



# Solaris ネーミングの設定と構成

---

Sun Microsystems, Inc.  
901 San Antonio Road  
Palo Alto, CA 94303  
U.S.A. 650-960-1300

Part Number 806-2722-10  
2000年3月

Copyright 2000 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303-4900 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。

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a).

本製品に含まれる HG 明朝 L と HG ゴシック B は、株式会社リコーがリコービイ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イスをもとに作成されたものです。平成明朝体 W3 は、株式会社リコーが財団法人 日本規格協会 文字フォント開発・普及センターからライセンス供与されたタイプフェイスをもとに作成されたものです。また、HG 明朝 L と HG ゴシック B の補助漢字部分は、平成明朝体 W3 の補助漢字を使用しています。なお、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 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) の商標もしくは登録商標です。

サンロゴマークおよび 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、オムロン株式会社が共同開発されたソフトウェアです。

Wnn6 は、オムロン株式会社が開発されたソフトウェアです。(Copyright OMRON Co., Ltd. 1999 All Rights Reserved.)

「ATOK」は、株式会社ジャストシステムの登録商標です。

「ATOK8」は株式会社ジャストシステムの著作物であり、「ATOK8」にかかる著作権その他の権利は、すべて株式会社ジャストシステムに帰属します。

「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。

本製品に含まれる郵便番号辞書 (7 桁/5 桁) は郵政省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なっています)。

本製品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'98』に添付のものを使用しています。© 1997 ビレッジセンター

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。

DtComboBox ウィジェットと DtSpinBox ウィジェットのプログラムおよびドキュメントは、Interleaf, Inc. から提供されたものです。(© 1993 Interleaf, Inc.)

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ないものとします。

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: Solaris Naming Setup and Configuration Guide

Part No: 806-1386-10

Revision A



# 目次

---

はじめに 13

パートI ネームサービスの設定

1. ネームサービススイッチの設定 21

ネームサービススイッチ 21

nsswitch.conf ファイル 21

    NIS+ 用デフォルトスイッチファイル 22

    NIS 用デフォルトスイッチファイル 23

    ファイル用デフォルトスイッチファイル 24

    LDAP 用デフォルトスイッチファイル 25

構成ファイルの変更 25

    セキュリティ上の留意点 26

    ネームサービススイッチを設定する 26

▼ 構成ファイルを変更する方法 26

DNS を使用できるようにする 27

    前提条件 28

    セキュリティ上の留意点 28

    DNS を使用できるようにする方法 - タスクマップ 28

▼ NIS+ クライアントで DNS を使用できるようにする方法 28

+/- 構文との互換性を確保する 29

- +/- 構文との互換性を追加する - タスクマップ 29
- セキュリティ上の留意点 30
- ▼ +/- 構文との互換性を追加する方法 30
- IPv6 アドレスを使用できるようにする 31
  - IPv6 を使用できるようにする方法 - タスクマップ 31
- ▼ NIS+ クライアントで IPv6 を使用できるようにする方法 32
- パートII NIS+ の設定と構成
- 2. NIS+ 入門 35
  - NIS+ の概要 35
  - 設定および構成の前に 36
  - 名前空間がすでに存在する場合の設定 37
  - 2 通りの構成方法 39
- 3. NIS+ 設定スクリプト - 紹介 41
  - NIS+ スクリプトについて 41
  - NIS+ スクリプトで実行すること 42
  - NIS+ スクリプトでは実行しないこと 42
- 4. スクリプトを使用した NIS+ の設定 45
  - NIS+ 設定の概要 46
  - NIS+ 名前空間サンプルの作成 48
    - NIS+ スクリプトのコマンド行の要約 49
  - NIS+ ルートサーバーの設定 51
    - nisserver を実行するための前提条件 52
  - ▼ ルートマスターサーバーを作成する方法 53
  - ▼ 誤った情報を変更する方法 56
  - ▼ Multihomed NIS+ ルートマスターサーバーの設定方法 57
  - NIS+ テーブルの生成 (populate) 59
    - nispopulate を実行するための前提条件 59
  - ▼ ルートマスターサーバーのテーブルを生成する方法 61

NIS+ クライアントマシンの設定	67
<code>nisclient</code> を実行するための前提条件	68
▼ 新しいクライアントマシンを初期設定する方法	68
クライアントマシンの追加作成	70
NIS+ クライアントユーザーの初期設定	71
<code>nisclient(1M)</code> を実行するための前提条件	71
▼ NIS+ ユーザーを初期設定する方法	72
NIS+ サーバーの設定	72
<code>rpc.nisd</code> を実行するための前提条件	73
クライアントを NIS+ サーバーとして構成する方法	74
サーバーの追加作成	75
ルート複製サーバーの作成	75
<code>nisserver</code> を実行するための前提条件	76
▼ ルート複製サーバーを作成する方法	77
▼ Multihomed NIS+ 複製サーバーの設定方法	79
サブドメインの作成	81
<code>nisserver</code> を実行するための前提条件	81
▼ 新しい非ルートドメインを作成する方法	82
ドメインの追加作成	84
新しいサブドメインのテーブルの生成	84
<code>nispopulate</code> を実行するための前提条件	85
マスターサーバーテーブルを生成する方法	87
サブドメイン複製サーバーの作成	88
<code>nisserver</code> を実行するための前提条件	88
▼ 複製サーバーを作成する方法	89
サブドメインの NIS+ クライアントマシンの初期設定	89
<code>nisclient</code> を実行するための前提条件	90
▼ 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	91

- サブドメインの NIS+ クライアントユーザーの初期設定 91
  - nisclient(1M) を実行するための前提条件 91
  - ▼ NIS+ サブドメインユーザーを初期設定する方法 92
  - NIS+ 名前空間サンプルで使用したコマンドのまとめ 92
- 5. ルートドメインの設定 95
  - ルートドメインの設定方法の概要 95
  - 標準構成と NIS 互換構成の手順の相違 96
  - ルートドメインの確立 96
    - 手順の要約 97
    - ルートドメインを確立する — タスクマップ 98
    - セキュリティ上の留意点 98
    - 前提条件 98
    - 必要な情報 99
  - ▼ ルートドメインを構成する方法 99
  - ルートドメイン構成の要覧 117
- 6. NIS+ クライアントの構成 119
  - NIS+ クライアントの設定方法の概要 119
  - クライアントを構成する 120
    - セキュリティ上の留意点 121
    - 前提条件 122
    - 必要な情報 123
    - クライアントの設定 — タスクマップ 123
  - ▼ NIS+ クライアントを設定する方法 123
  - ワークステーションのドメイン名を変更する 127
    - セキュリティ上の留意点 127
    - 必要な情報 127
    - ワークステーションのドメインの変更 — タスクマップ 127
  - ▼ クライアントのドメイン名を変更する方法 128

- NIS+ クライアントを初期設定する 129
  - ブロードキャストにより初期設定する 130
  - ホスト名により NIS+ クライアントを初期設定する 131
  - 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 133
- NIS+ クライアント構成の要覧 135
- 7. **NIS+ サーバーの構成 137**
  - NIS+ サーバーを設定する 137
    - 標準構成と NIS 互換構成の手順の相違 138
    - セキュリティ上の留意点 138
    - 前提条件 139
    - 必要な情報 139
    - NIS+ サーバーを初期設定する — タスクマップ 139
  - ▼ NIS+ サーバーを構成する方法 139
    - 既存のドメインに複製サーバーを追加する 142
      - NIS+ コマンドによる複製サーバーの構成 144
      - NIS+ コマンドを使って複製サーバーを構成する— タスクマップ 145
    - ▼ NIS+ コマンドを使って複製サーバーを構成する 145
      - nisrestore を使ってデータを複製サーバーにロードする 146
      - nisrestore を使ってデータを複製サーバーにロードする — タスクマップ 147
    - ▼ nisrestore を使って名前空間データをロードする 147
      - nisping を使ってデータを複製サーバーにロードする 148
      - nisping を使ってデータを複製サーバーにロードする — タスクマップ 149
    - ▼ nisping を使って名前空間データをロードする方法 149
  - サーバー構成の要覧 150
- 8. **非ルートドメインの構成 153**
  - 非ルートドメインを設定する 153

- 標準構成と NIS 互換構成の手順の相違 154
- セキュリティ上の留意点 155
- 前提条件 155
- 必要な情報 156
- 非ルートドメインを設定する — タスクマップ 156
- ▼ 非ルートドメインを設定する方法 156
- サブドメイン構成の要覧 163
- 9. **NIS+ テーブルの設定 165**
- テーブルの設定 165
- テーブルの生成の方法 166
- NIS+ テーブルをファイルから生成する方法 167
  - ファイルのセキュリティ上の留意点 168
  - 前提条件 168
  - 必要な情報 169
  - NIS+ テーブルをファイルから生成する — タスクマップ 169
- ▼ NIS+ テーブルをファイルから生成する方法 169
- NIS+ テーブルを NIS マップから生成する方法 175
  - マップのセキュリティ上の留意点 176
  - 前提条件 176
  - 必要な情報 177
  - NIS+ テーブルを NIS マップから生成する — タスクマップ 177
- ▼ NIS+ テーブルをマップから生成する方法 177
- NIS+ から NIS に情報を転送する 182
  - NIS から NIS+ に情報を転送する際のセキュリティ上の留意点 182
  - 前提条件 182
  - NIS+ から NIS に情報を転送する — タスクマップ 182
- ▼ NIS+ から NIS へ情報を転送する方法 182
- 所有者および管理者に対する Passwd へのアクセス制限 183



	Passwd 列のセキュリティ上の留意点	184
	前提条件	184
	必要な情報	184
	所有者および管理者に対する Passwd 列へのアクセス制限 — タスクマップ	184
▼	パスワード列へのアクセスを制限する方法	185
	テーブルの生成のまとめ	186
	パートIII NIS の設定と構成	
10.	NIS サービスの構成	193
	Solaris 8 オペレーティング環境の NIS	193
	NIS と IPv6	194
	NIS の構成を始める前に	194
	NISドメインの計画	194
	ドメインの計画	194
	NIS サーバーの選択	195
	NIS クライアントマシンの選択	195
	NIS の構成手順	195
	マスターサーバーの準備	196
	入力ファイルディレクトリ	196
	passwd ファイルと名前空間のセキュリティ	197
	マスターサーバーの準備 — タスクマップ	197
▼	NIS マップへの変換用ソースファイルの準備	198
	Makefile の準備	199
▼	ypinit によるマスターサーバーの設定	210
	マスターサーバー上での NIS サービスの開始	212
	NIS サービスの自動スタート	212
	コマンド行からの NIS の開始	212
	DNS の転送	213

- ypstop による NIS の停止 213
  - NIS スレーブサーバーの設定 213
    - スレーブサーバーの準備 214
    - NIS スレーブサーバーの設定 — タスクマップ 214
    - スレーブサーバーの設定 214
    - スレーブサーバー上での NIS サービスの開始 216
  - NIS クライアントの設定 216
    - NIS を使用するマシンの構成 216
  - NIS マップ 217
    - デフォルトの NIS マップ 217
    - NIS マップの修正 220
  - NIS の管理、問題解決、エラーメッセージ 220
  - パートIV **FNS** の設定と構成
- 11. **FNS** の設定と構成 223
  - FNS の設定 - 概要 224
  - リソース条件の決定 224
  - FNS 用の名前空間の準備 226
    - FNS 用の名前空間の準備 — タスクマップ 226
  - ▼ FNS 用の NIS+ サービスの準備 226
  - ▼ FNS 用の NIS サービスの準備 228
    - FNS 用のファイルを使用したネームサービスの準備 229
  - グローバルな FNS の名前空間コンテキストの作成 229
    - グローバルな FNS の名前空間コンテキストの作成 — タスクマップ 230
  - ▼ NIS+ の下での名前空間コンテキストの作成 230
  - ▼ NIS の下での名前空間コンテキストの作成 231
  - ▼ ローカルファイルの下での名前空間コンテキストの作成 232
  - FNS サービスの複製 233

	FNS サービスの複製 — タスクマップ	233
	▼ NIS+ の下での FNS の複製	233
	▼ NIS の下での FNS の複製	234
	▼ ファイルを使用したネームサービスの下での FNS の複製	235
	FNS の管理、問題解決、エラーメッセージ	236
	パートV DNS の設定と構成	
12.	DNS クライアントの設定	239
	Solaris DNS BIND の実装	240
	DNS サービスの設定	240
	クライアントの設定	241
	リゾルバ	241
	resolv.conf ファイルの作成	242
	/etc/nsswitch.conf ファイルの修正	243
13.	DNS サーバーの設定	245
	DNS サーバーの設定	245
	▼ DNS サーバーの設定	245
	サーバーの構成とデータファイルの名前	247
	構成ファイル	247
	DNS データファイルの名前	247
	データファイル	249
	\$INCLUDE ファイル	250
	ドメイン名	250
	デフォルトのドメイン名	250
	ドメイン名の末尾のドットについて	251
	resolv.conf ファイル	251
	▼ resolv.conf ファイルの設定	251
	named.conf ファイル	252
	BIND 4.9.x から BIND 8.1 への移行	255

サーバー機能の指定	255
主マスターサーバーの指定	256
副マスターサーバーの指定	258
キャッシュオンリーサーバーの指定	259
データファイルの設定	260
資源レコードの種類	260
named.ca ファイルの設定	261
hosts ファイルの設定	264
hosts.rev ファイルの設定	265
named.local ファイルの設定	266
サーバーの初期設定	267
設定の確認	267
サブドメインの設定	269
単一ゾーンのサブドメインの設定	270
複数ゾーンのサブドメインの設定	271
非インターネットルートマスターの設定	272
実例	273
サンプルブートファイル	274
サンプル resolv.conf ファイル	275
サンプル named.local ファイル	276
サンプル hosts ファイル	277
サンプル hosts.rev ファイル	278
サンプル named.ca ファイル	279
索引	281

## はじめに

---

このマニュアルでは、NIS+、NIS、FNS、LDAP、DNS の各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。このマニュアルは、Solaris™ 8 リリースシステム管理マニュアルセットの一部です。

---

### 対象読者

このマニュアルは、NIS+、NIS、FNS、DNS という 4 つの Solaris ネームサービスのどれか 1 つまたは複数を設定したいと考えているシステム管理者とネットワーク管理者を対象としています。読者にはシステム管理者としての経験があるものと想定しています。

このマニュアルでは Solaris ネームサービスに関連する概念を紹介しますが、ネットワークの基礎や、Solaris 環境の提供する管理ツールについては説明していません。ネットワークを管理する場合、このマニュアルでは、読者がすでにネットワークの動作に関する知識があり、好みのツールも選択しているものと想定しています。

(Solaris ネームサービス、ネームサービス関連用語、一般的なエラーメッセージの詳細は、『Solaris ネーミングの管理』を参照してください。)

---

### このマニュアルの構成

このマニュアルは、次の 5 部構成になっています。

## パート I, ネームサービスの設定

パート I では、`nsswitch.conf` を使用して、複数の異なるネームサービスを同時に利用する方法について説明します。

- 第 1 章：ネームサービススイッチとその詳細な構成手順について説明します。

## パート II, NIS+ の設定と構成

パート II では、NIS+ 名前空間を設定および構成する方法について説明します。第 2 章では、NIS+ とは何かを紹介합니다。第 3 章と第 4 章では、NIS+ 設定スクリプトを使って NIS+ 名前空間を設定する方法を説明します。第 5 章から第 9 章では、NIS+ の各種コマンドを使って NIS+ 名前空間を設定する方法を説明します。

- 第 2 章：NIS+ (Network Information Service Plus) の概要を説明し、NIS+ を設定する前に必要な作業を列挙します。さらに、NIS+ 名前空間の最低要件を明らかにし、NIS+ の設定方法を 2 通り紹介します。
- 第 3 章：NIS+ スクリプトの概要と、スクリプトを使ってできることとできないことを説明します。
- 第 4 章：NIS+ スクリプト (`nissserver`、`nispopulate`、`nisclient`) といくつかの NIS+ コマンドを組み合わせて基本的な NIS+ 名前空間を構成する方法について説明します。
- 第 5 章：NIS+ コマンドを使用してルートドメインと DES 認証を設定する詳細な方法について説明します。
- 第 6 章：NIS+ コマンドを使って NIS+ クライアントを設定する手順、および 3 通りの初期設定方式について説明します。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。
- 第 7 章：NIS+ コマンドを使って NIS+ (ルートマスターサーバーを除く) サーバーを設定し、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。
- 第 8 章：NIS+ コマンドを使ってサブドメイン (非ルートドメインともいう) を設定する手順を説明します。この手順には、マスターサーバーと複製サーバーを指定する方法も含まれます。
- 第 9 章：NIS+ コマンドを使ってマスターサーバー上で `/etc` ファイルまたは NIS マップから NIS+ テーブル生成する手順、NIS+ テーブルから NIS マップに情報を戻す方法、さらに `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。

## パート III, NIS の設定と構成

第 10 章：NIS (Network Information Service) の初期設定と初期構成について説明します。

## パート IV, FNS の設定と構成

第 11 章：NIS+、NIS、/etc のそれぞれの名前空間環境において、FNS (Federated Naming Service) の初期設定および管理の方法について説明します。

## パート V, DNS の設定と構成

このパートでは、DNS クライアントと DNS サーバーの設定方法について説明します。

- 第 12 章：DNS クライアントマシン上の DNS サービスの設定方法について説明します。
- 第 13 章：DNS ネームサーバーの設定方法について説明します。

---

## 関連マニュアル

NIS+、DNS に関してこのマニュアルの範囲を超えることがらについては、次に示すマニュアルを参照してください。これらのマニュアルは、日本語 Solaris 8 リリースシステム管理マニュアルセットの一部です。

- 『Solaris ネーミングの管理』 - すでに存在する NIS+ 名前空間のカスタマイズおよび管理の方法を説明しています。
- 『NIS+ への移行』 - NIS から NIS+ に移行する方法について説明しています。

Solaris 8 リリースマニュアルセット以外のマニュアル

- 『DNS and Bind』 - Cricket Liu、Paul Albitz 共著、浅羽登志也／上水流由香 監訳、アスキー出版局、1995年
- 『NFS and NIS』 - Hal Stern 著、君骨彰訳、砂原秀樹 監訳、アスキー出版局、1992年

---

## Sun のマニュアルの注文方法

専門書を扱うインターネットの書店 Fatbrain.com から、米国 Sun Microsystems™, Inc. (以降、Sun™ とします) のマニュアルをご注文いただけます。

マニュアルのリストと注文方法については、<http://www1.fatbrain.com/documentation/sun> の Sun Documentation Center をご覧ください。

---

## Sun のオンラインマニュアル

<http://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、検索をおこなうこともできます。

---

## 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、または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は区別して示します。	system% <b>su</b> password:
AaBbCc123	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。



表 P-1 表記上の規則 続く

字体または記号	意味	例
『 』	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、または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越える場合、バックスラッシュは継続を示します。	<pre>sun% grep `^#define \ XV_VERSION_STRING'</pre>

ただし AnswerBook2™ では、ユーザーが入力する文字と画面上のコンピュータ出力は区別して表示されません。

コード例は次のように表示されます。

■ C シェルプロンプト

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

■ Bourne シェルおよび Korn シェルのプロンプト

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

■ スーパーユーザーのプロンプト

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

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

| 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。

---

## 一般規則

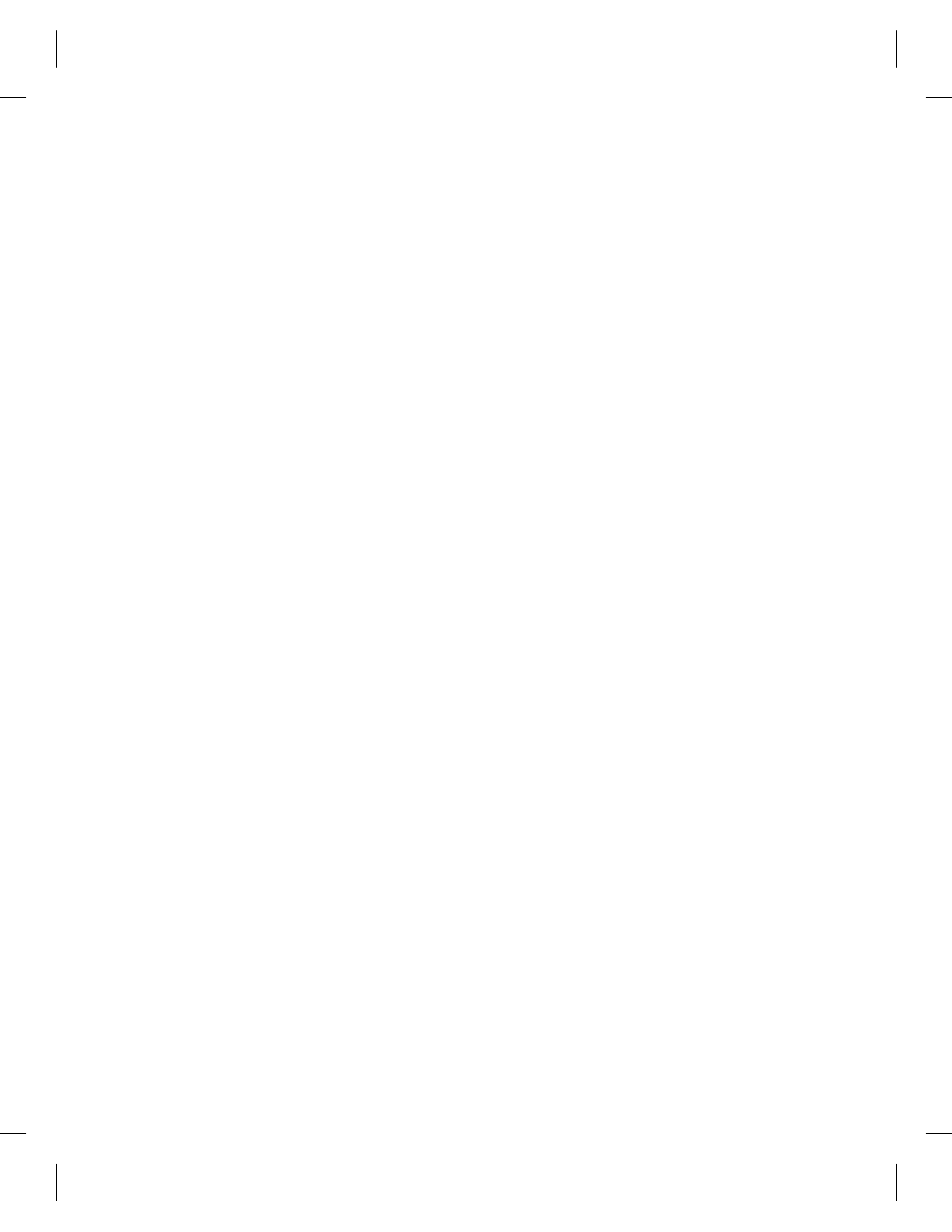
- このマニュアルでは、「IA」という用語は、Intel 32 ビットのプロセッサアーキテクチャを意味します。これには、Pentium、Pentium Pro、Pentium II、Pentium II Xeon、Celeron、Pentium III、Pentium III Xeon の各プロセッサ、および AMD、Cyrix が提供する互換マイクロプロセッサチップが含まれます。

## パート I | ネームサービスの設定

---

パート I では、`nsswitch.conf` を使用して、複数の異なるネームサービスを同時に使用する方法について説明します。

- 第 1 章



## ネームサービススイッチの設定

---

この章では、ネームサービススイッチとその構成手順について説明します。

---

### ネームサービススイッチ

ネームサービススイッチは、クライアントワークステーションやアプリケーションのネットワーク情報の入手方法を制御するためのもので、単に「スイッチ」ともいいます。アプリケーションは、このスイッチの内容を基に、どのネームサービスをどの順番で使うかを判断し、名前に関する情報を入手します。スイッチの実体は `nsswitch.conf` というファイルで、これは各マシン (ワークステーション) の `/etc` ディレクトリに格納されています。

---

### `nsswitch.conf` ファイル

各ワークステーションの `/etc` ディレクトリには `nsswitch.conf` ファイルがあります。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。`nsswitch.conf` ファイルの詳細は、『Solaris ネーミングの管理』を参照してください。

/etc/nsswitch.conf ファイルは、Solaris 8 リリースをインストールすると自動的に作成されます。また次の 4 つ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
- /etc/nsswitch.ldap

これらの 4 つの代替テンプレートファイルには、それぞれネットワーク情報の情報源として NIS+、NIS、ローカルファイル、または LDAP を使用する標準的なスイッチ構成が設定されています。DNS 用のテンプレートファイルは用意されていませんが、いずれの代替テンプレートファイルも DNS 用に編集して使用できます。詳細は、27ページの「DNS を使用できるようにする」を参照してください。Solaris オペレーティング環境をワークステーションに最初にインストールする際、インストーラはそのワークステーショ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。インストールの途中、選択されたネームサービスに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、NIS+ を使う場合、インストーラは nsswitch.nisplus を nsswitch.conf にコピーします。

インターネットに接続したネットワークの中で、DNS サービスを使ってインターネット上のホストにアクセスできる環境をユーザーに提供したいのであれば、27ページの「DNS を使用できるようにする」の説明に従って DNS 転送を可能にする必要があります。

通常の名前空間であれば (DNS の利用に関しては前述のとおり)、nsswitch.conf にコピーされたデフォルトのテンプレートファイルで十分に用が足りるはずです。

## NIS+ 用デフォルトスイッチファイル

Solaris 8 リリースで提供される NIS+ 用スイッチファイルは、nsswitch.nisplus です。

例 1-1 デフォルトの nsswitch.nisplus ファイル

```
#
# /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.
```

(続く)

```

#
# "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
# 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

```

## NIS 用デフォルトスイッチファイル

Solaris 8 リリースで提供される NIS 用スイッチファイルは、`nsswitch.nis` です。

例 1-2 デフォルトの `nsswitch.nis` ファイル

```

#
# /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

```

(続く)

```
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
```

## ファイル用デフォルトスイッチファイル

Solaris 8 リリースで提供される ローカルファイル用スイッチファイルは `nsswitch.files` です。

### 例 1-3 デフォルトの `nsswitch.files` ファイル

```
#
# /etc/nsswitch.files:
#
# 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 won't use netgroups at all.

netgroup: files
automount: files
aliases: files
services: files
sendmailvars: files
```



## LDAP 用デフォルトスイッチファイル

Solaris オペレーティング環境で提供される LDAP 用スイッチファイルは、`nsswitch.ldap` です。

例 1-4 LDAP 用スイッチファイルのテンプレート

```
#
# /etc/nsswitch.ldap:
#
# An example file that could be copied over to /etc/nsswitch.conf; it
# uses LDAP 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 ldap
group:       files ldap

hosts:       ldap [NOTFOUND=return] files

networks:    ldap [NOTFOUND=return] files
protocols:   ldap [NOTFOUND=return] files
rpc:         ldap [NOTFOUND=return] files
ethers:      ldap [NOTFOUND=return] files
netmasks:   ldap [NOTFOUND=return] files
bootparams:  ldap [NOTFOUND=return] files
publickey:   ldap [NOTFOUND=return] files

netgroup:    ldap

automount:   files ldap
aliases:     files ldap

# for efficient getservbyname() avoid ldap
services:    files ldap
sendmailvars: files
```

---

## 構成ファイルの変更

ワークステーションのネームサービスを変更する際は、そのワークステーションのスイッチファイルを新しいネームサービスに対応したものに変更する必要があります。たとえば、ワークステーションのネームサービスを NIS から NIS+ に変更する

場合、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+ をワークステーションにインストール  
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。

## セキュリティ上の留意点

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。

## ネームサービススイッチを設定する

表 1-1 タスクマップ：ネームサービススイッチを設定する

タスク	説明	指示の参照先
ネームサービス スイッチを設定する	NIS+ またはNIS 用の構成ファイルを設定 し、ワークステーションをリポートする	26ページの「構成ファイ ルを変更する方法」

### ▼ 構成ファイルを変更する方法

スイッチファイルの変更は次の手順で行います。

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 /etc ファイル用」

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

### 3. ワークステーションを再起動します。

nscd ネームサービスキャッシュデーモンはスイッチ情報をキャッシュに書き込みます。また、ライブラリ関数には nsswitch.conf ファイルが変更されてもスイッチ情報を読み直さないものがあります。したがって、ワークステーションを再起動して、nscd とこれらのライブラリ関数が最新スイッチの情報を持つようにする必要があります。

---

## DNS を使用できるようにする

ここでは、ローカルファイル用または NIS+ 用のネームサービススイッチ構成ファイルを設定して、ローカルファイルまたは NIS+ と共に DNS (Domain Name System) を使用できるようにする方法を説明します。NIS には DNS を参照する機能がもともと組み込まれているので、NIS を使用しているマシンではスイッチファイルの hosts 行に DNS のためのエントリを追加する必要はなく、追加すべきでもあ

