開するものであることを銘記してください。たとえば、新規にテーブルを作成するには、そのテーブルを格納するNIS+ ディレクトリオブジェクトに対する作成権が必要です。そのテーブルを作成した場合、そのデフォルト所有者になります。所有者として、自分自身にそのテーブルに対する作成権を与え、テーブルに新規のエントリを作成できます。テーブル内に新規のエントリを作成した場合、それらのエントリの所有者になります。所有者として、ほかのユーザーにテーブルレベルの作成権を与えることもできます。たとえば、テーブルのグループクラスにテーブルレベルの作成権を与えることができます。その場合、テーブルのグループのすべてのメンバーがテーブル内に新規のエントリを作成できます。新規のテーブルエントリを作成した、グループの個々のメンバーはそのエントリのデフォルト所有者になります。

NIS+ 管理者

NIS+管理者は、NIS+オブジェクトに対して「管理者権限」を持つユーザーです。管理者権限は、オブジェクトに対する作成および削除権限で構成されます。また、オブジェクトによっては変更権限が含まれる場合もあります (NIS+ アクセス権については、[224 ページの「NIS+ アクセス権について」](#)を参照してください)。

NIS+ オブジェクトを作成するものはだれでもそのオブジェクトに対する初期アクセス権を持つと設定します。もし作成者が管理権限をオブジェクトの所有者に限った場合 (初期状態では作成者) は、所有者だけがそのオブジェクトに対する管理権限を持ちます。一方、作成者が管理権限をオブジェクトのグループに与えた場合は、グループの全員がそのオブジェクトに対して管理権限を持つことになります。

あるオブジェクトに対して管理権限を持つものはだれでもそのオブジェクトの NIS+ 管理者とみなされます。

すなわち、NIS+ ソフトウェアは NIS+ 管理者を 1 人にしようとするものではないということです。

理論的には、管理権限をその他クラスに与えることも、未認証クラスに与えることもできます。これは、ソフトウェア上で実行可能です。しかし、管理権限をグループクラス以上に広げて与えてしまうと、NIS+ のセキュリティを実質的に無効にしています。もし管理権限をその他クラスや未認証クラスに与えた場合、NIS+ のセキュリティは保証されません。

NIS+ のパスワード、資格、およびコマンド

注 - NIS+ サービスは、サービス管理機能によって管理されます。このサービスに対する有効化、無効化、再起動などの管理操作を実行するには、svcadm コマンドを使用します。この機能を NIS+ で使用する方法については、89 ページの「NIS+ とサービス管理機能」を参照してください。この機能の概要については、『Solaris のシステム管理 (基本編)』の「サービスの管理 (概要)」を参照してください。詳細については、svcadm(1M) と svcs(1) の各マニュアルページも参照してください。

管理者のパスワード、資格、および鍵には次のコマンドを使用します。各コマンドの説明についてはマニュアルを参照してください。

- chkey - 主体の Secure RPC 鍵の組み合わせを変更します。必要のない場合は chkey を使わずに、passwd を使ってください。詳細については、251 ページの「NIS+ 主体の鍵の変更」を参照してください。
- keylogin - keyserv を使って主体の非公開鍵を復号化し格納します。
- keylogout - 格納されている非公開鍵を keyserv から削除します。
- keyserv - サーバーに非公開暗号鍵を格納させます。鍵の詳細については、250 ページの「NIS+ を使用したキーログイン」を参照してください。サービス管理機能を使用して NIS+ を管理する方法については、89 ページの「NIS+ とサービス管理機能」を参照してください。
- newkey - 公開鍵データベースに新規の鍵の組み合わせを作成します。
- nisaddcred - NIS+ 主体に資格を作成します。詳細は、239 ページの「資格情報の作成」および、247 ページの「NIS+ 資格情報の管理」を参照してください。
- nisauthconf - Diffie-Hellman 鍵の長さの表示あるいは設定を行います。
- nisupdkeys - ディレクトリオブジェクト内の公開鍵を更新します。詳細については、255 ページの「公開鍵の更新」を参照してください。
- passwd - 主体のパスワードを変更し管理します。詳細については、第 16 章を参照してください。

第 12 章

NIS+ 資格の管理

この章では、NIS+ 資格とその管理方法について説明します。

ヒント - NIS+ セキュリティタスクには、Solaris 管理コンソール ツールがあればより簡単に実行できるものがあります。

注 - NIS+ は、将来のリリースではサポートされなくなる可能性があります。NIS+ から LDAP への移行支援ツールは、Solaris 9 リリース以降で使用できます (『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』を参照)。詳細については、<http://www.sun.com/directory/nisplus/transition.html> を参照してください。

NIS+ 資格について

NIS+ 資格は、NIS+ のユーザーを識別するために使用します。この章では、NIS+ セキュリティシステム全体と、特にシステムで資格が果たす役割について、十分理解しているユーザーを想定しています。

これらのタスクを実行するのに使われるコマンドの構文、オプションなどの詳細については、NIS+ のマニュアルを参照してください。

注 - この章の DES 資格の説明は、192 ビット Diffie - Hellman DES 資格に適用することができます。これらは似てはいますが、その他の鍵の長さを使用した認証は細部では異なっています。コマンド行インタフェースを使用して鍵を操作するときは、その違いはユーザーにもシステム管理者にも意識されることはありません。指定した鍵の長さの表示や設定は `nisauthconf (1M)` を使って行います。

資格の機能

注 - この章では、NIS+ のコマンド一式を使用してクライアント情報を作成する方法について説明します。

資格および認証システムは他人になりすますのを防ぐシステムです。あるマシンの root 権限で `su` コマンドを使って、ログインしていないもしくは別のマシンにログインしている第 2 のユーザーになりすまし、第 2 のユーザーの NIS+ アクセス権を使用して NIS+ オブジェクトにアクセスすることを防ぎます。



注意 - NIS+ には、他人のユーザーログインパスワードを知っている者が、そのユーザーになりすましてユーザー ID と NIS+ アクセス権を使用することを妨げることはできません。また root 権限を持つユーザーが「同一」マシンからログインしている別のユーザーになりすますのを防ぐこともできません。

資格と資格情報

DES 資格をどのように作成しそれがどのように機能するのかを理解するには、実際の資格と資格を作成しチェックするのに使われる情報とを区別する必要があります。

- 「資格情報」。DES 資格を作成するのに使われるデータであり、サーバーが資格をチェックするのに使われる
- 「DES 資格」。主体を認証するために、主体からサーバーへ渡される一連の数字。主体が NIS+ 要求を行うたびに作成されチェックされる。DES 資格については、[232 ページの「DES 資格の詳細」](#)を参照してください。

認証

資格および認証プロセスを機能させるには、次の要素が必要です。

- 「主体の DES 資格情報」。この情報は各主体に NIS+ 管理者が最初に作成します。この情報は主体のホームドメインの cred テーブルに格納されます。主体の DES 資格情報は次のもので構成されます。
 - 「主体名」。ユーザーの完全ログイン ID もしくはマシンの完全ホスト名
 - 「主体の Secure RPC ネット名」。各主体は固有の Secure RPC ネット名を持っています (Secure RPC ネット名の詳細については、[232 ページの「DES 資格の Secure RPC ネット名」](#)を参照)。
 - 「主体の公開鍵」。
 - 「主体の非公開暗号鍵」。
- 「主体の LOCAL 資格」。
- 「サーバーの公開鍵」。各ディレクトリオブジェクトには、当該ドメインのすべてのサーバーの公開鍵の複製が格納されています。cred テーブルには各サーバーの DES 資格も格納されていることに注意してください。
- 「主体の公開鍵のキーサーバーの複製」。キーサーバーは、その時点でログインしている主体 (ユーザーまたはマシン) の公開鍵の複製を持っています。

主体の認証方法

承認プロセスには 3 つのフェーズがあります。

- 「準備フェーズ」。NIS+ 管理者がユーザーのログインに先立って行う設定作業 (ユーザーの資格情報の作成など)
- 「ログインフェーズ」。ユーザーがログインした時にシステムが行う動作
- 「要求フェーズ」。NIS+ 主体が、NIS+ サービスもしくは NIS+ オブジェクトに要求を発する時にソフトウェアが行う動作

これら 3 つのフェーズの詳細は、次の項目を参照してください。

資格準備フェーズ

ユーザーの資格情報は、nisclient スクリプトを使用すればもっとも簡単に作成できます。この節では、NIS+ のコマンド一式を使用してクライアント情報を作成する方法について説明します。

NIS+ 主体がログインする前に、NIS+ 管理者は当該主体 (ユーザーまたはマシン) のための資格情報を作成する必要があります。管理者は次のようにする必要があります。

- 各主体の公開鍵と非公開暗号鍵を作成します。これらの鍵は主体のホームドメインの cred テーブルに格納されます。これは nisaddcred コマンドを使って行われます ([243 ページの「NIS+ 主体の資格情報を作成する方法」](#)を参照)。
- サーバーの公開鍵を作成します ([255 ページの「公開鍵の更新」](#)を参照)。

ログインフェーズ - 詳細

主体がシステムにログインする時、次のステップが自動的に実行されます。

1. 主体のために `keylogin` プログラムが起動します。`keylogin` プログラムは `cred` テーブルから主体の非公開暗号鍵を取得し、主体のログインパスワードを使って復号化します。

注 - 主体のログインパスワードが `Secure RPC` パスワードと異なる場合、`keylogin` コマンドはパスワードを復号化できず、「cannot decrypt」などのエラーとなるか、メッセージなしでコマンドが失敗します。この問題の解説は、[235 ページの「Secure RPC パスワードとログインパスワードの問題」](#)を参照してください。

2. 復号化された主体の非公開暗号鍵は、要求フェーズでの利用に備えてキーサーバーに渡され格納されます。

注 - 復号化された非公開暗号鍵はユーザーが `keylogout` コマンドを実行するまでキーサーバーに格納されます。ユーザーが `keylogout` を実行せずにログアウトした (またはログアウトせずに帰宅してしまった) 場合には、復号化された非公開鍵がサーバーに格納されたままになっています。もしユーザーマシンの `root` 権限を持っただれかがユーザーログイン ID に `su` コマンドを使った場合、その人間はユーザーの復号化された非公開鍵を使用でき、ユーザーのアクセス承認を使って `NIS+` オブジェクトにアクセスできるようになります。このような事態を避けるため、ユーザーが業務を終了する時は確実に `keylogout` コマンドを使ってログアウトするように注意する必要があります。もしユーザーがシステムからログアウトもした場合は、業務に戻る時はログインし直すだけですみます。ユーザーが確実にログアウトしなかった場合は、業務に戻る時に確実に `keylogin` を実行する必要があります。

要求フェーズ - 詳細説明

`NIS+` 主体の要求が `NIS+` オブジェクトにアクセスするつど、その主体を認証するために `NIS+` ソフトウェアは複数段階のプロセスを実行します。

1. `NIS+` はオブジェクトドメインの `cred` テーブルをチェックします。
 - 主体が `LOCAL` 資格情報を持っていた場合、`LOCAL` 資格にあるドメイン情報を使い、主体のホームドメイン `cred` テーブルを見つけます。ここでは必要な情報を獲得します。
 - 主体が資格情報を持っていなかった場合、残りのプロセスは実行されず、主体の承認アクセスクラスには未認証クラスが与えられます。
2. ユーザーのホームドメインの `cred` テーブルから、`NIS+` がユーザーの `DES` 資格を取得します。ユーザーパスワードを使って非公開暗号鍵が復号化され、キーサーバーに格納されます。

3. NIS+ が NIS+ ディレクトリオブジェクトからサーバーの公開鍵を獲得します。
4. キーサーバーが、主体の復号化された非公開鍵とオブジェクトのサーバー (オブジェクトが格納されているサーバー) の公開鍵を使って、「共通鍵」を作成します。
5. 共通鍵を使って暗号化された「DES 鍵」が作成されます。これは、Secure RPC が乱数を発生させ、これを共通鍵を使って暗号化することで行われます。このため、DES 鍵は「乱数鍵」もしくは「乱数 DES 鍵」として参照される場合があります。
6. NIS+ が主体のサーバーの時刻情報に基づいてタイムスタンプ (DES 鍵を使って暗号化される) を作成します。
7. NIS+ が 15 秒のタイムウィンドウ (DES 鍵を使って暗号化される) を作成します。この「ウィンドウ」はタイムスタンプとサーバーの内部クロックとの間で許容される最長時間になります。
8. NIS+ が主体の DES 資格を作成します。これは次のもので構成されます。
 - 主体の cred テーブルに基づく主体の Secure RPC ネット名 (`unix.identifier@domain`)
 - キーサーバーから得た暗号化された主体の DES 鍵
 - 暗号化されたタイムスタンプ
 - 暗号化されたタイムウィンドウ
9. NIS+ が NIS+ オブジェクトの格納されているサーバーに次の情報を渡します。
 - アクセス要求
 - 主体の DES 資格
 - ウィンドウ判定子 (暗号化済)。ウィンドウ判定子は暗号化されたウィンドウに 1 を加えたもの
10. オブジェクトのサーバーがこの情報を受信します。
11. オブジェクトのサーバーが、資格の中の Secure RPC ネット名の一部を使って、主体のホームドメインにある cred テーブル内にある主体の公開鍵をチェックします
12. サーバーが主体の公開鍵とサーバーの非公開鍵を使ってもう一度共通鍵を作成します。この共通鍵は主体の非公開鍵とサーバーの公開鍵を使って作成されている共通鍵と一致する必要があります。
13. 共通鍵を使って、主体の資格の一部として受信している DES 鍵を暗号化します。
14. サーバーが、新しく復号化した DES 鍵を使って主体のタイムスタンプを復号化し、ウィンドウ判定子を使ってそれをチェックします。
15. その後、サーバーは、復号化しチェックしたタイムスタンプと自分の内部時計とを比較し、次のプロセスを実行します。
 - a. サーバーとの時間差がウィンドウ許容値を越えていた場合、要求は拒否され、プロセスが中止されて、エラーメッセージが表示されます。たとえば、タイムスタンプが 9:00 am であり、ウィンドウが 1 分であったとします。もしその要求をサーバーが受信して復号化したのが 9:01 am であれば、その要求は拒否されます。

- b. タイムスタンプがウィンドウ許容値内である場合、サーバーは主体から前に受信した要求のタイムスタンプよりも大きいかわかるチェックします。これによって、NIS+ 要求を正しい順序で処理しているか確認できます。
- 順序誤りの要求はエラーメッセージとともに拒否されます。たとえば、タイムスタンプが 9:00 am でありその主体から最後に受信した要求のタイムスタンプが 9:02 am だった場合、その要求は拒否されます。
 - 最後に受信した要求と同じタイムスタンプであった場合もエラーメッセージが表示されて拒否されます。これによって、要求を 2 度処理することはありません。たとえば、タイムスタンプが 9:00 am であり最後にその主体から受信した要求のタイムスタンプも 9:00 am だった場合、この要求は拒否されます。
16. タイムスタンプがウィンドウ許容値内であり、その主体からの最後の要求よりも大きかった場合に、サーバーはその要求を受け入れます。
17. サーバーが、要求をコンパイルし、タイムスタンプをその主体から受信した最後のものとして格納し、要求に基づいて動作します。
18. 要求に対する応答としてサーバーから受信した情報が信頼できるサーバーからのものであるかを主体が確認できるようにするため、サーバーは主体の DES 鍵を使ってタイムスタンプを暗号化し、データとともに主体に送り返します。
19. 主体側では、主体の DES 鍵を使って送り返されたタイムスタンプを復号化します。
- 復号化が成功した場合は、サーバーからの情報を要求者へ戻します。
 - 何らかの理由で復号化に失敗した場合は、エラーメッセージが表示されます。

DES 資格の詳細

DES 資格は次の内容で構成されます。

- 主体の「Secure RPC ネット名」(232 ページの「DES 資格の Secure RPC ネット名」を参照)
- 「確認 (verification)」フィールド (233 ページの「DES 資格確認フィールド」を参照)

DES 資格の Secure RPC ネット名

「Secure RPC ネット名」資格のこの部分は NIS+ の主体を特定するのに使用します。Secure RPC ネット名はすべて次の 3 つの要素で構成されます。

- 「接頭辞」。接頭辞は常に unix になります。
- 「識別子」。主体がクライアントユーザーの場合、ID フィールドはユーザーの UID です。主体がクライアントマシンの場合、ID フィールドはマシンのホスト名です。

- 「ドメイン名」。主体の DES 資格を含んでいるドメイン名がドメイン名になります (主体のホームドメイン)。

注 – NIS+ 主体名は常にドット (.) で終わりますが、Secure RPC ネット名はドット (.) で終わらないことに注意してください。

表 12-1 Secure RPC ネット名のフォーマット

主体	接頭辞	識別子	ドメイン	コメント例
ユーザー	unix	UID	ユーザーのパスワードエントリと DES 資格そのもののあるドメイン	unix.24601@sales.doc.com
マシン	unix	ホスト名	ドメイン名はマシンが domainname コマンドを実行すると返される	unix.machine7@sales.doc.com

DES 資格確認フィールド

確認フィールドは資格を確かめるために使用するフィールドです。cred テーブルに格納された資格情報に基づいて作成されます。

確認フィールドの構成は次のとおりです。

- 主体の非公開鍵と NIS+ サーバーの公開鍵に基づく、暗号化された主体の DES 鍵 (230 ページの「要求フェーズ - 詳細説明」参照)
- 暗号化されたタイムスタンプ
- タイムウィンドウ

DES 資格の作成方法

DES 資格を作成するには、keylogin コマンドを実行する必要がありますが、これは主体が資格を作成する前に実行します。keylogin コマンド (単に「keylogin」とする場合が多い) は、NIS+ 主体がログインする時に自動的に実行されます。図 12-2 を参照してください。

注 – 主体のログインパスワードが Secure RPC パスワードと異なる場合は、keylogin コマンドは成功しないことに注意してください。詳しくは、[235 ページの「Secure RPC パスワードとログインパスワードの問題」](#)を参照してください。

keylogin コマンドの目的は、主体が自身の非公開鍵にアクセスできるようにすることにあります。keylogin コマンドを実行すると、cred テーブルから主体の非公開鍵を取り込み (fetch)、主体の Secure RPC パスワード (非公開鍵は主体の「Secure RPC パスワード」を使って最初に暗号化される) を使ってそれを復号化し、将来の NIS+ 要求に備えてキーサーバーに格納します。

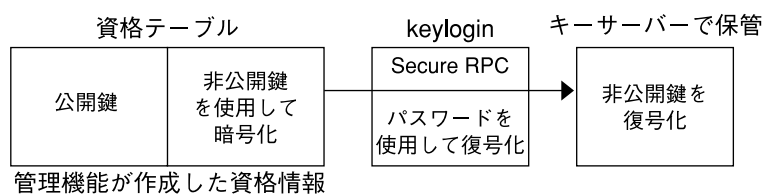


図 12-1 keylogin コマンドで主体の非公開鍵を作成

主体が DES 資格を作成するには、主体が要求を送る相手サーバーの公開鍵が必要です。この情報は主体のディレクトリオブジェクトに格納されています。主体がこの情報を獲得すれば資格の確認フィールドを作成できます。

まず、主体が種々の資格情報を暗号化するためにランダム DES 鍵を作成します。主体は自分の非公開鍵 (キーサーバーに格納されている) とサーバーの公開鍵を使って、ランダム DES 鍵を作成し暗号化するための共通鍵を作成します。次に DES 鍵で暗号化したタイムスタンプを作成し、それを確認フィールドでほかの資格関連情報の中に入れます。

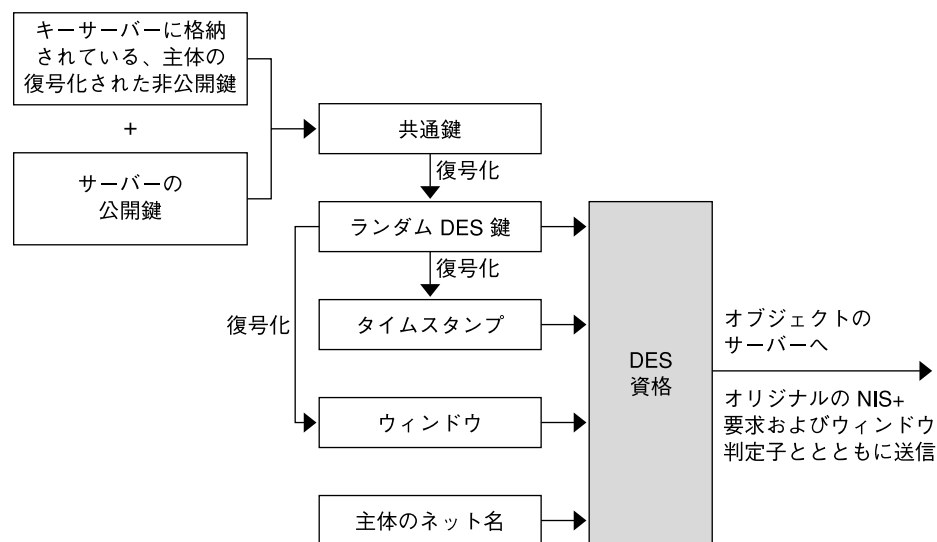


図 12-2 DES 資格の作成

Secure RPC パスワードとログインパスワードの問題

主体のログインパスワードと Secure RPC パスワードが一致しないと、ログイン時に keylogin はパスワードを復号化できません。keylogin はデフォルトで主体のログインパスワードを使用することになっていて、また非公開鍵は主体の Secure RPC パスワードを使用して暗号化されているからです。

この場合でも主体がシステムにログインすることは可能ですが、キーサーバーがそのユーザーのための復号化された非公開鍵を持っていませんから、未認証クラスが与えられることになります。ほとんどの NIS+ 環境において、未認証クラスにはほとんどの NIS+ オブジェクトに対する作成権、削除権、変更権を与えないように設定されますから、結果としてユーザーが NIS+ オブジェクトにアクセスしても「アクセス権拒否」などのエラーになります。

注 - 「ネットワークパスワード」と「Secure-RPC パスワード」を同意語として使用する場合があります。ネットワークパスワードの入力を求められたら Secure RPC パスワードを入力します。

別の承認クラスを得るには、この場合 keylogin プログラムを実行し、パスワード入力のプロンプトに主体の Secure RPC パスワードを入力する必要があります (250 ページの「NIS+ を使用したキーログイン」を参照)。

しかし `keylogin` を実行しても、現在のログインセッション以外には無効で、一時的な解決にしかありません。キーサーバーはこの方法によって、復号化された形でユーザーの非公開鍵を持つようになるのですが、ユーザーの `cred` テーブル中の非公開鍵が Secure RPC パスワード (ユーザーのログインパスワードとは異なっている) を使用して暗号化されているという点に変わりはないからです。次にログインした時には同じ問題が発生します。この問題を永久的に解決するには、ユーザーの Secure RPC パスワードではなくユーザーログイン ID を使って、`cred` テーブルにある非公開鍵を再度暗号化する必要があります。このためにはユーザーが `chkey -p` コマンドを実行します (251 ページの「NIS+ 主体の鍵の変更」の項を参照)。

Secure RPC パスワードがログインパスワードと異なる問題を永久的に解決するには、ユーザー (もしくはユーザーに対応している管理者) が次の手順を実行してください。

1. ログインパスワードを使用してログインします。
2. `keylogin` プログラムを実行して、キーサーバーに保存される非公開鍵を一時的に復号し、一時的な NIS+ アクセス権を得ます。
3. `chkey -p` を実行し、`cred` テーブル内の暗号化された非公開鍵を、ユーザーログインパスワードに基づいたものに永久的に変更します。
4. ログインセッションの終了の準備ができれば、システムをログオフする前に `keylogout` を実行します。
5. `logout` コマンドを使って、システムからログオフします。

キャッシュに保存された非公開鍵の問題

正しい資格を作成し、正しいアクセス権を与えられたのに、主体の要求が拒否される場合があります。この原因のほとんどは、サーバーの公開鍵が古いバージョンだった時に作られた、古いオブジェクトが残っていることにあります。この問題は通常次の方法で解決します。

- アクセスしようとしているドメインで、`nisupdkeys` を実行します (`nisupdkeys` コマンドの使用法については 255 ページの「`nisupdkeys` コマンド」を、この種の問題の解決方法については 448 ページの「資格情報が古い」を参照)。
- マシン上で `svcadm` コマンドを使用して NIS+ サービスを停止します。次に、`/var/nis/NIS_SHARED_DIRCACHE` を削除してから、NIS+ サービスを再起動します。

資格に関連する情報が格納されている場所

この節では、NIS+ 名前空間のどこに資格関連情報が格納されるのかを説明します。

公開鍵など、資格に関する情報は、名前空間内の様々な場所に保存されています。NIS+ は、情報を格納しているオブジェクトの生存期間に応じてこの情報を定期的に更新しますが、場合によっては、更新の間に同期が失われ、システムが正常に動作しなくなることがあります。資格関連の情報を格納するすべてのオブジェクト、テーブル、ファイル、および再設定方法を、表 12-2 に示します。

注 – NIS+ サービスは、サービス管理機能 (SMF) によって管理されます。rpc.nisd、keyserv、nis_cachemgr などの NIS+ デーモンを有効化、無効化、または再起動するには、svcadm コマンドを使用します。SMF を NIS+ で使用方法については、89 ページの「NIS+ とサービス管理機能」を参照してください。SMF の概要については、『Solaris のシステム管理 (基本編)』の「サービスの管理 (概要)」を参照してください。詳細については、svcadm(1M) と svc(1) の各マニュアルページも参照してください。

表 12-2 資格に関連する情報が格納されている場所

項目	保存対象	リセットおよび更新の方法
cred テーブル	NIS+ 主体の秘密非公開鍵と公開鍵。これらの鍵のマスターコピーとなる	nisaddcred を使用して新しい資格を作成する。これによって既存の資格が更新されるchkey を使用しても同様のことが行える
ディレクトリオブジェクト	個々のサーバーの公開鍵のコピー	ディレクトリオブジェクトに対して /usr/lib/nis/nisupdkeys コマンドを実行する
キーサーバー	その時点でログインされている NIS+ 主体の非公開鍵	主体ユーザーに対して keylogin を実行する。または主体マシンに対して keylogin -r を実行する
NIS+ デーモン	ディレクトリオブジェクトのコピー (そのサーバーの公開鍵のコピーが含まれる)	NIS+ サービスを無効にすることにより rpc.nisd デーモンとキャッシュマネージャを停止し、/var/nis から NIS_SHARED_DIRCACHE を削除する。その後、NIS+ サービスを再起動する
ディレクトリキャッシュ	ディレクトリオブジェクトのコピー (そのサーバーの公開鍵のコピーが含まれる)	-i オプションを指定して NIS+ キャッシュマネージャを再起動する
コールドスタートファイル	ディレクトリオブジェクトのコピー (そのサーバーの公開鍵のコピーが含まれる)	NIS+ サービスを停止する。/var/nis から NIS_COLD_START ファイルと NIS_SHARED_DIRCACHE ファイルを削除する。NIS+ サービスを再起動する

表 12-2 資格に関連する情報が格納されている場所 (続き)

項目	保存対象	リセットおよび更新の方法
passwd テーブル	ユーザーのパスワード	passwd -r nisplus コマンドを使用 するこれによって、NIS+ passwd テー ブル、cred テーブルの中でパスワー ドが更新される
passwd ファイル	ユーザーのパスワードまたはマシ ンのスーパーユーザーのパスワー ド	passwd -r nisplus コマンドを使用 する。スーパーユーザーとしてログイ ンしていても、自分のユーザー名でロ グインしていてもよい
passwd マップ (NIS)	ユーザーのパスワード	passwd -r nisplus コマンドを使用 する

cred テーブルの詳細

主体の資格情報は「cred テーブル」に格納されています。cred テーブルは NIS+ が標準で持つ 16 のテーブルの 1 つです。cred テーブルは 1 ドメインにつき 1 つ存在し、ドメインに属するクライアントマシンおよびマシンにログインできるユーザー (つまりドメインの主体) の資格に関する情報を持っています。cred テーブルはドメイン中の org_dir サブディレクトリにあります。



注意 - cred テーブルをリンクしないでください。各 org_dir ディレクトリはそれ自身の cred テーブルを持つ必要があります。ほかの org_dir の cred テーブルとのリンクも使用しません。

ドメイン内のすべてのマシンにログインできるユーザーすべての LOCAL 資格情報が cred テーブルに格納されています。cred テーブルにはまた、ホームドメインとしてドメインを持っているユーザーの DES 資格情報も格納されています。

cred テーブルの内容は niscat コマンドを使って見ることができます。第 19 章を参照してください。

表 12-3 に示すように、cred テーブルには 5 つの列があります。

表 12-3 cred テーブルの資格情報

	NIS+ 主体名	認証タイプ	認証名	公開データ	非公開データ
列名	cname	auth_type	auth_name	public_data	private_data

表 12-3 cred テーブルの資格情報 (続き)

	NIS+ 主体名	認証タイプ	認証名	公開データ	非公開データ
ユーザー	完全主体名	LOCAL	UID	グループ ID リスト	
マシン	完全主体名	DES	Secure RPC ネット名	公開鍵	暗号化された非公開鍵

2 列目の認証の種類ではかの 4 列の値の種類を判定します。

- 「LOCAL」。認証タイプが LOCAL の場合、残りの列には主体ユーザー名、UID、GID が入ります (最後の列は空白)。
- 「DES」。認証タイプが DES の場合、残りの列には主体名、Secure RPC ネット名、公開鍵、暗号化非公開鍵が入ります。これらの鍵はほかの情報と組み合わせられ、DES 資格の暗号化および復号化に使用されます。

資格情報の作成

資格情報を作成したり管理したりするには次の方法があります。

- Solaris 管理コンソール ツールを使用します。このツールを使用すると容易に資格を管理できます。個々の資格を管理する場合はこのツールの使用を推奨します。
- nisclient スクリプトを使用します。1 つの主体の資格を容易に作成し変更できます。便利な方法なので、個々の資格を管理する方法として推奨します。本書では nisclient スクリプトの使用方法を手順ごとに示しながら、資格情報を作成していきます。
- nispopulate スクリプトを使用します。これはすでに NIS+ マップもしくは /etc ファイルに資格情報を格納してある 1 つ以上の主体の資格を作成変更する便利な方法です。これは、NIS+ 主体グループの資格を管理する方法として推奨します。本書では nispopulate スクリプトの使用方法を手順ごとに示しながら、資格情報を作成していきます。100 ページの「NIS+ テーブルの生成 (populate)」を参照してください。
- nisaddcred コマンドを使用します。次の節では、nisaddcred コマンドを使って資格と資格情報を作成する方法を説明します。

nisaddcred コマンド

nisaddcred コマンドは資格情報を作成するコマンドです。

注 -nispopulate と nisclient スクリプトを使って資格情報を作成できます。これらも nisaddcred コマンドを使用します。これらのスクリプトは、nisaddcred コマンドよりも容易かつ効率的に使えます。ネットワークが特別な機能を必要とするのでなければ、スクリプトを使用してください。

nisaddcred コマンドは、LOCAL 資格と DES 資格情報の作成、更新、および削除を行います。資格情報を作成するには、適切なドメインの cred テーブルに対する作成権が必要です。資格を更新するには、cred テーブル、または少なくとも cred テーブルの特定エントリに対する変更権が必要です。資格を削除するには、cred テーブル、または cred テーブルのエントリに対する削除権が必要です。

■ 別の NIS+ 主体資格を作成または更新する場合

LOCAL 資格情報

```
nisaddcred -p uid -P principal-name local
```

DES 資格情報

```
nisaddcred -p rpc-netname -P principal-name des
```

■ 自分の資格を更新する場合

LOCAL 資格情報

```
nisaddcred -local
```

DES 資格

```
nisaddcred des
```

■ 資格を削除する場合

```
nisaddcred -r principal-name
```

資格関連のコマンド

この章で説明する nisaddcred コマンドに加えて、資格に関して有用な情報を提供するコマンドが2つ用意されています。次の表では、niscat コマンドと nismatch コマンドについて説明します。

表 12-4 その他の資格関連コマンド

コマンド	説明	参照する項目
niscat -o	ディレクトリの属性を一覧表示する。ディレクトリのサーバーの非公開鍵フィールドをチェックすることで、ディレクトリオブジェクトに非公開鍵が格納されているかわかる	330 ページの「ディレクトリのオブジェクト属性を表示する」

表 12-4 その他の資格関連コマンド (続き)

コマンド	説明	参照する項目
nismatch-	cred テーブル上で実行すると主体の資格情報を表示する	368 ページの「nismatch と nisgrep コマンド」

nisaddcred コマンドを使って資格情報を作成する方法

nisaddcred コマンドを使用して、LOCAL および DES 資格情報を作成します。

LOCAL 資格情報

LOCAL 資格情報を作成するために nisaddcred コマンドを使用すると、nisaddcred コマンドは主体のログインレコードから主体ユーザーの UID (および GID) を抽出し、ドメインの cred テーブルに置きます。

DES 資格情報

DES 資格情報を作成するのに使用した場合、nisaddcred は 2 つのプロセスを実行します。

1. 主体の Secure RPC ネット名を作成します。Secure RPC ネット名は、パスワードレコードから取得した主体のユーザー ID 番号とドメイン名 (たとえば、unix.1050@doc.com) を結合して作成されます。
2. 主体の非公開鍵と公開鍵を生成します。

nisaddcred が非公開鍵を暗号化するには、Secure RPC パスワードが必要です。nisaddcred コマンドを引数 -des で呼び出すと、主体の Secure RPC パスワードの入力を要求されます。通常このパスワードは主体のログインパスワードと同じです (異なる場合は、235 ページの「Secure RPC パスワードとログインパスワードの問題」に示す手順に従ってさらに操作が必要となります)。

nisaddcred コマンドは 1 対の乱数 (Diffie-Hellman 方式を使った 192 ビットの認証鍵) を作成します。この鍵は Diffie-Hellman の鍵ペア (key-pair) または省略して単に「鍵ペア」と呼びます。

この鍵ペアの一方が「非公開鍵」であり、もう一方が「公開鍵」になります。公開鍵は cred テーブルの公開データフィールドに置かれます。非公開鍵は、主体の Secure RPC パスワードで暗号化された後のみ、非公開データフィールドに置かれます。

nisaddcred:

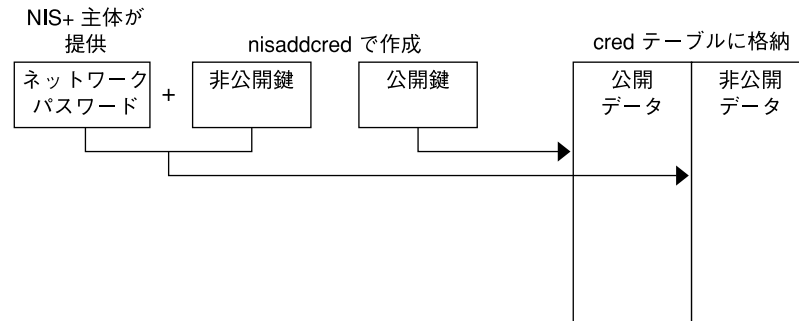


図 12-3 nisaddcred コマンドを使って主体の鍵を作成する方法

デフォルトでは、cred テーブルをすべての NIS+ 主体 (未認証主体さえも) が読み取れるので、セキュリティ上の予防措置として、主体の非公開鍵を暗号化しています。

Secure RPC ネット名と NIS+ 主体名

資格情報を作成する際、何度も主体の「RPC ネット名 (rpc-netname)」と「主体名 (principal-name)」を入力しなければなりません。どちらも独自の構文が必要です。

- 「Secure RPC ネット名」。Secure RPC ネット名は Secure RPC プロトコルによって決まるため、NIS+ の命名規則には従いません。

- ユーザーの場合: `unix.uid@domain`
- マシンの場合: `unix.hostname@domain`

Secure RPC ネット名がユーザーを識別する場合、ユーザーの UID が必要です。Secure RPC ネット名がマシンを識別する場合、マシンのホスト名が必要です。nisaddcred コマンドに使用する場合、Secure RPC ネット名の前に `-p` (小文字) フラグを入力してください。

Secure RPC ネット名は常に `unix` (すべて小文字) で始まりドメイン名で終わります。Secure RPC プロトコルに従うため、ドメイン名にはドットを付けません。

- 「主体名」。NIS+ 主体名は通常の NIS+ ネーミング規則に従いますが、常に完全指定名でなければなりません。構文は、`principal.domain` になります。

識別する対象がクライアントユーザーであっても、クライアントマシンであっても、主体名で始まり、その後にドットと完全なドメイン名が続き、最後はドットで終わります。資格の作成に使用する場合、常に先頭は `-p` (大文字) フラグで始まります。資格の削除に使用する場合、`-p` フラグは使用しません。

管理者のために資格情報を作成する方法

名前空間を最初に設定する場合、ドメインをサポートする管理者の資格情報を最初に作成します。一度管理者の資格情報を作成すると、ほかの管理者、クライアントマシン、およびクライアントユーザーの資格情報を作成できます。

自分の資格情報を作成しようとする、堂々めぐりに陥ります。つまり、ドメインの cred テーブルに対する作成権がなければ自分の資格情報を作成できず、もし NIS+ 環境が適切に設定されていれば、資格を持つまではそのような権限を持つこともできません。この循環から抜け出さなくてはなりません。これには、次の 2 つの方法があります。

- ドメインのマスターサーバーにスーパーユーザーとしてログインしている間に、資格情報を作成する
- ダミーのパスワードを使用して、ほかの管理者に自分の資格情報を作成してもらってから chkey コマンドを使用してパスワードを変更する

いずれの場合も、別の NIS+ 主体に資格情報を作成してもらうことになります。自分の資格情報を作成するには、[243 ページの「NIS+ 主体の資格情報を作成する方法」](#)の項を参照してください。

NIS+ 主体の資格情報を作成する方法

NIS+ 主体の資格情報はドメインの設定後いつでも作成できます。すなわち、cred テーブルがあれば作成できます。

NIS+ 主体の資格情報を作成する

- 主体のホームドメインの cred テーブルに対する作成権が必要
- サーバーは、主体の存在を認識している必要がある。この場合「サーバーが認識している」とは以下のことを意味する
 - 主体がユーザーの場合、ドメインの NIS+ passwd テーブルかサーバーの /etc/passwd ファイルのどちらかにエントリが必要
 - 主体がマシンの場合、ドメインの NIS+ host テーブルかサーバーの /etc/hosts ファイルのどちらかにエントリが必要

これらの条件を満たせば、-p と -P の両方のオプション付きで nisaddcred コマンドを実行できます。

LOCAL 資格情報

```
nisaddcred -p uid -P principal-name local
```

DES 資格情報

```
nisaddcred -p rpc.netname -P principal-name des
```

次の原則を銘記してください。

- 主体ユーザーの場合、LOCAL 資格と DES 資格の両方の情報を作成できます。
- 主体マシンの場合、DES 資格情報しか作成できません。
- 主体のホームドメイン (ユーザーまたはマシン) でだけ DES 資格情報を作成できます。
- ユーザーの LOCAL 資格情報はユーザーのホームドメインでもほかのドメインでも作成できます。

ユーザー主体 - 例

ユーザー ID が 11177 の morena という名前の NIS+ ユーザーの LOCAL および DES 資格情報を作成する例を次に示します。彼女の所属するドメインは doc.com. で、この例ではドメインの主体マシンから彼女の資格情報を入力します。

```
client# nisaddcred -p 11177 -P morena.doc.com. local
client# nisaddcred -p unix.11177@sales.doc.com \
-P morena.doc.com. des
Adding key pair for unix.11177@sales.doc.com
(morena.doc.com.).
Enter login password:
```

Enter login password: プロンプトに対する正しい応答は morena のログインパスワードです。彼女のログインパスワードを知らない場合は、ダミーパスワードを使用します。ダミーパスワードは、次の例のように、後で chkey コマンドを使って変更できます。

ダミーパスワードと chkey の使い方 - 例

ユーザーのログインパスワードを知らない場合、次に説明するようにダミーパスワードを使うことができます。

表 12-5 は、ダミーパスワードを使って資格情報を作成した管理者が、chkey コマンドを使ってパスワードを変更する方法を示します。この例では、UID が 119 である eiji という名前の管理者の資格情報を作成します。eiji はルートドメインに属しているので、rootmaster という名前のルートマスターサーバーから彼の資格情報を入力します。

表 12-5 管理者の資格情報を作成する - コマンドのまとめ

作業	コマンド
eiji の LOCAL 資格情報を作成する	rootmaster# nisaddcred -p 119 -P eiji.doc.com. local

表 12-5 管理者の資格情報を作成する -コマンドのまとめ (続き)

作業	コマンド
ei ji の DES 資格情報を作成する	<pre>rootmaster# nisaddcred -p unix.119@doc.com -P ei ji.doc.com. des Adding key pair for unix.119@doc.com (ei ji.doc.com.).</pre>
ei ji のダミーパスワードを入力する	<pre>Enter ei ji's login password: nisaddcred: WARNING: password differs from login passwd</pre>
ダミーパスワードを再度入力する	<pre>Retype password:</pre>
使用したダミーパスワードを ei ji に伝える	
ei ji がルートマスターにログイン	<pre>rootmaster% login: ei ji</pre>
ei ji が本当のログインパスワードを入力	<pre>Password:</pre>
ei ji はエラーメッセージを受けたが、ログインは許可される	<pre>Password does not decrypt secret key for unix.119@doc.com.</pre>
ei ji がログインを実行	<pre>rootmaster% keylogin</pre>
ei ji がダミーパスワードを入力	<pre>Password: dummy-password</pre>
	<pre>rootmaster% chkey -p Updating nisplus publickey database</pre>
ei ji が chkey を実行	<pre>Generating new key for 'unix.119@doc.com'.</pre>
ei ji が本当のログインパスワードを入力	<pre>Enter login password:</pre>
ei ji が本当のログインパスワードを再入力	<pre>Retype password: Done.</pre>

まず、通常の方法で ei ji の資格情報を作成しますが、ダミーのログインパスワードを使用します。NIS+ は警告を発して、再入力を要求します。再入力を行うと、この操作は完了します。ドメインの cred テーブルには、ダミーパスワードに基づく ei ji の資格情報が入っています。しかし、ドメインの passwd テーブル (あるいは /etc/passwd ファイル) には、まだ彼のパスワードエントリが残っているので、彼はシステムにログオンできます。

次に、eiji は本当のログインパスワードを入力して、ドメインのマスタースerver にログインします (ログイン手順では、passwd テーブルまたは /etc/passwd ファイルのパスワードをチェックするため)。そこから eiji は、まずダミーパスワードを使用して keylogin を実行し (cred テーブルをチェックするため)、chkey -p コマンドを使用して cred エントリを本当のパスワードに変更します。

別のドメインでの作成 - 例

これらの 2 つの例では、主体ユーザーのホームドメインのマスタースerver にログインしている間に、主体ユーザーの資格情報を作成しました。しかし、適切なアクセス権がある場合、別のドメインに資格を作成できます。次の構文でドメイン名を付加するだけです。

LOCAL 資格情報

```
nisaddcred -p uid -P principal-name local domain-name
```

DES 資格情報

```
nisaddcred -p rpc-netname -P principal-name des domain-name
```

次の例では、まず chou という名前のシステム管理者の LOCAL および DES 資格情報を、そのシステム管理者のホームドメイン (これは、たまたまルートドメイン) に作成し、次にその LOCAL 資格を doc.com ドメインに追加します。chou の UID は 11155 です。このコマンドは、ルートマスタースerver から入力します。簡単にするため、chou の正しいログインパスワードを入力しているものとします。

```
rmaster# nisaddcred -p 11155 -P chou.doc.com. local
rmaster# nisaddcred -p unix.11155@doc.com -P chou.doc.com. des
Adding key pair for unix.11155@doc.com (chou.doc.com.).
Enter login password:
rootmaster# nisaddcred -p 11155 -P chou.doc.com. local doc.com.
```

LOCAL 資格情報は、UID を NIS+ 主体名にマップします。クライアントユーザーである NIS+ 主体は、さまざまなドメインにさまざまな UID を持てますが、NIS+ 主体名は 1 つしか持てません。したがって、chou などの NIS+ 主体が自分のホームドメイン以外のドメインからログインする場合、そのドメインにパスワードエントリを持つだけではなく、そのドメインの cred テーブルに LOCAL 資格も持っていない必要ありません。

マシンの場合—例

主体「マシン」の資格情報を作成する例を次に示します。このホスト名は starshine1 で、ルートドメインに所属しています。したがって、この資格はルートマスタースerver から作成されます。この例では、ルートマスタースerver に root としてログインしている間に資格情報を作成します。しかし、有効な資格情報と適切なアクセス権をすでに持っている場合、自分のユーザー名でログインしているときに、資格を作成できます。

```
rootmaster# nisaddcred -p unix.starshine1@doc.com -P starshine1.doc.com. des
Adding key pair for unix.starshine1@doc.com
(starshine1.doc.com.).
Enter starshine1.doc.com.'s root login password:
Retype password:
```

パスワードプロンプトに対しては、主体マシンのスーパーユーザーパスワードを入力してください。もちろん、ダミーパスワードを使用することもできます。このダミーパスワードは、その主体マシンにスーパーユーザーとしてログインすれば、後で変更できます。

NIS+ 資格情報の管理

以下のいくつかのセクションでは、`nisaddcred` コマンドで資格情報を管理する方法について説明します。`nisaddcred` で資格情報を管理するには、`cred` テーブルに対する作成権、変更権、読み取り権、削除権が必要です。

自分の資格情報を更新する方法

自分の資格情報を更新することは、資格を作成することよりはるかに簡単です。自分のユーザー名でログインしているときに、次のような `nisaddcred` コマンドを入力するだけです。

```
# nisaddcred des
# nisaddcred local
```

別のユーザーの資格情報を更新する場合は、そのユーザーの資格情報を作成するのと同じ操作を行います。

資格情報を削除する方法

`nisaddcred` コマンドは、主体の資格情報を削除しますが、コマンドを実行しているローカルドメインだけから削除できます。

システム全体から完全に主体を削除するには、主体のホームドメインおよび `LOCAL` 資格情報のあるすべてのドメインから主体の資格情報を明示的に削除しなければなりません。

資格情報を削除するには、ローカルドメインの `cred` テーブルへの変更権が必要です。`-r` オプションを使い、完全 NIS+ 主体名で主体を指定します。

```
# nisaddcred -r principal-name
```

次の2つの例では、システム管理者 Morena の LOCAL と DES の資格情報を削除します。最初の例では、システム管理者のホームドメイン (doc.com.) から両方の資格情報を削除し、2 番目の例では、sales.doc.com. ドメインから LOCAL 資格情報を削除します。該当するドメインのマスタースerverから、それぞれがどう入力されるかに注意してください。

```
rootmaster# nisaddcred -r morena.doc.com.  
salesmaster# nisaddcred -r morena.doc.com.
```

資格が本当に削除されたことを確かめるには、次に示すように、cred テーブルに対して nismatch を実行します。nismatch の詳細は、[第 19 章](#)を参照してください。

```
rootmaster# nismatch morena.doc.com. cred.org_dir  
salesmaster# nismatch morena.doc.com. cred.org_dir
```


第 13 章

NIS+ 鍵の管理

この章では、NIS+ の鍵と、鍵を管理する方法について説明します。

ヒント - NIS+ セキュリティタスクには、Solaris 管理コンソール ツールがあればより簡単に実行できるものがあります。

注 - NIS+ は、将来のリリースではサポートされなくなる可能性があります。NIS+ から LDAP への移行支援ツールは、Solaris 9 リリース以降で使用できます (『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』を参照)。詳細については、<http://www.sun.com/directory/nisplus/transition.html> を参照してください。

NIS+ 鍵について

NIS+ 鍵は NIS+ の関連情報を暗号化するために使用します。

この章では、NIS+ セキュリティシステム全体と、特にシステムで鍵が果たす役割について、十分理解しているユーザーを想定しています。詳細については、[第 11 章](#)を参照してください。

NIS+ 鍵に関するコマンドとその構文やオプションについての詳しい説明は、[nis+\(1\)](#) のマニュアルページを参照してください。また、`nisaddcred` コマンドも鍵に関連のある働きをします。詳細については、[第 12 章](#)を参照してください。

注 - NIS+ サービスは、サービス管理機能 (SMF) によって管理されます。NIS+ サービスに対する有効化、無効化、再起動などの管理操作を実行するには、`svcadm` コマンドを使用します。SMF を NIS+ で使用する方法については、[89 ページの「NIS+ とサービス管理機能」](#)を参照してください。SMF の概要については、『Solaris のシステム管理 (基本編)』の「サービスの管理 (概要)」を参照してください。詳細については、`svcadm(1M)` と `svcs(1)` の各マニュアルページも参照してください。

NIS+ を使用したキーログイン

主体がログインする時、パスワードの入力を求めるプロンプトが現れます。このパスワードはユーザーがセキュリティゲートをパスし、ネットワークにアクセスするために必要なものです。またログインプロセスは、ユーザーのホームドメインの `cred` テーブルにあるユーザーの非公開鍵を復号化しキーサーバーへ渡します。キーサーバーはその復号化された非公開鍵を使って、ユーザーが NIS+ オブジェクトにアクセスするつど、ユーザーの認証を行います。

通常は、この場合だけ主体にパスワードが要求されます。しかし、`cred` テーブルにある主体の非公開鍵がユーザーのログインパスワードと異なるパスワードを使用して暗号化されていると、ログインプロセスはそれを `login` の際にログインパスワードを使って復号化できないので、キーサーバーに復号化した非公開鍵を提供できません。これは `cred` テーブル内にあるユーザーの非公開鍵が、ログインパスワードと異なる Secure RPC パスワードを使用して暗号化された場合に多く起こります。

注 - 「ネットワークパスワード」と「Secure-RPC パスワード」を同意語として使用する場合があります。

この問題を一時的に解決するには、ログインの後には必ず主体が `keylogin` コマンドを使用して、キーログインを実行する必要があります。-r フラグを使って、スーパーユーザー主体にログインし、ホストの `/etc/.rootkey` にスーパーユーザーの鍵を格納します。

主体ユーザーの場合

```
keylogin
```

主体マシンの場合 (一度だけ)

```
keylogin -r
```

しかしながら、もとのパスワードを使用して確実にキーログインを実行しても、そのログインセッションにおいて一時的に問題が解決するだけです。`cred` テーブルの非公開鍵はユーザーのログインパスワードと異なるパスワードを用いて暗号化されたま

まであり、次にユーザーがログインした時に同じ問題が起こります。この問題を永久的に解決するには、`chkey` コマンドを使用して、非公開鍵の暗号化に使ったパスワードをユーザーのログインパスワードに変更します (251 ページの「NIS+ 主体の鍵の変更」を参照)。

NIS+ 主体の鍵の変更

`chkey` コマンドは `cred` テーブルに格納されている NIS+ 主体の公開および非公開鍵を変更します。このコマンドは `passwd` テーブル内もしくは `/etc/passwd` ファイル内のいずれのエントリにも影響を与えません。

`chkey` コマンドについて

- 新規に鍵を作成し、パスワードを用いて非公開鍵を暗号化します。-p オプションをつけて実行した場合、`chkey` コマンドは既存の非公開鍵を新規のパスワードで再暗号化します。
- 新規に Diffie-Hellman の鍵ペアを作成し、提供されたパスワードを使用して非公開鍵を暗号化します。各主体に複数の Diffie-Hellman の鍵のペアが存在することが可能です。しかし、ほとんどの場合ユーザーは新規の鍵ペアを必要とせず、既存の非公開鍵を新規のパスワードを使って再暗号化しようとします。これを行うには、-p オプション付きで `chkey` を実行します。

これらのテーマの詳細は、マニュアルを参照してください。

注 - NIS+ 環境においては、どんな管理ツールまたは `passwd` (もしくは `nispasswd`) コマンドを用いてログインパスワードを変更しても、`cred` テーブル内の非公開鍵は新規のパスワードを使用して自動的に再暗号化されます。したがって、ログインパスワードの変更後に `chkey` コマンドを実行する必要はありません。

`chkey` コマンドはキーサーバー、`cred` テーブル、および `passwd` テーブルとの関連で動作します。`chkey` を実行するには次のようにします。

- ホームドメインの `passwd` テーブルにエントリが必要です。この条件を満たさなければ、エラーメッセージが返ります。
- キーサーバーに復号化された非公開鍵のあることを確認するために、`keylogin` を実行する必要があります。
- `cred` テーブルの変更権を持っていない限りなりません。この権限がない場合は「permission denied」などのエラーメッセージが返ります。
- `cred` テーブル内の非公開鍵の暗号化に使われた、もとのパスワードを知っている必要があります。ほとんどの場合、これは Secure RPC パスワードです。

ログインパスワードを使って非公開鍵を再暗号化するために `chkey` コマンドを使用するには、最初にもとのパスワードを用いて `keylogin` を実行し、次に [表 13-1](#) に示す手順で `chkey -p` を実行します。[表 13-1](#) は、主体ユーザーとして `keylogin` と `chkey` を実行する方法を示しています。

表 13-1 非公開鍵の再暗号化 -コマンドのまとめ

作業	コマンド
ログインする	<code>Sirius% login Login-name</code>
ログインパスワードを入力する	<code>Password:</code>
ログインパスワードと Secure RPC パスワードが異なっている場合、 <code>keylogin</code> を実行する	<code>Sirius% keylogin</code>
非公開鍵の暗号化に使用したもとのパスワードを入力する	<code>Password: Secure RPC password</code>
<code>chkey</code> を実行する	<code>Sirius% chkey -p</code> Updating nisplus publickey database Updating new key for 'unix.1199@Doc.com'.
ログインパスワードを入力する	<code>Enter login password: login-password</code>
ログインパスワードを再入力する	<code>Retype password:</code>

鍵の変更

次の節では、NIS+ 主体の鍵の変更方法を説明します。

注 - サーバーの鍵を変更するときは、常にドメイン内の全クライアントの鍵情報も更新する必要があります。その方法は [258 ページ](#) の「[クライアントの鍵情報を更新する](#)」で説明します。

ルートからルート鍵を変更する

ルートマスター (スーパーユーザーとして) からルートマスターサーバーの鍵を変更するには、[表 13-2](#) の手順を実行します。

表 13-2 ルートマスター鍵の変更 - コマンドのまとめ

作業	コマンド
新規 DES 資格を作成	rootmaster# nisaddcred des
NIS+ サービスを発見	rootmaster# svcs *nisplus*
NIS+ サービスを停止	rootmaster# svcadm disable -t /network/rpc/nisplus:default
-S 0 セキュリティオプションを削除	/lib/svc/method/nisplus ファイルを編集して -S 0 オプションを削除
セキュリティなしで NIS+ サービスを再起動	# svcadm enable network/rpc/nisplus
keylogout を実行 (以前の keylogin はタイムアウト)	rootmaster# keylogout -f
マスターがディレクトリに保管していた鍵を更新	rootmaster# nisupdkeys dirs
NIS+ サービスを発見	rootmaster# svcs *nisplus*
NIS+ サービスを停止	rootmaster# svcadm disable -t /network/rpc/nisplus:default
-S 0 セキュリティオプションを追加	/lib/svc/method/nisplus ファイルを編集して -S 0 オプションを追加
デフォルトセキュリティで NIS+ デーモンを再起動	# svcadm enable network/rpc/nisplus
keylogin を実行	rootmaster# keylogin

引数の意味は以下のとおりです。

- *dirs* は更新するディレクトリオブジェクト (rootmaster によって保管されたディレクトリオブジェクト)

表 13-2 に示すプロセスの最初の手順では、nisaddcred がルートマスターの cred テーブルを更新し、/etc/.rootkey を更新し、ルートマスターのキーログインを実行します。この時点では、マスターに保管されたディレクトリオブジェクトが更新されておらず、その資格情報とルートマスターとは同期がとれていません。この手順に続く表 13-2 の手順は、すべてのオブジェクトを正しく更新するのに必要です。

注 - サーバーの鍵を変更するときは、常にドメイン内の全クライアントの鍵情報も更新する必要があります。その方法は 258 ページの「クライアントの鍵情報を更新する」で説明します。

別のマシンからルート鍵を変更する手順

別のマシンからルートマスターの鍵を変更する場合、それに必要な NIS+ 資格とそれを行う承認を得ていなくてはなりません。

表 13-3 別のマシンによるルートマスターの鍵の変更 - コマンドのまとめ

作業	コマンド
新規の DES 資格を作成	othermachine% nisaddcred -p <i>principal</i> -P <i>nisprincipal</i> des
ディレクトリオブジェクトを更新	othermachine% nisupdkeys <i>dirs</i>
/etc/.rootkey を更新	othermachine% keylogin -r
クライアントとして再度初期化	othermachine% nisinit -cH

引数の意味はそれぞれ以下のとおりです。

- *principal* はルートマシンの Secure RPC ネット名。たとえば、以下のようになります。unix.rootmaster@doc.com (末尾のドットなし)
- *nis-principal* はルートマシンの NIS+ 主体名。たとえば、rootmaster.doc.com. (末尾のドットあり)
- *dirs* は更新するディレクトリオブジェクト (rootmaster で保管されたディレクトリオブジェクト)

nisupdkeys を実行する場合、関連したディレクトリオブジェクトのすべてを同時に更新するように注意します。すなわち、すべてを 1 つのコマンドで行います。分割して更新すると、認証エラーになります。

注 - サーバーの鍵を変更するときは、常にドメイン内の全クライアントの鍵情報も更新する必要があります。その方法は 258 ページの「クライアントの鍵情報を更新する」で説明します。

複製からルート複製の鍵を変更する手順

複製からルート複製の鍵を変更する手順を次に示します。

```
replica# nisaddcred des
replica# nisupdkeys dirs
```

引数の意味はそれぞれ以下のとおりです。

dirs は更新するディレクトリオブジェクト (replica で保管されたディレクトリオブジェクト)

nisupdkeys を実行する場合、関連したディレクトリオブジェクトのすべてを同時に更新するように注意します。すなわち、すべてを 1 つのコマンドで行います。分割して更新すると、認証エラーになります。

注 – サーバーの鍵を変更するときは、常にドメイン内の全クライアントの鍵情報も更新する必要があります。その方法は [258 ページの「クライアントの鍵情報を更新する」](#) で説明します。

ルート以外のサーバーの鍵の変更手順

サーバーからルート以外のサーバー (マスターまたは複製) の鍵を変更する手順を以下に示します。

```
subreplica# nisaddcred des
subreplica# nisupdkeys parentdir dirs
```

引数の意味はそれぞれ以下のとおりです。

- *parentdir* はルート以外のサーバーの親ディレクトリ (subreplica の NIS+ サーバーを持つディレクトリ)
- *dirs* は更新しようとするディレクトリオブジェクト (subreplica で保管されたディレクトリオブジェクト)

`nisupdkeys` を実行する場合、関連したディレクトリオブジェクトのすべてを同時に更新するように注意します。すなわち、すべてを 1 つのコマンドで行います。分割して更新すると、認証エラーになります。

注 – サーバーの鍵を変更するときは、常にドメイン内の全クライアントの鍵情報も更新する必要があります。その方法は、[258 ページの「クライアントの鍵情報を更新する」](#) で説明します。

公開鍵の更新

NIS+ サーバーの公開鍵は名前空間のあちこちに格納されています。サーバーに新規の資格情報を作成する場合、新規の鍵ペアが作成され `cred` テーブルに格納されます。しかし、名前空間ディレクトリオブジェクトには、まだサーバーの古い公開鍵のコピーが残っています。`nisupdkeys` コマンドを使用して、これらのディレクトリオブジェクトのコピーを更新します。

`nisupdkeys` コマンド

古い鍵ペアの保全が危うくなったり、非公開鍵の暗号化に使ったパスワードを忘れてしまったりして新規の鍵ペアを作成する場合、`nisupdkeys` を使用してディレクトリオブジェクト内の古い公開鍵を更新できます。

nisupdkeys コマンドで次の操作を行うことができます。

- 1 台の特定サーバーの鍵を更新する
- NIS+ ディレクトリオブジェクトをサポートしているサーバーすべての鍵を更新する
- サーバーの公開鍵をディレクトリオブジェクトから削除する
- サーバーの IP アドレスが変更された場合にそれを更新する

ただし、nisupdkeys は主体マシン上の NIS_COLD_START ファイルを更新できません。サーバーの鍵のコピーを更新するには、NIS+ クライアントが nisclient コマンドを実行しなければなりません。もしくは、NIS+ キャッシュマネージャを実行中であかつコールドスタートファイル内で1つ以上のサーバーを利用できる場合、主体はディレクトリオブジェクト上で生存期間がタイムアウトするまで待つことができます。タイムアウトが発生すると、キャッシュマネージャはコールドスタートファイルを自動的に更新します。生存期間のデフォルトは12 時間です。

nisupdkeys を使うには、NIS+ ディレクトリオブジェクトへの変更権が必要です。

公開鍵の引数の更新と例

nisupdkeys コマンドは /usr/lib/nis にあります。nisupdkeys コマンドは次の引数を使います。nisupdkeys コマンドの詳細と引数すべてのリストは、nisupdkeys (1M) のマニュアルページを参照してください。

表 13-4 nisupdkeys の引数

引数	効果
(引数なし)	カレントドメインのサーバーの鍵をすべて更新する
ディレクトリ名	ディレクトリ名で指定したディレクトリオブジェクトの鍵を更新する
-H <i>servername</i>	カレントドメインディレクトリオブジェクト内のサーバー名で指定したサーバーの鍵を更新する。ほかのドメインにあるサーバーの鍵を更新する場合は、完全ホスト名を使用する
-s -H <i>servername</i>	サーバー名で指定したサーバーで保管されたディレクトリオブジェクトすべての鍵を更新する
-C	鍵をクリアする

表 13-5 で公開鍵の更新手順の例を示します。

表 13-5 公開鍵の更新 - コマンド例

作業	コマンド
カレントドメイン (doc.com) のすべてのサーバーのすべての鍵を更新	<pre>rootmaster# /usr/lib/nis/nisupdkeys Fetch Public key for server rootmaster.doc.com. netname='unix.rootmaster@doc.com' rootmaster.doc.com. の公開鍵を更新 Public key: public-key</pre>
sales.doc.com ドメインのディレクトリオブジェクトをサポートしているすべてのサーバーの鍵を更新	<pre>salesmaster# nisupdkeys sales.doc.com (画面上には何も表示されません)</pre>
すべてのディレクトリ内のサーバー名が master7 であるサーバーの鍵を更新	<pre>rootmaster# nisupdkeys -H master7</pre>
sales.doc.com ディレクトリオブジェクトで保管された鍵をクリア	<pre>rootmaster# nisupdkeys -C sales.doc.com</pre>
カレントドメインのディレクトリオブジェクトのサーバー名が master7 であるサーバーの鍵をクリア	<pre>rootmaster# nisupdkeys -C -H master7</pre>

IP アドレスの更新

サーバーの IP アドレスを変更するか、またはアドレスを追加する場合、NIS+ アドレス情報を更新するために `nisupdkeys` を実行する必要があります。

1 つ以上のサーバーの IP アドレスを更新するには、`-a` オプション付きで `nisupdkeys` を使用します。

指定されたドメインのサーバーの IP アドレスを更新。

```
rootmaster# nisupdkeys -a domain
```

特定サーバーの IP アドレスを更新。

```
rootmaster# nisupdkeys -a -H server
```

クライアントの鍵情報を更新する

サーバーの鍵を変更するときは、常に全クライアントの鍵情報も更新する必要があります。NIS+ のサーバーは NIS+ のクライアントでもあります。そのため、あるサーバーの鍵を更新した場合は、それが NIS+ のサーバーであるか通常のクライアントであるかにかかわらず、ドメイン内にあるほかのすべてのマシンの鍵情報を更新する必要があります。

クライアントの鍵情報を更新するには、以下の 3 通りの方法があります。

- クライアント鍵情報は、クライアントの `niscclient` スクリプトを実行すれば、もっとも簡単に作成できます。
- 別の方法で個々のクライアントの鍵情報を更新するには、クライアント側で `nisinit` コマンドを実行します。339 ページの「クライアントを初期設定する」を参照してください。
- ドメイン内にある全てのマシンのクライアントの鍵情報を一括で更新するには、ドメインのディレクトリオブジェクトの生存期間の値を短くします。

クライアントの鍵情報を一括で更新する

サーバーの鍵を変更したあとで、ドメイン内にあるすべてのマシンのクライアントの鍵情報を一括更新できます。

1. **`nischttl`** コマンドを使用して、ドメインのディレクトリオブジェクトの生存期間 (**TTL**) を、すぐに満了するような小さな値にしてください。
たとえば、`sales.doc.com.` ドメイン内のサーバーの鍵を変更した場合、ディレクトリの **TTL** 値を 1 分にするには、以下のように入力します。

```
client% nischttl 60 sales.doc.com.
```

2. ディレクトリの **TTL** 値が満了すると、キャッシュマネージャはエントリを終了し、クライアントのために新しく更新された情報を入手します。
3. ディレクトリオブジェクトの **TTL** 値が満了したあとで、ディレクトリオブジェクトの **TTL** 値をデフォルト値に戻します。
たとえば、`sales.doc.com.` ドメインのディレクトリオブジェクトの **TTL** 値を 12 時間に戻すには、以下のように入力します。

```
client% nischttl 12h sales.doc.com.
```

TTL 値の使用の詳細は、347 ページの「**`nischttl`** コマンド」を参照してください。

第 14 章

拡張セキュリティ資格の管理

注 - NIS+ は、将来のリリースではサポートされなくなる可能性があります。NIS+ から LDAP への移行支援ツールは、Solaris 9 リリース以降で使用できます (『Solaris のシステム管理 (ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』を参照)。詳細については、
<http://www.sun.com/directory/nisplus/transition.html> を参照してください。

Diffie-Hellman 拡張鍵

NIS+ は、RPCSEC_GSS RPC(3N) セキュリティフレイバーを使用して、192 ビット Diffie-Hellman (RPC(3N) セキュリティフレイバー AUTH_DH) を超える RPC(3N) 層におけるより厳格なセキュリティを提供します。システム上で使用可能なセキュリティメカニズムのリストについては、`nisauthconf(1M)` コマンドを参照してください。また、これらのセキュリティメカニズムは、より厳格な暗号と各 NIS+ トランザクションの完全性を提供します。すなわち、各 NIS+ トランザクションのデータが変更されていないことが検証されます。

システム管理者は、後述のガイドラインに従って、NIS+ サーバー環境を構築する前または後に `nisauthconf(1M)` を実行することによって、より厳格なセキュリティメカニズムの利点を活用することができます。

注 - AUTH_DH は、以前は AUTH_DES と呼ばれていました。

新しい公開鍵ベースのセキュリティメカニズムへの移行

Diffie-Hellman 640 ビット (dh640-0) のような公開鍵暗号ファミリーのより厳格なセキュリティメカニズムを使用するには、既存の cred テーブルに各主体の新しい資格を追加する必要があります。後述の手順は、現在 Diffie-Hellman 192 ビット (RPC セキュリティフレイバー AUTH_DH) セキュリティを実行しているシステムを、Diffie-Hellman 640 ビット (RPC セキュリティフレイバー RPCSEC_GSS) セキュリティを実行するように変換する場合のものです。この移行手順は、もっともよくある場合を説明したのですが、公開鍵暗号ファミリーのどれか 1 つのセキュリティメカニズムタイプから、公開鍵暗号ファミリーの別のセキュリティメカニズムに変換する場合、原理は同じです。

注 – 以下の例では、\$PATH に /usr/lib/nis があることを前提にしています。

NIS+ セキュリティメカニズムの構成

NIS+ セキュリティの構成は、nisauthconf (1M) コマンドを使用して行います。nisauthconf はセキュリティメカニズムのリストを設定した順に取ります。セキュリティメカニズムは、secure_rpc (3N) で指定された 1 つまたは複数の認証フレイバーを使用することができます。des が唯一の指定されたメカニズムである場合、NIS+ はほかの NIS+ クライアントおよびサーバーの認証に AUTH_DH (旧 AUTH_DES) だけを使用します。NIS+ は、nisaddcred (1M) の場合を除いて、des より後のメカニズムは無視します。

```
nisauthconf [-v] [mechanism, ...]
```

この場合の mechanism は、システム上で使用可能な RPC セキュリティメカニズムです。使用可能なメカニズムのリストについては、nisauthconf (1M) を参照してください。メカニズムを指定しないと、現在構成されているメカニズムのリストが出力されます。

新しいセキュリティメカニズム資格の作成

NIS+ ユーザーおよびホスト主体ごとに、新しいメカニズムの資格情報を作成する必要があります。これを行うには、システムが現在のメカニズムによる認証を継続している間に、NIS+ を実行しているマシンの 1 つで `nisauthconf(1M)` コマンドを実行して新しい資格を作成しなければなりません。資格作成の基本の詳細は、[243 ページの「NIS+ 主体の資格情報を作成する方法」](#) も参照してください。

新しいセキュリティメカニズム資格 - 例

`des` から `dh640-0` へ変換します。`nisauthconf` がスーパーユーザーとして実行され、主体のホームディレクトリの中で作成権を持つ主体として `nisaddcred` が実行されている必要があります。サーバー名は `server1`、ユーザーの主体名は `morena` です。ユーザー `morena` の UID は `11177` です。

```
client# nisauthconf des dh640-0
client% nisaddcred -P server1.doc.com. -p unix.server1@doc.com dh640-0
(screen notices not shown)
client% nisaddcred -P morena.doc.com. -p unix.11177@doc.com -ldummy-password dh640-0
(screen notices not shown)
```

NIS+ ディレクトリオブジェクトへの新しい鍵の追加

すべてのサーバーの新しい資格が生成されたら、`nisupdkeys(1M)` を実行して、これらのサーバーが提供するすべてのディレクトリオブジェクトに新しい公開鍵を追加します。`nisupdkeys(1M)` コマンドを使用するには、その NIS+ ディレクトリオブジェクトに対する変更権を持っていないければなりません。詳細は、[255 ページの「公開鍵の更新」](#) を参照してください。



注意 - これらの NIS+ ディレクトリを提供するすべてのサーバー、およびこれらのディレクトリにアクセスするすべてのクライアントは、Solaris 7 以降を実行していないければなりません。

NIS+ ディレクトリオブジェクトへの新しい公開鍵の追加 - 例

この例では、新しい公開鍵を使用してサーバーが提供しているディレクトリは、doc.com、org_dir.doc.com.、および groups_dir.doc.com. です。更新は、マスターサーバー主体として行われます。新しいメカニズムを実行する前に、nisauthconf(1M) を使用して nisupdkeys を構成する必要があります。この例では、現在の認証メカニズムは des で、新しいメカニズムは dh640-0 です。

```
masterserver# nisauthconf dh640-0 des
masterserver# nisupdkeys doc.com.
               (screen notices not shown)
masterserver# nisupdkeys org_dir.doc.com.
               (screen notices not shown)
masterserver# nisupdkeys groups_dir.doc.com.
               (screen notices not shown)
```

新しいセキュリティメカニズム資格を受け入れるようにする NIS+ サーバーの構成

各サーバー上で NIS+ 認証を構成して、新旧両方の資格を受け入れるようにします。そのためには、nisauthconf(1M) および keylogin(1) を実行し、keyserv(1M) を再起動することが必要です。keylogin(1) コマンドは、各メカニズムの鍵を /etc/.rootkey に格納します。keylogin の基本の詳細は、[250 ページの「NIS+ を使用したキーログイン」](#)を参照してください。

新しいセキュリティメカニズム資格を受け入れるようにする NIS+ サーバーの構成 - 例

この例では、現在の認証メカニズムは des で、新しいメカニズムは dh640-0 です。ここでは順序が重要です。クライアントおよびサーバーの NIS+ 認証では、des エントリより後のメカニズムはすべて無視されます。

```
server# nisauthconf dh640-0 des
server# keylogin -r
               (screen notices not shown)
server# /etc/reboot
```

新しいセキュリティメカニズム資格を使用するようにするマシンの構成

サーバーで新しい資格が受け入れられるため、新しい資格によって認証するようにマシンを変換できます。そのためには、`nisauthconf(1M)` および `keylogin(1)` をスーパーユーザーとして実行し、再起動します。

新しいセキュリティメカニズム資格を使用するようにするマシンの構成 - 例

この例では、新しいメカニズムは `dh640-0` ですが、システムは、`dh640-0` 資格が使用可能でないか、あるいは `dh640-0` による認証に成功しなかった場合には、`des` 資格による認証も試みます。

```
workstation# nisauthconf dh640-0 des
workstation# keylogin -r
               (screen notices not shown)
workstation# /etc/reboot
```

次の例では、新しいメカニズムは `dh640-0` で、このメカニズムによる認証だけが行われます。システムを新しいメカニズムだけで認証するように構成する前に、キャッシュに書き込まれているディレクトリオブジェクトが新しいメカニズムの鍵を含むように、リフレッシュされることが必要です。これは、`nisshowcache(1M)` によって検証することができます。キャッシュに書き込まれているディレクトリオブジェクトがタイムアウトしてリフレッシュされるのを待つ代わりとして、次の方法があります。NIS+ サービスを停止し、`nisinit(1M)` を使用して新しい `NIS_COLD_START` を構築してから、NIS+ サービスを再起動します。

ディレクトリオブジェクトの手動によるリフレッシュ - 例

ディレクトリオブジェクトを手動でリフレッシュするには、`svcadm` コマンドを使用します。詳細は、`svcadm(1M)` のマニュアルページを参照してください。

```
# svcadm disable -t /network/rpc/nisplus:default
# nisinit -cH masterserver
# svcadm enable /network/rpc/nisplus:default
```



注意 - マシン主体およびこのマシンのすべてのユーザーが `cred` テーブルの中に `dh640-0` 資格を持つことが、`dh640-0` だけで認証するようにシステムを構成するのに必要です。

```
workstation# nisauthconf dh640-0
workstation# keylogin -r
```

```
(screen notices not shown)
workstation# /etc/reboot
```

新しい資格を保護するパスワードの変更

ダミーパスワードを持った新しい資格を与えられた各ユーザーは、`chkey(1)` を実行してログインパスワードに変換する必要があります。詳細については、[251 ページ](#)の「NIS+ 主体の鍵の変更」を参照してください。

新しい資格を保護するパスワードの変更 - 例

`chkey(1)` を実行してログインパスワードに変換します。

```
# chkey -p
(screen notices not shown)
```

新しいセキュリティメカニズム資格だけを受け入れるようにするサーバーの構成

低グレードのセキュリティメカニズムから高グレードのものに変換する場合には、新しい高グレードのセキュリティメカニズムタイプの資格だけを受け入れるように NIS+ サーバーを構成することによって、最大限のセキュリティが達成されます。新旧両方のメカニズムによって認証を行うようにサーバーを構成した後で、変換します。

新しいメカニズムだけを使用して認証を行うようにシステムを構成する前に、キャッシュに書き込まれているディレクトリオブジェクトをリフレッシュして、新しいメカニズムの鍵を含むようにし、`nisshowcache(1M)` を使用してこれを検証します。

新しいセキュリティメカニズム資格だけを受け入れるようにするサーバーの構成 - 例

各 NIS+ サーバー上で `nisauthconf(1M)` を実行し、再起動します。この例では、NIS+ サーバーは `dh640-0` 資格の認証だけを受け入れるように構成されます。


```
server# nisauthconf dh640-0
server# /etc/reboot
```

この段階で、オプションで、ディレクトリオブジェクトを更新して、古い公開鍵を削除できます。これはマスターサーバーから行う必要があり、新しいセキュリティメカニズムだけを使用して認証を行うサーバーが管理するディレクトリごとに一度、`nisupdkeys (1M)` を実行しなければなりません。この例では、更新されるディレクトリは、`doc.com`、`org_dir.doc.com.`、および `groups_dir.doc.com.` です。

```
masterserver# nisupdkeys doc.com.
               (screen notices not shown)
masterserver# nisupdkeys org_dir.doc.com.
               (screen notices not shown)
masterserver# nisupdkeys groups_dir.doc.com.
```

cred テーブルからの古い資格の削除

必要に応じて、`cred` テーブルから古いセキュリティメカニズムの資格を削除できます。そのためには、ローカルドメインの `cred` テーブルに対する変更権が必要です。詳細は、[247 ページの「資格情報を削除する方法」](#)の項を参照してください。

cred テーブルからの古い資格の削除 - 例

この例では、主体 `morena.doc.com` が自分の `des` 資格を `cred` テーブルから削除します。

```
master# nisaddcred -r morena.doc.com. dh192-0
```


第 15 章

NIS+ のアクセス権の管理

この章では、NIS+ アクセス権とその管理方法について説明します。

ヒント - NIS+ セキュリティタスクには、Solaris 管理コンソール ツールがあればより簡単に実行できるものがあります。

注 - NIS+ は、将来のリリースではサポートされなくなる可能性があります。NIS+ から LDAP への移行支援ツールは、Solaris 9 リリース以降で使用できます (『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』を参照)。詳細については、<http://www.sun.com/directory/nisplus/transition.html> を参照してください。

NIS+ のアクセス権について

NIS+ アクセス権によって、NIS+ ユーザーが実行できる操作とアクセス可能な情報が決定されます。この章では、NIS+ セキュリティシステム全体と、特にシステムでアクセス権が果たす役割について、十分理解しているユーザーを想定しています。詳しくは、[第 11 章](#)を参照してください。

NIS+ アクセス関連のコマンドとその構文、オプションについては、`nis+ (1)` のマニュアルページを参照してください。

承認およびアクセス権について

NIS+ 名前空間のセキュリティは、承認、アクセス権、および NIS+ 資格と認証の相互作用によって確保されています。これらの詳細については、[220 ページの「NIS+ の承認とアクセス - 紹介」](#)を参照してください。

承認クラス - 復習

[221 ページの「承認クラス」](#)に説明があるように、NIS+ のアクセス権は各クラスに与えられるものです。4 つの NIS+ クラスが用意されています。

- 「所有者」。所有者クラスは 1 つの NIS+ 主体です。デフォルトではオブジェクトの所有者は、オブジェクトを作成した主体になります。しかし、オブジェクトの所有者は所有権を別の主体に譲ることで所有者を変更できます。
- 「グループ」。グループクラスは 1 つ以上の NIS+ 主体の集まりです。1 つの NIS+ オブジェクトは 1 つだけの NIS+ グループを持つことができます。
- 「その他」。その他クラスには、NIS+ に認証された NIS+ 主体のすべて (すべての所有者クラスとグループクラス、および有効な DES 資格を提示したものすべて) が含まれます。
- 「未認証」。未認証クラスは、すべての適切な認証を受けられなかったもので構成されます。すなわち、有効な DES 資格を提示できなかったすべてのものです。

アクセス権 - 復習

[224 ページの「NIS+ アクセス権について」](#)で詳述したように、NIS+ には 4 種類のアクセス権があります。

- 「読み取り」。オブジェクトの読み取り権を持った主体はオブジェクトの内容を読み取ることができます。
- 「変更」。オブジェクトに対する変更権を持った主体はオブジェクトの内容を変更できます。
- 「削除」。オブジェクトに対する削除権を持った主体がオブジェクトを削除できます。
- 「作成」。上位レベルのオブジェクトに対する作成権を持った主体が、そのレベルの下位に新規のオブジェクトを作成できます。すなわち、NIS+ ディレクトリオブジェクトの作成権を持っていれば、そのディレクトリ内に新しいテーブルを作成できます。NIS+ テーブルの作成権を持っていれば、そのテーブル内に新しい列とエントリを作成できます。

これらの権限は論理的には、ディレクトリ、テーブル、列とエントリのように下位に展開するものであることを銘記してください。たとえば、新規にテーブルを作成するには、そのテーブルを格納する NIS+ ディレクトリオブジェクトに対する作成権が必

要です。そのテーブルを作成した場合、そのデフォルト所有者になります。所有者として、自分自身にそのテーブルに対する作成権を与え、テーブルに新規のエントリを作成できます。テーブル内に新規のエントリを作成した場合、それらのエントリの所有者になります。テーブルの所有者として、テーブルレベルの作成権をほかの人に与えることもできます。たとえば、自身のテーブルのグループクラスのテーブルレベルの作成権を与えることができます。その場合、テーブルのグループのすべてのメンバーがテーブル内に新規のエントリを作成できます。新規のテーブルエントリを作成した、グループの個々のメンバーはそのエントリのデフォルト所有者になります。

アクセス権の連鎖

承認クラスは連鎖の関係にあります。つまり、上位クラスは通常下位クラスにも属しており、自動的に下位クラスの権利を得ることになります。次のように機能します。

- 「所有者クラス」。オブジェクトの所有者はそのオブジェクトのグループに所属していても、いなくてもかまいません。所有者があるグループに属していると、そのグループに与えられている権利をすべて得ることになります。オブジェクトの所有者が自動的に、その他と未認証のクラスにも属することになり、自動的にこれら2つのクラスに与えられている権利を獲得することになります。
- 「グループクラス」。オブジェクトのグループメンバーは自動的にその他と未認証クラスに所属します。したがって、グループメンバーは自動的にその他クラスと未認証クラスのメンバーがそのオブジェクトに持っている権利を獲得します。
- 「その他クラス」。その他クラスは自動的に、未認証クラスがオブジェクトに対して持っている権利と同じ権利を持つことになります。
- 「未認証クラス」。未認証クラスは、オブジェクトが未認証クラスに与えている権利を持つだけです。

この基本原則は、下位クラスのアクセス権は自動的に上位クラスに与えられるということです。つまり、上位クラスは下位クラスよりも多くの権利を持つことができ、権利が少なくなることはないということです。この原則の例外は、所有者がグループのメンバーでない場合に、所有者の持っていない権利をグループクラスに与えることができることです。

アクセス権の割り当てと変更方法

NIS+ オブジェクトを作成した時、NIS+ はそのオブジェクトに所有者クラスとグループクラスのアクセス権のデフォルトセットを与えます。デフォルトでは、所有者はそのオブジェクトを作成した NIS+ 主体です。デフォルトのグループは、環境変数 `NIS_GROUP` の中で名前をつけられたグループです。

異なるデフォルト権利の指定

NIS+ は NIS+ オブジェクトが作成された時に自動的に付与されたデフォルト権利を変更する2つの異なった方法を提供しています。

- NIS_DEFAULTS 環境変数。NIS_DEFAULTS はセキュリティに関するデフォルト値を保管し、その 1 つはアクセス権です。このデフォルトアクセス権は、オブジェクトが作成された時にオブジェクトに自動的に付与されるものです (詳細については、[280 ページの「NIS+ デフォルトの表示 - nisdefaults コマンド」](#) を参照してください)。

NIS_DEFAULTS 環境変数の値を変更すると、変更後に作成されたオブジェクトに新規の値が与えられます。しかし、以前に作成されたオブジェクトは影響を受けません。

- -D オプションはいくつかの NIS+ コマンドに用いられます。NIS+ オブジェクトを作成するコマンドに -D オプションを使用すると、NIS_DEFAULTS 環境変数が指定したデフォルト権利を上書きします (詳細は、[283 ページの「デフォルトを無効にする」](#) を参照してください)。

既存オブジェクトへのアクセス権を変更する

NIS+ オブジェクトを作成する場合、既存のデフォルトアクセス権 (NIS_DEFAULTS 環境変数か -D オプションの指定かのいずれかによる) に対処する必要があります。デフォルト権利を変更するコマンドは次のとおりです。

- nischmod コマンド
- テーブルの列の場合は nistbladm コマンド

テーブル、列、およびエントリのセキュリティ

NIS+ テーブルに対するアクセス権を指定する方法には、次の 3 通りあります。

- 「テーブル」全体を対象にアクセス権を指定する
- 「エントリ」(行) 単位でアクセス権を指定する
- 「列」単位でアクセス権を指定する

列とエントリ (行) が交差する部分をフィールドといいます。データ値はすべてこの交差領域、つまりフィールドに入力します。

列とエントリレベルのアクセス権という仕組みを使用すると、テーブルレベルのアクセス権の制限があっても個々の行と列にアクセス権を追加できますが、テーブル全体へのアクセス権以上に列とエントリレベルの権利を制限することはできません。

- 「テーブル」。テーブルレベルは基本的なレベルです。テーブルレベルに付与されたアクセス権は、列ごとまたはエントリごとに特に変更された場合を除き、テーブル内のすべての部分に適用されます。テーブルレベルの権利はもっとも厳格であるべきです。

注 – 承認クラスは連結しているということに注意してください。上位クラスは、下位クラスに割り当てられた権利を取得していることになります。269 ページの「[アクセス権の連鎖](#)」を参照してください。

- 「列」。列レベルの権利という仕組みを使用すると、列ごとに追加アクセス権を付与できます。たとえば、その他クラスと未認証クラスにテーブルレベルの権利が何も付与されていなかったとします。この場合、この 2 つのクラスはテーブル内のデータに対して、読み取り権、変更権、作成権、削除権を持ちません。列レベルの権利という仕組みを使用すると、テーブルレベルの権限の制限を越えて、その他クラスのメンバーに対して特定の列のデータに対する読み取り権を付与できます。
一方、所有者クラスとグループクラスにテーブル全体の読み取り権が付与されている場合、列レベルの権利を使ってグループクラスの列の読み取り権を妨げることはできません。列のグループはテーブルのグループやエントリのグループと同じにはなりません。これらはまったく異なるグループです。
- 「エントリ (行)」。エントリレベルの権限という仕組みを使用すると、行ごとに追加アクセス権を付与できます。たとえば、個々のユーザーに指定したエントリに限って変更する権限を与えることができるのです。
エントリのグループはテーブルのグループとは同じである必要はなく、別のグループにできます。そうすることによって、特定のグループのメンバーに、ほかのグループに属するエントリに影響を与えないで、あるエントリのセットにアクセスする権限を付与できます。

テーブル、列、エントリの例

列またはエントリレベルのアクセス権は、追加アクセスを次の 2 つの方法で提供できます。1 つは権利を持つ主体を増やす方法で、もう 1 つは同じ主体に権利を追加する方法です。もちろん、これらを組み合わせることも可能です。以下にその例を示します。

テーブルオブジェクトが、そのテーブルの所有者に対して読み取り権を与えたとします。

表 15-1 テーブル、列、エントリの例 1

	未認証	所有者	グループ	その他
テーブルのアクセス権	----	r---	----	----

このことは、テーブルの所有者だけがテーブル全体の内容を読み取れることを意味します。所有者でない主体は、アクセスできません。テーブルのエントリ 2 にグループクラスに対して読み取り権を与えることができます。

表 15-2 テーブル、列、エントリの例 2

	未認証	所有者	グループ	その他
テーブルのアクセス権	----	r---	----	----
エントリ 2 のアクセス権	----	----	r---	----

テーブルの内容をすべて読み取れるのは所有者ですが、このテーブルのグループのメンバーであれば、この特定エントリの内容を読み取ることができます。次に、特定の列がその他クラスに読み取り権を与えたとします。

表 15-3 テーブル、列、エントリの例 3

	未認証	所有者	グループ	その他
テーブルのアクセス権	----	r---	----	----
エントリ 2 のアクセス権	----	----	r---	----
列 1 のアクセス権	----	----	----	r---

その他クラスのメンバーは例 1 のすべてのエントリを読み取ることができます。グループクラスのメンバーは、(その他クラスに属しているので) 列 1 のすべてとエントリ 2 の全列を読み取ることができます。*NP* になっているセルは、「グループ」、「その他」いずれのクラスも読み取りができません(アクセス権がない)。

表 15-4 テーブル、列、エントリの例 4

	列 1	列 2	列 2
エントリ 1	読み取り	*NP*	*NP*
エントリ 2	読み取り	読み取り	読み取り
エントリ 3	読み取り	*NP*	*NP*
エントリ 4	読み取り	*NP*	*NP*
エントリ 5	読み取り	*NP*	*NP*

異なるレベルの権利

この節では 4 つの権利 (読み取り、作成、変更、削除) が 4 つのアクセスレベル (ディレクトリ、テーブル、列、エントリ) とどのようにかわるのかを説明します。

種々の権利とレベルに関係した機能を次の表 15-5 にまとめます。

表 15-5 アクセス権、アクセスレベル、およびその機能

	ディレクトリ	テーブル	列	エントリ
読み取り	ディレクトリ内容のリスト	テーブル内容の読み取り	列内容の読み取り	エントリ (行) 内容の読み取り
作成	新規ディレクトリまたはテーブルオブジェクトの作成	新規エントリ (行) の追加	列に新規のデータを入力	エントリ (行) に新規のデータを入力
変更	オブジェクトの移動とオブジェクト名の変更	テーブル内の任意のデータを変更	列内のデータを変更	エントリ (行) 内のデータを変更
削除	テーブル等のディレクトリオブジェクトの削除	エントリ (行) の削除	列内のデータの削除	エントリ (行) 内のデータの削除

読み取り権

- 「ディレクトリ」。ディレクトリの読み取り権があれば、ディレクトリの内容を表示できます。
- 「テーブル」。テーブルの読み取り権があれば、テーブル内のすべてのデータを読み取れます。
- 「列」。列の読み取り権があれば、その列のすべてのデータを読み取れます。
- 「エントリ」。エントリの読み取り権があれば、そのエントリのすべてのデータを読み取れます。

作成権

- 「ディレクトリ」。ディレクトリレベルの作成権があれば、テーブル等の新規オブジェクトをディレクトリ内に作成できます。
- 「テーブル」。テーブルレベルの作成権があれば、テーブル内に新規のエントリを作成できますが、列は作成できません。テーブルレベルの作成権だけでは、テーブルに新規のエントリを追加できますが、新規の列を追加することはできません。
- 「列」。列の作成権があれば、列内のフィールドに新規のデータを入力できます。新規の列を作成できません。
- 「エントリ」。エントリの作成権があれば、その行のフィールドに新規のデータを入力できます (エントリレベルの作成権では新規の行を作成することはできません)。

変更権

- 「ディレクトリ」。ディレクトリレベルの変更権があれば、ディレクトリオブジェクトの移動と名前の変更ができます。

- 「テーブル」。テーブルレベルの変更権があれば、テーブル内のデータをすべて変更できます。新規の行を作成 (追加) できますが、新規の列は作成できません。空白フィールドにデータを入力することも可能です。
- 「列」。列の変更権があれば、その列の任意のフィールドのデータを変更できます。
- 「エントリ」。エントリの変更権があれば、その行の任意のフィールドのデータを変更できます。

削除権

- 「ディレクトリ」。ディレクトリレベルの削除権があれば、テーブル等のディレクトリ内の既存オブジェクトを削除できます。
- 「テーブル」。テーブルレベルの削除権があれば、テーブル内の既存のエントリ (行) を削除できますが、列は削除できません。削除できるのは既存のエントリだけで、既存の列は削除できません。
- 「列」。列の削除権があれば、その列の任意のフィールドのデータを削除できます。
- 「エントリ」。エントリの削除権があれば、その行の任意のフィールドのデータを削除できます。

アクセス権の格納場所

オブジェクトのアクセス権は、そのオブジェクトの定義の一部として指定され、格納されます。この情報は NIS+ テーブルには格納されません。

NIS+ オブジェクトのアクセス権の読み取り

アクセス権を読み取るには `niscat` コマンドを使用します。

```
niscat -o objectname
```

アクセス権を読み取るオブジェクト名を指定します。

このコマンドは、NIS+ オブジェクトに関する次の情報を返します。

- 「所有者」。所有権を持っている NIS+ 主体。通常はオブジェクトを作成した人ですが、もとの所有者が所有権を渡した場合もある
- 「グループ」。オブジェクトの NIS+ 主体
- 「未認証クラスのアクセス権」。認証された (有効な DES 資格を提示した) か否かにかかわらず、全員が持つ権利
- 「所有者クラスのアクセス権」。オブジェクトの所有者に付与されたアクセス権
- 「グループクラスのアクセス権」。オブジェクトのグループに付与されたアクセス権

- 「その他クラスのアクセス権」。認証された NIS+ 主体全てに付与されたアクセス権

4 つの承認クラスのアクセス権は、次のように 16 文字の文字列で表示されます。

```
r---rmcdr---r---
```

各文字がアクセス権の種類を表します。

- r は読み取り権
- m は変更権
- d は削除権
- c は作成権
- - はアクセス権のないことを表す

先頭の 4 文字は未認証に、次の 4 文字は所有者に、その次の 4 文字はグループに、そして最後の 4 文字はその他に、それぞれ与えられたアクセス権を表します。

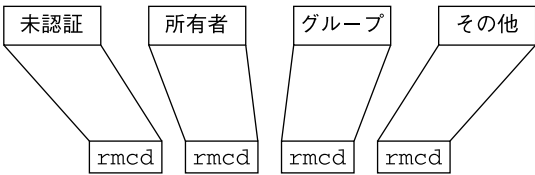


図 15-1 アクセス権の表示

注 – UNIX ファイルシステムとは異なり、先頭のアクセス権は未認証用であり、所有者用ではありません。

デフォルトのアクセス権

オブジェクトを作成すると、NIS+ はそのオブジェクトにデフォルトの所有者、グループ、およびアクセス権を割り当てます。デフォルトの所有者は、そのオブジェクトを作成する NIS+ 主体です。デフォルトのグループは、環境変数 NIS_GROUP の中で名前をつけられたグループです。デフォルトのアクセス権は次のようになります。

表 15-6 デフォルトのアクセス権

未認証	所有者	グループ	その他
-	読み取り権	読み取り権	読み取り権
-	変更権	-	-
-	作成	-	-

表 15-6 デフォルトのアクセス権 (続き)

未認証	所有者	グループ	その他
-	削除	-	-

環境変数 `NIS_DEFAULTS` のセットがある場合、`NIS_DEFAULTS` 内の値が新規のオブジェクトに適用されるデフォルト値を決定します。コマンド行でオブジェクトを作成した場合は、`-D` フラグを使ってデフォルト値以外を設定できます。

テーブルに対するアクセス権をサーバーが割り当てる方法

この節では、読み取り、変更、削除、作成の操作が行われる際、テーブルオブジェクト、エントリ、列に対するアクセス権をサーバーがどのように割り当てるかについて説明します。

注 - セキュリティレベル 0 では、サーバーはアクセス権を実行しないため、すべてのクライアントがテーブルオブジェクトに対する完全なアクセス権を付与されます。セキュリティレベル 0 は管理者だけがテストの目的で使用します。通常の業務にはレベル 0 を使用しないでください。

サーバーがアクセスを許可するか否かを決定する 4 つの要素があります。

- 主体が要求する処理の種類
- 主体がアクセスしようとしているテーブル、エントリ、または列
- その特定のオブジェクトに対して、主体が所属する承認クラス
- テーブル、エントリ、または列がその主体の承認クラスに割り当てたアクセス権

認証後に主体は、主体が有効な DES 資格を所持しているかを確認することで要求を行い、NIS+ サーバーは処理の種類と要求のオブジェクトを決定します。

- 「ディレクトリ」。オブジェクトがディレクトリかグループの場合、サーバーは 4 つのクラスに付与されている権利を知るためにオブジェクトの定義をチェックし、どのクラスに主体が所属しているか判定し、主体のクラスとそのクラスに付与された権利に基づいて、その要求を受け入れるかまたは拒否します。
- 「テーブル」。オブジェクトがテーブルの場合、サーバーは 4 つのクラスに付与されているテーブルレベルの権利を知るためにテーブルの定義をチェックし、どのクラスに主体が所属しているか判定します。所属しているクラスが要求処理を行うテーブルレベルの権利を持っていない場合は、サーバーは次にどの行または列にかかわる処理かを判定し、要求処理に必要な該当する行または列レベルのアクセス権があるかを決定します。

コマンドによるアクセス権の指定

ここでの説明では、NIS+ 環境のセキュリティレベルが 2 (デフォルト) であるものと想定しています。

この節では、この章で説明するコマンドを使用するときにアクセス権や所有者、グループ所有者、オブジェクトを指定する方法を説明します。

アクセス権の構文

この節では、承認とアクセス権に関する NIS+ コマンドに使われるアクセス権の構文について説明します。

クラス、演算子、および権利の構文

アクセス権は、環境変数で指定する場合もコマンドで指定する場合も、「クラス (class)」、「演算子 (operator)」、「権利 (right)」という 3 種類の引数で区別されます。

- 「クラス」。クラスは、「権利」が適用される NIS+ 主体のカテゴリを意味します。

表 15-7 アクセス権の構文 - クラス

クラス	目的
n	未認証: すべての未認証要求
o	オブジェクトまたはテーブルエントリの所有者
g	オブジェクトまたはテーブルエントリのグループ所有者
w	その他: すべての認証済み主体
a	すべて: 所有者、グループ、およびその他の省略形。これはデフォルト

- 「演算子」。演算子は、「権利」について行われる操作を示します。

表 15-8 アクセス権の構文 - 演算子

演算子	目的
+	「権利」によって指定されたアクセス権を追加する

表 15-8 アクセス権の構文 - 演算子 (続き)

演算子	目的
-	「権利」によって指定されたアクセス権を取り消す
=	「権利」によって指定されたアクセス権を明示的に変更する。つまり、既存の権利をすべて取り消し、新しいアクセス権に置き換える

- 「権利」。アクセス権そのものです。使用可能な値は次のとおりです。

表 15-9 アクセス権の構文 - 権利

権利	目的
r	オブジェクト定義またはテーブルエントリを読み取る
m	オブジェクト定義またはテーブルエントリを変更する
c	テーブルエントリまたは列を作成する
d	テーブルエントリまたは列を削除する

コンマ (,) で区切ることで、複数のコマンドを 1 つのコマンド行にまとめることができます。

表 15-10 クラス、演算子、権利の構文 - 例

操作	構文
読み取りアクセス権を「所有者」クラスに追加する	o+r
所有者、グループ、およびその他のクラスのアクセス権を、以前のアクセス権と異なるものに変更する	a=m
読み取りと変更の権利をその他と未認証クラスに追加する	wn+m
グループ、その他、および未認証クラスから 4 つの権利をすべて削除する	gwn-rmcd
所有者クラスに作成と削除の権利を追加し、その他と未認証クラスに読み取り権と変更権を追加する	o+cd,wn+rm

所有者とグループの構文

- 「所有者」。NIS+ グループを指定するには、NIS+ グループ名にドメイン名を付けて使います。
- 「グループ」。NIS+ グループを指定するには、NIS+ グループ名にドメイン名を付けて使います。

主体名は完全指定されていることに注意してください (*principalname.domainname*)。

所有者

principalname
グループ
groupname.domainname

オブジェクトとテーブルエントリの構文

オブジェクトとテーブルエントリは異なる構文を使います。

- オブジェクトは単純なオブジェクト名を使います。
- テーブルエントリはインデックス付きの名前を使います。

オブジェクト

objectname

テーブルエントリ

[columnname=value] , tablename

注 – この場合、角括弧 ([]?) は構文の一部であり、オプション記号ではありません。

インデックス付きの名前では、列と値のペアを複数指定できます。その場合、操作はすべての列と値のペアに一致するエントリにだけ適用されます。列と値のペアが増えると、検索条件が厳しくなります。

たとえば、以下のようになります。

表 15-11 オブジェクトとエントリの構文

種類	コメント例
オブジェクト	hosts.org_dir.sales.doc.com.
テーブルエントリ	`[uid=33555],passwd.org_dir.Eng.doc.com.`
2つの値のテーブルエントリ	`[name=sales,gid=2],group.org_dir.doc.com.`

列はインデックス付きの名前の特殊な形式になっています。列の操作は `nistbladm` コマンドでだけ可能なため、詳細は、[352 ページの「テーブルでの nistbladm コマンドの使用」](#) を参照してください。

NIS+ デフォルトの表示 - nisdefaults コマンド

nisdefaults コマンドは、名前空間内で現在有効な 7 つのデフォルトを表示します。これらのデフォルトは次のいずれかです。

- NIS+ ソフトウェアが提供するプリセット値
- 変数 NIS_DEFAULTS 値で指定 されるデフォルト (変数 NIS_DEFAULTS のセットがある場合)

オブジェクトを作成する時に -D オプション付きのコマンドを使って上書きしなければ、このマシン上でオブジェクトを作成すると自動的にデフォルト値を獲得することになります。

表 15-12 7 つの NIS+ デフォルト値と nisdefaults オプション

デフォルト	オプション	元データ	目的
ドメイン	-d	/etc/defaultdomain	コマンドを入力したマシンのホームドメインを表示する
グループ	-g	環境変数 NIS_GROUP	このシェルが作成する次のオブジェクトに付与されるグループを表示する
ホスト	-h	uname -n	マシンのホスト名を表示する
主体	-p	gethostbyname()	nisdefaults コマンドを入力した NIS+ 主体の完全指定ユーザー 名またはホスト名を表示する
アクセス権	-r	環境変数 NIS_DEFAULTS	このシェルが作成する次のオブジェクトまたはエントリに付与されるアクセス権を表示する。書式: -----rmcdr---r---
検索パス	-s	環境変数 NIS_PATH	検索パスの構文を表示する。これは NIS+ が情報を検索する時のドメインを示す。もし設定してあれば、環境変数 NIS_PATH の値を表示する
生存期間	-t	環境変数 NIS_DEFAULTS	このシェルが作成する次のオブジェクトに付与される生存期間を表示する。デフォルトは 12 時間
全部 (簡潔)	-a		7 つのデフォルトすべてを簡潔書式で表示する
冗長	-v		指定した値を冗長モードで表示する

これらのオプションを使用して、すべてのデフォルト値もしくはそのサブセットを表示できます。

- すべての値を冗長書式で表示するには、引数なしで nisdefaults コマンドを実行します。


```
master% nisdefaults
Principal Name : topadmin.doc.com.
Domain Name : doc.com.
Host Name : rootmaster.doc.com.
Group Name : salesboss
Access Rights : ----rmcdr---r---
Time to live : 12:00:00:00:00
Search Path : doc.com.
```

- すべての値を簡潔書式で表示するには、`-a` オプションを付けます。
- 値のサブセットを表示するには、適切なオプションを使用します。その値は簡潔モードで表示されます。権利と検索パスのデフォルトを簡潔モードで表示する例を次に示します。

```
rootmaster% nisdefaults -rs
----rmcdr---r---
doc.com.
```

- 値のサブセットを冗長モードで表示するには、`-v` フラグを使用します。

デフォルトセキュリティ値の設定

この節では、`nisdefaults` コマンド、環境変数 `NIS_DEFAULTS`、および `-D` オプションに関連したタスクを実行する方法を説明します。環境変数 `NIS_DEFAULTS` は次のデフォルト値を指定します。

- 所有者
- グループ
- アクセス権
- 生存期間

環境変数 `NIS_DEFAULTS` に設定した値はデフォルトとなり、そのシェルを使用して作成したすべての NIS+ オブジェクトに適用されます (`-D` オプション付きでコマンドを実行してデフォルト値に上書きした場合を除く)。

環境変数 `NIS_DEFAULTS` を指定することで、デフォルト値 (所有者、グループ、アクセス権、および生存期間) を指定できます。一度 `NIS_DEFAULTS` の値を設定するとそのシェルから作成したすべてのオブジェクトは、`-D` オプション付きでコマンドを実行して上書きした場合を除きそのデフォルトに設定されます。

NIS_DEFAULTS の値を表示する

`echo` コマンドを使って、環境変数の値をチェックできます。以下にその例を示します。

```
client% echo $NIS_DEFAULTS
owner=butler:group=gamblers:access=o+rmcd
```

nisdefaults コマンドを使用して、名前空間でアクティブな NIS+ デフォルトの一般的リストを表示することも可能です。を参照してください。

デフォルトを変更する

環境変数 NIS_DEFAULTS の値を変更することで、アクセス権、所有者、およびグループのデフォルトを変更できます。ユーザーのシェルに適切な環境コマンド (csh には setenv、sh と ksh には \$NIS_DEFAULTS=, export) を次の引数を付けて使用します。

- access=アクセス権 (277 ページの「コマンドによるアクセス権の指定」で説明されたフォーマットを使って)
- owner=所有者名
- group=グループ名

複数の引数をまとめる場合は、コロン (:) で区切ります。

owner= 主体名 :group= グループ名

表 15-13 に例をいくつか示します。

表 15-13 デフォルトの変更例

作業	例
所有者のデフォルトアクセス権に読み取り権を設定	client% setenv NIS_DEFAULTS access=o+r
デフォルトの所有者をホームドメインが doc.com. である abe に設定	client% setenv NIS_DEFAULTS owner=abe.doc.com.
2 つのコード行をまとめる	client% setenv NIS_DEFAULTS access=o+r:owner=abe.doc.com.

デフォルトを変更したシェルから作成されるすべてのオブジェクトとエントリは、指定した新規の値になります。テーブルの列またはエントリに対してはデフォルトを指定できません。列とエントリはテーブルのデフォルトをそのまま継承します。

NIS_DEFAULTS の値を再設定する

変数 NIS_DEFAULTS をオリジナルの値に再設定するには、ユーザーのシェルに適したフォーマットを使って、引数なしで変数の名前を入力します。

C シェルの場合

```
client# unsetenv NIS_DEFAULTS
```

Bourne シェルまたは Korn シェルの場合

```
client$ NIS_DEFAULTS=; export NIS_DEFAULTS
```

デフォルトを無効にする

次の NIS+ コマンドのどれかを使えば、NIS+ オブジェクトまたはテーブルエントリを作成する時に、いつでもデフォルトのアクセス権、所有者、およびグループを無効にできます。

- `nismkdir` - NIS+ ディレクトリオブジェクトを作成
- `nisaddent` - エントリを NIS+ テーブルに転送
- `nistbladm` - NIS+ テーブル内にエントリを作成

デフォルト値を無効にするには、[277 ページ](#)の「[コマンドによるアクセス権の指定](#)」の説明のように、コマンドの構文に `-D` オプションを挿入します。

デフォルトを設定する時と同様、複数の引数を 1 行のコマンド行で指定できます。なお、列とエントリの所有者とグループは常に同じであるため、これらを無効にすることはできません。

`nismkdir` コマンドを使用して、`sales.doc.com` ディレクトリを作成し、デフォルトアクセス権を無効にして所有者にだけ読み取り権を付与するには、次のように入力します。

```
client% nismkdir -D access=o+r sales.doc.com
```

オブジェクトとエントリのアクセス権を変更する

`nischmod` コマンドは、NIS+ オブジェクトまたはテーブルエントリのアクセス権を変更します。テーブル列のアクセス権は操作しません。列の場合、`nistbladm` コマンドに `-D` オプションを付けて実行してください。`nischmod` コマンドを使用するには、そのオブジェクトかエントリに対する変更権が必要です。

`nischmod` コマンドを使用して権利を追加する

オブジェクトまたはエントリに権利を追加する例を次に示します。

オブジェクト

```
nischmod class+right object-name
```

テーブルエントリ

```
nischmod class+right [column-name=value], table-name
```

sales.doc.com. ディレクトリオブジェクトのグループに読み取り権と変更権を追加する場合は、次のように入力します。

```
client% nischmod g+rm sales.doc.com.
```

hosts.org_dir.doc.com. テーブル内の name=abe エントリのグループに読み取り権と変更権を追加する場合は、次のように入力します。

```
client% nischmod g+rm '[name=abe],hosts.org_dir.doc.com.'
```

nischmod を使用して権利を削除する

オブジェクトまたはエントリの権利を削除するには、次のように入力します。

オブジェクト

```
nischmod class-right object-name
```

エントリ

```
nischmod class-right [column-name=value], table-name
```

sales.doc.com. ディレクトリオブジェクトのグループから作成権と削除権を削除するには次のように入力します。

```
client% nischmod g-cd sales.doc.com.
```

hosts.org_dir.doc.com. テーブル内の name=abe エントリのグループから削除権を削除する場合は、次のようにします。

```
client% nischmod g-d '[name=abe],hosts.org_dir.doc.com.'
```

列アクセス権を指定する

nistbladm コマンドは NIS+ テーブルに種々の操作を行うことができます。これについては [352 ページ](#)の「[テーブルでの nistbladm コマンドの使用](#)」にほとんどの説明があります。このコマンドは -c と -u の 2 つのオプションを使用してセキュリティ関連のタスクを実行できます。

- -c オプション。nistbladm コマンドを使用して テーブルを作成する際に、-c オプションを使用すると最初の列アクセス権を指定できます。

- `-u` オプション。`-u` オプションを `nistbladm` コマンドとともに使用すると、列アクセス権を変更できます。

テーブル作成時の列権利設定

テーブルが作成された時、列にはテーブルオブジェクトと同じ権利が付与されます。このテーブルレベルの権利は環境変数 `NIS_DEFAULTS` で指定されるか、またはテーブルを作成したコマンドの一部で指定されます。`nistbladm` コマンドでテーブルを作成する際に `-c` オプションを使用すれば、最初の列アクセス権を指定できます。このオプションを使用するには、テーブルを作成しようとするディレクトリの作成権が必要です。テーブル作成時に列権利を指定するには、次のように入力します。

```
nistbladm -c type `columnname=[flags] [, access]... tablename`
```

引数の意味はそれぞれ以下のとおりです。

- `type` はテーブルの種類を示す文字列です。テーブルの `type` は何にでもできます。
- `columnname` は列名です。
- `flags` には、列の種類を指定します。使用できるフラグには次のものがあります。
 - `s` (検索可能)
 - `i` (大文字と小文字を区別しない)
 - `c` (暗号化)
 - `B` (2進データ)
 - `x` (XDR 符号化データ)
- `access` は、[277 ページの「コマンドによるアクセス権の指定」](#)で説明した構文を使用したこの列のアクセス権です。
- `...` はそれぞれの型と権利のセットを持った複数の列を指定できることを示しています。
- `tablename` は完全指定名で表した作成しようとするテーブル名です。

テーブルを作成する際、列に独自の権利を付与するには、列の型とコンマの後の各列の等号記号にアクセス権を追加します。次のように、列をスペースで区切ります。

```
column=type, rights column=type, rights column=type, rights
```

以下の例では、`div` 特性で `depts` という名前のテーブルが `doc.com` ディレクトリに作成されます。列は `Name`、`Site` および `Manager` の3つです。第2、第3列のグループには、変更権が追加されます。

```
rootmaster% nistbladm -c div Name=S Site=S,g+m Manager=S,g+m depts.doc.com.
```

`nistbladm -c` オプションの詳細は、[第19章](#)を参照してください。

既存のテーブル列に権利を追加する

既存の NIS+ テーブルの列にアクセス権を追加するには、`nistbladm -u` オプション付きの `nistbladm` コマンドを使用します。このオプションを使用する場合、テーブル列の変更権が必要です。入力例を次に示します。

```
nistbladm -u [column= access,...],tablename
```

引数の意味はそれぞれ以下のとおりです。

- `column` は列名です。
- `access` は、277 ページの「[コマンドによるアクセス権の指定](#)」で説明した構文を使用したこの列のアクセス権です。
- ... は複数列のアクセス権を指定できることを示しています。
- `tablename` は完全指定名で表した作成しようとするテーブル名です。

権利を更新する列ごとに `column=access` のペアを使用します。複数の列を更新するにはコンマ (,) で区切り全体を角括弧 ([]) で囲みます。

```
[column=access, column=access, column=access]
```

このオプションの完全な構文については、[第 2 章](#)を参照してください。

次の例は、`hosts.org_dir.doc.com`. テーブルの `name` および `addr` 列のグループに、作成権と変更権を追加するものです。

```
client% nistbladm -u '[name=g+rm,addr=g+rm],hosts.org_dir..doc.com.'
```

テーブル列から権利を削除する

NIS+ テーブル列のアクセス権を削除するには、上記 286 ページの「[既存のテーブル列に権利を追加する](#)」で説明したように `-u` オプションを使用します。ただし + 記号を使って権利を追加するのではなく、- 記号を使って権利を削除します。

次の例は、`hosts.org_dir.doc.com`. テーブルの `hostname` 列のグループの読み取り権と変更権を削除するものです。

```
client% nistbladm -u 'name=g-rm,hosts.org_dir.doc.com.'
```

オブジェクトとエントリの所有権の変更

`nischown` コマンドは、1 つ以上のオブジェクトまたはエントリの所有者を変更します。このコマンドを使用するには、オブジェクトまたはエントリに対する変更権が必要です。テーブルの列の所有者はテーブルの所有者であるため、`nischown` コマンドでは列の所有者を変更できません。列の所有者を変更するには、テーブルの所有者を変更する必要があります。

nischown コマンドを使用してオブジェクトの所有者を変更する

オブジェクトの所有者を変更するには、次のように入力します。

```
nischown new-owner object
```

引数の意味はそれぞれ以下のとおりです。

- *new-owner* はそのオブジェクトの新規の所有者のユーザー ID で完全指定します。
- *object* は、オブジェクトの完全指定名です。

オブジェクト名と新規所有者名にドメイン名を必ず追加します。

次の例は、doc.com. ドメイン内の *hosts* テーブルの所有者を、ホームドメインが doc.com. でユーザー名 *lincoln* であるユーザーに変更するものです。

```
client% nischown lincoln.doc.com. hosts.org_dir.doc.com.
```

nischown コマンドを使用してテーブルエントリの所有者を変更する

テーブルエントリの所有者を変更する構文は、エントリを特定するのにインデックス付きエントリを使います。次に例を示します。この構文の詳細は、[279 ページの「オブジェクトとテーブルエントリの構文」](#)を参照してください。

```
nischown new-owner [column=value,...], tablename
```

引数の意味はそれぞれ以下のとおりです。

- *new-owner* はそのオブジェクトの新規の所有者のユーザー ID で完全指定します。
- *column* は、所有者を変更するエントリ (行) を特定する値を持った列名です。
- *value* は、所有者を変更するエントリ (行) を特定するデータ値です。
- ... は、複数のエントリで所有者の変更を指定できることを示します。
- *tablename* は、所有者を変更するエントリを含むテーブルの完全指定名です。

所有者名とテーブル名にドメイン名を必ず追加します。

次の例は、doc.com. ドメインのホストテーブル内のエントリの所有者を、ホームドメインが doc.com. であるユーザー *takeda* に変更するものです。そのエントリの *name* 列の値は *virginia* です。

```
client% nischown takeda.doc.com. '[name=virginia],hosts.org_dir.doc.com.'
```

オブジェクトまたはエントリグループの変更

`nischgrp` コマンドは、1 つ以上のオブジェクトまたはテーブルエントリのグループ所有者を変更します。このコマンドを使用するには、オブジェクトまたはエントリに対する変更権が必要です。テーブルの列に割り当てられたグループは、テーブルに割り当てられたグループと同じであるため、`nischgrp` コマンドは、列のグループを変更できません。列のグループ所有者を変更するには、テーブル所有者を変更する必要があります。

`nischgrp` コマンドを使用してオブジェクトのグループを変更する

オブジェクトのグループを変更するには、次の構文を使用します。

```
nischgrp group object
```

引数の意味はそれぞれ以下のとおりです。

- `group` は、オブジェクトの新規のグループの完全指定名です。
- `object` は、オブジェクトの完全指定名です。

オブジェクト名と新規のグループ名にはドメイン名を必ず追加します。

次の例は、`doc.com.` ドメイン内の `hosts` テーブルのグループを `admins.doc.com.` に変更するものです。

```
client% nischgrp admins.doc.com. hosts.org_dir.doc.com.
```

`nischgrp` コマンドを使用してテーブルエントリのグループを変更する

テーブルエントリのグループを変更する構文は、エントリを識別するためにインデックス付きのエントリを使用します (この構文については、[279 ページの「オブジェクトとテーブルエントリの構文」](#)に詳細説明があります)。以下に構文を示します。

```
nischgrp new-group [ column=value,...], tablename
```

引数の意味はそれぞれ以下のとおりです。

- `new-group` はオブジェクトの新規グループの完全指定名です。
- `column` は変更される特定のエントリ (行) を識別する値を持った列名です。

- *value* は変更されるグループの特定の エントリ (行) を識別するデータ値です。
- *tablename* は変更ことになるグループのエントリを含むテーブルの完全指定名です。
- ... は、複数エントリのグループの変更を指定できることを示します。

新規グループ名とテーブル名にはドメイン名を必ず追加します。

次の例は、`doc.com.` ドメインの `hosts` テーブル内のエントリのグループを `sales.doc.com.` に変更するものです。そのエントリの `name` 列の値は `virginia` です。

```
client% nischgrp sales.doc.com. '[name=virginia],hosts.org_dir.doc.com.'
```


第 16 章

パスワードの管理

この章では、一般ユーザー (NIS+ 主体) の観点から見た `passwd` コマンドの使用方法和、NIS+ 管理者によるパスワードシステムの管理方法を説明します。

ヒント - NIS+ セキュリティタスクには、Solaris 管理コンソール ツールがあればより簡単に実行できるものがあります。

注 - NIS+ は、将来のリリースではサポートされなくなる可能性があります。NIS+ から LDAP への移行支援ツールは、Solaris 9 リリース以降で使用できます (『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』を参照)。詳細については、<http://www.sun.com/directory/nisplus/transition.html> を参照してください。

パスワードの使用

マシンへのログインの際には、ユーザー名 (「ログイン ID」とも呼ばれる)、およびパスワードを入力する必要があります。ログイン ID は公開の情報ですが、パスワードを知っているのは所有者だけです。

ログイン

システムへのログインは以下の手順で行います。

1. **Login:** プロンプトで、ログイン ID を入力します。

2. **Password:** プロンプトで、パスワードを入力します。

(秘密を守るため、入力してもパスワードは画面に表示されません)

ログインに成功すると、本日のメッセージ (ない場合もある)、続いてコマンド行プロンプト、ウィンドウシステム、通常のアプリケーションなどが表示されます。

Login incorrect メッセージ

「Login incorrect」というメッセージは以下のことを意味します。

- 「誤ったログイン ID あるいはパスワードを入力した」。もっとも一般的な理由です。スペルを確認してもう一度やり直します。誤入力の回数は、ほとんどのシステムで「5」に制限されているので注意してください。
- 誤入力の回数が制限を超えると、「Too many failures - try later」というメッセージが表示され、一定の時間が経過するまで再試行ができなくなります。
- 指定された時間内にログインが成功しないと、「Too many tries; try again later」というメッセージが表示され、一定の時間が経過するまで再試行ができなくなります。
- 「管理者がパスワードをロックしている」。ログイン ID、パスワードともに正しく入力しているにもかかわらず「Login incorrect」メッセージが表示される場合は、システム管理者に問い合わせてください。
- 「パスワード使用権がシステム管理者によって取り消された」。ログイン ID、パスワードともに正しく入力しているにもかかわらず「Login incorrect」メッセージが表示される場合は、システム管理者に問い合わせてください。

will expire メッセージ

「Your password will expire in N days」メッセージ (または「Your password will expire within 24 hours」というメッセージ) は、「パスワードが、N 日あるいは 24 時間以内に有効期限に達する」ということを意味します。

このメッセージが表示されたら、パスワードをすぐに変更する必要があります (293 ページの「パスワードの変更」を参照)。

Permission denied メッセージ

ログイン ID とパスワードを入力後、「Permission denied」というメッセージが表示されて login: プロンプトに戻る場合があります。これは、管理者がパスワードをロックしたか、管理者がユーザーのアカウントを終了したか、パスワード権限の有効期限が切れたために、ログインに失敗したことを意味します。このような場合、管理者がパスワードロックを解除するか、アカウントを復旧するまではログインができません。システム管理者に問い合わせてください。

パスワードの変更

セキュリティを確保するため、パスワードは定期的に変更してください (パスワードを作成する場合の必要条件、および基準については、[294 ページの「パスワードの選択」](#)を参照)。

注 - 現在の `passwd` コマンドでは、以前 `nispasswd` で行なっていた操作がすべて行えます。NIS+ の名前空間に特有な操作を行うには、`passwd -r nisplus` を使用します。

パスワードの変更は以下の手順で行います。

1. システムプロンプトから **passwd** コマンドを実行します。
2. **Enter login password** (多少異なる場合がある) プロンプトで、従来のパスワードを入力します。
キー入力の内容は画面には表示されません。
 - 「Sorry:less than N days since the last change」というメッセージが表示されたら、「現在使用中のパスワードは、作成されてからまだ十分な時間が経過していないため変更できない」ということを意味します。この場合はシステムプロンプトに戻ります。システム管理者に「パスワードは、作成後何日経過すれば変更できるのか」を問い合わせてください。
 - 「You may not change this password」というメッセージが表示された場合は、変更がネットワーク管理者によって禁止されているということの意味します。
3. **Enter new password** プロンプトで、新しいパスワードを入力します。
キー入力の内容は画面には表示されません。
この時点で、新しいパスワードが必要条件を満たしているかどうかシステムによって確認されます。
 - 満たしている場合は再入力を求められます。
 - 満たしていない場合は、その旨を知らせるメッセージが表示されます。このときは、必要条件を満たす別のパスワードを入力する必要があります。パスワードの必要条件については、[295 ページの「パスワードの必要条件」](#)を参照してください。
4. **Re-enter new password** プロンプトで、新しいパスワードを再入力します。
キー入力の内容は画面には表示されません。
1 回目と 2 回目で入力したパスワードが異なっていた場合は、手順 1 から繰り返すようにプロンプトが表示されます。

注 - root のパスワードを変更した場合は、その直後に必ず `chkey -p` を実行する必要があります (`chkey -p` を使用し root の鍵を変更する方法の詳細については、[252 ページの「ルートからルート鍵を変更する」](#) および [253 ページの「別のマシンからルート鍵を変更する手順」](#) を参照してください)。 `chkey -p` を実行しないと、root で正しくログインできなくなります。

このメッセージは、「パスワードが有効期限を過ぎている」ということを意味します。つまり、パスワードを作成してから時間が経ちすぎているので、すぐに作成し直す必要があるということです (新しいパスワードを作成する場合の必要条件については、[294 ページの「パスワードの選択」](#) を参照してください)。

この場合、新しいパスワードの作成は以下の手順で行います。

1. **Enter login password** (多少異なる場合がある) プロンプトで、従来のパスワードを入力します。

キー入力の内容は画面には表示されません。

2. **Enter new password** プロンプトで、新しいパスワードを入力します。

キー入力の内容は画面には表示されません。

3. **Re-enter new password** プロンプトで、新しいパスワードを再入力します。

キー入力の内容は画面には表示されません。

パスワード変更の失敗

システムの中には、パスワード変更の試行回数、所要時間に制限を設けているものもあります。これらの制限は、他人がパスワードを推測して勝手にパスワードを変更してしまうのを防ぐために実装されています。

本人 (または本人になりすました他人) によるパスワード変更、ログインの試行回数、または所要時間が指定の範囲を超えた場合は、「Too many failures - try later」、または「Too many tries: try again later」といったメッセージが表示されます。この種のメッセージが表示されると、一定の時間 (システム管理者が指定) が経過するまでログイン、パスワードの変更は行えなくなります。

パスワードの選択

コンピュータのセキュリティの侵害の例としては、「ほかのユーザーのパスワードを推測によって盗む」というものが多くみられます。 `passwd` コマンドには、パスワードを推測しにくいものにするための基準がいくつか設けられていますが、ユーザーに関していくつかの情報を得るだけでパスワードがわかってしまう人もいます。したがって、「自分にとって覚えやすく他人にとって推測しにくい」というのが良いパスワードです。逆に悪いパスワードとは、「自分にとって覚えにくく (メモしないと覚えられない)、自分を知る他人にとっては推測しやすい」というものです。

パスワードの必要条件

パスワードの必要条件は以下のとおりです。

- 「長さ」。パスワードの長さは6文字以上にします(デフォルト)。また意味を持つのは最初の8文字だけです(つまり、「8文字より長いパスワードを作成することはできるが、システムは最初の8文字のみをチェックする」ということです)。パスワードの長さの最小値(文字数)はシステム管理者によって変更される可能性があるため、システムによって異なる場合があります。
- 「文字」。英文字2つ以上(大文字、小文字どちらでも良い)と数字または記号(@、#、%など)1つ以上で構成します。つまり、dog#food、dog2foodといったパスワードは使用できますが、dogfoodというパスワードは使用できません。
- 「ログインIDと同じにならないようにする」。パスワードをログインIDと同じにすることはできません。またログインIDの文字を並べ替えたものも使用できません(この場合、大文字と小文字は同じものとして考える)。たとえば、ログインIDがClaire2の場合は、e2clairはパスワードとして使用できません。
- 「古いパスワードと同じにならないようにする」。新しいパスワードは古いパスワードと比較して3文字以上異なっていなければなりません(この場合、大文字と小文字は同じものとして考える)。たとえば、古いパスワードがDog#fooDなら、新しいパスワードにdog#Meatを使用できますが、daT#Foodは使用できません。

悪いパスワードの例

悪いパスワードの例としては以下のようなものが考えられます。

- 自分の名前をもとにしたもの
- 家族やペットの名前
- 運転免許証の番号
- 電話番号
- 社会保険の番号
- 従業員番号
- 趣味に関係のある名前
- 季節に関係のある名前(たとえば12月にSantaを使うなど)
- 普通の辞書に載っている単語

良いパスワードの例

良いパスワードの例としては以下のようなものが考えられます。

- フレーズに数字や記号を加えたもの(例: beam#meup)
- フレーズを構成する単語の頭文字に番号や記号を加えた意味のない言葉(例: SomeWhere Over The RainBow の頭文字に7を加えた swotr7b)
- 単語の一部を数字や記号に置き換えたもの(例: snoopyではなく、sn00pyにする)

パスワードの管理

この節では、NIS+ 名前空間でパスワードを管理する方法について説明します。この節では、NIS+ セキュリティシステム全般、および特にログインパスワードがシステムで果たす役割について十分に理解していることを想定しています (詳細については、[第 11 章](#)を参照してください)。

注 – 現在の `passwd` コマンドでは、以前 `nispasswd` で行っていた操作がすべて行えます。NIS+ の名前空間に特有な操作を行うには、`passwd -r nisplus` を使用します。

nsswitch.conf ファイルの必要条件

`passwd` コマンドの使用や、パスワードの使用期間に関する設定を正しく行うには、`nsswitch.conf` ファイル中の `passwd` エントリがすべてのマシンにおいて正しくなければなりません。`passwd` コマンドが「パスワード情報をどこに要求するか」および「パスワード情報をどこで更新するか」は、このエントリによって決定されます。

`passwd` エントリの設定として考えられるのは、以下の 5 種類だけです。

- `passwd: files`
- `passwd: files nis`
- `passwd: files nisplus`
- `passwd: compat`
- `passwd: compat passwd_compat: nisplus`



注意 – 使用しているネットワークのすべてのマシン上に存在する `nsswitch.conf` ファイルは、必ず上記の `passwd` 構成のうちの 1 つを使用していなければなりません。上記の 5 つ以外の設定をすると、ログインができなくなります。

nispasswd コマンド

現在の `passwd` コマンドでは、以前 `nispasswd` で行っていた操作がすべて行えます。コマンド行で実行するときは、`nispasswd` ではなく `passwd` を使用します。

旧バージョンとの互換性を確保するため、`nispasswd` も完全な形で残っている点に注意してください。

yppasswd コマンド

現在の `passwd` コマンドでは、以前 `yppasswd` で行なっていた操作がすべて行えます。コマンド行で実行するときは、`yppasswd` ではなく `passwd` を使用します。

旧バージョンとの互換性を確保するため、`yppasswd` も完全な形で残っている点に注意してください。

passwd コマンド

`passwd` コマンドでは、パスワードに関する様々な操作が行えます。現在の `passwd` コマンドは、`nispasswd` コマンドの代わりとして使用できます。従来 `nispasswd` で行なっていた操作にも、`passwd` コマンドを使用するようにしてください。`passwd` コマンドのフラグ、オプション、引数の詳細は、マニュアルページを参照してください。

一般ユーザーが `passwd` コマンドで行える操作には以下のものがあります。

- 自らのパスワードの変更
- 自らのパスワード情報の表示

管理者が `passwd` コマンドで行える操作には以下のものがあります。

- ほかのユーザーにログイン時のパスワード変更を強制する
- ほかのユーザーのパスワードをロック (使用不可能に) する
- 「パスワード作成後、どのくらいの期間変更禁止にするか」を指定する
- 「パスワードが間もなく無効になる」という警告を出すタイミングを指定する
- パスワードの有効期間を指定する

passwd コマンドと `nsswitch.conf` ファイル

`passwd` などのコマンドがパスワード情報をどこから得て、どこに保存するのかということは、ネームサービススイッチで指定します。`nsswitch.conf` ファイルの `passwd` エントリの設定に使用する文字列はそれぞれ以下のことを意味します。

- `nisplus`。パスワード情報の獲得、変更、保存は、該当するドメインの `passwd` テーブルおよび `cred` テーブルで行う
- `nis`。パスワード情報の獲得、変更、保存は、`passwd` マップで行う
- `files`。パスワード情報の獲得、変更、保存は、`/etc/passwd` ファイルおよび `/etc/shadow` ファイルで行う

`passwd -r` オプション

`passwd` コマンドの `-r nisplus`、`-r nis`、`-r files` といった引数は、`nsswitch.conf` ファイルの設定よりも優先されます。これらの引数をつけて `passwd` コマンドを実行すると、「`nsswitch.conf` ファイルより優先される」というこ

とを知らせる警告メッセージが表示されます。警告メッセージ表示後も操作を続行すると、`nsswitch.conf` ファイルのシーケンスは無視され、`-r` によって指定された場所でパスワード情報の更新が行われます。

たとえば、`nsswitch.conf` ファイルにおいて `passwd` エントリが以下のように指定されている場合を考えてみましょう。

```
passwd: files nisplus
```

この場合、`passwd` コマンドを `-r` を使用しないで実行すると、パスワード情報のソース (情報の獲得、変更、保存が行われる場所) は `/etc/passwd` ファイルになります。しかし `-r nisplus` オプションを使用して `passwd` コマンドを実行すると、パスワード情報のソースは `/etc/passwd` ファイルから NIS+ の `passwd` テーブルに変更されます。

`-r` オプションは、「検索シーケンスが誤っていて `nsswitch.conf` ファイルが使用できない」という場合にのみ使用するようにします。たとえば、2 カ所に格納されているパスワード情報を更新する必要がある場合、1 つめについては `nsswitch.conf` ファイルで指定されているソースを使用できますが、2 つめについては別のソースを使用する必要があります。

変更しようとする、次のメッセージが表示されます。

```
Your specified repository is not defined in the nsswitch file!
```

このメッセージは、「パスワード情報の更新は、`-r` オプションで指定された場所で行なわれるが、`nsswitch.conf` ファイルでその場所がソースとして使用されるまでは更新の影響がまったく現れない」ということを意味します。たとえば、`nsswitch.conf` ファイルにおいて `passwd: files nis` という指定が行われているときに、`-r nisplus` オプションでパスワード使用期間の設定を行なったとします (つまり設定は NIS+ の `passwd` テーブルにおいて行われることになる)。パスワード情報のソースが `nsswitch.conf` ファイルにおいて NIS+ `passwd` テーブル以外の場所に指定されているため、この設定による影響はまったく現れません。

passwd コマンドと NIS+ 環境

この章で「NIS+ 環境とは」、「`nsswitch.conf` ファイルでパスワード情報のソースが `nisplus` に設定されている」、「`passwd` コマンドが `-r nisplus` という引数をつけて実行されている」という状況を指します。

passwd コマンドと資格

`passwd` コマンドは NIS+ 環境 (前節参照) で実行された場合、ユーザーに資格があってもなくても機能するよう設計されています。ただし資格のないユーザーが `passwd` コマンドで行えるのは、自らのパスワードの変更だけです。ほかのパスワード操作は、資格のある (認証された)、必要なアクセス権を持った (承認された) ユーザーだけが行えます。

passwd コマンドとアクセス権

承認およびアクセス権については、ユーザーがすべて適切な資格を持っているという前提で説明をします。

通常の NIS+ 環境では、passwd テーブルの所有者はいつでも制約なしにパスワード情報の変更ができます (デフォルトの場合)。つまり passwd テーブルの所有者は、読み取り、作成、変更、削除に関して完全に承認されている (アクセス権を与えられている) ということになります。また所有者は以下のことも行えます。

- nischown コマンドを使用して、テーブルの所有権を別のユーザーに与える
- テーブルのグループ、その他、未認証といったクラスに、読み取り権、作成権、変更権、削除権などを与える (この種の権利をその他クラス、未認証クラスに与えると NIS+ のセキュリティが弱くなる)
- 任意のクラスに与えられたアクセス権を、nisdefaults、nischmod、nistbladm などのコマンドを使用して変更する

注 - 与えられているアクセス権には関わりなく、その他クラス、未認証クラスのユーザーはすべてパスワードの使用期間の制約に従います。つまり、自分のパスワードであろうとほかのユーザーのパスワードであろうと、作成されてから一定の時間が経過するまでは変更ができないということです。また有効期間の過ぎたパスワードを変更しなければならないという点も、グループ、その他、未認証といったクラスのメンバーすべてに共通です。しかしパスワード使用期間に関する上記のような制約は、passwd テーブルの所有者には適用されません。

NIS+ 環境で passwd コマンドを使用する場合、行おうとする操作に関する承認 (アクセス権) が必要です。

表 16-1 passwd コマンドに関するアクセス権

操作の種類	必要な権利	アクセス対象となるオブジェクト
情報を表示する	読み取り権	passwd テーブルのエントリ
情報を更新する	変更権	passwd テーブルのエントリ
情報を追加する	変更権	passwd テーブル

passwd コマンドと鍵

NIS+ 環境で passwd コマンドを使用して主体のパスワードを変更しようとする、主体の非公開鍵が cred テーブル中で更新されます。

- cred テーブルの DES エントリに対してユーザーが変更権を持っていて、ログインパスワードと Secure RPC パスワードが同じであれば、passwd コマンドによって cred テーブルの非公開鍵が更新されます。

- cred テーブルの DES エントリに対する変更権がユーザーになく、ログインパスワードと Secure RPC パスワードが異なっている場合は、passwd コマンドによってパスワードだけが変更され、非公開鍵は変更されません。
- つまり cred テーブルの非公開鍵が、passwd テーブルに格納されているものとは異なったパスワードで作られることになります。この場合は、chkey コマンドを実行する、ログイン後に毎回 keylogin を実行するといった方法で、鍵を変更する必要があります。

passwd コマンドとほかのドメイン

ほかのドメインの passwd テーブルに対して操作をするには、passwd コマンドを以下のように使用します。

```
passwd [options] -D domainname
```

nistbladm コマンド

nistbladm は、passwd テーブルをはじめとする NIS+ テーブルに関する情報を作成、変更、表示するのに使用するコマンドです。



注意 - nistbladm コマンドを使用してパスワード操作をするには、nistbladm を passwd テーブルのシャドウ列に適用する必要があります。nistbladm をシャドウ列に適用するのは複雑で微妙な作業になります。passwd コマンドまたは Solaris 管理コンソールツールでより簡単に行える操作には、nistbladm コマンドを使用するべきではありません。

つまり以下のような操作には、nistbladm ではなく passwd コマンドや Solaris 管理コンソール ツールを使用してください。

- パスワードの変更
- パスワードの使用期間の設定
- パスワード作成後から変更が可能になるまでの時間の設定
- 「パスワードが間もなく無効になる」という警告が表示されるタイミングの設定
- パスワードの使用期間に関する設定の解除

nistbladm コマンドには以下の機能があります。

- 新しい passwd テーブルエントリの作成
- 既存のエントリの削除
- passwd テーブル中の UID フィールド、GID フィールドの更新
- passwd テーブルの、アクセス権などセキュリティに関する属性の更新
- ユーザーアカウントに関して「どれだけの期間使用できるか」、「どれだけの期間使用されなければ無効になるか」を設定する (312 ページの「パスワード使用権の有効期限」、313 ページの「ログインの間隔の最大値の指定」を参照)

nistbladm と シャドウ列のフィールド

nistbladm コマンドでは、シャドウ列の様々なフィールドの値を指定することによってパスワードパラメータの設定を行います。シャドウ列のフィールドの値は、以下のような形式で設定します。

	最小値	警告	期限	
nistbladm -m shadow=n1	:n2	:n3	:n4	:n5 :n6 :n7 [name=login], passwd.org_dir
	最終変更日	最大値	間隔	未使用

引数の意味はそれぞれ以下のとおりです。

- 「n1 最終更新日」。パスワードが最後に変更された日付です。1970 年 1 月 1 日からの日数で表されます。このフィールドの値は、パスワードが変更されると自動的に更新されます (日数に関する重要な情報は、[303 ページの「nistbladm と日数」](#)に示されています)。このフィールドが空白であったり、指定されている値が 0 の場合は、過去に一度もパスワードの変更が行われていないことを意味します。

このフィールドの値は、残りのフィールドの設定の基礎となることに注意してください。このフィールドの値が不当に変更されると、残りのフィールドの設定も不当なものになります。

- 「n2 最小値」。パスワードが変更されてから、次の変更が可能になるまでの期間です。たとえば、パスワード作成時の最終変更日フィールドの値が 9201 (1970 年 1 月 1 日から 9201 日経過したことを示す) で、最小値フィールドの値が 8 だとすると、9209 日目まではパスワードの変更ができません (詳細は、[309 ページの「パスワードの変更禁止期間の設定」](#)を参照)。

この値としては以下の 3 とおりが考えられます。

- 「0」または空白の場合は、変更禁止期間がないことを意味します。
- 「0 より大きい数字」は、この数字がパスワードの変更禁止期間 (単位: 日) になります。
- 「最大値 (n3) より大きい値」は、このフィールド値が最大フィールドの値より大きい場合、パスワードを変更できません。変更しようとすると、次のメッセージが表示されます。「You may not change this password」
- 「n3 最大値」は、パスワードが作成されてから、使用できなくなるまでの日数を示します。この日数を過ぎると、ログインの際新しいパスワードの設定が強制されます。たとえば、パスワード作成時の最終変更日の値が 9201 で、最大値が 30 の場合、9231 (9201+30) 日目が経過した後は、ログイン時に新しいパスワードの設定が強制されます (詳細は、[308 ページの「パスワードの有効期間の設定」](#)を参照)。

この値としては以下の 3 とおりが考えられます。

- 「0」は「次のログイン時にパスワードの変更を強制し、その後は有効期間を設定する機能を停止する」ということを意味します。
- 「0 より大きい値」は、パスワードの有効期間の日数を指定します。

- 「マイナス1 (-1)」は、「-1」は、有効期間を設定する機能を停止するということを意味します。つまり `passwd -x -1 username` と入力すると、使用期間に関して該当ユーザーに行われていた設定はすべて解除されます。このフィールドを空白にしておく、と、-1 として扱われます。
- 「n4 警告」は、パスワードの使用期間がその最大値に達する前に、パスワードを変更するように警告される日数です。たとえば、最終変更日時フィールドの値が 9201、最大値フィールドの値が 30、警告フィールドの値が 5 であると仮定します。この場合、日付が 9226 日 ($9201+30-5=9226$) 以降になると、ログインするたびに「パスワードを変更してください」という警告が表示されます。パスワードの警告時期の詳細については、[310 ページの「警告期間の設定」](#)を参照してください。
この値としては以下の 2 とおりが考えられます。
 - 警告メッセージが表示される期間がないことを示します。
 - 警告メッセージが表示される期間をその日数に指定します。
- 「n5 間隔」は、ログインとログインの間隔 (日数) の最大値です。ここで指定された日数を超える期間ログインを行わないと、ログインができなくなります。たとえば、間隔に 6 を指定した場合、6 日間ログインを行わないと 7 日目にはログインができなくなります (アカウントの最大ログイン間隔についての詳細は、[313 ページの「ログインの間隔の最大値の指定」](#)を参照)。
この値としては以下の 2 とおりが考えられます。
 - 「マイナス1 (-1)」は、最大ログイン間隔機能を解除します。任意の日数の間オフにすることができ、ログイン権限を失うことはありません。これがデフォルトです。
 - 最大ログイン間隔 (日数) を指定します。
- 「n6 期限」は、パスワードの有効期限を示します。1970 年 1 月 1 日からの日数で表されます。この日を過ぎると、ログインができなくなります。たとえば、期限が 9739 (1995 年 9 月 1 日) に設定されている場合、1995 年 9 月 2 日 (GMT) になるとログインができなくなります。このときログインを試行すると「Login incorrect」というメッセージが表示されます (詳細は、[312 ページの「パスワード使用権の有効期限」](#)を参照)。
この値としては以下の 2 とおりが考えられます。
 - 「マイナス1 (-1)」は、有効期限の設定を解除します。パスワードが有効期限を過ぎている場合でも、-1 を設定すると再び使えるようになります。有効期限を設定したくないときは、-1 を指定するようにしてください。
 - 有効期限の日付 (「1970 年 1 月 1 日から数えて何日目か」) です。現在以前の日付を設定すると、パスワードはすぐに無効になります。
- 「n7 未使用」というフィールドは、現在使用されていません。値を設定しても無視されます。
- 「Login」はユーザーのログイン ID を指定します。



注意 - nistbladm を使用して passwd テーブルのシャドウ列を設定する場合、すべてのフィールドに適切な値を指定する必要があります。空白のまま残したり、0 を入力したりしても「変更なし」という意味にはなりません。

ユーザー amy が、パスワードの最終変更日が 1995 年 5 月 1 日 (1970 年 1 月 1 日から数えて 9246 日目)、パスワード作成後の変更禁止期間が 7 日、パスワードの有効期間が 30 日、「パスワードが間もなく無効になる」という警告が表示されるのがパスワード作成から 26 日目以降、ログインとログインの間隔の最大値が 15 日、アカウントの有効期限が (1970 年 1 月 1 日から) 9255 日目という設定にするには、以下のように入力します。

```
nistbladm -m shadow=9246:7:30:5:15:9285 [name=amy], passwd.org.dir
```

nistbladm と日数

パスワードの使用期間に関するパラメータは、日数で表されるものがほとんどです。日数を指定する際には以下の規則を守る必要があります。

- 1970 年 1 月 1 日を 0 とします。つまり 1970 年 1 月 2 日は 1 になります。
- NIS+ では、日付の計算、カウントに GMT (Greenwich Mean Time) が使用されます。つまり日付が変更されるのは、GMT の午前 0 時です。
- 日数の指定には整数を使用します。小数は使用できません。
- パスワードのロックなどの動作は、指定された日付に行われます。たとえば、パスワード使用権の有効期限を 1995 年 1 月 2 日 (1970 年 1 月 1 日から 9125 日目) に設定した場合、この日は「ユーザーがパスワードを使用できる最後の日」となります。パスワードが使用できなくなるのは、次の日からです。

最終変更日、期限のどちらのフィールドも、入力するのは 1970 年 1 月 1 日から数えた日数です。たとえば、以下のようになります。

表 16-2 1970 年 1 月 1 日からの日数

日付	日数
1970 年 1 月 1 日	0
1970 年 1 月 2 日	1
1971 年 1 月 2 日	365
1997 年 1 月 1 日	9863

パスワード関連のコマンド

passwd および nistbladm コマンドと類似の機能を持つコマンドはほかにもあります。それぞれについて表 16-3 にまとめてあります。

表 16-3 関連コマンド

コマンド	目的
yppasswd	現在は passwd コマンドにリンク。yppasswd を起動すると passwd コマンドが起動される
nispasswd	現在は passwd コマンドにリンク。nispasswd を起動すると passwd コマンドが起動される
niscat	テーブルの内容を表示するのに使用

パスワード情報の表示

ドメイン中のユーザーのパスワード情報を表示するには、passwd コマンドを使用します。情報は、全ユーザーについて同時に表示することも、ユーザーごとに表示することもできます。

自分のパスワード情報

```
passwd -s
```

ドメインの全ユーザーの情報

```
passwd -s -a
```

個々のユーザー

```
passwd -s username
```

表示できるのは、エントリおよび列のうち読み取り権を持っているものだけです。エントリの表示形式は以下のとおりです。

- 使用期間に関する情報が省略されたもの: *username status*
- 使用期間に関する情報が付加されたもの: *username status mm/dd/yy min max warn expire inactive*

表 16-4 NIS+ パスワード情報表示の形式

列	目的	参照箇所
<i>username</i>	ユーザーのログイン名	
<i>status</i>	パスワードの状態。PS は、そのアカウントにパスワードがあることを示す。LK は、そのパスワードがロックされていることを示す。NP は、そのアカウントにパスワードがないことを示す	306 ページの「パスワードのロック」を参照
<i>mm/dd/yy</i>	パスワードが最後に変更された日付 (GMT で表す)	

表 16-4 NIS+ パスワード情報表示の形式 (続き)

列	目的	参照箇所
<i>min</i>	パスワード作成後の変更禁止期間 (日数)	309 ページの「パスワードの変更禁止期間の設定」を参照
<i>max</i>	パスワードの有効期間 (日数)	308 ページの「パスワードの有効期間の設定」を参照
<i>warn</i>	「パスワードが間もなく無効になる」という警告が表示されるまでの日数	310 ページの「警告期間の設定」を参照
<i>expire</i>	パスワードの有効期限 (日付)	312 ページの「パスワード使用権の有効期限」を参照
<i>inactive</i>	ログインとログインの間隔の最大値 (日数)。ログインとログインの間隔がここで指定した日数を超えると、ログインができなくなる	313 ページの「ログインの間隔の最大値の指定」を参照

別のドメインの `passwd` テーブルのエントリを表示するには、`-D` オプションを使用します。

ドメイン中の全ユーザー

```
passwd -s -a -D domainname
```

個々のユーザー

```
passwd -s -D domainname username
```

パスワードの変更

新しいパスワードを作成するときは、295 ページの「パスワードの必要条件」に示された必要条件を満たすようにします。

自分のパスワードの変更

自分のパスワードを変更するには、以下のように入力します。

```
station1% passwd
```

「古いパスワードの入力」、「新しいパスワードの入力」、「新しいパスワードの再入力 (確認のため)」という順にプロンプトが表示されるので、それに従って作業します。

他人のパスワードの変更

同じドメイン内に存在するほかのユーザーのパスワードを変更するには、以下のように入力します。

```
passwd username
```

ほかのドメイン内のユーザーのパスワードを変更するには、以下のように入力します。

```
passwd -D domainname username
```

NIS+ 環境 (298 ページの「passwd コマンドと NIS+ 環境」参照) で passwd コマンドを使用して他人のパスワードを変更する場合、passwd テーブルにおける該当ユーザーエントリへの変更権が必要になります。つまり、該当する passwd テーブルに対して変更権を持つグループのメンバーになる必要があります。このとき、該当ユーザーの古いパスワードや自分のパスワードを入力する必要はありません。確認のため新しいパスワードの入力を求めるプロンプトが 2 回表示されます。一致しない場合は、さらに 2 回入力する必要があります。

root のパスワードの変更

passwd コマンドで root のパスワードを変更した場合は、その直後に chkey -p を実行する必要があります。chkey -p を実行しないと、root で正しくログインできなくなります。

root のパスワードの変更手順は以下のとおりです。

1. **root** でログインします。
2. **passwd** コマンドで **root** のパスワードを変更します。
nispasswd は使用しないでください。
3. **chkey -p** を実行します。
必ず -p オプションを使用します。

パスワードのロック

NIS+ 環境 (298 ページの「passwd コマンドと NIS+ 環境」を参照) では、passwd テーブルの該当ユーザーのエントリに対する変更権を持つ管理者 (グループのメンバー) が、passwd コマンドを使用してそのパスワードをロックできます。パスワードがロックされると、そのアカウントは使用できなくなります。また、パスワードがロックされているユーザー名を使ってログインをしようとすると、「Login incorrect」というメッセージが表示されます。

該当ユーザーがすでにログインしている場合、パスワードをロックすることによる影響は現れないという点に注意してください。ただ、パスワードの入力が必要になる login、rlogin、ftp、telnet などの機能は使用できなくなります。