りません。ここで説明する手順は、`/etc` のローカルファイルか NIS+ を使用するマシンにだけ適用されます。`/etc` のローカルファイルまたは NIS+ を使っている場合の手順を次に示します。

## 前提条件

DNS を使用するマシンでは `/etc/resolv.conf` を正しく構成する必要があります。詳しくは、241ページの「リゾルバ」を参照してください。

## セキュリティ上の留意点

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。

## DNS を使用できるようにする方法 - タスクマップ

表 1-2 DNS を使用できるようにする方法

タスク	説明	指示の参照先
DNS を使用できるようにする方法	<code>/etc/nsswitch.conf</code> ファイルを修正して、NIS+ クライアントが DNS を使用できるようにする	28ページの「NIS+ クライアントで DNS を使用できるようにする方法」

### ▼ NIS+ クライアントで DNS 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` ファイルをオープンします。
3. ホスト情報のソースとして **DNS** を指定します。

DNS は、ホスト情報の情報源として、「他の情報源」と共に使用できません。`/etc/nsswitch.conf` の `hosts` 行を次のいずれかのように変更して、DNS をホスト情報の情報源とすることを指定します。

```
hosts: files dns
```

または

```
hosts: nisplus dns [NOTFOUND=return] files
```

または

```
hosts: dns nisplus [NOTFOUND=return] files
```

NIS クライアントの場合は、上記の指定をしないでください。この指定をすると名前を見つけることができない場合に、2 度 DNS から検索することになります。

4. /etc/nsswitch.conf ファイルを保存し、ワークステーションを再起動します。

nscd デーモンは、その起動時にスイッチ情報を読み込み、キャッシュに書き込むため、ここでワークステーショ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

ここでは、ネームサービスとして主に NIS または NIS+ を使用する場合に、/etc/passwd、/etc/shadow、/etc/group の各ファイルで使用される +/- 構文との互換性を確保する方法について説明します。

## +/- 構文との互換性を追加する - タスクマップ

表 1-3 +/- 構文との互換性を追加する

タスク	説明	指示の参照先
+/- 構文との互換性を追加する	/etc/passwd、/etc/shadow、および /etc/group ファイルを修正して、+/- 構文との DNS の互換性を追加する	30ページの「+/- 構文との互換性を追加する方法」

## セキュリティ上の留意点

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。

---

注 - NIS 互換性モードで実行されている NIS+ サーバーをサーバーとするクライアントマシン上で作業しているユーザーは、netgroup テーブルに対して ypcat を実行できません。実行した場合、エントリがあってもテーブルが空の場合と同じ結果となります。

---

### ▼ +/- 構文との互換性を追加する方法

1. スーパーユーザーとしてログインします。
2. /etc/nsswitch.conf ファイルをオープンします。
3. **passwd** と **group** の情報源を **compat** に変更します。
  - NIS を使う場合は次のように入力します。

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

- NIS+ を使う場合は次のように入力します。

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

これにより Solaris 1.x リリースと同じ構文を使用できます。ファイル内の +/- エントリに従って、/etc と NIS マップ (または NIS+ テーブル) を検索します。

4. + または + netgroup を /etc/passwd、/etc/shadow、/etc/group の各ファイルに追加します。



---

注意 - + または + netgroup のエントリを /etc/shadow および /etc/passwd に追加しないと、ローカルファイルに登録されていないユーザーは以後ログインできなくなります。

---

5. /etc/nsswitch.conf ファイルを保存し、ワークステーションを再起動します。

ライブラリ関数には nsswitch.conf ファイルが変更されてもスイッチ情報を読み直さないものがあります。したがって、ワークステーションを再起動して、nscd とこれらのライブラリ関数が最新スイッチの情報を持つようにする必要があります。

---

## IPv6 アドレスを使用できるようにする

nsswitch.conf ファイルは、IPv6 アドレスの検索基準を制御します。IPv6 は、32 ビットから 128 ビットまで IP アドレスサイズを大きくして、より多くのアドレス階層をサポートし、より多くのノードにアドレス指定できるようにします。IPv6 の構成と実装の詳細は、『Solaris のシステム管理 (第 3 巻)』の「IPv6 の概要」と「IPv4 から IPv6 への移行」を参照してください。

/etc/inet/ipnodes ファイルには、IPv4 と IPv6 のアドレスが格納されています。/etc/inet/ipnodes ファイルは、/etc/hosts ファイルと同じフォーマットを使用します。

## IPv6 を使用できるようにする方法 - タスクマップ

表 1-4 IPv6 を使用できるようにする方法

タスク	説明	指示の参照先
IPv6 を使用できるようにする	/etc/nsswitch.conf ファイルを修正して、NIS+ クライアントが IPv6 を使用できるようにする	32ページの「NIS+ クライアントで IPv6 を使用できるようにする方法」

## ▼ 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 を使用してください。このデフォルトを使用しない場合には、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 (ブート時の遅延など)。

4. ファイルを保存して、ワークステーションをリブートします。  
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、ここでワークステーションをリブートする必要があります。



## パート II NIS+ の設定と構成

---

パート II では、NIS+ 名前空間を設定および構成する方法について説明します。

- 第 2 章では、NIS+ とは何かを紹介します。
- 第 3 章と第 4 章では、NIS+ 設定スクリプトを使って NIS+ 名前空間を設定する方法を説明します。NIS+ の設定と構成には、NIS+ 設定スクリプトを使うと便利です。
- 第 5 章から第 9 章では、NIS+ の各種コマンドを使って NIS+ 名前空間を設定する方法を説明します。

パート II は次の 8 つの章で構成されています。

- 第 2 章
- 第 3 章
- 第 4 章
- 第 5 章
- 第 6 章
- 第 7 章
- 第 8 章
- 第 9 章



## NIS+ 入門

---

この章では、「Network Information Service Plus (NIS+)」の概要を説明します。また NIS+ を設定する前に行うべき作業を列挙し、NIS+ 名前空間の最低要件について確認したうえで、NIS+ の設定方法を 2 通り紹介します。

- 35ページの「NIS+ の概要」
- 36ページの「設定および構成の前に」
- 37ページの「名前空間がすでに存在する場合の設定」
- 39ページの「2 通りの構成方法」

---

### NIS+ の概要

NIS+ はネットワークネームサービス的一种です。NIS と似ていますが、機能はさらに豊富になっています。NIS の拡張機能というより新しいソフトウェアプログラムであるといえます。

NIS+ は (ワークステーションのアドレス、セキュリティ情報、メール情報、Ethernet インタフェースおよびネットワークサービスに関する情報などの) 情報を 1 ヲ所に格納して、ネットワーク上のすべてのワークステーショ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 UNIX のファイルシステムに似た階層構造になっているため、組織の論理上の階層に合わせて構成できます。名前空間における情報の配置は、物理的な配置とは関係ありません。NIS+ 名前空間は複数のドメインに分割が可能で、そ

それぞれ独立して管理できます。クライアントは、適切なアクセス権を持っていれば自分のドメインだけでなく他のドメインの情報にもアクセスできます。

NIS+ では、NIS+ 名前空間への情報の保存やその情報へのアクセスにクライアントサーバーモデルを使用します。それぞれのドメインは、複数のサーバーによってサポートされます。主となるサーバーを「マスター」サーバー、補助用のサーバーを「複製」サーバーと呼びます。ネットワーク情報は、NIS+ 内部のデータベースにある 16 個の標準 NIS+ テーブルに格納されます。マスターサーバーと複製サーバーの両方で NIS+ サーバーソフトウェアが動作しており、NIS+ テーブルのコピーを維持しています。マスターサーバーの NIS+ データに変更が加えられると、その内容は自動的に複製サーバーに順次反映されます。

NIS+ では高機能のセキュリティシステムによって、名前空間の構造と保存されている情報が保護されます。このシステムは、情報にアクセスしようとしているクライアントが正当なものであるかどうかを認証と承認によって確認します。「認証」では、情報を要求しているのがネットワーク上の正当なユーザーであるかどうかを確認します。「承認」では、ユーザーが情報を所有したり修正したりできるかどうかを確認します。

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、ワークステーションがネットワーク情報を検索する場所を指定します。ネットワーク情報が保存されているのは、ローカルの `/etc` ディレクトリ内のファイル、NIS、DNS、NIS+ です。ネームサービススイッチには、情報の種類ごとに異なる情報源を指定することもできます。

NIS+ の詳細は、『Solaris ネーミングの管理』を参照してください。

---

## 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。これには次の 2 つの作業があります。
  - 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、メモリ容量は
  - サーバーのディスク容量はどれくらい必要か

設計に関する詳細な検討項目や指針については、『NIS+ への移行』を参照してください。

- 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。37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。

---

## 名前空間がすでに存在する場合の設定

NIS ドメインがすでに存在する場合は、NIS+ 名前空間にも既存のものと同じフラットな構造のドメインを使用できます (階層構造にあとから変更可能)。NIS から NIS+ への移行は、『NIS+ への移行』を読んで、計画や準備に関する重要事項を確認してから行なってください。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章では、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。詳細は、『NIS+ への移行』を参照してください。

準備に関する主な注意事項は次のとおりです。

- 「ドメインとホストには同じ名前を使用しない」

たとえば、ドメインが sales の場合、マシンの名前には sales を付けしないでください。同様に、home という名前のマシンを使用している場合は、home という名前のドメインは作成しないでください。これはサブドメインの場合も同様です。たとえば、west という名前のマシンを使用している場合は、sales.west.myco.com というサブディレクトリを作成しないようにします。

- 「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 という名前は、sales-alpha という名前に置き換えます

- 「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。偽造されたエントリや誤ったエントリがなく、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、エントリの中で、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 つのファイルと、サブディレクトリ */var/nis/hostname* がありました。Solaris 2.5 の NIS+ では、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。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」

第 2 章、第 3 章で説明するように、`nisserver`、`nispopulate`、`nisclient` という 3 つのスクリプトを使って NIS+ 名前空間を構成します。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」

第 4 章から第 9 章で説明するように、NIS+ コマンドを使うとスクリプトを使う場合より柔軟に構成を決めることができますが、それだけ難しく、NIS+ 管理者としての経験が必要になります。スクリプトによる標準構成では要件を満たせないといった場合に限り使用してください。

---

注 - NIS+ コマンドを使って構成する場合、NIS+ ネームサービスを使う各マシンの `/etc` ディレクトリに正しい `nsswitch.conf` ファイルが必ず格納されるようにしてください (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、NIS+ ドメイン、NIS+ サーバー、NIS+ 名前空間を削除する方法については、『Solaris ネーミングの管理』を参照してください。





## NIS+ 設定スクリプト - 紹介

---

この章では、NIS+ スクリプトについて説明し、スクリプトの実行する作業と実行しない作業について解説します。

- 41ページの「NIS+ スクリプトについて」
- 42ページの「NIS+ スクリプトでは実行しないこと」

---

### NIS+ スクリプトについて

---



**注意** - 36ページの「設定および構成の前に」の手順を必ず実行してから、NIS+ スクリプトを実行してください。

---

3つのNIS+ スクリプト `nissserver`、`nispopulate`、`nisclient` 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。各スクリプトの動作内容を表 3-1 に示します。

表 3-1 NIS+ スクリプト

NIS+ スクリプト	機能
nissserver	ルートマスターサーバー、非ルートマスターサーバー、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、`nissserver(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。

`nissserver` 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非ルート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。



## スクリプトを使用した NIS+ の設定

---

この章では、`nisserver`、`nispopulate`、`nisclient` の各スクリプトと NIS+ コマンドを組み合わせて、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。

- 46ページの「NIS+ 設定の概要」
- 48ページの「NIS+ 名前空間サンプルの作成」
- 51ページの「NIS+ ルートサーバーの設定」
- 59ページの「NIS+ テーブルの生成 (`populate`)」
- 67ページの「NIS+ クライアントマシンの設定」
- 71ページの「NIS+ クライアントユーザーの初期設定」
- 72ページの「NIS+ サーバーの設定」
- 75ページの「ルート複製サーバーの作成」
- 81ページの「サブドメインの作成」
- 84ページの「新しいサブドメインのテーブルの生成」
- 88ページの「サブドメイン複製サーバーの作成」
- 89ページの「サブドメインの NIS+ クライアントマシンの初期設定」
- 91ページの「サブドメインの NIS+ クライアントユーザーの初期設定」
- 92ページの「NIS+ 名前空間サンプルでを使用したコマンドのまとめ」

また、この章では次の手順について説明します。

- 53ページの「ルートマスターサーバーを作成する方法」
- 56ページの「誤った情報を変更する方法」

- 61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」
- 68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」
- 72ページの「NIS+ ユーザーを初期設定する方法」
- 74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」
- 77ページの「ルート複製サーバーを作成する方法」
- 82ページの「新しい非ルートドメインを作成する方法」
- 87ページの「マスターサーバーテーブルを生成する方法」
- 89ページの「複製サーバーを作成する方法」
- 91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」
- 92ページの「NIS+ サブドメインユーザーを初期設定する方法」

---

## 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。NIS+ コマンド群を使って NIS+ 名前空間を設定する方法 (以降の章で説明) より簡単に実行できます。

(スクリプトの詳細は、`nisserver(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、『Solaris ネーミングの管理』の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。中央の列には作業の説明を、右端の列には各手順に必要なスクリプトまたは NIS+ コマンドをそれぞれ示します

表 4-1 NIS+ の推奨構成手順の概要

作業	説明	スクリプトまたは NIS+ 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。計画に関する要件と手順の詳細は、『NIS+ への移行』を参照。(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、37ページの「名前空間がすでに存在する場合の設定」および『NIS+ への移行』を参照。(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	nisserver
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続いてサーバーになる)。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	rpc.nisd
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 台または複数を経典ドメインの複製として指定する	rpc.nisd nisserver
非ルートドメインの構成	新しいドメインを作成する。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	rpc.nisd nisserver

表 4-1 NIS+ の推奨構成手順の概要 続く

作業	説明	スクリプトまたは NIS+ 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>
非ルート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を行う	<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+ 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、スクリプトの使用方法を説明します。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 互換モードの詳細は、『NIS+ への移行』および『Solaris ネーミングの管理』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非ルート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

NIS+ ドメインの例を作成する場合の NIS+ スクリプトとコマンドの一般的な順序を、表 4-2 に示します。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

目的	対象マシン	コマンド行
<code>/usr/lib/nis</code> を <code>root</code> 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	<code>setenv PATH \$PATH:/usr/lib/nis</code> または <code>PATH=\$PATH:/usr/lib/nis; export PATH</code>
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	<code>nisauthconf -dhkey-length-<i>alg-type</i> des</code>

表 4-2 NIS+ ドメインの構成に使うコマンド行のまとめ 続く

目的	対象マシン	コマンド行
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	<code>nisserver -r -d newdomain.</code> または <code>nisserver -Y -r -d newdomain.</code>
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	<code>nispopulate -F -p /files -d newdomain.</code> または <code>nispopulate -Y -d newdomain. -h NISservername -a NIS_server_ipaddress -y NIS_domain</code>
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	<code>nisgrpadm -a admin.domain. name.domain.</code>
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	<code>nisping -C domain.</code>
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	<code>nisclient -i -d domain. -h master1</code>
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	<code>nisclient -u</code>
<code>rpc.nisd</code> デーモンを起動 - NIS (および DNS) との互換性なし、またはありでクライアントをサーバーにするために必要	クライアントマシン。スーパーユーザーとして行う	<code>rpc.nisd</code> または <code>rpc.nisd -Y</code> または <code>rpc.nisd -Y -B</code>
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	<code>nisserver -R -d domain. -h clientname</code>
サーバーを非ルート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	<code>nisserver -M -d newsubdomain.domain. -h clientmachine</code>

表 4-2 NIS+ ドメインの構成に使うコマンド行のまとめ 続く

目的	対象マシン	コマンド行
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	<code>nispopulate -F -p /subdomaindirectory -d newsubdomain.domain.</code> または <code>nispopulate -Y -d newsubdomain.domain. -h NISservername -a NIS_server_ipaddress -y NIS_domain</code>
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	<code>nisserver -R -d subdomain.domain. -h clientname</code>
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	<code>nisclient -i -d newsubdomain.domain. -h subdomainmaster</code>
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	<code>nisclient -u</code>

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 - `nissserver` 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、『Solaris ネーミングの管理』およびこのマニュアルの第 1 章を参照してください。

---

## nissserver を実行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。

### 必要な情報

次の情報が必要です。

- ルートマスターサーバーになるワークステーション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の名前。ルートドメイン名は少なくとも 2 つの要素 (ラベル) で構成し、その末尾にドットを置きます (例 *something.com.*)。最後の要素は実際には何でもかまわないのですが、インターネットとの互換性を保つには、インターネット組織コード (表 4-3) または 2 ~ 3 文字の地域コード (日本であれば *.jp.*) にしなければなりません。

表 4-3 インターネット組織コード

ドメイン	種類
com	営利組織
edu	教育機関
gov	行政機関

表 4-3 インターネット組織コード 続く

ドメイン	種類
mil	軍事組織
net	主要ネットワークサポートセンター
org	非営利組織
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 - ドメインとホストでは、同じ名前を使用しないでください。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 という名前のマシンを使用している場合は、home という名前のドメインを作成しないでください。この注意事項は、サブドメインの場合にもあてはまります。たとえば、マシンに west という名前を付けている場合、sales.west.myco.com というサブドメインを作成しないでください。

## ▼ 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. 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+ group」とは、doc.com. ドメイン (ドメイン名は必ず最後にピリオドが付く) 内の情報を変更する許可が与えられたユーザーのグループを意味します。変更には削除も含まれます。この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、56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。56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については、『Solaris ネーミングの管理』および rpc.nisd(1M) 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、56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安全に終了します)。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せん。

```
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、59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 54ページの手順 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. であることを確認しました。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+** グループが正しければ **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。詳細は 74 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、53 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、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の場合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、79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**つのイーサネット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-1e0
192.168.11.y hostA hostA-11 hostA-1e1
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、他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`、`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 を実行するための前提条件

スクリプト `nispopulate` を実行するには、次の条件が必要です。

- 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。構成が完了した後は、いつでも各エントリを追加できます。後から追加する方が、不完全なエントリや破損したエントリを読み込むよりも簡単です。
- ファイル内の情報は、それがロードされるテーブルに適合する書式でなければなりません。『Solaris ネーミングの管理』および第 9 章では、対応する NIS+ テーブルに変換するテキストファイルで要求される書式について説明しています。
- 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` というドメインを使用している

場合、sales という名前の付いたマシンを使用しないでください。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。この注意事項は、サブドメインの場合にもあてはまります。たとえば、マシンに west という名前を付けている場合、sales.west.myco.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 というマシン名は無効ですが、これを sales-alpha とすれば有効になります。
- 初めてネットワークを設定する場合、十分なネットワーク情報がどこにも格納されていないことがあります。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必要があります。
- 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が必要です (この例では、マシン master1 上のユーザーは root)。

## 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
- 適切に編集された変換されるデータを持つテキストファイルへのパス
- 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 のドメイン名では大文字と小文字を区別しますが、NIS+ のドメイン名では区別しません。

---

## ▼ 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、63ページの手順 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て読めないことがあります。

---

注 - システムの /tmp 領域が不足すると、nispopulate スクリプトが異常終了することがあります。このようなことが起こらないように、環境変数 TMPDIR に別のディレクトリを設定できます。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 オプションは、入力ファイルが存在するディレクトリのパスを指定します (この例では、パスは /nis+files)。-d 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 doc.com.)。

NIS+ 主体のユーザーは root です。この例では、初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。sales.doc.com. はドメイン名の例です。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 ドメイン名は大文字と小文字が区別されることを忘れないでください。

NIS+ 主体のユーザーは root です。この例では、初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を要求します。情報が誤っている場合の操作は、56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.
```

(続く)

```
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、スクリプトは安全に終了します。`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まま残されることがあります。現在生成中であったテーブルは、一部だけが生成されていることがあります。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他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、他の管理者の 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。しかし、管理グループにほかの管理者を追加するには、

その前に `passwd` テーブルに管理者が登録されていることが必要です。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、72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。スワップ領域を増やして、再度ドメインのチェックポイントを実行してください。

---

## 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のこれ以上の初期化は必要ありません。この節では、`nisclient` スクリプトを使用して、デフォルトの設定で NIS+ クライアントを初期設定する方法を説明します。

- 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 - 68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。管理者は、明確にクライアントのネームサービススイッチファイルに DNS を指定する必要があります。DNS によるホスト名の検索については、『Solaris ネーミングの管理』およびこのマニュアルの第 1 章を参照してください。

---

## nisclient を実行するための前提条件

nisclient スクリプトを使用するには、次の条件が必要です。

- ドメインがすでに構成されていて、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

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ワークステーションの root のパスワード
- 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ない場合、そのデフォルトは現在のドメイン名となります。-h オプションは NIS+ サーバーのホスト名を指定します。

```
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非ルート (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(1M)` を実行するための前提条件

`nisclient(1M)` スクリプトを使用してユーザーを初期設定するには、次の条件が必要です。

- 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`)
- DES 認証を使用する場合には、クライアントマシンは、マスターサーバーと同じ Diffie-Hellman キー設定を使用

### 必要な情報

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。  
入力した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、51 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 - NIS+ 名前空間に SunOS 4.x クライアントが存在する場合にだけ、DNS 転送を設定する必要があります。NIS 互換モードの使用方の詳細は、『NIS+ への移行』を参照してください。

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。

この例では、マシン `client1` がサーバーに変更されます。この手順では、NIS+ スクリプトの代わりに NIS+ コマンド `rpc.nisd` を使用します。

## rpc.nisd を実行するための前提条件

`rpc.nisd` を実行するには、次の条件が必要です。

- 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 `root` 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はマスターサーバーと同じ Diffie-Hellman キー設定を使用する必要があります。

## 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の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 NIS+ サーバーを構成する方法

◆ NIS と互換性のないサーバーを構成するには、次のコマンドを入力します。

```
client1# rpc.nisd
```

## NIS と互換性のある NIS+ サーバーを構成する方法

1. サーバーの /etc/init.d/rpc ファイルを編集して、文字列 `-EMULYP=''-Y''` を含む行のコメントアウトを解除して有効にします。  
コメントアウトを解除するには、行の先頭から # を削除します。
2. スーパーユーザーとして、次のように入力します。

```
client1# rpc.nisd -Y
```

## NIS との互換性と DNS 転送機能を備えた NIS+ サーバーを構成する方法

次の手順を実行して、DNS 転送機能と 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。

1. サーバーの /etc/init.d/rpc ファイルを編集します。文字列 `-EMULYP=''-Y''` を含む行のコメントアウトを解除 (行の先頭から # を削除) して有効にします。  
コメントアウトを解除するには、行の先頭から # を削除します。

2. 上の行の引用符で囲まれた中に次のように `-B` を追加します。

```
-EMULYP=''-Y -B''
```

3. 次のコマンドをスーパーユーザーとして入力します。

```
client1# rpc.nisd -Y -B
```

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。ただし、ネットワークが複数のサブネットで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した方がいい場合があります。

### ■ 「サブネット」

複数のサブネットで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すれば、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配置については、『NIS+ への移行』を参照してください。

最適な複製サーバーの数については、『Solaris ネーミングの管理』を参照してください。

77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。144ページの「NIS+ コマンドによる複製サーバーの構成」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 を実行するための前提条件

`nissserver` を実行してルート複製サーバーを作成するには、次の条件が必要です。

- ドメインがすでに構成され、マスターサーバーが稼動している
- 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 1 つは必要
- 新たに作成した複製サーバーを `doc.com.` 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、67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 `rpc.nisd(1M)` を起動している。(具体的な手順については、72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ルート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

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が構成されることを示します。-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 を入力して続行します。

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、56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 を入力して続行します。

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. If you want to run this replica in NIS (YP)
compatibility mode, edit the /etc/init.d/rpc file on the replica server '
to uncomment the line which sets EMULYP to "-Y". This will ensure that
rpc.nisd will boot in NIS-compatibility mode. Then, restart rpc.nisd with
the '-Y' option. These actions should be taken after this script completes.
```

---

注 - 上記のメッセージはオプションの手順に関するものです。NIS (YP) 互換ではないルート複製サーバーを NIS 互換にする場合にだけ、`/etc/init.d/rpc` ファイルを変更します。つまり、NIS 互換のサーバーとして構成されていない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にだけ、このファイルの変更が必要になります。NIS 互換サーバーの作成については、74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。

---

4. [必要に応じて] 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. `rpc.nisd` を終了します。
  - b. サーバーの `/etc/init.d/rpc` ファイルをオープンして、`EMULYP` が `-Y` に設定されている行のコメントアウトを解除します。  
具体的には、`EMULYP` 行の先頭の `#` を削除します。
  - c. `rpc.nisd` を再起動します。
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、147ページの「`nisrestore` を使って名前空間データをロードする」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、149ページの「`nisping` を使って名前空間データをロードする方法」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成することができます。同じ手順を使ってサブドメインに複製サーバーを作成することもできます。

## ▼ 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 `nissserver` 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、57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、次のように入力します。

```
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、次のように入力します。

```
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. ルートマスター以外のサーバー上で、新しいサーバーがまだ稼動していない場合は、`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、次のように入力します。

```
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、次のように入力します。

```
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。nissserver 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、nissserver を使用して複製サーバーを設定します。たとえば、次のように入力します。

```
hostA# nissserver -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。nissserver 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 sub.doc.com. という新しい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 nissserver を使用します。

### nissserver を実行するための前提条件

nissserver 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、次の条件が必要です。

- 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 `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 `root` 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

次の情報が必要です。

- 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`」となる
- クライアントマシンの名前 (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

82ページの「新しい非ルート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。

---

注 - 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.` 内の NIS+ クライアントは、`west.sales.doc.com.` や `alameda.west.sales.doc.com.` などの下位の階層に属するドメインにサービスを提供できます。しかし、このクライアントは `doc.com.` ドメインにはサービスを提供できません。その理由は、`doc.com.` がドメイン `sales.doc.com.` より上位の階層にあるためです。ただし、ルート複製サーバーはこの規則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。

---

---

注 - 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。これによって、サーバーとドメイン階層の他の部分との通信を安全な状態に保ったまま、サーバーのドメイン名をそのクライアントが使用しているドメイン名と同じに設定することができます。 `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。

---

## ▼ 新しい非ルート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、新しい非ルート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、新しい非ルート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# nissserver -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)
```

新しい非ルート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、53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、56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

サーバーを新しい非ルート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+ 名前空間の作成を始めます。NIS+ 階層の計画については、第 2 章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 を実行するための前提条件

スクリプト nispopulate 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、次の条件が必要です。

- ファイル内の情報は、それがロードされるテーブルに合わせて書式設定さ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。変更を完了した後は、いつでも各エントリを追加できます。後から追加する方が、不完全なエントリや破損したエントリを読み込むよりも簡単です。
  - 初めてネットワークを設定する場合、十分なネットワーク情報がどこにも格納されていないことがあります。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
```

(続く)

```
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なければなりません。
- 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、61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、新しいドメイン (sales.doc.com.) のテーブルを生成するための入力についてだけ説明します。この手順の詳細は、61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)
```

このスクリプトの残りの出力については、61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 (75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、77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、次の条件が必要です。

- ドメインがすでに設定され、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 rpc.nisd が起動されている
- マスターサーバーにルート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

- 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、77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ワークステーションです。

---

注 - 91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。DNS によるホスト名の解決については、27ページの「DNS を使用できるようにする」を参照してください。

---

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、68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、次の条件が必要です。

- 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、ユーザーの名前は user1)

## 必要な情報

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 Secure RPC パスワード (nisplus)
- クライアントになるワークステーションの root 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、現在のドメイン名がデフォルトになります。-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、68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いければなりません。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、72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 (1M) を実行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、次の条件が必要です。

- 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

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、72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+ 名前空間サンプルのコマンド行のまとめ

目的	コマンド行
/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)	<pre># setenv PATH \$PATH:/usr/lib/nis または # PATH=\$PATH:/usr/lib/nis; export PATH</pre>
オプションで Diffie-Hellman キー長を設定する	<pre>master1# nisauthconf dh640-0 des</pre>
doc.com. ドメインのルートマスターサーバーを作成	<pre>master1# nisserver -r -d doc.com.</pre>
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	<pre>master1# nispopulate -F -p /nis+files -d doc.com. または master1# nispopulate -Y -d doc.com. -h salesmaster -a \ 130.48.58.111 -y sales.doc.com.</pre>
管理グループ (2) にメンバーを追加する	<pre>master1# nisgrpadm -a admin.doc.com. topadmin.doc.com. \ secondadmin.doc.com.</pre>
NIS+ データベースのチェックポイントを実行	<pre>master1# nisping -C org_dir.doc.com.</pre>
オプションで Diffie-Hellman キー長を設定する	<pre>client1# nisauthconf dh640-0 des</pre>
doc.com. ドメイン内の NIS+ クライアントマシンを初期設定	<pre>client1# nisclient -i -d doc.com. -h master1</pre>
ユーザーを NIS+ クライアントとして初期設定	<pre>client1user1prompt% nisclient -u</pre>
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	<pre>client1# rpc.nisd または client1# rpc.nisd -Y または client1# rpc.nisd -Y -B</pre>