すでにログインしているユーザーのパスワードをロックしても、そのユーザーが passwd コマンドでパスワードを変更するとロックは解除されます。

以下のようなことが有効です。

- 休暇などで不在になっているユーザーのパスワードを一時的にロックする。ほかの人に不正に使用されることを防止できる
- セキュリティ上の問題が発生している可能性があれば、すぐに一部のユーザーのパスワードをロックする
- 解雇になったユーザーのパスワードをすぐにロックする。ユーザーのアカウントを削除するより速く容易に行える上、そのアカウントに格納されているデータをそのまま容易に保管できる
- UNIX プロセスにパスワードを設定している場合には、この種のパスワードもロックできる。パスワードがロックされてもプロセスの実行はできるが、プロセスを使用してログインすることは (たとえパスワードを知っていても) できなくなる。多くの場合、プロセスは NIS+ 主体の形では設定されない。しかしプロセスのパスワード情報は、/etc ディレクトリのファイルに保存される。このとき /etc に格納されたパスワードをロックするにはファイルモードで `passwd` コマンドを実行する必要がある。

パスワードのロックは以下のように行います。

```
passwd -l username
```

パスワードロックの解除

パスワードのロックは、パスワードを変更すれば解除されます。この場合、「変更」とは必ずしもパスワードを新しいものに変えるという意味ではなく、パスワード変更の手順を踏むということです (元と同じパスワードに「変更」することも可能です)。ただし、新しいパスワードに変更することも可能です。

たとえば、jody というユーザーのパスワードのロックを解除するには、以下のように入力します。

```
station1% passwd jody
```

パスワードの使用期間に関する設定

設定により、パスワードの定期的な変更をユーザーに強制できます。

パスワードの有効期限については、以下のような設定が可能です。

- 次回のログイン時に強制的にパスワードを変更させる (詳細は、[308 ページの「パスワードの強制的な変更」](#)を参照)
- パスワードの有効期間 (日数) を設定する (詳細は、[308 ページの「パスワードの有効期間の設定」](#)を参照)
- パスワード作成後の変更禁止期間を設定する (詳細は、[309 ページの「パスワードの変更禁止期間の設定」](#)を参照)
- パスワードの有効期間が終わる前に「パスワードが間もなく無効になる」という警告メッセージが (ログイン時に) 表示されるタイミングを設定する (詳細は、[310 ページの「警告期間の設定」](#)を参照)

- ログインとログインの間隔の最大値 (日数) を指定する。指定された日数を超えてログインが行われないと、パスワードがロックされる (詳細は、[313 ページの「ログインの間隔の最大値の指定」](#)を参照)
- パスワードの有効期限 (日付) を設定する。この日を過ぎると、該当ユーザーはシステムにログインできなくなる (詳細は、[312 ページの「パスワード使用権の有効期限」](#)を参照)

上記の警告期間や有効期間などに達したのがログインの行われた後であった場合、影響は現れません。そのまま普通に動作し続けます。

影響が現れるのは、次のログイン時か、ログインを必要とする機能 (以下に例を示す) を使用した時です。

- login
- rlogin
- telnet
- ftp

パスワードの使用期間に関する設定は、ユーザーごとに行います。パスワードの使用期間に関する必要条件はユーザーによって違っている可能性があります。ユーザー一般にデフォルト設定を適用することもできます。詳細は、[307 ページの「パスワードの使用期間に関する設定」](#)を参照してください。

パスワードの強制的な変更

次回ログイン時のパスワード変更をユーザーに強制するには、以下の 2 種類の方法があります。

使用期間に関する設定を引き続き有効にする場合

```
passwd -f username
```

使用期間に関する設定を解除する場合

```
passwd -x 0 username
```

パスワードの有効期間の設定

パスワードの有効期間を設定するには、passwd コマンドの引数 *max* を使用します。有効期間は日数で指定します。有効期間が終了した後は、新しいパスワードをユーザーが作成しなければなりません。有効期間終了後に同じパスワードでログインしようとしても、「Your password has been expired for too long」というメッセージが表示され、新しいパスワードを作成しない限りログインを完了できません。

引数 *max* は以下の形式で使します。

```
passwd -x max username
```

引数の意味はそれぞれ以下のとおりです。

- *username* は、ユーザーのログイン ID
- *max* は、以下のうちいずれかになります。
 - 「0 より大きい値」は、パスワードの有効期間の日数を指定します。
 - 「0」は「次のログイン時にパスワードの変更を強制し、その後は有効期間を設定する機能を停止する」ということを意味します。
 - 「マイナス1 (-1)」は、「-1」は、有効期間を設定する機能を停止するということの意味します。つまり **passwd -x -1 username** と入力すると、使用期間に関して該当ユーザーに行われていた設定はすべて解除されます。

たとえば、schweik というユーザーに 45 日ごとのパスワード変更を強制するには、以下のように入力します。

```
station1% passwd -x 45 schweik
```

パスワードの変更禁止期間の設定

パスワード作成後の変更禁止期間を設定するには、引数 *min* を使用します。変更禁止期間が終了しないうちにパスワードを変更しようとする、と、「Sorry less than N days since the last change」というメッセージが表示されます。

引数 *min* は以下の形式で使用します。

```
passwd -x max -n min username
```

引数の意味はそれぞれ以下のとおりです。

- *username* は、ユーザーのログイン ID
- *max* は、パスワードの有効期間 (前のセクションを参照)
- *min* は、パスワード作成後の変更禁止期間

たとえば、ユーザー eponine のパスワードの、有効期間を 45 日間、変更禁止期間を 7 日間に設定する場合は、以下のように入力します。

```
station1% passwd -x 45 -n 7 eponine
```

min 引数には以下の規則があります。

- *min* 引数は必ずしも使用しなくて良い (つまり、パスワード変更禁止期間は必ずしも指定しなくて良い)
- *min* 引数は、必ず *-max* 引数と組み合わせて使用する。つまり、変更禁止期間を設定するには、有効期間も設定する必要がある
- *min* に *max* より大きい値を設定すると (例: **passwd -x 7 -n 8**)、ユーザーはパスワードを変更できなくなる。変更しようとする、と、「You may not change this password」というメッセージが表示される。*min* に *max* より大きな値を設定すると、次の 2 つの効果が得られる

- ユーザーによるパスワードの変更を禁止する。つまり、管理特権を有する一部の人物以外は、パスワードを変更できなくなる。たとえば、複数のユーザーに共通のグループパスワードを割り当て、そのパスワードに対して *min* 値より大きな *max* 値を設定すると、グループ内のユーザーは誰一人としてパスワードを変更できなくなる
- パスワードは *max* 値で設定される期限内のみ有効となる。しかも、*min* 値が *max* 値より大きい場合、ユーザーはパスワードを変更できない。つまり、*max* 値の示す有効期限が満了する前にパスワードの期限延長を図る手だては、ユーザーには一切与えられない。実際、有効期限が満了した後は、管理者の介在なしにユーザーはログインできなくなる

警告期間の設定

パスワードの有効期限以前の一定期間、ログイン時に「Your password will expire in N days」(*N* は日数) というメッセージが表示されるようにするには、引数 *warn* を使用します。

たとえば、パスワードの有効期限が 30 日間 (引数 *-max* で指定する) で、*warn* が 7 に設定されている場合、パスワード作成後 24 日目のログイン時に「Your password will expire in 7 days」という警告メッセージが表示されます。また翌日 (パスワード作成後 25 日目) には、警告メッセージは「Your password will expire in 6 days」となります。

警告メッセージは、電子メールで送信されるものでも、ユーザーのコンソールウィンドウに表示されるものでもないという点に注意してください。表示されるのは、ユーザーのログイン時のみです。つまり指定の期間ユーザーがログインしなければ警告メッセージは表示されません。

warn の値は、*max* の値との「関連によって機能する」ものであるという点に注意してください。つまり、「*max* によって設定される有効期限から逆算されるもの」ということです。*warn* の値が 14 であれば、「Your password will expire in N days」というメッセージの表示が開始されるのは、有効期限の 2 週間前です。

warn の値は *max* の値との関連によって機能します。*max* 値の指定がある場合のみ有効です。そのため、*max* 値の指定がない場合は、*warn* 値は意味がなくなります。

引数 *warn* は以下の形式で使用します。

```
passwd -x max -w warn username
```

引数の意味はそれぞれ以下のとおりです。

- *username* は、ユーザーのログイン ID
- *max* は、パスワードの有効期限 (日数) (308 ページの「パスワードの有効期間の設定」を参照)
- *warn* は、「有効期限の何日前から警告メッセージの表示を開始するか」を指定する

たとえば、*nilovna* というユーザーの、パスワードの有効期間を 45 日間とし、警告メッセージの表示を有効期限の 5 日前から開始する場合、以下のように入力します。

```
station1% passwd -x 45 -w 5 nilovna
```

warn 引数には以下の規則があります。

- *warn* 引数は必ずしも使用しなくて良い。*warn* の値を設定しなければ、警告メッセージは表示されない
- *warn* 引数は、必ず *max* 引数と組み合わせて使用する。つまり、警告メッセージの表示期間を指定するには、有効期間の指定が必要になる

注 – Solaris 管理コンソール を使用して、*warn* 値をユーザーのパスワード用に設定することもできます。

パスワードの使用期間に関する設定の解除

ユーザーのパスワードの有効期間を解除するには、以下の 2 つの方法があります。

有効期間を取り消す。パスワードはそのまま使用できる

```
passwd -x -1 username
```

次回ログイン時にパスワードを強制的に変更させた後、有効期間を取り消す

```
passwd -x 0 username
```

上記の方法では、*max* の値を 0 か -1 に設定する (詳細は、[308 ページの「パスワードの有効期間の設定」](#)を参照)

たとえば、ユーザー *mendez* のパスワードを次回ログイン時に強制的に変更させ、その後パスワードの有効期間を解除するには、以下のように入力します。

```
station% passwd -x 0 mendez
```

注 – Solaris 管理コンソール を使用しても、このパラメータをユーザーのパスワード用に設定できます。

この値を設定するには、*nistbladm* コマンドを使用できます。たとえば、ユーザー *otsu* のパスワードの有効期間を取り消し、変更の必要がないようにするには、以下のように入力します。

```
station1% nistbladm -m 'shadow=0:0:-1:0:0:0:0' [name=otsu],passwd.org_dir
```

nistbladm コマンドの使用の詳細は、[300 ページの「nistbladm コマンド」](#)を参照してください。

パスワード使用権の有効期限

引数 *expire* には「ユーザーのパスワード使用権がいつ無効になるか」を日付で指定します。ユーザーのパスワード使用権が無効になると、そのユーザーは有効なパスワードを所有できなくなります。該当ユーザーのログインをこの日以降禁止し、システムから締め出します。

たとえば、ユーザー *pete* の有効期限を 1997 年 12 月 31 日に設定すると、どんなパスワードを使用しても 1998 年 1 月 1 日以降 *pete* という ID ではログインができなくなります。ログインをしようとしても、「Login incorrect」というメッセージが表示されます。

「有効期間」と「有効期限」の違い

パスワード使用権の有効期限は、パスワードの有効期間とは異なっています。

- 「パスワードの有効期間」。厳密に区別をする必要のない場合には、有効期間が終了した後も変更されないパスワードのことを「有効期限の過ぎたパスワード」と呼ぶことがあります。しかしこのパスワードは、もう一度だけログインに使用できます。ただしその際に変更を強制されます。
- 「パスワード使用権の有効期限」。ユーザーのパスワード「使用権」が無効になると、そのユーザーはすべてのパスワードでログインできなくなります。つまり、ネットワークにログインする権限の有効期限が切れたことになります。

有効期限の設定

パスワード使用権が無効になっても、その影響が現れるのは、該当ユーザーがログインをしようとするときだけです。該当ユーザーがすでにログインしていた場合には、パスワード使用権が無効になったことによる影響は現れません。ただし、*rlogin*、*telnet* などログインを必要とする機能は使用できなくなり、いったんログアウトするとログインできなくなります。以上のことから、パスワード使用権に有効期限を設ける場合、毎日のワークセッション終了時には必ずログアウトするようユーザー全員に指示してください。

注 – Solaris 管理コンソール ツールを使用して有効期限の設定ができるときは、*nistbladm* を使用しないでください。Solaris 管理コンソール ツールの方が使いやすく、設定を誤る可能性が低くなります。

nistbladm コマンドでパスワード使用権の有効期限を指定するには、以下のように入力します。

```
nistbladm -m 'shadow=n:n:n:n:n6:n' [name=login],passwd.org_dir
```

引数の意味はそれぞれ以下のとおりです。

- *login* は、ユーザーのログイン ID です。

- *n* は、シャドウ列の (*n*5 以外の) フィールドの値です。
- *n*6 は、ユーザーのパスワード使用権の有効期限 (日付) を設定します。「1970 年 1 月 1 日から数えて何日目か」で表します (表 16-2 を参照)。このフィールドに指定する値には、以下の 2 種類があります。
 - 「マイナス1 (-1)」は、有効期限の設定を解除します。ユーザーのパスワードがすでに有効期限を過ぎていても、-1 を設定することによって再び使用できるようになります。有効期限を設定したくないときは、初めから -1 を指定しておきます。
 - 有効期限の日付 (「1970 年 1 月 1 日から数えて何日目か」) です。現在以前の日付を設定すると、ユーザーのパスワードはすぐに無効になります。

たとえば、ユーザー *pete* の有効期限を 1995 年 12 月 31 日に設定する場合は、以下のように入力します。

```
station1% nistbladm -m 'shadow=n:n:n:n:n:9493:n' [name=pete],passwd.org_dir
```



注意 - 空白のフィールドや、値の正しくないフィールドがあると機能しません。

パスワード使用権の有効期限の解除

有効期限を過ぎたパスワード使用権を再び使用できる状態にするには、*nistbladm* コマンドを使用して該当フィールド (*n*6) に -1 を指定します。例を以下に示します。

```
station1% nistbladm -m 'shadow=n:n:n:n:n:-1:n' [name=huck],passwd.org_dir
```

また、*nistbladm* で「*n*6」フィールドを将来の日付に設定し直すという方法も使用できます。

ログインの間隔の最大値の指定

引数 *inactive* には、「ログインからログインまでの間隔を最大何日あけることができるか」を指定します。この引数に指定された日数を超えてログインが行われなかった場合、該当ユーザーはマシンにログインできなくなります。ログインしようすると、「Login incorrect」というメッセージが表示されます。

この設定はネットワーク全体ではなく、個々のマシンに対して行われます。したがって、NIS+ 環境においてログイン間隔の最大値を設定するには、該当ユーザーのエントリをホームドメインの *passwd* テーブルに指定します。この設定はネットワーク上のすべてのマシンで該当ユーザーに対して適用されます。

たとえば、ユーザー *sam* の最大ログイン間隔を 10 日間に設定した場合を考えてみましょう。*sam* は 1 月 1 日にマシン A とマシン B にログインをした後、ログアウトしました。続いて 3 日後の 1 月 4 日、*sam* はマシン B にログインした後ログアウトしまし

た。さらに9日後の1月13日になると、samはマシンBにはログインできますがマシンAにはログインできません。マシンBにおいては最終ログインから9日間しか経過していませんが、マシンAにおいては13日間が経過しているからです。

最大ログイン間隔の設定は、まだ一度もログインをしていないマシンには適用されないという点に注意してください。一度もログインをしていないマシンには、引数 *inactive* の設定およびほかのマシンへのログイン実績がどのようになっていようとログインが可能です。



注意 – 「毎日の作業終了時には必ずログアウトをせよ」という指示がユーザーに対して行われていない場合、最大ログイン間隔の設定はしないでください。この設定はログインにだけ関係しており、ほかの形でシステムを使用する場合は考慮されていません。仮にユーザーが作業終了後もログインしたままの状態をシステムを放置しておく、ログインが発生しないため指定のログイン間隔をすぐ超えてしまいます。指定のログイン間隔を超えた状態で再起動やログアウトをすると、再びログインできなくなります。

注 – Solaris 管理コンソール ツールを使用して最大ログイン間隔の設定ができる場合は、*nistbladm* を使用しないでください。Solaris 管理コンソール ツールの方が使いやすく、設定を誤る可能性が低くなります。

最大ログイン間隔の設定を *nistbladm* コマンドで行う場合は、以下の形式を使用します。

```
nistbladm -m 'shadow=n:n:n:n5:n:n' [name=login],passwd.org_dir
```

引数の意味はそれぞれ以下のとおりです。

- *login* は、ユーザーのログイン ID です。
- *n* は、シャドウ列の (*n5* 以外の) フィールドの値です。
- *n5* は、最大ログイン間隔 (日数) です。指定される値には、以下の2種類があります。
 - 「マイナス1 (-1)」は、最大ログイン間隔機能を解除します。任意の日数の間オフにすることができ、ログイン権限を失うことはありません。これがデフォルトです。
 - 最大ログイン間隔 (日数) を指定します。

たとえば、ユーザー *sam* の最大ログイン間隔を7日間に指定する場合は、以下のように入力します。

```
station1% nistbladm -m 'shadow=n:n:n:n:7:n:n' [name=sam],passwd.org_dir
```

最大ログイン間隔の設定を解除する (ログインできなくなったユーザーを再びログインできるようにする) には、*nistbladm* で最大ログイン間隔を -1 に設定します。

パスワードの使用規則の設定 (およびそのデフォルト)

パスワードの使用規則に関する設定およびそのデフォルトについて説明します。

/etc/default/passwd ファイル

パスワード情報の獲得先が `nsswitch.conf` ファイルで `files` に指定されているすべてのユーザーのパスワードについて、4つのデフォルト設定を行うためのファイルです。`/etc/default/passwd` ファイルで行われたデフォルト設定は、`/etc` ディレクトリのファイルからパスワード情報を獲得しているユーザーにだけ適用されます。パスワード情報の獲得先が NIS マップあるいは NIS+ テーブルであるようなユーザーには適用されません。NIS+ サーバー上の `/etc/default/passwd` ファイルは、パスワード情報をローカルファイルから獲得しているローカルユーザーにだけ影響を与えます。NIS+ サーバー上の `/etc/default/passwd` ファイルは、NIS+ 環境または、`nsswitch.conf` ファイルでパスワード情報の獲得先が `nis`、`nisplus` に設定されているユーザーには影響を与えません。

`/etc/default/passwd` ファイルで行われる「パスワードに関する4つのデフォルト設定」とは、具体的には以下のものを指します。

- 有効期間 (週単位)
- 変更禁止期間 (週単位)
- 「パスワードが間もなく無効になる」という警告メッセージの表示される期間 (週単位)
- 最低文字数

`/etc/default/passwd` ファイルのデフォルト設定には、以下の規則があります。

- パスワード情報をローカルの `/etc` ディレクトリのファイルから得ているユーザーの場合、`password` コマンドまたは Solaris 管理コンソール で個々に行われた設定の方が、`/etc/default/passwd` のデフォルト設定よりも優先します。つまり `/etc/default/passwd` ファイルのデフォルト設定は、`passwd` テーブル中のエントリで (`/etc/default/passwd` の設定に対応した) 個別の設定が行われていないユーザーにだけ適用されるということです。
- `/etc/default/passwd` ファイルのすべてのデフォルト設定は、パスワードの長さを除いて、週単位として表現されます。パスワードの有効期間は、日数として表現されます。
- `MAXWEEKS`、`MINWEEKS`、および `WARNWEEKS` のデフォルト値は、の最終変更日を起点として計算されます。ただし、各警告値は、有効期限を起点として逆算されます。

`/etc/default/passwd` ファイルには、デフォルトでは以下のエントリがすでに含まれています。

```
MAXWEEKS=  
MINWEEKS=  
PASSELENGTH=
```

設定に必要な作業は、= の後に適切な数字を入力することだけです。= の後に数字が入っていないエントリは無効です。たとえば、MAXWEEKS に 4 を指定するには、以下のように入力します。

```
MAXWEEKS=4
MINWEEKS=
PASSLENGTH=
```

有効期限 (週単位)

/etc/default/passwd ディレクトリのファイルの MAXWEEKS により、ユーザーのパスワードの有効期間を週単位で設定できます。ユーザーのパスワードの有効期間を週単位で指定するためのエントリです。以下に示すとおり、"=" の後に適切な数字を入力するだけで指定ができます。

```
MAXWEEKS=N
```

N は週の数です。たとえば、MAXWEEKS=9 というようにします。

変更禁止期間 (週単位)

/etc/default/passwd ファイルの MINWEEKS デフォルト値には、ユーザーのパスワードの変更禁止期間を週単位で設定できます。パスワードの変更禁止期間を週単位で指定するためのエントリです。以下に示すとおり、"=" の後に適切な数字を入力するだけで指定ができます。

```
MINWEEKS=N
```

N は週の数です。たとえば、MINWEEKS=2 というようにします。

警告期間 (週単位)

/etc/default/passwd ファイルに WARNINWEEKS デフォルト値を追加し、パスワードの有効期間が切れて無効になる前に警告を表示する期間を、週単位で設定できます。たとえば、MAXWEEKS デフォルト値に 9 を設定したときに、パスワードが無効になる前の 2 週間に警告する場合は、MAXWEEKS に 7 を設定します。

MAXWEEKS のデフォルトを設定しない限り、WARNWEEKS のデフォルトを設定することはありません。

WARNWEEKS は、WARNWEEKS に指定された有効期限ではなく、パスワードの最終変更日を起点と考えることに注意してください。したがって WARNWEEKS に、MAXWEEKS 以上の値を設定することはできません。

WARNWEEKS のデフォルトは、MAXWEEKS のデフォルトと一緒に設定されていない限り無効です。

WARNWEEKS は、以下に示すとおり "=" の後に適切な数字を入力するだけで指定ができます。

WARNWEEKS=N

N は週の数です。たとえば、WARNWEEKS=1 というようにします。

最低文字数

passwd コマンドには、デフォルトで「パスワードの長さは6文字以上にする」という規則があります。しかし、/etc/default/passwd ファイルの PASSLENGTH を使用することによってこの規則を変更することが可能です。

パスワードの最低文字数を6以外の値にするには、以下に示すとおり PASSLENGTH= の後に適切な値を入力します。

PASSLENGTH=N

N は文字数です。たとえば、PASSLENGTH=7 というようにします。

パスワード変更時の制限

パスワード変更の際の入力試行回数と所要時間には、制限を設定できます。設定は rpc.nispasswd デーモン起動時の引数で行います。

入力試行回数を制限したり、タイムウィンドウを設定したりすることにより、「権利のない人が、パスワードを試行錯誤で知ることのできるものに変えてしまう」という危険をある程度 (完璧ではない) 回避できます。

試行回数の制限

パスワード変更時の誤入力試行回数を制限するには、rpc.nispasswd に *-a number* という引数 (*number* は試行回数) を指定します。rpc.nispasswd を起動するには、NIS+ マスターサーバー上にスーパーユーザー特権が必要です。

誤入力試行回数を4に制限する (デフォルトは3) 場合、以下のように入力します。

```
station1# rpc.nispasswd -a 4
```

この場合、4 回目のパスワード入力失敗すると、「Too many failures - try later」というメッセージが表示され、一定の時間が経過するまで該当ユーザーのパスワード変更はできなくなります。

所要時間の制限

パスワード変更時の所要時間 (ログインの所要時間) を制限するには、rpc.nispasswd に *-c minutes* という引数 (*minutes* には時間を分単位で指定する) を指定します。rpc.nispasswd を起動するには、NIS+ マスターサーバー上にスーパーユーザー特権が必要です。

たとえば、ユーザーが2分以内にログインしなければならないように設定するには、以下のように入力します。

```
station1# rpc.nispasswd -c 2
```

この場合、2分以内にパスワードの変更に成功しないとエラーメッセージが表示され、一定の時間が経過するまで該当ユーザーのパスワード変更はできなくなります。

第 17 章

NIS+ グループの管理

この章では、NIS+ グループとその管理方法について説明します。

ヒント - NIS+ セキュリティグループのタスクには、Solaris 管理コンソール ツールを利用するともっと簡単に実行できるものもあります。

注 - NIS+ は、将来のリリースではサポートされなくなる可能性があります。NIS+ から LDAP への移行支援ツールは、Solaris 9 リリース以降で使用できます (『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』を参照)。詳細については、<http://www.sun.com/directory/nisplus/transition.html> を参照してください。

Solaris グループ

Solaris の NIS+ 環境には、UNIX グループ、ネットグループ、NIS+ グループの 3 種類のグループがあります。

- 「UNIX グループ」。UNIX グループとは、特別な UNIX アクセス権を与えられたユーザーの集まりのことです。NIS+ 名前空間では、UNIX グループに関する情報は `org_dir` ディレクトリオブジェクト (`group.org_dir`) 内のグループテーブルに格納されます。UNIX グループのメンバーの追加、変更、または削除の方法については、第 19 章を参照してください。
- 「ネットグループ」。ネットグループとは、マシンとユーザー (ほかのマシン上でリモート操作を実行する権限を与えられている) の集まりのことです。NIS+ 名前空間では、ネットグループに関する情報は `org_dir` ディレクトリオブジェクト

(netgroup.org_dir) 内のネットグループテーブルに格納されます。ネットグループのメンバーの追加、変更、または削除の方法については、[第 19 章](#)を参照してください。

- 「NIS+ グループ」。NIS+ グループとは、NIS+ オブジェクトに対する特別なアクセス権 (ネームスペースの管理を許されることが多い) を与えられている NIS+ ユーザーの集まりのことをいいます。NIS+ グループに関する情報は groups_dir ディレクトリオブジェクト内のテーブルに格納されます。

NIS+ グループ

NIS+ グループは、NIS+ オブジェクトに対するアクセス権を 1 つ以上の NIS+ 主体に割り当てる場合に使用されます。これらのアクセス権については、[第 11 章](#)を参照してください。NIS+ グループに関する情報は、NIS+ groups_dir ディレクトリオブジェクト内のテーブルに格納されます。各グループは個別のテーブルを持ち、グループ名とテーブル名はそれぞれ対応します。たとえば、admin グループに関する情報は admin.groups_dir テーブルに格納されます。

NIS+ グループを作成したら、どれか 1 つには admin という名前を付けることをお勧めします。そして、この admin グループには、NIS+ アクセス権を持たせるユーザーを割り当ててください。必ずしも admin という名前にしなければならないわけでもないのですが、NIS+ マニュアルでは、NIS+ 管理権限を持たせるユーザーのグループを admin と呼んでいます。異なるユーザーと異なる権限を組み合わせる複数の NIS+ グループを作成することもできます。

注 - NIS+ グループのメンバー (ユーザー) の管理には、nisgrpadm コマンドを使います。グループテーブルの管理には、nislsl コマンドと nischgrp コマンドを使います。グループテーブルに対して nistbladm コマンドを使うことはできないので注意してください。

NIS+ グループ関連のコマンドとその構文、オプションの詳細は、nis+ (1) のマニュアルページを参照してください。

関連する管理コマンド

nisgrpadm コマンドを使ってほとんどのグループ管理作業を実行できますが、グループ管理に関連するコマンドには次のものがあります。

表 17-1 グループに関連するコマンド

コマンド	説明	参照する項目
nissetup	ドメインのグループが格納されるディレクトリである groups_dir を作成する	
nislsl	groups_dir ディレクトリの内容、つまり、ドメイン内の全グループを表示する。各グループは groups_dir に個別のテーブルを持ち、グループ名とテーブル名はそれぞれ対応する	330 ページの「nislsl コマンドでディレクトリを表示する」
nischgrp	グループを任意の NIS+ オブジェクトに割り当てる	288 ページの「オブジェクトまたはエントリグループの変更」
niscat	NIS+ グループのオブジェクト属性とメンバーを表示する	323 ページの「NIS+ グループについて niscat を使用する」
nisdefaults	新しい NIS+ オブジェクトに割り当てられるグループを表示する	280 ページの「NIS+ デフォルトの表示 - nisdefaults コマンド」

以上のコマンドの詳細 (構文、オプションなど) は、nis+(1) のマニュアルページを参照してください。

注 - NIS+ グループテーブルに対して nistbladm コマンドを使うことはできません。

NIS+ グループメンバーのタイプ

NIS+ グループには、明示的 (explicit)、暗黙的 (implicit)、および再帰的 (recursive) という 3 タイプのメンバーがあります。またメンバー以外の主体にも、同様の 3 つのタイプがあります。メンバーのタイプは、グループのメンバーを追加、削除する際に使用されます (324 ページの「nisgrpadm コマンド」を参照)。

メンバーのタイプ

- 「明示的なもの」。個々の NIS+ 主体です。これらは、すべてのグループ管理コマンドで、主体名によって識別されます。主体名は、デフォルトドメインから入力される場合、完全指定名を使用する必要はありません。
- 「暗黙的なもの」。NIS+ ドメインに所属するすべての NIS+ 主体です。これらは、* 記号とドットで始まるドメイン名によって識別されます。選択した処理は、グループ内のすべてのメンバーに適用されます。

- 「再帰的なもの」。ほかの NIS+ グループのメンバーである、すべての NIS+ 主体です。これらは、@ 記号で始まる NIS+ グループ名によって識別されます。選択した処理は、グループ内のすべてのメンバーに適用されます。

NIS+ グループは、明示的、暗黙的、および再帰的の 3 つすべてのカテゴリでのメンバー以外も受け付けます。メンバー以外の主体とは、「メンバーになることができるが、指定によってグループから排除されているもの」を指します。

メンバー以外の主体のタイプ

メンバー以外の主体はマイナス記号で始まり、メンバー主体と区別されます。

- 「明示的なもの」 - 名前の先頭に「-」を付けます。
- 「暗黙的なもの」 - ドメイン名の先頭に「-」に続いて「*」、「.」を付けます。
- 「再帰的なもの」 - グループ名の先頭に「-」に続いて「@」を付けます。

グループの構文

同じ主体について「グループに含まれる」という指定と「グループに含まれない」という指定があった場合、指定の順序に関係なく「含まれない」という指定が優先されます。たとえば、同じ主体について「グループに含まれる暗黙的なドメインのメンバーである」という指定と「グループに含まれない再帰的なグループのメンバーである」という指定の両方がある場合、後者の方が優先されます。

nisgrpadm コマンドを使用する場合、主体のタイプは表 17-2 のように指定します。

表 17-2 主体のタイプの指定方法

主体のタイプ	構文
明示的なメンバー	<i>username.domain</i>
暗黙的なメンバー	<i>*.domain</i>
再帰的なメンバー	<i>@groupname.domain</i>
メンバー以外 (明示的なもの)	<i>-username.domain</i>
メンバー以外 (暗黙的なもの)	<i>-*.domain</i>
メンバー以外 (再帰的なもの)	<i>@groupname.domain</i>

NIS+ グループについて niscat を使用する

niscat -o コマンドを使用して、NIS+ グループのオブジェクト属性を表示できます。

グループのオブジェクト属性を表示する方法

グループのオブジェクト属性を表示するには、そのグループが格納されている groups_dir ディレクトリへの読み取り権が必要です。ここでは、niscat -o とグループの完全指定名を使用します。この完全指定名には、次に示すようにその groups_dir サブディレクトリを含んでいなければなりません。

```
niscat -o group-name.groups_dir.domain-name
```

たとえば、以下ようになります。

```
rootmaster# niscat -o sales.groups_dir.doc.com.  
Object Name : sales  
Owner : rootmaster.doc.com.  
Group : sales.doc.com.  
Domain : groups_dir.doc.com.  
Access Rights : ----rmcdr---r---  
Time to Live : 1:0:0  
Object Type : GROUP  
Group Flags :  
Group Members : rootmaster.doc.com.  
                 topadmin.doc.com.  
                 @.admin.doc.com.  
                 *.sales.doc.com.
```

注 - nisgrpadm -l コマンドを使うと、メンバーリストはさらに整理されて表示されます。

グループ属性のいくつかは、環境変数 NIS_DEFAULTS から継承されます。ただし、このグループの作成時に環境変数が無効になっている場合を除きます。Group Flags フィールドは、現在使用されていません。グループメンバーのリストでは、* 記号はメンバーのドメインを、@ 記号はメンバーのグループをそれぞれ示します。

nisgrpadm コマンド

nisgrpadm コマンドは、NIS+ グループを作成したり、削除したり、さまざまな管理作業を実行します。nisgrpadm コマンドを使用するには、操作に必要なアクセス権を取得しなければなりません。

表 17-3 nisgrpadm に必要なアクセス権

操作の種類	必要なアクセス権	アクセス対象となるオブジェクト
グループの作成	作成	groups_dir ディレクトリ
グループの削除	削除	groups_dir ディレクトリ
メンバーの表示	読み取り	グループオブジェクト
メンバーの追加	変更	グループオブジェクト
メンバーの削除	変更	グループオブジェクト

nisgrpadm には、グループ作業用とグループメンバー作業用に 2 つの形式があります。

グループの作成または削除、あるいはメンバーの表示

```
nisgrpadm -c group-name.domain-name
nisgrpadm -d group-name
nisgrpadm -l group-name
```

メンバーの追加または削除、あるいはメンバーがグループに所属するかどうかの判定 (member... には、表 17-2 に示した、6 種類のメンバーのどのような組み合わせでも指定できます)。

```
nisgrpadm -a group-name member...
nisgrpadm -r group-name member...
nisgrpadm -t group-name member...
```

作成 (-c) 以外のすべての処理には、部分指定名 group-name を使用できます。ただし、-c オプションの場合でも、nisgrpadm では group-name 引数に groups_dir を使用する必要はありません。実際これは受け付けられません。

NIS+ グループを作成する

NIS+ グループを作成するには、グループのドメインの groups_dir ディレクトリに対する作成権が必要です。-c オプションと完全指定グループ名を使用します。

```
nisgrpadm -c group-name.
domainname
```

グループを作成すると、指定した名前の NIS+ グループテーブルが `groups_dir` に作成されます。`nislsls` コマンドを使うと、対応するテーブルが `groups_dir` に作成されているかどうかを確認できます。また、`niscat` コマンドを使うと、そのテーブル中のグループメンバーを一覧表示できます。

新しく作成したグループにはメンバーがありません。どのメンバーをどのグループに追加するかについては、[326 ページの「NIS+ グループにメンバーを追加する」](#)を参照してください。

次の例では、`admin` という名前の 3 つのグループを作成します。最初のグループは、`doc.com` ドメインに作成します。2 番目のグループは、`sales.doc.com` ドメインに作成します。3 番目のグループは、`manf.doc.com` に作成します。これらはすべて、それぞれのドメインのマスタースerverから作成されます。

```
rootmaster# nisgrpadm -c admin.doc.com.
Group admin.doc.com. created.
salesmaster# nisgrpadm -c admin.sales.doc.com.
Group admin.sales.doc.com. created.
manfmaster# nisgrpadm -c admin.manf.doc.com.
Group admin.manf.doc.com. created.
```

作成されるグループは、変数 `NIS_DEFAULTS` で指定されたオブジェクト属性をすべて継承します。つまり、その所有者、所有グループ、アクセス権、生存期間、および検索パスです。これらのデフォルトを表示するには、`nisdefaults` コマンドを使用します ([第 15 章](#)を参照)。オプションなしで使用すると、次のように出力されます。

```
rootmaster# nisdefaults
Principal Name : rootmaster.doc.com.
Domain Name : doc.com.
Host Name : rootmaster.doc.com.
Group Name :
Access Rights : ----rmcdr---r---
Time to live : 12:0:0
Search Path : doc.com.
```

所有者は `Principal Name`: フィールドに表示されます。所有者グループ名は、環境変数 `NIS_GROUP` を設定した場合にだけ表示されます。たとえば、C シェルを想定して `NIS_GROUP` を `net_admins.doc.com` に設定する場合は、次のように入力します。

```
rootmaster# setenv NIS_GROUP net_admins.doc.com
```

もちろん、グループの作成時に `-D` オプションを使用することによって、これらのデフォルトはどれでも変更できます。

```
salesmaster# nisgrpadm -D group=special.sales.doc.com.-c
admin.sales.doc.com. Group admin.sales.doc.com. created.
```

NIS+ グループを削除する

NIS+ グループを削除するには、グループのドメイン内の `groups_dir` ディレクトリに対する削除権が必要です。`-d` オプションを使用します。

```
nisgrpadm -d group-name
```

デフォルトドメインが正しく設定されている場合、完全指定グループ名を使用する必要はありません。ただし、別のドメイン内のグループを誤って削除しないように、まず (nisdefaults を使用して) チェックしなければなりません。次の例では test.sales.doc.com. グループを削除します。

```
salesmaster% nisgrpadm -d test.sales.doc.com.  
Group 'test.sales.doc.com.' destroyed.
```

NIS+ グループにメンバーを追加する

NIS+ グループにメンバーを追加するには、グループオブジェクトに対する変更権が必要です。-a オプションを使用します。

```
nisgrpadm -a group-name members...
```

321 ページの「NIS+ グループメンバーのタイプ」の記述どおり、主体 (明示的なメンバー)、ドメイン (暗黙的なメンバー)、およびグループ (再帰的なメンバー) を追加できます。デフォルトドメインに所属するメンバー名またはグループ名は、完全指定する必要がありません。この例では、デフォルトドメイン sales.doc.com. の NIS+ 主体 panza と valjean、および manf.doc.com. ドメインの主体 makeba を top-team.sales.doc.com. グループに追加します。

```
client% nisgrpadm -a Ateam panza valjean makeba.manf.doc.com.  
Added panza.sales.doc.com to group Ateam.sales.doc.com  
Added valjean.sales.doc.com to group Ateam.sales.doc.com  
Added makeba.manf.doc.com to group Ateam.sales.doc.com
```

この動作を確認するには、nisgrpadm -l オプションを使用します。「明示的なメンバー」のカテゴリにあるメンバーを探してください。

次の例では、doc.com. ドメイン内のすべての NIS+ 主体を staff.doc.com. グループに追加します。これは、doc.com. ドメイン内のクライアントから入力します。ドメイン名の前にある * 記号とドットに注意してください。

```
client% nisgrpadm -a Staff *.doc.com.  
Added *.doc.com. to group Staff.manf.doc.com.
```

次の例では、NIS+ グループ admin.doc.com. を admin.manf.doc.com. グループに追加します。これは manf.doc.com. ドメインのクライアントから入力します。グループ名の前にある @ 記号に注意してください。

```
client% nisgrpadm -a admin @admin.doc.com.  
Added @admin.doc.com. to group admin.manf.doc.com.
```

NIS+ グループのメンバーを表示する

NIS+ グループのメンバーを表示するには、グループオブジェクトに対する読み取り権が必要です。-l オプションを使用します。

```
nisgrpadm -l group-name
```

次の例では、admin.manf.doc.com. グループのメンバーを表示します。これは manf.doc.com. グループ内のクライアントから入力します。

```
client% nisgrpadm -l admin
Group entry for admin.manf.doc.com. group:
  No explicit members
  No implicit members:
  Recursive members:
  @admin.doc.com.
  No explicit nonmembers
  No implicit nonmembers
  No recursive nonmembers
```

NIS+ グループからメンバーを削除する

NIS+ グループからメンバーを削除するには、グループオブジェクトに対する変更権が必要です。-r オプションを使用します。

```
nisgrpadm -r group-name members. . .
```

次の例では、Ateam.sales.doc.com グループから NIS+ 主体 allende と hugo.manf.doc.com. を削除します。これは sales.doc.com.domain ドメイン内のクライアントから入力します。

```
client% nisgrpadm -r Ateam allende hugo.manf.doc.com.
Removed allende.sales.doc.com. from group Ateam.sales.doc.com.
Removed hugo.manf.doc.com. from group Ateam.sales.doc.com.
```

次の例では、admin.manf.doc.com. グループから admin.doc.com. グループを削除します。これは manf.doc.com. ドメイン内のクライアントから入力します。

```
client% nisgrpadm -r admin @admin.doc.com.
Removed @admin.doc.com. from group admin.manf.doc.com.
```

NIS+ グループのメンバーかどうかを調べる

ある NIS+ 主体が特定の NIS+ グループのメンバーであるかどうかを調べるには、そのグループオブジェクトに対する読み取り権が必要です。-t オプションを使用します。

```
nisgrpadm -t group-name members. . .
```

次の例では、NIS+ 主体 topadmin が admin.doc.com. グループに所属するかどうかを調べます。これは、doc.com. ドメイン内のクライアントから入力します。

```
client% nisgrpadm -t admin topadmin
topadmin.doc.com. is a member of group admin.doc.com.
```

次の例では、sales.doc.com. ドメインの NIS+ 主体 jo が admin.sales.doc.com. グループに所属するかどうかを調べます。これは、doc.com. ドメイン内のクライアントから入力します。

```
client% nisgrpadm -t admin.sales.doc.com. jo.sales.doc.com.  
jo.sales.doc.com. is a member of group admin.sales.doc.com.
```


第 18 章

NIS+ ディレクトリの管理

この章では、NIS+ ディレクトリオブジェクトとその管理方法について説明します。

注 - NIS+ は、将来のリリースではサポートされなくなる可能性があります。NIS+ から LDAP への移行支援ツールは、Solaris 9 リリース以降で使用できます (『Solaris のシステム管理 (ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』を参照)。詳細については、<http://www.sun.com/directory/nisplus/transition.html> を参照してください。

NIS+ ディレクトリ

NIS+ ディレクトリオブジェクトは、NIS+ ドメインに関連する情報の格納に使用されます。NIS+ ドメインには、それぞれ NIS+ ディレクトリ構造が関連付けられます。NIS+ ディレクトリの詳細については、第 2 章を参照してください。

NIS+ ディレクトリ関連のコマンドとその構文、オプションについては、`nis+ (1)` のマニュアルページを参照してください。

ディレクトリについて niscat コマンドを使用する

`niscat -o` コマンドを使えば、NIS+ ディレクトリのオブジェクト属性を表示できます。このコマンドを使うには、ディレクトリオブジェクトの読み取り権が必要です。

ディレクトリのオブジェクト属性を表示する

ディレクトリのオブジェクト属性を表示するには、`niscat -o` とディレクトリ名を使います。

```
niscat -o directory-name
```

たとえば、以下のようになります。

```
rootmaster# niscat -o doc.com.  
Object Name : doc  
Owner : rootmaster.doc.com.  
Group :  
Domain : Com.  
Access Rights : r---rmdr---r---  
Time to Live : 24:0:0  
Object Type : DIRECTORY  
.  
.
```

nisls コマンドでディレクトリを表示する

`nisls` コマンドは、NIS+ ディレクトリの内容を表示します。このコマンドを使うには、ディレクトリオブジェクトに対する読み取り権が必要です。

簡潔形式での表示

```
nisls [-dgLmMR] directory-name
```

詳細形式での表示

```
nisls -l [-gm] [-dLMR] directory-name
```

表 18-1 `nisls` コマンドのオプション

オプション	目的
-d	ディレクトリオブジェクト。ディレクトリの内容を表示するのではなく、別のオブジェクトのように扱う
-L	リンク。ディレクトリ名が実際にはリンクである場合、コマンドはリンクをたどり、リンクされたディレクトリの情報を表示する
-M	マスター。マスターサーバーだけから情報を取得する。これによって最新の情報が得られるが、マスターサーバーが使用中の場合は長時間を要することがある

表 18-1 nislsls コマンドのオプション (続き)

オプション	目的
-R	再帰的 (recursive)。ディレクトリを再帰的に表示する。つまり、ディレクトリ中にほかのディレクトリがある場合、それらの内容も表示する
-l	長形式。情報を長形式で表示する。長形式では、オブジェクトのタイプ、作成時間、所有者、およびアクセス権を表示する
-g	グループ。情報を長形式で表示するとき、ディレクトリ所有者ではなく、そのグループ所有者を表示する
-m	変更時間。情報を長形式で表示するとき、ディレクトリ作成時間ではなく、その変更時間を表示する

ディレクトリの内容を表示する (簡潔形式)

ディレクトリ内容をデフォルトの短形式で表示するには、次に示すオプションの内の 1 つまたは複数と、ディレクトリ名を使います。ディレクトリ名を指定しない場合、NIS+ はデフォルトのディレクトリを使います。

```
nislsls [-dLMR] directory-name
```

または

```
nislsls [-dLMR]
```

たとえば、次の nislsls の場合は、ルートドメイン doc.com. のルートマスターサーバーから入力します。

```
rootmaster% nislsls doc.com.:
org_dir
groups_dir
```

次に、ルートマスターサーバーから入力した別の例を示します。

```
rootmaster% nislsls -R sales.doc.com.
sales.doc.com.:
org_dir
groups_dir
groups_dir.sales.doc.com.:
admin
org_dir.sales.doc.com.:
auto_master
auto_home
bootparams
cred
.
```

ディレクトリの内容を表示する (詳細形式)

ディレクトリの内容を詳細形式で表示するには `-l` オプション、および次に示すオプションのうちの1つまたは複数を使います。`-g` と `-m` のオプションは、表示される属性を変更します。`-g` と `-m` のオプションは、表示される属性を変更します。ディレクトリ名を指定しない場合、NIS+ はデフォルトのディレクトリを使います。

```
nisls -l [-gm] [-dLMR] directory-name
```

または

```
nisls -l [-gm] [-dLMR]
```

次に示すのは、ルートドメイン `doc.com.` のマスターサーバーから入力した例です。

```
rootmaster% nisls -l
doc.com.
D r---rmdr---r--- rootmaster.doc.com. date org_dir
D r---rmdr---r--- rootmaster.doc.com. date groups_dir
```

nismkdir コマンド

注 - この節では、`nismkdir` コマンドを使用して既存のドメインにルート以外のサーバーを追加する方法を説明します。`nisserver` スクリプトを使用すれば、より簡単に追加できます。方法については、[第4章](#)を参照してください。

`nismkdir` コマンドは、ルート以外のNIS+ ディレクトリを作成し、これをマスターサーバーに関連付けます。ルートディレクトリを作成するには、[339 ページ](#)の「`nisinit` コマンド」で説明する `nisinit -r` コマンドを使います。`nismkdir` コマンドを使って、複製サーバーを既存のディレクトリに追加できます。

NIS+ ディレクトリを作成するには、いくつかの関連作業のほかに、いくつかの前提条件があります。

ディレクトリを作成するには、以下のように入力します。

```
nismkdir [-m master-server] \  
  directory-name
```

複製を既存のディレクトリに追加するには、以下のように入力します。

```
nismkdir -s replica-server \  
  directory-name  
nismkdir -s replica-server \  
  org_dir.directory-name
```

```
nismkdir -s replica-server \  
groups_dir.directory-name
```

ディレクトリを作成する

ディレクトリを作成するには、(ドメインのマスターサーバー上の) 親ディレクトリに対する作成権が必要です。まず -m オプションを使ってマスターサーバーを定義し、次に -s オプションを使って複製を定義します。

```
nismkdir -m master directory  
nismkdir -s replica directory
```

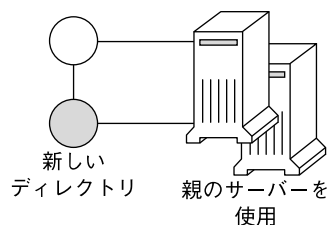


注意 -nismkdir は必ず (複製サーバーではなく) マスターサーバー上で実行してください。複製サーバー上でnismkdirを実行しないでください。複製サーバーで実行すると、マスターサーバーと複製サーバーの間で通信上の問題が発生します。

この例では、sales.doc.com. ディレクトリを作成してから、そのマスターサーバー smaster.doc.com. と複製サーバー repl.doc.com. を指定します。これはルートマスターサーバーから入力します。

```
rootmaster% nismkdir -m smaster.doc.com. sales.doc.com.  
rootmaster% nismkdir -m smaster.doc.com. org_dir.sales.doc.com.  
rootmaster% nismkdir -m smaster.doc.com. groups_dir.sales.doc.com.  
rootmaster% nismkdir -s repl.doc.com. sales.doc.com.  
rootmaster% nismkdir -s repl.doc.com. org_dir.sales.doc.com.  
rootmaster% nismkdir -s repl.doc.com. groups_dir.sales.doc.com.
```

名前空間 サーバー

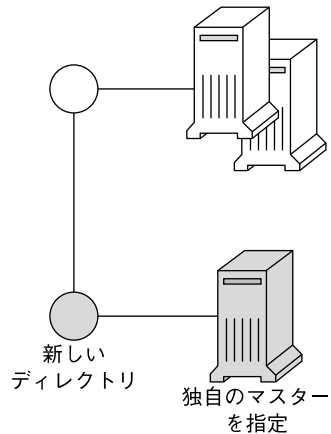


小規模またはテスト用の名前空間でないかぎりには推奨しませんが、nismkdir コマンドを使えば、独自のサーバーを指定する代わりに、親ディレクトリのサーバーを新しいディレクトリ用に使えます。次に 2 つの例を示します。

最初の例では、sales.doc.com. ディレクトリを作成し、これをその親ディレクトリのマスターと複製サーバーに関連付けます。

```
rootmaster% nismkdir sales.doc.com
```

名前空間 サーバー



2 番目の例では、sales.doc.com. ディレクトリを作成し、独自のマスターサーバーである smaster.doc.com. を指定します。

```
rootmaster% nismkdir -m smaster.doc.com. sales.doc.com.
```

複製サーバーは指定されないため、nismkdir を再び使用して複製を割り当てるまでは、新しいディレクトリにはマスターサーバーしかありません。sales.doc.com. ドメインがすでに存在する場合、上記の nismkdir コマンドは、salesmaster.doc.com. をその新しいマスターサーバーとし、その古いマスターサーバーを複製サーバーに格下げします。

複製サーバーを既存のディレクトリに追加する

この章では nismkdir コマンドを使用して複製サーバーを既存のシステムに追加する方法を説明します。nisserver スクリプトを使用すると、より簡単に追加できます。

以下の点に注意してください。

- ルートドメインサーバーはルートドメイン (またはその一部) に置く
- サブドメインサーバーは、サブドメインの 1 つ上の階層の親ドメイン (またはその一部) に置く。たとえば、名前空間に prime という名前のルートドメインと sub1 という名前のサブドメインがある場合は、次のようになる
 - prime ドメインにサービスを提供するマスターサーバーと複製サーバーは、prime がルートドメインであるため、prime ドメインの一部になる
 - sub1 サブドメインにサービスを提供するマスターサーバーと複製サーバーも、prime が sub1 の親ドメインであるため、prime ドメインの一部になる

- マスターサーバーまたは複製サーバーから複数のドメインにサービスを提供することもできるが、そのような措置は避けた方がよい

新しい複製サーバーを既存のディレクトリに割り当てるには、マスターサーバーに対して `-s` オプションと既存のディレクトリ名 (`org_dir` と `groups_dir`) を指定した `nismkdir` を使います。

```
nismkdir -s replica-server existing-directory-name
nismkdir -s replica-server org_dir. existing-directory-name
nismkdir -s replica-server groups_dir. existing-directory-name
```

`nismkdir` コマンドは、ディレクトリがすでに存在することを知っているため、再び作成しません。単に、追加の複製サーバーを割り当てるだけです。たとえば、新しい複製サーバーマシン名が `rep1` である場合は、以下のように入力します。

```
rootmaster% nismkdir -s repl.doc.com. doc.com.
rootmaster% nismkdir -s repl.doc.com. org_dir.doc.com.
rootmaster% nismkdir -s repl.doc.com. groups_dir.doc.com.
```



注意 - `nismkdir` は必ず (複製サーバーではなく) マスターサーバー上で実行してください。複製サーバー上で `nismkdir` を実行しないでください。複製サーバーで実行すると、マスターサーバーと複製サーバーの間で通信上の問題が発生します。

上記の説明に従って `nismkdir` を 3 回繰り返して実行した後は、以下の 3 つのディレクトリでマスターサーバーから `nisping` を実行する必要があります。

```
rootmaster# nisping doc.com.
rootmaster# nisping org_dir.doc.com.
rootmaster# nisping group_dir.doc.com.
```

この結果は以下のようになります。

```
rootmaster# nisping doc.com.
Pinging replicas serving directory doc.com. :
Master server is rootmaster.doc.com.
Last update occurred at Wed Nov 18 19:54:38 1995
Replica server is repl.doc.com.
Last update seen was Wed Nov 18 11:24:32 1995
Pinging ... repl.doc.com
```

マスターサーバーの `cron` ファイルに、`nisping` コマンドが少なくとも 24 時間に一度 (この 3 つのディレクトリに対して) 実行されるよう設定しておくことをお勧めします。

nisrmdir コマンド

nisrmdir コマンドでは、ディレクトリを削除したり、ディレクトリと複製サーバーを切り離すことができます。ディレクトリを削除するかディレクトリと複製サーバーを切り離すと、その NIS+ ドメインに対して、マシンは NIS+ 複製サーバーとしては機能しなくなります。

ディレクトリの削除では、まずマスターサーバーと複製サーバーをディレクトリから切り離し、次にそのディレクトリを削除します。

- ディレクトリを削除するには、その親ディレクトリに対する削除権が必要です。
- ディレクトリから複製サーバーを切り離すには、そのディレクトリに対する変更権が必要です。

nisrmdir コマンドの実行で問題が生じる場合は、[437 ページの「複製の失敗からの NIS+ ディレクトリの削除または分離」](#)を参照してください。

ディレクトリを削除する

ディレクトリ全体を削除し、そのマスターサーバーと複製サーバーを切り離すには、nisrmdir コマンドをオプションなしで実行します。

```
nisrmdir directory-name  
nisping domain
```

次の例では、doc.com. ディレクトリの下の manf.doc.com. ディレクトリを削除します。

```
rootmaster% nisrmdir manf.doc.com.  
rootmaster% nisping doc.com.
```

複製サーバーをディレクトリから切り離す

複製サーバーをディレクトリから切り離すには、まず、ディレクトリの org_dir と groups_dir サブディレクトリを削除します。その時は、nisrmdir コマンドに -s オプションを付けて実行します。サブディレクトリが削除されたら、親ドメインに戻って nisping を実行してください。

```
nisrmdir -s replicanameorg_dir.domain  
nisrmdir -s replicanamegroups_dir.domain  
nisrmdir -s replicaname domain  
nisping domain
```

次の例では、manfreplical サーバーを manf.doc.com. ディレクトリから切り離します。


```
rootmaster% nisrmdir -s manfreplica1 org_dir.manf.doc.com.
rootmaster% nisrmdir -s manfreplica1 groups_dir.manf.doc.com.
rootmaster% nisrmdir -s manfreplica1 manf.doc.com.
rootmaster% nisping manf.doc.com.
```

`nisrmdir -s` コマンドを実行したが、切り離し対象の複製サーバーがダウンしていた、または通信が途絶していたという場合、「Cannot remove replica *name*:attempt to remove a non-empty table」というエラーメッセージが出されます。このような場合は、マスターサーバー上で `nisrmdir -f -s replicaname` コマンドを実行すれば、強制的に切り離すことができます。ただし、`nisrmdir -f -s` コマンドでダウン状態の複製サーバーを切り離した場合、その複製サーバーがオンラインに復帰したらただちに `nisrmdir -f -s` コマンドを再実行して複製サーバーの `/var/nis` ファイルシステムの中を整理する必要があります。`nisrmdir -f -s replicaname` を再実行しないと、複製がサービスを再開した時に複製上に残された古い情報によって問題が発生する場合があります。

nisrm コマンド

`nisrm` コマンドは、標準の `rm` システムコマンドと似ています。このコマンドは、ディレクトリと空でないテーブルを除いて、名前空間からすべての NIS+ オブジェクトを削除します。`nisrm` コマンドを使うには、オブジェクトに対する削除権が必要です。しかし、この権利がない場合、`-f` オプションを使うことができます。このオプションは、アクセス権がなくても動作を強行しようと試みます。

`nisgrpadm -d` コマンド (325 ページの「NIS+ グループを削除する」を参照) を使えばグループオブジェクトを削除でき、`nistbladm -r` または `nistbladm -R` (359 ページの「テーブルを削除する」を参照) を使えばテーブルを空にできます。

ディレクトリ以外のオブジェクトは、次のコマンドで削除します。

```
nisrm [-if] object-name
```

表 18-2 `nisrm` 構文オプション

オプション	目的
-i	照会。オブジェクトを削除する前に確認を要求する。指定されたオブジェクト名が完全指定されていない場合、このオプションが自動的に使われる
-f	強制。たとえ適切なアクセス権がない場合でも、削除の強行を試みる。 <code>nischmod</code> コマンドを使ってアクセス権の変更を試み、再度オブジェクトの削除を試みる

ディレクトリ以外のオブジェクトを削除する

ディレクトリ以外のオブジェクトを削除するには、`nisrm` コマンドを使い、オブジェクト名を指定します。

```
nisrm object-name...
```

次の例では、名前空間からグループとテーブルを削除します。

```
rootmaster% nisrm -i admins.doc.com. groups.org_dir.doc.com.  
Remove admins.doc.com.? y  
Remove groups.org_dir.doc.com.? y
```

rpc.nisd デーモン

NIS+ サービスを有効にすると、rpc.nisd デーモンが起動します。NIS+ デーモンを起動するにはアクセス権は不要ですが、そのすべての前提条件と関連作業を知っておく必要があります。前提条件については、[111 ページの「NIS+ サーバーを設定するための前提条件」](#)を参照してください。

注 - NIS+ サービスは、サービス管理機能 (SMF) によって管理されます。このサービスに対する有効化、無効化、再起動などの管理操作を実行するには、svcadm コマンドを使用します。SMF を NIS+ で使用する方法については、[89 ページの「NIS+ とサービス管理機能」](#)を参照してください。SMF の概要については、『Solaris のシステム管理 (基本編)』の「サービスの管理 (概要)」を参照してください。詳細については、svcadm(1M) と svcs(1) の各マニュアルページも参照してください。

rpc.nisd デーモンを起動する

NIS+ サービスを有効にすると、rpc.nisd デーモンが起動します。サービスを起動するには、svcadm enable コマンドを使用します。

```
rootmaster# svcadm enable /network/rpc/nisplus:default
```

rpc.nisd デーモンを停止する

NIS+ サービスを無効にすると、rpc.nisd デーモンが停止します。まず、svcs コマンドを使って FMRI とサービスの状態を確認します。次に、svcadm disable コマンドを使ってサービスを停止します。その際、サービスが通常モードで動作しているか、NIS 互換モードで動作しているかは関係ありません。

```
rootmaster# svcs \*nisplus\  
STATE      STIME      FMRI  
online      Oct_07      svc:/network/rpc/nisplus:default  
  
rootmaster# svcadm disable /network/rpc/nisplus:default
```

rpc.nisd 構文オプションを変更する

デフォルトでは、NIS+ デーモンはセキュリティレベル 2 で起動し、通常モードで動作します。サービス管理機能を使って `rpc.nisd` デーモンを呼び出すときに特定のオプションを指定する場合は、そのオプションを含むように `/lib/svc/method/nisplus` ファイルを変更します。詳細については、[89 ページ](#)の「NIS+ とサービス管理機能」を参照してください。

表 18-3 に、一般的な `rpc.nisd` 構文オプションをいくつか示します。

表 18-3 一般的な `rpc.nisd` 構文オプション

オプション	目的
-Y	デーモンが NIS 互換モードで動作することを指定します。これにより、デーモンは NIS クライアントからの要求に応答できます。ルートマスターを含む任意のサーバーで、NIS+ デーモンを NIS 互換モードで起動できます。
-B	NIS 互換モードで動作している NIS+ デーモンに DNS 転送機能を追加します。このオプションでは、 <code>/etc/resolv.conf</code> ファイルが DNS ネームサーバーとの通信用に設定されている必要があります。
-S security-level	セキュリティレベルを指定する。0 は「NIS+ セキュリティを使用しない」、2 は「最高レベルの NIS+ セキュリティを使用する」という意味（レベル 1 はサポートされない）
-F	デーモンによって提供されるディレクトリのチェックポイント設定を強行する。この動作は、ディレクトリのトランザクションログを空にして、ディスク空間を解放することになる

nisinit コマンド

この節では、`nisinit` コマンドを使用してマシンクライアントを初期設定する方法について説明します。`nisclient` スクリプトを使用すると、より簡単に初期設定できます。[106 ページ](#)の「NIS+ クライアントマシンの設定」を参照してください。

`nisinit` コマンドは、NIS+ クライアントまたはサーバーとなるマシンを初期設定します。`rpc.nisd` コマンドと同様、`nisinit` コマンドを使うにはアクセス権は不要ですが、その前提条件と関連作業を知っておく必要があります。[156 ページ](#)の「NIS+ クライアントを初期設定する」を参照してください。

クライアントを初期設定する

クライアントを初期設定するには、次の 3 つの方法があります。

- ホスト名による方法

- ブロードキャストによる方法
- コールドスタートファイルによる方法

前提条件と関連作業は、方法によってそれぞれ異なります。たとえば、ホスト名によってクライアントを初期設定するには、その前にクライアントの `/etc/hosts` ファイルまたは `/etc/inet/ipnodes` ファイルに、使用するホスト名を登録しなければなりません。また `nsswitch.conf` ファイルの `hosts` の最初の選択肢として、`files` を指定する必要があります。IPv6 アドレスには、`hosts` の最初の選択肢として `ipnodes` を指定してください。nisinit コマンドを使う手順を次にまとめてみます。

ホスト名によってクライアントを初期設定するには、`-c` と `-H` のオプションを使い、クライアントがそのコールドスタートファイルを取得するサーバー名を指定します。

```
nisinit -c -H hostname
```

コールドスタートファイルによってクライアントを初期設定するには、`-c` と `-C` のオプションを使い、コールドスタートファイル名を指定します。

```
nisinit -c -C filename
```

ブロードキャストによってクライアントを初期設定するには、`-c` と `-B` のオプションを使います。

```
nisinit -c -B
```

ルートマスターサーバーを初期設定する

ルートマスターサーバーを初期設定するには、`nisinit -r` コマンドを使います。

```
nisinit -r
```

ここで次の情報が必要になります。

- ルートマスターサーバーとなるマシンのスーパーユーザーパスワード
- 新しいルートドメイン名。ルートドメイン名は少なくとも2つの要素(ラベル)で構成され、その末尾にドット(ピリオド)が打たれていなければならない(例: `something.com`)。最後の要素はインターネットの組織ドメイン(表 18-4 を参照)か、2~3文字の地域識別子(日本であれば `.jp`)のどちらかにするように決められている

表 18-4 インターネットの組織ドメイン

ドメイン	目的
com	営利団体
edu	教育機関

表 18-4 インターネットの組織ドメイン (続き)

ドメイン	目的
gov	行政機関
mil	軍事組織
net	主要ネットワークサポートセンター
org	非営利団体
int	国際組織

nis_cachemgr デーモン

nis_cachemgr は、すべての NIS+ クライアント上で動作する必要があります。キャッシュマネージャは、名前空間でもっともよく使われるディレクトリをサポートする NIS+ サーバーの位置情報キャッシュを管理します。位置情報はトランスポートアドレス、認証情報、生存期間値などです。

キャッシュマネージャが起動すると、クライアントのコールドスタートファイルから初期情報を取得し、それを /var/nis/NIS_SHARED_DIRCACHE ファイルにダウンロードします。

キャッシュマネージャは、クライアントマシンとして要求を行います。クライアントマシンには必ず適切な資格を持たせてください。そうしないと、性能が向上するどころか、キャッシュマネージャが性能を低下させます。

キャッシュマネージャの起動と停止

サービス管理機能 (SMF) の使用時は、キャッシュマネージャは NIS+ サービスに依存しているため、NIS+ サービスとともに起動または停止します。NIS+ サービスを起動、停止、または再起動するには、svcadm コマンドを使用します。

```
client% svcadm enable /network/rpc/nisplus:default
client% svcadm disable /network/rpc/nisplus:default
client% svcadm restart /network/rpc/nisplus:default
```

NIS+ サービスをいったん停止してから起動すると、キャッシュマネージャが再起動されますが、/var/nis/NIS_SHARED_DIRCACHE ファイル内の情報は保持されます。コールドスタートファイル内の情報は、キャッシュファイル内の既存の情報にそのまま追加されます。キャッシュファイルをクリアし、クライアントのコールドスタートファイルの内容からこれを再び初期設定するには、-i オプションを使用します。

nisshowcache コマンド

nisshowcache コマンドは、クライアントのディレクトリキャッシュの内容を表示します。

NIS+ キャッシュの内容を表示する

nisshowcache コマンドは、/usr/lib/nis 内にあります。このコマンドはキャッシュヘッダーとディレクトリ名だけを表示します。ルートマスターサーバーから入力した例を次に示します。

```
rootmaster# /usr/lib/nis/nisshowcache -v
Cold Start directory:
Name : doc.com.
Type : NIS
Master Server :
  Name : rootmaster.doc.com.
  Public Key : Diffie-Hellman (192 bits)
  Universal addresses (3)
  .
Replicate:
  Name : rootreplica1.doc.com.
  Public Key : Diffie-Hellman (192 bits)
  Universal addresses (3)
  .
  .
Time to live : 12:0:0
Default Access Rights :
```

ping とチェックポイントを実行する

NIS+ データセットに変更を加えると、その変更は、当該 NIS+ ドメイン (またはサブドメイン) のマスターサーバーのメモリーに格納されます。変更の記録はマスターサーバーのトランザクションログ (/var/nis/data/trans.log) にも残されます。

通常、NIS+ データセットに変更が加えられると、その 120 秒 (2 分) 後に、マスターサーバーから当該ドメインの複製サーバーにその変更の内容が転送されます。この転送プロセスのことを「ping」といいます。マスターサーバーから複製サーバーへの ping が実行されると、通知された変更内容に従って複製サーバーのデータセットが更新されます。これにより、変更された NIS+ データがマスターサーバーと複製サーバーの両方のメモリーに格納されることになります。

自動 ping プロセスが不調で複製サーバーのデータセットが更新されないこともあります。その場合は、[344 ページの「ping を強制的に実行する」](#)の説明に従って ping を強制的に実行する必要があります。複製サーバーが最新の NIS+ データどおりに正しく更新されているかどうか不安な場合は、[343 ページの「最新更新時間を表示する」](#)の説明に従って複製サーバーが最後に更新されたのはいつであるかを確認してください。

NIS+ データセットに対する変更は、サーバーのメモリーに格納され、トランザクションログに記録されたのち、ディスク上の NIS+ テーブルに書き込まれなければなりません。この NIS+ テーブルを更新することを「チェックポイントを実行する」といいます。

チェックポイントの実行は自動的には行われません。[344 ページの「ディレクトリにチェックポイントを実行する」](#)の説明に従ってチェックポイントコマンドを実行する必要があります。

nisping コマンド

nisping コマンドには次の用途があります。

- 複製サーバーが最後に ping されたのはいつであるかを表示する (詳細は、[343 ページの「最新更新時間を表示する」](#)を参照)
- 自動 ping サイクルが不調に終わった場合、マスターサーバーから複製サーバーへの ping を強制的に実行する (詳細は、[344 ページの「ping を強制的に実行する」](#)を参照)
- サーバーにチェックポイントを実行する (詳細は、[344 ページの「ディレクトリにチェックポイントを実行する」](#)を参照)

最新更新時間を表示する

-u オプションを指定して nisping コマンドを実行すると、ローカルドメインのマスターサーバーと複製サーバーが更新された時間が表示されます。

```
/usr/lib/nis/nisping -u [domain]
```

ほかのドメインで最後に更新が行われた時間を表示するには、コマンド行にそのドメイン名を指定します。-u オプションを指定して nisping コマンドを実行した場合、複製サーバーに対する ping は一切実行されない点に注意してください。

たとえば、ローカルドメイン doc.com. で複製サーバーが最後に更新された時間を表示するには、次のように入力します。

```
rootmaster# /usr/lib/nisping -u
Last updates for directory doc.com.:
Master server is rootmaster.doc.com.
Last update occurred at Wed Nov 25 10:53:37 1992
```

```
Replica server is rootreplica1.doc.com.  
Last update seen was Wed Nov 25 10:53:37 1992
```

ping を強制的に実行する

-u オプションを指定して `nisping` コマンドを実行した結果、複製サーバーが正しく更新されていないことがわかったとします。そのような場合は、`nisping` コマンドを実行することにより、マスターサーバーからドメイン内のすべての複製サーバーに対して、または特定の複製サーバーに対して、`ping` を強制的に実行できます。

すべての複製サーバーを `ping` の対象とする場合、次に示すように、`nisping` コマンドをオプションなしで実行します。

```
/usr/lib/nis/nisping
```

これにより、マスターサーバーからドメイン内のすべての複製サーバーへの `ping` が強制的に実行されます。ローカルドメイン `doc.com.` のすべての複製サーバーに対する `ping` を強制的に実行するには、次のように入力します。

```
rootmaster# /usr/lib/nis/nisping  
Pinging replicas serving directory doc.com.:  
Master server is rootmaster.doc.com.  
Last update occurred at Wed Nov 25 10:53:37 1992  
Replica server is rootreplica1.doc.com.  
Last update seen was Wed Nov 18 11:24:32 1992  
Pinging ... rootreplica1.doc.com.
```

ローカルドメインでないドメイン内のすべての複製サーバーに対し `ping` を実行するには、ドメイン名を指定します。

```
/usr/lib/nis/nisping domainname
```

特定の 1 台のホスト上のすべてのディレクトリにあるすべてのテーブルに対し `ping` を実行することもできます。この場合、`-a` オプションを使用します。

```
/usr/lib/nis/nisping -a hostname
```

ディレクトリにチェックポイントを実行する

スワップの頻度やディスク領域との関連でトランザクションログが大きくなりすぎる場合、各ドメインおよびサブドメインに対しては、少なくとも 24 時間に 1 回はチェックポイントを実行する必要があります。

注-大きなドメイン、または大きなトランザクションログを持つドメインに対してチェックポイントを実行すると、終了するまでにかなりの時間がかかり、その間NIS+ サーバーが拘束され、NIS+ サービスの速度が低下します。サーバーは、チェックポイントを実行している間もサービス要求に応答しますが、更新できません。できるだけ、システムが混んでいない時間帯を選んでチェックポイントを実行することをお勧めします。チェックポイントの実行スケジュールは cron ファイルで調整できます。

チェックポイントを実行するには、当該ドメインのマスターサーバー上で `nisping -c` コマンドを実行します。先にすべての複製サーバーに対して `ping` を実行してから、チェックポイントを実行することをお勧めします。その場合には、複製サーバーに対して常に最新のデータでチェックポイントを実行できます。

- 特定のディレクトリに対してチェックポイントを実行するには、次に示すように、`-c` オプションを指定して `nisping` コマンドを実行します。たとえば、次のように指定します。

```
rootmaster# /usr/lib/nis/nisping
rootmaster# /usr/lib/nis/nisping -c org_dir
```

- ローカルドメイン内のすべてのディレクトリに対してチェックポイントを実行するには、次に示すように、`-c -a` オプションを指定して `nisping` コマンドを実行します。

```
rootmaster# /usr/lib/nis/nisping
rootmaster# /usr/lib/nis/nisping -c -a
```

サーバーによってそのトランザクションログの情報がNIS+ テーブルに転送されると、ログファイル内のトランザクションは、ディスク領域の節約のため消去されます。

`doc.com`. ドメイン内のすべてのディレクトリに対してチェックポイントを実行するには、次のように入力します。

```
rootmaster# /usr/lib/nis/nisping -c -a
Checkpointing replicas serving directory doc.com. :
Master server is rootmaster.doc.com.
  Last update occurred at Wed May 25 10:53:37 1995
Master server is rootmaster.doc.com.
checkpoint has been scheduled with rootmaster.doc.com.
Replica server is rootreplica1.doc.com.
  Last update seen was Wed May 25 10:53:37 1995
Replica server is rootreplica1.doc.com.
checkpoint has been scheduled with rootmaster.doc.com.
```

nislog コマンド

nislog コマンドは、トランザクションログの内容を表示します。

```
/usr/sbin/nislog
/usr/sbin/nislog -h [number]
/usr/sbin/nislog -t [number]
```

表 18-5 nislog コマンドのオプション

オプション	目的
-h [num]	ログの先頭からトランザクションを表示する。数字を指定しなければ、最初のトランザクションから表示が開始される。0 を指定すると、ログヘッダーだけが表示される
-t [num]	ログの末尾からトランザクションを表示する。数字を指定しなければ、最後のトランザクションから表示が開始される。0 を指定すると、ログヘッダーだけが表示される
-v	冗長モード。

トランザクションログの内容を表示する

各トランザクションは、トランザクションの明細部とオブジェクト定義のコピー部の 2 つの部分から構成されます。

doc.com. ディレクトリが最初に作成されたときに作られたトランザクションログエントリを次の例に示します。「XID」はトランザクション ID を意味します。

```
rootmaster# /usr/sbin/nislog -h 1
NIS Log printing facility.
NIS Log dump:
  Log state : STABLE
Number of updates : 48
Current XID : 39
Size of log in bytes : 18432
***UPDATES***
@@@@@@@@@@@@@@@@TRANSACTION@@@@@@@@@@@@@@@@
#00000, XID : 1
Time : Wed Nov 25 10:50:59 1992
Directory : doc.com.
Entry type : ADD Name
Entry timestamp : Wed Nov 25 10:50:59 1992
Principal : rootmaster.doc.com.
Object name : org_dir.doc.com.
.....Object.....
Object Name : org_dir
Owner : rootmaster.doc.com.
```

```

Group : admin.doc.com.
Domain : doc.com.
Access Rights : r---rmcdr---r---
Time to Live : 24:0:0
Object Type : DIRECTORY
Name : 'org_dir.doc.com.'
Type: NIS
Master Server : rootmaster.doc.com.
.
.
.....
@@@@@@@@@@@@@@@@TRANSACTION@@@@@@@@@@@@@@@@
#00000, XID : 2

```

nischttl コマンド

nischttl コマンドは、名前空間内のオブジェクトまたはエントリの生存期間値を変更します。キャッシュマネージャは、この生存期間値を使って、キャッシュエントリをいつ期限切れにするかを決めます。生存期間を指定するには、合計秒数か、日、時間、分、秒の組み合わせのいずれかを使います。

オブジェクトまたはエントリに割り当てる生存期間値は、オブジェクトの安定性に左右されます。よく変化するオブジェクトの場合、生存期間値を低くします。安定したオブジェクトの場合、高い値を指定します。高い生存期間値であれば、1 週間などを指定し、低い値であれば 1 分未満を指定します。パスワードエントリの生存期間値は約 12 時間とし、1 日に 1 回パスワード変更ができるようにする必要があります。RPC テーブル内のエントリなど、あまり変化しないテーブルのエントリには、数週間の値を設定できます。

オブジェクトの生存期間を変更するには、そのオブジェクトに対する変更権が必要です。テーブルエントリの生存期間を変更するには、テーブルに対する変更権が必要であり、テーブルがなければエントリに対して、エントリもなければ変更したい列に対して変更権が必要です。

オブジェクトまたはテーブルエントリの現在の生存期間値を表示するには、[第 15 章](#)で説明する `nisdefaults -t` コマンドを使います。

オブジェクトの生存期間値を変更するには、以下のどちらかのように入力します。

```
nischttl time-to-live object-name
```

または

```
nischttl [-L] time-to-live object-name
```

エントリの生存期間値を変更するには、以下のいずれかのように入力します。

```
nischttl time-to-live \
[column=value,...], \
```

table-name

または

```
nischtttl [-ALP] time-to-live \
[column=value,...], \
table-name
```

time-to-live には以下のものを指定します。

- 「秒」。数字だけで文字を指定しなければ、単位が秒であると解釈される (例: 1234 は 1,234 秒)。数字の後に *s* をつけると、単位が秒であると解釈される (例: 987*s* は 987 秒)。日、時間、分などとともに指定する場合は、秒であることを明らかにするため *s* をつける
- 「分」。数字の後に *m* をつけると、単位が分であると解釈される (例: 90*m* は 90 分)
- 「時間」。数字の後に *h* をつけると、単位が時間であると解釈される (例: 9*h* は 9 時間)
- 「日」。数字の後に *d* をつけると、単位が日であると解釈される (例: 7*d* は 7 日)

以上の値は組み合わせて使うことができます。たとえば「4*d*3*h*2*m*1*s*」は、4 日 3 時間 2 分 1 秒を意味します。

nischtttl コマンドでは以下のフラグを使用できます。

表 18-6 *nischtttl* 構文のオプション

オプション	目的
A	すべて。 <i>[column=value]</i> 指定に合致する全エントリに変更を適用する
L	リンク。リンクをたどり、リンク自体ではなく、リンクされたオブジェクトまたはエントリを変更する
P	パス。条件を満足するエントリが 1 つ見つかるまで、パスをたどる

オブジェクトの生存期間を変更する

オブジェクトの生存期間を変更するには、生存期間の値とオブジェクト名を指定して *nischtttl* コマンドを入力します。*-L* コマンドを追加すれば、リンクされたオブジェクトにまで変更を拡張できます。

```
nischtttl -L time-to-live object-name
```

生存期間は秒か、日、時間、分、秒の組み合わせかで指定できます。前者の場合、秒数だけを指定します。後者の場合、日数、時間数、分数と秒数に「*s*、*m*、*h*、*d*」をつけてください。たとえば、以下ようになります。

```
client% nischtttl 86400 sales.doc.com.
client% nischtttl 24h sales.doc.com.
client% nischtttl 2d1h1m1s sales.doc.com.
```

最初の2つでは、`sales.doc.com.` ディレクトリの生存期間が86,400秒、つまり24時間に変更されます。3つめでは、`hosts` テーブル内の全エントリの生存期間が176,461秒、つまり2日と1時間1分1秒に変更されます。

テーブルエントリの生存期間を変更する

エントリの生存期間を変更するには、インデックス付きのエントリフォーマットを使います。`-A`、`-L`、または`-P`のどのオプションでも使えます。

```
nischttl [-ALP] time-to-live \  
  [column=value,...], \  
  table-name
```

注 - Cシェルを使用している場合、`[ ]`がメタキャラクタとして解釈されないように引用符で囲みます。

次に示す例は上記の例と似ていますが、オブジェクトではなく、テーブルエントリの値を変更します。

```
client% nischttl 86400 '[uid=99],passwd.org_dir.doc.com.'  
client% nischttl 24h '[uid=99],passwd.org_dir.doc.com.'  
client% nischttl 2d1h1m1s '[name=fred],hosts.org_dir.doc.com.'
```


第 19 章

NIS+ テーブルの管理

この章では、NIS+ テーブルとその管理方法について説明します。デフォルトの NIS+ テーブルの詳細については、表 10-1 を参照してください。

注 - NIS+ は、将来のリリースではサポートされなくなる可能性があります。NIS+ から LDAP への移行支援ツールは、Solaris 9 リリース以降で使用できます (『Solaris のシステム管理 (ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』を参照)。詳細については、<http://www.sun.com/directory/nisplus/transition.html> を参照してください。

NIS+ のテーブル

NIS+ で使用される情報は、NIS+ テーブルに格納されます (Solaris ソフトウェアで提供されるデフォルトの NIS+ システムテーブルについては、第 23 章を参照)。

NIS+ テーブル関連のコマンドと、その構文およびオプションの詳細については、NIS+ のマニュアルページを参照してください。

テーブルでの nistbladm コマンドの使用

注 - NIS+ テーブルに関連した作業には、可能な場合、Solaris 管理コンソール ツールを使用するとより簡単にできるものがあります。

nistbladm コマンドは NIS+ テーブル管理コマンドの中でもっとも重要なコマンドであり、NIS+ ディレクトリオブジェクトに格納されている NIS+ テーブル上で使います。nistbladm コマンドを使うと、NIS+ テーブルまたはそのエントリを作成、修正、削除できます。ただし、ディレクトリのないところにテーブルを作成はできません。同様に、テーブルと列が定義されていないところにエントリを追加できません。

テーブルを作成するには、そのテーブルが所属するディレクトリに対する作成権が必要です。テーブルを削除するには、そのディレクトリに対する削除権が必要です。テーブルの内容を変更 (エントリの追加、変更、または削除) するには、そのテーブルまたはエントリの変更権が必要です。

nistbladm 構文

nistbladm コマンドの一般的な構文は次のとおりです。

```
nistbladm options \  
[columnspec | columnvalue] \  
[tablename | indexedname]
```

引数の意味はそれぞれ以下のとおりです。

- *columnspec* には、テーブル内に作成する列を指定する (具体的な指定方法については、357 ページの「[テーブルの列を指定する](#)」を参照)
- *columnvalue* には、*tablename* によって表されるテーブルの中の特定のセルを指定する (具体的な指定方法については、353 ページの「[nistbladm と列の値](#)」を参照)
- *tablename* には、テーブル名 (*hosts.org_dir.doc.com.* など) を指定する
- *indexedname* には、特定のテーブルの中の特定のセル値を指定する (具体的な指定方法については、353 ページの「[nistbladm と列の値](#)」を参照)。 *indexedname* は本質的に、*columnvalue* と *tablename* の役割を兼ね備えている

表 19-1 nistbladm コマンドのオプション

オプション	説明
-a -A	既存の NIS+ テーブルにエントリを追加する。-a を指定した場合、nistbladm コマンドを実行すると既存のエントリが上書きされる場合には、エラーが返される。-A を指定した場合、nistbladm コマンドが既存エントリを上書きする場合でも強制的に実行される (359 ページの「エントリをテーブルに追加する」を参照)
-D defaults	別のデフォルト特性を使ってオブジェクトを作成する (詳細については、デフォルトの nistbladm(1) のマニュアルページを参照)
-d	テーブルを破棄する (359 ページの「テーブルを削除する」を参照)
-c	テーブルを作成する (356 ページの「テーブルを作成する」を参照)
-r -R	既存の NIS+ テーブルからエントリ (1 つまたは複数) を削除する。-r を指定した場合、複数のエントリの削除につながる nistbladm コマンドは実行されず、エラーが返される。-R を指定した場合、複数のエントリの削除につながる nistbladm コマンドであっても強制的に実行される (364 ページの「テーブルからエントリを削除する」を参照)
-m	テーブルエントリを修正するためのオプション。旧リリースとの互換性を維持するためにだけ残されている。エントリを修正するのであれば、-e オプションまたは -E オプションを使うほうが望ましい
-e -E	既存の NIS+ テーブルのエントリを修正する。-e を指定した場合、複数のエントリの変更につながる nistbladm コマンドは実行されず、エラーが返される。-E を指定した場合、複数のエントリの変更につながる nistbladm コマンドであっても強制的に実行される (362 ページの「エントリを修正する」を参照)

nistbladm と列の値

列の値は、テーブル内の個々のエントリを識別するために使われます。列の値の書式は次のとおりです。

```
columnname="value", \  
columnname="value", ...
```

引数の意味はそれぞれ以下のとおりです。

- *columnname* はテーブルの列名を示す
- *value* は列内の特定のセルの内容 (値) を示す。この値により、テーブルの行を識別することができる (*column=value* を指定してテーブルデータを作成または修正する場合は、必ず *value* を引用符で囲むこと)

たとえば、マシン名と IP アドレスを登録した *hosts* という名前のテーブルがあるとします。

表 19-2 hosts テーブルの例

IP アドレス	名前	別名
172.22.168.4	altair	
172.22.168.119	deneb	mail
172.22.168.120	regulus	dnsmaster
172.22.168.121	regulus	dnsmaster
172.22.168.11	sirius	

このサンプルテーブルの altair エントリ (行) を識別するには、次の 3 通りの *column=value* の指定方法が考えられます。

- name=altair
- address=172.22.168.4
- name=altair,address=172.22.168.4

上記のテーブルで注目すべき点としては、regulus という名前のマルチホームマシンに 2 つの IP アドレスが割り当てられていることです。この場合、ホストマシン *regulus* の *column=value* は 2 つの行を示します。そこで、最初の *regulus* 行だけを識別したいという場合は次のいずれかを入力します。

- address=172.22.168.120 or
- address=172.22.168.120,name=regulus, dnsmaster

注 - *nistbladm* コマンドの一部のオプションには、テーブル内のすべての列に *column=value* を指定しなければならないものがあります。

nistbladm、検索可能列、キー、列の値

NIS+ テーブルを作成する際は、*s* フラグまたは *i* フラグを指定し、1 つまたは複数の列が検索可能になるようにします (357 ページの「[テーブルの列を指定する](#)」を参照)。なお、*niscat -o tablename* と入力すれば、テーブルの列とその特性を一覧表示できます。

テーブル内で行の「識別」に使われるのが検索可能列です。検索可能列の値 (値の組み合わせ) はすべて一意でなければなりません。したがって、検索可能列が 1 つしかないテーブルの場合、各行の検索可能列の値はすべて一意でなければならず、重複は一切許されません。

ここで、*city* という名前の検索可能列と *country* という名前の検索不可列からなるテーブルを作成する場合を想定します。次に示すのは、正しい (検索可能列の値の組み合わせに重複がない) テーブルの例です。

City	Country
San Francisco	United States
Santa Fe	United States
Santiago	Chile

次は正しくない (city 列に重複がある) テーブルの例です。

City	Country
London	Canada
London	England

1 つのテーブルに検索可能列が複数ある場合は、検索可能列どうしの値の組み合わせが一意であればかまいません。ここでは、Lastname および Firstname という 2 つの検索可能列と、city という検索不可列からなるテーブルを作成する場合を想定します。次に示すのは、正しい (検索可能列の値の組み合わせに重複がない) テーブルの例です。

Lastname	Firstname	City
Kuznetsov	Sergei	Odessa
Kuznetsov	Rima	Odessa
Sergei	Alex	Odessa

次は正しくない (検索可能列の値の組み合わせに重複がある) テーブルの例です。

Lastname	Firstname	City
Kuznetsov	Rima	Odessa
Kuznetsov	Rima	Chelm

NIS+ のコマンドはいずれも、検索可能列の値に基づいて特定の行を識別します。

nistbladm とインデックス名

NIS+ には、テーブル名と列の値の検索基準の組み合わせ (これを「インデックス名」という) によって特定のテーブルの特定のエントリを識別する、というテーブル管理の方法もあります。インデックス名の書式は次のとおりです。

`[search_criteria] , tablename . directory`

`search_criteria` には検索基準を指定しますが、このとき角カッコ ([]) で囲むのを忘れないでください。書式は次のとおりです。

```
columnname=value, \  
columnname=value, ...
```

`columnname=value` にはテーブルの検索可能列の値を指定します (353 ページの「`nistbladm` と列の値」を参照)。

たとえば、表 19-2 の `altair` エントリを識別するには、次のようにインデックス名を指定します。

```
[addr=172.22.168.4,cname=altair],hosts.org_dir.doc.com.
```

`nistbladm -R` コマンドを使用すると、間になにも入れない角カッコ [] をすべてのテーブル列を指定するワイルドカードとして使用し、テーブル中のすべてのエントリを一度に削除できます。

nistbladm とグループ

Solaris の NIS+ 環境には 3 種類のグループがあります。

- 「UNIX グループ」。UNIX グループに関する情報は `groups.org_dir` テーブルに格納される。UNIX グループ情報の管理には `nistbladm` コマンドを使う
- 「ネットグループ」。ネットグループに関する情報は `netgroups.org_dir` テーブルに格納される。ネットグループ情報の管理には `nistbladm` コマンドを使う
- 「NIS+ グループ」。NIS+ グループに関する情報は `groups_dir` ディレクトリオブジェクトのテーブル (1 つまたは複数) に格納される。NIS+ グループ情報の管理には `nisgrpadm` コマンドを使う

注 – NIS+ グループの管理に `nistbladm` を使うことはできません。

その他のグループの詳細は、319 ページの「Solaris グループ」を参照してください。

テーブルを作成する

NIS+ テーブルには、少なくとも 1 つの列が必要で、その列のうち少なくとも 1 つが検索可能でなければなりません。NIS+ テーブルを作成するには、`nistbladm` コマンドに `-c` オプションを付けて使います。

```
nistbladm -c tabletype columnspec \  
... tablename
```

引数の意味はそれぞれ以下のとおりです。

- *tabletype* は、単にテーブルをあるテーブルクラスに所属するものとして識別する文字列です。任意の名前を使用できます。
- *columnspec* 引数には、各列の名前と特性を指定します。1 つの *columnspec* には、新規テーブルに含める、それぞれの列を入力してください。複数の *columnspec* を指定する場合は、次の例に示すように空白で区切ってください。

```
nistbladm -c tabletype columnspec columnspec \  
columnspec tablename
```

columnspec の書式については、357 ページの「テーブルの列を指定する」を参照してください。

テーブルの列を指定する

各 *columnspec* エントリは、以下の形式のように、2 つから 4 つの要素から成っています。

name=type,rights:

表 19-3 テーブルの列の構成要素

構成要素	目的
<i>name</i>	列の名前
=	等号記号 (必須)
<i>type</i>	(オプション) S、I、または C で列の種類を指定する。表 19-4 参照。このコマンドは省略可能です。 <i>type</i> を指定しないと、その列はデフォルトとなる
<i>rights</i>	(オプション) アクセス権を指定する。ここに指定したアクセス権は、テーブル全体や特定エントリに付与したアクセス権より優先される。 <i>access</i> を指定しないと、その列のアクセス権は、テーブル全体や特定エントリに付与したアクセス権になる。アクセス権の構文については、277 ページの「コマンドによるアクセス権の指定」を参照

列には、次に挙げる種類のうち 1 つを指定できます。

表 19-4 列の種類

種類	目的
	等号 (=) のみ。列の種類は指定しない。その列は検索可能にもならなければ、暗号化されることもない

表 19-4 列の種類 (続き)

種類	目的
S	検索可能
I	大文字と小文字の区別をしない。NIS+ コマンドで列を検索する場合、大文字と小文字を区別しない
C	暗号化する

NIS+ の各コマンドは、列全体を調べ、検索可能列の値に基づいて個々の行を識別します。検索可能列は `s` または `i` オプションで指定します (データベース用語では、検索可能列のことをキー列といいます)。テーブルの最初の列は必ず検索可能にします。その他の列は検索可能にする必要はありません。NIS+ テーブルのキーは、検索可能列に対して作成されます。このため、検索可能列が複数存在する場合は、最初の列から連続した領域に割り当てなければなりません。検索可能列以外の列がその領域に存在してはなりません。たとえば、テーブルの検索可能列が 1 列だけの場合は、最初の列に配置しなければなりません。検索可能列が 2 列の場合は、最初の 2 列に配置しなければなりません。検索可能列の詳細については、[354 ページの「nistbladm、検索可能列、キー、列の値」](#)を参照してください。

アクセス権だけを指定する場合、コンマを使用する必要はありません。-S、-I、-C フラグの 1 つか複数を指定する場合は、アクセス権の前にコンマを追加します。

以下の例では、上と同じテーブルが作成されますが、最初の 2 列にはその列に固有なアクセス権が付与されます。

```
master% nistbladm -c depts Name=I,w+m Site=w+m Name=C \
divs.mydir.doc.com.
```

テーブル作成時の列のアクセス権の指定については、[285 ページの「テーブル作成時の列権利設定」](#)を参照してください。

注 - NIS+ では、すべての列エントリが NULL で終了するものと仮定しています。NIS+ テーブルに情報を書き込むアプリケーションやルーチンは、列エントリをすべて NULL で終了するように構成しなければなりません。

自動マウントテーブルを追加作成する

自動マウントテーブルには 2 つの列しか作ることができません。1 番目の列には `key` という名前を付け、2 番目の列には `value` という名前を付けます。たとえば、`auto1` という名前の自動マウントテーブルを作成するには、次のように入力します。

```
master% nistbladm -c key-value key=S value= auto1.org_dir.doc.com.
```

テーブルを削除する

テーブルを削除するには、`-d` オプションを使ってテーブル名を入力します。

```
nistbladm -d tablename
```

テーブルを削除するには、その前にテーブルが空になっていなければなりません (364 ページの「[テーブルからエントリを削除する](#)」を参照)。次の例では、`doc.com.` ディレクトリから `divs` テーブルを削除します。

```
rootmaster% nistbladm -d divs.doc.com.
```

エントリをテーブルに追加する

エントリ (行) をテーブルに追加するには、`nistbladm` コマンドに `-a` オプションまたは `-A` オプションを指定し、続けて `column=value` のペア (1 つまたは複数)、テーブル名の順に指定します。あるいは、`nistbladm` コマンドに `-a` オプションまたは `-A` オプションを指定し、続けてインデックス名を指定します (355 ページの「[nistbladm とインデックス名](#)」を参照)。

```
nistbladm [-a | -A] indexedname
nistbladm [-a | -A] column="value" \
column="value" \
... tablename
```

`-a` オプションまたは `-A` オプションを指定して既存のテーブルに新しいエントリ (行) を追加する場合は、次の点に注意してください。

- `value` は必ず引用符で囲む。たとえば、新たに追加するエントリの `cname` 列の値を `deneb` にする場合は、`column=value` ペアを `cname="deneb"` と指定する
- テーブル内のすべての列の値を指定する
- 追加するエントリ (行) の列を空白にする場合は、`column=" "` と指定する。つまり、`value` には、引用符で囲んだ半角スペースを指定する

注 - NIS+ はネームサービスであり、設計上、そのテーブルにはオブジェクト本体ではなく、オブジェクトのリファレンスが格納されます。NIS+ は、最適化された状態では 10,000 におよぶオブジェクトをサポートし、すべてのテーブルのサイズを合計しても 10M バイトを超えることはありません。NIS+ では、1 つの列のフィールドサイズの合計が 7k を超えるようなテーブルはサポートされません。テーブルが大きすぎると、`rpc.nisd` は失敗します。

-a オプションを指定してエントリを追加する

nistbladm コマンドに -a オプションを指定しておく、追加するエントリと同じエントリがすでに存在する場合は、エントリの上書きは行われずにエラーが返されます。あるエントリの検索可能列の値が、追加するエントリの検索可能列の値とまったく同じである場合、そのエントリは「すでに存在しているもの」と判断されます。このとき、検索不可列の値は考慮されません。

-a オプションを指定する場合、次のように、テーブル内のすべての列の値を指定する必要があります。

```
nistbladm -a column="value" \  
column="value" \  
... tablename  
nistbladm -a indexedname
```

テーブルのすべての列名とその特性を表示する場合は、niscat -o tablename コマンドを実行します。

たとえば、depts テーブルに新しい行を追加する場合は、次に示すように、列の数だけ column=value ペアを指定します。

```
rootmaster% nistbladm -a Name='R&D' Site='SanFran' \  
Name='vattel' depts.doc.com.
```

なお、これと同じエントリをインデックス名を指定して追加する場合は、次のように入力します。

```
rootmaster% nistbladm -a [Name='R&D',Site='SanFran',\  
Name='vattel'],depts.doc.com.
```

いずれの場合も、次のようなテーブルとなります。

Dept	Site	Name
R&D	SanFran	vattel

C シェルを使っている場合は、角カッコを含む数式を引用符でオフセットすることもできます。

nistbladm コマンドでは 1 回につき 1 つのエントリしか追加できません。つまり、追加する行の数だけ nistbladm コマンドを実行する必要があります。

追加するエントリ (行) の各列と同じ値を持つ行がすでに存在する場合、nistbladm -a はエラーを返します。1 つのテーブルの中に同じ行が存在することはできません。検索可能列の値が同一である場合、それらの行は同一であるとみなされます。このとき、検索不可列の値は考慮されません。

たとえば、Dept 列と Site 列が検索可能であって、Name 列は検索可能ではないテーブルに対して nistbladm を実行すると、次の 2 つの行は同一であるとみなされます。

Dept (検索可能)	Site (検索可能)	Name (検索不可)
Sales	Vancouver	Hosteen
Sales	Vancouver	Lincoln

この場合、nistbladm -a を実行して Sales Vancouver Lincoln という行を作成できません。

しかし、複数の検索可能列のどちらか一方の値が重複するだけであれば、nistbladm -a を実行して新しい行を作成できます。たとえば、以下の2つのコマンドを実行すると、一部の値が異なるだけの2つの列を作成できます。

```
rootmaster% nistbladm -a Dept='Sales' \
    Site='Vancouver' Name='hosteen' staff.doc.com.
rootmaster% nistbladm -a Dept='Sales' \
    Site='SanFran' Name='lincoln' staff.doc.com.
```

これらのコマンドを実行すると、検索可能列の一部が同じで、すべてが同じではない2つの行が作成されます。

Dept	Site	Name
Sales	Vancouver	hosteen
Sales	SanFran	lincoln

-A オプションを指定してエントリを追加する

-A オプションは、nistbladm コマンドで既存のエントリを上書きするアプリケーションに使用します。-a と同様に、-A もテーブルにエントリを追加するためのオプションです。しかし、追加するエントリがすでに存在する場合、エラーを返して終了するのではなく、既存のエントリを上書きします。

-A オプションを指定する場合、エントリ内のすべての列の値を指定する必要があります。

ここで、Dept 列と Site 列が検索可能である次のテーブルを想定します。

Dept (検索可能)	Site (検索可能)	Name
Sales	SanFran	Lincoln

このテーブルに対して次のコマンドを実行します。

```
rootmaster% nistbladm -A Name=Sales Site=SanFran \
    Name=Tsosulu depts.doc.com.
```

Name=Sales Site=SanFran はすでに存在するので、オプションが -a であれば、上記のコマンドはエラーを返します。しかし、上記のコマンドでは -A が指定されているので、既存の行が上書きされます。

Dept	Site	Name
Sales	SanFran	Tsosulu

エントリを修正する

既存のエントリを編集 (修正) する場合は、-e オプションまたは -E オプションを指定します。Solaris リリースでは、旧リリースとの互換性を維持するため、-m オプションもサポートされています。新しいアプリケーションまたはコマンド行での操作には、-e オプションまたは -E オプションを使うことをお勧めします。

テーブル内の既存のエントリ (行) を編集するには、nistbladm コマンドに -e オプションまたは -E オプションを指定し、それに続けて、値を変更した *column=value* のペア (1 つまたは複数) とテーブル内の特定の行を示すインデックス名を指定します (355 ページの「nistbladm とインデックス名」を参照)。

```
nistbladm [-e | -E] column="value" \  
column="value" \  
... indexedname
```

-e オプションまたは -E オプションを指定して既存エントリ (行) を修正する場合は、次の点に注意してください。

- *value* は必ず引用符で囲む。たとえば、cname 列の値を deneb に変更する場合は、*column=value* ペアを
- 検索可能列の値は、1 行 (エントリ) 単位でしか変更できない
- エントリ (行) の列を空白にする場合は、*column=" "* と指定する。つまり、*value* には、引用符で囲んだ半角スペースを指定する

-e オプションを指定してエントリを編集する

-e オプションを指定した場合、1 つのエントリを編集します。複数エントリの検索可能列値を変更することになる場合は、編集は実行されず、エラーが返されます。このとき、検索不可列の値は考慮されません。

```
nistbladm column="value" \  
column="value" \  
... indexedname
```

-e オプションを指定した場合、変更する列の値を指定するだけです。

ここで次のテーブルについて考えます。

Dept	Site	Name
Sales	SanFran	Tsosulu

Name 列の値を Chandar に変更するには、次のように入力します。

```
master% nistbladm -e Name="Chandar" [Dept='Sales',Site='SanFran'],\
depts.doc.com.
```

修正後のテーブルは次のようになります。

Dept	Site	Name
Sales	SanFran	Chandar

(上記の例では、インデックス名に Name 列が含まれていません。これは、Name 列が検索可能ではないためです)。

C シェルを使っている場合は、角カッコを含む数式を引用符でオフセットすることもできます。

-e オプションを指定して検索可能列の値を編集するには、新たに指定した値が (インデックス名によって識別される) 1 行にだけ適用されることが条件になります。したがって、Dept 列の値を Manf に変更するには、次のように入力します。

```
master% nistbladm -e Dept="Manf" [Dept='Sales',Site='SanFran'],\
depts.doc.com.
```

Dept (検索可能)	Site (検索可能)	Name
Manf	SanFran	Chandar

しかし、検索可能列に Manf と SanFran という値を持つエントリがすでに存在する場合は、nistbladm -e はエラーを返します。

同一のエントリが対象である場合は、複数の列の変更を指定できます。たとえば、Dept 列と Name 列の値を変更するには、次のように入力します。

```
master% nistbladm -e Dept="Manf" Name="Thi" \
[Dept='Sales',Site='SanFran'],depts.doc.com.
```

Dept (検索可能)	Site (検索可能)	Name
Manf	SanFran	Thi

-E オプションを指定してエントリを編集する

-E オプションは、複数のエントリに影響する場合でも、nistbladm コマンドで既存のエントリを上書きする必要があるアプリケーションに使用します。

ここで次のテーブルを想定します。

Dept (検索可能)	Site (検索可能)	Name
Sales	SanFran	Chandar
Sales	Alameda	Achmed

このテーブルに対して次のコマンドを実行します。

```
master% nistbladm -E Site="Alameda" Mgr="Chu" \  
[Div='Sales',Site='SanFran'],depts.doc.com.
```

すると、Sales SanFran Chandar 行が Sales Alameda Chu に変わります。しかも、キーの値 Sales Alameda によって識別される Sales Alameda Achmed 行までもが変更の対象になります。その結果、コマンド実行前には2つあったはずの行が1つになります。

Dept (検索可能)	Site (検索可能)	Name
Sales	Alameda	Chu

これは複数の行を編集することになるので、オプションが -e であれば、上記のコマンドはエラーを返します。しかし、実際には -E が指定されているので、複数のエントリ (行) が編集されます。

テーブルからエントリを削除する

- テーブルから1つのエントリを削除する場合は、-r オプションを指定します (364 ページの「1つのエントリを削除する」を参照)。
- テーブルから複数のエントリを削除する場合は、-R オプションを指定します (365 ページの「複数のエントリを削除する」を参照)。

1つのエントリを削除する

テーブルから1つのエントリを削除するには、次に示すように -r オプションを指定します。

```
nistbladm -r indexed-name
```

次に示すコマンドでは、depts テーブルから Manf-1 エントリを削除します。

```
rootmaster% nistbladm -r [Dept=Manf-1,Site=Emeryville,Name=hosteen],\
depts.doc.com.
```

指定する列の値の数は最小限に減らすことができます。NIS+ では、列の値の重複が見つかり、行は 1 つも削除されることなく、エラーが返されます。次のテーブルから Manf-1 エントリを削除するには、次に示すように Site 列の値を指定するだけで済みます。

```
rootmaster% nistbladm -r [Site=Emeryville],depts.doc.com.
```

しかし、R&D エントリも Sales エントリと同じ値を持っているため、Site 列の値 (SanFran) を指定するだけでは Sales エントリを削除できません。

Dept	Site	Name
R&D	SanFran	kuznetsov
Sales	SanFran	jhill
Manf-1	Emeryville	hosteen
Manf-2	Sausalito	lincoln

複数のエントリを削除する

テーブルから複数のエントリを削除するには、次に示すように -R オプションを指定します。

```
nistbladm -R indexedname
```

オプションが -r であれば、最小限必要な列値を指定するだけで済みます。しかし、上記のコマンドでは -R が指定されているので、NIS+ が重複を見つけると、それに該当するすべてのエントリが削除されます。テーブル内の列の名前を確認したい場合は、niscat -o コマンドを実行します。次のコマンドを実行すると、Site 列の値が SanFran であるすべてのエントリが削除されます。

```
rootmaster% nistbladm -R [Site=SanFran],depts.doc.com.
```

Dept	Site	Name
Manf-1	Emeryville	hosteen
Manf-2	Sausalito	lincoln

-R オプションを使用して、テーブルからすべてのエントリを削除することができます。次に示すように、列の値を指定しなければテーブル内のすべてのエントリを削除できます。

```
rootmaster% nistbladm -R [],depts.doc.com.
```

nistbladm -R コマンドとともに使用すると、一對の空の角カッコはすべてのテーブル行を指定するワイルドカードとして解釈されます。

niscat コマンド

niscat コマンドは NIS+ テーブルの内容を表示します。このコマンドを使って、テーブルのオブジェクト属性を表示できます。表示するテーブル、エントリ、または列に対する読み取り権が必要です。

niscat コマンドの構文

テーブルの内容を表示するには、以下のように入力します。

```
niscat [-hM] tablename
```

テーブルのオブジェクト属性を表示するには、以下のように入力します。

```
niscat -o tablename  
niscat -o entry
```

表 19-5 niscat 構文のオプション

オプション	目的
-h	ヘッダー。テーブルエントリの上にあるヘッダー行を表示し、各列の名前を表示する
-M	マスター。マスターサーバーに収められているテーブルのエントリだけを表示する。これによって、最新情報を取得できる。デバッグにだけ使うようにする
-o	オブジェクト。列名、属性、サーバーなどの、テーブルについてのオブジェクト情報を表示する

テーブルの内容を表示する

テーブルの内容を表示するには、niscat にテーブル名を付けて実行します。

```
niscat tablename
```

次の例では、depts というテーブルの内容を表示します。

```
rootmaster% niscat -h depts.doc.com.  
#Name:Site:Name  
R&D:SanFran:kuznetsov  
Sales:SanFran:jhill  
Manf-1:Emeryville:hosteen  
Manf-2:Sausalito:lincoln
```

注 - *NP* は、ユーザーにそのエントリを表示するためのアクセス権がないことを示します。アクセス権は、テーブル、列、エントリ (行) ごとに与えられます。アクセス権の詳細は、[第 15 章](#)を参照してください。

テーブルまたはエントリのオブジェクト属性を表示する方法

テーブルのオブジェクト属性を表示するには、niscat -o にテーブル名を付けて実行します。

```
niscat -o tablename.org_dir
```

テーブルエントリのオブジェクト属性を表示するには、niscat -o を使い、インデックス付きの名前でエントリを指定します。

```
entry ::=column=value \  
... tablename | \  
[column=value,...],\  
tablename
```

次に、テーブルの例とテーブルエントリの例を示します。

「テーブル」

```
rootmaster# niscat -o hosts.org_dir.doc.com.  
Object Name : hosts  
Owner : rootmaster.doc.com.  
Group : admin.doc.com.
```

```

Domain : org_dir.doc.com.
Access Rights : ----rmcdr---r---
Time to Live : 12:0:0
Object Type : TABLE
Table Type : hosts_tbl
Number of Columns : 4
Character Separator :
Search Path :
Columns :
  [0] Name : cname
  Attributes : (SEARCHABLE, TEXTUAL DATA, CASE INS
  Access Rights: -----
  [1] Name : name
  Attributes : (SEARCHABLE, TEXTUAL DATA, CASE INS
  Access Rights: -----
  [2] Name : addr
  Attributes : (SEARCHABLE, TEXTUAL DATA, CASE INS
  Access Rights: -----
  [3] Name : comment
  Attributes : (TEXTUAL DATA)
  Access Rights: -----

```

「テーブルエントリ」

```

rootmaster# niscat -o [name=rootmaster],hosts.org_dir.doc.com.
Object Name : hosts
Owner : rootmaster.doc.com.
Group : admin.doc.com.
Domain : org_dir.doc.com.
Access Rights : ----rmcdr---r---
Time to Live : 12:0:0
Object Type : ENTRY
  Entry data of type hosts_tbl
  Entry has 4 columns.
.
#

```

nismatch と nisgrep コマンド

nismatch コマンドと nisgrep コマンドは、NIS+ テーブルを検索して、それぞれ特定の文字列または正規表現に一致するエントリを探します。これらのコマンドは、エントリ自体、またはエントリの検索できた回数のどちらかを表示します。nismatch コマンドと nisgrep コマンドの相違を表 19-6 に示します。

表 19-6 nismatch と nisgrep の 比較

機能	nismatch	nisgrep
検索指定	テキストのみ指定可能	正規表現が指定可能
速度	高速	低速
検索対象	検索可能列のみ	すべての列 (検索可能かどうかとは無関係)
検索構文	<i>column=string ... tablename [column= string,...] , tablename</i>	<i>column=exp ... tablename</i>

この節の例では、両方のコマンドの構文を説明します。

どちらのコマンドを使用する場合にも、検索するテーブルに対する読み取りアクセス権が必要です。

この節の例は、次に示す `depts.doc.com` というテーブルの値をベースにしています。最初の 2 つの列だけが検索可能です。

Name (S)	Site (S)	Name
R&D	SanFran	kuznetsov
Sales	SanFran	jhill
Manf-1	Emeryville	hosteen
Manf-2	Sausalito	lincoln
Shipping-1	Emeryville	tsosulu
Shipping-2	Sausalito	katabami
Service	Sparks	franklin

正規表現について

正規表現により、テキストと記号を組み合わせて使用し、特定の構成の列の値を検索できます。たとえば、正規表現 `'Hello'` は、`Hello` で始まる値を検索します。正規表現をコマンド行で使用するときは、必ず引用符で囲んでください。その理由は、正規表現記号の多くが **Bourne** シェルと **C** シェルでは特殊な意味をもつからです。たとえば、以下ようになります。

```
rootmaster% nisgrep -h greeting='Hello' phrases.doc.com.
```

正規表現の記号を [表 19-7](#) にまとめます。

表 19-7 正規表現記号

記号	説明
<code>^string</code>	<code>string</code> で始まる値を見つける
<code>string \$</code>	<code>string</code> で終わる値を見つける
<code>.</code>	ピリオドの数と等しい数の文字をもつ値を見つける
<code>[chars]</code>	角カッコ内の文字のどれかを含む値を見つける
<code>*expr</code>	<code>expr</code> についてゼロ回以上一致する値を見つける
<code>+</code>	1 回以上現われるものを見つける
<code>?</code>	任意の値を見つける
<code>\ 's-char'</code>	? や \$ などの特殊文字を見つける
<code>x y</code>	x または y の値を見つける

nismatch と nisgrep コマンドの構文

最初の列を検索するには、以下のように入力します。

```
nismatch string tablename
nisgrep reg-exp tablename
```

特定の列を検索するには、以下のように入力します。

```
nismatch column=string tablename
nisgrep column=reg-exp tablename
```

複数の列を検索するには、以下のように入力します。

```
nismatch column=string tablename ... \
nismatch [column=string,...],tablename
nisgrep column=reg-exp ... \
tablename
```

表 19-8 nismatch 構文と nisgrep 構文のオプション

オプション	目的
-c	カウント。エントリ自体ではなく、検索指定に一致したエントリのカウントを表示する
-h	ヘッダー。エントリの上のヘッダー行を表示し、各列名を表示する
-M	マスター。マスターサーバーに収められているテーブルのエントリだけを表示する。これによって、最新情報を取得できる。デバッグにだけ使うようにする

最初の列を検索する

テーブルの最初の列で特定の値を検索するには、最初の列の値と「*tablename*」を入力します。nismatch では、この値は文字列でなければなりません。nisgrep では、この値は正規表現でなければなりません。

```
nismatch [-h] string tablename
nisgrep [-h] reg-expression tablename
```

次の例では、depts テーブルを検索して、最初の列の値が R&D となっているすべてのエントリを探します。

```
rootmaster% nismatch -h 'R&D' depts.doc.com.
rootmaster% nisgrep -h 'R&D' depts.doc.com.
```

注 - & がシェルによってメタキャラクタとして解釈されないように、R&D が引用符で囲まれています。

特定の列を検索する

最初の列以外の特定の列を検索するには、次の構文を使います。

```
nismatch column=string tablename
nisgrep column=reg-expression tablename
```

次の例では、depts テーブルを検索して、第 2 列の値が SanFran となっているすべてのエントリを探します。

```
rootmaster% nismatch -h Site=SanFran depts.doc.com.
rootmaster% nisgrep -h Site=SanFran depts.doc.com.
```

複数の列を検索する

2 つ以上の列で一致するエントリを検索するには、次の構文を使います。

```
nismatch [-h] [column=string, ... \
  column=string, ...],tablename
nisgrep [-h] column=reg-exp ... \
  tablename
```

以下の例では、第 2 列の値が SanFran で、第 3 列の値が jhill のエントリを検索します。

```
rootmaster% nismatch -h [Site=SanFran,Name=jhill], depts.doc.com.
rootmaster% nisgrep -h Site=SanFran Name=jhill depts.doc.com.
```

nisln コマンド

nisln コマンドは、NIS+ オブジェクトとテーブルエントリの間でシンボリックリンクを作成します。すべての NIS+ 管理コマンドで、NIS+ オブジェクト間のリンクをたどるように指示する `-L` フラグを使用できます。

注 – テーブルエントリはリンクしないでください。あるテーブルからほかのテーブルへのリンクはできますが、あるテーブルのエントリからほかのテーブルのエントリへのリンクはできません。

ほかのオブジェクトまたはエントリへのリンクを作成するには、ソースオブジェクトまたはエントリ、つまりほかのオブジェクトまたはエントリを指すものに対する変更権が必要です。



注意 – cred テーブルをリンクしないでください。org_dir ディレクトリには、それぞれ専用の cred テーブルが必要です。org_dir cred テーブルもリンクしないでください。

nisln コマンドの構文

リンクを作成するには次のように入力します。

`nisln source target`

表 19-9 nisln 構文のオプション

オプション	目的
<code>-L</code>	リンクをたどる。ソース (source) 自体がリンクである場合、新しいリンクはこれにはリンクされず、そのリンク元のソースにリンクされる
<code>-D</code>	デフォルト。リンクされたオブジェクトに対して別のデフォルトセットを指定する。デフォルトについては、283 ページの「デフォルトを無効にする」を参照

リンクを作成する

オブジェクト間のリンク (テーブルとディレクトリ間のリンクなど) を作成するには、最初に「ソース (source)」、次に「リンク先 (target)」の順で、両方のオブジェクト名を指定します。テーブルエントリはリンクしないでください。

`nisln source-object target-object`

nissetup コマンド

`nissetup` コマンドは、`org_dir` ディレクトリと `groups_dir` ディレクトリ、それにフルセットの NIS+ テーブルを作成することによって、既存の NIS+ ディレクトリオブジェクトをドメインに展開します。しかし、データでテーブルを生成することはありません。これを行うには、[374 ページの「`nisaddent` コマンド](#)」で説明する `nisaddent` コマンドが必要です。ドメインにディレクトリを展開することは、ドメインの設定作業の一部です。

注 - 新しい NIS+ ドメインを設定するのであれば、`nissetup` コマンドを使うより、`nisserver` スクリプトを使った方が簡単です。`nisserver` の使用法の詳細については、[95 ページの「NIS+ ルートサーバーの設定」](#)を参照してください。

`nissetup` コマンドは、NIS クライアントをサポートするドメインにもディレクトリを展開できます。

`nissetup` を使用するには、テーブルを格納するディレクトリに対する変更権が必要です。

ディレクトリを NIS+ ドメインに展開する

`nissetup` コマンドは、ディレクトリ名を付けても付けなくても使えます。ディレクトリ名を付けない場合、このコマンドはデフォルトのディレクトリを使います。追加される各オブジェクトは、出力に表示されます。

```
rootmaster# /usr/lib/nis/nissetup doc.com.
org_dir.doc.com. created
groups_dir.doc.com. created
auto_master.org_dir.doc.com. created
auto_home.org_dir.doc.com. created
bootparams.org_dir.doc.com. created
cred.org_dir.doc.com. created
ethers.org_dir.doc.com. created
group.org_dir.doc.com. created
hosts.org_dir.doc.com. created
mail_aliases.org_dir.doc.com. created
sendmailvars.org_dir.doc.com. created
netmasks.org_dir.doc.com. created
netgroup.org_dir.doc.com. created
networks.org_dir.doc.com. created
```

```
passwd.org_dir.doc.com. created
protocols.org_dir.doc.com. created
rpc.org_dir.doc.com. created
services.org_dir.doc.com. created
timezone.org_dir.doc.com. created
```

ディレクトリを NIS 互換ドメインに展開する方法

NIS+ と NIS のクライアント要求をサポートするドメインにディレクトリを展開するには、`-Y` フラグを使います。作成するテーブルは、NIS のクライアント要求がアクセスできるように、未認証カテゴリに対する読み取り権が与えられます。

```
rootmaster# /usr/lib/nis/nissetup -Y Test.doc.com.
```

nisaddent コマンド

`nisaddent` コマンドは、テキストファイルまたは NIS マップからの情報を NIS+ テーブルにロードします。また、NIS+ テーブルの内容をテキストファイルに逆にダンプできます。NIS+ テーブルをはじめて生成する場合は、[100 ページの「NIS+ テーブルの生成 \(populate\)」](#)を参照してください。すべての前提条件と関連作業が説明してあります。

`nisaddent` を使用して、情報のある NIS+ テーブルから別のテーブルに (たとえば、別のドメインの同じ種類のテーブルに) 転送できますが、直接には転送できません。まず、テーブルの内容をファイルにダンプし、次にそのファイルをほかのテーブルにロードする必要があります。ただし、ファイル内の情報は正しくフォーマットされていなければなりません。各テーブルに必要なフォーマットを [第 10 章](#)で説明します。

情報をテーブルにロードするとき、置換 (replace)、追加 (append)、またはマージ (merge) の 3 つのオプションを自由に使用できます。追加オプションは、ソースエントリを NIS+ テーブルに単純に追加します。置換オプションの場合、NIS+ は、初めにテーブル内の既存のエントリをすべて削除してから、情報源からのエントリを追加します。大規模なテーブルでは、これによって大きなエントリセット (1 セットは既存エントリの削除用、ほかのセットは新エントリの追加用) がテーブルの `.log` ファイルに追加され、`/var/nis` 内の領域を占有し、複製への伝達に長時間を要することになります。

マージオプションでは、置換オプションと同じ結果が得られますが、異なるプロセスを使っており、複製に送信する動作数が大幅に減少します。このオプションでは、NIS+ は次の 3 種類のエントリを異なる方法で処理します。

- ソース内にだけ存在するエントリは、テーブルに「追加」される
- ソースとテーブルの両方に存在するエントリは、テーブル内で「更新」される
- NIS+ テーブルにだけ存在するエントリは、テーブルから「削除」される

大きなテーブルを更新する場合で、内容の変更があまりないときは、マージオプションを使用するとサーバーは多くの動作を節約できます。ソース内で重複していないエントリだけを削除する (置換オプションは全エントリを無差別に削除する) ため、重複エントリごとに 1 回の削除と 1 回の追加動作を省略できます。

はじめてテーブルに情報をロードする場合、テーブルオブジェクトに対する作成権が必要です。テーブル内の既存情報を上書きする場合、テーブルに対する変更権が必要です。

nisaddent コマンドの構文

テキストファイルから情報をロードするには、以下のように入力します。

```
/usr/lib/nis/nisaddent -f filename table-type\ [domain]
/usr/lib/nis/nisaddent -f filename \
-t tablename table-type [domain]
```

NIS マップから情報をロードするには、以下のように入力します。

```
/usr/lib/nis/nisaddent -y NISdomain table-type\
[domain]
/usr/lib/nis/nisaddent -y NISdomain -t tablename table-type [domain]
/usr/lib/nis/nisaddent -Y map table-type [domain]
/usr/lib/nis/nisaddent -Y map -t tablename table-type [domain]
```

NIS+ テーブルから情報をファイルにダンプするには、以下のように入力します。

```
/usr/lib/nis/nisaddent -d [-t tablename tabletype] \
> filename
```

ファイルから情報をロードする

ファイルの内容を NIS+ テーブルに転送するには、いくつかの方法があります。

- `-f` を単独で指定すると、`table-type` の内容が `filename` の内容に置き換えられます。

```
nisaddent -f filename table-type
```

- `-f` と `-a` を同時に指定すると、`filename` の内容が `table-type` に「追加」されます。

```
nisaddent -a -f filename table-type
```

- `-f` と `-m` を同時に指定すると、`filename` の内容が `table-type` の内容に「マージ」されます。

```
nisaddent -m -f filename table-type
```

次に示す 2 つの例では、テキストファイル `/etc/passwd.xfr` の内容が NIS+ `passwd` テーブルにロードされます。最初の例ではローカルドメインに、2 番目の例では別のドメインのテーブルにロードされます。

```
rootmaster# /usr/lib/nis/nisaddent -f /etc/passwd.xfr passwd
rootmaster# /usr/lib/nis/nisaddent -f /etc/shadow.xfr shadow
```

```
rootmaster# /usr/lib/nis/nisaddent -f /etc/passwd.xfr passwd sales.doc.com.
rootmaster# /usr/lib/nis/nisaddent -f /etc/shadow.xfr shadow sales.doc.com.
```

注 - /etc ディレクトリのファイルから NIS+ passwd テーブルを作成する場合は、nisaddent を 2 回 (/etc/passwd ファイルと /etc/shadow ファイルに対して 1 回ずつ) 実行する必要があります。

/etc/inet/ipnodes ファイル (IPv6 アドレス) のエントリを ipnodes.org_dir テーブルにマージするには、-v オプションと -f オプションを使用します。

```
rootmaster# /usr/lib/nis/nisaddent -mv -f /etc/inet/ipnodes ipnodes
```

もう 1 つの方法では、stdin をソースとして使います。しかし、stdin では -m オプションを使えません。次に例を示します。リダイレクト (->) やパイプ (-|) を使用することは可能です。ただし、別のドメインに出力が行われるような形でパイプを使用することはできません。

作業	コマンド
リダイレクト	<code>cat filename > nisaddent table-type</code>
リダイレクト処理後、追加する	<code>cat filename > nisaddent -a table-type</code>
リダイレクト処理後、別のドメインに追加する	<code>cat filename > nisaddent -a table-type NIS+ domain</code>
パイプ	<code>cat filename nisaddent table-type</code>
パイプ処理後、追加する	<code>cat filename nisaddent -a table-type</code>

NIS+ テーブルが、オートマウントテーブルの 1 つであるか標準以外のテーブルである場合、-t オプションと NIS+ テーブルの完全な名前を追加します。

```
master# nisaddent -f /etc/auto_home.xfr \
-t auto_home.org_dir.doc.com.key-value
master# nisaddent -f /etc/auto_home.xfr \
-t auto_home.org_dir.doc.com. key-value sales.doc.com.
```

NIS マップからデータをロードする

NIS マップから情報を転送する方法は 2 つあり、NIS ドメインを指定するか、または実際の NIS マップを指定します。ドメインを指定した場合、NIS+ は、table-type に基づいて、/var/yp/nisdomain 内のどのマップファイルをソースとして使うかを判断します。/var/yp/nisdomain は「ローカル」ファイルでなければなりません。

NIS+ テーブル形式	NIS マップ名
Hosts	hosts.byaddr
Nodes	ipnodes.byaddr
Passwd	passwd.byname
Group	group.byaddr
Ethers	ethers.byname
Netmasks	netmasks.byaddr
Networks	networks.byname
Protocols	protocols.byname
RPC	rpc.bynumber
Services	services.byname

NIS ドメインの指定によって転送するには、`-y` (小文字) オプションを使い、NIS+ テーブル形式に加えて NIS ドメインを指定します。

「テーブルの置換」

```
nisaddent -y nisdomain table-type
```

「テーブルの追加」

```
nisaddent -a -y nisdomain table-type
```

「テーブルのマージ」

```
nisaddent -m -y nisdomain table-type
```

デフォルトでは、`nisaddent` は NIS+ テーブルの内容を NIS マップの内容に置き換えます。追加またはマージを行うには、`-a` または `-m` のオプションを使います。次の例では、対応する NIS マップ (`passwd.byname`) からの NIS+ `passwd` テーブルを `old-doc` ドメインにロードします。

```
rootmaster# /usr/lib/nis/nisaddent -y old-doc passwd
```

2 番目の例でも同じことを行いますが、ローカルドメイン `doc.com.` の代わりに、`sales.doc.com.` ドメインに対して行います。

```
rootmaster# /usr/lib/nis/nisaddent -y old-doc passwd sales.doc.com.
```

NIS+ テーブルが、オートマウントテーブルの 1 つであるか非標準テーブルである場合、そのソースがファイルであるかのように、`-t` オプションと NIS テーブルの完全な名前を追加します。

```
rootmaster# nisaddent -y old-doc \
-t auto_home.org_dir.doc.com. key-value
rootmaster# nisaddent -y old-doc \
```

```
-t auto_home.org_dir.doc.com. key-value sales.doc.com.
```

マップファイルをドメイン用に使うのではなく、特定の NIS マップを指定したい場合、**-Y** (大文字) オプションを使い、マップ名を指定します。オプションを見つけやすいように、次の例では太字にしています。

```
rootmaster# nisaddent -Y hosts.byname hosts
rootmaster# nisaddent -Y hosts.byname hosts sales.doc.com.
```

NIS マップがオートマウントマップの 1 つであるか非標準マップである場合、**-Y** オプションと **-t** オプションを組み合わせます。

```
rootmaster# nisaddent -Y auto_home
-t auto_home.org_dir.doc.com. key-value
rootmaster# nisaddent -Y auto_home
-t auto_home.org_dir.doc.com. key-value sales.doc.com.
```

NIS+ テーブルの内容をファイルにダンプする

NIS+ テーブルの内容をファイルにダンプするには、**-d** オプションと **-t** オプションを使います。**-d** オプションは、ダンプするようコマンドに指示します。**-t** オプションは、NIS+ テーブルを指定します。

```
rootmaster# nisaddent -d auto_home
-t auto_home.org_dir.doc.com. key-value
rootmaster# nisaddent -d auto_home
-t auto_home.org_dir.doc.com. key-value sales.doc.com.
```

第 20 章

サーバー使用のカスタマイズ

この章では、NIS+ クライアントが使用するサーバーのカスタマイズおよび制御方法について説明します。

注 - NIS+ は、将来のリリースではサポートされなくなる可能性があります。NIS+ から LDAP への移行支援ツールは、Solaris 9 リリース以降で使用できます (『Solaris のシステム管理 (ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』を参照)。詳細については、<http://www.sun.com/directory/nisplus/transition.html> を参照してください。

NIS+ サーバーと NIS+ クライアント

クライアントマシン、ユーザー、アプリケーション、あるいはプロセスは、アクティブな NIS+ サーバー (マスターまたは複製) を検索して、そこから必要な情報を取り込みます。巨大なネットワーク、サブネットを多く持ったネットワーク、あるいは広域リンクにまたがったネットワークでは、サーバーの使用法をカスタマイズすることにより NIS+ の性能を強化できます。

デフォルトでのクライアントの検索動作

カスタマイズ前の状態で、nisprefadm コマンドによるサーバーの優先順位設定がないにも設定されていなければ、クライアントは、まず自分自身のローカルサブネット上の NIS+ サーバーから情報を入手しようとします。そのサブネットにアクティブなサーバーが見つければ、応答のあった最初のローカルサーバーから必要な情報を入手します。ローカルサブネットにサーバーがない場合、次にクライアントは、ローカルサブネットの外部を検索し、応答のあった最初の遠隔サーバーから必要な NIS+ 情報を入手します。

ネットワークが大規模で、混雑している場合、上記のデフォルトの検索動作では NIS+ の性能を十分に発揮させることができないことがあります。それは次のような理由によります。

- サブネット上の複数のサーバーが多数のクライアントに情報の配布を行なっている場合、クライアントのデフォルト検索パターンがランダムなために、ほかにあまり使われていないサーバーがあるのに、いくつかのサーバーの負荷が高くなりすぎる可能性があります。
- ローカルサブネットの外で NIS+ サーバーを検索する際、クライアントは、オーバーワークしているサーバーであっても、低速な広域ネットワーク接続方式 (たとえば、モデム) や、すでにトラフィック多い専用回線によってリンクされているサーバーであっても、最初に応答したものから情報を入手します。

優先サーバーを指定する

Solaris リリースには、新規機能 (サーバー使用のカスタマイズ) が搭載されています。この機能を使用すると、クライアントが NIS+ サーバーを検索する順序をコントロールできます。この新規機能では、次のような方法でサーバーの使用の度合いをバランスよくカスタマイズできます。

- クライアントが特定のサーバーを優先的に選択する (検索する) ように指定します。
- 使用できるローカルサーバーがない場合、クライアントが遠隔サーバーを使用してもよいかどうかを指定します。

指定した検索基準は、ドメイン内のすべてのクライアント、サブネット上のすべてのクライアント、またはマシンごとに独立したクライアントに適用できます。

注 - サーバー使用の優先順位を特定のマシンに設定すると、この優先順位は、そのマシンで実行中のユーザー、アプリケーション、処理、またはほかのクライアントすべてに適用されます。同じマシン上の別のクライアントに、異なるサーバー使用のパターンを設定できません。

広域ネットワーク (WAN) 上での NIS+

使用サーバーのカスタマイズは、多数のサブネットを持つ大規模ネットワークや、モデムまたは専用回線で接続された複数の地理的サイトにまたがるネットワークで使用すると特に効果があります。ネットワークの性能を最大にするには、サブネット間やより低速な接続によってリンクされたサイト間のトラフィックを最小限にする必要があります。これは、クライアントが使用できる NIS+ サーバーとその優先順位を指定することによって実現できます。このようにして、可能な限り NIS+ ネットワークトラフィックがローカルサブネットから出ないようにします。

サーバー使用の最適化 - 概要

この節では、サーバー使用のカスタマイズの概要について説明します。

`nis_cachemgr` が必要

使用サーバーをカスタマイズするには、クライアントで `nis_cachemgr` を実行している必要があります。クライアントマシンが `nis_cachemgr` を実行していない場合は、使用サーバーのカスタマイズ機能は利用できません。クライアントマシン上で、`nis_cachemgr` を実行していない場合は、そのクライアントは、[379 ページの「デフォルトでのクライアントの検索動作」](#) で説明した方法で識別された最初のサーバーを使用します。

グローバルテーブルまたはローカルファイル

`nisprefadm` コマンドは、次のように、その使用方法により、ローカルの `client_info` ファイルまたはドメインの `client_info` テーブルのどちらかを作成します。

- 「ファイル」。 `nisprefadm` を使用して、マシンの `/var/nis` ディレクトリに格納される、ローカルなマシン固有の `client_info` を作成できます。ローカルファイルは、サーバーの優先順位をそのマシンだけに指定するためのものです。マシンに、ローカルの `/var/nis/client_info` ファイルがある場合、そのマシンは、ドメインの `client_info.org_dir` テーブルに入っているサーバーの優先順位は無視します。ローカル `client_info` ファイルを作成するには、`-L` オプションを指定した `nisprefadm` を実行してください。
- 「テーブル」。 `nisprefadm` を使用して、各ドメインの NIS+ ディレクトリオブジェクト `org_dir` 内に格納される NIS+ `client_info` テーブルを作成できます。このテーブルは、次のものに対してサーバーの優先順位を指定できます。
 - 個別のマシンマシンにローカルの `/var/nis/client_info` ファイルがある場合、ドメインの `client_info` テーブルに指定されたマシンの優先順位はすべて無視されます。
 - 1つの特定のサブネット上にあるすべてのマシンサブネット上のマシンにローカルの `/var/nis/client_info` ファイルがあるか、またはテーブル内にそのマシン固有の優先順位が設定されている場合、そのマシンは、サブネットに指定された優先順位は無視します。

1つのサブネット上のすべてのマシンに適用されるグローバル `client_info` テーブルを作成するには、`-G` および `-c` オプションを指定して `nisprefadm` を実行してください。これについては、[388 ページの「グローバルサーバー優先順位を指定する」](#) で説明しています。

マシンに、以下で説明するような、固有のローカル `client_info` ファイルがある場合、そのマシンに対してグローバル `client_info` テーブル内に設定されたサーバーの優先順位はすべて無視されるので注意してください。マシンに、ローカル `client_info` ファイルがあるか、またはグローバル `client_info` テーブルにそのマシン固有のエントリがある場合は、そのサブネットに設定された優先順位は無視されます。



注意 - `client_info` ファイルと `client_info` テーブルを変更する場合は、`nistbladm` コマンドだけを使用してください。`nistbladm` などの、ほかの NIS+ コマンドは絶対に使用しないでください。

`client_info` テーブルまたは `client_info` ファイルを処理する場合は、`-g` または `-L` オプションを使用して、グローバルテーブル (`-g`) またはコマンドを実行中のマシンのローカルファイル (`-L`) のどちらにそのコマンドを適用させるかを指定する必要があります。

優先順位の番号の指定

サーバーの優先順位は、各サーバーに「優先順位を表す番号」を指定することによって制御されます。クライアントは、数値で指定された優先順位にしたがって NIS+ サーバーを検索します。検索の順序は、まず、優先順位格付け番号が小さいサーバーを先に検索し、次に大きい数値の付いたサーバーを検索します。

つまり、クライアントは、まず初めにゼロの優先順位の付いた NIS+ サーバーから名前空間情報を入手しようとします。使用できる優先順位 = 0 のサーバーがない場合、クライアントは、優先順位 = 1 のサーバーに問い合わせを行います。1 のサーバーが使用できない場合は 2 のサーバー、次に 3 というように、この検索は必要な情報が入手できるか検索するサーバーがなくなるまで続きます。

優先順位格付け番号は、`nisprefadm` コマンドを使用してサーバーに割り当てます。これについては、[388 ページの「グローバルサーバー優先順位を指定する」](#)で説明しています。

サーバーの優先順位番号は、`client_info` テーブルと `client_info` ファイル内に格納されます。マシンに、固有の `/var/nis/client_info` ファイルがある場合は、このファイル内に格納された優先順位番号が使用されます。マシンに、固有の `client_info` ファイルがない場合は、ドメインの `client_info.org_dir` テーブル内に格納された優先順位番号が使用されます。このような `client_info` テーブルと `client_info` ファイルを、「優先サーバーのリスト」、または単に「サーバーリスト」と呼びます。

各クライアントのサーバーの優先順位を制御することで、サーバーの使用法をカスタマイズします。たとえば、ドメインに `mailer` という名前のクライアントマシンがあり、このマシンは名前空間情報を頻繁に利用していて、さらにこのドメインには、マスターサーバー (`nismaster`) と複製サーバー (`replica1`) の両方があるとします。

ここで、mailer マシン用に nismaster に優先順位番号 1 を割り当て、replica1 に優先順位番号 0 を割り当てると、mailer マシンは、名前空間情報を、必ず最初に replica1 から入手しようとし、その後で nismaster に移ります。次に、このサブネット上にあるほかのすべてのマシンに対して、優先順位番号を nismaster サーバーには 0 を、replica1 には 1 を割り当てます。この結果、ほかのマシンは、必ず最初に nismaster に問い合わせを行います。

同じ優先順位番号を、ドメイン内の複数のサーバーに割り当てることができます。たとえば、nismaster1 と replica2 の両方に優先順位番号 0 を割り当て、replica3、replica4、replica5 に優先順位番号 1 を割り当てることができます。

サーバーの優先順位のデフォルト

client_info ファイルまたはテーブルがない場合は、キャッシュ管理プログラムが自動的に、ローカルサブネット上のすべてのサーバーにデフォルトの優先順位番号ゼロ (0) を割り当て、ローカルサブネットの外部のすべてのサーバーに無限大の優先順位を割り当てます。nisprefadm は、このデフォルトの優先順位番号を自由に変更するためのものです。

効率とサーバーの優先順位番号

クライアントは、すべてのサーバーを指定された優先順位番号を使用して昇順に検索しなければなりません。クライアントが指定された優先順位番号をもつすべてのサーバーを検索するには、5 秒以上必要です。つまり、ドメイン内に 1 つのマスターサーバーと 4 つの複製サーバーがあり、それぞれのサーバーに 0 から 4 の異なる優先順位番号を指定した場合、クライアントがこれらの優先レベルをすべて実行するには、25 秒以上かかるということです。

性能を最大にするために、サーバーの優先レベルを 2 つまたは 3 つ以上使用しないでください。たとえば、上記のような場合は、5 つのサーバーのうちの 1 つに優先順位=0 を割り当て、ほかすべてのサーバーには優先順位 1 を割り当てるか、または 5 つのサーバーのうち 2 つに優先順位 1 を、残りの 3 つのサーバーに 2 を割り当てるといった方法をとってください。

優先サーバー限定と全サーバー

サーバーリストには、クライアントが優先サーバーを検出できなかった場合の処置も指定できます。「優先サーバー」とは、優先順位にゼロが指定されたサーバー、または nisprefadm を使用して優先順位番号を割り当てたサーバーです。

デフォルトでは、クライアントは優先サーバーに到達できないと、検索モードを使用して、ネットワーク上のあらゆる場所を検索し、検出できるあらゆるサーバーを検索します。これについては、[379 ページの「デフォルトでのクライアントの検索動作」](#)

で説明しています。このデフォルト機能を、`nisprefadm -o` オプションを使用して変更し、クライアントが優先サーバーだけを使用し、使用できるサーバーがない場合でも、優先サーバー以外のサーバーには問い合わせしないように指定できます。詳細については、[394 ページの「優先サーバー限定指定」](#)を参照してください。

注 - マシンのドメインに優先サーバーが全くない場合、このオプションは無視されます。

優先順位の表示

現在特定のクライアントマシンに対して有効なサーバーの優先順位を表示するには、`-l` オプションを指定した `nisprefadm` を実行してください。これについては、[387 ページの「現行のサーバー優先順位の表示」](#)で説明しています。

サーバー名とクライアント名

サーバーまたはクライアントマシンを指定する場合、次の点に注意してください。

- サーバー名とクライアント名は、同じ NIS+ ドメイン内にあり、オブジェクトを個別に特定できれば、完全指定名である必要はありません。マシン名を単独で使用できます。
- サーバーまたはサブネットが別の NIS+ ドメインにある場合は、そのマシンを個別に特定できるだけのドメイン名を含める必要があります。たとえば、`sales.doc.com` ドメイン内にいて、`manf.doc.com` ドメイン内の `nismaster2` マシンを指定する必要がある場合、`nismaster2.manf` とだけ入力します。

サーバーの優先順位

以下のマシンにサーバーの優先順位を指定する方法は、それぞれ次のとおりです。

- 「個別のクライアントマシン」にサーバーの優先順位を指定するには、`-l` オプションを使用すると、`nisprefadm` を実行中のマシンにローカル `client_info` ファイルを作成します。`-G -c` マシンオプションを使用すると、グローバル `client_info` テーブル内にマシン固有の優先順位を作成します。
- 「サブネット上のすべてのマシン」にサーバーの優先順位を指定するには、`-G -c` サブネット番号オプションを使用してください。
- マシン固有のまたはサブネット固有の優先順位を持たない、現在のドメイン内にあ
るすべてのマシンにサーバーの優先順位を指定するには、`-G` オプションを使用し
てください。

サーバーの優先順位が有効になるタイミング

マシンまたはサブネットのサーバー優先順位の変更内容は、通常、指定したマシンがその `nis_cachemgr` データを更新するまでは有効になりません。マシンの `nis_cachemgr` がそのサーバー使用情報を更新するタイミングは、マシンがサーバーの優先順位をグローバル `client_info` テーブルまたはローカル `/var/nis/client_info` ファイルのどちらから入手するかによって決まります (381 ページの「グローバルテーブルまたはローカルファイル」を参照してください)。

- 「グローバルテーブル」から入手する場合サーバーの優先順位をグローバルテーブルから入手するマシンのキャッシュ管理プログラムは、マシンがブートされた時、または `client_info` テーブルの存続時間 (TTL) の値が満了した時に、マシン用のサーバーの優先順位を更新します。デフォルトでは、この TTL 値は 12 時間ですが、変更可能です。348 ページの「オブジェクトの生存期間を変更する」を参照してください。
- 「ローカルファイル」ローカルファイルからサーバーの優先順位を入手するマシンのキャッシュ管理プログラムは、サーバーの優先順位を 12 時間ごとに更新するか、または `nisprefadm` を実行してサーバーの優先順位を変更した時に必ず更新します。マシンを再起動しても、キャッシュ管理プログラムのサーバーの優先順位情報は更新されません。

ただし、`nisprefadm` に `-F` オプションを指定して実行すると、サーバーの優先順位の変更内容を強制的にただちに有効にできます。`-F` オプションを使用すると、`nis_cachemgr` で、ただちにその情報が更新されます。詳細は、398 ページの「優先順位の変更内容をただちに実現する方法」を参照してください。

`nisprefadm` コマンドの使用方法

ここからの節では、`nisprefadm` コマンドを使用して、サーバーの優先順位を設定、変更、削除する方法について説明します。

`nisprefadm` コマンドは、クライアントが優先的に選択するサーバーを指定するために使用します。

`nisprefadm` コマンドの構文は次のとおりです。

```
nisprefadm -a|-m|-r|-u|-x|-l -L|-G [-o type] \  
[-d domain] \  
[-C machine] \  
servers  
nisprefadm -F
```

表 20-1 nisprefadm コマンドのオプション

オプション	目的
-G	ドメインの org_dir ディレクトリ内に格納されるグローバル client_info テーブルを作成する。つまり、グローバルな優先されるサーバーリストを作成する。このオプションは、指定したサブネット上のすべてのマシンに対して優先順位を指定する場合は -c subnet、個別のマシンに優先順位を指定する場合は -c machine のどちらかと同時に使用しなければならない
-L	ローカルマシンの /var/nis ディレクトリに格納されるローカル client_info ファイルを作成する。つまり、コマンドを実行中のマシンにだけ適用される優先サーバーのリストを作成する
-o type	オプションを指定する。有効なオプションには、クライアントが接続できる優先サーバーがない場合、優先サーバー以外のサーバーを使用できるように指定する pref_type=all と、指定した優先サーバーだけをクライアントが使用するよう指定する pref_type=pref_only がある
-d domain	指定したドメインまたはサブドメインに、グローバルな優先サーバーの client_info テーブルを作成する
-C subnet	優先順位を適用するサブネットの番号
-C machine	クライアントマシン名
servers	1 つまたは複数の NIS+ サーバー。ここで指定されたサーバーは優先的に選択される
-a	サーバーリストに指定サーバーを追加する
-m	サーバーリストを変更する。たとえば、-m オプションを使用すると、1 つまたは複数のサーバーに指定された優先順位番号を変更できる
-r	サーバーリストから指定したサーバーを削除する
-u	サーバーリストを消去してから、指定したサーバーを追加する (つまり、現行のサーバーリストを優先サーバーの新規リストに置換する)
-x	サーバーリストを完全に削除する
-l	現行の優先サーバー情報を一覧表示 (表示) する
-F	優先サーバーのリストを強制的にただちに変更する

注 - `-C machine` オプションは、`-L` (ローカル) フラグとともに使用しても無効となるので、使用しないでください。たとえば、`nisprefadm` を `altair` マシン上で実行しているとします。ここで、`-L` フラグを使用して、指定した優先順位が `altair` のローカル `client_info` ファイルに書き込まれるように指定し、さらに、`-C vega` オプションを使用して、作成した優先順位が `vega` マシンに適用されるように指定します。すると、`nisprefadm` コマンドは、`vega` 用の優先順位を `altair` のファイルに書き込みます。しかし、`vega` は、サーバーの優先順位を必ず自身のローカル `client_info` ファイルまたはドメインのグローバル `client_info` テーブルから入手するため、これらの情報を参照することはありません。そのため、`-C` オプションは、`nisprefadm` に `-G` (グローバル) フラグを指定して実行する場合にだけ意味をもちます。

現行のサーバー優先順位の表示

現行のサーバー優先順位を表示するには、`nisprefadm` に `-l` オプションを指定して実行してください。

マシンに指定された優先順位の表示方法

- そのマシン上で、**`nisprefadm`** に **`-L`** と **`-l`** を指定して実行します。

```
sirius# nisprefadm -L -l
```

このようにすると、マシンのローカル `/var/nis/client_info` ファイル内に定義されたサーバーの優先順位がすべて表示されます。ローカルファイルがない場合は、情報は表示されず、シェルプロンプトに戻ります。

1 台のマシンに設定されたグローバルな優先順位の表示方法

- **`nisprefadm`** に、**`-l`**、**`-G`**、**`-C machinename`** オプションを指定して実行します。

```
sirius# nisprefadm -G -l -C machinename
```

`machinename` には、マシンの IP アドレス (番号) が入ります。

このようにすると、このサブネット用にドメインのグローバル `client_info` テーブル内に設定された優先順位が表示されます。

サブネットに設定されたグローバル優先順位の表示方法

● **nisprefadm** に、**-l**、**-G**、**-C subnet** オプションを指定して実行します。

```
sirius# nisprefadm -G -l -C subnet
```

subnet には、サブネットの IP アドレス (番号) が入ります。

このようにすると、このサブネット用にドメインのグローバル `client_info` テーブル内に設定された優先順位が表示されます。

優先順位格付け番号の指定方法

デフォルトでは、**-a** オプションの後ろにリストしたサーバーには、すべて優先順位番号ゼロが指定されます。別の優先順位番号を指定する場合は、次のように、サーバー名の直後に指定する番号をカッコで囲んで入れてください (**-a name (n)**)。 **-a name (n)** にはサーバー名が、**n** には優先順位番号が入ります。

たとえば、`replica2` サーバーに優先順位番号 3 を割り当てる場合は、次のように入力します。

```
# nisprefadm -G -a replica2(3)
```

注 - シェルの中には、次のように要素を 2 重引用符で囲まなければならないものがあります。 "name (n) "

サーバーの優先順位格付け番号の基礎知識については、[382 ページの「優先順位の番号の指定」](#)を参照してください。

グローバルサーバー優先順位を指定する

ローカルまたは遠隔ドメインに、グローバルサーバー優先順位を設定できます。優先順位は、個別のマシンと 1 つのサブネット上のすべてのマシンに設定できます。

この節では、NIS+ ドメインのマスターサーバーに存在するグローバル `client_info` テーブル内にサーバーの優先順位を設定する方法を説明します。マスターサーバー上にテーブルがある場合、NIS+ は、そのテーブルをドメインの既存の複製サーバー上に複製します。

- 個別のマシンにローカル `client_info` ファイルを作成する方法については、390 ページの「ローカルサーバー優先順位を指定する」を参照してください。
- グローバル `client_info` テーブルとローカル `client_info` ファイルの相違点については、381 ページの「グローバルテーブルまたはローカルファイル」を参照してください。

サーバーの優先順位番号を割り当てるには、次のどちらかを指定した `nisprefadm` を実行してください。

- 新規または別の優先サーバーを追加する場合は、`-a` オプションを指定します。
- 既存のサーバー優先順位を削除して、新規の優先順位を作成する場合は、`-u` オプションを指定します。

サブネットにグローバル優先順位を設定する方法

1 つのサブネット上のすべてのマシンのグローバルテーブルにサーバーの優先順位を割り当てるには、次のようにします。

- **`nisprefadm`** に **`-G`** および **`-C subnet`** オプションを指定して実行します。

```
#nisprefadm -G -a -C subnet servers (preferences)
```

引数の意味はそれぞれ以下のとおりです。

- `-C subnet` には、優先順位を適用するサブネットの IP 番号を指定します。
- `servers (preferences)` には、1 つまたは複数のサーバーが入ります。それらには優先順位格付け番号を指定できます。

たとえば、サブネット 172.22.123.123 が `nismaster` および `replica3` サーバーにデフォルトの優先順位番号ゼロを付け、`manf.replica6` サーバーには優先順位番号1 を付けて使用するよう指定する場合は、次のように入力してください。

```
polaris# nisprefadm -a -G -C 172.22.123.123 nismaster1 \
replica3 "manf.replica6(1)"
```

個別のマシンにグローバル優先順位を設定する方法

- **`-G`** と **`-C machine`** オプションを指定した **`nisprefadm`** を実行します。

```
#nisprefadm -G -a -C machine servers (preferences)
```

引数の意味はそれぞれ以下のとおりです。

- `-C machine` には、優先順位を適用するマシンを指定します。使用するシェルによっては、`machine` を引用符で囲む必要があります。

- *servers (preferences)* には、1 つまたは複数のサーバーが入ります。それらには優先順位格付け番号を指定できます。

たとえば、マシン *cygnus* に指定された現行の優先順位を、*replica7* と *replica9* の両方にデフォルトの優先順位番号ゼロを指定したものに置き換える場合は、次のように入力します。

```
polaris# nisprefadm -u -G -C cygnus replica7 replica9
```