表 4-4 NIS+ 名前空間サンプルのコマンド行のまとめ 続く

目的	コマンド行
ルート複製サーバーを作成	<code>master1# nisserver -R -d doc.com. -h client1</code>
サーバーを sales.doc.com. ドメインの非 ルートのマスターサーバーに変更	<code>master1# nisserver -M -d sales.doc.com. -h client2</code>
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	<code>client2# nispopulate -F -p /nis+files -d sales.doc.com.</code> または <code>client2# nispopulate -Y -d sales.doc.com. -h \ businessmachine -a 130.48.58.242 -y business.doc.com.</code>
マスターサーバーの複製を作成	<code>client2# nisserver -R -d sales.doc.com. -h client3</code>
sales.doc.com. ドメイン内の NIS+ クライアントを初期設定	<code>subclient1# nisclient -i -d sales.doc.com. -h client2</code>
ユーザーを NIS+ クライアント として初期設定	<code>subclient1user2prompt% nisclient -u</code>

## 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。

- 95ページの「ルートドメインの設定方法の概要」
- 96ページの「標準構成と NIS 互換構成の手順の相違」
- 96ページの「ルートドメインの確立」
- 117ページの「ルートドメイン構成の要覧」

既存のルートマスターサーバーの使用を中止し、別のマシンをルートマスターサーバーにする方法については、『Solaris ネーミングの管理』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・ルートドメインを構成する作業は、この章で説明する NIS+ コマンド群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。

---

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。他のパラメータはその後に指定します。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+ デーモンは `-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。起動方法については「ルートドメインを構成する方法」の 104 ページの手順 11 で説明します。標準の NIS+ ドメインについては 105 ページの手順 12 で説明します。

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 107 ページの手順 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。

---

## ルートドメインの確立

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。詳細な説明が必要ない場合は、117 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. **NIS+** のファイルを削除し、プロセスを終了します。
6. ルートドメインの管理グループを指定します。
7. 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。
8. **NIS+** デーモンを `-Y` で起動します (**NIS** 互換の場合)。**NIS+** デーモンを起動します (**NIS+** 互換の場合)。
9. **NIS+** デーモンが実行されていることを確認します。
10. 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。
11. ルートマスターサーバーの **DES** 資格を作成します。
12. ルートドメインの管理グループにルートマスターを追加します。
13. ルートドメインの管理グループを作成します。
14. ルートドメインの公開鍵を更新します。
15. **NIS+** キャッシュマネージャを起動します。
16. **NIS+** デーモンをセキュリティレベル **2** で再起動します。
17. 自分の **LOCAL** 資格をルートドメインに追加します。
18. 自分の **DES** 資格をルートドメインに追加します。

19. 他のシステム管理者の資格を追加します。

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

## ルートドメインを確立する — タスクマップ

表 5-1 ルートドメインを確立する

タスク	説明	指示の参照先
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 デモンの停止と起動を行う。keyerv デモンの強制終了と再起動を行う。余分な NIS+ 情報を削除する	99ページの「ルートドメインを構成する方法」

### セキュリティ上の留意点

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、構成およびテストの目的だけに使用します。

---

注 - NIS+ のセキュリティシステムは複雑です。NIS+ のセキュリティに精通していない場合は、『Solaris ネーミングの管理』のセキュリティ関連の章をもう一度読んでから NIS+ 環境の構成を始めてください。

---

### 前提条件

作業を開始する前に、以下のことを確認します。

- 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、他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。

- このサーバーが NIS 互換モードで動作し、Solaris リリース 1.x クライアント用に DNS 転送をサポートする場合、正しく構成された `/etc/resolv.conf` ファイルが必要です (241ページの「リゾルバ」を参照)。
- サーバーには、必ず他のユーザー 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。

- ルートマスターサーバーとなるワークステーションのスーパーユーザーパスワード
- 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他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、ワークステーションの現在のドメイン名を返します。



注意 - ドメインとホストで同じ名前を使用しないでください。たとえば、sales というドメインを使用している場合、sales という名前の付いたマシンは使用しないでください。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、サブドメインの場合にもあてはまります。たとえば、マシンに west という名前を付けている場合、sales.west.myco.com というサブディレクトリを作成しないでください。

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+ コマンドではありません。そのため、domainname コマンドは、ドメイン名の最後にドットを付ける NIS+ の表記規則には従っていません。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、25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 ファイルに対する変更が終了したら、nscd デーモンを終了して再起動します。

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 を終了して再起動する必要があります。詳しい手順は、第 1 章を参照してください。

6. 次のように入力し、keyserv を強制終了して再起動します。

```
rootmaster# cp /etc/nsswitch.nisplus /etc/nsswitch.conf
rootmaster# sh /etc/init.d/nscd stop
rootmaster# sh /etc/init.d/nscd start
rootmaster# ps -e | grep keyserv
  root 145 1 67 16:34:44 ? keyserv
.
.
rootmaster# kill -9 145
rootmaster# rm -f /etc/.rootkey
rootmaster# keyserv
```

7. **NIS+** のファイルを削除しプロセスを終了します。

現在作業しているワークステーションが、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、キャッシュマネージャがまだ実行中であれば、そのプロセスを終了します。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/*
rootmaster# ps -ef | grep nis_cachemgr
  root 295 260 10 15:26:58 pts/0 0:00 grep nis_cachemgr
  root 286 1 57 15:21:55 ? 0:01 /usr/sbin/nis_cachemgr
rootmaster# kill -9 286
```

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、これらを一時的に他のどこかに格納しておくことができます。

8. サーバーのデーモンを終了します。

現在作業しているワークステーションが、すでに NIS+ サーバーとして使用されている場合は、rpc.nisd または rpc.nispasswd が実行されているかどうか確認してください。実行されている場合は、どちらのデーモンも終了してください。

9. ルートドメインの管理グループを指定します。

109ページの手順 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 107ページの手順 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、ワークステーショ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。いずれにせよ、ルートマスターの場合を除くと、これらの2つのプロセスは別々の作業として行われます。

```
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。  
この下には `root.object.` という名前の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+ が内部目的のために名前空間の `root` 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、104ページの手順 11 または 105ページの手順 12 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、他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 リリース 2.4 以前では、/var/nis ディレクトリに *hostname.dict*、*hostname.log* という 2 つのファイルとサブディレクトリ /var/nis/*hostname* が存在していました。Solaris リリース 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 リリース 2.4 での名前にしてしまうと、Solaris リリース 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。

---

#### 11. NIS+ デーモンを -Y で起動します (NIS 互換の場合のみ)。

この手順は、NIS 互換モードでルートドメインを設定している場合にだけ実行します。標準の NIS+ ドメインを設定する場合は、この代わりに 105 ページの手順 12 を実行します。この手順には、NIS クライアントの DNS 転送機能をサポートするための操作手順が含まれます。

「a」では、NIS+ デーモンを NIS 互換モードで起動します。「b」では、NIS+ デーモンはこのサーバーが再起動されたとき、確実に NIS 互換モードで再起動します。「b」の後は、107 ページの手順 14 に進んでください。

- a. *rpc.nisd* に -Y、-B、-S 0 オプションを使用して実行します。

```
rootmaster# rpc.nisd -Y -B -S 0 options
```

-Y オプションを指定すると、NIS+ 要求だけでなく NIS 要求にも応答します。-B オプションを指定すると DNS 転送をサポートします。-s 0 フラグは、サーバーのセキュリティレベルに 0 を設定します。この時点では、ブートストラップ動作のためにこの設定が必要です。cred テーブルはまだ存在しないため、NIS+ 主体は資格をもつことができません。これより高いセキュリティレベルを使用した場合、ユーザーがサーバーから締め出されてしまいます。

- b. /etc/init.d/rpc ファイルを編集します。



/etc/init.d/rpc ファイル内で文字列 EMULYP='Y' を検索します。その行のコメント指定を解除し、DNS 転送機能を使用できるようにするために、-B フラグを追加します。

DNS 転送を使用する rpc ファイルの場合

```
EMULYP='Y -B'
```

DNS 転送を使用しない rpc ファイルの場合

```
EMULYP='Y'
```

DNS 転送機能を使用する必要がない場合、この行をコメント解除しますが、-B フラグは追加しません。

## 12. NIS+ デーモンを起動します (標準 NIS+ の場合のみ)。

rpc.nisd コマンドを使用しますが、必ず -s 0 フラグを追加してください。

```
rootmaster# rpc.nisd -s 0
```

-s 0 フラグは、サーバーのセキュリティレベルに 0 を設定します。この時点では、ブートストラップ動作のためにこの設定が必要です。cred テーブルはまだ存在しないため、NIS+ 主体は資格をもつことができません。これより高いセキュリティレベルを使用した場合、ユーザーがサーバーから締め出されてしまいます。

## 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。

104ページの手順 11 または 105ページの手順 12 を実行すると、名前空間には次のものが作成されます。

- ルートディレクトリオブジェクト (root.dir)
- NIS+ デーモン (rootmaster) を実行するルートマスターサーバー (rpc.nisd)
- 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、103ページの手順 10 で作成したディレクトリに格納されます。ls コマンドを使用して、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、`nislsls` コマンドを使用します。

```
rootmaster# nislsls -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+ デー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 -ef | 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
```

(続く)

続き

```
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。

nisls でルートディレクトリの内容を調べたとき (105ページの手順 12)、これが空であったことを思い出してください。現在は2つのサブディレクトリがありません。

```
rootmaster# nisls 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与えるには、keylogin -r を実行します (『Solaris ネーミングの管理』の、資格に関連のある章を参照)。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 (第 8 章を参照)。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、102ページの手順 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、110ページの手順 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 : ----rncdr---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 `admin.doc.com` グループに `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.
```

ルートマスターが本当にグループのメンバーであることを確認するには、`-l` オプションを指定して `nisgrpadm` コマンドを使用します (『Solaris ネーミングの管理』のグループ関連の章を参照)。

---

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。詳細は、『Solaris ネーミングの管理』のディレクトリ関連の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.
```

(続く)

続き

```
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+ キャッシュマネージャを起動します。

キャッシュマネージャは、NIS+ クライアント (この場合、ルートマスターサーバー) に関する位置情報のローカルキャッシュを管理します。初期の情報をクライアントのコールドスタートファイル (104ページの手順 11 または 105ページの手順 12 で作成) から入手し、これを `/var/nis` 内の `NIS_SHARED_DIRCACHE` という名前のファイルに保存します。

キャッシュマネージャを起動するには、次に示すように、`nis_cachemgr` コマンドを入力します。

```
rootmaster# nis_cachemgr
```

キャッシュマネージャがいったん起動されたら、このプロセスを明示的に終了させた場合を除いて、再起動する必要はありません。クライアントが再起動されたときには、`/var/nis` 内の `COLD_START` ファイルがキャッシュマネージャを自動的に起動するため、キャッシュマネージャを再起動する必要はありません。NIS+ キャッシュマネージャの詳細は、『Solaris ネーミングの管理』のディレクトリに関連する章を参照してください。



## 20. NIS+ デーモンをセキュリティレベル 2 で再起動します。

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、ルートマスターをセキュリティレベル 2 で再起動できます。まず既存のデーモンのプロセスを終了し、次にこれをセキュリティレベル 2 で再起動します。

標準 NIS+ ドメインの場合

```
rootmaster# ps -e | grep rpc.nisd
1081 ? 0:03 rpc.nisd -s 0
rootmaster# kill 1081
rootmaster# rpc.nisd
```

NIS 互換のルートドメインの場合、必ず `-Y` フラグ (および `-B` フラグ) を使用してください。

NIS 互換の NIS+ ドメインの場合

```
rootmaster# ps -e | grep rpc.nisd
1081 ? 0:03 rpc.nisd -Y -B -s 0
rootmaster# kill 1081
rootmaster# rpc.nisd -Y -B
```

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、設定およびテストの目的だけに使用します。

---

## 21. 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 コマンドの詳細は、『Solaris ネーミングの管理』の資格に関する章を参照してください。

## 22. 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。

## 23. 他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他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。

- 他の管理者に一時的な資格を作成するには、NIS+ モードで実行されている Solstice AdminSuite (使用できる場合) を使用すると簡単です。
- 2つ目は、他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. のシステム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つ目は、ダミーのパスワードを使用して、他の管理者に一時的な資格を作成する方法です。他の管理者、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。次の例で、その手順を示しています。

```
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。詳しくは、『Solaris ネーミングの管理』の資格に関連する章を参照してください。

#### 24. ルートドメインの管理グループに自分と他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、他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.
```

## 25. 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手順のまとめ

作業	コマンド
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	<code>rootmaster% su</code> Password: ルートパスワードを入力する
ドメイン名をチェックする	<code># domainname</code>
スイッチファイルをチェックする	<code># more /etc/nsswitch.conf</code>
NIS+ データを削除し、プロセスを終了する	<code># rm -rf /var/nis*</code>
管理グループを指定する	<code># NIS_GROUP=admin.doc.com.; export NIS_GROUP</code>
ルートマスターサーバーを初期設定する	<code># nisinit -r</code>
NIS 互換の場合のみ: デーモンを <code>-Y -B, S 0</code> で起動する	<code># rpc.nisd -Y -B -S 0</code> <code># vi /etc/inet.d/rpc</code>
EMULYP= <code>-Y -B</code> に変更する	
または、 標準 NIS+ の場合のみ: デーモンを <code>-S 0</code> で起動する	<code># rpc.nisd -S 0</code>
<code>org_dir</code> と、 <code>groups_dir</code> テーブルを作成する	<code># /usr/lib/nis/nissetup [-Y]</code>
ルートマスターサーバー用の DES 資格を作成する	<code># nisaddcred des</code> Enter login password:
管理グループを作成する	<code># nisgrpadm -c admin.doc.com.</code>

表 5-2 ルートドメインを設定する手順のまとめ 続く

作業	コマンド
ルートディレクトリに完全なグループ権を割り当てる	<code># nischmod g+rmcd doc.com.</code>
rootmaster を admin グループに追加する	<code># nisgrpadm -a admin.doc.com. rootmaster.doc.com.</code>
ルートディレクトリの鍵を更新する	<code># /usr/lib/nis/nisupdkeys doc.com.</code>
org_dir の鍵を更新する	<code># /usr/lib/nis/nisupdkeys org_dir.doc.com.</code>
groups_dir の鍵を更新する	<code># /usr/lib/nis/nisupdkeys groups_dir.doc.com.</code>
NIS+ キャッシュマネージャを起動する	<code># nis_cachemgr</code>
既存のデーモンを終了する	<code># ps -ef   grep rpc.nisd</code> <code># kill -9 process id</code>
NIS+ デーモンを再起動する (NIS 互換には -Y を、DNS には -B を使用する)	<code># rpc.nisd [-Y] [-B]</code>
自分の LOCAL 資格を追加する	<code># nisaddcred -p 11177 -P topadmin.doc.com. local</code>
自分の DES 資格を追加する	<code># nisaddcred -p unix.11177@doc.com -P \</code> <code>topadmin.doc.com. des</code> Enter login password:
他のシステム管理者の資格を追加する 他のシステム管理者を管理グループに追加する	<code># nisaddcred ... nisgrpadm -a admin.doc.com member</code>

## 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。ここで説明する手順は、NIS+ または NIS 互換であるかどうかにかかわらず、ルートドメインとサブドメインの両方に適用されます。

- 119ページの「NIS+ クライアントの設定方法の概要」
- 120ページの「クライアントを構成する」
- 129ページの「NIS+ クライアントを初期設定する」
- 131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」
- 133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」
- 127ページの「ワークステーションのドメイン名を変更する」

---

### NIS+ クライアントの設定方法の概要

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。手順の説明ではそれぞれの内容を詳細に説明し、関連する情報も示します。詳しい手順の説明が必要でない場合は、表 6-6 の「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。Solstice AdminSuite がある場合には、これを使用すると 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、126ページの手順 10 では、ブロードキャスト、ホスト名、またはコールドスタートファイルのうちのどれかを使用する方法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、126ページの手順 11 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、ワークステーションのドメイン名を変更する方法を取り上げています。

---

## クライアントを構成する

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** - ドメインとホストで同じ名前を使用しないでください。たとえば、sales というドメインを使用している場合、sales という名前の付いたマシンは使用しないでください。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、サブドメインの場合にも当てはまります。たとえば、マシンに west という名前を付けている場合、sales.west.myco.com というサブドメインを作成しないでください。

---

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
- ワークステーションの準備
- ワークステーションを 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、これらの手順を最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ワークステーション用の **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. `nscd` 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. `key serv` デーモンのプロセスを終了して再起動します。
12. `key login` を実行します。
13. クライアントを再起動します。

## セキュリティ上の留意点

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、これに対する変更権があれば、nischmod コマンドを使用して変更できます。詳しくは、『Solaris ネーミングの管理』の権利に関連する章を参照してください。

## 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、/etc/inet/ipnodes ファイルまたはドメインの hosts テーブル、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。sales-alpha というマシン名なら使用できます。

## 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ワークステーションのスーパーユーザーパスワード
- 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 — タスクマップ

表 6-1 クライアントの設定

タスク	説明	指示の参照先
クライアントの設定	クライアントの資格を作成する。クライアントワークステーションを準備して、NIS+ クライアントとして初期設定する	123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します。  
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ワークステーション用の **DES** 資格を作成します。  
-p と -P の引数を付けた nisaddcred コマンドを実行します。

```
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ワークステーションに対する 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` コマンドの詳細は『*Solaris* ネーミングの管理』の資格に関する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。

例

```
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ワークステーショ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、127ページの「ワークステーションのドメイン名を変更する」を参照し、次の 124ページの手順 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は、22 ページの「NIS+ 用デフォルトスイッチファイル」をご覧ください。

7. `nsswitch.conf` ファイルに何らかの変更を加えた場合 (または、新規にファイルにコピーした場合)、必ず次の表のように入力して `nscd` を停止してから再起動する必要があります。

```
client1# cp /etc/nsswitch.nisplus /etc/nsswitch.conf
client1# sh /etc/init.d/nscd stop
client1# sh /etc/init.d/nscd start
```

(第 1 章では、この時点でキーサーバーを停止して、再起動するよう説明していますが、同じ操作を 126 ページの手順 11 で行うため、この場合は停止および再起動の必要はありません。)

8. 手順 3 の情報を使用して、**Diffie-Hellman** キー長を設定します。

例 :

```
client# nisauthconf dh640-0 des
```

9. **NIS+** のファイルを削除し、プロセスを終了します。

作業しているワークステーションが、すでに NIS+ のサーバーまたはクライアントとして使用されていて、まだ実行中の場合、`/var/nis` ディレクトリに存在するファイルすべてを削除し、キャッシュマネージャを停止します。

```
client1# ls /var/nis
NIS_COLD_START NIS_SHARED_CACHE
client1# rm -rf /var/nis/*
client1# ps -ef | grep nis_cachemgr
root 295 260 10 15:26:58 pts/0 0:00 grep nis_cachemgr
root 286 1 57 15:21:55 ? 0:01 /usr/sbin/nis_cachemgr
client1# kill -9 286
```

/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が格納されている場合、ルートドメインの設定が終了するまでは、これを一時的に他の場所に格納しておくこともできます。

10. クライアントを初期設定します。

クライアントの初期設定は、ホスト名、コールドスタートファイル、またはブロードキャストのいずれかを使用して行うことができます。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、126ページの手順 11 に進みます。

11. keyserd デーモンを終了してから再起動します。

この手順では、非公開鍵をキーサーバー上に格納します。

a. keyserd デーモンを終了します。

この手順によって、キーサーバーが保持していたクライアントに関するスイッチ情報も更新されます。

b. /etc/.rootkey ファイルを削除します。

c. キーサーバーを再起動します。

次の例では、126ページの手順 11 の内容を示しています。

```
client1# ps -e | grep keyserd
root 145 1 67 16:34:44 ? keyserd
client1# kill 145
client1# rm -f /etc/.rootkey
client1# keyserd
```

d. keylogin -r を実行します。

この手順では、クライアントの非公開鍵をキーサーバーに格納します。また、クライアント上のスーパーユーザーが NIS+ を使用するために keylogin を行わなくてもすむように、/etc/.rootkey にコピーを保存します。-r オプションを付けて keylogin を実行します。パスワードの入力を求められたら、クライアントのスーパーユーザーパスワードを入力します。

このパスワードは、クライアントの DES 資格を作成するために与えられたパスワードと同じでなければなりません。

```
client1# keylogin -r
Password:
Wrote secret key into /etc/.rootkey
```

e. クライアントを再起動します。

---

## ワークステーションのドメイン名を変更する

この作業では、ワークステーションのドメイン名を変更します。ワークステーショ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、ワークステーショ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。

### セキュリティ上の留意点

この作業は、ドメイン名を変更するワークステーショ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

- ワークステーションのスーパーユーザーパスワード
- 新しいドメイン名

## ワークステーションのドメインの変更 — タスクマップ

表 6-2 クライアントの設定

タスク	説明	指示の参照先
ワークステーションのドメインの変更	domainname コマンドを使用して、クライアントワークステーションのドメインを変更する	128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. ワークステーションにログインし、スーパーユーザーになります。  
この例では、ワークステーションに `client1` を、新しいドメイン名に `doc.com.` を使用します。

```
client1% su  
Password:
```

2. ワークステーショ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、ワークステーションのドメインを `doc.com` に変更するのであれば、次のように入力します。

```
client1# domainname doc.com
```

ワークステーションが 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. 適切な時に、ワークステーションを再起動します。

/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、ワークステーションを再起動します。

この作業は、他のいくつかの作業の流れの中で行うものです。リブートは、ワークステーション上でのすべての作業が完了したことを確認してから行ってください。確認を怠ると、何度もリブートが必要になる可能性があります。

---

## 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 (130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 (131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 (133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最も簡単ですが、最も安全性の低い方法でもあります。ブロードキャストに回答した NIS+ サーバーはクライアントが自分自身のワ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回答するのは NIS+ サーバーだけですが、クライアントからは、ブロードキャストに回答したワークステーショ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。

## セキュリティ上の留意点

この作業は、クライアントのスーパーユーザーとして実行しなければなりません。

## 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 — タスクマップ

表 6-3 NIS+ クライアントを初期設定する

タスク	説明	指示の参照先
NIS+ クライアントを初期設定する	nisclient コマンドを使って、NIS+ クライアントを初期設定する	131ページの「ブロードキャストにより NIS+ クライアントを初期設定する方法」

## ブロードキャストにより NIS+ 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回答し、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回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。

## セキュリティ上の留意点

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。

## 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。

## NIS+ クライアントを初期設定する — タスクマップ

表 6-4 NIS+ クライアントを初期設定する

タスク	説明	指示の参照先
ホスト名によりクライアントを初期設定する	<code>nisinit</code> コマンドを使って、NIS+ クライアントをホスト名により初期設定する	132ページの「ホスト名によりクライアントを初期設定する方法」

## ホスト名によりクライアントを初期設定する方法

1. クライアントの `/etc/hosts` ファイルまたは `/etc/inet/ipnodes` ファイルを確認します。  
クライアントが、信頼できるサーバーのエントリを持っていることを確認します。
2. 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 `-H` のオプションを付けて実行します。次の例では、信頼できるサーバーとして `rootmaster` を使用します。

```
Client1# nisinit -c -H rootmaster
This machine is in the doc.com. NIS+ domain.
Setting up NIS+ client ...
All done.
```

`nisinit` ユーティリティは、クライアントの `/etc/hosts` ファイルまたは `/etc/inet/ipnodes` ファイル内でサーバーのアドレスを探します。したがって、サーバーにドメイン名を付加しないでください。ドメイン名を付加した場合、このユーティリティはサーバーのアドレスを見つけることができません。

## 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

ここでは、NIS+ クライアントを初期設定するために、別の NIS+ クライアント (できれば同じドメインから) の `COLD_START` を使用します。NIS+ クライアントを設定する方法としてはこれが最も安全です。これにより、クライアントは、信頼できるサーバーから確実に NIS+ 情報を得ることができます。これはホスト名やブロードキャストによる初期化では保証されません。

## セキュリティ上の留意点

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。

## 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。

## NIS+ クライアントを初期設定する — タスクマップ

表 6-5 NIS+ クライアントを初期設定する

タスク	説明	指示の参照先
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	134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. 他のクライアントの COLD\_START ファイルをコピーします。

他のクライアントの 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 COLD\_START 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。これは最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方法のまとめ

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

表 6-6 クライアントを設定する方法のまとめ 続く

作業	コマンド
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	<code>client1# more /etc/nsswitch.conf</code>
Diffie-Hellman キー長を設定する	<code>client1# nisauthconf dh640-0 des</code>
<code>/var/nis</code> の下のファイルを削除する	<code>client1# rm -rf /var/nis/*</code>
クライアントを初期設定する	<code>client1# nisinit -c -H rootmaster</code>
キーサーバーのプロセスを終了して、再起動する	<code>client1# ps -ef   grep keyserver</code> <code>client1# kill -9 process-id</code> <code>client1# keyserver</code>
クライアント上でキーログインを実行する	<code>client1# keylogin -r</code> password:
クライアントを再起動する	<code>client1# init 6</code>



## 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。

- 137ページの「NIS+ サーバーを設定する」
- 142ページの「既存のドメインに複製サーバーを追加する」

この章で説明する内容は、ルートマスターを除くすべての NIS+ サーバーに適用されます。つまり、NIS 互換モードで稼動しているかどうかにかかわらず、ルート複製サーバー、非ルートマスターサーバー、または非ルート複製サーバーに適用されます。章末では個々の作業についてまとめています。

既存の NIS+ サーバーの機能を停止し、他のマシンで置き換える方法については、『Solaris ネーミングの管理』を参照してください。

---

### NIS+ サーバーを設定する

NIS+ サーバーを設定する作業は、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。

## 標準構成と NIS 互換構成の手順の相違

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 (96ページの「標準構成と NIS 互換構成の手順の相違」を参照)。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` オプションを追加する) 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。これについては、「NIS+ サーバーを構成する方法」の140ページの手順 2 (標準の NIS+ サーバーの場合は、141ページの手順 3) で説明します。

---

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。この副デーモンは、主デーモンを終了させた場合は、必ず別個に終了させなければなりません。

---

構成プロセス全体のまとめを次に示します。

1. 新しい複製サーバー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` を指定して起動します (NIS 互換の場合のみ)。
3. NIS+ デーモンを起動します (標準の NIS+ の場合のみ)。

## セキュリティ上の留意点

---

注 - NIS+ のセキュリティシステムは複雑です。NIS+ のセキュリティに精通していない場合は、『Solaris ネーミングの管理』のセキュリティ関連の章をもう一度読んでから NIS+ 環境の構成を始めてください。

---

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。

---

## 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
- サーバーが NIS 互換モードで稼働し、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 (第 1 章を参照)

## 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+ サーバーを初期設定する — タスクマップ

表 7-1 NIS+ サーバーを初期設定する

タスク	説明	指示の参照先
NIS+ サーバーを初期設定する	NIS 互換または NIS+ のみをサポートするように、NIS+ サーバーを初期設定する	139ページの「NIS+ サーバーを構成する方法」

### ▼ NIS+ サーバーを構成する方法

1 つ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、あまりお勧めしません。

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、120ページの「クライアントを構成する」に従って、ワークステーションを NIS+ クライアントとして設定した後、これを再起動したことを前提としています。ワークステーションを再起動すると、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。ワークステーションを再起動しなかった場合、ここでキャッシュマネージャ (nis\_cachemgr) を再起動します。

2. NIS+ デーモンを -Y で起動します (NIS 互換のみの場合)。

この手順は、サーバーを NIS 互換モードで設定する場合にだけ実行します。標準の NIS+ サーバーを設定する場合は、この代わりに 141ページの手順 3を実行します。

この手順には、NIS クライアントの DNS 転送機能をサポートするための操作説明も含まれています。この手順は、2つに分かれています。最初の手順では、NIS+ デーモンを NIS 互換モードで起動します。2つめの手順では、サーバーが再起動されたときに、NIS+ デーモンが NIS 互換モードで再起動するように設定します。

- a. rpc.nisd に -Y と -B のフラグを付けて実行します。

```
compatserver# rpc.nisd -Y -B
```

-Y オプションは、NIS+ 要求だけでなく NIS 要求にも応答するインタフェースを呼び出します。-B オプションは DNS 転送をサポートします。

- b. /etc/init.d/rpc ファイルを編集します。

/etc/init.d/rpc ファイル内で文字列 EMULYP="-Y" を検索して、この文字列を含む行のコメント指定を解除します。

DNS 転送機能を使用するには、EMULYP="-Y" に -B フラグを追加します。DNS 転送機能が必要ない場合は、コメント指定の解除だけを実行し、-B フラグは追加しないでください。

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は、nislog コマンドを使用して調べることができます。詳細は、『Solaris ネーミングの管理』のディレクトリ関連の章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 リリース 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、それに関わりのあ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、マスターサーバー、複製サーバーのいずれにするかを決定します (第 8 章を参照)。この作業はこの手順で完了です。作業の要約については、150 ページの「サーバー構成の要覧」を参考にしてください。

### 3. NIS+ デーモンを起動します (標準の NIS+ のみの場合)。

rpc.nisd コマンドを実行します。

```
server# rpc.nisd
```

NIS+ デーモンが本当に実行されていることを確認するには、次のように ps コマンドを実行します。

```
server# ps -ef | grep rpc.nisd
root 1081 1 16:43:33 ? 0:01 rpc.nisd
root 1087 1004 11 16:44:09 pts/1 0:00 grep rpc.nisd
```

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 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
```

compatserver.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は、nislog コマンドを使用して調べることができます。詳しくは、『Solaris ネーミングの管理』のディレクトリ関連の章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 リリース 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、それに関わりのあ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 150 ページの「サーバー構成の要覧」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

NIS+ サービスを常に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で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」

複数のサブネットで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、複製サーバーの分散配置については、『NIS+ への移行』を参照してください。

理想的な複製サーバーの数については、『Solaris ネーミングの管理』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、75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、時間が長くなる場合があります。

#### ■ 「NIS+ コマンドセット」

NIS+ コマンドを使うには、144ページの「NIS+ コマンドによる複製サーバーの構成」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。nisbackup と nisrestore を使うと、nisserver スクリプトを使うより、短時間でデータセットをロードできます。

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 スクリプトと NIS+ コマンドのどちらを使った場合でも、新しい複製サーバーの構成が完了すると、nisping を使用することで、該当する名前空間のデータセットがマスターサーバーにより、ネットワーク経由で自動的に新しい複製サーバーにロードされます。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、148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 (146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+ コマンドによる複製サーバーの構成

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。

### セキュリティ上の留意点

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。

### 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 (137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— タスクマップ

表 7-2 NIS+ コマンドを使って複製サーバーを構成する

タスク	説明	指示の参照先
NIS+ サーバーを設定する	NIS+ コマンドを使って NIS+ サーバーを設定する	145ページの「NIS+ コマンドを使って複製サーバーを構成する」

### ▼ NIS+ コマンドを使って複製サーバーを構成する

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. `rpc.nisd` が稼働中であることを確認します。
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** - `nismkdir` 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 `nismkdir` コマンドを実行すると、マスターサーバーと複製サーバーとの間で通信上の問題が生じます。

---

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。NIS+ データセットのロードには、2 通りの方法があります。

- 「`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。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に対する応答が遅れることがあります (詳細は、148 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)。

- 「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 (146 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)。他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。

### **セキュリティ上の留意点**

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。

## 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 (137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 (144ページの「NIS+ コマンドによる複製サーバーの構成」を参照)

## nisrestore を使ってデータを複製サーバーにロードする — タスクマップ

表 7-3 nisrestore を使ってデータを複製サーバーにロードする

タスク	説明	指示の参照先
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	147ページの「nisrestore を使って名前空間データをロードする」

## ▼ nisrestore を使って名前空間データをロードする

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. 複製サーバー上の rpc.nisd を終了させます。  
マスターサーバーから複製サーバーへの名前空間データの自動ロード (nisping による) が中断されます。
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。  
この手順は『Solaris ネーミングの管理』に詳しく説明されています。以下の例では、nisbackup コマンドを使って master1 を /var/master1\_bakup ディレクトリにバックアップします。

```
master1# nisbackup -a /var/master1_bakup
```

`nisrestore` を使って新しい複製サーバーを構成する最も簡単な方法は、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 (148ページの手順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。

この手順は『*Solaris* ネーミングの管理』に詳しく説明されています。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で使うものである場合、あるいは `nisrestore` が必要なデータを検証または見つけることができないという旨のエラーメッセージが出た場合は、次に示すように `-f` オプション付きで実行してみてください。

```
replica2# nisrestore -f -a /var/master1_backup
```

4. 新しい複製サーバー上で `rpc.nisd` を再実行します。

詳細は、139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。

## セキュリティ上の留意点

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。

## 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 (137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 (144ページの「NIS+ コマンドによる複製サーバーの構成」を参照)

## nisping を使ってデータを複製サーバーにロードする — タスクマップ

表 7-4 nisping を使ってデータを複製サーバーにロードする

タスク	説明	指示の参照先
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	149ページの「nisping を使って名前空間データをロードする方法」

### ▼ nisping を使って名前空間データをロードする方法

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.
```

次のような画面が表示されます。

```
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の詳細は、『Solaris ネーミングの管理』のディレクトリ関連の章を参照してください。

---

## サーバー構成の要覧

表 7-5、表 7-6 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、最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する必要があります。また、ここでは各コマンドに対するサーバーの応答は割愛しています。

表 7-5 複製サーバー replica2 を doc.com. に追加する

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 - 上記の例でも説明したように、新しい複製サーバーにデータをロードするのであれば、nisping を使うより NIS+ のバックアップ機能や復元機能を使った方が簡単です。詳細は、146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」と『Solaris ネーミングの管理』を参照してください。

表 7-6 非ルートマスターサーバーを起動するには

作業	コマンド
ルート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 -B フラグを使用してデーモンを起動する	server# rpc.nisd -Y - B
NIS 互換モードの場合のみ: EMULYP= -Y -B に変更する	server# vi /etc/inet.d/rpc
NIS+ モードの場合のみ: デーモンを起動する	server# rpc.nisd





## 非ルートドメインの構成

---

この章では、NIS+ コマンドセットを使ってサブドメイン (非ルートドメイン) 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。

- 153ページの「非ルートドメインを設定する」
- 163ページの「サブドメイン構成の要覧」

作業のまとめは、表 8-2 にあります。

---

### 非ルートドメインを設定する

---

注 - 非ルートドメインを設定する作業は、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する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。

最初に非ルートドメインのサーバーを構成してから、非ルートドメインを構成してください。

非ルート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、最も効率的な順序に並べています。

## 標準構成と NIS 互換構成の手順の相違

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 (96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。

構成プロセス全体のまとめは次のとおりです。

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します。
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設定します。
7. ドメインの管理グループにサーバーを追加します。
8. 他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。

## セキュリティ上の留意点

注 - NIS+ のセキュリティシステムは複雑です。NIS+ のセキュリティに精通していない場合は、『Solaris ネーミングの管理』のセキュリティ関連の章を読んでから NIS+ 環境の構成を始めてください。

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---rmdrmdr---  
:
```

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、『Solaris ネーミングの管理』のグループ関連の章で説明する `nisgrpadm` コマンドを使用します。

## 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 (157ページの手順 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 (156ページの手順 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) (160ページの手順 8)

## 非ルートドメインを設定する — タスクマップ

表 8-1 非ルートドメインを設定する

タスク	説明	指示の参照先
非ルートドメインを設定する	NIS+ コマンドを使って非ルートドメインを設定する	156ページの「非ルートドメインを設定する方法」

## ▼ 非ルートドメインを設定する方法

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すべてのアクセス権があります。

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、159ページの手順 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、158ページの手順 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。いずれも `NIS_GROUP` に `admin.sales.doc.com.` を設定します。

「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します。

`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` 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 `nismkdir` コマンドを実行するとマスターサーバーと複製サーバーとの間で通信上の問題が生じます。

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 : ----rmdir---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が次のように表示されます。

```
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
```

(続く)

```
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+ ファイルと共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、156 ページの手順 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ます。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.
```

この手順はグループを作成するだけであり、メンバーの指定は行いません。指定は 161 ページの手順 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` コマンドの詳細な使用法は、『Solaris ネーミングの管理』のアクセス権に関連する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.
```

サーバーが実際にグループのメンバーであることを確認するには、`-l` オプションを付けて `nisgrpadm` コマンドを実行します (『Solaris ネーミングの管理』のグループに関する章を参照)。

```
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. 他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他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。しかし、これはスーパーユーザーとして実行しなければなりません。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、他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。ネットワークパスワードを変更する方法については、『*Solaris* ネーミングの管理』の資格および鍵に関する章を参照してください。

---

注 - 上記の 160 ページの手順 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しないでください。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。

この手順は、他の管理者がダミーパスワードを変更していなくても実行できます。-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源と 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、最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、作業の詳細について十分に理解しておいてください。また、この表では各コマンドに対するサーバーの応答は割愛しています。

表 8-2 まとめ - サブドメインを設定する方法

作業	コマンド
ドメインマスターにスーパーユーザーとしてログインする	<code>smaster% su</code>
ドメインの管理グループを指定する	<code># NIS_GROUP=admin.sales.doc.com.</code> <code># export NIS_GROUP</code>
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	<code># nismkdir -m smaster -s salesreplica sales.doc.com.</code>
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	<code># /usr/lib/nis/nissetup sales.doc.com.</code>
管理グループを作成する	<code># nisgrpadm -c admin.sales.doc.com.</code>
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	<code># nischmod g+rmcd sales.doc.com.</code>
管理グループにサーバーを追加する	<code># nisgrpadm -a admin.sales.doc.com. smaster.doc.com. \ sreplica.doc.com.</code>

表 8-2 まとめ - サブドメインを設定する方法 続く

作業	コマンド
他の管理者の資格を追加する	<pre># nisaddcred -p 22244 -P juan.sales.doc.com. local # nisaddcred -p unix.22244@sales.doc.com. \ -P juan.sales.doc.com. DES</pre>
ドメインの管理グループに管理者を追加する	<pre># nisgrpadm -a admin.sales.doc.com. juan.sales.doc.com.</pre>

## 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。

- 166ページの「テーブルの生成の方法」
- 167ページの「NIS+ テーブルをファイルから生成する方法」
- 175ページの「NIS+ テーブルを NIS マップから生成する方法」
- 182ページの「NIS+ から NIS に情報を転送する」
- 183ページの「所有者および管理者に対する Passwd へのアクセス制限」

---

### 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、Solstice AdminSuite 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。(167ページの「NIS+ テーブルをファイルから生成する方法」参照)
- NIS マップから作成する。(175ページの「NIS+ テーブルを NIS マップから生成する方法」参照)
- nispopulate スクリプトを使用して作成する。(59ページの「NIS+ テーブルの生成 (populate)」および 84ページの「新しいサブドメインのテーブルの生成」)
- Solstice AdminSuite 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。

- 情報源内だけに存在するエントリは、テーブルに追加する
- 情報源とテーブルの両方に存在するエントリは、テーブルで更新する
- NIS+ テーブルだけに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、内容があまり変わらないファイルやマップで更新する場合、マージ用のオプションを使用すると、サーバーの処理を大幅に減らすことができます。情報源内で重複していないエントリだけを削除する (置換オプションでは、単純にすべてのエントリを削除する) ため、重複するエントリに対する削除と追加の処理を 1 回ずつ少なくできます。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方法

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。

手順の概要は次のとおりです。

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します。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。詳細は、168ページの「ファイルのセキュリティ上の留意点」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
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` ファイルを転送します。
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. テーブルにチェックポイントを実行します。

## ファイルのセキュリティ上の留意点

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は必要です。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+ のセキュリティに精通していない場合は、『Solaris ネーミングの管理』のセキュリティ関連の章を読んでから NIS+ 環境の設定を始めてください。

---

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。

## 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なければなりません。



- 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です。NIS+ の要件については、『NIS+ への移行』を参照してください。
- 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、59ページの「nispopulate を実行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ないようにしてください。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ません。たとえば、sales.alpha というマシン名は無効です。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 — タスクマップ

表 9-1 NIS+ テーブルをファイルから生成する

タスク	説明	指示の参照先
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	169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. 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します。

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、この操作を実際に行っている (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がありません。ここでは、C シェルユーザーの場合と Bourne シェルまたは Korn シェルのユーザーの場合の 2 つの例を示します。

C シェルの場合

```
rootmaster# setenv PATH $PATH:/usr/lib/nis
```

## Bourne 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 度に 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、173ページの手順 6 に、automounter ファイルの場合は、173ページの手順 7 に、passwd ファイルと shadow ファイルの場合は、173ページの手順 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]
```

初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を使用します。ルートドメインのマスターサーバー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一つのドメインのクライアントです。たとえば、`sales.doc.com`。マスターサーバーは `doc.com`。ドメインに所属します。そのマスターサーバーからテーブルを `sales.doc.com`。ドメインに生成するには、`nisaddent` 文に `sales.doc.com`。ドメイン名を追加しなければなりません。たとえば、次のようになります。

```
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。`nisaddent` については、『Solaris ネーミングの管理』のテーブルに関連する章を参照してください。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。詳細は、『Solaris ネーミングの管理』のテーブルに関する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
.
```

障害追跡のこつ：`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。定期的に行うにしてもいいのですが、重要な変更を加えた場合は、できるだけ早い機会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は、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方法

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します。
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。
4. `nisaddent` を使用して、次のマップを 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. テーブルにチェックポイントを実行します。

## マップのセキュリティ上の留意点

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。

## 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る。また、ユーザーと同じ名前の付いたマシンは使用できない
- マシン名にはドット (ピリオド) を使用できない。たとえば、`sales.alpha` というマシン名にはできない。`sales-alpha` というマシン名は可能



## 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 — タスクマップ

表 9-2 NIS+ テーブルを NIS マップから生成する

タスク	説明	指示の参照先
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	177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します。

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、この操作を実際に行っている (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 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 ファイル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、「NIS+ テーブルをファイルから生成する方」の 173 ページの手順 6 に、オートマウントファイルの場合は、173 ページの手順 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
```

初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。

```
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` コマンドの詳細は、『*Solaris* ネーミングの管理』のテーブルに関連する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、ログインしているワークステーショ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. オートマウンタ情報を転送します。

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、オートマウンタの「テキストファイル」を転送するときに必要なように、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上の留意点

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。

## 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 — タスクマップ

表 9-3 NIS+ から NIS に情報を転送する

タスク	説明	指示の参照先
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	182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 列のセキュリティ上の留意点

- 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。

## 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 — タスクマップ



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

タスク	説明	指示の参照先
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	185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。詳細は、『Solaris ネーミングの管理』のアクセス権に関連する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、185 ページの手順 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、最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 まとめ : NIS+ テーブルへのファイルの転送

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.xfer cred
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 まとめ : NIS+ テーブルへのマップの転送

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

表 9-6 まとめ : NIS+ テーブルへのマップの転送 続く

作業	コマンド
publickey マップをファイルにダンプする	<code># makedbm -u /var/yp/olddoc/publickey.byname &gt; \ /etc/publickey.xfr</code>
新しい資格を削除する	<code># vi /etc/publickey.xfr</code>
publickey ファイルを転送する	<code># nisaddent -a -f /etc/publickey.xfr -t cred.org_dir \ publickey</code>
オートマウントマップを転送する	<code># 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</code>
テーブルディレクトリにチェックポイントを実行する	<code># nisping -C org_dir</code>

表 9-7 まとめ : NIS+ テーブルを NIS マップに転送する

作業	コマンド
NIS+ サーバーにログインする	<code>dualserver#</code>
NIS+ テーブルをファイルに転送する	<code># /usr/lib/nis/nisaddent -d [-ttable] tabletype &gt; filename</code>
ファイルを NIS マップに転送する	<code># makedbm flags output-file NIS-dbm-file</code>

表 9-8 まとめ：パスワード列へのアクセス権を変更する

作業	コマンド
ドメインのマスタサーバーにログインする	<code>rootmaster#</code>
テーブルの既存の権利をチェックする	<code># niscat -o passwd.org_dir</code>
テーブルに新しい権利を割り当てる	<code># nischmod og=rmcd,nw= passwd.org_dir</code>
列に新しい権利を割り当てる	<code># nistbladm -u passwd=o+r shadow=n+r passwd.org_dir</code>
新しい権利を確認する	<code># niscat -o passwd.org_dir</code>



## パート III NIS の設定と構成

---

パート III では NIS (Network Information Service) サービスの設定と構成の方法を説明します。次の 1 つの章で構成されます。

- 第 10 章





## NIS サービスの構成

---

この章では、NIS (Network Information Service) の初期設定と構成を説明します。

- 193ページの「Solaris 8 オペレーティング環境の NIS」
- 194ページの「NIS の構成を始める前に」
- 194ページの「NIS ドメインの計画」
- 195ページの「NIS の構成手順」
- 196ページの「マスターサーバーの準備」
- 212ページの「マスターサーバー上での NIS サービスの開始」
- 213ページの「NIS スレーブサーバーの設定」
- 216ページの「NIS クライアントの設定」
- 217ページの「NIS マップ」

NIS の一般的な説明と概要については、『Solaris ネーミングの管理』を参照してください。

---

### Solaris 8 オペレーティング環境の NIS

次の NIS の機能は新規のものか、または Solaris 8 オペレーティング環境で変更されました。

## NIS と IPv6

新しい ipnodes マップ (ipnodes.byaddr および ipnodes.byname) が、NIS に追加されました。これらのマップは、IPv4 および IPv6 の両方のアドレスを格納しています (ipnodes(4) のマニュアルページを参照)。NIS クライアントとサーバーは、IPv4 または IPv6 のどちらかの RPC トランスポートを使用して通信することができます。

---

## NIS の構成を始める前に

NIS の構成を始める前に、以下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 ファイルを適切に構成する (詳細は、第 1 章を参照)
- NIS ドメインの計画 (194 ページの「NIS ドメインの計画」を参照)

---

## NIS ドメインの計画

NIS のサーバーやクライアントとしてマシンの構成を始める前に、NIS ドメインの計画が必要です。

### ドメインの計画

NIS ドメインに入れるマシンを決めてください。NIS ドメインはネットワークと同一である必要はありません。ネットワークには複数の NIS ドメインが存在でき、また NIS ドメインに属さないマシンもネットワーク上に存在できます。

NIS ドメイン名を選択します。NIS ドメイン名は、最長 256 文字の長さにはできますが、短い名前の方が実用的です。ドメイン名が 32 文字を超えないように制限するとよいでしょう。ドメイン名は大文字と小文字を区別します。便宜上、インターネットのドメイン名を NIS ドメイン名のもととして使用できます。たとえば、インターネットのドメイン名が doc.com の場合には、NIS ドメイン名を doc.com にします。doc.com を 2 つのドメインに分けて、1 つを営業部門に、もう片方を製造部門に使用する場合には、1 つは sales.doc.com、もう片方は manf.doc.com と名前をつけます。

マシンが NIS サービスを使用できるようになる前に、正確な NIS ドメイン名とマシン名を設定する必要があります。マシン名は、マシンの `/etc/nodename` ファイルによって設定され、マシンのドメイン名は、マシンの `/etc/defaultdomain` ファイルによって設定されます。これらファイルはブート時に読み取られ、内容はそれぞれ `uname -S` と `domainname` コマンドに使用されます。ディスクレスマシンはブートサーバーからファイルを読み取ります。

## NIS サーバーの選択

NIS サーバーになるマシンを決定します。マスターサーバーになるマシン (後で変更可能) を 1 つ選択します。スレーブサーバーを作る場合には、それを決定します (NIS と NIS の条件の概要については、『Solaris ネーミングの管理』を参照)。

## NIS クライアントマシンの選択

NIS クライアントになるマシンを決定します。一般的には、ドメイン内のすべてのマシンは NIS クライアントになるように設定されますが、これは必須ではありません。

---

## NIS の構成手順

Solaris 8 リリースソフトウェアのインストール、`nsswitch.conf` ファイルの設定、ドメインの計画が終わったら、次の NIS の構成手順を実行する必要があります。

1. マスターサーバーの準備をする (196ページの「マスターサーバーの準備」を参照)
2. NIS のマスターサーバーを構成する (210ページの「ypinit によるマスターサーバーの設定」を参照)
3. マスターサーバーで NIS デーモンを起動する (212ページの「マスターサーバー上での NIS サービスの開始」を参照)
4. スレーブサーバーを構成する (213ページの「NIS スレーブサーバーの設定」を参照)

## 5. NIS クライアントマシンを構成する (216ページの「NIS クライアントの設定」を参照)

---

注・文脈によっては、「マシン」名は「ホスト」名や「ワークステーション」名と呼ばれる場合があります。この節では「マシン」を使用しますが、画面メッセージや NIS マップ名によっては、「ホスト」や「ワークステーション」を使用する場合があります。

---

次の節ではこれら手順の詳細を説明します。

---

## マスターサーバーの準備

マスターサーバーの設定では、マスターサーバーの情報源 (入力テキストファイル) の NIS マスターサーバーマップへの変換が必要です。これを行う前にすべき準備があります。

## 入力ファイルディレクトリ

入力ファイルはマスターサーバーの `/etc` ディレクトリか、その他のディレクトリに置かれます。`/etc` に置くのは望ましくありません。マップの内容が、マスターサーバーのローカルファイルの内容と同じになるからです。これは `passwd` と `shadow` のファイルに特有の問題です。ユーザー全員がマスターサーバーのマップにアクセスして、`passwd` マップを通じて YP クライアントのすべてに `root` のパスワードが渡されるためです。この詳細は、197ページの「`passwd` ファイルと名前空間のセキュリティ」を参照してください。

他のディレクトリにソースファイルを置く場合に、*your-choice* がソースファイルの格納に使用するディレクトリの名前であるときは、`/var/yp` 内の `Makefile` を修正して、`DIR=/etc` 行を `DIR=/your-choice` に変更する必要があります。これによって、クライアント上のファイルのようにサーバー上のローカルファイルを取り扱うことができます。`Makefile` を編集する前に、編集前の `Makefile` を別の名前のファイルに保存しておくことをお勧めします。

## passwd ファイルと名前空間のセキュリティ

passwd マップは特殊なケースです。古い Solaris 1.x の passwd ファイル形式の他に、この NIS の実装は、NIS のパスワードマップの構築用の入力として Solaris 8 リリースの /etc/passwd と /etc/shadow ファイル形式を受け付けます。

セキュリティ上の理由で未承認のルートアクセスを防ぐために、NIS のパスワードマップの構築に使用されるファイルには、root に対するエントリを含めないでください。このため、パスワードマップはマスターサーバーの /etc ディレクトリに置かれたファイルから構築しないでください。パスワードマップの構築に使用されるパスワードファイルは、root エントリが削除されていて、また未承認アクセスから保護されるディレクトリに置かれる必要があります。

たとえば、マスターサーバーのパスワード入力ファイルは、/var/yp のようなディレクトリや、または選択した任意のディレクトリに格納し、ファイル自体は他のファイルへのリンクにはせず、またその位置を Makefile に指定します。/usr/lib/netsvc/yp/ypstart スクリプトは、Makefile に指定された構成に従って自動的に正確なディレクトリオプションを設定します。

入力ファイルが /etc 以外のディレクトリにある場合には、PWDIR (パスワードマクロ) を変更して、passwd と shadow のファイルのあるディレクトリを参照するようにします。passwd マップ入力ファイルの格納に使用するディレクトリ名が *your-choice* の場合には、PWDIR=/etc の行を PWDIR=*your-choice* に変更します。



---

注意 - ディレクトリ内で PWDIR によって指定された passwd ファイルに root に対するエントリが含まれないようにしてください。

---

## マスターサーバーの準備 — タスクマップ

表 10-1 マスターサーバーの準備

タスク	説明	指示の参照先
マスターサーバーの準備	NIS マップへのファイルの変換	198ページの「NIS マップへの変換用ソースファイルの準備」
マスターサーバーの準備	ypinit を使用して設定する	210ページの「ypinit によるマスターサーバーの設定」

## ▼ NIS マップへの変換用ソースファイルの準備

NIS マップへの変換用の入力ファイルを準備します。

1. マスターサーバーの入力ファイルを確認して、システム環境の最新の状態が反映されていることを確認します。

次のファイルを確認します。

- auto.home または auto\_home
- auto.master または auto\_master
- bootparams
- ethers
- group
- hosts
- ipnodes
- netgroup
- netmasks
- networks
- passwd
- protocols
- rpc
- services

■ shadow

2. passwd 以外のこれら入力ファイルのすべてを、選択した DIR ディレクトリにコピーします。
3. passwd ファイルを選択した PWDIR ディレクトリにコピーします。
4. /etc/mail/aliases ファイルを確認します。  
他の入力ファイルと異なり、/etc/mail/aliases ファイルは別のディレクトリに移動できません。このファイルは /etc/mail ディレクトリになければなりません。ドメイン全体で利用しようとするメールエイリアスがすべて含まれていることを検証して、/etc/mail/aliases ファイルが完全なことを確認します。詳細は、aliases(4) のマニュアルページを参照してください。
5. 入力ファイルからすべてのコメント、その他の余計な行や情報を取り除きます。これら操作は、sed または awk のスクリプトを通じて、またはテキストエディタを使用して行えます。Makefile は入力ファイルから不要なエントリをある程度自動的に削除しますが、手作業で検証し、取り除いてから、実行するようにしてください。
6. すべての入力ファイルのデータが正しい形式になっていることを確認します。入力ファイルのデータは、それぞれのファイル用の形式になっている必要があります。該当するマニュアルページを参照して、各ファイルが正しい形式になっていることを確認します。

## Makefile の準備

入力ファイルを確認して、それを入力ファイルディレクトリにコピーした後に、NIS サービスが使用する ndbm 形式のマップに、その入力ファイルを変換する必要があります。210ページの「ypinit によるマスターサーバーの設定」の節で説明するとおり、これはマスターサーバーで ypinit が実行されると自動的に行われます。

ypinit スクリプトは、プログラム make を呼び出します。このプログラムは、/var/yp ディレクトリに置かれた Makefile を使用します。例 10-1 のようなデフォルトの Makefile が、/var/yp ディレクトリで提供されます。この Makefile は、入力ファイルの ndbm 形式マップへの変換に必要なコマンドを含みます。

Makefile はそのまま使用でき、必要なら修正もできます。デフォルトの Makefile を修正するときは、将来必要な場合に備えて最初にオリジナルのデフォルト Makefile をコピーして格納してください。次の Makefile への修正のうちの 1 つまたは複数を実行する必要がある場合があります。

■ 「デフォルトではないマップ」

デフォルトではない自分専用の入力ファイルを作成していて、それを NIS マップに変換する場合には、その入力ファイルを Makefile に追加する必要があります。

■ DIR の値

196ページの「入力ファイルディレクトリ」で説明するように、/etc 以外のディレクトリに格納されたソースファイルを Makefile に使用させるには、Makefile の DIR の値を使用するディレクトリに変更する必要があります。この値を Makefile で変更するときは、行をインデントしないでください。

■ PWDIR の値

/etc 以外のディレクトリに格納された passwd、shadow、または adjunct 入力ファイルを Makefile に使用させるには、Makefile の PWDIR の値を使用するディレクトリに変更する必要があります。この値を Makefile で変更するときは、行をインデントしないでください。

■ 「ドメイン名リゾルバ」

現在のドメインにはないマシンに対して、NIS サーバーがドメイン名リゾルバを使用するようにするには、Makefile の行 B= をコメント化し、B= -b のコメントを解除 (有効化) します。

例 10-1 修正前のデフォルトの Makefile

```
#
# Copyright (c) 1996-1999, by Sun Microsystems, Inc.
# All rights reserved.
#
#pragma ident "@(#)Makefile 1.25 99/06/01 SMI"
#
#----
# It is somewhat confusing to note that Solaris 2.x uses /etc/auto_master
# instead of the 4.x /etc/auto.master file name because of NIS+ treating a
# "." in a special way.
#
# Set the following variable to "-b" to have NIS servers use the domain name
# resolver for hosts not in the current domain.
#B=-b
```