遠隔ドメインにグローバル優先順位を設定する方法

遠隔ドメイン内の個別のマシン、または遠隔ドメイン内の 1 つのサブネット上にあるすべてのマシンにサーバーの優先順位を割り当てるには、次のようにします。

- **nisprefadm** に **-C**、**-G**、**-d** オプションを指定して実行します。

```
#nisprefadm -a -G -C name \  
-d domain servers(preferences)
```

引数の意味はそれぞれ以下のとおりです。

- *name* には、サブネットの IP 番号またはマシン名が入ります。このコマンドを使用して行なった変更は、ここに指定したサブネットまたはマシンに適用されます。
- *domainname* には、遠隔ドメイン名が入ります。
- *servers (preferences)* には、1 つまたは複数のサーバーが入ります。それらには優先順位格付け番号を指定できます。

たとえば、デフォルトの優先順位番号ゼロを指定した *nismaster2* サーバーを、遠隔ドメイン *sales.doc.com* 内のサブネット *10.10.111.11* の優先サーバーのリストに追加する場合は、次のように入力します。

```
polaris# nisprefadm -a -G -C 10.10.111.11 -d sales.doc.com. nismaster2
```

ローカルサーバー優先順位を指定する

ここでは、ローカル *client_info* ファイルの作成および変更方法について説明します。ローカル *client_info* ファイルは、このファイルが存在するマシンにサーバーの優先順位を指定するためのものです。

マシンに */var/nis/client_info* ファイルがある場合、このマシンは、NIS+ サーバー上のグローバル *client_info* テーブルではなく、そのマシンのローカルファイルからサーバーの優先順位を入手します。つまり、ローカルファイルはグローバルテーブルよりも優先されます。

- NIS+ サーバーのグローバル `client_info` テーブルを作成する方法については、[388 ページの「グローバルサーバー優先順位を指定する」](#)を参照してください。
- グローバル `client_info` テーブルとローカル `client_info` ファイルの相違点については、[381 ページの「グローバルテーブルまたはローカルファイル」](#)を参照してください。

サーバーの優先順位を割り当てるには、次のどちらかを指定した `nisprefadm` を実行してください。

- 新規または別の優先サーバーを追加する場合は、`-a` オプションを指定します。
- 既存のサーバー優先順位を削除して、新規の優先順位を作成する場合は、`-u` オプションを指定します。

ローカルマシン上に優先順位を設定する方法

`nisprefadm` コマンドを実行中のローカルマシンにサーバーの優先順位を割り当てるには、次のようにします。

- **`nisprefadm`** に `-L` オプションと `-a` または `-u` オプションを指定して実行します。

```
#nisprefadm -a -L servers (preferences)
```

ここで、`servers (preferences)` には、1 つまたは複数のサーバーが入ります。それらには優先順位格付け番号を指定できます。

たとえば、`deneb` マシンが NIS+ 情報を、まず初めにデフォルトの優先順位番号ゼロが指定された `replica3` から検索し、次に `manf.doc.com` ドメイン内の (優先順位番号 1 が指定された) サーバー `replica6` を検索するように指定する場合は、次のように入力します。

```
deneb# nisprefadm -a -L replica3 replica6.manf(1)
```

サーバーの優先順位を変更する

サーバーの優先順位番号は、変更することができ、また別のサーバーに優先順位番号の割り当てを移すこともできます

優先サーバーまたはサーバーに割り当てた優先順位番号を変更するには、`nisprefadm` に `-m oldserver=newserver (n)` オプションを指定して実行してください。

サーバーの優先順位番号を変更する方法

- **nisprefadm** に **-m server=server(new)** オプションを指定して実行します。

```
#nisprefadm -L|-G -C name -m oldserver=newserver(n)
```

引数の意味はそれぞれ以下のとおりです。

- **L|G** には、ローカルファイルとグローバルテーブルのどちらを修正するかを指定します。
- **-C name** には、サブネットの IP 番号またはマシン名が入ります。このオプションは、**-G** オプションを使用している場合にだけ使用します。このコマンドを使用して行なった変更は、ここに指定したサブネットまたはマシンに適用されます。
- **-m** は、サーバーリストを変更するオプションです。
- **old server** には、その優先順位番号を変更したいサーバー名が入ります。
- **new server(n)** には、サーバー名とその新しい優先順位番号が入ります。

たとえば、deneb マシン上で、deneb のローカル `client_info` ファイルの `replica6.manf` サーバーに指定された番号を 2 に変更する場合は、次ように入力します。

```
deneb# nisprefadm -L -m replica6.manf=replica6.manf(2)
```

優先順位リスト内で 1 つのサーバーを別のサーバーに置換する方法

優先順位リスト内で、1 つのサーバーを別のサーバーに変更するには、次のようにします。

- **nisprefadm** に **-m oldserver=newserver** オプションを指定して実行します。

```
#nisprefadm -L|-G -C name -m \
    oldserver=newserver(prefnumber)
```

引数の意味はそれぞれ以下のとおりです。

- **-L|-G** には、ローカルまたはドメイン全体のどちらのサーバーリストを変更するかを指定します。
- **-C name** には、サブネットの IP 番号またはマシン名が入ります。このオプションは、**-G** オプションを使用している場合にだけ使用します。このコマンドを使用して行なった変更は、ここに指定したサブネットまたはマシンに適用されます。
- **-m** は、サーバーリストを変更するオプションです。
- **oldserver** には、置換する対象となる旧サーバーが入ります。
- **newserver(prefnumber)** には、優先サーバーリスト内の旧サーバーの位置に入る新規サーバー (優先順位格付け番号を指定できる) を入力します。

-G オプションを使用して、グローバル `client_info` テーブル内のサーバーを置換する場合には、その置換内容は -c オプションで特定したサブネットまたはマシンにだけ適用されるので注意してください。それ以外にリストされている置換対象 (旧) サーバーは影響を受けません。

たとえば、3つのサブネットを持つドメインを所有していて、この中の2つのサブネット用に `replica1` サーバーが優先サーバーのリストに入っている場合を考えます。これ以上 `replica1` を使用しないので、使用サーバーから除外する場合は、`nisprefadm -m` を実行して、最初のサブネットの `replica1` を新規サーバーに置換します。この時、2番目のサブネットに対しても同様の処理を行うまでは、`replica1` はそのサブネットの優先サーバーのリストに入っています。個別のマシンの優先サーバーでも同様です。

たとえば、ドメインのグローバル `client_info` テーブル内で、サブネット `172.21.123.12` 用に指定された `replica3` サーバーを `replica6` サーバーに置換し、`replica6` に優先順位番号 1 を割り当てる場合は、次のように入力します。

```
nismaster# nisprefadm -G -C 172.21.123.12 -m replica3 replica6(1)
```

優先順位リストからサーバーを削除する方法

優先順位リストから 1 つまたは複数のサーバーを削除するには、次のようにします。

- **nisprefadm** に **-r** オプションを指定して実行します。

```
#nisprefadm -L|-G -C name -r servers
```

引数の意味はそれぞれ以下のとおりです。

- **-L|-G** には、ローカルまたはドメイン全体のサーバーリストのどちらを変更するのかを指定します。
- **-C name** には、サブネットの IP 番号またはマシン名が入ります。このオプションは、**-G** オプションを使用している場合にだけ使用します。このコマンドを使用した優先サーバーの削除は、ここに名前を指定したサブネットまたはマシンに適用されます。
- **-r** は、*servers* に名前を指定したサーバーをリストから削除するオプションです。

たとえば、ドメインのグローバル `client_info` テーブル内から、マシン `polaris` に指定された `replica3` および `replica6.manf` サーバーを削除する場合は、次のように入力します。

```
polaris# nisprefadm -G -C polaris -r replica3 replica6.manf
```

優先サーバーリスト全体を置換する方法

グローバル `client_info` テーブルまたはマシンのローカル `client_info` ファイルどちらかの中にある、サブネットまたはマシンに指定された優先サーバーのリスト全体を置換するには、`nisprefadm` に `-u` オプションを指定して実行してください。

`-u` オプションは、マシンまたはサブネットに指定されたサーバーの優先順位を、まず初めにすべて削除してから指定した新規の優先順位を追加しますが、これ以外は、`-a` オプションと同じ処理を行います。既存の優先順位がある場合、`-a` オプションでは旧リストに新しい優先順位を追加します。

`-u` オプションの使用例については、[389 ページの「個別のマシンにグローバル優先順位を設定する方法」](#)を参照してください。

優先サーバー限定指定

優先サーバーが使用できない場合に、クライアントがとる動作を指定できます。

デフォルトでは、クライアントは、優先サーバーに到達できない場合、検出できたサーバーであればどれでも使用してしまいます。優先サーバー限定オプションを設定すると、クライアントが優先サーバーだけを使用するように指定できます。優先サーバー限定オプションと全サーバーオプションの基礎知識については、[383 ページの「優先サーバー限定と全サーバー」](#)を参照してください。

優先サーバーが使用できない場合のクライアントの動作を指定するには、`nisprefadm` に `-o value` オプションを指定して実行してください。

優先サーバーだけを指定する方法

サーバーリストを使用するクライアントが、リスト内に記述されたサーバーだけから NIS+ 情報を入手するように指定するには、次のようにします。

- `nisprefadm` に `-o pref_only` オプションを指定して実行してください。

```
#nisprefadm -L|-G -o pref_only
```

引数の意味はそれぞれ以下のとおりです。

- `-L|-G` には、ローカルまたはドメイン全体のサーバーリストのどちらを変更するかを指定します。
- `-o pref_only` により、クライアントがリスト内に記述されたサーバーだけから NIS+ 情報を入手するように指定されます。

注 – このオプションは、優先サーバーが全くないドメインでは無視されます。

たとえば、altair のローカル client_info ファイル内に、altair が必ず優先サーバーを使用し、altair の優先サーバーリストにないサーバーは使用しないように指定するには、次のように入力します。

```
altair# nisprefadm -L -o pref_only
```

優先サーバー以外のサーバーを使用する方法

サーバーリストを使用するクライアントが、優先サーバーが使用できない場合に、リストに記載されていないサーバーから NIS+ 情報を入手するように指定するには、次のようにします。

- **nisprefadm** に **-o all** オプションを指定して実行してください。

```
#nisprefadm -L|-G -o all
```

引数の意味はそれぞれ以下のとおりです。

- **-L|-G** には、ローカルまたはドメイン全体のサーバーリストのどちらを変更するかを指定します。
- **-o all** は、クライアントが、優先サーバーが使用できない場合に、リストに記載されていないサーバーから NIS+ 情報を入手するように指定するオプションです。

注 – これは、デフォルトの動作です。そのため、**-o all** オプションは、以前に **-o pref_only** オプションを使用して優先サーバーだけを指定していた場合に限り使用する必要があります。

たとえば、altair が優先サーバーを使用できない場合には、altair のローカル client_info ファイル内に、優先サーバー以外のサーバーを使用できるように指定し直すには、次のように入力してください。

```
altair# nisprefadm -L -o all
```

サーバーの優先順位の使用の終了

使用サーバーのカスタマイズ機能の使用を終了し、[379 ページの「デフォルトでのクライアントの検索動作」](#)で説明した NIS+ 情報の入手方法に戻すことができます。

サーバーの優先順位の使用を終了するには、`nisprefadm` に `-x` オプションを指定して実行してください。

注 - サーバーの優先順位の使用を終了する場合、クライアントは、[385 ページ](#)の「サーバーの優先順位が有効になるタイミング」で説明していることがらが通常どおり進行するまではサーバーの優先順位の使用を終了しません。また、サーバーの優先順位の使用を強制的にただちに終了させることもできます。これについては、[397 ページ](#)の「サーバーの優先順位の有効化」で説明しています。

グローバルサーバー優先順位を削除する方法

- `nisprefadm` に `-G` および `-x` オプションを指定して実行してください。

```
#nisprefadm -G -x
```

これにより、グローバルサーバーの優先順位が削除されます。

- ローカルサーバーの優先順位が指定されていないクライアントマシンは、[379 ページ](#)の「デフォルトでのクライアントの検索動作」で説明したように NIS+ 情報入手します。
- ローカル `/var/nis/client_info` ファイルで設定されたサーバーの優先順位があるクライアントマシンは、引き続きそのファイルに指定されたサーバーを使用します。

ローカルサーバー優先順位を削除する方法

ローカル優先順位を終了するということは、次の異なる 3 つの事項のいずれか 1 つを意味すると考えられます。

- 特定のマシンで、サーバーの優先順位にローカル `client_info` ファイルの使用を終了し、かわってドメインのグローバル `client_info` テーブル内にそのサブネット用に設定された優先順位を使用したい場合
- このマシンで、サーバーの優先順位にローカル `client_info` ファイルの使用を終了し、かわってドメインのグローバル `client_info` テーブル内にそのマシン固有に設定された優先順位を使用したい場合
- 特定のマシンで、サーバーの優先順位をまったく使用したくない場合。マシンがサーバーの優先順位を使用しない場合、そのマシンは、[379 ページ](#)の「デフォルトでのクライアントの検索動作」に説明した方法で NIS+ 情報入手する

ローカルからグローバルサブネットの優先順位に置換する方法

- マシンの `/var/nis/client_info` ファイルを削除します。

```
# rm /var/nis/client_info
```

この結果、マシンはドメインのグローバル `client_info` テーブル内でマシンのサブネットに指定された優先順位を使用するようになります。

ローカルからマシン固有のグローバル優先順位に置換する方法

1. マシンの `/var/nis/client_info` ファイルを削除します。

```
# rm /var/nis/client_info
```

2. `-g` と `-c` オプションを使用して、グローバルテーブル内にそのマシン用の優先順位を指定します。

詳細は、[389 ページ](#)の「個別のマシンにグローバル優先順位を設定する方法」を参照してください。

マシンにサーバーの優先順位の使用を中止させる方法

1. マシンの `/var/nis/client_info` ファイルを削除します。

```
# rm /var/nis/client_info
```

マシンのドメインにグローバル `client_info` テーブルがない場合、必要な処理はこの手順だけです。ドメインに `client_info` テーブルがある場合は、[手順 2](#)に進んでください。

2. 空の `/var/nis/client_info` ファイルを作成します。

```
# touch /var/nis/client_info
```

マシンに固有の `/var/nis/client_info` ファイルがある場合、そのマシンは `client_info` テーブルのグローバル優先順位は使用しません。マシンに空の `/var/nis/client_info` ファイルがある場合、そのマシンはどの優先順位も使用せず、[379 ページ](#)の「デフォルトでのクライアントの検索動作」で説明した方法で NIS+ 情報を入手します。

サーバーの優先順位の有効化

使用サーバーの変更内容は、通常、クライアントマシンを再起動するか、またはマシンのキャッシュ管理プログラムを更新した時に有効になります。

ローカル `client_info` ファイル (`-L` オプション) を使用するローカルマシン上で、`nisprefadm` を使用してサーバーの優先順位の設定または変更した場合は、その変更内容はただちに有効になります。

グローバル `client_info` テーブル (`-G` オプション) からサーバーの優先順位を入手しているマシンの場合、`nisprefadm` に `-F` オプションを指定して実行すると、サーバーの優先順位に加えた変更を強制的にただちに有効にできます。

```
# nisprefadm -F
```

`-F` オプションとは、マシンのキャッシュ管理プログラムに、そのサーバーの優先順位情報をドメインのグローバル `client_info` テーブルを使って強制的にただちに更新させるオプションです。`nisprefadm -F` を実行するマシンが、`/var/nis` 内にそのマシン固有のローカル `client_info` ファイルを持つ場合、そのマシン上で `nisprefadm -F` を実行しても無効になります。

注 `-F` オプションは、ほかの `nisprefadm` オプションと同時に使用できません。`nisprefadm -F` コマンドは、必ず、このコマンドを適用するマシン上で、単独で使用してください。`-G` オプションを使用して、ドメイン内にあるすべてのマシンのキャッシュ管理プログラムを更新できません。`nisprefadm -F` コマンドは、各マシン上でそれぞれ実行する必要があります。

優先順位の変更内容をただちに実現する方法

新しく作成、または変更したサーバーリストを、指定したマシン上で強制的にただちに有効にするためには、次のようにします。

- マシン上で、**`nisprefadm`** に **`-F`** オプションを指定して実行します。

```
# nisprefadm -F
```

たとえば、`vega` の優先サーバーリストへの変更内容を、強制的にただちに実現させるには (ローカルまたはグローバルのどちらの場合でも)、次のように入力してください。

```
vega# nisprefadm -F
```

第 21 章

NIS+ のバックアップと復元

この章では、NIS+ 名前空間のバックアップ方法と復元方法について説明します。

NIS+ のバックアップ機能と復元機能を使用すると、NIS+ 名前空間の保存と復元を素早く簡単に行うことができます。また、これらの機能を使用すると、新しい複製サーバーを簡単に作成でき、さらにそのサーバーをオンラインにするためにかかる時間を削減できます。これらのタスクは、次の 2 種類のコマンドを使用して実行します。

- `nisbackup`. NIS+ ディレクトリオブジェクトをバックアップする
- `nisrestore`. NIS+ ディレクトリオブジェクトを復元する

注 - NIS+ は、将来のリリースではサポートされなくなる可能性があります。NIS+ から LDAP への移行支援ツールは、Solaris 9 リリース以降で使用できます (『Solaris のシステム管理 (ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』を参照)。詳細については、

<http://www.sun.com/directory/nisplus/transition.html> を参照してください。

`nisbackup` を使用して名前空間をバックアップする

`nisbackup` コマンドは、1 つまたは複数の NIS+ ディレクトリオブジェクト、または 1 つの名前空間全体を、指定した UNIX ファイルシステムのディレクトリにバックアップします。

注 - `nisbackup` コマンドは、必ずマスターサーバー上で実行してください。複製サーバー上では絶対に実行しないでください。

`nisbackup` コマンドは、バックアップコマンドが動作するように設定された時点の NIS+ 名前空間をコピーします。この記録には、現行のすべての NIS+ データと、認証されたネットワーク管理者が NIS+ 名前空間に入力した変更が含まれます。ただし、まだ NIS+ テーブルにチェックポイントされていない (NIS+ テーブルに記入されていない) ものは除きます。このバックアップ処理は、NIS+ データのチェックや訂正は行いません。テーブル内のデータが破壊されると、破壊されたデータは、有効なデータと見分けがつかない状態にバックアップされます。

`nisbackup` コマンドは、マシンがそのマスターサーバーであるディレクトリオブジェクトだけをバックアップします。つまり、`nisbackup` は、マスターサーバー上でだけ使用でき、複製サーバー上では使用できません。

- マシンが、ドメインとサブドメイン両方の NIS+ ディレクトリオブジェクトのマスターサーバーである場合、`nisbackup` をこのマシン上で実行するとドメインとサブドメイン両方のディレクトリオブジェクトをバックアップできます。
- ただし、マシンが、1つのディレクトリオブジェクトのマスターサーバーで、さらに別のディレクトリオブジェクトの複製サーバーである場合、`nisbackup` を実行してそのマシンがマスターサーバーであるディレクトリオブジェクトをバックアップできますが、複製サーバーのオブジェクトはバックアップされません。

バックアップ処理が、ほかの処理に割り込まれたり、またはその処理を正常に終了できない場合は、処理を停止し、転送先ディレクトリ内に格納された以前のバックアップファイルをすべて復元します。

`nisbackup` の構文

`nisbackup` コマンドで使用する構文は、次のとおりです。

```
nisbackup [-v] [-a] backupdir objects
```

引数の意味はそれぞれ以下のとおりです。

- `backupdir` には、バックアップファイルを格納する転送先ディレクトリが入ります。たとえば、`/var/master1_backup` です。
- `objects` には、バックアップする NIS+ ディレクトリオブジェクトが入ります。たとえば、`org_dir.doc.com` です。複数の NIS+ ディレクトリオブジェクトを、スペースで区切って入れることができます。

`nisbackup` コマンドには、次のようなオプションを指定できます。

表 21-1 nisrestore コマンドのオプション

オプション	目的
-v	冗長モード。このモードは、追加情報を出力する
-a	すべて。サーバーがマスターである NIS+ ディレクトリオブジェクトをすべてバックアップする。これには、このサーバーがマスターであるサブドメインのディレクトリオブジェクトも含まれる。ただし、ほかのマスターサーバーを持つサブドメインのディレクトリオブジェクトはバックアップされない

nisbackup コマンドは、バックアップする NIS+ ディレクトリオブジェクトのマスターサーバー上で実行する必要があります。

バックアップする NIS+ ディレクトリオブジェクトを指定する場合、そのディレクトリ名には完全指定名、または部分指定名を使用できます。

マルチレベルディレクトリをバックアップする場合、下位ディレクトリのバックアップファイルは、自動的にバックアップ転送先ディレクトリのサブディレクトリ内に配置されます。

nisbackup によるバックアップの対象

nisbackup を使用する場合、nisbackup はサーバー固有のコマンドであることに注意してください。-a オプションを使用するかどうかに関係なく、nisbackup は、このコマンドを実行中のサーバーがマスターサーバーであるディレクトリだけをバックアップします。ほかにマスターサーバーを持つ NIS+ ディレクトリオブジェクトは、バックアップされません。

たとえば、submaster1 は、sales.doc.com. ディレクトリオブジェクトのマスターサーバーで、さらに west.sales.doc.com. ディレクトリオブジェクトの複製サーバーでもあるとします。この場合、submaster1 上で nisbackup を実行すると、sales.doc.com. ディレクトリオブジェクトだけがバックアップされます。

このサーバー固有の原則には次のものがあります。

- 「全 NIS+ 名前空間」。複数のドメインのすべての名前空間に、NIS+ バックアップを実行する場合で、ルートマスターサーバーが全サブドメインのマスターサーバーでもある場合は、ルートマスター上で nisbackup に -a オプションを指定して実行します。ただし、スーパーユーザーのマスターサーバーが、すべてのサブドメインのマスターサーバーでない場合は、すべての名前空間のバックアップを完全に実施するために、nisbackup をほかのマスターサーバー上でも実行する必要があります。
- 「サブドメイン」。1 つまたは複数のサブドメインの NIS+ バックアップを実行する場合、サブドメインのマスターサーバー上で nisbackup を実行する必要があります。ルートマスターなどの 1 つのマシンが、1 つまたは複数のサブドメインのマスターでもある場合、そのマシン上で nisbackup に -a オプションを指定して実行します。

バックアップ転送先ディレクトリ

バックアップ転送先ディレクトリは、バックアップ対象のサーバーが使用できるものでなければなりません。サーバー上に物理的にマウントしていない転送先ディレクトリを使用するのも良い方法です。この場合、サーバーがダメージを受けても、バックアップディレクトリは使用可能です。

独立した転送先ディレクトリは、バックアップ対象となるマスターサーバーごとに使用する必要があります。混乱を避けるために、マスターサーバーのマシン名を転送先ディレクトリ内に組み込むと良いでしょう。たとえば、master1 マシン上で実行した nisbackup の転送先ディレクトリは、/var/master1_backup という名前にします。



注意 – 指定した 1 つの転送先ディレクトリに対して、複数のサーバーをバックアップしないでください。異なるマスターサーバーには、必ず、異なる転送先ディレクトリを使用してください。それは、指定した転送先ディレクトリに、1 つまたは複数の NIS+ ディレクトリオブジェクトをバックアップするたびに、このディレクトリ内のこれらの NIS+ ディレクトリオブジェクト用のそれ以前のバックアップファイルが上書きされるからです。

NIS+ のバックアップを日付順に保存する

バックアップファイルを日付順に保存するには、少なくとも次の 2 つの方法があります。

- 「別々の転送先ディレクトリとして保存」。異なる転送先ディレクトリは、バックアップした日付ごとに保持できます。たとえば、
/var/master1_backup/July14、/var/master1_backup/July15、など。この方法は簡単ですが、ディスク領域がかなり必要です。
- 「ファイルシステムのバックアップ」。NIS+ バックアップを日付順に保存するもっとも一般的な方法は、使用する通常ファイルシステムのバックアップメソッドに、バックアップ転送先ディレクトリを単に組み込むだけの方法です。これを簡単に行うには、nisbackup コマンドを crontab ファイルから実行するか、または Solstice バックアップルーチン内から実行します。nisbackup のようなコマンドを、システムのバックアッププロシージャとして自動的に実行するように指定する方法については、Solstice の説明書を参照してください。

特定の NIS ディレクトリをバックアップする

特定の NIS+ ディレクトリオブジェクトをバックアップするには、これらのディレクトリをバックアップ転送先ディレクトリの後ろに入力します。

たとえば、ルート、sales ドメイン、manf ドメインの 3 つの org_dir ディレクトリオブジェクトを /master1_backup ディレクトリにバックアップするには、nisbackup を master1 マシン上で次のように実行します。

```
master1# nisbackup /var/master1_backup org_dir org_dir.sales org_dir.manf
```

すべての NIS+ 名前空間をバックアップする

すべての NIS+ 名前空間をバックアップする場合は、ルートマスターサーバー上で、`nisbackup` コマンドに `-a` オプションを指定して実行します。

`-a` オプションを使用する場合は、バックアップする NIS+ ディレクトリオブジェクトは指定しません。サーバー上とそのサーバーの下にあるサブドメインのすべての NIS+ ディレクトリオブジェクトは、自動的にバックアップされます。

たとえば、`doc.com.` 名前空間を `/master1_backup` ディレクトリにバックアップするには、ルートマスター上で、`nisbackup` を次のように実行してください。

```
rootmaster# nisbackup -a /var/master1_backup
```

バックアップディレクトリの構造

ドメイン上でバックアップを実行すると、バックアップ転送先ディレクトリ内に、NIS+ ディレクトリオブジェクトごとにサブディレクトリが作成されます。これらのサブディレクトリ名は、完全指定の NIS+ ディレクトリオブジェクト名の末尾にピリオドが付いたものになります。

`-a` オプションを使用して NIS+ オブジェクト全体を完全にバックアップした場合は、3つの関連ディレクトリオブジェクト (`domain.`、`org_dir.domain.`、`groups_dir.domain.`) がすべてバックアップされ、3つのターゲットサブディレクトリが作成されます。複数のオブジェクトをバックアップすると、サブディレクトリはバックアップしたそれぞれのオブジェクトごとに作成されます。

複数の NIS+ ディレクトリオブジェクトのバックアップサブディレクトリは、それがサブドメインであるかどうかに関係なく、親バックアップ転送先ディレクトリのサブディレクトリになるので注意してください。つまり、`nisbackup` は、親バックアップ転送先ディレクトリの下にドメインの階層を複製しません。その代わりに、バックアップサブディレクトリはすべて、転送先ディレクトリの単純なサブディレクトリになります。

たとえば、`doc.com.` のルート、`sales`、`manf` のディレクトリオブジェクトを `/var/master1_backup` ディレクトリにバックアップする場合、[図 21-1](#) に示すように、`/var/master1_backup` ディレクトリ内には9個のサブディレクトリが作成されます。

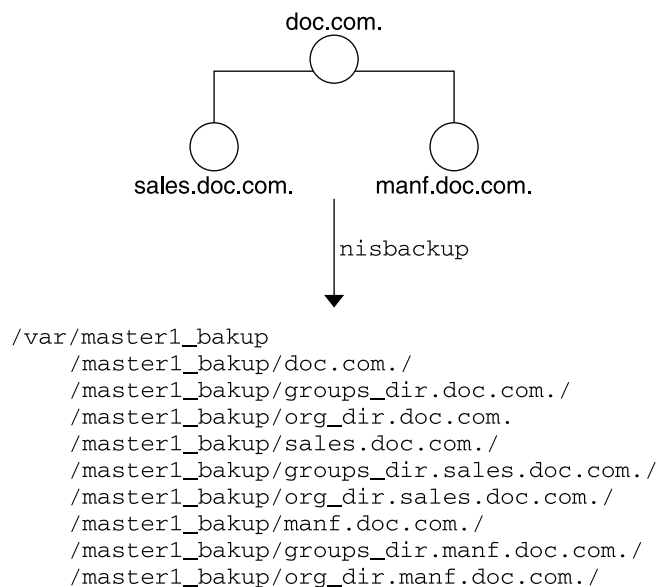


図 21-1 nisbackup によって作成されたディレクトリの例

バックアップファイル

バックアップ転送先ディレクトリには、この転送先ディレクトリにバックアップされた最新の NIS+ ディレクトリオブジェクトを表示する `backup_list` ファイルが入っています。

各サブディレクトリには、ファイルが 2 つと `/data` サブディレクトリが 1 つ組み込まれます。このファイルを次に示します。

- `data.dict` このディレクトリにバックアップされた NIS+ ディレクトリオブジェクトの NIS+ データ辞書の入った XDR コード化ファイル
- `last.upd` このディレクトリにバックアップされた NIS+ ディレクトリオブジェクトに関する時刻スタンプ情報が入ったバイナリファイル

各 `/data` サブディレクトリには、1 つまたは複数の以下のファイルが入っています。

- `root.object`。NIS+ ルートディレクトリオブジェクトの説明の入った XDR コード化ファイル。たとえば、`/master1_backup/doc.com/data/root.object`
- `root_dir`。これらのオブジェクトのルートディレクトリとサーバー情報内に組み込まれた NIS+ オブジェクトの説明が入った XDR コード化ファイル。たとえば、`/master1_backup/doc.com/data/root_dir`
- `table.directory`。バックアップの実行時に、NIS+ テーブル内に表示されていたデータと関連するすべての NIS+ ログファイル内に含まれるデータがすべて入った XDR コード化ファイル。バックアップされた NIS+ ディレクトリオブジェクト内

に NIS+ テーブルがある場合、対応する *table.directory* バックアップファイルが、そのディレクトリオブジェクトの /data サブディレクトリ内に作成される。

たとえば、それぞれの NIS+ *org_dir* ディレクトリには、*hosts* テーブルが含まれるため、各 *target/org_dir.domain/data* サブディレクトリには、*hosts.org_dir* がある。たとえば、
/master1_bakup/org_dir.doc.com./data/hosts.org_dir。

指定したディレクトリオブジェクト内に表示されたユーザー作成の NIS+ テーブルは、標準 NIS+ テーブルとして同じ方法でバックアップされる

- NIS+ グループ情報の入った XDR コード化ファイル。このファイルは、対応する NIS+ *groups_dir* 転送先ディレクトリに格納される

nisrestore を使用して NIS+ 名前空間を復元する

nisrestore コマンドによって、nisbackup を使用して作成したバックアップファイル内に格納されたデータと一致する NIS+ ディレクトリオブジェクトが再現されます。このコマンドを使用すると、NIS+ サーバーの復元、壊れたディレクトリオブジェクトの置換、または新しい NIS+ サーバーに NIS+ データを読み込みます。

nisrestore を実行するための前提条件

nisrestore を使用するには、nisrestore から NIS+ データを受け取るマシンは、NIS+ サーバーとして設定されている必要があります (NIS+ サーバーの設定の詳細については、第 4 章を参照)。この場合「サーバーが認識している」とは以下のことを意味する

- マシンを、NIS+ クライアントとして初期化しておく必要があります。
- マシンを NIS 互換モードで実行し、ドメイン名システム (DNS) 転送をサポートする場合は、そのマシンには、適切に構成された /etc/resolv.conf ファイルが必要です。
- 名前空間内でほかのサーバーが実行中の時、あるサーバーで nisrestore を使用する場合は、nisrestore は、ほかのサーバーを検証して、このサーバーがサーバーに復元するバックアップ NIS+ オブジェクトを配布するように構成されているかどうかを確認します。実行中のサーバーがほかにはない場合は、nisrestore に -f オプションを指定して実行する必要があります。つまり、nisrestore がチェックするほかのサーバーがある場合は、-f オプションを使用する必要はありません。また、使用できるサーバーがほかにはない場合、たとえば、1 台のマスターサーバーを復元するときに、機能複製サーバーがほかにはない場合は、-f オプションを使用する必要があります。



注意 – 上記の 3 つの前提条件への追加条件として、マシン上で `rpc.nisd` デーモンを実行しないでください。必要に応じて、NIS+ サービスを停止して `rpc.nisd` のプロセスを終了してから、`nisrestore` を実行してください。

nisrestore の構文

`nisrestore` では、次の構文を使用します。

```
nisrestore [-fv] [-a] [-t] backupdir [directory_objects]
```

引数の意味はそれぞれ以下のとおりです。

- `backupdir` には、NIS+ オブジェクトの復元に使用するバックアップファイルの入ったディレクトリを入力します (`/var/master1_bakup`)。たとえば、`/var/master1_bakup` です。
- `directory_objects` には、復元する NIS+ ディレクトリオブジェクトを入力します (`org_dir.doc.com`)。たとえば、`org_dir.doc.com` です。複数の NIS+ ディレクトリオブジェクトを、スペースで区切って入れることができます。`nisrestore` に `-a` オプションを指定して実行する場合は、特定のディレクトリオブジェクトは指定しません。

`nisrestore` コマンドには、以下のオプションを指定できます。

表 21-2 `nisrestore` コマンドのオプション

オプション	目的
-a	すべて。バックアップディレクトリ内に入っている NIS+ ディレクトリオブジェクトをすべて復元する
-f	サーバーが、ディレクトリオブジェクトのサーバーリストに記載されているかどうかを検査せずに、強制的に復元を行う。このオプションは、ルートマスターサーバーを復元する時、または“オブジェクトを検出できません”といった種類のエラーを受け取った場合に使用する必要がある
-v	冗長モード。このモードは、追加情報を出力する
-t	このオプションを使用すると、バックアップディレクトリ内に格納された NIS+ ディレクトリオブジェクトがすべて表示される。オブジェクトの復元は行われない

nisrestore を使用する

NIS+ バックアップファイルから NIS+ データを復元するには、`nisrestore` コマンドを使用します。

たとえば、org_dir.doc.com. ディレクトリオブジェクトを replica1 サーバーに復元する場合は、root として replica1 にログインし、405 ページの「[nisrestore を実行するための前提条件](#)」で説明した前提条件が満たされていることを確認してから、以下のように nisrestore を実行します。

```
replica1# nisrestore /var/master1_backup org_dir.doc.com.
```

nisrestore には、以下の項目が適用されます。

- 「損傷した名前空間」。損傷した、または破壊された NIS+ 名前空間を復元するには、復元する NIS+ ディレクトリオブジェクトのすべてのサーバー上で nisrestore コマンドを実行する必要があります。
- 「検出エラー」。nisrestore が必要なデータを確認できないか、または検出できないというエラーメッセージを受け取った場合は、-f オプションを使用する必要があります。

たとえば、master1 という名前のルートマスターサーバー上に NIS+ データをロードし直す場合は、次のように入力します。

```
master1# nisrestore -f -a /var/master1_backup
```

- 「ディレクトリ名」。復元する NIS+ ディレクトリオブジェクトを指定する場合は、完全指定または部分指定ディレクトリ名を使用します。

バックアップと復元を使用して複製サーバーを設定する

NIS+ バックアップおよび復元機能を使用すると、NIS+ データを新しい複製サーバーに速く読み込むことができます。名前空間が広い場合は、この方法の方が、nisping を使用するよりもマスターサーバーからのデータを非常に速く入手できます。

nisbackup と nisrestore を使用して新しい複製サーバーを設定するには、次の手順を行います。

1. マスター上で **nisservice** を実行し、新しい複製サーバーを作成します。
2. 新しい複製サーバー上の **NIS+** サービスを停止します。
svcadm disable を使用します。この処理は、nisping コマンドを使用した、マスターから複製への名前空間データの自動転送に割り込んで実行されます。
3. マスターサーバー上で、**nisbackup** を実行します。
4. 新しい複製サーバー上で **nisrestore** を実行し、**NIS+** データを読み込みます。
5. 新しい複製サーバー上で **NIS+** サービスを再起動します。
svcadm restart を使用します。

サーバーマシンを置換する

`nisbackup` と `nisrestore` を使用すると、サーバーとして使用中のマシンと別のマシンをすぐに置換できます。たとえば、旧サーバーを新しい高速のサーバーと交換すると、ネットワークのパフォーマンスを向上させることができます。

マシンを置換する場合の必要条件

NIS+ サーバーとして使用中のマシンをほかのマシンに置き換える場合には、次の条件が必要です。

- 新しいマシンには、置換する旧マシンと同じ IP アドレスを割り当てます。
- 新しいマシンには、置換する旧マシンと同じマシン名を割り当てます。
- 新しいマシンは、置換する旧マシンと同じサブネットに接続します。

サーバーマシンの置換方法

サーバーマシンを置換する場合は、次の手順に従ってください。

1. 旧サーバーが管理するドメインのマスターサーバー上で **`nisbackup`** を実行します。
詳細は、[403 ページの「すべての NIS+ 名前空間をバックアップする」](#)を参照してください。置換する旧サーバーがマスターサーバーである場合もあるので注意してください。この場合は、この旧マスターサーバー上で `nisbackup` を実行します。
2. 旧サーバーの `/var/nis/NIS_COLD_START` ファイルをバックアップディレクトリにコピーします。
3. 旧サーバーの `/etc/.rootkey` ファイルをバックアップディレクトリにコピーします。
4. 旧サーバーをネットワークから切り離します。
5. 新しいサーバーをネットワークに接続します。
6. 新しいサーバーに旧サーバーと同じ IP アドレス (番号) を割り当てます。
7. 新しいサーバーに旧サーバーと同じマシン名を割り当てます。
8. 必要に応じて、新しいサーバー上の **NIS+** サービスを停止します。
9. 新しいサーバー上で **`nisrestore`** を実行し、**NIS+** データを読み込みます。
詳細は、[405 ページの「`nisrestore` を使用して NIS+ 名前空間を復元する」](#)を参照してください。

10. **.rootkey** ファイルを、バックアップディレクトリから新しいサーバーの **/etc** にコピーします。
11. **NIS_COLD_START** ファイルを、バックアップディレクトリから新しいサーバーの **/var/nis** にコピーします。
12. 新しいサーバーを再起動します。

第 22 章

NIS+ の削除

この章では、NIS+ ディレクトリ管理コマンドを使用して、クライアントまたはサーバーから NIS+ を削除する方法と、NIS+ 名前空間全体を削除する方法について説明します。

NIS+ 複製サーバーを、ディレクトリから分離し、そのドメインの複製サーバーとして機能しないようにする場合は、336 ページの「[nisrmdir コマンド](#)」を参照してください。

注 - NIS+ は、将来のリリースではサポートされなくなる可能性があります。NIS+ から LDAP への移行支援ツールは、Solaris 9 リリース以降で使用できます (『Solaris のシステム管理 (ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』を参照)。詳細については、<http://www.sun.com/directory/nisplus/transition.html> を参照してください。

注 - NIS+ サービスは、サービス管理機能によって管理されます。このサービスに対する有効化、無効化、再起動などの管理操作を実行するには、svcadm コマンドを使用します。SMF を NIS+ で使用する方法については、89 ページの「[NIS+ とサービス管理機能](#)」を参照してください。SMF の概要については、『Solaris のシステム管理 (基本編)』の「サービスの管理 (概要)」を参照してください。詳細については、svcadm (1M) と svc (1) の各マニュアルページも参照してください。

クライアントマシンから NIS+ を削除する

この節では、クライアントマシンから NIS+ を削除する方法について説明します。ただし、クライアントマシンから NIS+ を削除しても、ネットワークから NIS+ ネームサービスを削除したことにはならないので注意してください。ネットワークから NIS+ ネームサービスを削除して、NIS または /etc ディレクトリのファイルをネームサービスとして使用する状態に戻す場合は、[414 ページの「NIS+ 名前空間を削除する」](#)を参照してください。

nisclient を使用してインストールした NIS+ を削除する

[第 4 章](#)の説明に沿って `nisclient -i` スクリプトを使用して、NIS+ クライアントとして設定されたクライアントマシンから NIS+ を削除するには、`-r` オプションを付けて `nisclient` を実行します。

```
client# nisclient -r
```

`nisclient -r` では、`nisclient -i` の最新の処理が 1 回分取り消されるだけです。つまり、`-i` 実行以前にクライアントによって使用されていたネーミングシステム (NIS や /etc ディレクトリのファイルなど) が再び使用されるようになります。

NIS+ コマンドでインストールした NIS+ を削除する

[第 4 章](#)の説明に沿って `nisaddcred`、`domainname`、および `nisinit` コマンドを使用して、NIS+ クライアントとして設定されたクライアントマシンから NIS+ を削除するには、次の手順を行います。

1. **.rootkey** ファイルを削除します。

```
client# rm -f /etc/.rootkey
```

2. キーサーバーを停止します。

```
client# svcadm disable /network/rpc/keyserv
```

3. **NIS+** サービスを停止します。

これにより、`rpc.nisd` デーモンと `nis_cachemgr` が停止します。

```
client# svcadm disable /network/rpc/nisplus:default
```

4. ネームサービスキャッシュ (**nscd**) を停止します。

```
client# svcadm disable /system/name-service-cache:default
```

5. **/var/nis** ディレクトリとその下のファイルを削除します。

```
clientmachine# rm -rf /var/nis/*
```

サーバーから NIS+ を削除する

この節では、NIS+ サーバーから NIS+ を削除する方法を示します。

ただし、サーバーから NIS+ を削除しても、ネットワークから NIS+ ネームサービスを削除したことにはならないので注意してください。ネットワークから NIS+ ネームサービスを削除して、NIS または `/etc` ディレクトリのファイルをネームサービスとして使用する状態に戻す場合は、[414 ページの「NIS+ 名前空間を削除する」](#)を参照してください。

注 - NIS+ サーバーとして使用しているマシンを、別のマシンに置換できます。
[408 ページの「サーバーマシンを置換する」](#)を参照してください。

サーバーから NIS+ を削除する手順は以下のとおりです。

1. クライアントから **NIS+** を削除する作業を行います。

NIS+ サーバーは NIS+ クライアントでもあります。このため、クライアントに関連する NIS+ の部分を最初に削除する必要があります。このためには `nisclient -r` ([412 ページの「nisclient を使用してインストールした NIS+ を削除する」](#)を参照) か、NIS+ コマンド ([412 ページの「NIS+ コマンドでインストールした NIS+ を削除する」](#)を参照) を使用します。

2. サーバーの **groups_dir** ディレクトリと **org_dir** ディレクトリを削除します。

```
server# nisrmdir -f groups_dir.domainname
server# nisrmdir -f org_dir.domainname
```

3. キーサーバーを停止します。

```
client# svcadm disable /network/rpc/keyserv
```

4. **NIS+** サービスを停止します。

これにより、`rpc.nisd` デーモンと `nis_cachemgr` が終了します。

```
server# svcadm disable /network/rpc/nisplus:default
```

5. ネームサービスキャッシュ (**nscd**) を停止します。

```
server# svcadm disable /system/name-service-cache:default
```

6. **/var/nis** ディレクトリとその下のファイルを削除します。

```
rootmaster# rm -rf /var/nis/*
```

NIS+ 名前空間を削除する

NIS+ 名前空間を削除し、NIS または **/etc** ディレクトリのファイルをネームサービスとして使用する状態に戻す手順は以下のとおりです。

1. ルートマスターから **.rootkey** ファイルを削除します。

```
rootmaster# rm -f /etc/.rootkey
```

2. ルートマスターのルートドメインから **groups_dir** サブディレクトリと **org_dir** サブディレクトリを削除します。

```
rootmaster# nisrmdir -f groups_dir.domainname
```

```
rootmaster# nisrmdir -f org_dir.domainname
```

domainname には、ルートドメイン名 (**doc.com** など) が入ります。

3. ルートドメインを削除します。

```
rootmaster# nisrmdir -f domainname
```

domainname には、ルートドメイン名 (**doc.com** など) が入ります。

4. キーサーバーを停止します。

```
client# svcadm disable /network/rpc/keyserv
```

5. **NIS+** サービスを停止します。

これにより、**rpc.nisd** デーモンと **nis_cachemgr** が終了します。

```
rootmaster# svcadm disable -t /network/rpc/nisplus:default
```

6. ネームサービスキャッシュ (**nscd**) を停止します。

```
rootmaster# svcadm disable -t /system/name-service-cache:default
```

7. 新しいドメインを作成します

```
rootmaster# domainname name
```

name には、新しいドメイン名 (**NIS+** インストール前のドメイン名など) が入ります。

8. 既存の **/etc/defaultdomain** ファイルを削除します。

```
rootmaster# rm /etc/defaultdomain
```

9. **/etc/defaultdomain** ファイルを、新しいドメイン名を使用して作成し直します。

```
rootmaster# domainname > /etc/defaultdomain
```

10. **nsswitch.conf** ファイルをもとのファイルに戻します。

サーバーを **nisserver -r** を使用して設定した場合は、以下のコマンドを使用します。

```
rootmaster# cp /etc/nsswitch.conf.no_nisplus /etc/nsswitch.conf
```

また、デフォルトスイッチテンプレートファイルの1つをコピーする方法もあります。NIS スイッチのデフォルトファイルテンプレートを使用する場合は、以下のコマンドを入力します。

```
rootmaster# cp /etc/nsswitch.nis etc/nsswitch.conf
```

/etc ファイルのデフォルトスイッチファイルテンプレートを使用する場合は、以下のコマンドを入力します。

```
rootmaster# cp /etc/nsswitch.files etc/nsswitch.conf
```

11. **/var/nis** ディレクトリとその下のファイルを削除します。

```
rootmaster# rm -rf /var/nis/*
```

12. **NIS+** サービスを開始します。

```
rootmaster# svcadm enable /network/rpc/nisplus:default
```


第 23 章

NIS+ テーブルの情報

この章では、Solaris ソフトウェアで提供されているデフォルトの NIS+ テーブルについて概要を説明します。対応するマニュアルページも参照してください。

- 419 ページの「auto_home テーブル」
- 419 ページの「auto_master テーブル」
- 420 ページの「bootparams テーブル」
- 422 ページの「client_info テーブル」
- 423 ページの「ethers テーブル」
- 423 ページの「group テーブル」
- 424 ページの「hosts テーブル」
- 424 ページの「mail_aliases テーブル」
- 425 ページの「netgroup テーブル」
- 426 ページの「netmasks テーブル」
- 427 ページの「networks テーブル」
- 427 ページの「passwd テーブル」
- 429 ページの「protocols テーブル」
- 430 ページの「rpc テーブル」
- 430 ページの「services テーブル」
- 431 ページの「timezone テーブル」

注 - NIS+ は、将来のリリースではサポートされなくなる可能性があります。NIS+ から LDAP への移行支援ツールは、Solaris 9 リリース以降で使用できます (『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』を参照)。詳細については、<http://www.sun.com/directory/nisplus/transition.html> を参照してください。

NIS+ のテーブル

NIS+ 環境では、ほとんどの名前空間の情報は、NIS+ テーブルに格納されます。

ネームサービスがないと、ほとんどのネットワーク情報は /etc ファイルに格納され、ほとんどすべての NIS+ テーブルが対応する /etc ファイルを持ちます。NIS サービスでは、/etc ファイルにほとんど対応する NIS マップにネットワーク情報を格納しました。

注 - この章では、NIS+ の一部として配布されたものだけを説明します。ユーザーとアプリケーション開発者は、目的に応じて NIS+ との互換性があるテーブルを頻繁に作成します。ユーザーと開発者によって作成されたテーブルの詳細は、マニュアルを参照してください。

すべての NIS+ テーブルは、groups_dir ディレクトリオブジェクトに格納される admin とグループテーブルを除いて、ドメインの org_dir NIS+ ディレクトリオブジェクトに格納されます。

注 - テーブルエントリはリンクしないでください。テーブルはほかのテーブルにリンクされますが、あるテーブルのエントリを別のテーブルのエントリにリンクしないでください。

NIS+ テーブルとほかのネームサービス

Solaris システムでは、ネームサービスのスイッチファイル (nsswitch.conf) によって、1 つ以上のソースを異なる名前空間の情報に対して指定できます。NIS+ テーブルのほかに、ソースは NIS マップ、DNS ゾーンファイル、/etc テーブルにすることができます。これらをスイッチファイルで指定する順番によって、異なるソースから情報が組み合わされる方法が決まります。スイッチファイルの詳細は第 1 章を参照してください。

NIS+ テーブル入力ファイルフォーマット

これらテーブルのどれかに対して入力ファイルを作成している場合、ほとんどのテーブルは、次の 2 つのフォーマットの条件を共有します。

- エントリごとに 1 行を使う
- 1 つ以上のスペースまたはタブで列を分ける

特定のテーブルに、異なるまたは追加のフォーマットの条件がある場合には、「入力ファイルフォーマット」の項で説明します。

auto_home テーブル

auto_home テーブルは間接オートマウントマップです。このマップによって NIS+ クライアントは、ドメイン内の任意のユーザーのホームディレクトリをマウントできます。このテーブルは、各ユーザーのホームディレクトリのマウントポイント、各ホームディレクトリの位置と、マウントオプションがあればそれを指定してマウントします。これは間接マップであるため、マウントポイントの最初の部分は auto_master テーブルに指定され、デフォルトでは /home となります。マウントポイントの 2 番目の部分 (つまり、/home の下にあるサブディレクトリ) は auto_home マップ内のエントリによって指定され、ユーザーごとに異なります。

auto_home テーブルには 2 つの列があります。

表 23-1 auto_home テーブル

列	コメント	説明
キー	マウントポイント	ドメイン内のユーザーのログイン名
値	オプションと位置	ユーザーごとのマウントオプションがあればそれと、ユーザーのホームディレクトリの位置

たとえば、以下のようになります。

```
costas barcelona:/export/partition2/costas
```

ユーザー costas のホームディレクトリは、サーバー barcelona 上にあり、ディレクトリ /export/partition2/costas 内にあります。またこのホームディレクトリは、クライアントの /home/costas ディレクトリの下にマウントされます。このエントリにはマウントオプションはありません。

auto_master テーブル

auto_master テーブルは、ドメイン内のすべてのオートマウントマップを含みます。直接マップの場合、auto_master テーブルは、単にマップ名を提供するだけです。間接マップの場合、このテーブルは、そのマウントポイントの先頭ディレクトリとマップ名の両方を提供します。auto_master テーブルには次の 2 つの列があります。

表 23-2 auto_master テーブル

列	コメント	説明
キー	マウントポイント	マップがマウントされる先頭ディレクトリ。マップが直接マップの場合、これは /- で表されるダミーディレクトリである
値	マップ名	オートマウントマップの名前

たとえば、auto_master テーブルに次のエントリがあるとします。

```
/home auto_home
/-auto_man
/programs auto_programs
```

最初のエントリは auto_home マップを指定します。このエントリによって、auto_home マップのすべてのエントリに対して、マウント先の最上位ディレクトリ /home が指定されます。auto_home マップは、間接マップです。2 番目のエントリには、auto_man マップを指定します。このマップは直接マップであるため、このエントリはマップ名だけを与えます。auto_man マップは、それ自体で、その各エントリに対するマウントポイントの完全パス名だけではなく、最上位ディレクトリも提供します。3 番目のエントリは auto_programs マップを指定します。このエントリはマウントポイントの最上位ディレクトリを提供するため、auto_programs マップは間接マップです。

すべてのオートマウントマップは NIS+ テーブルとして格納されます。デフォルトでは、Solaris システムは必須の auto_master マップ、および非常に便利な auto_home マップを提供します。

1 つのドメインに対してさらに多くのオートマウントマップを作成できますが、これらは必ず NIS+ テーブルとして格納し、auto_master テーブルに登録してください。その他のオートマウントマップを作成して、auto_master (ユーザーに対して作成された) に追加するときは、列名は「キー」と「値」にする必要があります。オートマウントの詳細は、オートマウント、または NFS ファイルシステムの関連マニュアルを参照してください。

bootparams テーブル

bootparams テーブルは、ドメイン内のすべてのディスクレスマシンに関する構成情報をもっています。ディスクレスマシンは、ネットワークに接続されていてもハードディスクをもたないマシンです。ディスクレスマシンにはディスク装置がないため、自分のファイルとプログラムをネットワーク上のサーバーのファイルシステムに持っています。また、自分の構成情報 (つまり「ブートパラメータ」) をサーバーに持っています。

このような構成になっているため、すべてのディスクレスマシンには、この情報がどこに格納されているかを知っている初期設定プログラムがあります。ネットワークにネームサービスがない場合、このプログラムはサーバーの `/etc/bootparams` ファイル内でこの情報を探します。ネットワークが NIS+ ネームサービスを使う場合、このプログラムは、代わりに `bootparams` テーブル内でこの情報を探します。

`bootparams` テーブルは、ディスクレスマシンに関するどんな構成情報でも格納できます。このテーブルには 2 つの列があり、構成キーとその値を格納します。デフォルトでは、このテーブルは各マシンのルート、スワップ、およびダンプの各パーティションの位置を格納するように設定されます。

デフォルトの `bootparams` テーブルには 2 つの列しかありませんが、列を使用して、次に示す情報を提供します。

表 23-3 `bootparams` テーブル

列	コメント	説明
キー	ホスト名	ディスクレスマシンの正式ホスト名、 <code>hosts</code> テーブルで指定
値	構成	ルートパーティション: マシンのルートパーティションの位置 (サーバー名とパス) スワップパーティション: マシンのスワップパーティションの位置 (サーバー名とパス) ダンプパーティション: マシンのダンプパーティションの位置 (サーバー名とパス) インストールパーティション ドメイン

「入力ファイルのフォーマット」

列はタブ文字で区切ります。バックスラッシュ (`\`) は、1 つのエントリを複数の行で指定するのに使用します。ルート、スワップ、およびダンプの各パーティション用のエントリのフォーマットを次に示します。

```
client-name root=server:path \
swap=server:path \
dump=server:path \
install=server:path \
domain=domainname
```

次に例を示します。

```
buckarooroot=bigriver:/export/root1/buckaroo \
swap=bigriver:/export/swap1/buckaroo \
dump=bigriver:/export/dump/buckaroo \
install=bigriver:/export/install/buckaroo \
domain=sales.doc.com
```

x86 ベースのマシンでは、使用できるパラメータが増えます。詳細は、`bootparams` (4) のマニュアルページを参照してください。

client_info テーブル

client_info テーブルは、それが存在するドメインのサーバー設定を格納するために使われるオプションの内部 NIS+ テーブルです。このテーブルは、nisprefadm コマンドで作成および保管されます。



注意 - nisprefadm だけを使って、このテーブルで作業してください。このテーブルでは、ほかの NIS+ コマンドは使わないでください。

cred テーブル

NIS+ 主体の資格に関する情報を格納するテーブルです。cred テーブルは各ドメインに 1 つ存在し、ドメインに属するクライアントマシンおよびマシンにログインできるクライアントユーザー (つまり、ドメイン中の主体) の資格に関する情報を持っています。cred テーブルはドメインの org_dir サブディレクトリにあります。

注 - cred テーブルはリンクしないでください。org_dir ディレクトリには、それぞれ専用の cred テーブルが必要です。org_dir cred テーブルもリンクしないでください。

cred テーブルには以下の 5 つの列があります。

表 23-4 cred テーブル

NIS+ 主体名	認証タイプ	認証名	公開データ	非公開データ
主体ユーザー名	LOCAL	UID	グループ ID リスト	
主体ユーザー名またはマシン名	DES	Secure RPC ネット名	公開鍵	暗号化非公開鍵

2 番目の「認証タイプ」という列の値により、ほかの 4 つの列の値のタイプが決定されます。

- 「LOCAL」。認証タイプが LOCAL の場合、残りの列には主体ユーザー名、UID、GID が入ります (最後の列は空白)。

- 「DES」。認証タイプが DES の場合、残りの列には主体名、Secure RPC ネット名、公開鍵、暗号化非公開鍵が入ります。これらの鍵はほかの情報と組み合わせられ、DES 資格の暗号化および復号化に使用されます。

資格および cred テーブルの詳細は、[第 12 章](#)を参照してください。

ethers テーブル

ethers テーブルは、インターネット上のマシンの 48 ビット Ethernet アドレスに関する情報を格納しています。次の 3 列があります。

表 23-5 ethers テーブル

列	コメント	説明
アドレス	Ethernet アドレス	マシンの 48 ビット Ethernet アドレス
名前	公式なホスト名	マシンの名前、hosts テーブルで指定
コメント	コメント	エントリに関するコメント (必要に応じて)

Ethernet アドレスの形式は次のとおりです。

n:n:n:n:n:n hostname

ここで *n* は 0 ~ FF の 16 進数で、1 バイトを表します。このアドレスのバイト列は常に最上位のバイトから始まるネットワーク順になっています。

group テーブル

group テーブルは、UNIX ユーザーグループについての情報を格納します。group テーブルには 4 つの列があります。

表 23-6 group テーブル

列	説明
名前	グループ名
パスワード	グループのパスワード

表 23-6 group テーブル (続き)

列	説明
グループ ID	グループの数値 ID
メンバー	グループメンバー名、コンマで区切る

Solaris の以前のリリースでは、ローカル `/etc/group` ファイルで `+/-` 構文を使用して、NIS グループマップ内のエントリを結合または上書きしました。Solaris システムではネームサービススイッチファイルを使用してマシンの情報ソースを指定するため、この仕様は不要になります。Solaris 2.x システムで必要なのは、クライアントの `/etc/nsswitch.conf` ファイルを編集して `files` を指定し、そのあとグループ情報のソースとして `nisplus` を指定することです。これによって、`group` テーブルの内容がクライアントの `/etc/group` ファイルの内容に効果的に追加されます。

hosts テーブル

`hosts` テーブルは、ドメイン内の全マシン名とそれらの IP アドレスを関連付けます。マシンは、通常は NIS+ クライアントでもあります。必ずしもその必要はありません。`bootparams`、`group`、および `netgroup` などのほかのテーブルは、このテーブルに収められたネットワーク名に依存しています。これらのテーブルは、ネットワーク名を使用して、ホームディレクトリやグループメンバー情報などのほかの属性を個々のマシンに割り当てます。`hosts` テーブルには 4 つの列があります。

表 23-7 hosts テーブル

列	説明
アドレス	マシンの IP アドレス (ネットワーク番号とマシン ID 番号)
正式名	マシンの正式名
名前	マシンを識別するために、ホスト名の代わりに使われる任意の名前
コメント	エントリに関するコメント (必要に応じて)

mail_aliases テーブル

`mail_aliases` テーブルは、`sendmail` によって認識されるドメインのメール別名を含んでいます。これには 4 つの列があります。

表 23-8 mail_aliases テーブル

列	説明
別名	別名の名前
メンバーリスト	この別名に送信されたメールを受信するメンバーが収められているリスト。メンバーはユーザー、マシン、またはほかの別名とすることができる
コメント	エントリに関するコメント (必要に応じて)
オプション	(マニュアルページを参照)

「入力ファイルのフォーマット」

各エントリのフォーマットは次のとおりです。

```
alias-name:member[, member]...
```

1 つのエントリが複数の行にまたがるときには、バックスラッシュを使用します。

netgroup テーブル

netgroup テーブルは、リモートマウント、ログインとシェルのアクセス権をチェックするために使用される、ネットワーク全域にわたるグループを定義します。リモートマウントに使用されるネットグループのメンバーは、マシンです。リモートログインとシェルの場合、これらのメンバーはユーザーです。

注 - サービスを提供している NIS+ サーバーが互換モードで動作している場合、クライアントマシンのユーザーは netgroup テーブルに対して ypcat を実行できません。実行すると、エントリの有無に関わらず「テーブルが空である」という結果が出ます。

netgroup テーブルには 6 つの列があります。

表 23-9 netgroup テーブル

列	コメント	説明
名前	グループ名	ネットワークグループ名
グループ	グループ名	このグループを構成するグループ名

表 23-9 netgroup テーブル (続き)

列	コメント	説明
ホスト	ホスト名	ホスト名
ユーザー	ユーザー名	ユーザーのログイン名
ドメイン	ドメイン名	ドメイン名
コメント	コメント	エントリに関するコメント (必要に応じて)

「入力ファイルのフォーマット」

入力ファイルは、1 つのグループ名と任意の数のメンバーから構成されます。

```
groupname member-list ...
```

メンバー指定は、ほかのネットグループ名、または 3 つのフィールドを順序正しく並べたものとして行うことができます。ネットグループ名、3 つのフィールドを両方指定することもできます。

```
member-list ::= groupname | (hostname, username, domainname)
```

最初のフィールドには、グループに属するマシン名を指定します。2 番目のフィールドに、グループに属するユーザー名を指定します。3 番目のフィールドには、メンバー指定が有効となっているドメインを指定します。

指定のないフィールドは、ワイルドカードを示します。たとえば、次の netgroup には、すべてのドメインのユーザーとすべてのマシンが含まれます。

```
everybody ( , , )
```

フィールド内のダッシュは、ワイルドカードの反対で、このグループに所属するマシンやユーザーがないことを示します。次に 2 つの例を示します。

```
(host1, -, doc.com.) (-, joe, doc.com.)
```

最初の指定では、1 台のマシン host1 を doc.com. ドメインに入れますが、すべてのユーザーを除外します。2 番目の指定では、1 人のユーザーを doc.com. ドメインに入れますが、すべてのマシンを除外します。

netmasks テーブル

netmasks テーブルには、標準のインターネットサブネットに使われるネットワークマスクが収められています。このテーブルには 3 つの列があります。

表 23-10 netmasks テーブル

列	説明
アドレス	ネットワークの IP 番号
マスク	ネットワーク上で使うネットワークマスク
コメント	エントリに関するコメント (必要に応じて)

ネットワーク番号に関しては、マシンアドレスで使用される通常の IP ドット表記を使用できますが、マシンアドレスの代わりにゼロを残します。たとえば、

```
172.31.0.0 255.255.255.0
```

というエントリでは、クラス B ネットワーク 172.31.0.0 が、サブネットフィールドに 24 ビット、ホストフィールドに 8 ビットを持つことを意味します。

networks テーブル

networks テーブルにはインターネットのネットワークが含まれます。このテーブルは、Network Information Control Center (NIC) で管理される公式のネットワークテーブルから作成されるのが普通ですが、これにローカルネットワークを追加しなければならないこともあります。これには 4 つの列があります。

表 23-11 networks テーブル

列	説明
正式名	ネットワークの正式名、インターネットが提供
アドレス	ネットワークの公式 IP 番号
名前	ネットワークの非公式名
コメント	エントリに関するコメント (必要に応じて)

passwd テーブル

passwd テーブルには、ドメイン内のユーザーのアカウントに関する情報が入っています。これらのユーザーは、一般的には NIS+ 主体ですが、必ずしもその必要はありません。ただし、ユーザーが NIS+ 主体である場合、それらの資格はここには収められず、ドメインの cred テーブルに格納されることに注意してください。passwd テーブルは、その他 (または未認証) に対して通常読み取り権を与えます。

注 - スーパーユーザー (ユーザー ID = 0) のエントリをこのテーブルに入れることはできません。ルートのパスワード情報は、/etc ディレクトリ上のファイルに格納してください。

passwd テーブル内の情報は、ユーザーのアカウントが作成されたときに追加されます。

passwd テーブルには次の列があります。

表 23-12 passwd テーブル

列	説明
名前	ユーザーのログイン名であり、ユーザーのアカウントが作成されたときに割り当てられる。この名前には大文字を含める必要はない。最大 8 文字まで
パスワード	ユーザーの暗号化されたパスワード
UID	ユーザーの ID 番号であり、ユーザーのアカウントが作成されたときに割り当てられる
グループ ID	ユーザーのデフォルトグループの ID 番号
GCOS	ユーザーの実名と、ユーザーがメールメッセージ見出しの「From:」フィールドに入りたい情報。この列が「"&"」の場合、単にユーザーのログイン名を使用する
ホーム	ユーザーのホームディレクトリのパス名
シェル	ユーザーの初期シェルプログラム。デフォルトは Bourne シェル： /usr/bin/sh
シャドウ	(表 23-13 を参照)

passwd テーブルには、さらにシャドウ列があります。この列には、ユーザーアカウントに関して、次に示すような制限情報が格納されています。

表 23-13 passwd テーブルのシャドウ列

項目	説明
最終変更日	1970 年 1 月 1 日からパスワードの最終変更日までの日数
最小値	推奨されるパスワード変更間隔の最小日数
最大値	パスワードが有効な最大日数
警告	ユーザーパスワードが期限切れになる前にユーザーが警告を受ける日数

表 23-13 passwd テーブルのシャドウ列 (続き)

項目	説明
間隔	ユーザーに許される休止日数
期限	ユーザーのアカウントが無効となる絶対日付
未使用	将来のために確保。現在は 0 に設定されている

Solaris の以前のリリースでは、ローカル `/etc/passwd` ファイル内で `+/ -` 構文を使って、NIS パスワードマップ内のエントリを統合または上書きしました。Solaris 2x リリースではネームサービススイッチファイルを使用してマシンの情報ソースを指定するため、この仕様は不要になります。Solaris 2.x システムで必要なのは、クライアントの `/etc/nsswitch.conf` ファイルを編集して `files` を指定し、それに続けて `passwd` 情報のソースとして `nisplus` を指定することです。これによって、`passwd` テーブルの内容が `/etc/passwd` ファイルの内容に効果的に追加されます。

しかし、それでも `+/ -` 方式を使用したい場合は、クライアントの `nsswitch.conf` ファイルを編集します。NIS を使用している場合、`passwd` のソースに `compat` を指定します。NIS+ を使用している場合は、`passwd_compat: nisplus` を追加してください。

protocols テーブル

`protocols` テーブルにはインターネットで使われるプロトコルが含まれます。これには 4 つの列があります。

表 23-14 protocols テーブル

列	説明
正式名	プロトコル名
名前	プロトコルの識別に使われる非公式な別名
番号	プロトコルの番号
コメント	プロトコルに関するコメント

rpc テーブル

rpc テーブルには、RPC プログラム名が含まれます。これには 4 つの列があります。

表 23-15 rpc テーブル

列	説明
正式名	プログラム名
名前	プログラムの起動に使用できる別の名前
番号	プログラムの番号
コメント	RPC プログラムに関するコメント

rpc テーブルの入力ファイルの例を次に示します。

```
#
# rpc file
#
rpcbind      00000      portmap      sunrpc      portmapper
rusersd      100002      rusers
nfs          100003      nfsprog
mountd       100005      mount        showmount
walld        100008      rwall        shutdown
sprayd       100012      spray
llockmgr     100020
nlockmgr     100021
status       100024
bootparam    100026
keyserver    100029      keyserver
nisd         100300      rpc.nisd
#
```

services テーブル

services テーブルは、インターネット上で使用できるインターネットサービスに関する情報を格納しています。これには 5 つの列があります。

表 23-16 services テーブル

列	説明
正式名	サービスの公式インターネット名
名前	サービスを要求できる代替名のリスト
プロトコル	サービスを提供するためのプロトコル (たとえば 512/tcp)
ポート	ポート番号
コメント	サービスに関するコメント

timezone テーブル

timezone テーブルは、ドメイン内の全マシンのデフォルトの時間帯を含んでいます。デフォルトの時間帯はインストール中に使用されますが、インストーラがこれを無効にすることができます。このテーブルには 3 つの列があります。

表 23-17 timezone テーブル

列	説明
名前	ドメイン名
時間帯	時間帯の名前 (たとえば US/Pacific)
コメント	時間帯に関するコメント

その他のデフォルトのテーブル

以下のテーブルも、デフォルトで用意されています。

- audit_user
- auth_attr
- exec_attr
- prof_attr
- user_attr

詳細については、『man pages section 4』を参照してください。

第 24 章

NIS+ の問題解決

この章では、問題をいくつかの種類に分類しています。各問題について、一般的な症状を示し問題について説明してから、推奨する対策を 1 つまたは複数挙げています。

また、付録 A では、NIS+ の詳細な共通エラーメッセージをアルファベット順に掲載しています。

注 - NIS+ は、将来のリリースではサポートされなくなる可能性があります。NIS+ から LDAP への移行支援ツールは、Solaris 9 リリース以降で使用できます (『Solaris のシステム管理 (ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』を参照)。詳細については、<http://www.sun.com/directory/nisplus/transition.html> を参照してください。

NIS+ のデバッグオプション

NIS_OPTIONS の環境変数を設定して、NIS+ デバッグオプションを制御できます。

オプションは、二重引用符で囲まれたオプションセットとともにスペースで区切られた NIS_OPTIONS コマンドの後に指定されます。各オプションに *name=value* のフォーマットがあります。値は、特定のオプションに従って整数、文字列、ファイル名になります。整数値のオプションに対して、値が指定されていない場合には、デフォルト値は 1 になります。

NIS_OPTIONS は次のオプションを認識します。

表 24-1 NIS_OPTIONS オプションと値

オプション	値	アクション
debug_file	<i>filename</i>	デバッグの指定ファイルへの出力を指示する。このオプションが指定されていない場合は、デバッグは stdout に出力する
debug_bind	<i>Number</i>	サーバーの選択プロセスに関する情報を表示する
debug_rpc	1 または 2	値が 1 の場合には、NIS+ に対する RPC コールを表示する。値が 2 の場合には、RPC コールと RPC の内容、引数、結果の両方を表示する
debug_calls	<i>Number</i>	NIS+ API へのコールと、アプリケーションに戻される結果を表示する
pref_srvr	<i>String</i>	nisprefadm コマンドによって生成されるのと同じフォーマットで、優先サーバーを指定する (表 20-1 を参照)。これは nis_cachemgr で指定された優先サーバーを上書きする
server	<i>String</i>	特定のサーバーを設定する
pref_type	<i>String</i>	現在実装されていない

例を以下に示します (C シェルの使用を前提)。

- デバッグメッセージを表示するには、以下を入力します。

```
setenv NIS_OPTIONS "debug_calls=2 debug_bind debug_rpc"
```

- API コールの簡単なリストを得て、それをファイル /tmp/CALLS に格納するには、以下を入力します。

```
setenv NIS_OPTIONS "debug_calls debug_file=/tmp/CALLS"
```

- 特定のサーバーに送られた API コールの簡単なリストを得るには、以下を入力します。

```
setenv NIS_OPTIONS "debug_calls server=sirius"
```

NIS+ の管理上の問題

この節では、日常的な NIS+ 名前空間の管理作業を行なっているときに発生する可能性のある問題について説明します。一般的な症状には次のようなものがあります。

- 操作時に表示される「Illegal object type」のメッセージ
- その他の「オブジェクトの問題」のエラーメッセージ
- 初期設定のエラー
- チェックポイントのエラー
- ユーザーをグループに追加するときのエラー
- ログが大きすぎる場合、ディスク容量が不足した場合、ログの切り捨てのエラー
- groups_dir や org_dir が削除できない

無効なオブジェクトの問題

「症状」

- 操作時に表示される「Illegal object type」のメッセージ
- その他の「オブジェクトの問題」のエラーメッセージ

このエラーメッセージには、いくつかの原因が考えられます。

- 検索可能な列がない状態で、テーブルを作成しようとした。
- データベース処理が DB_BADOBJECT の状態を返しました (db(3N) のエラーコードの詳細は、nis_db(3N) のマニュアルページを参照してください)。
- 長さを 0 に指定して、データベースオブジェクトを追加または変更しようとした。
- 所有者を指定せずに、オブジェクトを追加しようとした。
- その処理はディレクトリオブジェクトを想定していますが、指定したオブジェクトは、ディレクトリオブジェクトではありませんでした。
- ディレクトリを LINK オブジェクトにリンクしようとした。
- テーブルエントリにリンクしようとした。
- グループオブジェクトではないオブジェクトを、nisgrpadm コマンドに渡しました。
- グループオブジェクトの処理が想定されていましたが、指定したオブジェクトのタイプはグループオブジェクトではありませんでした。
- テーブルオブジェクトの処理が想定されていましたが、指定したオブジェクトはテーブルオブジェクトではありませんでした。

nisinit のエラー

次の点をチェックしてください。

- NIS+ が動作していることを、ping によって確認できるか
- -H オプションで指定した NIS+ サーバーが、適切なサーバーであるか、また現在も動作しているか
- rpc.nisd がサーバー上で動作しているか
- その他クラスが、このドメインの読み取り権を持っているか
- このマシンで、ネットマスクが正しく設定されているか

チェックポイントのエラーが続く

チェックポイント処理 (たとえば、nisping -C コマンドによる操作) が、継続してエラーを起こしている場合は、十分なスワップ空間やディスク容量があるかどうか確認してください。syslog 内のエラーメッセージもチェックします。core ファイルがディスク空間を使い果たしていないかどうかチェックします。

ユーザーをグループに追加することができない

ユーザーをそのドメイン内のグループのメンバーとして追加する前に、そのユーザーは、ドメインの cred テーブルの中で LOCAL の資格を持つ NIS+ 主体クライアントになっていなければなりません。DES の資格を持っているだけでは、十分ではありません。

ログが大きくなりすぎた

nisping -C を使用して定期的にチェックポイントを実行していないと、ログファイルが極端に大きくなる場合があります。マスターサーバーのすべての複製サーバーが更新されるまでは、マスター上にあるログはクリアされません。複製がダウンしている場合や、サービスが行われていない場合、複製サーバーと通信できない場合は、その複製サーバーに対応するマスターをクリアできません。このため複製がダウンしているか、または一定時間使用できない場合には、[336 ページの「ディレクトリを削除する」](#)で説明したとおり、マスターから複製を削除する必要があります。ディレクトリ org_dir とサブディレクトリ groups_dir を最初に削除してから、ディレクトリ自体を削除してください。

ディスク容量の不足

十分なディスク容量が確保できない場合は、様々なエラーメッセージが表示されます(詳細は、[461 ページの「ディスク容量の不足」](#)を参照)。

トランザクションログファイルを切り捨てることができない

まずはじめに、問題のファイルが存在するか、読み込み可能か、そのファイルに書き込み権が割り当てられているかどうかチェックしてください。

- トランザクションログは、ls /var/nis/trans.log で表示できます。
- ファイルの存在、アクセス権、読み取り権については、nislsl -l と niscat で確認できます。
- 関係するメッセージは、syslog でチェックできます。