(続く)



```

B=
DIR =/etc
#
# If the ipnodes (IPv6 hosts file) lives in a directory other than
# /etc/inet, then you'll need to change the following line.
#
INETDIR=/etc/inet
#
# If the passwd, shadow and/or adjunct files used by rpc.yppasswdd
# live in directory other than /etc then you'll need to change the
# following line.
# DO NOT indent the line, however, since /etc/init.d/yp attempts
# to find it with grep "^PWDIR" ...
#
PWDIR =/etc
DOM = `domainname`
NOPUSH = ""
ALIASES = /etc/mail/aliases
YPDIR=/usr/lib/netsvc/yp
SBINDIR=/usr/sbin
YPBBDIR=/var/yp
YPPUSH=${YPDIR}/yppush
MAKEDBM=${SBINDIR}/makedbm
MULTI=${YPDIR}/multi
REVNETGROUP=${SBINDIR}/revnetgroup
STDEETHERS=${YPDIR}/stdethers
STDHOSTS=${YPDIR}/stdhosts
MKNETID=${SBINDIR}/mknetid
MKALIAS=${YPDIR}/mkalias

CHKPIPE= || ( echo "NIS make terminated:" $@ 1>&2; kill -TERM 0 )

k:
@if [ ! ${NOPUSH} ]; then ${MAKE} ${MFLAGS} -k all; \
else ${MAKE} ${MFLAGS} -k all NOPUSH=${NOPUSH};fi

all: passwd group hosts ipnodes ethers networks rpc services protocols \
netgroup bootparams aliases publickey netid netmasks c2secure \
timezone auto.master auto.home \
auth.attr exec.attr prof.attr user.attr audit.user

c2secure:
-@if [ -f ${PWDIR}/security/passwd.adjunct ]; then \
if [ ! ${NOPUSH} ]; then ${MAKE} ${MFLAGS} -k \
passwd.adjunct.time group.adjunct.time; \
else ${MAKE} ${MFLAGS} -k NOPUSH=${NOPUSH} \
passwd.adjunct.time group.adjunct.time; \
fi; \
fi

passwd.time: ${PWDIR}/passwd ${PWDIR}/shadow
-@if [ -f ${PWDIR}/security/passwd.adjunct ]; then \
(nawk 'BEGIN { FS=":"; OFS=":" } /^[a-zA-Z0-9_]/ { $$2 = "##" $$1; \

```

(続く)

```

printf "%s\t%s\n", $$1, $$0 }' $(PWDIR)/passwd $(CHKPIPE) | \
$(MAKEDBM) - $(YPDBDIR)/$(DOM)/passwd.byname; \
(nawk 'BEGIN { FS=":"; OFS=":" } /^[a-zA-Z0-9_]/ { $$2 = "##" $$1; \
printf "%-10d\t%s\n", $$3, $$0 }' $(PWDIR)/passwd $(CHKPIPE) | \
$(MAKEDBM) - $(YPDBDIR)/$(DOM)/passwd.byuid; \
elif [ -f $(PWDIR)/shadow ]; then \
(nawk 'BEGIN { FS=":"; OFS=":"; while ( getline < "$(PWDIR)/shadow" > 0) \
shadow[$$1] = $$2; } /^[a-zA-Z0-9_]/ { $$2 = shadow[$$1]; \
printf "%s\t%s\n", $$1, $$0 }' $(PWDIR)/passwd $(CHKPIPE) | \
$(MAKEDBM) - $(YPDBDIR)/$(DOM)/passwd.byname; \
(nawk 'BEGIN { FS=":"; OFS=":"; while ( getline < "$(PWDIR)/shadow" > 0) \
shadow[$$1] = $$2; } /^[a-zA-Z0-9_]/ { $$2 = shadow[$$1]; \
printf "%-10d\t%s\n", $$3, $$0 }' $(PWDIR)/passwd $(CHKPIPE) | \
$(MAKEDBM) - $(YPDBDIR)/$(DOM)/passwd.byuid; \
else \
(awk 'BEGIN { FS=":"; OFS="\t"; } /^[a-zA-Z0-9_]/ { print $$1, $$0 }' \
$(PWDIR)/passwd $(CHKPIPE)) | $(MAKEDBM) - $(YPDBDIR)/$(DOM)/passwd.byname; \
(awk 'BEGIN { FS=":"; OFS="\t"; } /^[a-zA-Z0-9_]/ { printf("%-10d ", $$3); \
print $$0 }' $(PWDIR)/passwd $(CHKPIPE)) | $(MAKEDBM) - \
$(YPDBDIR)/$(DOM)/passwd.byuid; \
fi
@touch passwd.time;
@echo "updated passwd";
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) passwd.byname; fi
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) passwd.byuid; fi
@if [ ! $(NOPUSH) ]; then echo "pushed passwd"; fi

group.time: $(DIR)/group
@(awk 'BEGIN { FS=":"; OFS="\t"; } /^[a-zA-Z0-9_]/ { print $$1, $$0 }' \
$(DIR)/group $(CHKPIPE)) | $(MAKEDBM) - $(YPDBDIR)/$(DOM)/group.byname;
@(awk 'BEGIN { FS=":"; OFS="\t"; } /^[a-zA-Z0-9_]/ { printf("%-10d ", $$3); \
print $$0 }' $(DIR)/group $(CHKPIPE) | $(MAKEDBM) - \
$(YPDBDIR)/$(DOM)/group.bygid;
@touch group.time;
@echo "updated group";
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) group.byname; fi
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) group.bygid; fi
@if [ ! $(NOPUSH) ]; then echo "pushed group"; fi

ipnodes.time: $(INETDIR)/ipnodes
@($ (MULTI) -n $(B) -l $(INETDIR)/ipnodes);
@($ (STDHOSTS) -n $(INETDIR)/ipnodes $(CHKPIPE)) | \
(awk 'BEGIN { OFS="\t"; } $$1 !~ /^#/ { print $$1, $$0 }' $(CHKPIPE) | \
$(MAKEDBM) $(B) - $(YPDBDIR)/$(DOM)/ipnodes.byaddr;
@touch ipnodes.time;
@echo "updated ipnodes";
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) ipnodes.byname; fi
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) ipnodes.byaddr; fi
@if [ ! $(NOPUSH) ]; then echo "pushed ipnodes"; fi

hosts.time: $(DIR)/hosts
@($ (MULTI) $(B) -l $(DIR)/hosts);
@($ (STDHOSTS) $(DIR)/hosts $(CHKPIPE)) | \
(awk 'BEGIN { OFS="\t"; } $$1 !~ /^#/ { print $$1, $$0 }' $(CHKPIPE) | \

```

(続く)

```

$(MAKEDBM) $(B) - $(YPDBDIR)/$(DOM)/hosts.byaddr;
@touch hosts.time;
@echo "updated hosts";
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) hosts.byname; fi
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) hosts.byaddr; fi
@if [ ! $(NOPUSH) ]; then echo "pushed hosts"; fi

ethers.time: $(DIR)/ethers
@($(STDETHERS) $(DIR)/ethers $(CHKPIPE)) \
| (awk '{print $$1, $$0; for (i = 3; i <= NF; i++) print $$i, $$0}' $(CHKPIPE)) \
| $(MAKEDBM) - $(YPDBDIR)/$(DOM)/ethers.byaddr

@ (awk 'BEGIN { OFS="\t"; } $$1 !~ /^#/ { print $$2, $$0 }' \
$(DIR)/ethers $(CHKPIPE)) | \
$(MAKEDBM) - $(YPDBDIR)/$(DOM)/ethers.byname;
@touch ethers.time;
@echo "updated ethers";
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) ethers.byname; fi
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) ethers.byaddr; fi
@if [ ! $(NOPUSH) ]; then echo "pushed ethers"; fi

networks.time: $(DIR)/networks
@(sed -e "/^#/d" -e s/#.*$$// $(DIR)/networks $(CHKPIPE)) |( awk \
'{print $$1, $$0; for (i = 3; i <= NF; i++) print $$i, $$0}' \
$(CHKPIPE) )| $(MAKEDBM) - $(YPDBDIR)/$(DOM)/networks.byname;
@ (awk 'BEGIN { OFS="\t"; } $$1 !~ /^#/ { print $$2, $$0 }' \
$(DIR)/networks $(CHKPIPE)) | $(MAKEDBM) - $(YPDBDIR)/$(DOM)/networks.byaddr;
@touch networks.time;
@echo "updated networks";
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) networks.byname; fi
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) networks.byaddr; fi
@if [ ! $(NOPUSH) ]; then echo "pushed networks"; fi

services.time: $(DIR)/services
@ (awk 'BEGIN { OFS="\t"; } $$1 !~ /^#/ { print $$2, $$0 }' \
$(DIR)/services $(CHKPIPE)) | $(MAKEDBM) - $(YPDBDIR)/$(DOM)/services.byname;
@ (awk 'BEGIN { OFS="\t"; } \
$$1 !~ /^#/ { split($$2, pp, "/"); printf("%s/%s %s\n", $$1, pp[2], $$0); \
if (seen[$$1] == "") { \
printf("%s %s\n", $$1, $$0); seen[$$1]=$$1; } \
for (i = 3; i <= NF && $$i !~ /^#/; i++) \
printf("%s/%s %s\n", $$1, pp[2], $$0)}' \
$(DIR)/services $(CHKPIPE)) | \
$(MAKEDBM) $(B) - $(YPDBDIR)/$(DOM)/services.byservicename

@touch services.time;
@echo "updated services";
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) services.byname; fi
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) services.byservicename; fi
@if [ ! $(NOPUSH) ]; then echo "pushed services"; fi

rpc.time: $(DIR)/rpc
@ (awk 'BEGIN { OFS="\t"; } $$1 !~ /^#/ { print $$2, $$0 }' \
$(DIR)/rpc $(CHKPIPE)) | $(MAKEDBM) - $(YPDBDIR)/$(DOM)/rpc.bynumber;

```

(続く)

```

@touch rpc.time;
@echo "updated rpc";
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) rpc.bynumber; fi
@if [ ! $(NOPUSH) ]; then echo "pushed rpc"; fi

protocols.time: $(DIR)/protocols
@(awk 'BEGIN { OFS="\t"; } $$1 !~ /^#/ { print $$2, $$0 }' \
$(DIR)/protocols $(CHKPIPE)) | $(MAKEDBM) - \
$(YPDBDIR)/$(DOM)/protocols.bynumber;

@(sed -e "/^#/d" -e s/#.*$$// $(DIR)/protocols $(CHKPIPE)) |(awk \
'{print $$1,$$0; for (i = 3;i <= NF;i++) print $$i, $$0}' \
$(CHKPIPE))| $(MAKEDBM) - $(YPDBDIR)/$(DOM)/protocols.bynumber;

@touch protocols.time;
@echo "updated protocols";
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) protocols.bynumber; fi
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) protocols.bynumber; fi
@if [ ! $(NOPUSH) ]; then echo "pushed protocols"; fi

netgroup.time: $(DIR)/netgroup
@$(MAKEDBM) $(DIR)/netgroup $(YPDBDIR)/$(DOM)/netgroup
@$(REVNETGROUP) < $(DIR)/netgroup -u $(CHKPIPE)| $(MAKEDBM) - $(YPDBDIR)/$(DOM)/
netgroup.byuser
@$(REVNETGROUP) < $(DIR)/netgroup -h $(CHKPIPE)| $(MAKEDBM) - $(YPDBDIR)/$(DOM)/
netgroup.byhost
@touch netgroup.time;
@echo "updated netgroup";
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) netgroup; fi
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) netgroup.byuser; fi
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) netgroup.byhost; fi
@if [ ! $(NOPUSH) ]; then echo "pushed netgroup"; fi

bootparams.time: $(DIR)/bootparams
@(sed -e '/^#/d' -e s/#.*$$// -e 's/[ ][*]$$/' \
-e '/\$\$/s/\$\$/ /' $(DIR)/bootparams $(CHKPIPE))\
|(awk '/ $$/ {printf "%s", $$0} !/ $$/ {print}' $(CHKPIPE))\
|(sed -e 's/[ ][*]*/g' $(CHKPIPE))\
| $(MAKEDBM) - $(YPDBDIR)/$(DOM)/bootparams;
@touch bootparams.time;
@echo "updated bootparams";
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) bootparams; fi
@if [ ! $(NOPUSH) ]; then echo "pushed bootparams"; fi

aliases.time: $(ALIASES)
@cp $(ALIASES) $(YPDBDIR)/$(DOM)/mail.aliases;
@/usr/lib/sendmail -bi -oA$(YPDBDIR)/$(DOM)/mail.aliases;
$(MKALIAS) $(YPDBDIR)/$(DOM)/mail.aliases $(YPDBDIR)/$(DOM)/mail.byaddr;
@rm $(YPDBDIR)/$(DOM)/mail.aliases;
@touch aliases.time;
@echo "updated aliases";
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) mail.aliases; fi
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) mail.byaddr; fi
@if [ ! $(NOPUSH) ]; then echo "pushed aliases"; fi

```

(続く)

```

netmasks.time: $(DIR)/netmasks
$(MAKEDBM) $(DIR)/netmasks $(YPBDDIR)/$(DOM)/netmasks.byaddr;
@touch netmasks.time;
@echo "updated netmasks";
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) netmasks.byaddr; fi
@if [ ! $(NOPUSH) ]; then echo "pushed netmasks"; fi

publickey.time: $(DIR)/publickey
@(sed "/^#/d" < $(DIR)/publickey $(CHKPIPE)) | $(MAKEDBM) - $(YPBDDIR)/$(DOM)/
publickey.byname;
@touch publickey.time;
@echo "updated publickey";
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) publickey.byname; fi
@if [ ! $(NOPUSH) ]; then echo "pushed publickey"; fi

netid.time: $(PWDIR)/passwd $(DIR)/group $(DIR)/hosts $(DIR)/netid
$(MKNETID) -q -p $(PWDIR)/passwd -g $(DIR)/group -h $(DIR)/hosts -m $(DIR)/
netid > .ypjunk;
$(MAKEDBM) .ypjunk $(YPBDDIR)/$(DOM)/netid.byname;
@rm -f .ypjunk;
@touch netid.time;
@echo "updated netid";
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) netid.byname; fi
@if [ ! $(NOPUSH) ]; then echo "pushed netid"; fi

# Old way. Could be restored by PSARC decision.
#
#passwd.adjunct.time: $(PWDIR)/security/passwd.adjunct
#@ (awk 'BEGIN { FS=":"; OFS="\t"; } /^[a-zA-Z0-9_]/ { print $$1, $$0 }' $(PWDIR)/security/
passwd.adjunct $(CHKPIPE)) | \
# $(MAKEDBM) -s - $(YPBDDIR)/$(DOM)/passwd.adjunct.byname;
# @chmod 600 $(YPBDDIR)/$(DOM)/passwd.adjunct.byname.dir;
# @chmod 600 $(YPBDDIR)/$(DOM)/passwd.adjunct.byname.pag;
# @touch passwd.adjunct.time
# @echo "updated passwd.adjunct";
# @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) passwd.adjunct.byname; fi
# @if [ ! $(NOPUSH) ]; then echo "pushed passwd.adjunct"; fi

passwd.adjunct.time: $(PWDIR)/security/passwd.adjunct $(PWDIR)/shadow
-@if [ -f $(PWDIR)/shadow ]; then \
  (nawk 'BEGIN { FS=":"; while (getline < "$(PWDIR)/shadow" > 0) \
    shadow[$$1] = $$2; } /^[a-zA-Z0-9_]/ { $$2 = shadow[$$1]; OFS=":"; \
    printf "%s\t%s\n", $$1, $$0 }' $(PWDIR)/security/passwd.adjunct $(CHKPIPE)) | \
  $(MAKEDBM) -s - $(YPBDDIR)/$(DOM)/passwd.adjunct.byname; \
else \
  (awk 'BEGIN { FS=":"; OFS="\t"; } /^[a-zA-Z0-9_]/ { print $$1, $$0 }' \
  $(PWDIR)/security/passwd.adjunct $(CHKPIPE)) | \
  $(MAKEDBM) -s - $(YPBDDIR)/$(DOM)/passwd.adjunct.byname; \
fi
@chmod 600 $(YPBDDIR)/$(DOM)/passwd.adjunct.byname.dir;
@chmod 600 $(YPBDDIR)/$(DOM)/passwd.adjunct.byname.pag;
@touch passwd.adjunct.time

```

(続く)

```

@echo "updated passwd.adjunct";
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) passwd.adjunct.byname; fi
@if [ ! $(NOPUSH) ]; then echo "pushed passwd.adjunct"; fi

group.adjunct.time: $(PDIR)/security/group.adjunct
@(awk 'BEGIN { FS=":"; OFS="\t"; } /^[a-zA-Z0-9_]/ { print $$1, $$0 }' $(PDIR)/security/
group.adjunct $(CHKPIPE) | \
$(MAKEDBM) -s - $(YPDBDIR)/$(DOM)/group.adjunct.byname;
@chmod 600 $(YPDBDIR)/$(DOM)/group.adjunct.byname.dir;
@chmod 600 $(YPDBDIR)/$(DOM)/group.adjunct.byname.pag;
@touch group.adjunct.time
@echo "updated group.adjunct";
@if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) group.adjunct.byname; fi
@if [ ! $(NOPUSH) ]; then echo "pushed group.adjunct"; fi

timezone.time: $(DIR)/timezone
-@if [ -f $(DIR)/timezone ]; then \
sed -e "/^#/d" -e s/#.*$$// $(DIR)/timezone \
| awk '{for (i = 2; i<=NF; i++) print $$i, $$0}' \
| $(MAKEDBM) - $(YPDBDIR)/$(DOM)/timezone.byname; \
touch timezone.time; \
echo "updated timezone"; \
if [ ! $(NOPUSH) ]; then \
$(YPPUSH) timezone.byname; \
echo "pushed timezone"; \
else \
: ; \
fi \
else \
echo "couldn't find $(DIR)/timezone"; \
fi

auto.master.time: $(DIR)/auto_master
-@if [ -f $(DIR)/auto_master ]; then \
sed -e "/^#/d" -e s/#.*$$// $(DIR)/auto_master \
| $(MAKEDBM) - $(YPDBDIR)/$(DOM)/auto.master; \
touch auto.master.time; \
echo "updated auto.master"; \
if [ ! $(NOPUSH) ]; then \
$(YPPUSH) auto.master; \
echo "pushed auto.master"; \
else \
: ; \
fi \
else \
echo "couldn't find $(DIR)/auto_master"; \
fi

auto.home.time: $(DIR)/auto_home
-@if [ -f $(DIR)/auto_home ]; then \
sed -e "/^#/d" -e s/#.*$$// $(DIR)/auto_home \
| $(MAKEDBM) - $(YPDBDIR)/$(DOM)/auto.home; \
touch auto.home.time; \
echo "updated auto.home"; \

```

(続く)

```

if [ ! $(NOPUSH) ]; then \
  $(YPPUSH) auto.home; \
  echo "pushed auto.home"; \
else \
: ; \
fi \
else \
echo "couldn't find $(DIR)/auto_home"; \
fi

auth.attr.time: $(DIR)/auth_attr
-@if [ -f $(DIR)/auth_attr ]; then \
  sed -e "/^#/d" -e s/#.*$$// $(DIR)/auth_attr \
  |sed -e '/\$\$/{:l' -e 'N;s/\n//;t h' -e ':h' \
  -e 's/\$\$/\;/;t l' -e } \
  | (nawk 'BEGIN { FS=":"; OFS=":" } /^[a-zA-Z0-9_]/ \
  {printf "%s:%s\n", $$1, $$0 }' $(CHKPIPE)) \
  | $(MAKEDBM) -S ":" -E - $(YPDBDIR)/$(DOM)/auth_attr; \
  touch auth.attr.time; \
  echo "updated auth_attr"; \
  if [ ! $(NOPUSH) ]; then \
    $(YPPUSH) auth_attr; \
    echo "pushed auth_attr"; \
  else \
: ; \
fi \
else \
echo "couldn't find $(DIR)/auth_attr"; \
fi

exec.attr.time: $(DIR)/exec_attr
-@if [ -f $(DIR)/exec_attr ]; then \
  sed -e "/^#/d" -e s/#.*$$// $(DIR)/exec_attr \
  |sed -e '/\$\$/{:l' -e 'N;s/\n//;t h' -e ':h' \
  -e 's/\$\$/\;/;t l' -e } \
  | (nawk 'BEGIN { FS=":"; OFS=":" } /^[a-zA-Z0-9_]/ \
  {printf "%s:%s:%s\n", \
  $$1, $$6, $$0 }' $(CHKPIPE)) \
  | $(MAKEDBM) -S ":" -E -D 1 - $(YPDBDIR)/$(DOM)/exec_attr; \
  touch exec.attr.time; \
  echo "updated exec_attr"; \
  if [ ! $(NOPUSH) ]; then \
    $(YPPUSH) exec_attr; \
    echo "pushed exec_attr"; \
  else \
: ; \
fi \
else \
echo "couldn't find $(DIR)/exec_attr"; \
fi

prof.attr.time: $(DIR)/prof_attr
-@if [ -f $(DIR)/prof_attr ]; then \

```

(続く)

```

sed -e "/^#/d" -e s/#.*$$// $(DIR)/prof_attr \
|sed -e '/\$\$/{:l' -e 'N;s/\n//;t h' -e ':h' \
-e 's/\$\$/\;/;t l' -e } \
| (nawk 'BEGIN { FS=":"; OFS=":" } /^[a-zA-Z0-9_]/ \
{printf "%s:%s\n", $$1, $$0 }' $(CHKPIPE)) \
| $(MAKEDBM) -S ":" -E - $(YPDBDIR)/$(DOM)/prof_attr; \
touch prof.attr.time; \
echo "updated prof_attr"; \
if [ ! $(NOPUSH) ]; then \
$(YPPUSH) prof_attr; \
echo "pushed prof_attr"; \
else \
; ; \
fi \
else \
echo "couldn't find $(DIR)/prof_attr"; \
fi

user.attr.time: $(DIR)/user_attr
-@if [ -f $(DIR)/user_attr ]; then \
sed -e "/^#/d" -e s/#.*$$// $(DIR)/user_attr \
|sed -e '/\$\$/{:l' -e 'N;s/\n//;t h' -e ':h' \
-e 's/\$\$/\;/;t l' -e } \
| (nawk 'BEGIN { FS=":"; OFS=":" } /^[a-zA-Z0-9_]/ \
{printf "%s:%s\n", $$1, $$0 }' $(CHKPIPE)) \
| $(MAKEDBM) -S ":" -E - $(YPDBDIR)/$(DOM)/user_attr; \
touch user.attr.time; \
echo "updated user_attr"; \
if [ ! $(NOPUSH) ]; then \
$(YPPUSH) user_attr; \
echo "pushed user_attr"; \
else \
; ; \
fi \
else \
echo "couldn't find $(DIR)/user_attr"; \
fi

audit.user.time: $(DIR)/audit_user
-@if [ -f $(DIR)/audit_user ]; then \
sed -e "/^#/d" -e s/#.*$$// $(DIR)/audit_user \
|sed -e '/\$\$/{:l' -e 'N;s/\n//;t h' -e ':h' \
-e 's/\$\$/\;/;t l' -e } \
| (nawk 'BEGIN { FS=":"; OFS="\t" } /^[a-zA-Z0-9_]/ \
{print $$1, $$0 }' $(CHKPIPE)) \
| $(MAKEDBM) -s - $(YPDBDIR)/$(DOM)/audit_user; \
touch audit.user.time; \
echo "updated audit_user"; \
if [ ! $(NOPUSH) ]; then \
$(YPPUSH) audit_user; \
echo "pushed audit_user"; \
else \
; ; \
fi \

```

(続く)



```
else \  
  echo "couldn't find $(DIR)/audit_user"; \  
fi  
  
passwd: passwd.time  
group: group.time  
hosts: hosts.time  
ipnodes: ipnodes.time  
ethers: ethers.time  
networks: networks.time  
rpc: rpc.time  
services: services.time  
protocols: protocols.time  
netgroup: netgroup.time  
bootparams: bootparams.time  
aliases: aliases.time  
publickey: publickey.time  
netid: netid.time  
passwd.adjunct: passwd.adjunct.time  
group.adjunct: group.adjunct.time  
netmasks: netmasks.time  
timezone: timezone.time  
auto.master: auto.master.time  
auto.home: auto.home.time  
auth.attr:auth.attr.time  
exec.attr:exec.attr.time  
prof.attr:prof.attr.time  
user.attr:user.attr.time  
audit.user:audit.user.time  
$(DIR)/netid:  
$(DIR)/timezone:  
$(DIR)/auto_master:  
$(DIR)/auto_home:  
$(PWDIR)/shadow:  
$(DIR)/auth_attr:  
$(DIR)/exec_attr:  
$(DIR)/prof_attr:  
$(DIR)/user_attr:  
$(DIR)/audit_user:
```

Makefile は、all の下にリストされるデータベースのそれぞれに対して、適切な NIS マップを作成します。データは、makedbm で処理され、mapname.dir と mapname.pag の 2 つのファイルに保存されます。この両方はマスターサーバーの /var/yp/domainname ディレクトリに置かれます。

#### Makefile

は、/PWDIR/passwd、/PWDIR/shadow、/PWDIR/security/passwd.adjunct ファイルから passwd マップを適宜作成します。

## ▼ ypinit によるマスターサーバーの設定

/usr/sbin/ypinit シェルは、マスターサーバー、スレーブサーバー、クライアントが NIS を使用するよう設定します。また最初に make を実行して、マスターサーバー上にマップを作成します。

ypinit を使用して、新規に NIS マップをマスターサーバーに構築するには、次の手順に従います。

1. マスターサーバーの **root** になって、確実にネームサービスで **NIS** ではなく /etc ファイルから情報を得るようにします。次のように入力します。

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

2. /etc/hosts ファイルまたは /etc/inet/ipnodes ファイルを編集して、**NIS** サーバーのそれぞれの名前と **IP** アドレスを追加します。
3. 次のように入力して、新しいマップをマスターサーバーで構築します。

```
# /usr/sbin/ypinit -m
```

4. ypinit は、**NIS** スレーブサーバーになる他のマシンの入力を要求します。作業中のサーバー名と **NIS** スレーブサーバー名を入力します。
5. 致命的ではないエラーが発生したときすぐに手続きを終了するか、または致命的ではないエラーが発生しても継続するかを ypinit が問い合わせます。y を入力します。

y を選択すると、ypinit は最初の問題が発生すると終了します。問題を解決して、ypinit を再開できます。ypinit を最初に実行する場合には、この手順を実行するようにしてください。継続する場合には、発生する問題をすべて手作業で解決してから、ypinit を再度実行します。

---

注・マップファイルの一部が存在しない場合に、致命的ではないエラーが表示されることがあります。これは NIS の機能に影響するエラーではありません。自動的に作成されない場合には、マップを手作業で追加できます。デフォルトの NIS マップすべての説明は表 10-3 を参照してください。

---

6. `ypinit` が、`/var/yp/domainname` ディレクトリ内の既存のファイルを破棄してもよいかを問い合わせます。

このメッセージは、NIS が以前に設定されている場合にだけ表示されます。これに `yes` と回答して、NIS の新しいマップを設定します。

7. いったんサーバーのリストを構築すると、`ypinit` は `make` を起動します。

```
# make
```

このプログラムは、`/var/yp` に置かれる `Makefile` (デフォルトまたは修正されたもの) に含まれる命令を使用します。`make` コマンドは、指定したファイルにコメント行があれば、それを取り除き、それに対して `makedbm` を実行します。適切なマップを作成して、各マップにマスターサーバー名を設定します。

マスターサーバー上で `domainname` コマンドを実行した時に返されるドメイン名以外のドメイン用のマップの転送が `Makefile` で行われる場合、次のように `ypinit` シェルスクリプト中で起動される `make` コマンドに適切なドメイン名を変数 `DOM` で指定することで、確実にマップが正しいドメインに転送されるようにすることができます。

```
# make DOM=domainname password
```

これは `password` マップを、マスターサーバーが属するドメインではなく目的のドメインに転送します。

8. 次のように入力して、ネームサービスとして **NIS** を使用します。

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

これは、現在のスイッチファイルをデフォルトの NIS 用スイッチファイルに置き換えます。このファイルは必要に応じて編集可能です。

## 複数の NIS ドメインをサポートするマスターサーバー

通常 NIS マスターサーバーは、NIS ドメインだけをサポートします。しかしマスターサーバーを使用して、複数のドメインをサポートする場合には、追加ドメイン用にサーバーを設定する際に、前述の節で説明した手順を若干修正する必要があります。

サーバー上で `domainname` コマンドを実行します。コマンドが戻すドメイン名は、サーバーのデフォルトドメインです。前述の節で説明した手順は、そのドメインへのサービスの設定では正しく機能します。他のドメインに対してサービスを構成するには、`ypinit` シェルスクリプトを次のように修正する必要があります。

```
# make DOM=correct-domain passwd
```

`correct-domain` は、サービスを設定している他のドメインの名前であり、`passwd` は `make` ターゲットです。このコマンドは `password` マップを、マスターサーバーが属するドメインではなく目的のドメインに転送します。

---

## マスターサーバー上での NIS サービスの開始

マスターマップが作成されると、NIS デーモンをマスターサーバーで起動して、サービスを開始できます。これを行うには、`ypserv` をサーバー上で起動して、`ypbind` を実行する必要があります。クライアントがサーバーに情報を要求すると、`ypserv` デーモンは NIS マップ内で検索し、クライアントからの情報の要求に対応します。

サーバー上で NIS サービスを起動するには 2 つの方法があります。

- ブートプロセス中に `/usr/lib/netsvc/yp/ypstart` スクリプトを自動的に起動
- コマンド行から `ypstart` を実行

### NIS サービスの自動スタート

`ypinit` の実行による NIS マスターサーバーの構成後、マシンのブート時に `ypstart` が自動的に起動され、`ypserv` を起動します (210ページの「`ypinit` によるマスターサーバーの設定」を参照)。

### コマンド行からの NIS の開始

NIS サービスをコマンド行から開始するには、以下を実行します。

```
# /usr/lib/netsvc/yp/ypstart
```

---

注 - 起動後に `ypserv` が呼び出しに応答できるようになるまでに若干の遅延があります。プログラムまたはスクリプトの内部から呼び出す場合には、`ypstart` の実行後に 3 秒から 5 秒間スリープ状態にしてください。

---

## DNS の転送

Solaris 8 リリースでは、`/etc/resolv.conf` ファイルがある場合には、`ypstart` は DNS の転送を使用するように `ypserv` を起動します。DNS の転送を使用しない場合には、`/usr/lib/netsvc/yp/ypstart` スクリプトを編集して、`-d` オプションを `ypserv` コマンドから削除します。その後マシンをリブートする必要があります。

DNS での NIS の使用については、第 13 章と『Solaris ネーミングの管理』を参照してください。

## ypstop による NIS の停止

NIS を停止するには、`ypstop` コマンドを実行します。

```
# /usr/lib/netsvc/yp/ypstop
```

---

## NIS スレーブサーバーの設定

ネットワークは 1 つ以上のスレーブサーバーを持つことができます。スレーブサーバーを持つことで、マスターサーバーが利用できない場合にも NIS サービスを継続して確保できます。

## スレーブサーバーの準備

ypinit を実際に実行して、スレーブサーバーを作成する前に、domainname コマンドを NIS スレーブごとに実行して、ドメイン名がマスターサーバーと一致していることを確認します。

---

注・ドメイン名は大文字と小文字を区別します。

---

ネットワークが適切に機能していることを確認してから、NIS スレーブサーバーを構成してください。特に、rcp を使用して NIS のマスターサーバーから NIS スレーブサーバーにファイルを送れるかを確認してください。

## NIS スレーブサーバーの設定 — タスクマップ

表 10-2 NISスレーブサーバーの設定

タスク	説明	指示の参照先
NIS スレーブサーバー	NIS スレーブサーバー	214ページの「スレーブサーバーの設定」

## スレーブサーバーの設定

マスターサーバー上での NIS サービスの開始後、次のように新しいスレーブサーバーを作成できます。

1. **root** として、スレーブサーバー上で /etc/hosts ファイルまたは /etc/inet/ipnodes ファイルを編集して、他のすべての **NIS** サーバー名と IP アドレスを追加します。
2. スレーブサーバー上の /var/yp にディレクトリを変更します。
3. 次のように入力して、スレーブサーバーにするマシンをクライアントとして初期化します。

```
# /usr/sbin/ypinit -c
```

ypinit コマンドが、NIS サーバーのリストの入力を促します。作業中のローカルマシン (スレーブ) の名前を最初に入力してからマスターサーバーを入力し、その後にドメイン内の他の NIS スレーブサーバーをネットワーク的に近いものから遠いものの順番で入力します。

注 - スレーブサーバーにするマシンを最初に NIS クライアントとして構成して、マスターサーバーから最初に NIS マップを得ることができるようになります (詳細は、216ページの「NIS クライアントの設定」を参照)。

4. ypbind が実行中かどうかを判断するには、次のように入力します。

```
# ps -ef | grep ypbind
```

リストが表示されたら、ypbind は実行中です。

5. ypbind が実行中なら、次のように入力して停止します。

```
# /usr/lib/netshvc/yp/ypstop
```

6. 次のように入力して、ypbind を再開します。

```
# /usr/lib/netshvc/yp/ypstart
```

7. 次のように入力して、このマシンをスレーブとして初期化します。

```
# /usr/sbin/ypinit -s master
```

この場合 *master* は、既存の NIS マスターサーバーのマシン名です。

この節で説明した手順を、NIS スレーブサーバーとして構成するマシンごとに繰り返します。

## スレーブサーバー上での NIS サービスの開始

これでスレーブサーバーでデーモンを開始して、NIS サービスを開始できます。次のとおりに入力して、既存のすべての yp プロセスを停止する必要があります。

```
# /usr/lib/netstvc/yp/ypstop
```

スレーブサーバーで ypserv を開始して ypbind を実行するには、次のとおりに入力します。

```
# /usr/lib/netstvc/yp/ypstart
```

これとは別に、スレーブサーバーをリブートして、デーモンを自動的に開始することもできます。

---

## NIS クライアントの設定

次の 2 つの手続きを実行して、NIS をマシンで使用できるようにします。

- 第 1 章で説明したように、nsswitch.conf ファイルを正しく設定する
- NIS を使用するマシンを次に説明するとおりに構成する

## NIS を使用するマシンの構成

以下にネームサービスとして NIS を使用するマシンを構成する方法を 2 通り説明します。

- 「ypinit」

NIS を使用するクライアントマシンの構成に推奨されているのは、マシンに root としてログインして、ypinit -c を実行する方法です。

```
# ypinit -c
```

NIS サーバー (クライアントはそこからネームサービス情報を得る) を指定するように要求されます。必要なだけマスターサーバーやスレーブサーバーを指定できます。



指定するサーバーはドメイン内のどこにあってもかまいません。クライアントにネットワーク的に近いサーバーから遠いサーバーの順に指定することをお勧めします。

#### ■ 「ブロードキャスト方式」

NIS のクライアントマシンを構成する古い方式です。マシンに `root` としてログインし、`domainname` コマンドでドメイン名を設定してから、`ypbind` を実行します。

```
# domainname doc.com
# ypbind -broadcast
```

`ypbind` を実行すると、NIS サーバーをローカルサブネットで探して、見つければ接続します。この検索を「ブロードキャスト」と呼びます。クライアントのローカルサブネットに NIS サーバーがない場合には接続されず、クライアントマシンは NIS サービスから名前空間データを得ることができません。

---

## NIS マップ

NIS が使用する名前空間データは、NIS マップ群に格納されます。本質的に NIS マップは、2 つの列で構成される表です。

### デフォルトの NIS マップ

表 10-3 は、デフォルトの NIS マップ、それに含まれる情報、NIS の実行中に対応する管理ファイルをオペレーティングシステムが参照するかどうかを説明します。

表 10-3 NIS マップの説明

マップ名	対応する NIS 管理ファイル	説明
bootparams	bootparams	クライアントがブート中に必要とするファイルのパス名を含む。root、swap など
ethers.byaddr	ethers	マシン名と Ethernet アドレスを含む。Ethernet アドレスはマップ内のキー
ethers.byname	ethers	キーが Ethernet アドレスではなくマシン名であることを除けば ethers.byaddr と同じ
group.bygid	group	グループ ID をキーとしてグループセキュリティ情報を含む
group.byname	group	グループ名をキーとしてグループセキュリティ情報を含む
hosts.byaddr	hosts	IPv4 アドレスをキーとしてマシン名と IPv4 アドレスを含む
hosts.byname	hosts	マシン (ホスト) 名をキーとしてマシン名と IPv4 アドレスを含む
ipnodes.byaddr	ipnodes	IP アドレスをキーとしてマシン名と IP アドレス (IPv4 および IPv6 アドレスの両方) を含む
ipnodes.byname	ipnodes	マシン (ホスト) 名をキーとしてマシン名と IP アドレス (IPv4 および IPv6 アドレスの両方) を含む
mail.aliases	aliases	エイリアスをキーとしてエイリアスとメールアドレスを含む
mail.byaddr	aliases	メールアドレスをキーとしてメールアドレスとエイリアスを含む
netgroup.byhost	netgroup	マシン (ホスト名) をキーとして、グループ名、ユーザー名、マシン名を含む
netgroup.byuser	netgroup	キーがユーザー名であることを除くと netgroup.byhost と同じ

表 10-3 NIS マップの説明 続く

マップ名	対応する NIS 管理ファイル	説明
netgroup	netgroup	キーがグループ名であることを除くと netgroup.byhost と同じ
netid.byname	passwd, hosts, group	UNIX 形式の認証に使用。ネット名のデータベースを含む。netid ファイルがある場合には、他のファイルを使用して利用できるデータの他にそれが参照される
netmasks.byaddr	netmasks	アドレスをキーとして、IP のサブネット化で使用されるマスクを含む
networks.byaddr	networks	アドレスをキーとして、システムに対して既知のネットワーク名とアドレスを含む
networks.byname	networks	キーがネットワーク名であることを除くと、networks.byaddr と同じ
passwd.adjunct.byname	passwd と shadow	C2 クライアント用の監査情報と隠蔽されたパスワード情報を含む
passwd.byname	passwd と shadow	ユーザー名をキーとして、パスワード情報を含む
passwd.byuid	passwd と shadow	キーがユーザー ID であることを除くと passwd.byname と同じ
protocols.byname	protocols	プロトコル名をキーとして、システムに対して既知のネットワークプロトコルを含む
protocols.bynumber	protocols	キーがプロトコル番号であることを除くと、protocols.byname と同じ
rpc.bynumber	rpc	システムに対して既知のプログラム番号と RPC の名前を含む。キーは RPC のプログラム番号

表 10-3 NIS マップの説明 続く

マップ名	対応する NIS 管理ファイル	説明
services.byname	services	ネットワークに対して既知のインターネットサービスをリストする。キーはポートとプロトコル
services.byservice	services	ネットワークに対して既知のインターネットサービスをリスト
ypservers	なし	ネットワークに対して既知の NIS サーバーをリスト

## NIS マップの修正

作成後の NIS マップの修正については、『Solaris ネーミングの管理』を参照してください。

## NIS の管理、問題解決、エラーメッセージ

NIS の管理、問題解決、エラーメッセージの詳細は、『Solaris ネーミングの管理』を参照してください。

## パート **IV**    **FNS** の設定と構成

---

パート IV では、FNS (Federated Naming Service) を NIS+、NIS、/etc の名前空間環境で初期設定する方法と構成する方法を説明します。

- 第 11 章



## FNS の設定と構成

---

この章では、NIS+、NIS、ファイルのネームサービス環境での FNS (Federated Naming Service) の初期設定と構成の方法を説明します。「ファイル」のネームサービスとは、NIS+ や NIS ではなく、/etc ファイルからデータを得るネームサービスのことです。

- 224ページの「FNS の設定 - 概要」
- 224ページの「リソース条件の決定」
- 226ページの「FNS 用の名前空間の準備」
- 229ページの「グローバルな FNS の名前空間コンテキストの作成」
- 233ページの「FNS サービスの複製」
- 236ページの「FNS の管理、問題解決、エラーメッセージ」

FNS の一般的な説明や概要については、『Solaris ネーミングの管理』を参照してください。

---

注 - 『Solaris ネーミングの管理』には、FNS の要約、設定と構成の手順の簡単な説明、プログラミング例を提供する「FNS の手引き」の章があります。システム管理者は、この章を参照してください。

---

---

## FNS の設定 - 概要

Solaris 8 リリースソフトウェアのインストール後に次の作業を実行して、FNS を設定する必要があります。

1. サーバーが **FNS** を取り扱えることを確認します。224ページの「リソース条件の決定」を参照してください。
2. **FNS** 用の名前空間を準備します。226ページの「FNS 用の名前空間の準備」を参照してください。
3. **FNS** 名前空間のコンテキストを設定します。次の **2** つの方法があります。
  - a. **1** つのプロセスで、すべてのコンテキストをグローバルに作成します。229ページの「グローバルな FNS の名前空間コンテキストの作成」を参照してください。
  - b. **FNS** コンテキストを個別に作成します。『**Solaris** ネーミングの管理』を参照してください。
4. **FNS** の複製サーバーを設定します。233ページの「FNS サービスの複製」を参照してください。

組織の大きさによっては、FNS の設定が完了するまでに数時間と、名前空間の準備にもある程度の時間が必要になります。

---

## リソース条件の決定

インストールの手順に進む前に、FNS をサポートしているサーバーに十分なメモリーとディスク領域があることを最初に確認する必要があります。企業レベルのネームサービス (NIS+、NIS、ファイル) で必要な領域の他に FNS 用の領域が必要です。

一般的に確実な方法として、ユーザーとホストごとに、約 17K バイトのディスク領域とスワップ領域が必要になります。このディスク領域が配置される場所と、その計算方法は、使用している企業レベルのネームサービスによって異なります。



- 「NIS+」

ドメインまたサブドメインに対して FNS サーバーとして機能するマシンに、ディスク領域が必要です。NIS+ 環境では、FNS の `ctx_dir` ディレクトリを提供するサーバーは、`org_dir` など標準の NIS+ ディレクトリを提供するのと同じサーバーである必要はありません。サーバーの負荷を均等にするために、大規模な構成で使用しているサイトの多くでは、NIS+ と FNS のサーバーに別個のマシンを使用しています。NIS+ 環境で FNS サーバーに必要な領域は、サーバーがネームサービスを提供するドメインまたはサブドメイン内のユーザーとホストの数で決まります。

- 「NIS」

ドメインに対して FNS サーバーとして機能するマシンにディスク領域が必要です。NIS 環境では、FNS をホストするサーバーは NIS をホストするのと同じサーバーである必要はありません。サーバーの負荷を均等にするために、大規模な構成で使用しているサイトの多くでは、NIS と FNS のサーバーに別個のマシンが使用されています。NIS 環境で FNS サーバーに必要な領域は、ドメイン内のユーザーとホストの数で決まります。

- 「ファイル」

企業レベルのネームサービスがファイルのときは、FNS に必要なディスク領域は、`/var/fn` をマウントするマシンの `/etc/users` と `/etc/hosts` のファイル内のユーザーとホストの数で決まります。マシンごとに `/var/fn` ディレクトリがある場合には、必要な領域は各マシンのユーザーとホストのファイルで決まります。`/var/fn` がマシンにマウントされ、FNS によってネットワーク上の残りのマシンにエクスポートされる場合には、`/var/fn` を提供するマシンに必要な領域は、マシンの `/etc/users` と `/etc/hosts` のファイル内のユーザーとホストの数で決まります。

たとえば、1,200 のユーザーとホストを持つ NIS+ ドメインで FNS 環境をサポートするには、以下が必要になります。

- 基礎となる名前空間に必要な領域 (NIS+、NIS、ファイル) の他に、少なくとも 20M バイトのディスク領域
- FNS 用に 40M バイト のスワップ領域

## FNS 用の名前空間の準備

この節では、`fncreate` を実行して FNS コンテキストを設定する前の準備について説明します。準備は該当する企業レベルのネームサービスによって異なります。以下を参照してください。

### FNS 用の名前空間の準備 — タスクマップ

表 11-1 FNS 用の名前空間の準備

タスク	説明	指示の参照先
FNS 用の名前空間の準備	NIS マップへのファイルの変換	198ページの「NIS マップへの変換用ソースファイルの準備」
FNS 用の名前空間の準備	NIS サービスの準備	228ページの「FNS 用の NIS サービスの準備」
FNS 用の名前空間の準備	ファイルベースのネーミングの準備	229ページの「FNS 用のファイルを使用したネームサービスの準備」

### ▼ FNS 用の NIS+ サービスの準備

FNS 名前空間を設定する前に以下の作業を行います。

1. **NIS+** ドメインが正確に設定されていることを確認します。  
NIS+ドメインと関連のサブドメインが、FNS の構成前に設定済みである必要があります。つまり `hosts` と `passwd` など NIS+ の標準のテーブルが既に存在し、サブコンテキストが生成されている必要があります。
2. ドメインの `hosts.org_dir` と `passwd.org_dir` のテーブルが、すべてのホスト名とユーザー名を持つサブコンテキストが生成されていることを確認します。

niscat と nismatch のコマンドを使用して、これらテーブルの内容を確認できます。

3. NIS\_GROUP 環境変数を、**FNS** オブジェクトを管理するグループの名前に設定します。

この変数を最初に設定しないと、fncreate コマンドで FNS 設定を完了できません。fncreate コマンドがユーザーとホストのコンテキストを作成するとき、それらはコマンドを実行したシステム管理者ではなく、そのホストとユーザーの所有となります。NIS\_GROUP の設定によって、グループのメンバーになっているシステム管理者は、オブジェクトを所有していなくてもコンテキストを修正できます。

Cシェルの場合、次のように NIS\_GROUP に fns\_admins.doc.com を設定します。

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

4. [必要に応じて] **NIS+** マスターサーバー以外のマシンで **FNS** を実行するように指定します。

FNS が使用する、すべての NIS+ オブジェクトは、NIS+ ドメインの ctx\_dir ディレクトリの下に保管されます。5,000 以上のユーザーやホストを有する大規模なドメインでは、FNS が使用する ctx\_dir が、groups\_dir のような標準の NIS+ ディレクトリをサポートするサーバーとは別のサーバーによってサポートされるようにしてください。これは必須ではありませんが、推奨されています。別個のサーバーを使用することで、1つのサーバーに過剰な負荷がかからなくなります。またこれによって、FNS による NIS+ の使用の管理と NIS+ 自体の管理を分離できます。

ドメインに対する NIS+ マスターサーバーではないマシンによって FNS が提供されるように指定するには、ドメインに対する FNS ホストとして機能するマシン上に ctx\_dir ディレクトリオブジェクトを手作業で作成する必要があります。この手順を省略すると、FNS はドメインの NIS+ ルートマスターサーバーにインストールされます。

FNS のマスターサーバーになるマシンを指定するには次のようにします。

- a. **NIS+** ドメインに ctx\_dir ディレクトリを作成します。

たとえば、doc.com ドメイン内で fns\_server と名前のついたマシン上に ctx\_dir 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で次の

コマンドを実行します。ドメイン名の最後にドットが付いていることに注意してください。

```
nismaster# nismkdir -m fns_server ctx_dir.doc.com.
```

nismkdir コマンドによる NIS+ ディレクトリの作成については、『Solaris ネーミングの管理』を参照してください。

---

注 - 「サブドメイン」用の FNS の ctx\_dir を作成する場合、ctx\_dir を提供する FNS サーバーとして指定するマシンはサブドメイン内に存在する必要があります。親ドメイン内のマシンは指定できません。これとは対照的に、サブドメインの NIS+ のマスターサーバーは常に、それが機能する 1 つ上のドメインに存在します。つまり、NIS+ のサブドメインに対して FNS を構成しているときに、NIS+ と FNS の両方で同じサーバーを使用する場合には、そのサーバーはサブドメインの上のドメインに存在します。しかし、NIS+ と FNS で異なるサーバーを使用する場合には、NIS+ のマスターサーバーはその上のドメインに存在し、FNS サーバーはそれが機能するサブドメインに存在します。

---

- b. nisls コマンドを使用して、ctx\_dir ディレクトリが作成されたことを検証します。

```
rootmaster# nisls doc.com.ctx_dir
```

- c. nisping を実行して、ディレクトリをチェックポイントします。

```
# /usr/lib/nis/nisping -C ctx_dir.doc.com.
```

## ▼ FNS 用の NIS サービスの準備

FNS 名前空間を設定する前に、以下の作業を行います。

- ◆ hosts.byname、user.byname、printer.conf.byname のマップが完全、正確、最新であることを確認します。

---

注 - 他の任意の NIS マップに対して異なるマスターサーバーを割り当てるのと同じ手順で、FNS マップに異なるマスターサーバーを割り当てることができます。詳細は、『Solaris ネーミングの管理』を参照してください。

---

## FNS 用のファイルを使用したネームサービスの準備

ファイルを使用したネームサービスとは、NIS+ または NIS ではなく、/etc ファイルからデータを得るネームサービスのことです。

/var/fn ディレクトリをマシンごとにインストールする場合には、一般的に次の手順をマシンごとに実行する必要があります。1つのマシンから /var/fn ディレクトリのマウントとエクスポートをする場合には、次の手順を /var/fn をエクスポートするマシンで実行する必要があります。

- ◆ /etc/hosts と /etc/passwd のファイルが完全で、すべてのユーザー名とホスト名を含むことを確認してください。

---

## グローバルな FNS の名前空間コンテキストの作成

この節では、企業または NIS+ ドメインに対する名前空間をグローバルに作成する方法を説明します。

FNS の名前空間は、fncreate コマンドを使用して作成されます。

```
# fncreate -t org org//
```

もしくは次のコマンドを使用します。

```
# fncreate -t org org/domain/
```

この場合 *domain* 名は、NIS+ドメイン名またはサブドメイン名です。

fncreate コマンドは、指定された編成と、そのすべてのサブコンテキスト用のデフォルトのコンテキストを作成します。これには編成内のユーザーとホストに対するコンテキストとサブコンテキストが含まれます。

個々の FNS コンテキストを手作業で作成する方法は、『Solaris ネーミングの管理』を参照してください。

## グローバルな FNS の名前空間コンテキストの作成 — タスクマップ

表 11-2 グローバルな FNS の名前空間コンテキストの作成

タスク	説明	指示の参照先
グローバルな FNS の名前空間の作成	NIS+ の下での FNS の名前空間コンテキストの作成	230ページの「NIS+ の下での名前空間コンテキストの作成」
グローバルな FNS の名前空間コンテキストの作成	NIS の下での FNS の名前空間の作成	231ページの「NIS の下での名前空間コンテキストの作成」
グローバルな FNS の名前空間コンテキストの作成	ファイルの下での FNS の名前空間の作成	232ページの「ローカルファイルの下での名前空間コンテキストの作成」

### ▼ NIS+ の下での名前空間コンテキストの作成

企業レベルの主なネームサービスが NIS+ であるときは、NIS+ ドメインまたはサブドメインごとに名前空間のコンテキストを個別に作成する必要があります。

- NIS+ ドメインまたはサブドメインが既に存在している必要があります。
- NIS+ と FNS の両方で同じサーバーを使用する場合には、ドメイン (サブドメイン) のマスターサーバーで `fncreate` コマンドを実行する必要があります。NIS+ と FNS で異なるサーバーを使用する場合には、FNS サーバーとして機能するマシンで `fncreate` コマンドを実行する必要があります。異なるマシンを使用する場合には、前記の「FNS 用の NIS+ サービスの準備」の227ページの手順 4 に従って FNS サーバーを最初に準備する必要があります。

- NIS+ 管理の完全な認証が必要です。NIS+ の認証については、『Solaris ネーミングの管理』を参照してください。

たとえば、そのドメイン用の NIS+ マスターサーバーである `submaster` マシン上の `manf.doc.com` サブドメイン用にコンテキストを作成するには、次のようにします。

- ◆ サブドメインのマスターで、`fncreate` を実行します。

```
submaster# fncreate -t org org/manf.doc.com./
```

これで、NIS+ `manf.doc.com` サブドメイン用の編成コンテキスト、サブドメインの `passwd.org_dir` テーブルのすべてのユーザーとサブドメインの `hosts.org_dir` テーブルのすべてのホストに対するコンテキストとサブコンテキストが作成されます。

NIS+ と FNS のサーバーに異なるマシンを使用するには、FNS サーバーとして使用するマシン上で前述のコマンドを実行します。NIS+ ではないサーバーを FNS サーバーとして設定する方法についての説明は、前記の「FNS 用の NIS+ サービスの準備」の227ページの手順4を参照してください。

- ◆ `nisping` コマンドを使用して `ctx_dir` ディレクトリをチェックポイントにします。

```
# /usr/lib/nis/nisping -C ctx_dir.manf.doc.com.
```

---

注 - 数千のユーザーやホストを有する大規模な組織では、最初の `fncreate` 操作に数時間、それ以降のチェックポイントにも数時間かかる場合があります。

---

## ▼ NIS の下での名前空間コンテキストの作成

企業レベルの主なネームサービスが NIS であるときは、企業に対して1つのドメインだけが存在します。その企業全体のドメインに対して名前空間コンテキストが作成されます。

- NIS ドメインが既に存在している必要があります。

- FNS マスターサーバー上で root によって `fncreate` コマンドが実行される必要があります。通常、これは NIS マスターサーバーになりますが、別のサーバーを選択することもできます。

たとえば、NIS マスターサーバーでもある `fns_master` というマシンで `doc.com` ドメイン用にコンテキストを作成するには、次のようにします。

- ◆ ドメインマスターで `fncreate` を次のように実行します。

```
fns_master# fncreate -t org org//
```

これで NIS ドメイン `doc.com` 用の編成コンテキストと、NIS サーバーの `passwd` マップのすべてのユーザーとサーバーの `hosts` マップの、すべてのホストに対するコンテキストと関連のサブコンテキストが作成されます。

---

注 - コンテキストマップを作成したら、他の任意の NIS マップに異なるマスターを割り当てるのと同じ手順で、同じマシンをマスターサーバーに割り当てることができます。FNS マップはすべて、`.ctx` または `.attr` のいずれかで終わる名前を持ちます。詳細は、『Solaris ネーミングの管理』を参照してください。

---

## ▼ ローカルファイルの下での名前空間コンテキストの作成

企業レベルの主なネームサービスがファイルであるときは、コンテキストはシステムに対して作成されます。

- `/var/fn` ディレクトリが存在するマシン上の `/etc/passwd` と `/etc/hosts` のファイルは誤りがなく完全に生成されている必要があります。
- `fncreate` コマンドは、`/var/fn` ディレクトリが存在するマシン上で root によって実行される必要があります。

たとえば、システム用のコンテキストを作成するには次のようにします。

- ◆ `/var/fn` ディレクトリのあるマシンで、`fncreate` を次のように実行します

```
server1# fncreate -t org org//
```



これで、マシン内の `/etc/passwd` ファイルのすべてのユーザーとマシン内の `/etc/hosts` ファイルのすべてのホスト用にシステム、コンテキスト、関連のサブコンテキストに対する編成コンテキストが作成されます。

## FNS サービスの複製

FNS のネームサービスの性能と信頼性が重要な大規模で重要性の高いネットワークでは、FNS サービスを複製してください。

### FNS サービスの複製 — タスクマップ

表 11-3 FNS サービスの複製

タスク	説明	指示の参照先
FNS サービスの複製	NIS+ の下での FNS サービスの複製	233ページの「NIS+ の下での FNS の複製」
FNS サービスの複製	NIS の下での FNS の複製	234ページの「NIS の下での FNS の複製」
FNS サービスの複製	ファイルの下での FNS の複製	235ページの「ファイルを使用したネームサービスの下での FNS の複製」

#### ▼ NIS+ の下での FNS の複製

マスターサーバーで FNS 名前空間が設定されたら、その他の複製サーバーを各ドメインに追加して、ドメインの `ctx_dir` を提供するサーバーにします。複製サーバーによって、サーバーの可用性と性能を拡張できます。

1. **FNS** のマスターサーバーで `nismkdir` コマンドを実行して、`ctx_dir` ディレクトリ用の複製を追加します。

たとえば、マシン `fnsrserver` を `doc.com`. ドメインの FNS の複製サーバーにします。

```
# nismkdir -s fnsrserver ctx_dir.doc.com.
```

2. `nisping` コマンドで `ctx_dir` ディレクトリをチェックポイントします。

```
# /usr/lib/nis/nisping -C ctx_dir.doc.com.
```

FNS の複製は一定の間隔でチェックポイントしてください。間隔は、数日に一回程度をお勧めします。選択する間隔は、FNS 名前空間への変更頻度によって異なります。

## ▼ NIS の下での FNS の複製

FNS 名前空間がドメインのマスターサーバーで設定された後に、スレーブサーバーを追加して、サーバーの可用性と機能を拡張できます。

1. `root` として、スレーブサーバーの `/etc/hosts` ファイルを編集して、他のすべての **NIS** サーバーの名前と **IP** アドレスを追加します。
2. スレーブサーバー上で、`/var/yp` に現在のディレクトリを変更します。
3. クライアントとしてスレーブサーバーを初期化するには、以下を入力します。

```
# /usr/sbin/ypinit -c
```

`ypinit` コマンドが、NIS サーバーのリストの入力を促します。作業中のローカルスレーブの名前を最初に入力してからマスターサーバーを入力し、その後にドメイン内の他の NIS スレーブサーバーをネットワーク的に近いものから遠いものの順番で入力します。

---

注 - NIS クライアントとして、新しいスレーブサーバーを最初に構成して、マスターサーバーから最初に NIS マップを得ることができるようにします。詳細は、216ページの「NIS クライアントの設定」を参照してください。

---

4. `ypbind` が実行中かどうかを判断するには、次のとおりに入力します。

```
# ps -ef | grep ypbind
```

リストが表示されたら、`ypbind` は実行中です。

5. `ypbind` が実行中なら、次のとおりに入力して停止させます。

```
# /usr/lib/netshvc/yp/ypstop
```

6. 以下を入力して `ypbind` を再起動します。

```
# /usr/lib/netshvc/yp/ypstart
```

7. このマシンをスレーブとして初期化するには、以下を入力します。

```
# /usr/sbin/ypinit -s master
```

この場合、`master` は既存の NIS マスターサーバーのマシン名です。

8. スレーブサーバーの `yp` プロセスを停止します。

```
# /usr/lib/netshvc/yp/ypstop
```

9. `yp` サービスを再起動します。

```
# /usr/lib/netshvc/yp/ypstart
```

これとは別に、スレーブサーバーをリブートしてデーモンを自動的に開始することもできます。

## ▼ ファイルを使用したネームサービスの下での FNS の複製

主なネームサービスがファイルになっているときはサーバーの複製はありません。

---

## **FNS** の管理、問題解決、エラーメッセージ

FNS の管理、問題解決、エラーメッセージについては、『Solaris ネーミングの管理』を参照してください。

## パート V DNS の設定と構成

---

パート V では、DNS (Domain Name System) の概要と、DNS クライアントおよび DNS サーバーの設定方法について説明します。次の 2 つの章があります。