もっとも可能性のある原因は、適切なアクセス権を割り当てられているものの、ディスク容量が不足していることです。チェックポイント処理では、ログを切り捨て一時ファイルを削除する前に、まずログ一時ファイルのコピーを作成します。このため、一時ファイルを作成するための十分なディスク容量がない場合は、チェックポイント処理を進めることができません。使用可能なディスク容量をチェックし、必要であれば容量を確保してください。

ドメイン名の混同

NIS+ の多くのコマンドや操作にとって、ドメイン名は重要な役割を果たします。ルートサーバーを除いて、NIS+ のすべてのマスターと複製は、それ自身がサービスを提供するドメインより上にあるドメインのクライアントであるということに特に注意してください。サーバーまたは複製サーバーを、それ自身がサービスを提供するドメインのクライアントとして誤って扱った場合、「Generic system error」や「Possible loop detected in namespace *directoryname: domainname*」というエラーメッセージが表示されます。

たとえば、altair というマシンが、subdoc.doc.com. ドメインのクライアントだとします。サブドメイン subdoc.doc.com. のマスターサーバーが sirius というマシンだとすると、sirius は doc.com. ドメインのクライアントになります。したがって、ドメインの指定や変更を行うときは、混同しないように次のルールに注意してください。

1. クライアントマシンは、特定のドメインかサブドメインに所属します。
2. 特定のサブドメインにサービスを提供するサーバーや複製サーバーは、そのドメインより上にあるサブドメインのクライアントです。
3. 2 の規則の唯一の例外は、ルートマスターサーバーとルート複製サーバーです。これらは、それ自身がサービスを提供するドメインのクライアントになります。つまり、ルートドメインのクライアントになるのは、ルートマスターとルート複製だけです。

したがって、上の例では、マシン altair の完全指定名は alladin.subdoc.doc.com. です。マシン sirius の完全指定名は sirius.doc.com. です。sirius.subdoc.doc.com. という名前は正しくありません。sirius は doc.com. のクライアントで、subdoc.doc.com. のクライアントではないため、エラーの原因になります。

org_dir や groups_dir を削除できない

親ディレクトリを削除する前に、必ず org_dir と groups_dir を削除してください。ドメインの groups_dir と org_dir を削除する前に、nisrmdir を使用してドメインを削除すると、これらの 2 つのサブディレクトリは、どちらも削除できなくなります。

複製の失敗からの NIS+ ディレクトリの削除または分離

複製サーバーからディレクトリを削除または分離する場合には、最初にディレクトリの org_dir と groups_dir のサブディレクトリを削除してから、ディレクトリ自体を削除します。各サブディレクトリが削除された後に、削除しようとするディレクトリの親ディレクトリで nisping を実行する必要があります (336 ページの「[ディレクトリを削除する](#)」を参照)。

nisping の操作に失敗すると、ディレクトリは完全に削除または分離されません。

この状態が発生したら、次の手順を実行して、修正する必要があります。

1. 複製上の `/var/nis/rep/org_dir` を削除します。
2. `org_dir.domain` が、複製上の `/var/nis/rep/serving_list` に表示されないことを確認してください。
3. `domain` で `nisping` を実行します。
4. マスターサーバーから `nisrmdir -f replica_directory` を実行します。

分離しようとしている複製サーバーがダウンしているか、または通信不能である場合に `nisrmdir -s` コマンドは「Cannot remove replica name :attempt to remove a non-empty table」というエラーメッセージが出されます。

このような場合は、マスターサーバー上で `nisrmdir -f -s replicaname` コマンドを実行すれば、強制的に切り離すことができます。ただし、`nisrmdir -f -s` コマンドでダウン状態の複製サーバーを切り離した場合、その複製サーバーがオンラインに復帰したらただちに `nisrmdir -f -s` コマンドを再実行して複製サーバーの `/var/nis` ファイルシステムの中を整理する必要があります。`nisrmdir -f -s replicaname` を再実行しないと、複製がサービスを再開した時に複製上に残された古い情報によって問題が発生する場合があります。

NIS+ データベースの問題

この節では、名前空間のデータベースとテーブルに関連する問題を説明します。一般的な症状には次のようなものがあります。処理内容に関係する次の表現が含まれているエラーメッセージが表示されます。

- Abort_transaction:Internal database error
- Abort_transaction:Internal Error, log entry corrupt
- Callback:select failed
- CALLBACK_SVC:bad argument

`rpc.nisd` が失敗します。

[444 ページの「NIS+ の所有権とアクセス権の問題」](#) も参照してください。

`rpc.nisd` の複数の親プロセス

「症状」

次の表現が含まれている様々なエラーメッセージが表示されます。これらのメッセージは、データベースやトランザクションログが壊れていることを意味しています。

- Log corrupted
- Log entry corrupt
- Corrupt database
- Database corrupted

「考えられる原因」

複数の独立した `rpc.nisd` デーモンを実行させています。通常の動作では、`rpc.nisd` はほかの `rpc.nisd` デーモンを子プロセスとして生成できます。このこと自体は問題はありません。しかし、1 台のマシン上で 2 つの `rpc.nisd` デーモンが親として動作している場合は、互いのデータを上書きし、ログとデータベースを壊してしまいます (通常、このような現象は `rpc.nisd` が手作業で起動された場合にのみ発生します)。

「診断」

`ps -e | grep rpc.nisd` を実行します。親の `rpc.nisd` プロセスは、1 つしか動作していないことを確認します。

「対策」

「親」としての `rpc.nisd` エントリが複数ある場合は、1 つを残して、ほかのデーモンの動作を終了させなければなりません。`svcadm disable` を使用して不要なサービスを停止してから、`ps` コマンドをもう一度実行して、不要なデーモンが終了したかどうかを確認します。

NIS+ データベースが壊れている場合は、壊れていないデータベースを、最新のバックアップから復元します。次に、バックアップの時点よりあとで、名前空間に加えられた変更を反映します。しかし、ログも壊れている場合は、バックアップを行ったあとで名前空間に加えられた変更を、再度手作業で行わなければなりません。

rpc.nisd の失敗

NIS+ テーブルが大きすぎると、`rpc.nisd` は失敗します。

「診断」

`nislsls` を使って、NIS+ テーブルの大きさをチェックします。7K バイト以上のテーブルでは、`rpc.nisd` は失敗します。

「対策」

NIS+ テーブルの大きさを小さくします。ネームサービスとして NIS+ は、オブジェクト自体ではなく、オブジェクトへのリファレンスを格納するために設計されています。

NIS+ と NIS の互換性の問題

この節では、NIS と NIS+ や以前のシステムとの間の互換性に関連する問題、またスイッチ構成ファイルに関連する問題を説明します。一般的な症状には次のようなものがあります。

- `nsswitch.conf` ファイルが正しく実行されない

- 処理内容に関する表現を含むエラーメッセージが表示される

処理内容に関する次の表現が含まれているエラーメッセージが表示される

- Unknown user
- Permission denied
- Invalid principal name

ユーザーがパスワードを変更したあと、ログインできない

「症状」

新しいユーザーや、最近パスワードを変更したユーザーが、ログインできません。または、特定のマシンからログインできますが、ほかのマシンからログインできません。そのようなユーザーに対して、次の表現が含まれているエラーメッセージが表示されることがあります。

- Unknown user *username*
- Permission denied
- Invalid principal name

「最初に考えられる原因」

NIS マシン上でパスワードが変更されました。

NIS+ の名前空間サーバーがサービスを提供しているドメインの中で、NIS を実行している Solaris システムマシン上で、ユーザーまたはシステム管理者が `passwd` コマンドを使用してパスワードを変更した場合、そのユーザーのパスワードは、そのマシンの `/etc/passwd` ファイルの中だけで変更されています。そのユーザーが、ネットワーク上のほかのマシンを使用してログインしても、そのマシン上では新しいパスワードは認識されません。NIS+ のパスワードテーブルに格納されている、古いパスワードを使用しなければなりません。

「診断」

そのユーザーの古いパスワードが、NIS+ のほかのマシンでまだ有効かどうかチェックしてください。

「対策」

NIS+ を実行しているマシンで、`passwd` コマンドを使用して、ユーザーのパスワードを変更します。

「2 番目に考えられる原因」

パスワードの変更を行っても、システム全体に反映されるまでに時間がかかります。

「診断」

名前空間の変更がドメインやシステム全体に反映されるまでに、ある程度の時間がかかります。ドメインのサイズや複製サーバーの台数により、数秒間で済むこともあれば、何十分もかかることがあります。

「対策」

変更結果がドメインに伝わるまで、常識的に受け入れられる程度の時間、待つだけです。または、`nisping org_dir` コマンドを使用して、システムを同期させることもできます。

nsswitch.conf ファイルが正しく実行されない

変更した (または、新しくインストールした) `nsswitch.conf` ファイルが正しく実行されません。

「症状」

新しい `nsswitch.conf` ファイルをインストールしたか、または既存のファイルを変更しましたが、システムがその変更結果を反映していません。

「考えられる原因」

`nsswitch.conf` ファイルのインストールや変更を行なった後は、必ずマシンを再起動して、変更結果を有効にしなければなりません。`nscd` が `nsswitch.conf` ファイルをキャッシュに書き込むためです。

「対策」

`nsswitch.conf(4)` のマニュアルページの情報を参照して、現在の `nsswitch.conf` ファイルをチェックしてください。必要に応じてファイルを修正し、マシンを再起動します。

NIS+ オブジェクトが見つからない問題

この節では、NIS+ がオブジェクトや主体を見つけることができない問題について説明します。一般的な症状には次のようなものがあります。

処理内容に関係する次の表現が含まれているエラーメッセージが表示されます。

- Not found
- Not exist
- Can't find suitable transport for *name*
- Cannot find
- Unable to find
- Unable to stat

構文やスペリングの誤り

NIS+ のオブジェクトが見つからない場合、もっとも可能性のある原因は、名前の入力を間違えたことです。構文をチェックし、正しい名前を使用しているかどうか確認してください。

正しくないパス名

「オブジェクト」が見つからない場合、考えられる原因として、正しくないパスを指定していることが挙げられます。指定したパスが正しいことを確認してください。また、NIS_PATH 環境変数が正しく設定されているかどうか確認してください。

ドメインレベルが正しく指定されていない

すべてのサーバーは、それ自身がサービスを提供しているドメインではなく、その上にあるドメインのクライアントであることに注意してください。この規則には、2つの例外があります。

- ルートマスターサーバーとルート複製サーバーは、ルートドメインのクライアントです。
- NIS+ のドメイン名は、ピリオドで終わります。完全指定名を使用する場合は、ドメイン名の終わりにピリオドをつけなければなりません。ドメイン名の終わりにピリオドをつけないと、NIS+ は、それが部分指定名であると想定します。この規則の例外としては、/etc/defaultdomain ファイルの中では、マシンのドメイン名の終わりにピリオドをつけません。/etc/defaultdomain ファイルの中で、マシンのドメイン名にピリオドをつけると、起動時に「Could not bind to server serving domain name」というエラーメッセージが表示され、ネットワークへの接続に問題が生じます。

オブジェクトが存在しない

NIS+ のオブジェクトが削除されたため、または作成されていないために、現在は存在してなくて、見つからないこともあります。nisls -l を使用して、そのドメインに目的のオブジェクトが存在するかどうかチェックしてください。

複製サーバーの同期遅延

NIS+ のオブジェクトの作成や変更を行うと、処理が完了して、特定の複製サーバーの情報が更新されるまでに、ある程度の遅れが発生します。通常の動作では、マスターからその複製から名前空間情報の照会が行われます。クライアントは、照会を複数のサーバー (マスターと複製) に自動的に分散し、システムの負荷のバランスを取ります。これは、どの時点でも、名前空間の情報をどのマシンが返してくるかわからないということを意味します。新しく作成されたオブジェクトや変更されたオブジェクト

に関係するコマンドが、更新された情報をまだマスターから受け取っていない複製に送られた場合、「オブジェクトが見つかりません」というタイプのエラーか、同期していない古い情報を受け取ることになります。同様に、`nislsls` のような、全体に関係するコマンドを使用して、まだ更新されていない複製サーバーに照会を行なった場合、新しく作成されたオブジェクトが含まれていないリストを受け取る可能性があります。

`nisping` を使用すると、同期していない状態にある複製サーバーを同期させることができます。

代わりに、NIS+ のコマンドで `-M` オプションを指定して、そのコマンドがドメインのマスターサーバーから名前空間の情報を受け取るように指定することも可能です。この方法を使用すると、確実に最新の情報を取得し、使用できます。ただし、`-M` オプションは、必要なときにだけ使用してください。複製サーバーを使用して名前空間のサービスを行う最大の理由は、負荷を分散して、ネットワークの効率を向上させることにあります。

ファイルが見つからないか壊れている

`/var/nis/data` ディレクトリの中の1つまたは複数のファイルが、壊れているか削除されています。最新のバックアップから、これらのファイルを復元してください。

旧バージョンの `/var/nis` について

Solaris 2.4 以前では、`/var/nis` ディレクトリに `hostname.dict`、`hostname.log` という2つのファイルが含まれていました。また、`/var/nis/hostname` というサブディレクトリも含まれていました。Solaris 2.5 においては、2つのディレクトリ名は `trans.log`、`data.dict`、サブディレクトリ名は `/var/nis/data` となります。

`nisinit` など NIS+ 設定プロシージャによって作成された `/var/nis`、`/var/nis/data` といったディレクトリ、およびその下のファイルは、名前を変更しないでください。

Solaris 2.5 ではこれらのファイルの内容も変更されており、Solaris 2.4 以前との互換性はなくなっています。したがって、これらのファイルやディレクトリを Solaris 2.4 での名前にしてしまうと、Solaris 2.4、および 2.5 以降の `rpc.nisd` で機能しなくなりますので名前の変更をしないようにしてください。ディレクトリ名もファイル名も変更しないでください。

名前の中の空白

「症状」

ある時にはオブジェクトが存在し、別の時には存在しないことがあります。NIS+ や UNIX の特定のコマンドが NIS+ のオブジェクトが存在しない、または見つからないと報告しますが、別のコマンドは同じオブジェクトを見つけることができます。

「診断」

`nisls` を使用して、オブジェクト名を探します。オブジェクト名を注意深くチェックして、その名前が実際は空白で始まっているかどうか確認してください。NIS+ のコマンド行を使用して NIS+ のオブジェクトを作成するときに、フラグの後に誤って空白文字を 2 つ入力すると、NIS+ のコマンドの中には、2 番目の空白文字をオブジェクト名の一部と解釈するものがあります。

「対策」

NIS+ のオブジェクト名が空白で始まっている場合は、名前を変更して空白を取り除くか、いったん削除して初めから作成し直します。

オートマウントを使用できない問題

「症状」

ほかのホスト上のディレクトリに移動できません。

「考えられる原因」

NIS+ 環境では、NIS+ の必要条件に合わせて、オートマウントの名前を変更しなければなりません。`/etc/auto*` テーブル名の一部としてピリオドが使用されている場合、NIS+ はそれらのテーブルをアクセスすることができません。たとえば、NIS+ は `auto.direct` というファイルにアクセスすることができません。

「診断」

`nisls` と `niscat` を使用して、オートマウントのテーブル名が正しく割り当てられているかどうか確認します。

「対策」

ピリオドを下線 (`_`) に変更します。たとえば、`auto.direct` を `auto_direct` という名前に変更します。これらのテーブルを参照している可能性のあるほかのマップも変更してください。

テーブルエントリ間でのリンクが機能しない

`nisln` コマンド (またはその他のコマンド) を使って、テーブルエントリ間でのリンクは作成できません。NIS+ コマンドはエントリレベルでのリンクは追跡しません。

NIS+ の所有権とアクセス権の問題

この節では、ユーザーの所有権とアクセス権に関連する問題を説明します。一般的な症状には次のようなものがあります。

処理内容に関係する次の表現が含まれているエラーメッセージが表示されます。

- Unable to stat name
- Unable to stat NIS+ directory name
- Security exception on LOCAL system
- Unable to make request
- Insufficient permission to ...
- You name do not have secure RPC credentials

一般的に観察されるその他の現象

- ユーザーまたはスーパーユーザーが、名前空間に関する作業を行うことができない

アクセス権がない

アクセス権に関連してもっとも頻繁に発生する問題は、もっとも単純な問題です。行おうとしている業務に必要なアクセス権が、割り当てられていません。対象としているオブジェクトを指定して `niscat -o` を使用し、どのアクセス権が割り当てられているか確認します。ほかのアクセス権も必要な場合は、ユーザー自身、オブジェクトの所有者、システム管理者のうちのどれかが、そのオブジェクトのアクセス権要件の変更を行う (詳細は、[第 15 章](#)を参照) か、必要なアクセス権を持つグループに自分自身を追加 (詳細は、[第 17 章](#)を参照) することができます。

資格がない

ユーザーやマシンに適切な資格がない場合は、ほとんどの操作を行なったときにエラーが発生します。ホームドメインの `cred` テーブルを対象にして `nismatch` を使用し、正しい資格を割り当てられているかどうか確認します (資格に関連した問題の詳細については、[452 ページの「無効になった資格」](#)を参照)。

サーバーがセキュリティレベル 0 で動作している

セキュリティレベル 0 で動作しているサーバーは、NIS+ の主体の資格の作成や管理を行いません。

セキュリティレベル 0 で動作しているサーバーで `nispasswd` を試みると、次のエラーメッセージが表示されます。「*You name do not have secure RPC credentials in NIS+ domain domainname*」。

セキュリティレベル 0 は、管理者が名前空間の初期設定やテストを行う際にだけ使用されます。一般のユーザーがアクセスするような環境で使用すべきではありません。

ユーザーのログインがマシン名と同じ

マシン名と同じものを、ユーザーのログイン ID とすることはできません。ユーザー名と同じ名前をマシンに割り当てる (またはその逆) と、最初の主体は、セキュリティに関係するアクセス権を必要とする動作を行うことができなくなります。2 番目の主体の鍵が、cred テーブルの中にある、最初の主体の鍵を上書きするからです。さらに、2 番目の主体は、最初の主体に割り当てられていたアクセス権を持つようになります。

たとえば、saladin というログイン名を持つユーザーが、名前空間の中で読み込み専用のアクセス権を割り当てられていたとします。次に、saladin という名前を持つマシンをドメインに追加します。ユーザー saladin は、何らかの種類のアクセス権を必要とする名前空間の操作を行うことができなくなります。そして、マシン saladin のスーパーユーザーは、名前空間の中で、読み込み専用のアクセス権だけを割り当てられます。

「症状」

- ユーザーやマシンが「permission denied」エラーメッセージを受け取ります。
- ユーザーまたはスーパーユーザーが、keylogin を正しく実行できなくなります。
- 「Security exception on LOCAL system.UNABLE TO MAKE REQUEST.」というエラーメッセージが表示されます。
- 最初の主体が読み込みアクセスを割り当てられていない場合、2 番目の主体が、本来表示できるはずのオブジェクトを見ることができなくなります。

注 - nisclient や nisaddcred を実行したときに、「Adding Key」ではなく「Changing Key」というメッセージが表示された場合は、そのドメインの中で、ユーザー名またはホスト名がすでに重複しています。

「診断」

nismatch を実行して、hosts テーブルや passwd テーブル内のホストとユーザーを表示し、各テーブルの中に、同じホスト名やユーザー名が存在しないか確認します。以下に例を示します。

```
nismatch username passwd.org_dir
```

次に、ドメインの cred テーブルを対象にして nismatch を実行し、重複しているホスト名やユーザー名に、どのようなタイプの資格が割り当てられているかを調べます。LOCAL と DES 両方の資格が割り当てられている場合、cred テーブルのエントリはユーザーを表わしています。DES の資格だけが割り当てられている場合、エントリはマシンを表わしています。

「対策」

マシン名を変更します。ユーザー名を変更するより、マシン名を変更することをお勧めします。次に、cred テーブルからそのマシンのエントリを削除し、nisclient を使用して、マシンを NIS+ のクライアントとして初期設定します。nistbladm を使用して、そのマシンのもとの名前に対する別名を hosts テーブルの中に作成することもできます。必要に応じて、cred テーブル内のユーザーの資格を変更します。

正しくない資格

[452 ページの「無効になった資格」](#)を参照してください。

NIS+ のセキュリティの問題

この節では、パスワード、資格、暗号、その他セキュリティに関係した一般的な問題を取り上げます。

セキュリティ問題の症状

処理内容に関係する次の表現が含まれているエラーメッセージが表示されます。

- Authentication error
- Authentication denied
- Cannot get public key
- Chkey failed
- Insufficient permission to
- Login incorrect
- Keyserve fails to encrypt
- No public key
- Permission denied
- Password [problems]

ユーザーまたはスーパーユーザーは、名前空間に関する作業を行うことができません ([444 ページの「NIS+ の所有権とアクセス権の問題」](#)を参照)。

「Login Incorrect」というメッセージが表示された

原因としてもっとも多いのが、パスワードの誤入力です。もう一度入力するようユーザーに指示してください。「記憶しているパスワードが正しいか」、「パスワードでは大文字と小文字が区別されることを理解しているか」、「アルファベットの o と数字の 0、アルファベットの l と数字の 1 などを混同していないか」といったことについても確認してください。

「login incorrect」メッセージの原因としてはほかに以下のようなものが考えられます。

- パスワードが管理者によってロックされている (306 ページの「パスワードのロック」、307 ページの「パスワードロックの解除」を参照)。
- 指定の日数以上の期間、ログインが行われなかったため、パスワードがロックされた (313 ページの「ログインの間隔の最大値の指定」を参照)。
- パスワードが有効期限を過ぎた (312 ページの「パスワード使用権の有効期限」を参照)。

パスワードについては、第 16 章を参照してください。

パスワードがロック状態、期限切れ、または無効である

「Permission denied, password expired」といったタイプのメッセージは、「ユーザーのパスワードが有効期限を過ぎた」、「またはパスワード使用権が無効になった」といった理由で表示されることがもっとも多くなっています。パスワードについては、第 16 章を参照してください。

- 308 ページの「パスワードの有効期間の設定」を参照
- 312 ページの「パスワード使用権の有効期限」を参照

資格情報が古い

資格、アクセス権ともに正しいにもかかわらず、クライアントの要求が拒否されるという場合もあります。これはおそらく、名前空間のどこかに古い情報が存在することが原因です。

資格情報の保存と更新

公開鍵など、資格に関する情報は、名前空間内の様々な場所に保存されています。NIS+ は、情報を格納しているオブジェクトの生存期間に応じてこの情報を定期的に更新しますが、場合によっては、更新の間に同期が失われ、システムが正常に動作しなくなることがあります。その結果、一部の操作が行えなくなります。表 24-2 は、資格に関する情報を保存するオブジェクト、テーブル、ファイルと、そのリセットの方法を示したものです。

表 24-2 資格に関する情報の保存場所

項目	保存対象	リセットおよび更新の方法
cred テーブル	NIS+ 主体の非公開鍵と公開鍵。これらの鍵のマスターコピーとなる	<code>nisaddcred</code> を使用して新しい資格を作成する。これによって既存の資格が更新される。 <code>chkey</code> を使用しても同様のことが行える

表 24-2 資格に関する情報の保存場所 (続き)

項目	保存対象	リセットおよび更新の方法
ディレクトリオブジェクト	個々のサーバーの公開鍵のコピー	ディレクトリオブジェクトに対して /usr/lib/nis/nisupdkeys コマンドを実行する
キーサーバー	その時点でログインされている NIS+ 主体の非公開鍵	主体ユーザーに対して keylogin を実行する。 あるいは、主体マシンに対して keylogin -r を 実行する
NIS+ デーモン	ディレクトリオブジェクトのコピー (そのサーバーの公開鍵のコピーが含まれる)	NIS+ サービスを無効にすることにより rpc.nisd デーモンとキャッシュマネージャを 停止し、/var/nis から NIS_SHARED_DIRCACHE を削除する。その後、 NIS+ サービスを再起動する
ディレクトリ キャッシュ	ディレクトリオブジェクトのコピー (そのサーバーの公開鍵のコピーが含まれる)	-i オプションを指定して NIS+ キャッシュマ ネージャを再起動する
コールドスタート ファイル	ディレクトリオブジェクトのコピー (そのサーバーの公開鍵のコピーが含まれる)	NIS+ サービスを停止する。/var/nis から NIS_COLD_START ファイルと NIS_SHARED_DIRCACHE ファイルを削除する。 NIS+ サービスを再起動する
passwd テーブル	ユーザーのパスワードまたはマシン のスーパーユーザーのパスワード	passwd -r nisplus コマンドを使用する。こ れによって、NIS+ passwd テーブル、cred テーブルの中でパスワードが更新される
passwd ファイル	ユーザーのパスワードまたはマシン のスーパーユーザーのパスワード	passwd -r nisplus コマンドを使用する。 スーパーユーザー、一般ユーザーのどちらでロ グインしてもよい
passwd マップ (NIS)	ユーザーのパスワードまたはマシン のスーパーユーザーのパスワード	passwd -r nisplus を使用する

古くなったキャッシュに保存された鍵の更新

情報が古くなる原因としてもっとも多いのが、サーバーの公開鍵のバージョンが古くなることです。一般に、この問題を解決するには、アクセスするドメインに対して nisupdkeys を実行します (nisupdkeys コマンドの使用の詳細は、[第 12 章](#)を参照)。

鍵の中にはファイルやキャッシュに保存されているものもあるため、nisupdkeys ですべての問題を解決できるわけではありません。鍵を手作業で更新しなければならない場合もあります。この場合は、「サーバーの公開鍵の内容は、公開鍵が作成された後どのように名前空間オブジェクトに伝えられるのか」ということについて理解する必要があります。サーバーの公開鍵の伝播には、一般に以下の 5 つの段階があります。それぞれの詳細について説明します。

1: サーバーの公開鍵が作成される

NIS+ サーバーはまず第一に NIS+ クライアントです。つまり、NIS+ サーバーの公開鍵は、NIS+ クライアントの公開鍵と同様に、`nisaddcred` コマンドによって生成されます。公開鍵はその後、サーバーが実際にサポートするドメインではなく、サーバーのホームドメインの `cred` テーブルに保存されます。

2: 公開鍵の内容がディレクトリオブジェクトに伝えられる

NIS+ ドメインと NIS+ サーバーの設定後は、サーバーとドメインを関係づけることができます。この「関係づけ」は、`nismkdir` コマンドで行います。`nismkdir` コマンドによってサーバーとディレクトリオブジェクトの関係づけが行われる際、サーバーの公開鍵も `cred` テーブルからドメインのディレクトリオブジェクトにコピーされます。たとえば、サーバーが `doc.com.` ルートドメインのクライアントで、`sales.doc.com.` ドメインのマスターサーバーになっているという場合を考えてみましょう。

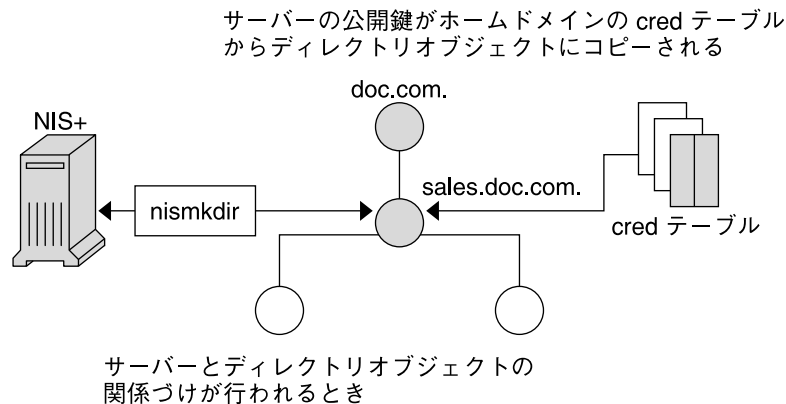


図 24-1 ディレクトリオブジェクトへの公開鍵の伝播

公開鍵は、`cred.org_dir.doc.com.` ドメインから、`sales.doc.com.` ディレクトリオブジェクトにコピーされます。以上のことは、`niscat -o sales.doc.com.` というコマンドを使用して確認できます。

3: ディレクトリオブジェクトの内容がクライアントファイルに伝えられる

`nisinit` ユーティリティまたは `nisclient` スクリプトを使用すれば、すべての NIS+ クライアントを初期化できます。

ほかの類似のコマンドと同様、`nisinit` (または `nisclient`) では、コールドスタートファイル `/var/nis/NIS_COLDSTART` が作成されます。コールドスタートファイルは、クライアントのディレクトリキャッシュ `/var/nis/NIS_SHARED_DIRCACHE` の初期化に使用されます。コールドスタートファイルには、クライアントのドメイン中のディレクトリオブジェクトのコピーが含まれています。ディレクトリオブジェクトには、すでにサーバーの公開鍵のコピーが含まれているため、これで公開鍵の内容はクライアントのコールドスタートファイルに伝えられたことになります。

また、クライアントがホームドメインの外のサーバーに対して要求をした場合、リモートドメインのディレクトリオブジェクトのコピーが、クライアントの NIS_SHARED_DIRCACHE ファイルに保存されます。クライアントのキャッシュの内容は、nisshowcache コマンドを使用して調べることができます。

複製サーバーがドメインに追加されるか、サーバーの鍵が更新されるまでは、鍵の伝播はこの段階にとどまります。

4: 複製サーバーがドメインに追加された場合の処理

複製サーバーがドメインに追加されると、nisping コマンドによって NIS+ テーブル (cred テーブルを含む) が新しい複製サーバーにダウンロードされます。これによって、元のサーバーの公開鍵も複製サーバーの cred テーブルに保存されます。

5: サーバーの公開鍵が更新された場合の処理

サーバーの DES 資格 (サーバーのルート ID) を変更すると、公開鍵も変更されます。その結果、サーバー用に cred テーブルに保存される公開鍵が、以下の場所に保存されるものと矛盾します。

- 複製サーバーの cred テーブル (数分の間)
- サーバーがサポートするドメイン中の、メインディレクトリオブジェクト (生存期間終了まで)
- サーバーがサポートするドメイン中の、クライアントの NIS_COLDSTART ファイルおよび NIS_SHARED_DIRCACHE ファイル (生存期間終了まで、通常 12 時間)
- サーバーがサポートするドメインに要求をしたクライアントの、NIS_SHARED_DIRCACHE ファイル (生存期間終了まで)

サーバーの鍵は、ほとんどの場所において数分 ~ 12 時間で自動的に更新されます。すぐに更新するには、以下のコマンドを使用します。

表 24-3 サーバーの鍵の更新

保存場所	コマンド	参照する項目
複製サーバーの cred テーブル (nisping を使用しなくても、テーブルは数分で自動的に更新される)	nisping	343 ページの「nisping コマンド」
サーバーがサポートするドメインのディレクトリオブジェクト	nisupdkeys	255 ページの「nisupdkeys コマンド」
クライアントの NIS_COLDSTART ファイル	nisinit -c コマンド	339 ページの「nisinit コマンド」
クライアントの NIS_SHARED_DIRCACHE ファイル	nis_cachemgr	341 ページの「nis_cachemgr デモン」

注 -nis_cachemgr の再起動は、既存の nis_cachemgr を停止してから行います。
nis_cachemgr を停止するには、svcadm disable
/network/rpc/nisplus:default を使用して NIS+ サービスを無効にします。

無効になった資格

主体 (ユーザーかマシン) の資格が無効になっている場合は、その主体は nisls のようなコマンドも含め、名前空間の操作や処理を行うことができなくなってしまう。資格が無効になると、未認証クラスに割り当てられるアクセス権も含め、すべてのアクセス権が失われるからです。

「症状」

ユーザーまたはスーパーユーザーが、名前空間に関する作業を行うことができなくなります。名前空間のどのような操作を行っても、「permission denied」というタイプのエラーメッセージが表示されます。ユーザーまたはスーパーユーザーは、nisls を実行することも不可能になります。

「考えられる原因」

鍵の破損、物理的な破損、古い資格、その他何らかの不適切な点が、
/etc/.rootkey ファイルの中にあります。

「診断」

snoop を使用して、不適切な資格を識別します。

または、その主体がリスト表示できる場合は、その主体としてログインを行い、主体が間違いなく承認されているはずのオブジェクトを対象として、NIS+ コマンドを実行します。たとえば、ほとんどの場合、未認証クラスにオブジェクトの読み込みは承認されているはずで、そこで、cred テーブルの中にリストされているすべての主体は、nisls object コマンドを正しく実行できるはずで、このコマンドを実行しても「permission denied」エラーが発生する場合は、おそらく、その主体の資格は無効になっています。

「対策」

- 「一般のユーザー」の場合、keylogout を実行してから、次に keylogin を実行して、その主体のログインを試みます。
- 「root」すなわち「スーパーユーザー」の場合、keylogout -f を実行してから、keylogin -r を実行します。

Keyserv のエラー

keyserv が、セッションを暗号化できません。このタイプの問題には、いくつかの原因が考えられます。

「考えられる原因と対策」

- クライアントが `keylogin` を実行していません。 `keylogin` を実行したかどうか確認します。クライアントが正しく `keylogin` を実行したかどうかを確認めるには、(そのクライアントで) `nisdefaults -v` を実行します。 `Principal Name` という行に「not authenticated」というメッセージが返れば、クライアントが正しく `keylogin` を実行していないことになります。
- クライアント(またはホスト)が、LOCAL や DES のような適切な資格を持っていません。クライアントの `cred` テーブルを対象にして `niscat` を実行し、クライアントが適切な資格を持っているかどうか確認します。必要に応じて、[243 ページの「NIS+ 主体の資格情報を作成する方法」](#)に従って資格を追加します。
- `keyserv` デーモンが動作していません。 `ps` コマンドを使用して、`keyserv` が動作しているかどうか確認します。動作していない場合は、`keyserv` を起動してから `keylogin` を実行します。
- `keyserv` が動作している場合は、Secure RPC や NIS+ のコールを行う終始動作しているはずのプロセスが、まだ動作していません。たとえば、`automountd`、`rpc.nisd`、`sendmail` などが動作していません。これらのプロセスが正しく動作しているかどうか確認します。動作していない場合は再起動します。

マシンが以前は NIS+ のクライアントだった

このマシンが、同じドメインの中で NIS+ のクライアントとして初期設定されている場合は、試みに `keylogin -r` を実行し、Secure RPC パスワードプロンプトでスーパーユーザーのログインパスワードを入力します。

cred テーブルにエントリがない

`cred` テーブルの中に主体(ユーザーまたはホスト)の NIS+ のパスワードが存在することを確認するために、主体のホームドメインの中で次のコマンドを実行します。

```
nisgrep -A cname=principal cred.org_dir.domainname
```

`nisgrep` を別のドメインで実行する場合は、`domainname` には完全な名前を指定する必要があります。

ドメイン名が変更されている

ドメイン名は変更すべきではありません。

既存のドメイン名を変更すると、認証の問題が発生するはずです。ネットワーク全体で、オブジェクトの中に完全指定のドメイン名が埋め込まれているからです。

ドメイン名を変更しないでください。すでにドメイン名を変更してしまい、認証の問題や、ドメイン名に関する「malformed」や「illegal」などの表現が含まれているメッセージが表示されている場合は、ドメイン名をもとの名前に戻します。ドメイン

名を変更したい場合は、次の手順に従ってください。「新しい名前」を使用して「新しいドメイン」を作成し、新しいドメインでマシンをサーバーやクライアントとして設定し、これらが正しく動作していることを確認した上で、古いドメインを削除します。

マシンを新しいドメインに変更する

NIS+ のクライアントとなっているマシンを、ほかのドメインのクライアントに変更したい場合は、`/etc/.rootkey` ファイルを削除し、ネットワーク管理者から受け取ったパスワードか `niscclient` スクリプトから取り出したパスワードを使用して、`nispopulate` スクリプトを実行し直します。

`/etc/passwd` ファイルの中にある NIS+ のパスワードとログインパスワード

NIS+ のパスワードは、NIS+ の `passwd` テーブルの中に格納されています。ログインパスワードは、NIS+ の `passwd` テーブルか、各ユーザーの `/etc/passwd` ファイルの中に格納されています。ユーザーパスワードと NIS+ のパスワードは、同じでも異っていてもかまいません。`/etc/passwd` ファイル内のパスワードを変更するには、`nsswitch.conf` ファイルでの設定を「files」にするか、「-r files」というフラグを指定して `passwd` コマンドを実行する必要があります。

`nsswitch.conf` ファイルは、どの目的にどのファイルを使用するか指定します。`nsswitch.conf` ファイルが、システムの照会に対して誤った場所を指示している場合は、パスワードやアクセス権のエラーが発生するはずですが、

Secure RPC パスワードとログインパスワードが異なる

主体のログインパスワードと Secure RPC パスワードが一致しないと、ログイン時に `keylogin` はパスワードを復号化できません。`keylogin` はデフォルトで主体のログインパスワードを使用することになっていて、また非公開鍵は主体の Secure RPC パスワードを使用して暗号化されているからです。

この場合、主体はシステムにログインすることはできますが、NIS+ においては未認証クラスとして扱われます。キーサーバーが、そのユーザーの非公開鍵を復号化されていない状態のまま持っているからです。NIS+ 環境では多くの場合、未認証クラスによる NIS+ オブジェクトへのアクセス (作成、削除、変更) が拒否されるため、この状態でユーザーが NIS+ オブジェクトにアクセスしようとしても「permission denied」といったタイプのエラーになってしまいます。

注 – 「ネットワークパスワード」と「Secure RPC パスワード」は、このコンテキストでは同義語として扱われる場合があります。ネットワークパスワードの入力を求められたら Secure RPC パスワードを入力します。

別の承認クラスを得るには、この場合 `keylogin` プログラムを実行し、パスワード入力のプロンプトに主体の Secure RPC パスワードを入力する必要があります (250 ページの「NIS+ を使用したキーログイン」を参照)。

しかし `keylogin` を実行しても、現在のログインセッション以外には無効で、一時的な解決にしかありません。キーサーバーはこの方法によって、復号化された形でユーザーの非公開鍵を持つようになるのですが、ユーザーの `cred` テーブル中の非公開鍵が Secure RPC パスワード (ユーザーのログインパスワードとは異なっている) を使用して暗号化されているという点に変わりはないからです。ログインし直してしまえば、状況はまったく元どおりになってしまいます。根本的に問題を解決するためには、`cred` テーブルの非公開鍵を、Secure RPC パスワードではなくログイン ID に基づいたものに変更する必要があります。この作業は、`chkey` プログラムを使用しています (251 ページの「NIS+ 主体の鍵の変更」を参照)。

Secure RPC パスワードとログインパスワードが異なっているという問題を根本的に解決するには、具体的には以下の作業を行います。

1. ログインパスワードを使用してログインします。
2. `keylogin` プログラムを実行して、キーサーバーに保存される非公開鍵を一時的に復号し、一時的な NIS+ アクセス権を得ます。
3. `chkey -p` 実行して、`cred` テーブル中の暗号化された非公開鍵を、ユーザーのログインパスワードに基づいたものに固定的に変更します。

/etc/.rootkey ファイルがすでに存在している

「症状」

「insufficient permission」や、「permission denied」などの、様々なエラーメッセージ

「考えられる原因」

サーバーやクライアントの設定や初期設定を行なったときに、`/etc/.rootkey` ファイルがすでに存在していました。以前そのマシンに NIS+ をインストールしたことがあり、NIS+ を削除したとき、または NIS や `/etc` への変更を行なったときに、`.rootkey` ファイルを削除しなかったためにこのような状態が起こります。

「診断」

`/etc` ディレクトリで `ls -l` と `nisls -l org_dir` を実行し、`/etc/.rootkey` の日付を、`cred` テーブルの日付と比較します。`/etc/.rootkey` の日付が明らかに `cred` テーブルより古い場合は、ファイルがあらかじめ存在していたことが考えられます。

「対策」

問題のあるマシンで、ルートとして `keylogin -r` コマンドを実行し、そのマシンをもう一度クライアントとして設定し直します。

root のパスワードを変更したための問題

「症状」

マシンのルートのパスワードを変更した結果、変更結果が反映されなかったか、スーパーユーザーとしてログインできなくなりました。

「考えられる原因」

注 - セキュリティ上の理由から、`passwd` テーブルの中に、`UserID 0` という項目を、設けるべきではありません。

`root` のパスワードを変更した際、`root` に対して `chkey -p` を実行していなかったり、何らかの問題が発生したことにより、変更が正しく行われませんでした。

「対策」

管理特権を持つユーザー (つまり、管理特権が割り当てられているグループに所属するメンバー) としてログインし、`passwd` を使用して、元のパスワードに戻します。元のパスワードが正しく機能するかどうか確かめてください。正しく機能すれば、`passwd` を使用してパスワードを変更した後、`chkey -p` を実行します。



注意 - NIS+ の名前空間の設定が終わり、すでに動作している状態でも、ルートマスターのマシンを使ってルートのパスワードを変更することは可能です。しかし、ルートマスターの鍵は変更しないでください。これらは、サブドメイン内のすべてのクライアント、複製サーバー、サーバーの中のすべてのディレクトリオブジェクトに埋め込まれているからです。`chkey` をルートで実行する際、必ず `-p` オプションを指定するようにすれば、ルートマスターの鍵を変更する必要はなくなります。

NIS+ の性能の低下とシステムのハングアップの問題

この節では、性能の低下とシステムのハングアップの問題を取り上げます。

性能の低下の症状

処理内容に関係する次の表現が含まれているエラーメッセージが表示されます。

- 「Busy try again later」

- 「Not responding」

次の一般的な現象が観察されることもあります。

- コマンドを実行しても、長時間、何も行われていないように見える
- システムやシェルが、キーボードやマウスの操作に全く反応しない
- NIS+ の動作が通常よりも遅い

チェックポイントの実行

だれかが `nisping` か `nisping -C` どちらかのコマンドを実行しました。または、`rpc.nisd` デーモンが、チェックポイントを実行しています。



注意 – 再起動しないでください。 `nisping` を実行しないでください。

`nisping` やチェックポイントを実行すると、サーバーは反応が遅くなったり、ほかのコマンドにすぐに応答しなくなったりすることがあります。名前空間のサイズに応じて、このようなコマンドが完了するまでに、かなりの時間を要します。これらのコマンドの実行中に、だれかが同様のコマンドを実行すると、所要時間は何倍にもなります。再起動も行わないでください。この種の問題は自然に解決します。サーバーが `nisping` やチェックポイントの実行を完了するまで待つだけです。

マスターサーバーと複製サーバーの同期が完全に行われている場合、同期が完了するまで、複製サーバーはサービスを停止します。再起動しないで待機してください。

変数 NIS_PATH

`NIS_PATH` 変数が、明快かつ単純な値に設定されているかどうか確認します。たとえば、デフォルトでは `org_dir.$:$` に設定されています。複雑な `NIS_PATH`、特にほかの変数を含んでいる変数は、システムのを低下させ、特定の操作にエラーを引き起こす可能性があります (詳細は、78 ページの「環境変数 `NIS_PATH`」を参照)。

デフォルト以外のテーブルパスを指定するために `nistbladm` を使用することは避けてください。デフォルト以外のテーブルパスを指定すると、性能は低下します。

テーブルパス

テーブルパスは使用しないでください。性能を低下させることになります。

複製サーバーが多すぎる

1 つのドメインの中に複製サーバーが多すぎる場合は、複製作業を行なっている間はシステムの性能が低下します。1 つのドメインまたはサブドメインの中に、10 台より多くの複製サーバーを置くことは避けてください。ドメインの中の複製サーバーが 5 台より多い場合は、何台かを取り除いて性能が改善されるかどうか調べます。

再帰的なグループ

再帰的なグループとは、ほかのグループを包含するグループのことです。グループの中にほかのグループを含めると、システム管理者として行う作業は減少しますが、システムの速度は低下します。再帰的なグループを使うべきではありません。

起動時の NIS+ データベースのログが大きい

rpc.nisd は、起動時に各ログを参照します。ログが大きい場合は、起動時の参照に時間がかかることがあります。rpc.nisd を起動する前に、nisping -C を使用して、チェックポイントを実行することをお勧めします。

マスターの rpc.nisd デーモンが終了している

「症状」

-M オプションを指定して要求をマスターサーバーに送ったときに、マスターのマシンで rpc.nisd デーモンが終了していると、「server not responding」タイプのエラーメッセージが発生し、更新を行うことは認められません。-M オプションを指定しなかったときは、要求は機能している複製サーバーに自動的に転送されます。

「考えられる原因」

ホームディレクトリ名やホスト名の一部として大文字を使うと、rpc.nisd デーモンが終了することがあります。

「診断」

最初に、サーバー自体が起動されて、動作しているかどうか確認します。動作している場合は、ps -ef | grep rpc.nisd を実行し、デーモンが動作しているかどうか調べます。

「対策」

デーモンが終了している場合は、svcadm enable /network/rpc/nisplus:default を使用して NIS+ サービスを再起動します。たびたび rpc.nisd デーモンが終了する場合は、ご購入先にご連絡ください。

nis_cachemgr がない

「症状」

あるマシンがほかのドメインにある名前空間オブジェクトを探すのに、非常に長い時間がかかっています。

「考えられる原因」

`nis_cachemgr` を実行していません。

「診断」

`ps -ef | grep nis_cachemgr` を実行して、`nis_cachemgr` が動作しているかどうか確認します。

「対策」

NIS+ サービスを有効にすることにより、そのマシンで `nis_cachemgr` を起動します。

NIS+ のインストール後、サーバーの起動が非常に遅い

「症状」

NIS+ のスクリプトを使用して NIS+ をインストールした後、サーバーの動作が非常に遅く、緩慢です。

「考えられる原因」

`nispopulate` スクリプトを動作させた後、`nisping -C -a` を実行していません。

「対策」

`nisping -C -a` を実行して、できるだけ早くチェックポイントを実行します。

niscat がエラーメッセージ「Server busy.Try Again」を返す

「症状」

`niscat` を実行すると、サーバーがビジーであるというエラーメッセージが表示されます。

「考えられる原因」

- 再同期を実行しているときに重い負荷がかかっている、サーバーがビジーです。
- サーバーがスワップ空間を使い果たしました。

「診断」

`swap -s` を実行して、サーバーのスワップ空間をチェックします。

「対策」

NIS+ を実行するには、適度のスワップ空間とディスク容量を用意すべきです。必要に応じて、空間を増やします。

ホスト名を変更した後で、NIS+ の照会がハングする

「症状」

NIS+ サーバーのホスト名を完全指定することは、推奨されていません。このような指定を行なった後、NIS+ の照会を実行すると、エラーメッセージを表示することなくハングアップが発生しました。次の可能性をチェックします。

「考えられる原因」

完全指定のホスト名は、次の条件を満たしていなければなりません。

- ホスト名のドメイン部は、`domainname` コマンドの結果と完全に一致している
- ホスト名を完全指定名に変更した後、`/etc` や `/etc/inet` 内のファイルに、新しいホスト名の情報を反映させる
- ホスト名の終わりにピリオドをつける

「対策」

NIS+ サービスと、そのホストやサーバーで実行中の関連プロセスをすべてを停止します。上の 2 つの条件に合わせて、ホスト名を変更します。`nisinit` を使用してサーバーを初期設定します。ホスト名が正しいことを確認した後もハングアップが発生する場合は、この節のほかの考えられる原因をチェックしてください。

NIS+ のシステムリソースの問題

この節では、メモリーやディスク容量のようなシステムリソースが不足したときの問題を取り上げます。

リソースの問題

処理内容に関係する次の表現が含まれているエラーメッセージが表示されます。

- No memory
- Out of disk space
- 「Cannot [do something] with log」などのメッセージ
- Unable to fork

メモリーの不足

システムのメモリーやスワップ空間が不足すると、NIS+ の様々な問題やエラーメッセージに遭遇します。一時的な解決策として、不必要なウィンドウやプロセスを終了させることが考えられます。必要に応じて、ウィンドウシステムを終了し、端末のコマンド行を使用します。それでもメモリー不足を知らせるメッセージが表示されるときは、スワップ空間やメモリーを追加するか、十分なスワップ空間やメモリーを持つシステムに切り替えます。

特定の状況では、アプリケーションやプロセスがメモリーを無駄に消費し、使用メモリーサイズが極端に大きくなることがあります。次のコマンドを実行すると、アプリケーションやプロセスの現在のサイズを知ることができます。

```
ps -el
```

sz (サイズ) の列には、各プロセスの現在のメモリーサイズが表示されています。必要に応じて、サイズが同程度と思われるアプリケーションやプロセスを比較し、メモリーサイズが極端に大きいものが存在しないかどうか調べます。

ディスク容量の不足

ディスク容量が不足すると、様々なエラーメッセージが表示されます。ディスク領域の不足に共通する原因は、定期的に tmp ファイルの削除やログファイルの切り捨てをしていないことです。切り捨てを行わないと、ログファイルと tmp ファイルは常時増加します。これらのファイルの増大速度はシステムごとに異なり、同じシステムでも状態によって変化します。非効率的なシステムや、名前空間に問題を抱えているシステムでは、ログファイルは非常に急速に増大します。

注 - 多くの問題が発生しているときは、ログファイルと /tmp のファイルを頻繁にチェックします。ディスク容量が不足してほかの問題が発生する前に、ログファイルを切り捨て、/tmp ファイルを削除します。また、ルートディレクトリとホームディレクトリで core ファイルを探して削除します。

ログファイルを切り捨てるには、システムのチェックポイントを設けます。チェックポイントのプロセスがある程度の時間を要し、完了するまではシステムの速度を低下させることに注意してください。また、ファイルの切り捨てを行う前に完全なコピーを作成するので、ある程度のディスク容量を必要とします。

システムのチェックポイントを実行するには、nisping -c を実行します。

プロセス数の不足

過度に負荷の大きいマシンでは、マシンの構成の中で指定されている、同時に実行できる最大のプロセス数に達する可能性があります。この結果、「unable to fork」のようなメッセージが表示されます。この問題を解決するために推奨されている方法は、不必要なプロセスを終了させることです。それでも問題が持続する場合は、システムの管理マニュアルの説明に従って、より多くのプロセスを扱えるようにシステムを構成し直してください。

NIS+ のユーザーの問題

この節では、一般ユーザーが遭遇する NIS+ の問題を取り上げます。

ユーザーの問題

- ユーザーがログインできない
- ユーザーがほかのドメインにリモートログインできない

ユーザーがログインできない

ユーザーがログインできない原因としては、以下のように様々なものが考えられます。

- パスワードを忘れた新しいパスワードを設定します。別のマシンのユーザーの場合は、`nispasswd` を使用します。この作業は NIS+ 管理者だけが行えます。
- パスワードを間違えて入力したユーザーが、正しいパスワードを覚えているかどうか、また「パスワードでは大文字と小文字が区別される」、「アルファベットの `o`、`l` と数字の `0`、`1` は、まったく別のものである」という点を理解しているかどうかを確認します。
- 「Login incorrect」というタイプのメッセージが表示される。単純に「パスワードの入力を間違えた」という以外の原因については、[447 ページの「「Login Incorrect」というメッセージが表示された」](#)を参照してください。
- ユーザーのパスワード使用権が有効期限を過ぎている ([312 ページの「パスワード使用権の有効期限」](#)を参照)。
- 指定された期間を超えてログインが行われなかった ([313 ページの「ログインの間隔の最大値の指定」](#)を参照)。
- `nsswitch.conf` ファイルの設定が正しくない。`passwd` エントリの設定は以下の 5 つのうちのいずれかにします。
 - `passwd: files`
 - `passwd: files nis`
 - `passwd: files nisplus`
 - `passwd: compat`
 - `passwd: compat passwd_compat: nisplus`

これ以外の設定をすると、ユーザーがログインできなくなります

(詳細は、[296 ページの「nsswitch.conf ファイルの必要条件」](#)を参照)。

ユーザーが新しいパスワードを使ったときにログインできない

「症状」

最近パスワードを変更したユーザーが、ログインできません。または、特定のマシンからログインできますが、ほかのマシンからログインできません。

「考えられる原因」

- パスワードの変更を行なっても、システム全体に反映されるまでに時間がかかります。試しに、古いパスワードを使ってログインしてみてください。
- NIS+ が動作していないマシンで、パスワードを変更しました。

ユーザーがリモートドメインにログインできない

「症状」

ユーザーが `rlogin` を使用して、ほかのドメインへのログインを試みましたが、「Permission denied」メッセージが表示されて、拒絶されました。

「考えられる原因」

ほかのドメインにリモートログイン (`rlogin`) するには、ユーザーはそのドメインで LOCAL の資格を持っているなければなりません。

「診断」

そのドメインで、`nismatch username.domainname.cred.org_dir` を実行し、LOCAL の資格を持っているかどうか調べます。

「対策」

リモートドメインから `nisaddcred` を使用し、そのドメインでの LOCAL の資格をユーザーに割り当てます。

ユーザーがパスワードを変更できない

原因としてもっとも多いのが、古いパスワードの入力を間違えた (または忘れた) ということです。

ほかには以下のような原因が考えられます。

- パスワードの最小値に、最大値よりも大きい値が設定されている (309 ページの「パスワードの変更禁止期間の設定」を参照)。
- パスワードがロックされている、あるいはパスワードの有効期限が過ぎている (447 ページの「「Login Incorrect」というメッセージが表示された」、448 ページの「パスワードがロック状態、期限切れ、または無効である」を参照)。

NIS+ に関するその他の問題

この節では、上記の問題に当てはまらない問題を取り上げます。

NIS+ の動作

特定のホストが NIS+ を実行しているかどうか知りたい場合があります。NIS+ が動作しているか知るには、スクリプトが必要です。

NIS+ が動作しているかどうかを確認するには、次の 2 つの方法があります。

- `svcs \*nisplus\*` コマンドを使用して、NIS+ がオンラインになっているかどうかを確認する
- 次のことを確認する
 - `nis_cachemgr` が動作している
 - ホストに `/var/nis/NIS_COLD_START` ファイルがある
 - `nisls` の実行に成功した

複製サーバーの更新のエラー

「症状」

更新が成功しなかったことを示すエラーメッセージが表示されます。次のメッセージが更新の成功を示すことに注意してください。「`replica_update: number updates number errors`」

「考えられる原因」

次のエラーメッセージのどれかが表示された場合、サーバーがビジーであり、更新を延期すべきことがわかります。

- Master server busy, full dump rescheduled
- `replica_update` error result was Master server busy full dump rescheduled, full dump rescheduled
- `replica_update: master server busy, rescheduling the resync`
- `replica_update: master server busy, will try later`
- `replica_update: nis dump result Master server busy, full dump rescheduled`
- `nis_dump_svc: one replica is already resyncing`

これらのメッセージは、NIS+ のエラーコード定数 `NIS_DUMPLATER` (ある複製がすでに再同期を実行している) によって、または併用することで生成されます。

次のメッセージは、ほかの問題が起こっていることを示します。

- `replica_update: error result was ...`
- `replica_update: nis dump result nis_perror error string`
- `rootreplica_update: update failednis dump result nis_perror string-variable: could not fetch object from master`

-c (診断チャンネルのオープン) オプションを指定した `rpc.nisd` が動作している場合は、マスターサーバーか複製サーバーのシステムログに、詳細な情報が記録されることもあります。

これらのメッセージは、次のような潜在的な問題が起こっていることを示しています。

- サーバーが割り当てることができる子プロセスを使い果たした
- 読み込み専用の子プロセスがダンプを行うよう要求された
- ほかの複製サーバーが、現在同期を実行している

「診断」

複製サーバーとマスターサーバー両方のシステムログから、詳細な情報を探します。情報が記録されている場合でも、詳細の程度は、システムのエラー報告レベルと、`-C` オプション (診断) を指定して `rpc.nisd` を実行したかどうかによって異なります。

「対策」

ほとんどの場合、これらのメッセージは、システムが修正することのできる、ソフトウェアの小さな問題が発生したことを意味しています。あるコマンドを実行した結果、これらのメッセージが表示された場合は、しばらく待って、もう一度同じコマンドを実行します。これらのメッセージが頻繁に表示される場合は、`/etc/syslog.conf` ファイル内のしきい値レベルを変更します。詳細は、`syslog.conf` のマニュアルページを参照してください。

付録 A

エラーメッセージ

この付録では、Solaris ソフトウェアの DNS、NIS、および NIS+ ネームサービスで使われる一般的なエラーメッセージを、アルファベット順に説明します。メッセージごとに説明を行い、該当するものがある場合は、解決策や、このマニュアルのほかの部分の参照箇所を示します。

エラーメッセージについて

この章で説明したエラーメッセージの一部は、マニュアルページでさらに詳しく説明します。

エラーメッセージの内容

エラーメッセージは、ポップアップウィンドウ、シェルツールのコマンド行、ユーザーコンソールウィンドウや、各種のログファイルに表示または記録されます。
/etc/syslog.conf ファイルに指定されているエラーの重大度の基準を上げたり下げたりすることもできます。

ほとんどの場合、入力したコマンドか、コマンドが送られたコンテナオブジェクト(ファイル、マップ、テーブル、またはディレクトリ)によって、エラーメッセージが生成されます。しかし時には、コマンドに応答したサーバーによって、エラーメッセージが生成されることもあります(メッセージは通常、syslog に記録される)。たとえば、「permission denied」メッセージは、ほとんどの場合ユーザーかマシンが原因となって発生しますが、コマンドやマシンから要求された機能を実行するのに必要なアクセス権を、サーバー上のソフトウェアが持っていない場合に発生することもあります。

同様に、一部のコマンドは、非常に多くの種類のオブジェクトに対して検索や照会を行います。オブジェクトによっては、原因が明確であるとは限りません。オブジェクトのアクセス権 (読み取り専用の状態、利用できないなどの状態) が原因で、エラーメッセージが返されることがあります。このような場合は、どのオブジェクトが原因で問題が起こったのか、メッセージからわからない場合もあります。

通常の操作では、ネーミングなどのソフトウェアとサーバーは、関数ルーチン呼び出しを行います。時として、これらの呼び出しが障害を起こし、エラーメッセージが生成されることがあります。また、ユーザーが入力したコマンドをクライアントやサーバーが処理する前に、それ以外のコマンドによる呼び出しが障害を起こし、エラーメッセージが生成されることもあります。そのようなエラーメッセージは、あたかも今入力したコマンドに対する応答のように見えるかもしれませんが、実際はそれまでの操作に対する応答です。

注 - 名前空間で作業を行なっているときに、遠隔手続き呼び出し (RPC) によってエラーメッセージが生成される可能性もあります。これらの RPC エラーメッセージは、このマニュアルでは説明していません。ご使用のシステムのマニュアルを参照してください。

状況によって異なる意味

NIS+ のソフトウェアのどの部分がエラーメッセージを生成したかにより、同じエラーメッセージがやや異なる意味を持つことがあります。たとえば、`nislsls` コマンドが「Not Found」というメッセージを生成した場合、指定された名前の NIS+ のオブジェクトが存在しないことを意味します。しかし、`nismatch` コマンドが同じメッセージを表示した場合、検索基準に一致するテーブル項目が見つからなかったことを意味します。

エラーメッセージのアルファベット順ソート規則

この付録のエラーメッセージは、次の規則に従ってアルファベット順に示します。

- 大文字と小文字を区別しません。つまり、「A」で始まるメッセージと「a」で始まるメッセージは同じ順になります。
- アルファベット以外の記号は無視します。つまり、`_svcauth_des` で始まるメッセージは、文字「S」で始まるほかのメッセージと同じ順になります。
- 「NIS+」で始まる (または「NIS+」を含む) エラーメッセージは、アルファベット順で「NIS」で始まる (または「NIS+」を含む) メッセージの後になります。
- エラーメッセージの前に、エラーメッセージを生成したホスト、アプリケーション、プログラム、ルーチンの日付または名前が付き、その後にコロンが付く場合があります。この場合には、コマンドの最初の名前はメッセージをアルファベット順にするために使われます。

- メッセージの中には、ユーザー ID、プロセス番号、ドメイン名などの変数が含まれています。この章では、これら変数は斜体の文字で示されます。変数は任意のものにできるので、この章に示すメッセージの分類には含まれていません。たとえば、実際には `sales: is not a table` (`sales` は変数) と表示されるメッセージは、この付録では、「`name: is not a table`」と表現しており、「`is not a table`」として文字「I」で始まるメッセージに入っています。
- 「`**ERROR: domainname does not exist`」のように、アスタリスクで始まるエラーメッセージは、NIS+ のインストールスクリプトや設定スクリプトによって生成されたものです。メッセージは、アスタリスクを無視し、最初の文字でソートされます。

エラーメッセージ内の番号

- 多くのメッセージには、IP アドレスが含まれます。IP アドレスは `n.n.n.n` で示されます。
- エラーメッセージにプロセス ID 番号、項目の数などのような番号が含まれる場合があります。エラーメッセージ内の番号は `nnnn` で示されます。

NIS+、FNS に共通するエラーメッセージ

`abort_transaction: Failed to action NIS+ objectname`
`abort_transaction` ルーチンが、サーバーのクラッシュやほかの回復不可能なエラーにより、不完全なトランザクションから抜けることに失敗しました。詳細については、『*Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)*』の「NIS の障害追跡」を参照してください。

`abort_transaction: Internal database error abort_transaction:`
`Internal error, log entry corrupt NIS+ objectname`
 これら 2 つのメッセージは、名前空間のデータベースやログが壊れていることを意味します。詳細については、『*Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)*』の「NIS の障害追跡」を参照してください。

`add_cleanup: Cant allocate more rags.`
 このメッセージは、システムで使用するメモリーが不足していることを示しています。メモリー不足に関する問題については、『*Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)*』の「NIS の障害追跡」を参照してください。

add_pingitem: Couldn't add *directoryname* to pinglist (no memory)
メモリー不足の問題については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

add_update: Attempt add transaction from read only child.
add_update Warning: attempt add transaction from read only child
読み取り専用の子プロセス `rpc.nisd` が、ログに項目を追加しようとしていました。ログの中にときどきこのメッセージが記録されても、重大な問題ではありません。しかし、頻繁に記録される場合は、ご購入先にご連絡ください。

Attempting to free a free rag!
このメッセージは、`rpc.nisd` にソフトウェア上の問題が起こったことを意味します。`rpc.nisd` は、異常終了したはずですが、`ps -ef | grep rpc.nisd` を実行して、`rpc.nisd` が現在も動作しているかどうか確認します。まだ動作している場合は、NIS+ サービスを停止し、前回と同じオプションで再起動します。デーモンが動作していない場合は、前回と同じオプションで NIS+ サービスを起動します。`/var/nis` をチェックして、`core` ファイルのダンプが行われていないか確認します。`core` ファイルが存在する場合は、削除してください。

注 - `-YB` オプションを指定して NIS+ サービスを起動した場合は、`rpc.nisd_reply` デーモンも終了しなければなりません。

Attempt to remove a non-empty table
`nistbladm` が、まだエントリを含んでいる NIS+ テーブルを削除しようとしていました。または、`nisrmdir` が、ファイルあるいはサブディレクトリを含むディレクトリを削除しようとしていました。

- テーブルを削除する場合には、`niscat` を使ってテーブルの内容をチェックし、`nistbladm` を使って既存の内容を削除します。
- ディレクトリを削除する場合には、`nisls -l -R` を使って既存のファイルまたはサブディレクトリをチェックして、それらを最初に削除します。
- `nisrmdir -s` を使って、ドメインから複製を分離するときに、複製がダウンしているか、またはマスターとの通信不能の状態にある場合に、このエラーメッセージが表示されます。このような場合は、マスターサーバー上で `nisrmdir -f -s replicaname` コマンドを実行すれば、強制的に切り離すことができます。しかし、`nisrmdir -f -s replicaname` を使って通信不能な複製を分離する場合には、複製がオンライン状態に戻ったらすぐに、`nisrmdir -f -s replicaname` を再実行して、複製の `/var/nis` ファイルシステムを消去する必要があります。`nisrmdir -f -s replicaname` を再実行しないと、複製がサービスを再開した時に複製上に残された古い情報によって問題が発生する場合があります。

このメッセージは、NIS+ のエラーコード定数 `NIS_NOTEMPTY` によって生成されます。詳細な情報は、`nis_tables(3N)` のマニュアルページを参照してください。

authdes_marshall: DES encryption failure

認証データの DES 暗号化に失敗しました。考えられる原因は次のとおりです。

- ライブラリ関数か引数が壊れています。
- DES 暗号化チップを使用している場合、そのチップに問題があります。

ご購入先にご連絡ください。

authdes_refresh: keyserv is unable to encrypt session key

keyserv プロセスは、公開鍵を使用してセッション鍵の暗号化を行うことができませんでした。詳細については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

authdes_refresh: unable to encrypt conversation key

keyserv プロセスは、公開鍵を使用してセッション鍵の暗号化を行うことができませんでした。このメッセージが表示された場合、通常はユーザーの側で何らかの操作を行う必要があります。考えられる原因は次のとおりです。

- keyserv プロセスが終了しているか、応答しません。ps -ef を使用して、keyserv プロセスが keyserv のホストで動作しているかどうかチェックします。動作していない場合は、keyserv を起動し、keylogin を実行します。
- クライアントが keylogin を実行していません。クライアント側で keylogin を実行し、問題が解決できるかどうか調べます。
- クライアントのホストに、資格が割り当てられていません。クライアントのホームドメインの cred テーブルに対して nismatch を使用して、クライアントのホストに適切な資格が割り当てられているかどうか調べます。割り当てられていない場合は、DES の資格を作成します。
- DES 暗号化に失敗しました。エラーメッセージ「authdes_marshall: DES encryption failure」を参照してください。

セキュリティ鍵の問題の詳細については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

authdes_refresh: unable to synchronize clock

このエラーメッセージは、クライアントとサーバーそれぞれのクロックが同期していないことを示しています。通常は、この問題は自動的に訂正されます。しかし、このメッセージの後に、タイムスタンプに関連したエラーが発生した場合は、手作業でクロックを同期させなければなりません。この問題が再度発生した場合は、rpcbind が正しく機能しているかどうかチェックします。

authdes_refresh: unable to synch up w/server

クライアントとサーバーそれぞれのクロックを同期させることに失敗しました。サーバー上の rpcbind プロセスが応答しなかったことが原因となった可能性もあります。サーバー上で ps -ef を実行し、rpcbind が動作しているかどうか調べます。動作していない場合は、rpcbind を再度起動します。このメッセージの後に、タイムスタンプに関係したエラーが発生した場合は、rdate servername を使用して、クライアントのクロックを手作業でサーバーのクロックに同期させなければなりません。

authdes_seccreate: keyserv is unable to generate session key

このエラーメッセージは、keyserv がこのセッション用のランダムな DES キーを生成できなかったことを示しています。このメッセージが表示された場合、ユーザーの側で何らかの操作を行う必要があります。

- keyserv が正しく動作しているかどうかチェックします。動作していない場合は、automountd、rpc.nisd、sendmail などのように、secure RPC を使用したり NIS+ のコールを行う、長時間にわたって動作するプロセスとともに、keyserv を再度起動します。次に、keylogin を実行します。
- keyserv が起動されていて正しく動作している場合は、このエラーをログに記録したプロセスをもう一度起動します。

authdes_seccreate: no public key found for servername

クライアント側で、サーバー名 servername の DES の資格を取得できませんでした。このメッセージが表示された場合、ユーザーの側で何らかの操作を行う必要があります。

- servername に DES の資格が割り当てられているかどうかチェックします。割り当てられていない場合は、DES の資格を作成します。
- スイッチ構成ファイルをチェックして、指定されているネームサービスを調べ、そのサービスが応答するかどうか確認します。応答しない場合は、ネームサービスを再度起動します。

authdes_seccreate: out of memory

詳細については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

authdes_seccreate: unable to gen conversation key

keyserv プロセスは、ランダムな DES キーを生成できませんでした。もっとも可能性の高い原因は、keyserv プロセスが停止しているか、応答しないことです。ps -ef を使用して、ホスト上で keyserv プロセスが動作しているかどうか確認します。動作していない場合は、keyserv を起動し、次に keylogin を実行します。

keyserv を起動し直しても問題が解決しない場合は、secure RPC を使用したり NIS+ をコールするプロセス (たとえば、automountd、rpc.nisd、sendmail) が動作していない可能性があります。これらのプロセスが動作しているかどうかチェックし、動作していない場合は再度起動します。

セキュリティ鍵の問題の詳細については、『Solaris のシステム管理 (ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

`authdes_validate: DES decryption failure`

「`authdes_marshal: DES encryption failure`」を参照してください。

`authdes_validate: verifier mismatch`

クライアントがサーバーに送ったタイムスタンプが、サーバーから受け取ったタイムスタンプに一致していません (Secure RPC セッションの中では回復できません)。考えられる原因は次のとおりです。

- クライアントかサーバーのキャッシュに格納されているセッション鍵またはタイムスタンプのデータが壊れています。
- サーバーがキャッシュから、まだ有効なセッションのセッション鍵を削除しました。
- ネットワークデータが壊れています。

コマンドを再度実行してください。

`CacheBind: xdr_directory_obj failed.`

もっとも可能性のある原因は次のとおりです。

- `xdr_directory_obj` ルーチンに渡されたパラメータが不適切です。直前に入力したコマンドの構文が正確かどうかチェックします。
- システムへのメモリーの割り当てに失敗しました。メモリーの問題については、『Solaris のシステム管理 (ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。
- コマンドの構文が正確で、システムもメモリー不足を起こしていないと思われる場合は、ご購入先にご連絡ください。

`Cache expired`

オブジェクトのキャッシュから返されたエントリが、有効期限を過ぎています。つまり、持続時間の値が 0 になり、エントリが変更された可能性があることを示しています。フラグ `-NO CACHE` が検索用関数に渡された場合は、その検索関数は動作を繰り返し行い、有効期限を過ぎていないオブジェクトのコピーの取得を試みます。

このメッセージは、NIS+ のエラーコード定数 `NIS_CACHEEXPIRED` によって生成されます。詳細な情報は、`nis_tables` と `nis_names` のマニュアルページを参照してください。

`Callback: - select failed message nnnn`

内部システムコールがエラーを起こしました。ほとんどの場合、この問題は自動的に修正されます。自動的に修正されない場合は、`rpc.nisd` が異常終了していないかどうか確認します。異常終了している場合は、再度起動します。この問題が頻繁に発生する場合は、ご購入先にご連絡ください。

CALLBACK_SVC: bad argument

内部システムコールがエラーを起こしました。ほとんどの場合、この問題は自動的に修正されます。自動的に修正されない場合は、rpc.nisd が異常終了していないかどうか確認します。異常終了している場合は、再度起動します。この問題が頻繁に発生する場合は、ご購入先にご連絡ください。

Cannot grow transaction log error *string*

システムは、ログファイルへの追加を行うことができません。理由は *string* が示すとおりです。もっともよく考えられる原因は、ディスク容量の不足です。詳細については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

Cannot truncate transaction log file

ログにチェックポイントを実行しようとしたますが、rpc.nisd デーモンは、チェックポイントが設定されたエントリをログから削除した後で、ログファイルの圧縮を試みました。このルーチンが失敗する要因については、ftruncate のマニュアルページを参照してください。『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」も参照してください。

Cannot write one character to transaction log, *errormessage*

rpc.nisd デーモンは、現在のトランザクションを使用して、トランザクションログに更新部分を追加しようとしたますが失敗しました。原因は、関数が返した *message* の中に示されています。書き込みルーチンのマニュアルページの中に詳細が記されていることもあります。

Can't compile regular expression *variable*

keypat 内の正規表現が不適切だったために、nisgrep コマンドによってエラーメッセージが返されました。

Can't get any map parameter information.

詳細については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

Can't find name service for passwd

「nsswitch.conf ファイルがない」、「nsswitch.conf ファイルに passwd のエントリがない」、「passwd のエントリが意味をなさない」、「passwd のエントリの形式が正しくない」のいずれかを意味するエラーメッセージです。

Can't find *name* 's secret key

考えられる原因は次のとおりです。

- 正しくないパスワードを入力しました。
- cred テーブルの中に、*name* のエントリがない
- NIS+ が鍵を復号できませんでした (エントリが壊れている可能性があります)。
- nsswitch.conf ファイルが、cred テーブルの中に記録されている NIS+ パスワードとは異なるパスワードを使用して、/etc/passwd ファイルの中に格納されているローカルパスワードの照会を行なっています。

これらの問題を診断および解決する方法については、『*Solaris* のシステム管理 (ネーミングとディレクトリサービス : *DNS*、*NIS*、*LDAP* 編)』の「*NIS* の障害追跡」を参照してください。

`checkpoint_log: Called from read only child ignored.`

このメッセージは、読み取り専用プロセスが、親プロセスに限定されている操作を試みて、その操作が異常終了したことを示す状態メッセージです。なんらかの操作を行う必要はありません。

`checkpoint_log: Unable to checkpoint, log unstable.`

安定した状態にないログに対して、チェックポイントを実行しようとした。つまり、ログは再同期、更新、チェックポイントのいずれかの状態にありました。ログが安定した状態になるまで待つて、`nisping` コマンドを再度実行します。

`check_updaters: Starting resync.`

システムの状態メッセージです。なんらかの操作を行う必要はありません。

`Child process requested to checkpoint!`

このメッセージは、システムが修正することのできる、ソフトウェアの小さな問題が発生したことを意味しています。これらのメッセージが頻繁に表示される場合は、`/etc/syslog.conf` ファイル内のしきい値レベルを変更します。詳細は、`syslog.conf` のマニュアルページを参照してください。

`Column not found: columnname`

指定した列名は、指定したテーブルの中に存在しません。

`Could not find string 's secret key`

考えられる原因は次のとおりです。

- 正しくないパスワードを入力しました。
- `cred` テーブルの中に、`name` のエントリがありません。
- `NIS+` が鍵を復号できませんでした (エントリが壊れている可能性があります)。
- `nsswitch.conf` ファイルで、公開鍵の指定方法を間違えています。`cred` テーブルの中に登録されている `NIS+` パスワードとは異なるパスワードを使用して、`/etc/publickey` ファイルの中に格納されているローカル公開鍵の照会を行なっています。

これらの問題を診断および解決する方法については、『*Solaris* のシステム管理 (ネーミングとディレクトリサービス : *DNS*、*NIS*、*LDAP* 編)』の「*NIS* の障害追跡」を参照してください。

`Could not generate netname`

`keylogin` を実行する際、`Secure RPC` のソフトウェアが、ユーザーの `UID` に対応する `Secure RPC` のネット名を生成できませんでした。考えられる原因は次のとおりです。

- マシンのホームドメインにある `NIS+` の `cred` テーブルで、`LOCAL` の資格が割り当てられていません。

- /etc/passwd ファイル内にあるローカルエントリの UID が、NIS+ の passwd テーブル内にある UID と違っています。

string: could not get secret key for '*string*
考えられる原因は次のとおりです。

- 正しくないパスワードを入力しました。
- cred テーブルの中に、name のエントリがありません。
- NIS+ が鍵を復号できませんでした (エントリが壊れている可能性があります)。
- nsswitch.conf ファイルで、公開鍵の指定方法を間違えています。cred テーブルの中に登録されている NIS+ パスワードとは異なるパスワードを使用して、/etc/publickey ファイルの中に格納されているローカル公開鍵の照会を行なっています。

これらの問題を診断および解決する方法については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

Couldn't fork a process!

サーバーは、コールバックの要求を満たす子プロセスをフォークできませんでした。おそらく、システムが最大のプロセス数に達したためです。不要なプロセスをいくつか終了するか、システムが処理できるプロセス数を増やしてください。詳細については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

Couldn't parse access rights for column *string*

このメッセージは、通常 `nistbladm -u` コマンドで、演算子として + (プラス記号)、- (マイナス記号)、= (等号) 以外の記号を入力したときに返されます。ほかに考えられる原因として、列の権利をコンマで区切っていないことや、アクセス権の種類として r、d、c、m のどれでもない文字を指定したことが挙げられます。エントリのタイプによるエラーの場合は、構文をチェックしてください。入力为正しくてもエラーが発生する場合は、テーブルが壊れていることも考えられます。

Database for table does not exist

テーブルの参照に失敗しました。原因については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

このメッセージは、NIS+ のエラーコード定数 `NIS_NOSUCHTABLE` によって生成されます。詳細な情報は、`nis_tables` と `nis_names` のマニュアルページを参照してください。

`_db_add`: child process attempting to add/modify `_db_addib`:
non-parent process attempting an add

これらのメッセージは、読み取り専用プロセスや、親プロセス以外のプロセスが、データベース内のオブジェクトを追加または変更しようとしたことを示します。ほとんどの場合、なんらかの操作を行う必要はありません。この問題が頻繁に発生する場合は、ご購入先にご連絡ください。

`db_checkpoint: Unable to checkpoint string`

このメッセージは、なんらかの理由で、NIS+ がディレクトリにチェックポイントを実行できなかったことを示します。ほとんどの場合、ディスクが一杯になっています。詳細については、『Solaris のシステム管理 (ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

`_db_remib: non-parent process attempting an remove _db_remove:
non-parent process attempting a remove`

これらのメッセージは、読み取り専用プロセスや、親プロセス以外のプロセスが、テーブルエントリを削除しようとしたことを示します。ほとんどの場合、なんらかの操作を行う必要はありません。この問題が頻繁に発生する場合は、ご購入先にご連絡ください。

`Do you want to see more information on this command?`

このメッセージは、スクリプトのコマンド行の中に、なんらかの構文エラーかスペルミスがあることを示します。

`Entry/Table type mismatch`

テーブルでエントリを追加または変更しようとして、ほかのタイプのテーブルからエントリを引き渡した場合に、このメッセージが表示されます。たとえば、列数が等しくないことが考えられます。更新に使っているテーブルのタイプが正確に一致しているかどうかチェックします。

このメッセージは、NIS+ のエラーコード定数 `NIS_TYPERMISMATCH` によって生成されます。詳細な情報は、`nis_tables(3N)` のマニュアルページを参照してください。

`**ERROR: chkey failed again. Please contact your network
administrator to verify your network password.`

このメッセージは、ネットワークのパスワードの入力ミスを示します。

- マシンをはじめて初期設定するときは、ネットワーク管理者に連絡を取り、ネットワークのパスワードを確認します。
- このマシンが、同じドメインの中で NIS+ のクライアントとして初期設定されたことがある場合は、Secure RPC のパスワードプロンプトで、root のログインパスワードを入力してみます。
- このマシンが現在 NIS+ のクライアントであり、ほかのドメインのクライアントに変更したい場合は、`/etc/.rootkey` ファイルを削除し、ネットワーク管理者から受け取ったパスワード (または `nispopulate` スクリプトから生成したパスワード) を使用して、`nisclient` スクリプトを再度実行します。

`Error: Could not create a valid NIS+ coldstart file`

このメッセージは、NIS+ の初期設定ルーチン `nisinit` によって生成されます。この後に、「lookup:…」で始まるメッセージが表示されます。2 番目のメッセージは、NIS+ の有効なコールドスタートファイルが作成できなかった理由を示します。

****ERROR: could not restore file *filename***

このメッセージは、NIS+ が、*filename.no_nisplus* を *filename* に変更できなかったことを示します。

システムコンソールを使用して、システムエラーメッセージの有無をチェックします。

- システムエラーメッセージがある場合は、そのエラーメッセージの示す問題を解決し、`nisclient -i` を再度実行します。
- システムエラーメッセージがない場合は、そのファイルを手作業で名前を変更し、`nisclient -i` を再度実行します。

****ERROR: Couldn't get the server NIS+_server's address.**

このスクリプトは、指定されたドメインのサーバーの IP アドレスを取り出すことができませんでした。/etc/hosts ファイルまたは /etc/inet/ipnodes ファイル内のサーバー *NIS+_server* に、手作業で IP アドレスを追加し、`nisclient -i` を再度実行します。

****ERROR: directory *directory-path* does not exist.**

このメッセージは、正しくないディレクトリパス名を入力したことを示します。正しいディレクトリパス名を入力します。

****ERROR: *domainname* does not exist.**

このメッセージは、存在しないドメインを複製しようとしたことを示します。

- *domainname* のスペルが正しくない場合は、正しいドメイン名を指定して、このスクリプトをもう一度実行します。
- *domainname* のドメインが存在しない場合は、そのドメインを作成します。次に、複製を行うことができます。

****ERROR: *parent-domain* does not exist.**

このメッセージは、コマンド行で指定されたドメインの親ドメインが存在しないことを示します。このメッセージは、ルートマスター以外のサーバーの設定を行なっているときにだけ表示されるはずです。

- ドメイン名のスペルが正しくない場合は、正しいドメイン名を指定して、もう一度このスクリプトを実行します。
- このドメインの親ドメインが存在しない場合は、最初に親ドメインを作成し、次にこのドメインを作成します。

****ERROR: Don't know about the domain "*domainname*". Please check your *domainname*.**

このメッセージは、認識できないドメイン名が入力されたことを示します。正しいドメイン名を指定して、もう一度このスクリプトを実行します。

****ERROR: failed dumping *tablename* table.**

このスクリプトは、指定されたテーブルのダンプに失敗したため、cred テーブルを生成できませんでした。

- `niscat tablename .org_dir` の実行に失敗した場合は、すべてのサーバーが動作しているかどうか確認し、もう一度このスクリプトを実行して、`tablename` テーブルを生成します。
- `niscat tablename .org_dir` が実行できる場合は、NIS+ のサーバーが一時的にビジーだったために、エラーが発生した可能性があります。もう一度このスクリプトを実行して、`tablename` テーブルを生成します。

****ERROR: host *hostname* is not a valid NIS+ principal in domain *domainname*. This host name must be defined in the credential table in domain *domainname*. Use `nisclient -c` to create the host credential**

あるマシンを NIS+ のサーバーにする前に、そのマシンを NIS+ のクライアントとし、適切な資格を割り当てる必要があります。マシンを NIS+ のルート複製サーバーに変換するには、最初にそのマシンをルートドメインの NIS+ クライアントにする必要があります。ドメインに新しいクライアントを追加するための指示に従い、`nissserver -R` を再度実行します。

マシンを NIS+ のルート以外のマスターサーバーや複製サーバーに変換するには、そのマシンを、サービスを提供するドメインの親ドメインの NIS+ クライアントにしておかなければなりません。ドメインに新しいクライアントを追加するための指示に従い、`nissserver -M` か `nissserver -R` を再度実行します。

ルートマスターサーバーを設定するときは、この問題は発生しないはずです。

Error in accessing NIS+ cold start file is NIS+ installed?

このメッセージは、NIS+ がマシンにインストールされていない場合や、何らかの理由で `/var/nis/NIS_COLD_START` ファイルが見つからないかアクセスできない場合に発生します。`/var/nis/NIS_COLD_START` ファイルが存在しているかどうかチェックします。ファイルが存在している場合は、パスが正しく設定されているかどうか、また `NIS_COLD_START` に正しいアクセス権が割り当てられているかどうか確認します。次に、古いコールドスタートファイル名の変更か削除を行い、`nisclient` スクリプトをもう一度実行して、NIS+ をマシンにインストールします。

このメッセージは、NIS+ エラーコード定数 `NIS_COLDSTART_ERR` を送信したキャッシュマネージャにより生成されます。ファイルにアクセスできない原因の詳細については、`write` および `open` のマニュアルページを参照してください。

Error in RPC subsystem

これは致命的なエラーで、RPC サブシステムになんらかの障害が発生したことを示します。通常は、クライアント側かサーバー側に、RPC 要求が失敗した理由を示す `syslog` メッセージがあります。

このメッセージは、NIS+ のエラーコード定数 `NIS_RPCERROR` によって生成されます。詳細な情報は、`nis_tables` と `nis_names` のマニュアルページを参照してください。

****ERROR: it failed to add the credential for root.**

NIS+ のコマンド `nisaddcred` は、ルートマスターサーバーの設定を行なっているときに、`root` の資格を作成できませんでした。システムコンソールを使用して、システムエラーメッセージの有無をチェックします。

- システムエラーメッセージが発生している場合は、そのエラーメッセージの示す問題を解決し、`nissserver` を再度実行します。
- システムエラーメッセージが発生していない場合は、`rpc.nisd` プロセスが動作しているかどうかチェックします。動作していない場合は、`rpc.nisd` をもう一度起動し、`nissserver` を再度実行します。

****ERROR: it failed to create the tables.**

NIS+ のコマンド `nissetup` は、ディレクトリやテーブルを作成できませんでした。システムコンソールを使用して、システムエラーメッセージの有無をチェックします。

- システムエラーメッセージが発生している場合は、そのエラーメッセージの示す問題を解決し、`nissserver` を再度実行します。
- システムエラーメッセージが発生していない場合は、`rpc.nisd` プロセスが動作しているかどうかチェックします。動作していない場合は、`rpc.nisd` をもう一度起動し、`nissserver` を再度実行します。

****ERROR: it failed to initialize the root server.**

NIS+ のコマンド `nisinit -r` は、ルートマスターサーバーの初期設定に失敗しました。システムコンソールを使用して、システムエラーメッセージの有無をチェックします。システムエラーメッセージがある場合は、そのエラーメッセージの示す問題を解決し、`nissserver` を再度実行します。

****ERROR: it failed to make the *domainname* directory**

`nissserver` を実行して、ルート以外のマスターを作成しているときに、NIS+ のコマンド `nismkdir` は、*domainname* という新しいディレクトリの作成に失敗しました。親ドメインに、新しいドメインを作成するために必要な作成権が割り当てられていません。

- そのドメインの所有者ではなく、親ドメインのグループメンバーでもない場合は、所有者か親ドメインのグループメンバーとして、そのスクリプトをもう一度実行します。
- `rpc.nisd` が、作成中のドメインのサーバーとなるマシンで動作していない場合は、`rpc.nisd` をもう一度起動します。

****ERROR: it failed to promote new master for the *domainname* directory**

`nissserver` スクリプトを使ってルート以外のマスターを作成しているときに、NIS+ のコマンド `nismkdir` は、ディレクトリ *domainname* の新しいマスターの変換に失敗しました。

- このドメインの親ドメインで変更権を割り当てられていない場合は、親ドメインの所有者かグループメンバーとして、そのスクリプトをもう一度実行します。

- `rpc.nisd` が、変換先のドメインのサーバーで動作していない場合は、これらのサーバー上で `rpc.nisd` をもう一度起動し、`nissserver` を再度実行します。

****ERROR: it failed to replicate the *directory-name* directory**
NIS+ のコマンド `nismkdir` は、ディレクトリ *directory-name* の新しい複製サーバーを作成できませんでした。

- `rpc.nisd` が、複製サーバーを作成したいドメインのマスターサーバー上で動作していない場合は、マスターサーバーで `rpc.nisd` をもう一度起動し、`nissserver` を再度実行します。
- `rpc.nisd` が、新しい複製サーバー上で動作していない場合は、新しい複製サーバーで `rpc.nisd` をもう一度起動し、`nissserver` を再度実行します。

****ERROR: invalid group name. It must be a group in the *root-domain* domain.**

このメッセージは、ルートマスターサーバーを構成しているときに、不適切なグループ名が使用されたことを示します。*root-domain* の適切なグループ名を指定して、`nissserver -r` をもう一度実行します。

****ERROR: invalid name "*client-name*" It is neither an host nor an user name.**

このメッセージは、不適切な *client-name* が入力されたことを示します。

- *client-name* のスペルが正しくない場合は、正しい *client-name* を指定して、`nisclient -c` をもう一度実行します。
- *client-name* のスペルは正しくても、適切なテーブルの中に存在していない場合は、適切なテーブル内に *client-name* を記述して、`nisclient -c` をもう一度実行します。たとえば、ユーザーのクライアントは `passwd` テーブルの中に存在し、ホストのクライアントは `hosts` テーブルの中に存在します。

****ERROR: *hostname* is a master server for this domain. You cannot demote a master server to replica. If you really want to demote this master, you should promote a replica server to master using `nissserver` with the M option.**

同じドメインの中で、マスターサーバーを直接複製サーバーに変換することはできません。しかし、複製サーバーのホスト名を指定して `nissserver -M` を実行することにより、複製サーバーを新しいマスターサーバーに変換できます。同時に、元のマスターは自動的に複製サーバーに変換されます。

****ERROR: missing hostnames or usernames.**

このメッセージは、コマンド行でクライアント名が入力されなかったことを示します。クライアント名を指定して、`nisclient -c` をもう一度実行します。

****ERROR: NIS+ group name must end with a "."**

このメッセージは、ピリオドで終わる完全指定のグループ名が指定されなかったことを示します。完全指定のグループ名を使用して、このスクリプトをもう一度実行します。

****ERROR: NIS+ server is not running on *remote-host*. You must do the following before becoming an NIS+ server: 1. become an NIS+ client of the parent domain or any domain above the domain which you plan to serve. (nisclient) 2. start the NIS+ server. (rpc.nisd)**

このメッセージは、NIS+ に変換しようとしているリモートマシン上で、`rpc.nisd` が動作していないことを示します。`nisclient` スクリプトを使用して、そのマシンを親ドメイン、またはサービスを提供するドメインより上位のドメインの NIS+ クライアントにします。次に、*remote-host* 上で `rpc.nisd` を起動します。

****ERROR: nisinit failed.**

`nisinit` は、`NIS_COLD_START` ファイルを作成できませんでした。

次のことをチェックします。

- `-H` オプションで指定した NIS+ サーバーが動作しているかどうか (ping を使用して確認する)
- 正しいドメイン名を入力したかどうか
- `rpc.nisd` がサーバー上で動作しているかどうか
- 未認証クラスに、そのドメインでの読み取り権が割り当てられているかどうか

****ERROR: NIS map transfer failed. *tablename* table will not be loaded.**

NIS+ は、このテーブルの NIS マップを、NIS+ データベースに転送できませんでした。

- NIS サーバーのホストが動作している場合は、もう一度このスクリプトを実行します。一時的な障害が原因で、エラーが発生した可能性があります。
- すべてのテーブルでこのエラーが発生した場合は、ほかの NIS サーバーを使用して、このスクリプトをもう一度実行します。

****ERROR: no permission to create directory *domainname***

親ドメインに、新しいドメインを作成するために必要な作成権が割り当てられていません。そのドメインの所有者ではなく、親ドメインのグループメンバーでもない場合は、所有者か親ドメインのグループメンバーとして、そのスクリプトをもう一度実行します。

****ERROR: no permission to replicate directory *domainname*.**

このメッセージは、ドメインを複製するためのアクセス権が割り当てられていないことを示します。このドメインの所有者かグループメンバーとして、スクリプトをもう一度実行します。

error receiving zone transfer

DNSのエラーメッセージです。通常これは、主サーバーのDNS ファイルの1つの構文エラーを示します。詳細については、『Solaris のシステム管理 (ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

****ERROR: table *tablename* .org_dir.domainname does not exist." *tablename* table will not be loaded."**

スクリプトは、NIS+ のテーブル *tablename* を見つけることができませんでした。

- *tablename* のスペルが正しくない場合は、正しいテーブル名を指定して、スクリプトをもう一度実行します。
- *tablename* が NIS+ の標準テーブルの 1 つであり、かつ *tablename* テーブルが存在しない場合は、*nissetup* を使用して、そのテーブルを作成します。
tablename が NIS+ の標準テーブルではなく、また存在しない場合は、*nistbladm* を使用して、独自のテーブル *tablename* を作成します。次に、このスクリプトを再度実行して、このテーブルを生成します。
- *tablename* テーブルが存在する場合は、NIS+ のサーバーが一時的にビジーであったためにエラーが発生した可能性があります。このスクリプトをもう一度実行して、*tabelname* テーブルを生成します。

****ERROR: this name "*clientname*" is in both the passwd and hosts tables. You cannot have an username same as the host name.**

client-name が、*passwd* テーブルと *hosts* テーブルの両方に記述されています。両方のテーブルに、共通の名前を記述することは認められていません。*passwd* と *hosts* どちらかのテーブルから、手作業でこのエントリを削除します。次に、*nisclient -c* を再度実行します。

****ERROR: You cannot use the -u option as a root user.**

このメッセージは、スーパーユーザーが *nisclient -u* を実行しようとしたことを示します。*-u* オプションは、普通のユーザーだけを初期設定するためのものです。スーパーユーザーは、NIS+ クライアントとして初期設定する必要がありません。

****ERROR: You have specified the Z option after having selected the X option. Please select only one of these options [*list*]. Do you want to see more information on this command?**

いま実行しているスクリプトでは、リスト表示されているオプションのうち、どれか 1 つしか使用できません。

- *y* と入力すると、詳細な情報が表示されます。
- *n* と入力すると、スクリプトは停止して、元のプロンプトに戻ります。

スクリプトを終了した後で、オプションを 1 つだけ指定して、もう一度実行します。

****ERROR: you must specify a fully qualified groupname.**

このメッセージは、ピリオドで終わる完全指定のグループ名が指定されなかったことを示します。完全指定のグループ名を使用して、このスクリプトをもう一度実行します。

****ERROR: you must specify both the NIS domainname (-y) and the NIS server host name (-h).**

このメッセージは、NIS のドメイン名と NIS サーバーのホスト名的一方または両方を指定しなかったことを示します。プロンプトかコマンド行で、NIS のドメイン名と NIS サーバーのホスト名を入力します。

****ERROR: you must specify one of these options: -c, -i, -u, -r.**

このメッセージは、これらのオプション -c、-i、-u、-r のどれかがコマンド行に指定されていなかったことを示します。正しいオプションを指定して、スクリプトをもう一度実行します。

****ERROR: you must specify one of these options: -r, -M or -R"**

このメッセージは、-r、-M、-R のどのオプションも指定されなかったことを示します。正しいオプションを指定して、スクリプトをもう一度実行します。

****ERROR: you must specify one of these options: -C, -F, or -Y**

このメッセージは、-Y、-F どちらのオプションも入力しなかったことを示します。正しいオプションを指定して、スクリプトをもう一度実行します。

****ERROR: You must be root to use -i option.**

このメッセージは、普通のユーザーが `nisclient -i` を実行しようとしたことを示します。`nisclient -i` を実行できるアクセス権を割り当てられているのは、スーパーユーザーだけです。

Error while talking to callback proc

サーバーがクライアントからコールバックされているときに、サーバー上で RPC エラーが発生しました。この時点でトランザクションは異常終了し、まだ送信されていないデータは破棄されます。サーバー上の `syslog` で詳細をチェックします。

このメッセージは、NIS+ のエラーコード定数 `NIS_CBERROR` によって生成されます。詳細な情報は、`nis_tables(3N)` のマニュアルページを参照してください。

First/Next chain broken

このメッセージは、コールバックルーチンが結果を返しているときに、クライアントとサーバーの接続が解除されたことを示します。プロセスの実行中にサーバーが終了すると、このようなメッセージが発生することがあります。

このメッセージは、NIS+ のエラーコード定数 `NIS_CHAINBROKEN` によって生成されます。

Generic system error

要求をしているときに、何らかの一般的なシステムエラーが発生しました。システムの `syslog` の記録をチェックし、サーバーから返されたエラーメッセージを探します。

このメッセージは通常、サーバーがクラッシュしたか、データベースが壊れたことを示します。このメッセージは、サーバーまたは複製サーバーの名前を正しく指定していない場合にも生成される可能性があります。たとえば、サーバーや複製サー

バーが自らサービスを提供するドメインより上位ドメインに属しているが、サービスを提供しているドメインに属しているように指定した場合に生成されます。詳細については、『*Solaris* のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

このメッセージは、NIS+ のエラーコード定数 `NIS_SYSTEMERROR` によって生成されます。詳細な情報は、`nis_tables` と `nis_names` のマニュアルページを参照してください。

Illegal object type for operation

これらの問題については、『*Solaris* のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

このメッセージは、NIS+ のエラーコード定数 `DB_BADOBJECT` によって生成されます。

insufficient permission to update credentials.

このメッセージは、動作に必要なアクセス権が割り当てられていない状態で、`nisaddcred` コマンドを実行したときに発生します。テーブル、列、エントリどれかのレベルでアクセス権が不足している可能性があります。`niscat -o cred.org_dir` を使用して、`cred` テーブルでどのようなアクセス権が割り当てられているを確認します。アクセス権を追加する場合は、ユーザー自身またはシステム管理者がオブジェクトのアクセス権の要件を変更するか、必要なアクセス権が割り当てられているグループにユーザーを追加します。

アクセス権の詳細については、『*Solaris* のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

Invalid Object for operation

- *Name context* - 関数に渡された名前は、NIS+ の有効な名前ではありません。
- *Table context* - 指定されたオブジェクトは、このテーブルにとって有効な NIS+ エントリのオブジェクトではありません。列数が一致しない場合や、テーブルの中で関連する列とデータ型 (たとえば、バイナリかテキストか) が異なる場合に、このメッセージが発生する可能性があります。

このメッセージは、NIS+ のエラーコード定数 `NIS_INVALIDOBJ` によって生成されます。詳細な情報は、`nis_tables` と `nis_names` のマニュアルページを参照してください。

invalid usecs Routine_name: invalid usecs

このメッセージは、`struct time stamp` タイプの変数フィールド `tv_usecs` の値が、1 秒未満マイクロ秒単位の値より大きいときに発生します。通常は、何らかのタイプのソフトウェアエラーが原因です。

tablename is not a table

tablename という名前を持つオブジェクトは、テーブルオブジェクトではありません。たとえば、`nisgrep` や `nismatch` のコマンド行で、テーブルオブジェクト以外のオブジェクトを指定すると、このメッセージが発生します。

Link Points to illegal name

渡された名前を LINK タイプのオブジェクトとして解決しましたが、そのオブジェクトの内容が不適切な名前を指しています。

テーブルエントリをリンクさせることができません。このエラーメッセージはエントリレベルでリンクを作成しようとするると発生します。

このメッセージは、NIS+ のエラーコード定数 NIS_LINKNAMEERROR によって生成されます。詳細な情報は、nis_tables と nis_names のマニュアルページを参照してください。

Load limit of *number* reached!

サーバー上で子プロセスの数がすでに上限に達しているときに、子プロセスを作成しようとした。このメッセージは、メッセージをログに記録する基準のレベルが LOG_WARNING を含む場合にのみ、サーバーのシステムログに記録されます。

login and keylogin passwords differ.

nispasswd を使用してパスワードを変更し、システムはパスワードを変更しましたが、cred テーブル内の資格のエントリを新しいパスワードに更新できず、さらに、passwd テーブルの中で元のパスワードに復元もできない場合に、このメッセージが発生します。このメッセージの後に、次の指示が表示されます。

Use NEW password for login and OLD password for
keylogin. Use "chkey -p" to reencrypt the credentials with
the new login password. You must keylogin explicitly after
your next login.

この指示の後に、なぜ元のパスワードに復元することができなかったのか、理由を示す状態メッセージが表示されます。これらのメッセージが表示された場合は、その指示に従ってください。

Login incorrect

このメッセージの原因としてもっとも多いのは、パスワードのタイプミスです。もう一度入力し直してください。また覚えているパスワードが正しいのかも確認してください。特に、大文字と小文字、アルファベットの o と数字の 0、アルファベットの l と数字の 1 を間違えないよう注意してください。

このメッセージが生成されるその他の原因については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

log_resync: Cannot truncate transaction log file

ログにチェックポイントを実行しようとしたが、rpc.nisd デーモンは、チェックポイントが設定されたエントリをログから削除した後で、ログファイルの圧縮を試みました。このルーチンが失敗する要因については、ftruncate のマニュアルページを参照してください。『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」も参照してください。

Malformed Name or illegal name

関数に渡された名前は、NIS+ の有効な名前ではありません。

考えられる理由の一つは、だれかがドメイン名を変更したことです。既存のドメイン名は変更すべきではありません。詳細については、『*Solaris* のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

このメッセージは、NIS+ のエラーコード定数 `NIS_BADNAME` によって生成されます。詳細な情報は、`nis_tables(3N)` のマニュアルページを参照してください。

_map_addr: RPC timed out.

プロセスやアプリケーションが、必要なデータの取得や NIS+ からのホスト名の解決を行うデフォルト時間内に、NIS+ と通信できませんでした。ほとんどの場合、しばらく待機すれば解決します。パフォーマンスの遅れに関する問題の詳細については、『*Solaris* のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

Master server busy full dump rescheduled

このメッセージは、マスターサーバーがビジーであったため、複製サーバーがマスターサーバーから得られたフルダンプを使って自らを更新することができなかったことを示します。詳細については、『*Solaris* のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

String Missing or malformed attribute

属性の名前が、テーブル内の指定された列に一致していません。または、属性に値が割り当てられていません。

コマンドの構文エラーが原因となっている可能性があります。*string* から、何が間違っているのか推測できます。一般的な原因は、スペルミス、等号 (=) を不適切な場所に置いたこと、列名やテーブル名の間違いなどです。

このメッセージは、NIS+ のエラーコード定数 `NIS_BADATTRIBUTE` によって生成されます。詳細な情報は、`nis_tables(3N)` のマニュアルページを参照してください。

Modification failed

コマンドを実行しているときに、だれかがグループ名を変更したため、`nisgrpadm` がこのメッセージを返しました。だれかがグループの操作を行なっているかどうかチェックします。次に、このコマンドを再度実行します。

このメッセージは、NIS+ のエラーコード定数 `NIS_IBMODERROR` によって生成されます。

Modify operation failed

試みた変更が、何らかの理由で失敗しました。

このメッセージは、NIS+ のエラーコード定数 `NIS_MODFAIL` によって生成されます。詳細な情報は、`nis_tables` と `nis_names` のマニュアルページを参照してください。

Name not served by this server

指定された名前へのサービスを提供していないサーバーに、要求を行いました。通常、このメッセージは発生しないはずです。しかし、内蔵のサーバー検索メカニズムを使用しない場合は、独自のメカニズムが壊れた結果、発生する可能性もあります。

ほかに考えられる原因は次のとおりです。

- コールドスタートファイルが壊れています。 `/var/nis/NIS_COLD_START` を削除して、再起動します。
- ローカルキャッシュの期限切れのような、キャッシュの問題。NIS+ サービスを停止することにより `nis_cachemgr` を終了し、
`/var/nis/NIS_SHARED_DIRCACHE` ファイルを削除して、再起動します。
ルートディレクトリ以外の場所で問題が発生した場合は、単純にドメインのキャッシュマネージャのプロセスを終了し、コマンドをもう一度実行します。
- だれかが複製サーバーから、このディレクトリを削除しました。

このメッセージは、NIS+ のエラーコード定数 `NIS_NOT_ME` によって生成されます。詳細な情報は、`nis_tables` と `nis_names` のマニュアルページを参照してください。

Named object is not searchable

テーブル名を NIS+ のオブジェクトとして解決しましたが、オブジェクトが検索できませんでした。

このメッセージは、NIS+ のエラーコード定数 `NIS_NOTSEARCHABLE` によって生成されます。詳細な情報は、`nis_tables(3N)` のマニュアルページを参照してください。

Name/entry isn't unique

複数のエントリを返す特定の検索基準に基づく動作が要求されました。たとえば、`nistbladm -r` を使用して、`passwd` テーブルからユーザーを削除しようとした。しかし、このテーブルには、そのユーザーに対応するエントリが2つ存在したため、次のメッセージが表示されます。

```
mymachine# nistbladm -r [name=arnold],passwd.org_dir
Can't remove entry: Name/entry isn't unique
```

`-r` ではなく `-R` オプションを使用すると、コマンドを複数のエントリに適用できます。たとえば、`arnold` に対応するすべてのエントリを削除するには、次のように入力します。

```
mymachine# nistbladm -R [name=arnold],passwd.org_dir
```


NIS+ error

「NIS+ サーバーがエラーを返したが、passwd コマンドがエラーの種類を特定できなかった」という場合に使用されるエラーメッセージです。

NIS+ operation failed

この一般的なエラーメッセージは、めったに発生しないはずです。通常はシステムが修正することのできる、ソフトウェアの小さな問題が発生したことを意味しています。これらのメッセージが頻繁に表示される場合や、システムにより修正されないとと思われる場合は、ご購入先にご連絡ください。

このメッセージは、NIS+ のエラーコード定数 NIS_FAIL によって生成されます。

string: NIS+ server busy try again later.

原因については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

NIS+ server busy try again later.

サーバーは現在ビジー状態です。後でコマンドをもう 1 度実行します。

原因については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

NIS+ server for *string* not responding still trying

原因については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

NIS+ server not responding

原因については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

NIS+ server needs to be checkpointed. Use nisping -C*domainname*



注意 – すぐにチェックポイントを設定してください。

サーバーのシステムログが LOG_CRIT のレベルにある場合は、このメッセージが生成されます。これは、ログが大きくなりすぎたことを示します。nisping -C *domainname* を使用してチェックポイントを実行し、ログを切り捨てます。

ログサイズの詳細については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

NIS+ servers unreachable

このソフトエラーは、指定したテーブルオブジェクトが置かれているディレクトリにサービスを提供するサーバーに到達できなかったことを示します。ネットワークの障害やサーバーのクラッシュが起こった場合に、このメッセージが発生する可能性があります。もう一度実行するだけで、成功する可能性もあります。

`nis_tables(3N)` と `nis_names(3N)` のマニュアルページの `-HARD_LOOKUP` フラグの説明を参照してください。

このメッセージは、NIS+ のエラーコード定数 `NIS_NAMEUNREACHABLE` によって生成されます。

NIS+ service is unavailable or not installed

NIS+ のサービスが利用できないか、インストールされていません。このメッセージは、NIS+ のエラーコード定数 `NIS_UNAVAIL` によって生成されます。

NIS+: write ColdStart File: xdr_directory_obj failed

もっとも可能性のある原因は次のとおりです。

- パラメータが正しくありません。直前に入力したコマンドの構文が正確かどうかチェックします。
- システムへのメモリーの割り当てに失敗しました。メモリーの問題については、『*Solaris* のシステム管理 (ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。
- コマンドの構文が正確で、システムもメモリー不足を起こしていないと思われる場合は、ご購入先にご連絡ください。

`nis_checkpoint_svc: readonly child instructed to checkpoint ignored.`

このメッセージは、読み取り専用プロセスが、親プロセスに限定されている操作を試みて、その操作が異常終了したことを示す状態メッセージです。なんらかの操作を行う必要はありません。

`nis_dumplog_svc: readonly child called to dump log, ignore`

このメッセージは、読み取り専用プロセスが、親プロセスに限定されている操作を試みて、その操作が異常終了したことを示す状態メッセージです。なんらかの操作を行う必要はありません。

`nis_dump_svc: load limit reached.`

システムで許可されている子プロセスの最大数に達しました。

`nis_dump_svc: one replica is already resyncing.`

マスターとの同期を実行できる複製は、1 度に 1 つだけです。後でコマンドをもう 1 度実行します。

上記の 3 つのエラーメッセージについては、『*Solaris* のシステム管理 (ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

`nis_dump_svc: Unable to fork a process.`

フォークを行うシステムコールが障害を起こしました。考えられる原因については、`fork(2)` のマニュアルページを参照してください。

`nis_mkdir_svc: read-only child called to mkdir, ignored`

このメッセージは、読み取り専用プロセスが、親プロセスに限定されている操作を試みて、その操作が異常終了したことを示す状態メッセージです。なんらかの操作を行う必要はありません。

`nis_ping_svc: read-only child was ping ignored`

このメッセージは、読み取り専用プロセスが、親プロセスに限定されている操作を試みて、その操作が異常終了したことを示す状態メッセージです。なんらかの操作を行う必要はありません。

`nis_rmdir_svc: readonly child called to rmdir, ignored`

このメッセージは、読み取り専用プロセスが、親プロセスに限定されている操作を試みて、その操作が異常終了したことを示す状態メッセージです。なんらかの操作を行う必要はありません。

`nisaddcred: no password entry for uid userid nisaddcred: unable to create credential.`

これら2つのメッセージが発生するのは、`nispopulate` スクリプトを実行しているときです。`NIS+` のコマンド `nisaddcred` は、リモートドメイン上で、ユーザー ID `userid` に `LOCAL` の資格を割り当てることに失敗しました。このメッセージは、リモートドメインで `passwd` テーブルを生成しているときにだけ発生します。

問題を解決するには、ローカルの `passwd` テーブルに、次のテーブルパスを追加します。

```
# nistbladm -u -p passwd.org_dir. remote-domain passwd.org_dir
```

`remote-domain` は、`nispopulate` を実行したときに `-d` オプションで指定したドメインと同じものでなければなりません。このスクリプトをもう一度実行して、`passwd` テーブルを生成します。

`No file space on server`

`NIS+` のサービスが利用できないか、インストールされていません。

このメッセージは、`NIS+` のエラーコード定数 `NIS_NOFILESPACE` によって生成されます。

`No match`

もっとも可能性の高い原因は、インデックスつきの名前を指定するために角括弧を使用し、そのためにシェルが解釈できなくなったことです。たとえば、インデックスつきの名前の両側を角括弧で囲む場合は、全体をさらに引用符で囲まない限り、シェルは角括弧を解釈することができません。そのため、次のメッセージが表示されます。

```
# nistbladm -m shell=/bin/csh [name=miyoko],passwd.org_dir No match
```

正しい構文は次のとおりです。

```
# nistbladm -m shell=/bin/csh `[name=miyoko],passwd.org_dir`
```

No memory

システムのメモリーが不足しているため、指定された操作を行うことができません。メモリーの問題の詳細については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

Non NIS+ namespace encountered

名前を完全に解決できませんでした。通常は、関数に渡された名前を解決した結果、NIS+ のネームツリーの中にある名前空間が得られたことを示します。つまり、この名前は登録されていないディレクトリの中にあります。このメッセージが発生した場合、エラーとともに、DIRECTORY タイプの NIS+ オブジェクトが返されます。

このメッセージは、NIS+ のエラーコード定数 NIS_FOREIGNNS によって生成されます。詳細な情報は、nis_tables と nis_names のマニュアルページを参照してください。

No password entry for uid *userid* No password entry found for uid *userid*

これらのメッセージは両方とも、ユーザーの資格の作成や追加を試みたときに、passwd テーブルの中にそのユーザーが見つからなかったことを示します。資格の作成や追加を行う前に、passwd テーブルの中にユーザーを追加しておかなければなりません。

- もっとも可能性のある原因は、コマンド行でユーザーの *userid* のスペルミスをしたことです。コマンド行の構文が正確か、またミススペルがないかチェックします。
- 正しいドメインにいるか、またはコマンド行で正しいドメインを指定しているかチェックします。
- コマンド行が正しい場合は、passwd テーブルをチェックして、入力した *userid* のユーザーが存在するかどうか確認します。次のように nismatch を使用します。

```
mymachine# nismatch uid=userid passwd.org_dir.
```

passwd テーブルの中にユーザーが含まれていない場合は、資格を作成する前に、nistbladm か nisaddent を使用して、passwd テーブルにユーザーを追加します。

No shadow password information

「制御に必要な情報が欠けているため、パスワードの有効期間の設定が行えなかった」ということを意味します。

Not found *String* Not found

Names context - 名前空間の中に、指定された名前が存在しません。

Table context - 検索基準に一致するエントリが、テーブルの中に存在しません。検索基準が `null` (すべてのエントリを返す) の場合は、このテーブルが空であり、安全に削除できることを意味します。

-FOLLOW_PATH フラグが設定されている場合にこのエラーが発生するのは、検索基準に一致するエントリが、パス内のどのテーブルにも含まれていないことを意味します。

このメッセージは、NIS+ のエラーコード定数 `NIS_NOTFOUND` によって生成されます。詳細な情報は、`nis_tables` と `nis_names` のマニュアルページを参照してください。

この種類の問題については、『*Solaris* のシステム管理 (ネーミングとディレクトリサービス: *DNS*、*NIS*、*LDAP* 編)』の「*NIS* の障害追跡」を参照してください。

Not Found no such name

このエラーメッセージは、テーブルオブジェクトが置かれているはずの、指定されたディレクトリが存在しないことを示します。サーバーは、自らサービスを提供するテーブルの親サーバーであるはずですが、このテーブルがどのディレクトリに置かれているか認識していません。

このメッセージは、NIS+ のエラーコード定数 `NIS_NOSUCHNAME` によって生成されます。詳細については、`nis_tables` と `nis_names` のマニュアルページを参照してください。

この種類の問題については、『*Solaris* のシステム管理 (ネーミングとディレクトリサービス: *DNS*、*NIS*、*LDAP* 編)』の「*NIS* の障害追跡」を参照してください。

Not master server for this domain

このメッセージは、複製サーバー上のデータベースを直接更新しようとしたことを意味します。

また、ネームサービスを提供するサーバーに変更の要求が行われたものの、そのサーバーがマスターサーバーでなかったことを意味することもあります。ディレクトリオブジェクトが変更され、その結果新しいマスターサーバーが指定された場合に、これが発生します。`/var/nis/NIS_SHARED_DIRCACHE` ファイル内にディレクトリオブジェクトのキャッシュ済みコピーを保持しているクライアントは、NIS+ サービスを停止し、`/var/nis/NIS_SHARED_DIRCACHE` ファイルを削除して、NIS+ サービスを再起動します。

このメッセージは、NIS+ のエラーコード定数 `NIS_NOTMASTER` によって生成されます。詳細な情報は、`nis_tables` と `nis_names` のマニュアルページを参照してください。

Not owner

オブジェクトの所有者だけに認められている操作を試みましたが、あなたは所有者ではありません。

このメッセージは、NIS+ のエラーコード定数 `NIS_NOTOWNER` によって生成されます。

Object with same name exists

すでに存在する名前を追加しようとした。この名前を追加するには、最初に既存の名前を削除して新しい名前を追加するか、既存のオブジェクトの名前を変更します。

このメッセージは、NIS+ のエラーコード定数 `NIS_NAMEEXISTS` によって生成されます。詳細な情報は、`nis_tables` と `nis_names` のマニュアルページを参照してください。

parse error: *string* (key *variable*)

`nisaddent` コマンドが `/etc` ディレクトリ内のデータベースファイルを使用しているときに、ファイルのエントリにエラーが見つかったと、このメッセージが表示されます。最初の *string* は問題について説明し、*key* の右側の *variable* は、障害のあるエントリを示します。`/etc/passwd` ファイルで問題が起こった場合は、`/usr/sbin/pwck` を使用して、このファイルをチェックします。

Partial Success

「要求の受信は正しく行われたが、対応するエントリがなかった」という点を除き、`NIS_NOTFOUND` とほぼ同じです。

このエラーが発生すると、サーバーは、エントリではなくテーブルオブジェクトのコピーを返します。パスの処理や、その他のローカルポリシーの実装をクライアントが行えるようにするためです。

このメッセージは、NIS+ のエラーコード定数 `NIS_PARTIAL` によって生成されます。詳細は、`nis_tables(3N)` のマニュアルページを参照してください。

Passed object is not the same object on server

「オブジェクトを名前空間から削除しようとしたが、削除の対象となるオブジェクトとは別のオブジェクトが要求の中で渡されたために処理が異常終了した」ということを意味します。

このメッセージは、NIS+ のエラーコード定数 `NIS_NOTSAMEOBJ` によって生成されます。詳細な情報は、`nis_tables` と `nis_names` のマニュアルページを参照してください。

Password does not decrypt secret key for *name*

考えられる原因は次のとおりです。

- 正しくないパスワードを入力しました。
- `cred` テーブルの中に、*name* のエントリがありません。
- NIS+ が鍵を復号できませんでした (エントリが壊れている可能性があります)。
- Secure RPC パスワードとログインパスワードが一致していません。
- `nsswitch.conf` ファイルが、`cred` テーブルの中に記録されている NIS+ パスワードとは異なるパスワードを使用して、`/etc/passwd` ファイルの中に格納されているローカルパスワードの照会を行なっています。実際に暗号化されているパスワードは、ローカルの `/etc/shadow` ファイルに格納されています。

これらの問題を診断および解決する方法については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

Password has not aged enough

「パスワードが新しすぎるため、変更ができない」ということを意味します。パスワードは、作成されてから一定の日数 (N 日) が経過するまでは変更ができません。

Permission denied

ある操作を試みましたが、必要なアクセス権が割り当てられていなかったため、このエラーが発生しました。詳細については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

このエラーメッセージは、ログイン、パスワードの問題、あるいは NIS+ のセキュリティの問題に関係があります。発生原因としてもっとも一般的なのは、「ユーザーのパスワードが管理者によってロックされている」あるいは「ユーザーのアカウントが終了している」ということです。

Permissions on the password database may be too restrictive

「NIS+ テーブル中のパスワードフィールドの内容を参照する (あるいは使用する) 権限を、ユーザーが持っていない」ということを意味します。NIS+ のアクセス権については、[第 15 章](#)を参照してください。

Please notify your System Administrator

passwd コマンドでパスワード情報を更新しようとして表示された場合は、「何らかの理由 (数多く考えられる) で成功しなかった」ということを意味します。成功しない理由としては、「サービスが利用できない」、「必要なサーバーが停止している」、「アクセス権がないといった類の問題がある」などがあげられます。セキュリティの問題については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

Please check your /etc/nsswitch.conf file

「nsswitch.conf ファイルで、パスワード更新についてサポートされていない設定が行われている」ということを意味します。どのような設定がサポートされているかは、[296 ページ](#)の「[nsswitch.conf ファイルの必要条件](#)」を参照してください。

Probable success

Name context - 要求は成功しました。しかし、返されたオブジェクトはオブジェクトキャッシュからのものであり、サーバーから直接返されたものではありません。オブジェクトキャッシュからオブジェクトを取り出したい場合は、検索用関数をコールするときに、-NO_CACHE フラグを指定しなければなりません。

Table context - 要求は成功しました。しかし、検索パスの中でテーブルを見つけることができなかったため、テーブルがアクセス可能だった場合と比べると、得られた結果は異なっている可能性があります。

このメッセージは、NIS+ のエラーコード定数 `NIS_S_SUCCESS` によって生成されます。詳細な情報は、`nis_tables` と `nis_names` のマニュアルページを参照してください。

Probably not found

指定されたエントリは、このテーブルの中に存在しません。しかし、パス内のすべてのテーブルが検索されたわけではないので、それらのテーブルの中に、このエントリが含まれている可能性もあります。

このメッセージは、NIS+ のエラーコード定数 `NIS_S_NOTFOUND` によって生成されます。詳細な情報は、`nis_tables(3N)` のマニュアルページを参照してください。

Query illegal for named table

要求された構造体をクライアントのライブラリに渡すときに、問題が検出されました。

このメッセージは、NIS+ のエラーコード定数 `NIS_BADREQUEST` によって生成されます。詳細な情報は、`nis_tables(3N)` のマニュアルページを参照してください。

Reason: can't communicate with ypbind.

『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

replica_update: Child process attempting update, aborted

このメッセージは、読み取り専用プロセスが更新を試みて、それが失敗したことを示す状態メッセージです。

replica_update: error result was string

このメッセージは、複製サーバーへのダンプを行う際に問題が起こったことを示します (*string* は理由を示します)。詳細については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

**replica_update: error result was Master server busy, full dump
rescheduled replica_update: master server busy rescheduling the
resync. replica_update: master server is busy will try later.
replica_update: nis dump result Master server busy, full dump
rescheduled**

これらのメッセージはどれも、サーバーがビジーなので、ダンプが後で行われることを示します。

replica_update: nis dump result nis_perror errorstring

このメッセージは、複製サーバーへのダンプを行う際に問題が起こったことを示します (*error string* は理由を示します)。詳細については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

replica_update: *nnnn* updates *nnnn* errors
更新に成功したことを示す状態メッセージです。

replica_update: WARNING: last_update (*directoryname*) returned 0!
NIS+ のプロセスは、このディレクトリに対応するトランザクションログに、最後に更新されたタイムスタンプを見つけることができませんでした。この結果、システムは、問題の起こったディレクトリについて完全な同期を実行しなければなりません。

Results Sent to callback proc
これは状態メッセージです。なんらかの操作を行う必要はありません。

このメッセージは、NIS+ のエラーコード定数 NIS_CBRESULTS によって生成されます。詳細な情報は、nis_tables(3N) のマニュアルページを参照してください。

root_replica_update: update failed *string*: could not fetch object from master.
このメッセージは、複製サーバーへのダンプを行う際に問題が起こったことを示します。詳細については、『Solaris のシステム管理 (ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

RPC failure: "RPC failure on yp operation."
このメッセージは、NIS クライアントの nsswitch.conf ファイルが、nis ではなく files に設定され、サーバーが /etc/hosts ファイルまたは /etc/inet/ipnodes ファイルに含まれていないときに、ypcat によって戻されます。

Security exception on local system. UNABLE TO MAKE REQUEST.
ユーザーのログイン ID がマシン名と重複したときに、このメッセージが表示されることがあります。詳細については、『Solaris のシステム管理 (ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

date: hostname: sendmail (nnnn) : gethostbyaddr failed
この問題の一般的な原因の 1 つに、0 を前に付けた状態での NIS+、NIS、ファイル、DNS データセットの IP アドレスへの入力があります。たとえば、151.029.066.001 のような IP アドレスは入力しないでください。このアドレスの正確な入力方法は、151.29.66.1 です。151.29.66.1.

Server busy, try again
サーバーがビジーのため、要求をすぐに処理できません。

- 追加、削除、変更の操作を行なったときにこのメッセージが返されるのは、あるディレクトリのサービスを提供しているマスターサーバーが使用できないか、データベースのチェックポイントが実行中であることを意味します。
- サーバーが内部の状態を更新しているときに、このメッセージが返されることもあります。

- `nis_list` を実行した場合、クライアントがコールバックを指定し、サーバーがリソース不足でコールバックを行えないことも考えられます。

サーバーが使用可能になるのを待って、コマンドをもう一度実行します。

このメッセージは、NIS+ のエラーコード定数 `NIS_TRYAGAIN` によって生成されます。詳細な情報は、`nis_tables` と `nis_names` のマニュアルページを参照してください。

Server out of memory

ほとんどの場合、このメッセージは致命的な結果が発生したことを示します。サーバーがヒープ領域を使い果たしたことを意味します。

このメッセージは、NIS+ のエラーコード定数 `NIS_NOMEMORY` によって生成されます。詳細な情報は、`nis_tables` と `nis_names` のマニュアルページを参照してください。

Sorry

ログイン、またはパスワードの変更が拒否されたときに表示されるメッセージです。拒否の理由が表示されないのは、「表示された理由を読むことで、権利のない人がシステムに不正にアクセスできるようになる可能性がある」というセキュリティ上の理由からです。

Sorry: less than *nn* days since the last change

「パスワードが新しすぎるため、変更ができない」ということを意味します。パスワードは、作成されてから一定の日数 (*N* 日) が経過するまでは変更ができません。詳細は、[293 ページの「パスワードの変更」](#)を参照してください。

_svcauth_des: bad nickname

クライアントから受け取ったニックネームが不適切か、壊れています。原因は、おそらくネットワークの混雑です。このメッセージの重大度は、有効にしているセキュリティのレベルに依存します。セキュリティレベルが低い場合は、このメッセージは情報を示すだけです。レベルが高い場合は、このコマンドをもう一度実行しなければならない可能性があります。

_svcauth_des: corrupted window from *principalname*

送られたウィンドウが、バリファイアで送られたものと一致しません。

このメッセージの重大度は、有効にしているセキュリティのレベルに依存します。セキュリティレベルが低い場合は、このメッセージは主に情報を示すために表示されます。高いセキュリティレベルでは、しばらくたってからコマンドを再実行するか、もしくは次に説明するような修正作業を行う必要があります。

考えられる原因は次のとおりです。

- サーバーの鍵の組が変更されています。クライアントが、サーバーの古い公開鍵を使って、その一方でサーバーが `keyserv` でキャッシュされた新しい非公開鍵を持っています。 `keylogin` をクライアントとサーバーの両方で実行してください。

- クライアントの鍵の組が変更されて、クライアントがクライアントシステムで `keylogin` を実行していません。このためシステムはまだ、クライアントの古い非公開鍵をサーバーに送付しており、そのサーバーはクライアントの新しい公開鍵を使用しています。このため、2つの鍵は一致しません。`keylogin` を再度クライアントとサーバーの両方で実行してください。
- ネットワークでデータが壊れました。コマンドをもう一度実行してください。これで機能しない場合には、`snoop` コマンドを使って、ネットワークの問題を調査、修正してください。その後で、再度クライアントとサーバーの両方で `keylogin` を実行してください。

`_svcauth_des: decryption failure`

特定の認証データの DES 復号化に失敗しました。考えられる原因は次のとおりです。

- ライブラリ関数か引数が壊れています。
- DES 暗号化チップを使用している場合、そのチップに問題があります。

このメッセージの重大度は、有効にしているセキュリティのレベルに依存します。セキュリティレベルが低い場合は、このメッセージは主に情報を示すために表示されます。レベルが高い場合は、ご購入先に連絡してください。問題が、DES 暗号化チップに関連すると思われる場合は、ご購入先に連絡してください。

`_svcauth_des: decryption failure for principalname`

特定の認証データの DES 復号化に失敗しました。考えられる原因は次のとおりです。

- ライブラリ関数か引数が壊れています。
- DES 暗号化チップを使用している場合、そのチップに問題があります。

このメッセージの重大度は、有効にしているセキュリティのレベルに依存します。セキュリティレベルが低い場合は、このメッセージは主に情報を示すために表示されます。レベルが高い場合は、ご購入先に連絡してください。問題が、DES 暗号化チップに関連すると思われる場合は、ご購入先に連絡してください。

`_svcauth_des: invalid timestamp received from principalname`

クライアントから受け取ったタイムスタンプが壊れています。または、サーバーが間違った鍵を使って、復号しようとしています。考えられる原因は次のとおりです。

- ネットワークの混雑。コマンドをもう一度実行します。
- サーバーのキャッシュから、このクライアント用のエントリが削除されました。ネットワークの負荷をチェックします。

`_svcauth_des: key_decryptsessionkey failed for principalname`

`keyserv` プロセスは、特定の公開鍵を使用してセッションキーを復号化することに失敗しました。考えられる原因は次のとおりです。

- `keyserv` プロセスが終了しているか、応答しません。`ps -e` を使用して、`keyserv` ホスト上で `keyserv` プロセスが動作しているかどうか確認します。動作していない場合は、NIS+ サービスを再起動し、`keylogin` を実行しま

す。

- サーバー主体のキーログインが行われていません。サーバー主体のキーログインを実行します。
- サーバー主体 (ホスト) に資格が割り当てられていません。クライアントのホームドメインにある cred テーブルを対象にして、`nismatch hostname.domainname.cred.org_dir` を実行します。必要に応じて、新しい資格を作成します。
- `keyserv` は再度起動されているようですが、長時間にわたって動作するアプリケーション `rpc.nisd`、`sendmail`、`automountd` など再度起動する必要があります。
- DES 暗号化に失敗しました。ご購入先にご連絡ください。

`_svcauth_des: no public key for principalname`

サーバーがクライアントの公開鍵を取得できません。考えられる原因は次のとおりです。

- 主体に公開鍵がありません。主体のホームドメインにある cred テーブルを対象に、`niscat` を実行します。テーブルの中で、この主体に DES の資格が割り当てられていない場合は、`nisaddcred` を使用して資格を作成し、その主体で `keylogin` を実行します。
- `nsswitch.conf` ファイルで指定されたネームサービスが応答しません。

`_svcauth_des: replayed credential from principalname`

サーバーは要求を受け取り、キャッシュの中でそのクライアント名に一致するエントリを見つけましたが、受け取った要求の会話鍵のタイムスタンプが、キャッシュの中に格納されているタイムスタンプより古いものでした。

このメッセージの重大度は、有効にしているセキュリティのレベルに依存します。セキュリティレベルが低い場合は、このメッセージは主に情報を示すために表示されます。レベルが高い場合は、次の説明に従って問題を解決するための操作を行なってください。

考えられる原因は次のとおりです。

- クライアントとサーバーそれぞれのクロックが同期していません。`rdate` を使用して、クライアントのクロックをサーバーのクロックに同期させます。
- サーバーがランダムな順序で要求を受け取っています。マルチスレッドアプリケーションを使用している場合に、これが発生します。アプリケーションが `tcp` をサポートしている場合、`/etc/netconfig` (または `NETPATH` 環境変数) を `tcp` に設定します。

`_svcauth_des: timestamp is earlier than the one previously seen from principalname`

クライアントからいくつかのコールを受け取りましたが、後から受け取ったコールが、最初のコールより古いタイムスタンプを示しています。このメッセージの重大度は、有効にしているセキュリティのレベルに依存します。セキュリティレベルが低い場合は、このメッセージは主に情報を示すために表示されます。レベルが高い場合は、次の説明に従って問題を解決するための操作を行なってください。

考えられる原因は次のとおりです。

- クライアントとサーバーそれぞれのクロックが同期していません。rdate を使用して、クライアントのクロックをサーバーのクロックに同期させます。
- サーバーのキャッシュから、このクライアント用のエントリが削除されました。サーバーはキャッシュの中に、現在のクライアントの情報を維持しています。キャッシュのサイズは、64 個のクライアントに相当します。

`_svcauth_des: timestamp expired for principalname`

ウィンドウを受信する際、デフォルトでは 35 秒以内の間隔が規定されていますが、クライアントから受け取ったタイムスタンプが、この範囲に収まっていません。このメッセージの重大度は、有効にしているセキュリティのレベルに依存します。セキュリティレベルが低い場合は、このメッセージは主に情報を示すために表示されます。レベルが高い場合は、次の説明に従って問題を解決するための操作を行ってください。

考えられる原因は次のとおりです。

- サーバーやネットワークが低速なので、35 秒のウィンドウでは間隔が不足しています。
- クライアントとサーバーそれぞれのクロックがかなりずれているため、ウィンドウの間隔に収まりません。rdate を使用して、クライアントのクロックをサーバーのクロックに同期させます。
- サーバーのキャッシュから、このクライアント用のエントリが削除されました。操作をもう一度実行します。

`Too Many Attributes`

サーバーに渡された検索基準が、検索可能なテーブルの列より多くの属性を持っています。

このメッセージは、NIS+ のエラーコード定数 `NIS_TOOMANYATTRS` によって生成されます。詳細な情報は、`nis_tables(3N)` のマニュアルページを参照してください。

`Too many failures - try later`

`Too many tries; try again later`

このメッセージは、「ログイン、またはパスワードの変更のやり直しの回数が多すぎる、あるいは時間がかかりすぎる」ということを意味します。詳細は、[292 ページの「Login incorrect メッセージ」](#)または、[294 ページの「パスワード変更の失敗」](#)を参照してください。

Unable to authenticate NIS+ client

サーバーがクライアントのコールバック要求を実行したときに、RPC の `clnt_call()` から `RPC_AUTHERR` という状態を受け取ると、このメッセージが生成されます。通常これは、古い認証情報が原因となっています。システムが、更新されていないキャッシュから取り出したデータを使用した場合や、認証情報が最近変更され、その変更結果がサーバーに伝わっていない場合に、このような古い認証情報が発生します。ほとんどの場合、この問題は短時間のうちに自動的に解決されます。

この問題が自動的に解決されない場合は、次のいずれかの問題があると考えられます。

- `/var/nis/NIS_SHARED_DIRCACHE` ファイルが壊れています。NIS+ サービスを停止し、`/var/nis/NIS_SHARED_DIRCACHE` ファイルを削除して、NIS+ サービスを再起動します。
- `/var/nis/NIS_COLD_START` ファイルが壊れています。このファイルを削除し、`nisinit` を実行して、このファイルを再度作成します。
- `/etc/.rootkey` ファイルが壊れています。`keylogin -r` を実行します。

このメッセージは、NIS+ のエラーコード定数 `NIS_CLNTAUTH` によって生成されます。

Unable to authenticate NIS+ server

ほとんどの場合、システムが容易に修正することのできる、ソフトウェアの小さなエラーが発生したことを意味しています。サーバーがRPCの `clnt_call` から `RPC_AUTHERR` という状態を受け取ると、このメッセージが生成されます。

この問題が自動的に解決されない場合は、`/var/nis/NIS_COLD_START` ファイル、`/var/nis/NIS_SHARED_DIRCACHE` ファイル、`/etc/.rootkey` ファイルのどれかが壊れていることが考えられます。

このメッセージは、NIS+ のエラーコード定数 `NIS_SRVAUTH` によって生成されます。

Unable to bind to master server for name '*string*'

これらの問題については、『Solaris のシステム管理 (ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。この特別なメッセージは、`/etc/defaultdomain` ファイルの中にあるサーバーのドメイン名の右端にピリオドを追加したときに発生することがあります。

Unable to create callback.

サーバーは、ユーザーのマシンのコールバックサービスと通信できませんでした。この結果、データを返すことができませんでした。

詳細な情報は、`nis_tables(3N)` のマニュアルページを参照してください。

Unable to create process on server

NIS+ のサービスルーチンが、サポートしていない手続き番号を指定する要求を受け取ると、このエラーが生成されます。

このメッセージは、NIS+ のエラーコード定数 `NIS_NOPROC` によって生成されます。

`string: Unable to decrypt secret key for string.`
考えられる原因は次のとおりです。

- 正しくないパスワードを入力しました。
- `cred` テーブルの中に、`name` のエントリがありません。
- NIS+ が鍵を復号化できませんでした。おそらく、エントリが壊れている可能性があります。
- `nsswitch.conf` ファイルが、`cred` テーブルの中に記録されている NIS+ パスワードとは異なるパスワードを使用して、`/etc/passwd` ファイルの中に格納されているローカルパスワードの照会を行なっています。

これらの問題を診断および解決する方法については、『*Solaris* のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

Unknown error

NIS+ のエラー処理ルーチンが、未登録のタイプのエラーを受け取ったときに、このメッセージが表示されます。

Unknown object

返されたオブジェクトのタイプが、未登録のタイプです。

このメッセージは、NIS+ のエラーコード定数 `NIS_UNKNOWNOBJ` によって生成されます。詳細な情報は、`nis_names` のマニュアルページを参照してください。

`update_directory: mmmn objects still running.`

複製を更新するときに、サーバーがディレクトリを更新している間、表示されるメッセージです。なんらかの操作を行う必要はありません。

User *principalname* needs Secure RPC credentials to login but has none.

キーログインが成功しなかったことを示します。この問題は一般に、`/etc/shadow` とリモートの NIS+ パスワードテーブルとで記憶しているパスワードが異なっているために発生します。

Warning: couldn't reencrypt secret key for *principalname*

この問題の原因としてもっとも可能性が高いのは「Secure RPC パスワードがログインパスワードと異なっていて (またはローカルの `/etc/shadow` とリモートの NIS+ パスワードテーブルとで記憶しているパスワードが異なっていて)、まだ明示的に `keylogin` を実行していない」ということです。この種類の問題の詳細については、『*Solaris* のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

WARNING: db::checkpoint: could not dump database: No such file or directory

このメッセージは、チェックポイント実行の際に、システムがデータベースファイルをオープンできなかったことを示しています。考えられる原因は次のとおりです。

- データベースファイルが削除されています。
- サーバーがファイル記述子を使い果たしました。
- ディスクに問題があります。
- ユーザーまたはホストに、必要なアクセス権が割り当てられていません。

WARNING: db_dictionary::add_table: could not initialize database from scheme

データベーステーブルを初期設定できませんでした。考えられる原因は次のとおりです。

- システムリソースに問題があります。『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。
- コマンド構文での新しいテーブルの指定が正しくありませんでした。
- データベースが壊れています。

WARNING: db_query::db_query:bad index

ほとんどの場合、このメッセージは、インデックス付きの名前の指定が正しくないことを示しています。指定されたテーブルの中に、インデックス付きの名前が存在するかどうか確認します。コマンドのスペルと構文にエラーがないかチェックします。

**WARNING: domain *domainname* already exists.

このメッセージは、作成しようとしたドメインがすでに存在することを意味しています。

- ルート以外のマスターサーバーを新しく変換しようとしている場合や、直前に起こった nisserver の問題を回復しようとしている場合は、このスクリプトを続けて実行します。
- *domainname* のスペルが正しくない場合は、正しいドメイン名を指定して、このスクリプトをもう一度実行します。

**WARNING: failed to add new member *NIS+_principle* into the *groupname* group. You will need to add this member manually: 1.

/usr/sbin/nisgrpadm -a groupname *NIS+_principal*

NIS+ のコマンド nisgrpadm が、NIS+ のグループ *groupname* に新しいメンバーを追加することに失敗しました。次のように入力して、NIS+ のこの主体を追加します。

/usr/sbin/nisgrpadm -a groupname *NIS+_principal*

****WARNING:** failed to populate *tablename* table.
nisaddent コマンドは、NIS+ の *tablename* テーブルをロードできませんでした。通常、この警告メッセージの前に、詳細なエラーメッセージが表示されます。

****WARNING:** hostname specified will not be used. It will use the local hostname instead.

このメッセージは、-H オプションを指定して、リモートホスト名を入力したことを示します。nisserver -r スクリプトを使って、ルートマスターサーバーと同様の方法でリモートマシンを構成できません。

- ローカルマシンを NIS+ のルートマスターサーバーに変換したい場合は、別に操作を行う必要はありません。nisserver -r スクリプトは、入力されたホスト名を無視します。
- リモートホスト (ローカルマシンではなく) を NIS+ のルートマスターサーバーに変換したい場合は、このスクリプトを終了します。リモートホストを対象にして、nisserver -r を再度実行します。

****WARNING:** *hostname* is already a server for this domain. If you choose to continue with the script, it will try to replicate the *groups_dir* and *org_dir* directories for this domain.

このメッセージは、複製サーバーを作成しようとしたドメインの中で、すでに *hostname* が複製サーバーとなっていることを警告します。

- nisserver によって以前に発生した問題を解決するためにこのスクリプトを実行している場合は、スクリプトを続けて実行します。
- *hostname* を間違って入力した場合は、正しいホスト名を指定して、このスクリプトをもう一度実行します。

****WARNING:** *alias-hostname* is an alias name for host *canonical_hostname*.
You cannot create credential for host alias.

このメッセージは、nisclient -c の名前リストに、ホストの別名を入力したことを示します。ホストの別名に資格を作成すべきではないので、標準的なホスト名にその資格を作成するかどうか、スクリプトが尋ねています。

****WARNING:** file *directory-path/tablename* does not exist! *tablename* table will not be loaded.

このスクリプトは、*tablename* の入力ファイルを見つけることができませんでした。

- *directory-path/tablename* が間違って入力されている場合は、正しいテーブル名を指定して、このスクリプトをもう一度実行します。
- *directory-path/tablename* ファイルが存在しない場合は、適切なデータを使ってファイルを作成し更新します。次に、このスクリプトを再度実行して、このテーブルを生成します。

```

**WARNING: NIS auto.master map conversion failed. auto.master
table will not be loaded.
    auto.master テーブル内のすべてのピリオドを下線に変換しているときに、
    auto.master マップの変換に失敗しました。別の NIS サーバーを使って、この
    スクリプトをもう一度実行します。

**WARNING: NIS netgroup map conversion failed. netgroup table
will not be loaded.
    netgroup マップの中で NIS のドメイン名を NIS+ のドメイン名に変換している
    ときに、netgroup マップの変換に失敗しました。別の NIS サーバーを使って、
    このスクリプトをもう一度実行します。

**WARNING: nisupdkeys failed on directory domainname. This script
will not be able to continue. Please remove the domainname
directory using 'nisrmdir'.
    NIS+ のコマンド nisupdkeys は、リスト指定されたディレクトリオブジェクト
    内の鍵を更新することに失敗しました。新しいドメインにサービスを提供すること
    になっている新しいマスターサーバー上で rpc.nisd が動作していない場合は、
    rpc.nisd を再度起動します。次に、nisrmdir を使用して、domainname ディレ
    クトリを削除します。最後に、nisserver を再度実行します。

WARNING: nisupdkeys failed on directory directory-name You will need
to run nisupdkeys manually: 1. /usr/lib/nis/nisupdkeys directory-name
    NIS+ のコマンド nisupdkeys は、リスト指定されたディレクトリオブジェクト
    内の鍵を更新することに失敗しました。次のように入力して、ディレクトリオブ
    ジェクト内の鍵を手作業で更新します。

    # /usr/lib/nis/nisupdkeys directory-name

**WARNING: once this script is executed, you will not be able to
restore the existing NIS+ server environment. However, you can
restore your NIS+ client environment using "nisclient -r" with the
proper domainname and server information. Use "nisclient -r" to
restore your NIS+ client environment.
    これらのメッセージは、このスクリプトを以前に実行して NIS+ サーバーをすでに
    設定している場合に表示されます。このスクリプトの実行を続行すると、必要に応
    じて NIS+ 関連のファイルが削除および再作成されることを示しています。

■ NIS+ のファイルを削除してもよい場合は、このスクリプトを続けて実行しま
    します。

■ NIS+ のファイルを保存するには、Do you want to continue? プロンプト
    で n と入力し、このスクリプトを終了します。次に、NIS+ のファイルを別の
    ディレクトリに保存し、このスクリプトをもう一度実行します。

```

****WARNING:** this script removes directories and files related to NIS+ under /var/nis directory with the exception of the NIS_COLD_START and NIS_SHARED_DIRCACHE files which will be renamed to <file?.no_nisplus. If you want to save these files, you should abort from this script now to save these files first.

すぐ上の「WARNING: once this script is executed,...」のメッセージを参照してください。

****WARNING:** you must specify the NIS domainname.

このメッセージは、プロンプトで NIS のドメイン名を入力しなかったことを示しています。プロンプトで、NIS サーバーのドメイン名を入力します。

****WARNING:** you must specify the NIS server *hostname*. Please try again.

このメッセージは、プロンプトで NIS のサーバーのホスト名を入力しなかったことを示しています。プロンプトで、NIS サーバーのホスト名を入力します。

Window verifier mismatch

これは、_svcauth_des コードによって生成されたデバッグ用メッセージです。鍵がキャッシュから削除されたため、ペリファイアが無効になっている可能性があります。このメッセージが生成されると、_svcauth_des は、AUTH_BADCRED の状態を返します。

You (*string*) do not have Secure RPC credentials in NIS+ domain '*string*'

コマンドを実行するために必要とされる資格が割り当てられていないサーバーで nispasswd を実行すると、このメッセージが生成されることがあります。セキュリティレベル 0 で動作しているサーバーは、資格の作成や維持を行わないことに注意してください。

資格、所有権、およびアクセス権の詳細については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

You may not change this password

パスワードの変更が管理者によって禁止されていることを意味します。

You may not use nisplus repository

「コマンド行で -r nisplus が使用されたが、NIS+ パスワードテーブルに適切なエントリがない」ということを意味します。パスワードテーブルに必要なエントリがあるかどうか確認してください。また nsswitch.conf ファイルに nisplus というエントリを追加してみてください。

Your password has been expired for too long

Your password is expired

上記の 2 つのメッセージは、パスワードの有効期間に関するものです。パスワードの有効期間を過ぎているため、すぐに変更する必要があることを示しています。詳細は、292 ページの「will expire メッセージ」を参照してください。

Your password will expire in *nn* days

Your password will expire within 24 hours

上記の2つのメッセージは、パスワードの有効期間に関するものです。パスワードがまもなく期限切れになるため、すぐに変更する必要があることを示しています。詳細は、[292 ページの「will expire メッセージ」](#)を参照してください。

Your specified repository is not defined in the nsswitch file!

「-r オプションでリポジトリのパスワード情報を更新しようとしたが、該当するパスワードのリポジトリが nsswitch.conf ファイルのパスワードエントリにない」という意味の警告メッセージです。使用したコマンド自体は機能し、-r フラグで指定したリポジトリへの変更はすべて行われます。しかし変更の対象となっているリポジトリが nsswitch.conf ファイルで指定されているものではないため、該当するリポジトリを nsswitch.conf ファイルで指定するまでこの変更の影響はまったく表れません。

たとえば、このスイッチファイルの passwd エントリに、files nis と指定されているとします。このファイルに対して

```
passwd -r nisplus
```

を使用してパスワードの有効期限を設定しようとしても、無視されます。スイッチファイルの設定は、files nis のままです。

verify_table_exists: cannot create table for string nis_perror message.

テーブルを対象にして操作を行う場合、NIS+ は最初にテーブルが存在するかどうかを確認します。テーブルが存在しない場合は、NIS+ はそのテーブルを作成しようとしています。作成できない場合は、このエラーメッセージを返します。string は、検索や作成ができなかったテーブルを示します。nis_perror message の部分は、問題の原因に関する情報を示します。メッセージのこの部分を、独立したメッセージと考えて、この付録にある説明を参照できます。このタイプの問題の考えられる原因は次のとおりです。

- サーバーはディレクトリの複製として追加されるだけであり、ディレクトリオブジェクトを持つことはできません。nisping -c により、チェックポイントを実行します。
- ディスク容量を使い果たしました。詳細については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。
- データベースが壊れています。
- その他のタイプのソフトウェアエラー。ご購入先にご連絡ください。

yppcat: can't bind to NIS server for domain *domainname*. Reason: can't communicate with ypbind.

『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』の「NIS の障害追跡」を参照してください。

yppoll: can't get any map parameter.