- 第 12 章
- 第 13 章



## DNS クライアントの設定

---

この章では、クライアントマシン上での DNS (Domain Name System) サービスの設定方法について説明します。

- 240ページの「Solaris DNS BIND の実装」
- 240ページの「DNS サービスの設定」
- 241ページの「クライアントの設定」
- 241ページの「リゾルバ」
- 242ページの「`resolv.conf` ファイルの作成」
- 243ページの「`/etc/nsswitch.conf` ファイルの修正」

---

注 - DNS の最も一般的かつ重要な役割は、ローカルなネットワークをグローバルなインターネットに接続することです。そして、インターネットに接続するためには、親ドメインの管理者にネットワークの IP アドレスを登録してもらう必要があります。管理者は、ネットワークの地理的な位置と、親ドメインの種類によって異なります。ドメイン管理者にネットワークを登録してもらう方法については、本書では説明を割愛しています。

---

詳細は、『*DNS and Bind*』(Cricket Liu & Paul Albitz 著、浅羽登志也／上水流由香監訳、アスキー出版局、1995) を参照してください。

---

## Solaris DNS BIND の実装

Solaris 8 リリースには、ユーザーの便宜を図るため、コンパイル済みの BIND (Barkeley Internet Name Domain) 8.1.2 が付属しています。コンパイルにあたっては、最大公約数のサイトのニーズを満たすべく、各種オプションを設定しました。このコンパイル済みの BIND が要件に合わない場合は、公開されているソースコードから独自にコンパイルすることができます。

Solaris 8 リリースに付属の BIND は、次のオプションでコンパイルされています。

- 「RFC1535」  
無効。暗黙的検索リストがサポートされなくなるのを避けるため
- 「逆クエリー」  
有効。SunOS 4.x の正常動作には不可欠
- 「偽名ロギング」  
無効。無用のメッセージが次々に出される事態を避けるため
- 「デフォルトドメイン名」  
DNS ドメイン名が `/etc/resolv.conf` に設定されていない場合、あるいは `LOCALDOMAIN` 環境変数で設定されていない場合、`libresolv` は NIS または NIS+ ドメイン名から DNS ドメイン名を取得する。`/etc/nsswitch.conf` ファイルの `hosts` 行の先頭に `nisplus` または `nis` が指定されていることを前提とする
- 「ユーティリティスクリプト」  
実装しない
- 「テストプログラム」  
`dig`、`dnsquery`、`host` の各テストプログラムは、`nslookup` および `nstest` と共通する部分があるため、いずれも実装しない

---

## DNS サービスの設定

DNS サービスを設定するには、次の 2 つの手順を実行します。



1. クライアントマシン上で **DNS** サービスを設定します (詳細は、この章で説明)。
2. 第 13 章の説明に従って、**DNS** サーバーを設定します。

---

## クライアントの設定

クライアントマシン上の DNS を設定するには、次の 2 つの手順を実行します。

- 242ページの「`resolv.conf` ファイルの作成」の手順に従って `/etc/resolv.conf` ファイルを作成します。
- 27ページの「DNS を使用できるようにする」の手順に従って、`/etc/nsswitch.conf` ファイルを作成します。

DNS サーバーとして使う予定のホスト上に DNS サービスを設定する場合は、ブートファイルとデータファイルも設定する必要があります。詳細は、第 13 章を参照してください。

---

## リゾルバ

DNS クライアントはダイナミックライブラリルーチン群 (「リゾルバ」と総称する) を使用して、リモートホストの位置を調べます。リゾルバはネームサーバー上の DNS データベースを照会します。最終的に返されるのは、リゾルバによって要求されたマシンのホスト名あるいは IP アドレスです。DNS ネームサーバーは、自らのローカルドメインの外のサーバーのクライアントとなるので、やはりリゾルバを動作させます。

DNS ネームサーバーは、いくつかのファイルを使用して、そのデータベースをロードします。リゾルバのレベルでは、必要な情報を取得できるサーバーのアドレスを登録するファイル (`/etc/resolv.conf`) が必要です。リゾルバは `resolv.conf` ファイルを読み取り、ローカルドメインの名前とネームサーバーの位置を見つけます。リゾルバはローカルドメイン名を設定し、リゾルバルーチンに指示して、登録されたネームサーバーに情報を照会させます。通常、ネットワーク上の各 DNS クライアントシステムは、その `/etc` ディレクトリ内に `resolv.conf` ファイルが必

要です。resolv.conf ファイルがない場合、クライアントは IP アドレス 127.0.0.1 のサーバーをデフォルトで使います。

リゾルバがホストのアドレス (またはアドレスに対応する名前) を探さなければならないときには、照会パッケージを構築し、/etc/resolv.conf に登録されたネームサーバーにこれを送信します。サーバーは、その照会にローカルに応答するか、または他のサーバーのサービスを使ってリゾルバに答えを返します。

---

## resolv.conf ファイルの作成

例 12-1 に、doc.com ドメインのクライアント (非サーバー) マシン用の簡単な resolv.conf ファイルの例を示します。

例 12-1 resolv.conf ファイル

```
; Sample resolv.conf file for the machine polaris
domain doc.com
; try local name server
nameserver 127.0.0.1
; if local name server down, try these servers
nameserver 123.45.6.1
nameserver 111.22.3.5
; sort the addresses returned by gethostbyname(3c)
sortlist
130.155.160.0/255.255.240.0
130.155.0.0
```

このファイルの最初の行では、ドメイン名を次の書式で指定します。

```
domain domainname
```

ここで、*domainname* はインターネット管理組織 (このドキュメントの執筆時点では InterNIC。日本では JPNIC) に登録されている名前です。

---

注 - ドメイン名の末尾にスペースまたはタブを使うことはできません。ドメイン名の最後の文字を入力したら、必ずキャリッジリターンで強制改行してください。

---

2 行目では、ループバックネームサーバーを次の書式で指定します。

```
nameserver 127.0.0.1
```

それ以降の行では、主 DNS ネームサーバー、副 DNS ネームサーバー、またはキャッシュオンリーネームサーバーの IP アドレスを最大 3 つまで指定します。4 つ以上指定することはできません。各行の書式は次のとおりです。

```
nameserver IP_address
```

ここで、*IP\_address* の部分には、実際に主 DNS ネームサーバーまたは副 DNS ネームサーバーの IP アドレスを指定します。リゾルバは、必要な情報が見付かるまで、ここに指定されている順番どおりにネームサーバーを探していきます。

*/etc/resolv.conf* ファイルの 5 行目は、アドレス *sortlist* を次の書式で表示します。:

```
sortlist  
addresslist
```

*addresslist* は、*gethostbyname(3c)* によって戻されるアドレスのソートの順番を指定します。上記の列では、*gethostbyname* は、IP アドレス 130.155.0.0 より先に 1 組のネットマスク 130.155.160.0/255.255.240.0 を戻します。

---

## ***/etc/nsswitch.conf* ファイルの修正**

クライアントマシンを DNS に対応させる方法は、企業レベルで使われているネームサービスによって異なります。

- 「NIS+」

企業レベルで主として使っているネームサービスが NIS+ の場合、27ページの「DNS を使用できるようにする」の手順に従います。

- 「NIS」

企業レベルで主として使っているネームサービスが NIS で、かつ、それが正しく設定されている場合、何もする必要はありません。構成に問題がなければ、NIS はインストール直後の状態のまま DNS を使用できます。

- 「ファイルベース」

企業レベルで主として使っているネームサービスが `/etc` ファイルベースのサービスの場合、27ページの「DNS を使用できるようにする」の手順に従います。

`nsswitch.conf` ファイルの詳細は、『Solaris ネーミングの管理5』を参照してください。

## DNS サーバーの設定

---

この章では、DNS (Domain Name System) ネームサーバーを設定する方法について説明します。

- 245ページの「DNS サーバーの設定」
- 247ページの「サーバーの構成とデータファイルの名前」
- 250ページの「ドメイン名」
- 251ページの「`resolv.conf` ファイル」
- 252ページの「`named.conf` ファイル」
- 255ページの「サーバー機能の指定」
- 260ページの「データファイルの設定」
- 267ページの「サーバーの初期設定」
- 267ページの「設定の確認」
- 269ページの「サブドメインの設定」
- 272ページの「非インターネットルートマスターの設定」
- 273ページの「実例」

---

### DNS サーバーの設定

#### ▼ DNS サーバーの設定

DNS サーバーを設定する手順は次のとおりです。

1. サーバーを **DNS** クライアントとして設定します (サーバーの `resolv.conf` ファイルの設定も含む)。  
詳細は、第 12 章を参照してください。
2. ブートファイルを設定します。  
詳細は、252ページの「`named.conf` ファイル」を参照してください。
3. データファイルを設定します。  
詳細は、260ページの「データファイルの設定」を参照してください。次の 4 つのデータファイルを設定する必要があります。
  - a. `named.ca` ファイルを設定します。  
詳細は、261ページの「`named.ca` ファイルの設定」を参照してください。
  - b. `hosts` ファイルを設定します。  
詳細は、264ページの「`hosts` ファイルの設定」を参照してください。
  - c. `hosts.rev` ファイルを設定します。  
詳細は、265ページの「`hosts.rev` ファイルの設定」を参照してください。
  - d. `named.local` ファイルを設定します。  
詳細は、266ページの「`named.local` ファイルの設定」を参照してください。
4. サーバーの初期設定を行います。  
詳細は、267ページの「サーバーの初期設定」を参照してください。
5. サーバーをテストします。  
詳細は、267ページの「設定の確認」を参照してください。

---

注 - DNS の最も一般的な役割は、ローカルなネットワークをグローバルなインターネットに接続することです。そして、インターネットに接続するためには、親ドメインの管理者にネットワークの IP アドレスを登録してもらう必要があります。誰が管理するのかは、ネットワークの地理的な位置と、親ドメインの種類によって異なります。ドメイン管理者にネットワークを登録してもらう方法については、本書では説明をしていません。

---

---

## サーバーの構成とデータファイルの名前

in.named デーモンを正しく機能させるには、1つの構成ファイルと4つのデータファイルが必要です。



---

**注意** - この例で使われている IP アドレスとネットワーク番号、および本書で使われているサンプルコードは、説明に具体性を持たせるために仮に決めたものです。実際のネットワークやホストに使われていることがありますので、そのまま使うのは避けてください。

---

### 構成ファイル

マスターサーバーの構成ファイルは、`/etc/named.conf` です (252ページの「`named.conf` ファイル」を参照)。構成ファイルには、ドメイン名と、ホスト情報が含まれているファイル名が記述されています (`named.conf` ファイルの詳細は、『Solaris ネーミングの管理』を参照)。

### DNS データファイルの名前

内部で一貫性が取れていれば、ゾーンデータファイルには何でも好きな名前を付けることができます。この柔軟性のために他のサイトで作業をしたり、DNS 関連の他のマニュアルなどを参照する場合に、混乱するかもしれません。

たとえば、Sun のマニュアルや大多数の Solaris サイトで使われているファイル名は、『*DNS and BIND*』(Paul Albelz & Crilcet Liu 著、浅羽登志也／上水流由香監訳、アスキー出版局、1995年) で使われているファイル名とは異なります。そしてこれら2派の命名方法は、『*Name Server Operations Guide for BIND*』(カリフォルニア州立大学刊、ハブリックドメイン) の命名方法とも若干の相違があります。

さらに、本書とその他の DNS 関連のマニュアルでは、説明にはファイルの主な役割を表す総称名を使い、レコード例には具体的な固有の名前を使っています。たとえば、Solaris のネームサービスに関するマニュアルでは、ファイルの機能や役割を説明する場合は `hosts` という総称名を使い、例では `db.doc` や `db.sales` といった名前を使っています。

参考のため、表 13-1 でこれら3派の BIND ファイル名を比較します。

表 13-1 ファイル名

Solaris	O'Reilly その他	カリフォルニア州立 大学バークレイ校	ファイルの内容と役割
/etc/named.conf (全て同じファイル名)			BIND 8.1 では新たに named.conf ファイルを追加して、従来の named.boot ファイルと置きかえる。この構成ファイルにはセキュリティ、スタートアップオプション、ロギングが追加されている。このファイルによって、稼働中のサーバーの種類を指定して、すべてのゾーンまたはサーバーではなく、ゾーンごとにまたはサーバーごとにオプションを選択的に適用する。ドメイン名とデータファイル名が記述されている
/etc/resolv.conf (全て同じファイル名)			各クライアント (DNS サーバーを含む) 上に存在するファイル。DNS 情報を探すためにクライアントが照会するサーバーを示す
named.ca	db.cache db.root	root.cache	ルートサーバー名とそのアドレスがリストされている
総称名: hosts	総称名: db.domain	総称名: hosts	サーバーがサービスを提供するローカルゾーン内のマシンに関する全データが格納されている
例: db.doc	例: db.movie	例: ucbhosts	
db.sales	db.fx		
総称名: hosts.rev	総称名: db.ADDR	総称名: hosts.rev	逆変換 (アドレスから名前への変換) が可能な特殊ドメイン in-addr.arpa. 内のゾーンを指定する
例: doc.rev	例: db.192.249.249 db.192.249.253		



表 13-1 ファイル名 続く

Solaris	O'Reilly その他	カリフォルニア州立 大学バークレイ校	ファイルの内容と役割
named.local	総称名: db.cache 例: db.127.0.0	named.local	ローカルループバックインタフェース (ローカルホスト)用のアドレスを指定 する
\$INCLUDE 全てで同じ規則			データファイル内の \$INCLUDE() 文 によって指定されるファイル

## データファイル

必要なデータファイルは次の 4 種類あります。

- /var/named/named.ca

named.ca ファイルの詳細は、261ページの「named.ca ファイルの設定」、および『Solaris ネーミングの管理』を参照してください。内部で一貫性さえとれていれば、このファイルには何でも好きな名前を付けることができます。

- /var/named/hosts

hosts ファイルの詳細は、264ページの「hosts ファイルの設定」、および『Solaris ネーミングの管理』を参照してください。

hosts はファイルの目的や中身を表す総称名です。しかし、/etc/hosts との混乱を避けるため、このファイルには hosts 以外の名前を付けることをお勧めします。最も一般的な名前の例は、db.domainname です。たとえば、doc.com ドメインにある hosts ファイルであれば、その名前は db.doc となります。

ゾーンがいくつかに分かれている場合、各ゾーンにはそれぞれ専用の hosts ファイルが必要です。そして、各ゾーンの hosts ファイルにはそれぞれ固有の名前が付けられていなければなりません。たとえば、doc.com と sales.doc.com に分けられている DNS ドメインであれば、一方の hosts ファイルの名前は db.doc、もう一方の名前は db.sales とします。

- /var/named/hosts.rev

hosts.rev ファイルの詳細は、265ページの「hosts.rev ファイルの設定」、および『Solaris ネーミングの管理』を参照してください。

hosts.rev はファイルの目的や中身を表す総称名です。ゾーンがいくつかに分かれている場合、各ゾーンにはそれぞれ専用の hosts.rev ファイルが必要です。そして、各ゾーンの hosts.rev ファイルにはそれぞれ固有の名前が付けられていなければなりません。たとえば、doc.com と sales.doc.com に分けられている DNS ドメインであれば、一方の hosts.rev ファイルの名前は doc.rev、もう一方の名前は sales.rev とします。

■ /var/named/named.local

named.local ファイルの詳細は、266ページの「named.local ファイルの設定」、および『Solaris ネーミングの管理』を参照してください。内部で一貫性が取られてさえいれば、このファイルには何でも好きな名前を付けることができます。

## \$INCLUDE ファイル

DNS データファイルの \$INCLUDE() 宣言で名前が指定されているファイルのことをインクルードファイルといいます。\$INCLUDE ファイルを使ってデータを型ごとに別々のファイルに分割しておくとう便利です。(詳細は、『Solaris ネーミングの管理』を参照)。

---

## ドメイン名

「ドメイン名」とは、ローカルネットワークの中で DNS 管理ファイルを共有する複数のシステムを 1 つのグループとして扱い、そのグループに付けた名前のことです。ドメイン名は、ネットワーク情報サービスデータベースが正常に動作するために必要です。

## デフォルトのドメイン名

DNS がデフォルトで使用するドメイン名は resolv.conf ファイルに指定されています。

- resolv.conf ファイルがない場合、あるいは resolve.conf ファイルにデフォルトのドメイン名が指定されていない場合で、しかも、企業レベルで使っている

ネームサービスが NIS+ または NIS のどちらかである場合は、DNS はそれらのサービスからデフォルトのドメイン名を取得します。

- `resolv.conf` ファイルがない場合、あるいは `resolv.conf` ファイルにドメイン名が指定されていない場合で、しかも、NIS+ と NIS のどちらも使っていない場合は、そのドメインを指定するすべてのマシンに `resolv.conf` ファイルを格納するか (241ページの「リゾルバ」を参照)、`LOCALDOMAIN` 環境変数を設定しなければなりません。

## ドメイン名の末尾のドットについて

各種 DNS ファイルに関連して、ドメイン名の末尾のドットには次のような規則があります。

- `hosts`、`hosts.rev`、`named.ca`、`named.local` の各データファイルの中では、ファイル名の末尾にドットを付けます。たとえば、`sales.doc.com.` は、これらのファイルの中では有効です。
- `named.boot` ファイル内、または `resolv.conf` ファイルの中では、ドメイン名の末尾にドットを付けません。たとえば、`sales.doc.com` は、これらのファイルの中では有効です。

---

## `resolv.conf` ファイル

次に `resolv.conf` ファイルの設定方法を説明します。

### ▼ `resolv.conf` ファイルの設定

ここでは、`doc.com` ドメインのサーバーに使う簡単な `resolv.conf` ファイルの例を示します。

例 13-1 DNS サーバー用 `resolv.conf` ファイルの例

```
;
; /etc/resolv.conf file for dnsmaster (sirius)
;
domain                doc.com
```

(続く)

続き

```
nameserver      0.0.0.0
nameserver      111.22.3.5
```

/etc/resolv.conf ファイルの最初の行には、ドメイン名が次の書式で指定されています。

```
domain domainname
```

ここで、*domainname* の部分には、インターネット管理団体 (本書執筆の時点では InterNIC。日本では JPNIC) によって登録されている名前を指定します。

注・ドメイン名の末尾にスペースまたはタブを使うことはできません。ドメイン名の最後の文字を入力したら、必ずキャリッジリターンで強制改行してください。

2 行目には、サーバー自体を次の書式で指定します。

```
nameserver 0.0.0.0
```

それ以降の行では、副 DNS ネームサーバー、またはキャッシュ専用ネームサーバーの IP アドレスを 1 つまたは 2 つ指定します。リゾルバはこれらの行を照会して該当するアドレスを識別します。各行の書式は次のとおりです。

```
nameserver IP_address
```

ここで、*IP\_address* の部分には、副 DNS ネームサーバー、またはキャッシュ専用ネームサーバーの IP アドレスを指定します。リゾルバは、必要な情報が見つかるまで、ここに指定されている順番どおりにネームサーバーを探していきます。

## named.conf ファイル

BIND 8.1 では新たに構成ファイル /etc/named.conf を追加し

て、/etc/named.boot ファイルと置きかえます。/etc/named.conf ファイルは、主、副、キャッシュ専用のネームサーバーを確立し、サーバーが権限を持つゾーンと、初期データを取得するために読み取るデータファイルを指定します。

/etc/named.conf ファイルには、以下の機能を実装するための文が含まれます。

- アクセスコントロールリスト (ACL) を用いたセキュリティ ACL には、NIS+ ホストが読み取り/書き込みアクセスを持つ IP アドレスの集合を定義する
- ロギング動作の指定
- すべてのゾーンにではなく、ゾーンのセットに対して選択的に適用されるオプション

サーバーのスタートアップスクリプト `/etc/init.d/inetsvc` によってデーモンが起動されたとき、構成ファイルは `in.named` によって読み取られます。

`named.conf` ファイルは、文とコメントによって記述されます。文はセミコロンで終わります。複数の文から成るブロックを含む文もあります。ブロック内の各文もセミコロンで終わります。

表 13-2 `named.conf` で使用する文

<code>acl</code>	アクセスコントロール用に使用する、IP アドレスの一致リストを名前を付けて定義する。アドレスの一致リストは 1 つまたは複数の IP アドレス (ドットで区切った 10 進表記) または IP プレフィックス (ドットで区切った 10 進表記の後にスラッシュとネットマスクのビット数が付く) を指定する。名前を付けた IP アドレスの一致リストは、他の場所で使用する前に <code>acl</code> 文で定義されている必要がある。順方向の参照は許可されない
<code>include</code>	<code>include</code> 文がある箇所にインクルードファイルを挿入する。 <code>include</code> を使用することで、管理しやすいまとまりに構成情報を分割することができる
<code>key</code>	特定のネームサーバーでの認証と承認に使用されるキー ID を指定する。 <code>server</code> 文を参照
<code>logging</code>	サーバーが記録するログの種類と、ログメッセージの送り先を指定する
<code>options</code>	グローバルなサーバー構成のオプションを制御して、他の文に対するデフォルト値を設定する
<code>server</code>	リモート用ネームサーバーに関して、指定された構成オプションを設定する。すべてのサーバーに対してではなく、サーバーごとに選択的にオプションを適用する
<code>zone</code>	ゾーンを定義する。すべてのゾーンに対してではなく、ゾーンベースごとに選択的にオプションを適用する

例 13-2 コード例 13-2 主サーバーに対するマスター構成例

```
options {
    directory "/var/named";
    datasize 2098;
    forward only;
    forwarders {
        99.11.33.44;
    };
    recursion no;
    transfers-in 10;
    transfers-per-ns 2;
    allow-transfer {
        127.0.1.1/24;
    };
};

logging {
    category queries { default_syslog; };
};

include "/var/named/abcZones.conf"

// here are the names of the primary files
zone "cities.zn" {

    type master;
    file "db.cities.zn";
};

zone "0.0.127.in-addr.arpa" {
    type master;
    file "db.127.cities.zn";
};

zone "168.192.in-addr.arpa" {
    type master;
    file "db.cities.zn.rev";
};

zone "sales.doc.com" {
    type slave;
    file "slave/db.sales.doc";
    masters {
        192.168.1.151;
    };
};

zone "168.192.in-addr.arpa" {
    type slave;
    file "slave/db.sales.doc.rev";

    masters {
        192.168.1.151;
    };
};
```

(続く)

```
};
```

## BIND 4.9.x から BIND 8.1 への移行

スーパーユーザーになって、Korn シェルスクリプト `/usr/sbin/named-bootconf` を実行し、BIND 4.9.x の `named.boot` ファイルを BIND 8.1 の `named.conf` ファイルに変換します。`named-bootconf(1M)` を参照してください。

---

注 - Solaris 8 では、`named.boot` ファイルは無視されます。

---

## サーバー機能の指定

DNS サーバーには次の役割のいくつかを果たします。

- 「ゾーン主マスターサーバー」

各ゾーンには、主マスターサーバーを1つずつ指定します。主マスターサーバーとは、そのゾーンにおける正規の機能を有するサーバーのことをいいます (256 ページの「主マスターサーバーの指定」を参照)。

- 「ゾーン副マスターサーバー」

各ゾーンには、「副」マスターサーバーを1つまたは複数指定します。副マスターサーバーは、そのゾーンにおける主マスターサーバーから DNS を取得するサーバーのことをいいます。副サーバー上のデータは直接変更しません。あるゾーンの主サーバー上のデータファイルが変更された場合、主サーバーを通じて副サーバーのファイルが更新されます (258 ページの「副マスターサーバーの指定」を参照)。

- 「キャッシュオンリーサーバー」

DNS データのキャッシュを維持するという意味では、すべてのサーバーがキャッシュサーバーであるといえます。キャッシュオンリーサーバーは、`in-addr.arpa.` ドメイン以外のどのゾーンのマスターサーバーでもないサーバーです。(259 ページの「キャッシュオンリーサーバーの指定」を参照)。

## ■ 「ルートドメインサーバー」

インターネットに接続されているネットワークでは、ルートドメインサーバーがインターネット上に置かれることとなります。この場合は、`named.ca` ファイルに各ルートドメインサーバーの IP アドレスを指定するだけで済みます (261 ページの「`named.ca` ファイルの設定」を参照)。ネットワークがインターネットに接続されていない場合は、独自にルートドメインサーバーを設定しなければなりません (272 ページの「非インターネットルートマスターの設定」を参照)。

これらのさまざまなサーバーの機能は同じマシンで実行できます。たとえば、あるゾーンで主マスターサーバーとして使っているマシンを、他のゾーンでは副マスターサーバーとして使うことができます。本書で「主」、「副」、「キャッシュ専用」とマシンを形容している場合、それは特定のマシンを指しているのではなく、あるゾーンでマシンに割り当てられた役割を指しています。

これらのサーバーの役割の詳細は、『*Solaris* ネーミングの管理』を参照してください。

## 主マスターサーバーの指定

あるゾーンであるマシンを主サーバーとして使うことを指定する場合は、そのマシン (サーバー) 上の `named.boot` ファイルに次の 3 つの「主」レコードを書き込みます。

### 1. ゾーンの「主」レコードを作成します。

このレコードは、そのサーバーを主サーバーとして使用するゾーンを指定し、正規の `hosts` ファイルの場所を示すものです。この「主」レコードは次の 3 つのフィールドで構成されます。

- 第 1 フィールド - サーバーを主 (primary) サーバーとして指定する
- 第 2 フィールド - 対象のゾーンを指定する
- 第 3 フィールド - `hosts` ファイルを指定する

次に示すブートファイルの行は、あるサーバーを `doc.com` ゾーンで主サーバーとして使い、正規の `hosts` ファイルとして `db.doc` を使うことを示すものです。

```
primary    doc.com    db.doc
```

### 2. ゾーンの逆変換の「主」レコードを作成します。



このレコードは、そのサーバーを逆アドレス変換 (つまり、doc.com の逆アドレスドメイン) の主サーバーとして使うことを指定し、正規の hosts ファイルの場所を示すものです。この「主」レコードには 3 つの構成要素があります。第 1 フィールドではサーバーを主サーバーとして指定します。第 2 フィールドでは対象のゾーンを指定します。第 3 フィールドでは hosts.rev ファイルを指定します。

あるゾーンにおける逆アドレスドメインは、そのゾーンにおける IP アドレスを逆にならべ、最後に in-addr.arpa を配したものです。たとえば、doc.com ゾーンの IP アドレスが 123.45.6. だとすると、逆アドレスドメインは 6.45.123.in-addr.arpa になります。

次に示すブートファイルの行は、そのサーバーを doc.com ゾーンの逆アドレスドメインで主サーバーとして使い、正規の hosts ファイルとして doc.rev を使うことを示すものです。

```
primary 6.45.123 . in-addr.arpa doc.rev
```

3. ローカルループバック (ホスト) の逆アドレス関連の「主」レコードを作成します。

このレコードは、そのサーバーをループバックホストの主サーバーとして使うことを指定し、正規の hosts ファイルの場所を示すものです。この「主」レコードには 3 つの構成要素があります。第 1 フィールドではサーバーを主サーバーとして指定します。第 2 フィールドではループバックホストの逆アドレスを指定します。第 3 フィールドでは hosts.rev ファイルを指定します。

---

注 - ループバックホストは常に、0.0.127.in-addr.arpa といった書式で識別されます。

---

次に示すブートファイルの行は、そのサーバーをループバックホストの逆アドレスドメインで主サーバーとして使い、正規の hosts ファイルとして named.local を使うことを示すものです。

```
primary 0.0.127.in-addr.arpa named.local
```

## 副マスターサーバーの指定

あるゾーンであるマシンを副サーバーとして使うことを指定する場合は、そのマシン (サーバー) 上の `named.boot` ファイルに「副」レコードを書き込みます。別々のレコードにより、サーバーをそのゾーン、逆アドレスドメイン、およびループバックホストの副サーバーとして指定できます。

この「副」レコードは次の3つのフィールドで構成されます

- 第1フィールド - サーバーを副サーバーとして指定する
- 第2フィールド - 対象のゾーンを指定する
- 第3フィールド - そのゾーンの主サーバーの IP アドレスを指定する。副サーバーは主サーバーから正規データを取得する