『Solaris のシステム管理 (ネーミングとディレクトリサービス : *DNS*、*NIS*、*LDAP* 編)』の「NIS の障害追跡」を参照してください。

付録 B

本書『Solaris のシステム管理 (ネーミングとディレクトリサービス : NIS+ 編)』の更新情報

Solaris 10

このリリースには、NIS+ に関連した新しい機能は特にありません。

FNS

FNS は、Solaris でサポートされなくなりました。FNS の関連情報は、Solaris 10 System Administrator Collection から削除されました。

サービス管理機能 (SMF)

現時点では、NIS+ はサービス管理機能 (SMF) によって管理されます。NIS+ サービスを起動、停止、または再起動するときは、svcadm コマンドを使用します。SMF によって管理される NIS+ プロセスには、rpc.nisd と keyserve があります。nis_cachemgr は、rpc.nisd サービスに連動して起動および停止します。これらのサービスが個別に起動および停止することはなくなりました。

SMF を使用すると、NIS+ でのオプションの動作の実装にも影響します。SMF を NIS+ で使用する方法については、[89 ページの「NIS+ とサービス管理機能」](#)を参照してください。

用語集

data encryption standard (DES) (データ暗号化規格)	アメリカ商務省標準局によって開発された、データの暗号化と復号化のために一般的に使用される高度なアルゴリズム。「SUN-DES-1」の項も参照。
DES	「data encryption standard (DES) (データ暗号化規格)」の項を参照。
DNS	「ドメインネームシステム (DNS)」の項を参照。
DNS ゾーン	ネットワークドメイン内の管理境界であり、1 つまたは複数のサブドメインから構成されるのが普通。
DNS ゾーンファイル	DNS ソフトウェアがドメイン内の全ワークステーションの名前と IP アドレスを格納する一連のマップ。
DNS 転送	NIS サーバーまたは NIS 互換設定の NIS+ サーバーは、自分で応答できない要求を DNS サーバーに転送する。
GID	「グループ ID」の項を参照。
IP	インターネットプロトコル。インターネットプロトコル体系の「ネットワーク層」プロトコル。
IP アドレス	ネットワーク内の各ホストを識別する一意な番号。
MIS (management information system or service)	経営情報システム
NIS	ネットワーク上のシステムとユーザーについての重要な情報が収められている分散型ネットワーク情報サービス。NIS データベースは、「マスターサーバー」とすべての「スレーブサーバー」に格納されている。
NIS+	ネットワーク上のシステムとユーザーについての階層情報が収められている分散型ネットワーク情報サービス。NIS+ データベースは、「マスターサーバー」とすべての「複製サーバー」に格納されている。

NIS+ オブジェクト	NIS+ のドメイン、ディレクトリ、テーブル、グループのこと。「ドメイン」、「ディレクトリ」、「グループ」、「テーブル」の項を参照。
NIS+ 環境	管理上使用する用語。「該当する <code>nsswitch.conf</code> ファイルで、 <code>nisplus</code> が情報源として指定されている」という状況を指す。また「コマンドが、NIS+ 名前空間のオブジェクトを操作するようにオプションを付けて実行されている」という状況 (例: <code>passwd -r nisplus</code>) を指すこともある。
NIS+ 主体	「主体」の項を参照。
NIS+ トランザクションログ	名前空間内のオブジェクトに関する NIS+ テーブル宛のデータ更新が収められたファイル。名前空間内の変更は、複製サーバーに伝達されるまでは、トランザクションログに格納される。トランザクションログは、マスターサーバーの複製サーバーがすべて更新された後で、クリアされる。
NIS 互換モード	NIS+ の構成の 1 つであり、このモードでは、NIS クライアントは NIS+ テーブルに格納されたデータにアクセスできる。また、NIS+ サーバーは NIS クライアントと NIS+ クライアントからの情報要求に応答できる。
NIS マップ	NIS によって使用されるファイルであり、たとえば、ネットワーク上の全ユーザーのパスワードエントリや、ネットワーク上の全ホストマシンの名前など、特定種類の情報を格納する。NIS サービスの一部であるプログラムは、これらのマップを参照する。「NIS」の項を参照。
NNSP (Next naming system pointer)	「次のネーミングシステム参照」の項を参照。
pinging	NIS+ データに加えられた変更の内容が、NIS+ マスターサーバーから当該ドメインの複製サーバーに送られるプロセス。
RPC	遠隔手続き呼び出し (RPC) を参照。
Secure RPC パスワード	Secure RPC プロトコルにおいて必要となるパスワードのこと。非公開鍵の暗号化に使用される。このパスワードはユーザーのログインパスワードと同じでなければならない。
TCP	「Transport Control Protocol (TCP)」の項を参照。
TCP/IP	Transport Control Protocol/Internet Protocol の頭文字。このプロトコル体系は、最初はインターネット用に開発された。「インターネット」プロトコル体系とも呼ばれる。Solaris 8 リリース ネットワークは、デフォルトでは TCP/IP 上で動作する。
Transport Control Protocol (TCP)	インターネットプロトコル体系での主要な転送プロトコルであり、高信頼性でコネクション型の全二重ストリームを提供する。配送には IP を使用する。「TCP/IP」の項を参照。

X.500	開放型システム間相互接続 (OSI) 規格で定義されたグローバルディレクトリサービス。
アクセス権	NIS+ 主体 (principal) のクラスに割り当てられるアクセス権であり、NIS+ オブジェクト上でこれらのクラスが実行できる動作を決定する。読み取り、変更、作成、または 削除の 4 種類がある。
アプリケーションレベルの ネームサービス	ファイル、メール、印刷などのサービスを提供するアプリケーションに組み込まれているネームサービスのこと。アプリケーションレベルのネームサービスは、エンタープライズレベルネームサービスの下に位置する。エンタープライズレベルのネームサービスが提供するコンテキストの中に、アプリケーションレベルネームサービスのコンテキストを組み込むことができる。
暗号化鍵	「データ暗号化鍵」の項を参照。
インターネット	一連のルーターによって相互接続された世界規模のネットワークの集まり。これらのネットワークは 1 つの大きな仮想ネットワークとして機能できる。
インターネットアドレス	「TCP/IP」を使用するホストに割り当てられた 32 ビットのアドレス。「ドット表記」の項を参照。
インデックス付き名前	テーブル内のエントリの識別に使用されるネーミング形式。
遠隔手続き呼び出し (RPC)	分散コンピューティングのクライアントサーバーモデルを実現する、簡単で一般的なパラダイム。与えられた引数を使用することによって、要求がリモートシステムに送信され、指定された手順が実行される。そして、その結果が呼び出し側に返される。
エンタープライズレベル のネームサービス	エンタープライズレベルネットワーク内のマシン (ホスト) 名、ユーザー名、ファイル名を管理するサービスのこと。
エンタープライズレベル のネットワーク	「エンタープライズレベル」のネットワークといっても、ケーブルや赤外線ビーム、無線などを利用した LAN (Local Area Network) の場合もあれば、ケーブルや直通電話接続を利用して複数の LAN を結んだものもある。エンタープライズレベルのネットワーク内では、DNS や X.500/LDAP などのグローバルネームサービスを使用せずに、どのマシンからでも任意のマシンにアクセスできる。
エントリ	データベーステーブルの中の、一列のデータのこと。
親ドメイン	「ドメイン」の項を参照。
鍵 (暗号化)	鍵の管理および配布システムの一部として、ほかの鍵の暗号化と復号化に使用される鍵。「データ暗号化鍵」の項も参照。
キーサーバー	非公開鍵を格納する Solaris システムのプロセス。
キー (列)	NIS+ テーブルエントリのデータは、そのテーブルのキーとは無関係に、どの列からでもアクセスできる。
逆解決	DNS ソフトウェアを使用して、ワークステーションの IP アドレスをワークステーション名に変換するプロセス。

キャッシュマネージャ	NIS+ クライアントのローカルキャッシュ (NIS_SHARED_DIRCACHE) を管理するプログラム。これらのクライアントによってもっとも頻繁に使用されるディレクトリをサポートする NIS+ サーバーについての位置情報 (トランスポートアドレス、認証情報、生存期間など) を格納するために使用される。
クライアント	(1) NIS+ では、NIS+ サーバーに対して NIS+ サービスをリクエストする主体 (マシンまたはユーザー) がクライアント。 (2) ファイルシステムのクライアントサーバーモデルでは、計算パワーや大きな記憶容量などの計算サーバーの資源にリモートアクセスするマシンがクライアント。 (3) クライアントサーバーモデルでは、「サーバープロセス」からサービスにアクセスする「アプリケーション」がクライアント。このモデルでは、クライアントとサーバーは同じマシン上または別のマシン上で動作可能。
クライアントサーバーモデル	ネットワークサービスおよびこれらのモデルユーザープロセス (プログラム) を説明する一般的な方法の 1 つ。これらの例としては、「ドメイン名システム (DNS)」のネームサーバー / ネームリゾルバパラダイム、および「NFS」やディスクレスホストなどのファイルサーバー / ファイルクライアント関係がある。「クライアント」の項も参照。
グループ	(1) 共通の名前で参照されるユーザーの集り。 (2) NIS+ では、NIS+ オブジェクトに共通のアクセス権を与えられたユーザーの集まり。NIS+ グループの情報は、NIS+ グループテーブルに格納されている。 (3) UNIX では、ユーザーのファイルへのアクセスを決定する。デフォルトユーザーグループと標準ユーザーグループの 2 つがある。
グループ ID	ユーザーのデフォルト「グループ」を識別する番号。
グローバルネームサービス	電話回線、衛星回線、その他の通信システムにより連結された世界中のエンタープライズレベルネットワークの名前を管理するサービスのこと。この世界中のネットワークの集合体がいわゆる「インターネット」である。グローバルネームサービスでは、ネットワーク名だけでなく、任意のネットワーク内の個々のマシンやユーザーも識別できる。
広域ネットワーク (WAN)	地理的に離れた複数の LAN (Local Area Network) またはシステムを、電話回線、光ファイバ、衛星などを使用して接続したネットワークのこと。
公開鍵	数学的に生成された 1 対の番号の公開構成要素であり、非公開鍵と組み合わせれば DES 鍵が生成される。この DES 鍵を使用すれば、情報の暗号化と復号化を行える。公開鍵は、すべてのユーザーとマシンが使用できる。どのユーザーやマシンにも、固有の公開鍵と非公開鍵が 1 対ある。

コールドスタートファイル	クライアントが初期設定されたときにクライアントに与えられる NIS+ ファイル。クライアントがホームドメイン内のマスターサーバーと通信を開始できるだけの情報が収められている。
子ドメイン	「ドメイン」の項を参照。
サーバー	(1) NIS+、NIS、DNS では、ネットワークにネームサービスを提供するホストマシンのこと。 (2) ファイルシステムの「クライアントサーバーモデル」では、サーバーとは計算資源 (計算サーバーとも呼ばれる) と大きな記憶容量を備えたマシン。クライアントマシンはリモートアクセスが可能であり、これらの資源を使用できる。ウィンドウシステムのクライアントサーバーモデルでは、サーバーとはアプリケーションまたは「クライアントプロセス」にウィンドウサービスを提供するプロセス。このモデルでは、クライアントとサーバーは同じマシン上または別のマシン上で動作可能。 (3) ファイルの提供を実際に処理する「デーモン」。
サーバーリスト	「優先サーバーリスト」の項を参照。
サブネット	ルーティングを簡単にするため、1 つの論理ネットワークを小さな物理ネットワークに分割する方式。
資格	NIS+ 主体 (principal) についての認証情報。クライアントソフトウェアは、各要求と一緒にこの情報を NIS+ サーバーに送信する。この情報によって、ユーザーまたはマシンの確認検査が行われる。
識別名	X.500 ディレクトリ情報ベース (DIB) のエントリ。ルートから指定エントリまでのパスに沿ったツリーの各エントリから選択した属性で構成される。
主体	NIS+ 情報のユーザーのうち、名前空間に資格が格納されているものを指す。また NIS+ サーバーに対してなんらかのリクエストを行うユーザー、マシンを指す。クライアントユーザー、クライアントマシンの 2 種類がある。 ■ 「ルート主体」マシンの root ユーザー (ユーザー ID = 0) のこと。DES 資格だけ必要。 ■ 「ユーザー主体」root ユーザーを除くすべてのユーザー (ユーザー ID > 0)。ローカル資格、および DES 資格が必要。
スレーブサーバー	(1) ネットワーク情報サービス (NIS) データベースのコピーを管理するサーバーシステム。このシステムには、ディスクとオペレーティングシステムの完全なコピーがある。 (2) スレーブサーバーは、NIS+ では「複製サーバー」と呼ばれる。

チェックポイント設定	NIS+ データセットの変更 (サーバーのメモリーに格納され、トランザクションログに記録される) をディスク上の NIS+ テーブルに書き込む一連のプロセス。つまり、NIS+ データセットの変更を反映して NIS+ テーブルを更新すること。
ディレクトリ	(1) NIS+ においては、NIS+ オブジェクトのコンテナのこと。NIS+ テーブル、グループ、サブディレクトリなどを指す。 (2) UNIX においては、ファイルまたはサブディレクトリのコンテナのこと。
ディレクトリキャッシュ	ディレクトリオブジェクトに関連付けられたデータの格納に使用されるローカルファイル。
データ暗号化鍵	暗号化を行うプログラムに使用されるデータを暗号化および復号化するための鍵。「鍵 (暗号化)」の項も参照。
テーブル	NIS+ においては、NIS+ データを行および列の中に持つ 2 次元的な (リレーショナルでない) データベースオブジェクトのこと (NIS における「NIS マップ」は、「列を 2 つ持つ NIS+ テーブルに似ている」)。NIS+ データは、テーブルの形で保存される。(NIS における「NIS マップ」は、「列を 2 つ持つ NIS+ テーブルに似ている」)。NIS+ データは、テーブルの形で保存される。NIS+ では、定義済み (システム) テーブルが 16 個提供される。各テーブルにはそれぞれ異なる情報が格納されています。
テーブルの生成 (populate)	ファイルまたは NIS マップから NIS+ テーブルにデータを入れること。
ドット表記	32 ビット整数用の構文表現であり、10 進表記された 4 つの 8 ビット数が小数点 (ドット) で区切って表現される。172.27.67.20 のように、インターネットでの IP アドレスを表現するために使用される。
ドメイン	(1) NIS+ では、NIS+ によって管理されるオブジェクト (階層構造になっている) のグループ。最上位のドメイン (ルートドメイン) 1 つと、サブドメイン 0 個以上からなる。ドメインおよびサブドメインは、地理的、組織的、機能的な基準によって編成される。 ■ 「親ドメイン」階層構造の中で、現在のドメインのすぐ上のドメインを表す相対的な名称。 ■ 「子ドメイン」階層構造の中で、現在のドメインのすぐ下のドメインを表す相対的な名称。 ■ 「ルートドメイン」現在の NIS+ 階層の最上位のドメイン。 (2) インターネットではネーミング階層の一部で、通常、Local Area Network (LAN)、Wide Area Network (WAN)、またはその一部に相当する。構文上、インターネットドメイン名は小数点 (ドット) によって区切られた一連の名前 (ラベル) から構成される。たとえば、<code>sales.doc.com</code>。

	(3) ISO の開放型システム間相互接続 (OSI) では、「ドメイン」は、MHS プライベート管理ドメイン (PRMD) やディレクトリ管理ドメイン (DMD) などのように、複雑な分散システムの管理パーティションとして使用されるのが普通。
ドメイン名	ローカルネットワーク上のシステムグループに割り当てられた名前であり、管理ファイルを共有する。ネットワーク情報サービスのデータベースが正常に動作するためにはドメイン名が必要。「ドメイン」の項を参照。
ドメイン名サービス (DNS)	インターネットで使用されるネットワーク情報サービスのこと。すなわち DNS は、ドメイン名とマシン名をインターネットなどのエンタープライズ外部のアドレスにマッピングする場合のネーミングポリシーとメカニズムを提供する。
名前解決	ワークステーションやユーザーの名前をアドレスに変換するプロセス。
名前空間	(1) 名前空間はユーザー、ワークステーション、そしてアプリケーションがネットワーク上で必ず必要とする情報を格納している。 (2) ネーミングシステムで使用される名前セット。 (3) NIS+ 名前空間 - NIS+ ソフトウェアによって使用される階層型ネットワーク情報の集まり。 (4) NIS 名前空間 - NIS ソフトウェアによって使用される非階層型ネットワーク情報の集まり。 (5) DNS 名前空間 - DNS ソフトウェアを使用するネットワークワークステーションの集まり。
認証	ある NIS+ サーバーが、NIS+ 名前空間に対するアクセス要求の送信者を識別できるかどうかの判定。認証された要求は、所有者、グループ、およびその他という承認カテゴリに分類される。送信者が識別できない未認証要求は、未承認カテゴリに入れられる。
ネームサーバー	1 つ以上のネットワークネームサービスを実行するサーバー。
ネームサービス	マシン、ユーザー、プリンタ、ドメイン、ルーターなどの、ネットワーク上の名前とアドレスを管理するネットワークサービスのこと。
ネームサービススイッチ	NIS+ クライアントがそのネットワーク情報を獲得できるソースを定義する構成ファイル (/etc/nsswitch.conf)。
ネットワークパスワード	「Secure RPC パスワード」を参照。
ネットワークマスク	ソフトウェアが、ローカルサブネットアドレスをそれ以外のインターネットプロトコルアドレスから分離するために使用する番号。

非公開鍵	数学的に生成された 1 対の番号の非公開構成要素であり、公開鍵と組み合わせれば DES 鍵が生成される。この DES 鍵を使用すれば、情報の暗号化と復号化を行える。送信側の公開鍵は、その鍵の所有者だけが使用できる。どのユーザーやマシンにも、固有の公開鍵と非公開鍵が 1 対ある。
複製サーバー	NIS+ サーバーのうち、ドメインのマスター NIS+ サーバーデータベースの複製を持つもの。複製サーバーは、NIS+ サーバーソフトウェアを実行し、NIS+ テーブルのコピーを管理する。複製サーバーによって、NIS+ サービスの可用性が向上する。NIS+ ドメインには個々に 1 つ以上の複製サーバーが必要である（「複製サーバー」は、NIS 名前空間では「スレーブサーバー」と呼ばれる）。
マスターサーバー	ドメイン内の NIS データベースのマスターコピーを保持しているサーバーのこと。名前空間に対する変更は、必ずマスターサーバーのデータベース上で行う。ドメイン中に複数のマスターサーバーを作成できない。
メール交換レコード	DNS ドメイン名、およびこれらに対応するメールホストのリストが収められているファイル。
メールホスト	サイトの電子メールのルーターおよびレシーバとして機能するワークステーション。
優先サーバー	クライアントマシンから見て、優先 NIS+ サーバーとは、そのクライアントが名前空間に関する情報を得る際に、ほかのサーバーより先に照会を試みるサーバーのことをいう。あるクライアントまたはドメインの優先サーバーリストにあるサーバーは、そのクライアントまたはドメインの優先サーバーであると考えられる。
優先サーバーリスト	client_info テーブルまたは client_info ファイルのこと。優先サーバーリストには、あるクライアントマシンまたはドメインから見た優先サーバーが指定される。
優先順位の番号	クライアントマシンが名前空間に関する情報を取得する際に、どの NIS+ サーバーから順に照会するかを示す番号。マシンは、ある優先順位番号を持つすべてのサーバーに要求を送ったあと、その次に高いランク番号を持つサーバーに要求を送る。たとえば、ランク番号 0 の NIS+ サーバーに要求を送った後、ランク番号 1 のサーバーに要求を送る。
ルートドメイン	「ドメイン」の項を参照。
ルート複製サーバー	NIS+ サーバーのうち、ルートドメインのマスター NIS+ サーバーデータベースの複製を持つものを指す。
ルートマスターサーバー	NIS+ ルートドメインのマスターサーバーのこと。
レコード	「エントリ」の項を参照。
ローカルエリアネットワーク (LAN)	1 つの地理的なサイトの中にある複数のシステムを、データやソフトウェアの共有、交換の目的で接続したもの。

索引

数字・記号

+/- 構文

compat, 39, 40

passwd_compat, 39

\$NIS_DEFAULTS, 282

NP, 367

A

aliases ファイル, 101, 122

API, NIS+, 54

attempt to remove a non-empty table
メッセージ (NIS+), 438

Authentication denied メッセージ (NIS+)
, 447

Authentication error メッセージ (NIS+)
, 447

auto_home テーブル, 419
nsswitch.conf ファイル, 33
列, 419

auto_home マップ, 420

auto_man マップ, 420

auto_master テーブル, 419-420, 420
nsswitch.conf ファイル, 33
オートマウントマップの追加, 207
列, 420

auto_master マップ, 420

auto_programs マップ, 420

B

backup_list ファイル, 404

bootparams テーブル, 420-421

入力ファイルのフォーマット, 421

Busy try again later メッセージ (NIS+)
, 456

C

Callback: - select failed メッセージ
(NIS+), 438

CALLBACK_SVC: bad argument メッセージ
(NIS+), 438

Cannot [do something] with log type メッセージ (NIS+), 460

Cannot find メッセージ (NIS+), 441

Cannot get public key メッセージ (NIS+)
, 447

Cannot remove replica メッセージ (NIS+)
, 438

Can't find suitable transport メッセージ (NIS+), 441

Changing Key メッセージ, 446

chkey, 145, 226, 236, 243, 244, 246, 251, 252,
300, 455

Chkey failed メッセージ, 447

root のパスワードの変更, 294

Chkey failed メッセージ (NIS+), 447

client_info テーブル (NIS+), 422

client_info ファイル, 387, 390

client_info ファイルとテーブル, 381, 382
格付け番号, 382-383, 383

client_info ファイルとテーブル (続き)

サブネット, 384

単一のクライアント, 384

変更, 382

core ファイル, 435

Corrupt database メッセージ (NIS+), 438

cred テーブル, エントリがない, 453

cred テーブル

詳細, 238-239

内容を表示する, 367

認証の種類, 239

リンク, 238

リンクできない, 372

列, 422

cred テーブル (NIS+), 422-423

リンクを使用できない, 422

crontab ファイル, バックアップ (NIS+), 402

cron ファイル, 335, 345

D

data.dict ファイル, 56, 57, 80, 138, 166, 404

Database corrupted メッセージ (NIS+), 438

data encryption standard (データ暗号化規格), 513

/data ディレクトリ (NIS+), 404

dbm ファイル, 193

defaultdomain ファイル, NIS+ のアンインストール, 414

DES, 513

.dict ファイル, 56, 80, 443

directory name error メッセージ, 437

dir ディレクトリオブジェクト, 142

.dir ファイル, 193

DNS, 519

DNS, 513

DNS ゾーン, 513

DNS ゾーンファイル, 513

DNS 転送, 338-339, 513

domainname, 135, 156

インストールした NIS+ の削除, 412

末尾のドット, 135

domain name error メッセージ, 437

E

echo, 281

entry corrupt メッセージ (NIS+), 438

/etc/auto* テーブル, 444

/etc/bootparams ファイル, 421

/etc/defaultdomain, 442

NIS+ のアンインストール, 414

/etc/hosts, 100, 108, 117

/etc/nsswitch.conf, 38, 39, 95, 136

/etc/nsswitch.files, 37

/etc/nsswitch.nis, 37

/etc/nsswitch.nisplus, 37

/etc/passwd, 95, 134

/etc/passwd ファイル

nisaddent, 376

nisaddent コマンド, 191

/etc/resolv.conf, 134

/etc/resolv.conf ファイル, 155

/etc/.rootkey, 253, 455

/etc/.rootkey files, 141

/etc/.rootkey ファイル, サーバー (NIS+) の置換, 408

/etc/shadow, nisaddent, 376

/etc/shadow ファイル, nisaddent コマンド, 191

/etc/syslog.conf, エラーメッセージ, 467

/etc ディレクトリ, 455

/etc ファイル, 40, 50, 55, 415

ethers テーブル, 423

列, 423

ethers テーブル (NIS+), 423

アドレス形式, 423

F

FMRI, svc:/network/rpc/nisplus, 111-113

full dump rescheduled メッセージ (NIS+), 464

G

Generic system error メッセージ, 437

gethostbyname(), 29

getipnodebyname(), 29

getpwnam(), 29

getpwuid(), 29

getXbyY(), 29
GID, 513
group.org_dir ディレクトリ, 319
groups_dir, 356
groups_dir
 作成nissetup, 139, 179
groups_dir ディレクトリ, 59, 75, 223, 224, 320
 インストールした NIS+ の削除, 413, 414
groups.org_dir テーブル, 356
group テーブル, 223, 423-424
 列, 423-424

H

/hostname ディレクトリ, 166
hosts テーブル, 424
 列, 424
hosts ファイル, 101, 108, 122

I

Illegal object type メッセージ (NIS+), 435
Insufficient permission メッセージ (NIS+), 445, 447
insufficient permission メッセージ (NIS+), 455
Invalid principal name メッセージ (NIS+), 440
IP, 513
IPv6
 使用できるようにする, 39
 スイッチファイル, 39
IP アドレス, 513
 IP アドレス、更新, 257
 更新, 257

K

keylogin, 226, 230, 233, 234, 235, 236, 246, 250-251, 252, 300, 455
 Secure パスワードとログインパスワードが異なる, 454
keylogout, 226, 230, 236

keyserv, 136, 154, 226, 453
 エラー, 452
keyserver, nsswitch.conf ファイルおよび, 34
Keyserv fails to encrypt メッセージ (NIS+), 447

L

last.upd ファイル, 404
/lib/svc/method/nisplus, 112, 113
 -B オプション, 113
 -Y オプション, 112, 113
/lib/svc/method/nisplus ファイル, 90, 338-339
.log, 212
Log corrupted メッセージ (NIS+), 438
log entry corrupt メッセージ (NIS+), 438
login, 250
Login incorrect メッセージ, 292
login incorrect メッセージ, 462
Login incorrect メッセージ (NIS+), 447
Login Incorrect メッセージ (NIS+), 447-448
.log ファイル, 56, 80, 192, 374
 旧バージョンのファイル, 443
log ファイル, ディスク容量の不足, 461
ls, 455

M

mail_aliases テーブル, 424-425
 入力ファイルのフォーマット, 425
 列, 425
make, 198
makedbm, 195
MIS (management information system or service), 513
modify, 268
multihomed NIS+ 複製サーバー,
 /etc/hosts, 117
multihomed NIS+ ルートマスターサーバー,
 /etc/hosts, 100

N

netgroup.org_dir ディレクトリ, 320
netgroups.org_dir テーブル, 356
netgroup テーブル, 425-426
 wildcards, 426
 入力ファイルのフォーマット, 426
 列, 425-426
netmasks テーブル, 列, 427
netmasks テーブル (NIS+), 426-427
networks テーブル, 427
 列, 427
networks テーブル (NIS+), 427
newkey, 226
NIS, 513
 NIS+, 55-56
 NIS+, NIS 互換モード, 51
 NIS+ から情報を転送する, 197-198
 NIS+ 環境のマシン, 55
 NIS+ と使う, 49
 NIS+ との違い, 47-50
 NIS+ の互換性の問題, 439-441
 NIS+ 環境内のサブドメイン, 55
 マシン上のファイルベースのネーミング, 56
NIS+, 513
 API, 54
 attempt to remove a non-empty
 table メッセージ, 438
 Authentication denied メッセージ, 447
 Authentication error メッセージ, 447
 Busy try again later メッセージ, 456
 Callback: - select failed メッセージ, 438
 CALLBACK_SVC: bad argument メッセージ, 438
 Cannot [do something] with log" type
 メッセージ, 460
 Cannot find メッセージ, 441
 Cannot get public key メッセージ, 447
 Cannot remove replica メッセージ, 438
 Can't find suitable transport
 メッセージ, 441
 Changing Key メッセージ, 446
 Chkey failed メッセージ, 447
 Corrupt database メッセージ, 438
 cred テーブルにエントリがない, 453
 Database corrupted メッセージ, 438
 directory name error メッセージ, 437
 domain name error メッセージ, 437

NIS+ (続き)

entry corrupt メッセージ, 438
 /etc/passwd 中にあるパスワード, 454
full dump rescheduled メッセージ, 464
Generic system error メッセージ, 437
groups_dir、削除できない, 437
Illegal object type メッセージ, 435
Insufficient permission メッセージ, 445, 447
insufficient permission メッセージ, 455
Invalid principal name メッセージ, 440
Keyserver fails to encrypt メッセージ, 447
Log corrupted メッセージ, 438
log entry corrupt メッセージ, 438
Login Incorrect メッセージ, 447-448
Login incorrect メッセージ, 447
NIS, 55-56
NIS_DUMPLATER, 464
NIS_PATH 変数, 78-79
nis dump result nis_perror メッセージ, 464
nisinit のエラー, 435
NIS 互換モード, 51, 215
NIS との違い, 47-50
NIS に情報を転送する, 197-198
NIS の互換性の問題, 439-441
NIS マシン, 55
No memory メッセージ, 460
No public key メッセージ, 447
Not exist メッセージ, 441
Not found メッセージ, 441-444
not have secure RPC credentials
 メッセージ, 445
Not responding メッセージ, 457
one replica is already resyncing
 メッセージ, 464
org_dir、削除できない, 437
Out of disk space メッセージ, 460
password expired メッセージ, 448
Permission denied メッセージ, 440, 447,
 463
permission denied メッセージ, 446, 452,
 455
Possible loop detected in
 namespace メッセージ, 437

NIS+ (続き)

- replica_update: メッセージ, 464
- rescheduling the resync メッセージ, 464
- rlogin、ユーザーがログインできない, 463
- .rootkey ファイルがすでに存在している, 455-456
- root のパスワードを変更したための問題, 456
- rpc.nisd 終了している, 458
- rpc.nisd の失敗, 439
- rpc.nisd、問題, 438
- Security exception メッセージ, 445, 446
- Server busy. Try Again メッセージ, 459
- Unable to find メッセージ, 441
- Unable to fork メッセージ, 460
- UNABLE TO MAKE REQUEST メッセージ, 446
- Unable to make request メッセージ, 445
- Unable to stat メッセージ, 441, 445
- Unknown user メッセージ, 440
- アクセス, 220-225
- アクセス権, 224-225
- アクセス権の問題, 444-447
- 新しいパスワードが使えない, 462-463
- オートマウントを使用できない, 444
- オブジェクトが存在しない、メッセージと問題, 442
- 「オブジェクトの問題」メッセージ, 435
- 親ドメイン内のサーバー, 71
- 完全指定名, 72
- 管理者, 225
- 管理上の問題, 434-438
- キャッシュマネージャ, 67
- キャッシュマネージャがない, 458-459
- 切り捨てることができないログ, 436
- クライアント, 65-72
- クライアントでもあるサーバー, 71-72
- グループクラス, 221, 222-223
- グループに追加できない, 436
- グループ名, 75-76
- 検索パス, 208-210
- 広域ネットワーク (WAN), 380
- 更新, 63-64
- 構成, 87-91

NIS+ (続き)

- 構成方法, 80-81
- コールドスタートファイル, 67-71
- コマンド, 52-54
- サーバー, 61-64
- サーバーの起動が遅い, 459
- サーバー (複製), 63
- サーバー (マスター), 63
- 再帰的なグループ, 458
- サブネット, 166
- 資格, 218-220
- 主体, 50
- 主体, 65-72
- 主体の名前, 77
- 照会がハングする, 460
- 使用できる記号名, 77
- 承認, 50, 216, 220-225
- 承認クラス, 221-224
- 所有権の問題, 444-447
- 所有者クラス, 221, 222
- スイッチファイルの問題, 441
- 生存期間, 68
- 性能の問題, 456-460
- セキュリティ, 50
- セキュリティコマンド, 226
- セキュリティの概要, 215-218
- セキュリティの問題, 447-456
- セキュリティレベル, 217-218
- 設定スクリプト, 83-84, 93-94, 94
- 設定の前に, 54-56, 79-80
- その他クラス, 221, 223
- チェックポイントのエラー, 435
- ディスク容量の不足, 436, 461
- ディレクトリ, 59-60
- ディレクトリ (UNIX), 56-57
- ディレクトリキャッシュ, 67
- ディレクトリ、削除できない, 437
- ディレクトリ名, 75
- テーブル, 49, 206
- テーブルエントリ名, 76
- テーブル間でのリンク, 444
- テーブルの更新, 211-212
- テーブルの構造, 205-210
- テーブルの設定, 210-212
- テーブルパス, 457
- テーブル名, 75-76
- テスト, 463-464
- デバッグ, 434-438

NIS+ (続き)

- ドメインの問題, 442
- ドメイン名, 74-75, 80
- ドメイン名が変更されている, 453-454
- トランザクションログ, 63-64
- 名前空間サンプル, 87, 92-94
- 名前空間の構造, 57-58
- 名前展開, 77-78
- 名前の中の空白, 443-444
- 認証, 50
- 認証, 216, 218-220
- パスワード, ログインの失敗, 440-441
- パスワードが異なる, 454-455
- パスワードコマンド, 218
- パスワードを変更できない, 463
- パフォーマンス, 166
- ファイル, 56-57
- ファイルの問題, 443
- 複製, ディレクトリを削除できない, 437-438
- 複製サーバーが多すぎる, 457
- 複製サーバーが同期していない, 442-443
- 複製サーバーの更新のエラー, 464-465
- 複製サーバーの同期遅延, 442-443
- 部分指定名, 72
- プロセス数の不足, 461
- ホスト名, 76-77, 80
- マシンを新しいドメインに変更する, 454
- 未認証クラス, 221, 223
- 命名規則, 72-78
- メッセージ, 464
- メモリーの不足, 460-461
- 問題解決, 434-465
- ユーザーがログインできない, 462
- ユーザーの問題, 461-463
- リソースの問題, 460-461
- リモートサイト, 167
- ルートサーバー, 80
- ログが大きすぎる, 436, 458
- nis_cachemgr, 52
- nis_checkpoint, 52
- NIS_COLD_START ファイル, 89, 143, 158, 159, 160
 - 置換するサーバー (NIS+), 408
- NIS_DEFAULTS, 281
- NIS_DEFAULTS, 270, 276, 280, 281
 - 値の表示, 281-282
 - 再設定, 282-283
- NIS_DUMPLATER, 464
- NIS_GROUP, 275, 325
- NIS_GROUP, 設定, 178
- NIS_OPTIONS
 - オプション, 434
 - デバッグ, 433
- NIS_PATH, パフォーマンスへの影響, 457
- NIS_PATH, 問題, 442
- NIS_PATH 変数, 78-79
- NIS_SHARED_DIRCACHE ファイル, 341
- NIS_SHARED_DIRCACHE ファイル, 67
- NIS+ オブジェクト, 514
- NIS+ 環境, 298**
- NIS+ 環境, 514
- NIS+ キャッシュ, 内容を表示する, 342
- NIS+ グループ, 320
 - NIS+, 320
 - NIS_DEFAULTS, 325
 - NIS_GROUP, 325
 - nistbladm, 356
 - 暗黙的なメンバー, 321
 - 暗黙的なメンバー以外の主体, 322
 - 管理グループ, 145
 - 構文 (NIS+), 322
 - 再帰的, 性能の低下, 458
 - 再帰的なメンバー, 322
 - 再帰的なメンバー以外の主体, 322
 - 削除, 337
 - 削除 (NIS+), 325-326
 - 作成 (NIS+), 324-325
 - 指定 (NIS+), 322
 - 属性、表示, 323
 - 明示的なメンバー, 321
 - 明示的なメンバー以外の主体, 322
 - メンバー以外, 322
 - メンバーであるかどうかを調べる (NIS+), 327-328
 - メンバーの削除 (NIS+), 327
 - メンバーのタイプ (NIS+), 321-322
 - メンバーの追加 (NIS+), 326
 - メンバーの表示 (NIS+), 326-327
 - メンバー、リスト, 323
 - ユーザーを追加できない, 436
- NIS+ 主体, 514
- NIS+ ディレクトリ, 329
 - nis_cachemgr, 341
 - niscat, 329-330
 - nisinit, 339-341
 - nisls, 330-332

NIS+ ディレクトリ (続き)

- nismkdir, 332-335
- nisping, 342-345
- nisping の強制実行, 344
- nism, 337-338
- nismrmdir, 336-337
- nisshowcache, 342
- オブジェクトの削除, 337-338
- 削除, 336
- 作成, 333-334
- 属性の表示, 330
- チェックポイント, 342-345
- ドメインに展開する, 373-374, 374
- トランザクションログ, 346-347
- 内容のリスト, 331, 332
- 複製サーバーの作成, 333
- 複製サーバーの追加, 334-335, 335
- 複製サーバーを NIS+ ディレクトリから切り離す, 336-337
- ルート以外の NIS+ ディレクトリの作成, 332
- ルートディレクトリの作成, 332

NIS+ テーブル, 49, 351, 418-419

- auto_home テーブル, 419
- auto_master テーブル, 419-420, 420
- bootparams テーブル, 420-421
- client_info テーブル, 422
- cred テーブル, 422-423
- cred テーブルの内容を表示する, 367
- ethers テーブル, 423
- group テーブル, 423-424
- hosts テーブル, 424
- links, 372-373
- mail_aliases テーブル, 424-425
- netgroup テーブル, 425-426
- netmasks tables, 426-427
- networks テーブル, 427
- nisaddent, 373, 374-378
- niscat, 366
- nisgrep, 368-371
- nisln, 372-373
- nismatch, 368-371
- nissetup, 373-374
- nistbladm, 352-356
- NIS マップからの生成, 192-196
- NIS マップからのデータ転送, 376-378
- passwd テーブル, 427-429
- protocols テーブルs, 429

NIS+ テーブル (続き)

- rpc テーブル, 430
- services テーブル, 430-431
- timezone テーブル, 431
- 演算子, 370
- エントリの NULL 終了, 358
- エントリの削除, 364-366, 365-366
- エントリの修正, 362-364
- エントリのセキュリティ, 271-272
- エントリの追加, 359-362, 361-362
- エントリの編集, 362-364, 364
- 最大サイズ, 359
- 削除, 359
- 作成, 356-358
- 自動マウントテーブルの追加, 358
- スワップ空間, 145
- 正規表現, 369-370
- 生成のオプション, 186
- 生成方法, 185
- セキュリティ, 270-274
- セキュリティとレベル, 272-274
- 設定 (コマンド), 185-186
- 設定 (スクリプト), 100-106
- 属性の表示, 367-368
- 置換, 186
- 追加, 186
- データ転送, 374
- データの転送 オプション, 374
- テーブルパス, 性能の低下, 457
- テーブルを空にする, 337
- 内容を表示する, 366
- 入力ファイル, 418-419
- ファイルから生成する, 187-192
- ファイルからデータを転送する, 375-376
- ファイルにデータをダンプする, 378
- ほかのネームサービス, 418
- マージ, 186
- 問題, 435
- リンクが機能しない, 444
- リンクできない cred テーブル, 372
- リンクできないエントリ, 372
- 列の検索, 371
- 列の構成要素, 357
- 列の指定, 357-358
- 列の種類, 357-358
- 列のセキュリティ, 271-272
- ワイルドカード, 370

NIS+ トランザクションログ, 514

NIS+ の削除, 412-413
 クライアントマシンから削除する, 412-413
 サーバーから削除する, 413-414
 名前空間からの削除, 414-415
 nisaddcred, 140, 153, 181, 226, 239, 241, 243, 247
 DES 資格、追加, 144
 LOCAL 資格、追加, 143
 インストールした NIS+ の削除, 412
 鍵の変更, 253, 254, 255
 資格、作成方法, 241-242
 資格の管理, 247-248
 資格の更新, 247
 資格の削除, 240, 247-248
 資格の作成, 240, 244, 246
 資格の変更, 240
 システム管理者の資格を追加, 144
 タイムスタンプ, 231
 nisaddent, 195
 nisaddent, 145, 187, 190, 191, 194, 195, 197, 211, 283, 373, 374-378
 nisaddent, 196
 nisaddent
 NIS マップからのデータ転送, 376-378
 passwd ファイル, 376
 構文, 375
 自動マウントテーブル, 376
 データからファイルを転送, 375-376
 データの転送オプション, 374
 テーブル、標準以外, 376
 ファイルにデータをダンプする, 378
 nisbackup, 170, 399-405, 408
 nisbackup, 上書き, 402
 nisbackup
 オプション, 401
 構文, 400-401
 自動化, 402
 全名前空間, 403
 ディレクトリ、個別のバックアップ, 402-403
 バックアップディレクトリの構造, 403-404
 バックアップファイル, 404-405
 ファイルシステムのバックアップ, 402
 マスターサーバーのみ, 401
 割り込み, 400
 niscat, 106, 142, 143, 151, 169, 190, 240, 274, 329-330, 354, 360, 365, 366
 NP, 367
 cred テーブルの内容を表示する, 367
 niscat (続き)
 Server busy. Try Again メッセージ, 459
 オブジェクト属性の表示, 367-368
 オプション, 366
 グループの属性, 323
 グループのメンバー, 323
 構文, 366
 ディレクトリの属性, 329-330, 330
 nischgrp, 52, 288, 320
 グループの変更, 288-289
 nischmod, 52, 180, 200, 270, 283
 アクセス権、削除, 284
 アクセス権、追加, 283-284
 nischown, 52, 286, 299
 所有者、変更, 287
 nischttl, 52, 68, 347-349
 鍵情報の更新, 258
 nisclient, 107, 109, 127-128, 229, 239, 256, 258, 413, 447
 nisclient, 125-127, 127-128
 nisclient, インストールした NIS+ の削除, 412
 nisclient スクリプト, 83, 84, 106-109
 DNS, 85
 multihomed NIS+ 複製サーバー, 117
 クライアントの追加, 109
 サブドメインクライアント, 125-127
 前提条件, 107, 109-110
 ユーザーの初期設定, 109-110, 110, 127-128, 128
 nisdefaults, 52, 280-281, 453
 TTL, 347
 生存期間, 347
 表示オプション, 280
 nis dump result nis_perror メッセージ (NIS+), 464
 nisgrep, 52, 368-371
 演算子, 370
 オプション, 370-371
 検索、最初の列を, 371
 検索、特定の列を, 371
 検索、複数の列を, 371
 構文, 370-371
 正規表現, 369-370
 ワイルドカード, 370
 nisgrpadm, 52, 84, 105, 141, 142, 145, 180, 181, 320, 324, 326, 356

nisgrpadm (続き)
 アクセス権, 324
 グループの構文, 324
 グループの削除, 325-326, 337
 グループの作成, 324-325
 グループの属性、表示, 323
 グループのメンバー、リスト, 323
 グループメンバーの構文, 324
 構文, 322
 メンバーであるかどうかを調べる, 327-328
 メンバーの削除, 327
 メンバーの追加, 326
 メンバーの表示, 326-327
 問題, 435
 nisinit, 52, 56, 137, 138, 158, 159, 160, 258, 339-341, 443, 460
 インストールした NIS+ の削除, 412
 クライアント、初期設定, 339-340
 ルートディレクトリ, 332
 ルートマスター、初期設定, 340-341
 nisinitの問題, 435
 nisln, 52, 372-373
 cred テーブル, 372
 オプション, 372
 構文, 372
 nisln, テーブルエントリ, 372
 nisl
 リンクの作成, 372-373
 nislog, 52, 346-347
 オプション, 346
 トランザクションログの内容を表示する, 346-347
 nisls, 52, 320, 330-332, 331, 332, 444
 ディレクトリの内容, 330-332, 332
 nismatch, 52, 240, 248, 368-371
 Changing Key メッセージ, 446
 演算子, 370
 オプション, 370-371
 検索、最初の列を, 371
 検索、特定の列を, 371
 検索、複数の列を, 371
 構文, 370-371
 正規表現, 369-370
 ワイルドカード, 370
 nismkdir, 52, 137, 169, 178, 210, 283, 332-335, 333
 ディレクトリの作成、親でないルート, 333
 nismkdir, 複製サーバーの作成, 333
 nismkdir
 複製サーバーの追加, 334-335, 335
 マスターサーバーのみ, 333
 ルート以外のディレクトリの作成, 332, 333-334
 ルートディレクトリを作成できない, 332
 nispasswd, 218, 293, 296
 nisping, 52, 170, 171, 172, 173, 196, 212, 343, 345
 forcing, 344
 最新更新時間, 343-344
 性能への影響, 457
 チェックポイント, 192
 チェックポイントのエラー, 435
 チェックポイントの設定, 196
 ディレクトリ、チェックポイント, 342-345
 ディレクトリ、チェックポイントの設定, 344-345
 テーブルの作成, 191
 複製サーバーの設定, 171-172, 172
 複製サーバーの追加, 335
 複製サーバーをディレクトリから切り離す, 336
 nispopulate, 211, 239
 nispopulate スクリプト, 83, 84, 100-106
 multihomed NIS+ 複製サーバー, 117
 multihomed NIS+ ルートマスターサーバー, 100
 管理グループ, 105
 セキュリティ, 101, 122
 前提条件, 102
 ドメインの追加, 121-124, 123
 入力ファイル, 101
 必要なスワップ領域, 106
 ファイルから生成, 123-124
 ファイルの準備, 101, 122
 マップから生成, 124
 nisprefadm, 52, 383, 384, 385, 389, 395, 398
 client_info テーブル, 422
 オプション, 386-387
 格付け番号, 382-383, 383
 格付け番号の指定, 388
 クライアント名, 384
 グローバルテーブル, 381-382
 グローバル優先順位の設定, 388-390, 390
 構文, 385
 サーバー使用、概要, 381-385
 サーバーの優先順位変更, 391-393

nisprefadm (続き)
 サーバー名, 384
 サブネット, 384
 単一のクライアント, 384
 有効化, 397-398, 398
 優先サーバー以外のサーバー使用, 395
 優先サーバー限定指定, 394-395
 優先サーバーの指定, 380
 優先順位の使用の終了, 396
 優先順位の番号の変更, 392
 優先順位の表示, 384, 387, 388
 リストからのサーバー削除, 393
 リスト内サーバー置換, 392-393
 リストの置換, 394
 ローカルファイル, 381-382
 ローカルマシン上での優先順位の設定, 391
 nisrestore, 52, 170, 171, 405-407
 resolve.conf ファイル, 405
 rpc.nisd, 406
 オプション, 406
 検出エラー, 407
 構文, 406
 サーバーの置換, 408-409
 サーバーの複製, 408-409
 前提条件, 405-406
 損傷した名前空間の復元, 407
 ディレクトリの復元, 407
 ディレクトリ名, 407
 手順, 406-407
 複製サーバーの設定, 169-170, 407
 nisrm, 52, 337-338
 nisrmdir, 52, 336-337
 オブジェクトの削除, 337-338
 ディレクトリの削除, 336
 ディレクトリを削除できない, 437
 複製サーバーを切り離す, 336-337
 nisserver, 118-121, 124-125, 332, 415
 複製 (NIS+)、設定, 407
 nisserver スクリプト, 83, 84, 95, 114, 210
 DNS と NIS+ サーバー, 113
 Multihomed NIS+ 複製サーバー, 117-118
 multihomed NIS+ 複製サーバー, 118
 multihomed NIS+ ルートマスターサーバー, 99-100
 NIS+ サーバー, 113
 NIS+ 複製サーバー, 111-113
 NIS+ ルート複製サーバー, 113-118, 115-116
 サブドメイン, 118-121, 120-121
 nisserver スクリプト (続き)
 サブドメイン複製サーバー, 124-125
 前提条件, 95-96
 デフォルトのセキュリティレベル, 97
 ドメインの追加, 121
 ルートサーバー, 95-100, 98-99
 nissetup, 98, 132, 210, 373-374
 automounter テーブル, 191
 groups_dir の作成, 139, 179
 org_dir の作成, 139, 179
 passwd テーブル, 191
 テーブルの作成, 139, 179
 ドメインに展開するディレクトリ, 373-374, 374
 nisshowcache, 52, 68, 342
 内容を表示する, 342
 nisstat, 52
 nistbladm, 52, 200, 210, 211, 270, 283, 284, 314, 352-356, 354, 364, 457
 NIS+ グループ, 320, 321, 356
 UNIX グループ, 356
 warn の値, 302
 アクセス権、列, 284-286, 286
 インデックス名, 355-356
 エントリの上書き, 361-362
 エントリの強制, 361
 エントリの削除, 364-366, 365-366
 エントリの修正, 362-364
 エントリの追加, 359-362
 エントリの編集, 362-364, 364
 オプション, 353
 間隔, 302
 管理者の資格、追加, 144
 キー, 354-355
 期限 値, 302
 グループ, 356
 検索可能列, 354-355
 構文, 352-353
 最小値, 301
 最大値, 301
 自動マウントテーブルの追加, 358
 シャドウ列のフィールド, 301-303
 追加のエントリ, 360
 テーブルの削除, 359
 テーブルの作成, 356-358
 テーブルを空にする, 337
 同一のエントリ, 360
 日数, 303

nistbladm (続き)
 ネットグループ, 356
 パスワード, 300-303
 パスワードの有効期間, 311
 パスワードの有効期限、解除, 313
 未使用 値, 302
 有効期限、設定, 312-313
 列エントリの NULL 終了, 358
 列の値, 353-354
 列の構成要素, 357
 列の指定, 357-358
 列の種類, 357-358
 ログインの間隔、設定, 313-314, 314
 nistest, 52
 nisupdkeys, 52, 226, 236, 254, 255, 256, 257
 鍵の更新、例, 256-257, 257
 コールドスタートファイル, 256
 引数, 256
 古くなった鍵の更新, 449
 NIS 互換モード, 97, 514
 起動, 338-339
 個々のマシン上の NIS, 55
 設定 (コマンド), 132, 176
 NIS マップ, 514
 NNSP, 514
 No memory メッセージ (NIS+), 460
 No public key メッセージ (NIS+), 447
 Not exist メッセージ (NIS+), 441
 Not found メッセージ (NIS+), 441
 not have secure RPC credentials
 メッセージ, 445
 Not responding メッセージ (NIS+), 457
 nscd, 153
 nscd デーモン, 39, 136
 nsswitch.conf files, 問題, 441
 nsswitch.conf で IPv6 を使用できるように
 する, 39
 nsswitch.conf ファイル, 29, 34, 39, 51, 95,
 136, 150, 296
 nsswitch.conf ファイル, 38
 nsswitch.conf ファイル
 auto_home テーブル, 33
 auto_master テーブル, 33
 compat, 39, 40
 keyserver エントリ, 34
 NIS+ と NIS の互換性の問題, 439
 NIS+ の問題, 441
 NOTFOUND=continue, 32
 nsswitch.conf ファイル (続き)
 nsswitch.files ファイル, 35
 nsswitch.nisplus ファイル, 35
 nsswitch.nis ファイル, 35
 passwd_compat, 39
 publickey エントリ, 34
 SUCCESS=return, 32
 timezone テーブル, 33
 TRYAGAIN=continue, 32
 UNAVAIL=continue, 32
 インストール, 38-39
 インストールした NIS+ の削除, 415
 応答, 32
 オプション, 32
 検索基準, 31, 32-33
 更新, 41
 構文が正しくない, 33
 コメント, 34
 状態メッセージ, 31-32, 32
 情報のソース, 31
 続行, 32
 デフォルトテンプレートファイル, 35-37
 デフォルトファイル, 37-38
 テンプレート, 30, 34-38
 動作, 32
 パスワード, 462
 パスワード情報および, 40-41
 ファイルの選択, 38-39
 フォーマット, 30
 変更, 32
 見つからない, 33
 メッセージ, 状態, 31-32
 例, 35-36, 36
 nsswitch.files ファイル, 37
 nsswitch.nis, 36
 nsswitch.nisplus ファイル, 37
 nsswitch.nis ファイル, 37
 NULL 終了, 358
O
 one replica is already resyncing
 メッセージ (NIS+), 464
 org_dir
 作成nissetup, 139, 179
 org_dir. ディレクトリ, 438
 org_dir ディレクトリ, 59, 75, 210, 224

org_dir ディレクトリ (続き)
インストールした NIS+ の削除, 413, 414
org_ ディレクトリオブジェクト, 142
Out of disk space メッセージ (NIS+), 460

P

.pag ファイル, 193
passwd, 218, 226, 294, 296, 297-300, 304, 306, 440
NIS+ 環境, 298
nispasswd, 296
rlogin の問題, 463
root、変更, 294
yppasswd, 297
アクセス権, 299
鍵, 299-300
休暇の場合のロック, 307
強制的な変更, 308, 311
passwd, 強制的な変更, 311
passwd
警告期間、設定, 310-311
資格, 298
使用期間、設定の解除, 311
情報の表示, 304-305
パスワードの使用期間, 307-314
パスワードの変更, 293, 305-306
パスワードの有効期限, 308
パスワードのロック, 306-307
パスワードロックの解除, 307
変更禁止期間、設定, 309-310
ほかのドメイン, 300
有効期間, 308-309
ユーザーがパスワードを変更できない, 463
ユーザーの問題, 462
passwd テーブル (NIS+), 427-429
+/- 構文, 429
シャドウ列の情報, 428-429
列, 428
passwd ファイル, 95, 101, 122, 134
nisaddent, 376
password expired メッセージ, 448
password -r コマンド, 41
passwords, 変更後のログイン失敗, 440-441
password will expire メッセージ, 292
Permission denied メッセージ, 292
permission denied メッセージ, 446, 452

Permission denied メッセージ (NIS+), 440, 447, 463
permission denied メッセージ (NIS+), 455
pinging, 514
Possible loop detected in namespace
メッセージ, 437
protocols テーブル, 429
列, 429
protocols テーブル (NIS+), 429
publickey files, 191, 195
publickey マップ, 195

R

replica_update: メッセージ (NIS+), 464
rescheduling the resync メッセージ
(NIS+), 464
resolv.conf ファイル, 134, 155
resolv.conf ファイル, 405
rlogin, 463
問題, 463
rlogin, ユーザーの問題, 462
root_dir ファイル (NIS+), 404
root.object ファイル, 137, 404
.rootkey ファイル, 412, 455
NIS+ のアンインストール, 414
既存, 455-456
サーバー (NIS+) の置換, 408
root ディレクトリオブジェクト, 142
RPC, 514
rpc.nisd, 56, 85, 97, 114, 115, 119, 125, 138, 338-339, 407, 458, 460
DNS 転送, 338-339
NIS 互換モード, 338-339
オプション, 339
起動, 338
構文オプションの変更, 339
失敗, 359
終了, 458
セキュリティレベル, デフォルト, 338-339
停止, 338-339
テーブルのサイズ, 359
復元 (NIS+), 406
複数の親プロセス, 439
rpc.nisd デーモン, 85, 99

rpc.nispasswd
パスワードの最大ログイン所要時
間, 317-318
パスワードの試行回数の制限, 317
パスワードのログインの失敗, 317-318
rpc テーブル
例, 430
列, 430
rpc テーブル (NIS+), 430

S

Secure RPC ネット名, 144, 231, 232-233
主体名, 242
詳細, 242
Secure RPC パスワード, 514
security, シェドウ列のフィールド, 301-303
Security exception メッセージ, 446
Security exception メッセージ (NIS+)
, 445
sendmail, mail_aliases テーブル, 424
Server busy. Try Again メッセージ, 459
services テーブル, 列, 431
services テーブル (NIS+), 430-431
setenv, 282
shadow ファイル, 101, 122
nisaddent, 376
SMF, 「サービス管理機能」を参照
snoop, 452
Sorry: less than メッセージ, 293
svc:/network/rpc/nisplus, 111-113
svcadm, 実行するための前提条件, 111-112
switch ファイル, nsswitch.nis, 36
syslog.conf ファイル, エラーメッセー
ジ, 467
syslog ファイル, チェックポイントのエラー
, 435

T

TCP, 514
TCP/IP, 514
timezone テーブル, 33, 431
列, 431
timezone テーブル (NIS+), 431
/tmp/CALLS ファイル, 434

TMPDIR, 103
/tmp ディレクトリ, 160
tmp ファイル, ディスク容量の不足, 461
trans.log ファイル, 56, 80, 138, 166, 342
Transport Control Protocol, 514
TTL, 347-349
nisdefaults, 347
値, 348
オブジェクトの生存期間の変更す
る, 348-349
オプション, 348
テーブルエントリの生存期間の変更, 349
変更, 347

U

Unable to find メッセージ (NIS+), 441
Unable to fork メッセージ (NIS+), 460
UNABLE TO MAKE REQUEST メッセージ, 446
Unable to make request メッセージ (NIS+)
, 445
Unable to stat メッセージ (NIS+), 441, 445
Unknown user メッセージ (NIS+), 440
User ID 0, 456
/usr/bin ディレクトリ, 56
/usr/lib/netsvc/yp/ypstart スクリプ
ト, 155
/usr/lib/nis, 342
/usr/lib/nis, 256
/usr/lib/nis/nisupdkeys, 142
/usr/lib/nis ディレクトリ, 56, 187, 189,
192, 194
/usr/lib ディレクトリ, 56
/usr/sbin ディレクトリ, 56

V

/var/nis, 337, 374
/var/nis/client_info, 387
/var/nis/data.dict, 57, 80, 138
/var/nis/data/trans.log, 342
/var/nis/data ディレクトリ, 56, 80, 137,
138, 166, 443
/var/nis/NIS_COLD_START, 置換するサー
バー (NIS+), 408

/var/nis/NIS_SHARED_DIRCACHE ファイル, 341
/var/nis/NIS_SHARED_DIRCACHE ファイル, 236
/var/nis/rep/org_dir ディレクトリ, 438
/var/nis/rep/serving_list ファイル, 438
/var/nis/root.object, 137
/var/nis/trans.log, 80, 138, 166
/var/nis ディレクトリ, 56, 80, 112, 136, 138, 154, 166
インストールした NIS+ の削除, 414, 415
旧バージョンのファイル名, 443
/var/yp, 376
/var/yp ディレクトリ, 193

W

WAN, NIS+, 380
WARNING: password differs from login password, 141

X

X.500, 515
XDR コード化 (NIS+), 404
xfr ファイル, 187

Y

ypcat, 40
netgroup テーブル, 425
yppasswd, 297
ypserv, 155
ypupdate, 51
ypxfr, 51

あ

アクセス権, 84, 515
アプリケーションレベルのネームサービス, 515
暗号化鍵, 515

い

インターネット, 515
ルートドメイン, 95-96
インターネットアドレス, 515
インデックス付き名前, 515
インデックス名 (NIS+ テーブル), 355-356

え

エラーメッセージ
アルファベット順表示, 468-469
エラーメッセージ内の番号, 469
解釈, 468
しきい値の表示, 467-468
内容, 467-468
遠隔手続き呼び出し (RPC), 515
エンタープライズレベルのネームサービス, 515
エンタープライズレベルのネットワーク, 515
エントリ, 271
エントリ, 515

お

オートマウンタ, マップの追加, 207
「オブジェクトの問題」メッセージ (NIS+), 435
親ドメイン, 515

か

鍵, 249-250
1 対の鍵, 241-242
DES, 231
keylogin, 250-251
passwd, 299-300
共通, 231
クライアントの鍵情報の更新, 258
公開, 主体, 229, 231
公開, サーバー, 229, 231
更新, 255-257, 257
サーバーの公開鍵, 231, 236
主体の非公開鍵, 230
タイムスタンプ, 231
手作業で更新, 449
非公開, サーバー, 231

鍵 (続き)

- 非公開, 主体, 229
- 非公開鍵の再暗号化, 252
- 非公開、サーバー, 229
- 非公開、主体, 230, 231
- 古くなった鍵の更新, 449
- 変更, 251-252, 252-255, 254, 255
- 乱数 DES, 231

鍵 (暗号化), 515

管理グループ

- 作成, 141, 178

き

キー

- 古くなったキーの更新, 449-452
- 問題と対策, 449-452

キー (NIS+ テーブル), 354-355

キーサーバー, 515

キー (列), 515

逆解決, 515

キャッシュマネージャ, 67, 341, 516

起動, 89, 143, 341

サーバーの優先順位, 385

サーバーの優先順位 (NIS+), 381

停止, 341

ない, 458-459

く

クライアント, 516

NIS+, 65-72

NIS+ 設定 (コマンド), 149-150, 152-154

NIS+ 設定 (スクリプト), 106-109, 109, 125-127

NIS+ の初期設定, 339-340

NIS+ の初期設定 (コマンド), 154, 156-161

NIS+ の初期設定 (スクリプト), 110

NIS+ の設定 (スクリプト), 107

NIS+ ユーザーの初期化 (スクリプト)

, 109-110

NIS+ ユーザーの初期設定 (スクリプト)

, 109-110, 127-128, 128

鍵情報の更新, 258

検索動作 (NIS+), 379-380

クライアント (続き)

コールドスタートによる初期化 (NIS+)

, 159-161

情報の更新, 258

ブロードキャストによる初期設定 (NIS+)

, 157

ホスト名による初期化 (NIS+), 158-159, 159

優先サーバーの指定 (NIS+), 380

クライアントサーバーモデル, 516

グループ, 319-320, 516

UNIX, 319

ネットグループ, 320

変更, 288-289

グループ ID, 516

グループクラス, 269

グループクラス, 221, 222-223, 268

グループクラスのアクセス権, 274

グローバルネームサービス, 516

こ

広域ネットワーク (WAN), 516

公開鍵, 516

更新, 142

コールドスタートによる初期化 (NIS+), 159-161

コールドスタートファイル, 67-71, 517

nisupdkeys, 256

子ドメイン, 517

さ

サーバー, 517

Multihomed NIS+ 複製サーバーの設定 (スクリプト), 117-118

multihomed NIS+ ルートマスターの設定 (スクリプト), 99-100

NIS+, 61-64

NIS+ 設定 (スクリプト), 111-113, 113

NIS+ 内で置かれているドメイン, 334

NIS+ 複製サーバー, 63

NIS+ 複製サーバー、最新更新時間, 343-344

NIS+ 複製サーバー、チェックポイント, 342-345

NIS+ 複製サーバーの作成, 333

NIS+ 複製サーバーの設定 (スクリプト)

, 111-113

サーバー (続き)

- NIS+ 複製サーバーの追加, 334-335, 335
- NIS+ マスター, 63
- NIS+ ルート複製サーバーの設定 (スクリプト), 113-118, 115-116
- アクセス権の割り当て, 276
- サブドメイン複製サーバーの設定 (スクリプト), 124-125
- サブネット, 114
- 設定 (NIS+ コマンド), 163-166
- 置換 (NIS+), 408-409
- 複製 (NIS+), 復元による設定, 407
- 複製サーバーの設定と
 - nisrestore, 169-170
- 複製の設定 (NIS+ コマンド), 166-172
- 複製の設定と nisping, 171-172, 172
- サーバーの優先順位 (NIS+), 379-380, 384
- 格付け番号, 382-383, 383
- 格付け番号の指定, 388
- キャッシュマネージャ, 385
- クライアントの検索動作, 379-380
- クライアント名, 384
- グローバル, 381-382
- グローバル優先順位の設定, 388-390, 390
- サーバー使用, 概要, 381-385
- サーバー名, 384
- サブネット, 384
- 使用の終了, 395-397
- 全サーバー, 383-384
- 単一のクライアント, 384
- デフォルト, 383
- 番号の変更, 392
- 必要なキャッシュマネージャ, 381
- 表示, 384, 387-388, 388
- 変更, 391-393
- 有効化, 397-398, 398
- 有効になるタイミング, 385
- 優先サーバー以外のサーバーを使用, 395
- 優先サーバー限定, 383-384
- 優先サーバー限定指定, 394-395
- 優先サーバーの指定, 380
- リストからのサーバー削除, 393
- リスト内サーバー置換, 392-393
- リストの置換, 394
- ローカル, 381-382
- ローカルマシン上での設定, 391
- サーバーリスト, 517

サービス管理機能

- NIS+, 89
 - SMF を使用しないとき, 90
 - rpc.nisd -x とともに使用する, 90
- 削除, 268
- 作成, 268
- サブドメイン
 - 設定 (スクリプト), 118-121, 120-121
- サブネット, 517
 - NIS+, 166
 - NIS+ ルート複製サーバー, 114

し

- 資格, 218-220, 227-228, 517
 - cred テーブルの詳細, 238-239
 - DES, 219, 228, 232-233
 - DES 確認フィールド, 233
 - DES 資格の構成要素, 231
 - DES 資格の作成, 233-235
 - DES 資格の詳細, 232-236
 - LOCAL, 219-220
 - LOCAL 資格の追加, 143
 - passwd, 298
 - Secure RPC ネット名, 242
 - WARNING: password differs from
 - login password, 141
 - 格納, 236-238
 - 管理, 247-248
 - 管理者, 244-245
 - 管理者、追加, 144
 - 管理者のための資格情報, 243
 - クライアント, 153
 - 更新, 247
 - 削除, 247-248
 - 作成, 140, 180, 239-247
 - 作成方法, 241-242
 - 資格情報, 228
 - 資格の削除, 240
 - 資格の作成, 240, 243-247, 246
 - 資格のタイプとユーザータイプ, 220
 - 資格の変更, 240
 - システム管理者の資格を追加, 144
 - 主体の認証, 229-232, 230
 - 主体名, 242
 - タイムスタンプ, 231
 - 認証コンポーネント, 228-229

資格 (続き)

- マシン, 218
- 無効, 452
- 問題と対策, 448
- ユーザー, 218
- リセット, 448

識別名, 517
シャドウ列, 428
主体, 50
主体, 517
承認, 50
所有者クラス, 269
所有者クラス, 221, 222, 268
新機能, NIS+ でのサービス管理機能, 89

す

スイッチファイル, `nsswitch.files`, 36-37
スレーブサーバー, 517
スワップ空間, NIS+ テーブル, 145

せ

性能, NIS+ サーバー検索, 380
セキュリティ
NIS+ アクセス権, 224-225
NIS+ コマンド, 226
NIS+ と, 50
NIS+ の概要, 213-215, 215-218
NIS 互換モード, 215
`passwd column` へのアクセスの制限, 198-200
`-s o` フラグ, 138
Secure RPC ネット名, 242
アクセス, 220-225
アクセス権, 267, 268-276, 269-270
アクセス権、組み合わせ, 269
アクセス権、構文, 277-279, 278, 279
アクセス権 (削除), 274
アクセス権、削除, 284
アクセス権 (作成), 273
アクセス権、追加, 283-284
アクセス権、デフォルト, 269-270
アクセス権とレベル, 272-274
アクセス権の割り当て, 276
アクセス権 (変更), 273-274

セキュリティ (続き)

アクセス権、変更, 270, 283-284
アクセス権 (読み取り), 273
アクセス権、読み取り, 274-275
アクセス権、列, 284-286, 286
アクセス権、連鎖, 269
鍵, 249-250
間隔, 302
管理者, 225
管理者の資格, 244-245
管理者のための資格情報, 243
期限 値, 302
グループクラス, 221, 222-223
グループの変更, 288-289
警告, 302
構文、アクセス権, 277-279, 278, 279
コマンドによるアクセス権の指定, 277-279
サーバーのアクセス権割り当て, 276
最小値, 301
最大値, 301
削除権, 274
作成権, 273
資格, 218-220
承認, 216, 220-225
承認クラス, 221-224
所有者クラス, 221, 222
所有者、変更, 287
その他クラス, 221, 223
テーブル, 270-274
テーブルとエントリ, 271-272
テーブルとエントリのアクセス権, 271-272
テーブルと列, 271-272
テーブルとレベル, 272-274
テーブルのアクセス権, 270-274, 272-274
デフォルト、設定, 281-283
デフォルトの表示, 280-281
デフォルトのレベル (スクリプト), 97
デフォルト、表示, 280
デフォルト、変更, 282
認証, 216, 218-220
パスワードコマンド, 218
非デフォルトのアクセス権, 283
非デフォルトのアクセス権指定, 283
変更権, 273-274
未使用 値, 302
未認証クラス, 221, 223
読み取り権, 273
列アクセス権, 271-272

セキュリティ (続き)

レベル, 85

レベル (NIS+), 217-218

設定

DNS と NIS+ サーバー, 113

Multihomed NIS+ 複製サーバー (スクリプト), 117-118

multihomed NIS+ ルートマスターサーバー (スクリプト), 99-100

NIS+, 87-91

NIS+ クライアント (コマンド), 149-150, 152-154

NIS+ クライアント (スクリプト), 106-109, 109

NIS+ クライアントの初期設定, 154, 156-161

NIS+ サーバー (スクリプト), 113

NIS+ サーバーの置換, 408-409

NIS+ スクリプト, 83-84

NIS+ 設定スクリプト, 93-94, 94

NIS+ テーブル (コマンド), 185-186

NIS+ テーブル (スクリプト), 100-106

NIS+ の準備, 54-56

NIS+ の準備, 79-80

NIS+ 複製サーバー (スクリプト), 111-113, 124-125

NIS+ ユーザーの初期化 (スクリプト), 109-110

NIS+ ユーザーの初期設定 (スクリプト), 109-110, 110, 127-128, 128

NIS+ ルート複製サーバー (スクリプト), 113-118, 115-116

PATH 変数, 96

管理グループ (スクリプト), 105

コールドスタートによる初期化 (NIS+), 159-161

サーバー (NIS+ コマンド), 163-166

サブドメインクライアント (スクリプト), 125-127

サブドメインの設定 (スクリプト), 118-121, 120-121

スイッチファイル, 38

スワップ領域 (スクリプト), 106

テーブル (NIS+), 210-212

デフォルトのセキュリティレベル, 97

ドメイン (NIS+ コマンド), 175-182

ドメインの追加, 121, 123

ドメイン名, 101

複製 (NIS+), 復元による設定, 407

設定 (続き)

複製サーバー (NIS+ コマンド), 166-172

複製の設定と nisping, 171-172, 172

複製の設定と nisrestore, 169-170

ブロードキャストによる初期設定 (NIS+), 157

ホスト名, 101

ホスト名による初期化 (NIS+), 158-159, 159

ルートサーバー (スクリプト), 95-100, 96-98, 98-99

ルートドメイン (NIS+ コマンド), 131-147, 132-145

そ

その他クラス, 269

その他クラス, 221, 223, 268

た

タイムスタンプ, 231

ち

チェックポイント設定, 518

置換 (NIS+ テーブル), 186

つ

追加 (NIS+ テーブル), 186

て

ディスク容量の不足 (NIS+), 461

ディレクトリ, 518

ディレクトリキャッシュ, 67, 518

データ暗号化鍵, 518

テーブル, 518

テーブル (NIS+), 206

インデックス名, 76

エントリ, 207-208

エントリ名, 76

オートマウントマップの追加, 207

テーブル (NIS+) (続き)

- 検索パス, 208-210
- 更新, 211-212
- 構造, 205-210
- 設定, 210-212
- 名前, 75-76
- リンクの作成, 372-373
- 列, 207-208

テーブルの生成 (populate), 518

デーモン

- NIS+, チェックポイントの設定, 344-345
- nscd, 39
- nscd デーモン, 136
- rpc.nisd, 338-339
- rpc.nisd, DNS 転送, 338-339
- rpc.nisd, NIS 互換, 338-339
- rpc.nisd, 起動, 338
- rpc.nisd, セキュリティレベル, 338-339
- rpc.nisd, 停止, 338-339
- rpc.nisd デーモン, 85
- rpc.nisd デーモン, 99
- rpc.nisd の失敗, 439
- rpc.nisd の終了, 458
- rpc.nisd, 問題, 438

と

独自のテーブル, 84

ドット表記, 518

ドメイン, 60-61, 518

- NIS+ ドメイン名, 74-75
- NIS と NIS+ の共存, 55
- passwd, 300
- 管理グループ, 178
- 設定 (NIS+ コマンド), 175-182
- 設定 (スクリプト), 121, 123
- 名前, 80, 96, 101
- ホストのドメインの変更, 155-156
- ルート、設定, 95-100

ドメイン名, 519

- 変更 (NIS+), 453-454

トランザクションログ

- nislog, 346-347
- XID, 346
- 内容を表示する, 346-347

トランザクションログの中の XID, 346

な

名前解決, 519

名前空間, 519

- NIS+ の準備, 79-80

に

入力ファイル, 211

認証, 50

認証, 519

- 主体、認証方法, 229-232, 230
- タイムスタンプ, 231

ね

ネーミング

- NIS+, 56-57

- NIS+ ディレクトリ, 59-60

- NIS+ の構造, 57-58

ネームサーバー, 519

ネームサービス, 519

ネームサービススイッチ, 519

ネットワークパスワード, 519

ネットワークマスク, 519

は

パスワード

- Login incorrect メッセージ, 292

- NIS+** 環境, 298

- nistbladm, 300-303

- not have secure RPC credentials
メッセージ, 445

- nsswitch.conf ファイル, 296, 297-298,
462

- nsswitch.conf ファイルのオーバーラ
イド, 297-298

- passwd, 297-300

- Permission denied メッセージ, 292

- rlogin 問題, 463

- root のパスワードを変更したための問
題, 456

- root、変更, 294

- Secure パスワードとログインパスワードが異
なる, 454-455

パスワード (続き)

- Sorry: less than メッセージ, 293
- will expire メッセージ, 292
- 新しいパスワードが使えない, 462-463
- 管理, 296-318
 - 休暇の場合のロック, 307
 - 強制的な変更, 308
 - 警告期間、設定, 310-311
 - 最大ログイン所要時間, 317-318
 - 試行回数の制限, 317
 - 使用, 291-295
 - 使用期間, 307-314
 - 使用期間、設定の解除, 311
 - 使用規則、デフォルトの設定, 315-318
 - 使用権の有効期限、解除, 313
 - 使用権 (ユーザー), 312-313
 - 選択, 294-295
 - デフォルトの使用規則、設定, 315-318
- パスワードを変更できない, 463
- 必要条件, 295
- 変更, 293-294, 305-306
- 変更禁止期間、設定, 309-310
- 有効期間, 308-309, 312
- 有効期限, 308
- 有効期限、設定, 312-313
- 有効期限 (ユーザー), 312
- ユーザーの問題, 462
- ログイン, 291-292
- ログインの間隔、設定, 313-314, 314
- ログインの失敗, 317-318
- ロック, 306-307
- ロックの解除, 307

パスワードコマンド, 218

パスワード情報

- nsswitch.conf ファイル, 40-41, 297-298
- nsswitch.conf ファイルのオーバーライド, 297-298
- Secure RPC パスワード, 235-236
- Secure RPC パスワードとログインパスワードの違い, 235-236
- 間隔, 302
- 期限 値, 302
- 警告, 302
- 最小値, 301
- 最大値, 301
- シャドウ列のフィールド, 301-303
- 日数, 303
- 表示, 304-305

パスワード情報 (続き)

- 未使用 values, 302
- ログインパスワード, 235-236
- ログインパスワードとSecure RPC パスワードの違い, 235-236

パスワードデータ

- passwd column へのアクセスの制限, 198-200

バックアップと復元 (NIS+), 399-405

- XDR コード化, 404
- 上書き, 402
- サーバーの置換, 408-409
- サーバーの複製, 408-409
- サブディレクトリ, 400
- サブドメイン, 401
- 実行されないデータチェック, 400
- 自動化, 402
- 全名前空間, 401, 403
- 転送先ディレクトリ, 402
- 特定のディレクトリ, 402-403
- バックアップディレクトリ, 403-404
- バックアップファイル, 404-405
- 日付順, 402
- ファイルシステムのバックアップ, 402
- 復元, 405-407
- マスターサーバーのみ, 400, 401
- マスター上だけのデータ, 401

パフォーマンス, NIS+, 166

ひ

- 非公開鍵, 520

ふ

- フィールド, 270
- 複製サーバー, 520
- ブロードキャストによる初期設定 (NIS+), 157
- プロセス数の不足 (NIS+), 461

ほ

- ホスト (データ)
 - 名前, 96
 - 名前内部のドット, 80

ホスト (マシン)
NIS+ のホスト名, 76-77
ドメインの変更, 155-156
名前, 80, 96
ホスト名による初期設定 (NIS+), 158-159, 159

ま
マージ (NIS+ テーブル), 186
マスターサーバー, 520

み
未認証クラス, 269
未認証クラス, 221, 223

め
メール交換レコード, 520
メールホスト, 520
メッセージ (NIS+), 464
メモリーの不足 (NIS+), 460-461

ゆ
ユーザー
初期設定 (NIS+), 109-110, 127-128, 128
優先サーバー, 520
優先サーバーリスト, 520
優先順位の番号, 520

り
リポジトリ
更新, 41
複数のリポジトリを使用する, 41
リモートサイト, NIS+, 167

る
ルートサーバー
初期設定, 340-341

ルートサーバー (続き)
設定 (NIS+ コマンド), 131-147, 132-145
設定 (スクリプト), 95-100, 96-98, 98-99
ルートドメイン, 520
NIS 互換モード, 132, 176
インターネットルートドメイン, 95-96
設定 (NIS+ コマンド), 131-147, 132-145
設定 (スクリプト), 95-96, 96-98
名前, 95
ルート複製サーバー, 520
ルートマスターサーバー, 520

れ
レコード, 520

ろ
ローカルエリアネットワーク (LAN), 520
ログイン, 291-292