「副」レコードでは、必須フィールドに続けて1つまたは複数の任意フィールドを設けることができます。任意フィールドには次の種類があります。

- 「副サーバー」

主サーバーの IP アドレスに続けて、他の副サーバーの IP アドレスを指定できます。これにより、副サーバーが情報を入手できる情報源が増えます。一方、状況によっては、副サーバーの IP アドレスを指定することで、パフォーマンスが下がることも考えられます (IP アドレスがマルチホーム主サーバーのもう一方のネットワークアドレスである場合を除く)。

- 「バックアップファイル」

主サーバー (および任意指定の副サーバー) の IP アドレスに続けて、バックアップ `hosts` ファイルの名前を指定できます。バックアップファイル名を指定すると、副サーバーはそのバックアップファイルからデータをロードし、続いて主サーバー (および任意指定の副サーバー) をチェックしてバックアップアプリケーションファイルのデータが最新のものであるかどうかを確認します。その結果、最新ではないことが分かった場合は、主サーバーから受け取った情報に基づいて更新されます。

次に示すブートファイルの行は、あるサーバーを `doc.com` ゾーンと逆アドレスドメインの副サーバーとして使うことを示します。さらに、その副サーバーが IP アドレス `129.146.168.119` の主サーバーから正規データを受け取り、サーバー `192.146.168.38` をゾーンデータの副情報源として使い、最初に `doc.com.bakup` ファイルから初期データをロードすることを示します。

```
secondary doc.com 129.146.168.119 192.146.168.38 doc.com.bakup
```

```
secondary 4.0.32.128.in-addr.arpa 129.146.168.119
```

この章で紹介するさまざまなサンプルファイルの中で、サーバー `dnssecondary` (IP アドレス `192.146.168.38` の `sirius` マシンの別名) のブートファイルがたびたび登場しますが、そのいずれもが上記のブートファイルを指しています。

注 - 1 台のサーバーは、1 つまたは複数のゾーンの主サーバーとして機能し、さらに 1 つまたは複数のゾーンの副サーバーとしても機能できます。これを決めるのは、ブートファイル内のエントリの組み合わせです。

## キャッシュオンリーサーバーの指定

キャッシュオンリーサーバーは正規データは一切管理しません。キャッシュオンリーサーバーは照会を行い、`in.named` ファイルにリストされているホストを照会して必要な情報を探します。つまり、キャッシュオンリーサーバーは、正規のネームサーバーと同様の照会を行いますが、正規データそのものは一切管理しません。

例 13-3 では、キャッシュオンリーサーバー用のブートファイルのサンプルを示します。

例 13-3 キャッシュオンリーサーバーのブートファイルの例

```
;
; Sample named.boot file for caching-only name server
;
; type          domain          source file or host
;
directory /var/named
cache      .          named.ca
primary    0.0.127.in-addr.arpa  named.local
```

サーバーをキャッシュオンリーサーバーとして指定するための行は特に必要ありません。ブートファイル内に `secondary` または `primary` など、権限に関する行が

ないということが、キャッシュオンリーサーバーであると判断する根拠になります (以下に示すものを除く)。

- ブートファイルの `directory` 行
- ブートファイルの `primary 0.0.127.in-addr.arpa` 行
- ブートファイルの `cache . named.ca` 行

---

## データファイルの設定

DNS デーモン `in.named` が使用するすべてのデータファイルは、標準資源レコード書式で記述されます。標準資源レコード書式では、ファイルの各行は、資源レコード (RR) と呼ばれるレコードです。各 DNS データファイルには決められた資源レコードが必要です。

資源レコード、その書式、フィールド、特殊文字、制御行 (エントリ) の詳細は、『Solaris ネーミングの管理』を参照してください。

## 資源レコードの種類

最も一般的に使用される資源レコードの種類を表 13-3 に示します。資源レコードは通常、表 13-3 の順番で入力されますが、必ずしもそうしなければならないということではありません。

表 13-3 一般的に使用される資源レコードの種類

種類	説明
SOA	権限の始まり
NS	ネームサーバー
A	IPv4 インターネットアドレス (名前からアドレスへ)
AAAA	IPv6 インターネットアドレス (名前からアドレスへ)
PTR	ポインタ (アドレスから名前へ)
CNAME	標準名 (ニックネーム)

表 13-3 一般的に使用される資源レコードの種類 続く

種類	説明
TXT	テキスト情報
MX	メール交換

これらの資源レコードの詳細は、『Solaris ネーミングの管理』を参照してください。

これ以降に示すサンプルファイルでは、@ は現在のゾーンまたは現在の起点を示します。セミコロン (;) で始まる行はコメントです。

## named.ca ファイルの設定

/var/named/named.ca ファイルには、ルートサーバーの名前とアドレスが格納されています。ルートサーバー名は NS レコードに、アドレスは A レコードに示されています。この named.ca ファイルを使用するサーバーごとに、NS レコードと A レコードを追加する必要があります。

named.ca ファイルの入手方法あるいは作成方法は、ネットワークがインターネットに接続されているかどうかによって異なります。

## インターネット named.ca ファイル

ネットワークがインターネットに接続されているのであれば、InterNIC Registration Service (本書執筆の時点) から次の手段で named.ca ファイルを入手できます。

### ■ 匿名 FTP

FTP サイトは ftp.rs.internic.net、ファイル名は /domain/named.root です。

### ■ Gopher

Gopher サイトは rs.internic.net です。ファイルは named.root であり、これは「InterNIC Registration Service」メニューの「InterNIC Registration Archives」サブメニューで見つけることができます。

本書で説明した命名規則に従うならば、named.root を /var/named/named.ca に移動します。

例 13-4 インターネット named.ca ファイルの例

```

;
; formerly NS1.ISI.EDU
.
B.ROOT-SERVERS.NET. 3600000 NS B.ROOT-SERVERS.NET.
3600000 A 128.9.0.107
;
; formerly C.PSI.NET
.
C.ROOT-SERVERS.NET. 3600000 NS C.ROOT-SERVERS.NET.
3600000 A 192.33.4.12
;
; formerly TERP.UMD.EDU
.
D.ROOT-SERVERS.NET. 3600000 NS D.ROOT-SERVERS.NET.
3600000 A 128.8.10.90
;
; formerly NS.NASA.GOV
.
E.ROOT-SERVERS.NET. 3600000 NS E.ROOT-SERVERS.NET.
3600000 A 192.203.230.10
;
; formerly NS.ISC.ORG
.
F.ROOT-SERVERS.NET. 3600000 NS F.ROOT-SERVERS.NET.
3600000 A 192.5.5.241
;
; formerly NS.NIC.DDN.MIL
.
G.ROOT-SERVERS.NET. 3600000 NS G.ROOT-SERVERS.NET.
3600000 A 192.112.36.4
;
; formerly AOS.ARL.ARMY.MIL
.
H.ROOT-SERVERS.NET. 3600000 NS H.ROOT-SERVERS.NET.
3600000 A 128.63.2.53
;
; formerly NIC.NORDU.NET
.
I.ROOT-SERVERS.NET. 3600000 NS I.ROOT-SERVERS.NET.
3600000 A 192.36.148.17
;
; temporarily housed at NSI (InterNIC)
.
J.ROOT-SERVERS.NET. 3600000 NS J.ROOT-SERVERS.NET.
3600000 A 198.41.0.10
;
; temporarily housed at NSI (InterNIC)
.
K.ROOT-SERVERS.NET. 3600000 NS K.ROOT-SERVERS.NET.
3600000 A 198.41.0.11
;
; temporarily housed at ISI (IANA)
.
L.ROOT-SERVERS.NET. 3600000 NS L.ROOT-SERVERS.NET.
3600000 A 198.32.64.12
;
; temporarily housed at ISI (IANA)
.
M.ROOT-SERVERS.NET. 3600000 NS M.ROOT-SERVERS.NET.
3600000 A 198.32.65.12
; End of File

```

## 非インターネット named.ca ファイル

ネットワークがインターネットに接続されていない場合は、独自の named.ca ファイルを作成する必要があります。それには、サーバーのどれか 1 つをルートサーバーとし、DNS サーバーごとにそのルートサーバーを指す named.ca ファイルを作成します。

たとえば、ドメイン private においてマシン ourroot を非インターネットルートサーバーに指定する場合を想定します。ourroot の IP アドレスが 192.1.1.10 であるとする、named.ca ファイルには次の行を書き込みます。

```
ourroot.private. 999999 IN A 192.1.1.10
```

キャッシュファイルも SOA レコード、各ドメインおよびサブドメインの NS レコード、各サーバーの A レコードを必要とします。

たとえば、ourroot の他に、ourprimary と oursecondary という 2 つの DNS ネームサーバーがあるとします。その場合、DNS サーバー上の named.ca ファイルはいずれも次のようになります。

### 例 13-5 named.ca ファイル (非インターネット) の例

```
;
@ IN SOA ourroot.private. hermit.ourroot.private (
    1997071401 ; serial number (YYYYMMDD##)
    10800 ; refresh after 3 hours
    3600 ; retry after 1 hour
    604800 ; expire after 1 week
    86400 ) ; minimum TTL of 1 day
;
ourroot.private. 999999 IN A 192.1.1.10
;
private. IN NS ourprimary.private.
1.1.192.in-addr.arpa IN NS ourprimary.private.
ourprivate.private. IN A 192.1.1.1
;
private. IN NS oursecondary.private.
1.1.192.in-addr.arpa IN NS oursecondary.private.
oursecondary.private. IN A 192.1.1.2
```

インターネットに接続されていないドメインの設定の詳細は、272ページの「非インターネットルートマスターの設定」を参照してください。

## hosts ファイルの設定

hosts ファイルには、ゾーン内にある全マシンの全データが収められています。複数のドメインにまたがっているゾーンの場合は、そのゾーンを構成する全ドメインの全マシンがそのゾーンのホストファイルに列挙されます (270ページの「単一ゾーンのサブドメインの設定」を参照)。

注 - hosts という名前はファイルの役割や中身を表す総称名です。この総称名をそのまま使うと /etc/hosts と紛らわしいので、この種のファイルは hosts 以外の名前にすることをお勧めします。ドメイン内に複数のゾーンがある場合は、各ゾーンに1つずつ hosts ファイルを置き、しかも、各ゾーンの hosts ファイルには一意の名前を付けなければなりません。たとえば、DNS ドメイン内に doc.com と sales.doc.com という2つのゾーンがある場合は、1つを db.doc、もう1つを sales.db.doc という名前にするとよいでしょう。

各ゾーンには個別の、一意の名前を持つ hosts ファイルが必要です。複数のゾーンが存在する場合は、各ゾーンの hosts ファイルには他のゾーンのマスター (主、副) サーバーに関する情報も含める必要があります。詳細は、271ページの「複数ゾーンのサブドメインの設定」を参照してください。

### 例 13-6 hosts ファイルの例

```
;
; SOA rec
doc.com IN SOA sirius.doc.com sysop.centauri.doc.com (
    1997071401      ; serial number (YYYYMMDD##)
    10800          ; refresh every 3 hours
    10800          ; retry every 3 hours
    604800         ; expire after a week
    86400 )        ; TTL of 1 day

; Name Servers
doc.com          IN NS  sirius.doc.com
sales.doc.com    IN NS  altair.sales.doc.com

; Addresses
localhost        IN A  127.0.0.1
sirius           IN A  123.45.6.1
rigel            IN A  123.45.6.112

antares          IN A  123.45.6.90
polaris          IN A  123.45.6.101
procyon          IN A  123.45.6.79
tauceti          IN A  123.45.6.69
altair.sales.doc.com IN A  111.22.3.4
; aliases
durvasa          IN  CNAME sirius.doc.com
```

(続く)



続き

```
dnsmastr          IN CNAME sirius.doc.com
dnssales          IN CNAME altair.sales.doc.com
```

hosts ファイルは、通常、次の 5 つの要素で構成されています。

- SOA (権限の始まり) レコード
- 1 つまたは複数の NS (ネームサーバー) レコード。主および副の DNS ネームサーバーを示す
- A (アドレス) レコード。ゾーン内の各ホストに必要
- ゾーン内のホストの別名に対する CNAME (標準名) レコード
- MX (メール交換) レコード

これらの資源レコードの詳細は、『Solaris ネーミングの管理』を参照してください。

## hosts.rev ファイルの設定

hosts.rev は逆マッピングを設定するファイルです。

注 - hosts.rev という名前はファイルの役割や中身を表す総称名です。ドメイン内に複数のゾーンがある場合は、各ゾーンに 1 つずつ hosts.rev ファイルを置き、しかも、各ゾーンの hosts.rev ファイルには一意の名前を付けなければなりません。たとえば、DNS ドメイン内に doc.com と sales.doc.com という 2 つのゾーンがある場合は、1 つを doc.rev、もう 1 つを sales.rev という名前にするとよいでしょう。

### 例 13-7 hosts.rev ファイルの例

```
; SOA rec
6.45.123.in-addr.arpa. IN SOA sirius.doc.com sysop.centauri.doc.com (
    1997071401 ; serial number (YYYYMMDD##)
    10800      ; refresh every 3 hours
    10800      ; retry every 3 hours
```

(続く)

続き

```

                                604800      ; expire after a week
                                86400 )      ; TTL of 1 day
; Name Servers
6.45.123.in-addr.arpa.  IN  NS  sirius.doc.com
1                       IN  PTR  sirius.doc.com
```

hosts.rev ファイルは、通常、次の3つの要素で構成されています。

- SOA (権限の始まり) レコード
- 1つまたは複数の NS (ネームサーバー) レコード。主および副の DNS ネームサーバーを示す。サーバー名は省略形では指定できない
- PTR レコードゾーン内の各ホストに必要。マシン名は省略形では指定できない

これらの資源レコードの詳細は、『Solaris ネーミングの管理』を参照してください。

## named.local ファイルの設定

named.local ファイルは、ネームサーバーのローカルループバックインタフェースを設定します。

例 13-8 named.local ファイルの例

```

; SOA rec
0.0.127.in-addr.arpa. IN SOA sirius.doc.com sysop.centauri.doc.com (
                                1997071401      ; serial number (YYYYMMDD##)
                                10800             ; refresh every 3 hours
                                10800             ; retry every 3 hours
                                604800            ; expire after a week
                                86400 )          ; TTL of 1 day
; Name Servers
0.0.127.in-addr.arpa.  IN  NS  sirius.doc.com
1                       IN  PTR  localhost.
```

named.local ファイルは、通常、次の3つの要素で構成されています。

- SOA (権限の始まり) レコード。ゾーンの始まりを示す。named.local ファイルが置かれるホストの名前もこれに含まれる
- 1 つまたは複数の NS (ネームサーバー) レコード。主および副の DNS ネームサーバーを示す。サーバー名およびドメイン名は省略形では指定できない
- localhost の PTR レコード

これらの資源レコードの詳細は、『Solaris ネーミングの管理』を参照してください。

---

## サーバーの初期設定

サーバーの初期設定の手順は次のとおりです。

1. named.conf 構成ファイルとその他必要なファイルを設定します。
2. in.named を実行します。

```
# /usr/sbin/in.named
```

コマンドラインから in.named を実行する代わりに、リポートするという方法もあります。

---

## 設定の確認

ブートファイルとデータファイルを設定し、in.named を実行したら、設定が正しく行われたかどうかを次の手順で確認してください。

1. syslog ファイルをオープンして、エラーメッセージが書き込まれていないかどうか確認します。  
エラーメッセージと障害追跡の仕方については、『Solaris ネーミングの管理 Solaris Naming Administration Guide - ja』を参照してください。
2. nslookup を実行して、ローカルドメインのホスト名を確認します。

```
dnsmaster% nslookup altair
Server: dnsmaster.doc.com
Address: 192.146.168.5
Name: altair.doc.com
Address: 192.146.168.10
```

- 異常が見つからなければ、ネームサーバーはおそらく正常に機能しています。
- 「Can't find」または「can't initialize address」といったメッセージがサーバーに表示された場合、あるいは「Non-existent domain」といったメッセージが表示された場合は、サーバーがブートファイルまたはホストファイルに設定されていない可能性があります。
- 「can't find name」または「Non-existent domain」といったメッセージがサーバーに表示された場合は、検索したサーバーがサーバーの hosts ファイルに書き込まれていないか、resolv.conf ファイルのドメイン情報に誤りがある、あるいは、それ以外のサーバーの問題がある可能性があります。

### 3. nslookup を実行してリモートドメイン名を検索します。

インターネットに接続されているネットワークの場合、リモートドメイン名を検索します。インターネットに接続されていないネットワークの場合は、他のゾーンにサブドメインがあれば、その名前を検索します。たとえば、インターネット上のリモートドメイン名 `internic.net` を検索するには、次のように入力します。

```
dnsmaster% nslookup internic.net
Server: dnsmaster.doc.com
Address: 192.146.168.
Name: internic.net
Addresses: 198.41.0.9, 198.41.0.6, 198.41.0.5, 198.41.0.8
```

- 異常が見つからなければ、ネームサーバーはおそらく正常に機能しています。
- 上記のコマンドを実行してもリモートドメイン名が表示されない場合は、インターネットとの接続に問題があることが原因の1つとして考えられます。
- あるいは、named.ca ファイルが正しくインストールまたは設定されていないことも考えられます。

(もう一度 nslookup を実行すると、「non-authoritative」というメッセージが出るはずですが、これは無視してかまいません。2回目の実行では、リモートネームサーバーからではなく、キャッシュから応答が来ています。)

#### 4. リモートドメインから自分のドメインのホスト名を検索します。

インターネットに接続されているネットワークの場合、リモートドメインに行き、そこから自分のドメインのホスト名を検索します。インターネットに接続されていないネットワークの場合は、他のゾーンがあれば、そこから自分のドメインのホスト名を検索します。

たとえば、インターネット上のリモートドメインから自分のドメインにあるホスト名を検索するには、nslookup コマンドに続けて、引数を2つ指定します。1つめは検索対象のホスト名、2つめは nslookup コマンドを実行するネームサーバー名です。具体的には次のように指定します。

```
remotemachine9% nslookup altair remotemaster.foo.org.  
Server: remotemaster.foo.org  
Address: 123.231.12.22  
Name: altair.doc.com  
Addresses: 111.22.3.4
```

- 異常が見つからなければ、ネームサーバーはおそらく正常に機能しています。
- 上記のコマンドを実行しても探しているマシンが見つからない場合は、ドメインが親ドメイン (上記の例では .com) の管理者と正しく登録されていないことが原因の1つとして考えられます。

---

## サブドメインの設定

サブドメインの設定方法には、次の 2 通りがあります。

- 「単一ゾーン」

最も簡単な方法は、サブドメインを親ドメインのゾーンに含めることです。こうすると、1 セットの DNS サーバーとデータファイルでドメインに関係なくすべてのマシンを管理できます。詳細は、270 ページの「単一ゾーンのサブドメインの設定」を参照してください。

単一ゾーン方式の長所は、管理が簡素化され簡単なことです。短所は 1 セットのサーバーですべてのゾーンのドメインにあるマシンを管理しなければならないということです。マシンの数が多すぎると、サーバーの負荷が大きくなり過ぎ、パフォーマンスが低下することがあります。

- 「複数ゾーン」

異なるゾーンに異なるドメインを含めることができます。あるゾーンのクライアントから他のゾーンのホストに関する情報をどうやって集めるかを考えなければならないので、この方法は上の方法より複雑です。詳細は、271 ページの「複数ゾーンのサブドメインの設定」を参照してください。

複数ゾーン方式の長所は、ドメインごとにその中のマシンを管理するサーバーセットを変更できるということです。つまり、サーバーの負荷を分散させ、1 セットのサーバーに負荷が集中するのを防ぐことができます。短所は、設定時の作業が複雑になることです。

## 単一ゾーンのサブドメインの設定

複数のドメインで構成されているゾーンのデータファイルには、そのゾーンによってカバーされる各ドメインのすべてのマシンとサーバーに関わるレコードが必要です。

複数のドメインで構成されているゾーンを設定するのも、単一ドメインで構成されているゾーンを設定するのも、やることは基本的に同じです。唯一の相違は、リモートドメインのマシンを識別できるようにするために、`hosts` ファイルには完全指定のドメイン名を使用しなければならないということです。サーバーのローカルドメインにあるマシンであれば、`hosts` ファイルにマシン名しか指定されていなくても識別できます。しかし、他のドメインにあるマシンを識別するには、完全指定のドメイン名、つまり `machine.domain` という書式で指定しなければなりません。

`hosts.rev` ファイルと `named.local` ファイルに指定するサーバー名やマシン名は、完全指定のドメイン名を使用する必要があります。しかし、これはゾーンがいくつのドメインで構成されているかを問いません。

## 複数ゾーンのサブドメインの設定

異なるゾーンのサブドメインを設定するのは、1つのゾーンに複数のドメインを含めるのよりも複雑です。というのも、さまざまなゾーンにあるクライアントが、他のゾーンの DNS 情報を得る方法を指定しなければならないからです。

ネットワークを複数のドメインに分ける場合、ドメインを階層化します。必ず最上位のドメインがあって、その下に1つまたは複数のサブドメインがあります。サブドメインの下にサブドメインを作ることができます。しかし、どのサブドメインにも、階層構造の中で決まった場所があります。ドメイン名は左から右に読んでいくと、階層内におけるドメインの位置を示していることがわかります。たとえば、doc.com ドメインは sales.doc.com の上にあり、west.sales.doc.com ドメインは sales.doc.com ドメインの下にあることがわかります。

DNS ゾーンはそれが含むドメインから階層を取り込みます。ネットワークのトップドメインを含むゾーンはトップゾーンになります。トップドメインの下のサブドメインを1つまたは複数含むゾーンは、ゾーンの階層でいえばトップゾーンの下にゾーンになります。DNS 情報があるゾーンから別のゾーンへ移動させるということは、このゾーン階層の中を上下に移動させるということです。つまり、各ゾーンは、すぐ上のゾーンに情報を渡すにはどうするか、すぐ下のゾーンに情報を渡すにはどうするかを、専用のデータファイルに指定しておく必要があります。

複数のゾーンで構成されているネットワークの中で、DNS 情報があるゾーンから別のゾーンへ正確に転送させるために必要なことを以下に示します。

### ■ hosts.rev ファイル

すぐ上のゾーンにある1つまたは複数のマスターサーバー名を指し示す PTR レコードが各 hosts.rev ファイルに必要です。上位ゾーンのサーバーを指し示すということを除けば、この種の PTR レコードは、ファイル内のその他の PTR レコードとまったく同じのものです。

### ■ hosts ファイルの NS レコード

すぐ下のゾーンにあるネームサーバー名を指し示すゾーン NS レコードが各 hosts ファイルに必要です。この種の NS レコードは、その最初のフィールドに、下のゾーン名が指定されていなければなりません。ゾーンの名前は、そのゾーンの host ファイルの SOA レコードに指定されています。

### ■ hosts ファイルの A レコード

すぐ下のゾーンにあるネームサーバーの IP アドレスを指し示す A レコードが各 hosts ファイルに必要です。この種の A レコードは、その最初のフィールド

に、下のゾーン名が指定されていなければなりません。ゾーン名は、そのゾーンの host ファイルの SOA レコードに指定されています。

表 13-4 は、2 つのゾーンを有するネットワークを示します。

## 非インターネットルートマスターの設定

インターネットに接続されているネットワークの場合、ルートドメインサーバーがルートドメインのインターネットサイト上に置かれることになります。この場合は、キャッシュファイルにそのサイトの IP アドレスを書き込むだけで済みます。261ページの「インターネット named.ca ファイル」を参照してください。

ネットワークに接続されていないネットワークの場合、ローカルネットワーク上のルートレベルドメインに主および副のネームサーバーを設定する必要があります。これにより、ネットワーク内のすべてのドメインで一貫性のある正規のサーバーを持つことができます。さもないと、いずれのマシンも照会を処理することができません。

たとえば、private という名前の非インターネットドメインで ourroot というマシンをルートサーバーとして指定する場合を想定します。ourroot マシンの IP アドレスが 192.1.1.10 だとすると、次のような手順を踏むことになります。

1. ルートサーバーを指し示す named.ca ファイルを主マスターサーバー上に作成します。

次のように入力します。

```
ourroot.private. 999999 A 192.1.1.10
```

named.ca ファイルには、内部のドメインとサブドメイン用の個別の NS レコードと、サーバー用の個別の A レコードが必要です。詳細は、272ページの「非インターネットルートマスターの設定」を参照してください。

2. ネットワーク上のすべての DNS ネームサーバーのブートファイルに、ルートドメインのキャッシュリソースレコードを追加します。

次のように入力します (詳細は、247ページの「構成ファイル」を参照)。

```
cache . named.ca
```



3. ルートサーバーのブートファイルから、ルートドメインのキャッシュリソースレコードを削除します。

つまり、ourroot のブートファイルから `cache.named.ca` 行を削除します。

4. ルートサーバーのブートファイルに、ルートドメインの `primary` 行を追加します。

たとえば、ourroot のブートファイルに次の行を追加します。

```
primary . ourroot.private
```



**注意** - 設定後、インターネットに接続することにした場合は、すべてのサーバーの `named.ca` ファイルを最新のインターネットファイルに置き換え、それまでルートサーバーだったマシンのブートファイルで、`primary . rootserver` 行を `cache . named.ca` 行に書き換える必要があります。

---

## 実例

この節では、この章で説明した例に基づいて、サンプルのインターネット接続ネットワークを想定し、そこで使う DNS を設定するために必要なファイルを示します。



**注意** - この例で使われている IP アドレスとネットワーク番号、および本書で使われているサンプルコードは、説明に具体性を持たせるために仮に決めたものです。それらは、実際のネットワークやホストに使われていることがありますので、そのまま使うのは避けてください。

この実例の前提条件

- インターネットに接続されている
- 2つのネットワークが存在 (ドメインは `doc.com` と `sales.doc.com`)。DNS ゾーンも別々に管理
- `doc.com` ドメインおよび `doc.com` ゾーンが `sales.doc.com` サブドメインおよび `sales.doc.com` ゾーンの上のトップゾーンである

- 2つのネットワークはどちらも個別のネットワーク番号を持っている

表 13-4 ネットワークドメインとゾーン構成の例

名前/ゾーン	番号
doc.com	123.45.6
sales.doc.com	111.22.3

- 各ゾーンにマスターサーバーと副サーバーがあり、doc.comの主サーバーが sales.doc.comの副サーバーを兼ねている

表 13-5 ネットワーク DNS サーバーの例

ゾーン	ホスト名	役割	アドレス	標準名
doc.com	sirius	doc.comの主サーバー	123.45.6.1	dnsmaster
doc.com	deneb	doc.comの副サーバー	111.22.3.5	dnssecond
sales.doc.com	altair	sales.doc.comの主サーバー	111.22.3.4	dnssales
sales.doc.com	altair	sales.doc.comの副サーバー	123.45.6.1	dnsmaster

## サンプルブートファイル

次に示すのは、2つのネットワークで使われている3つのサーバーのブートファイルです。

例 13-9 dnsmastr サーバー用ブートファイルの例

```

; named.boot file on the dnsmastr (sirius)
;
; files required by in.named are located here

```

```

directory /var/named
; here are the names of the primary files
cache      .                named.ca
primary    doc.com          db.doc
primary    0.0.127.in-addr.arpa  named.local
primary    6.45.123.in-addr.arpa  doc.rev
;This system is also the secondary for the sales.doc.com domain
secondary  sales.doc.com     111.22.3.4  db.sales
secondary  3.22.111.in-addr.arpa  111.22.3.4  sales.rev

```

例 13-10 dnssales サーバー用ブートファイルの例

```

; named.boot file on the dnssales (altair)
;
; in.named is located here
directory /var/named
; here are the names of the primary files
cache      .                named.ca
primary    sales.doc.com     db.sales
primary    0.0.127.in-addr.arpa  db.127.0.0
primary    3.22.111.in-addr.arpa  db.192.168.8

```

例 13-11 dnssecond サーバー用ブートファイルの例

```

; named.boot file on the dnsecond (deneb)
directory /var/named
cache      .                named.ca
secondary  doc.com          123.45.6.1 doc.com
secondary  6.45.123.in-addr.arpa  123.45.6.1 doc.123.45.6

```

## サンプル resolv.conf ファイル

次に示すのは、2つのネットワークで使われている3つのサーバーの resolv.conf ファイルです。そのホストが in.named を稼動していない場合、そのローカルホストのアドレスをネームサーバーとして使うことはできません。

例 13-12 dnsmaster サーバー用 resolve.conf ファイルの例

```
;
; /etc/resolv.conf file for dnsmaster (sirius)
;
domain          doc.com
nameserver      0.0.0.0
nameserver      111.22.3.5
```

例 13-13 dnssales サーバー用 resolve.conf ファイル

```
;
; /etc/resolv.conf file for dnssales (altair)
;
domain          sales.doc.com
nameserver      111.22.3.4
nameserver      123.45.6.1
```

例 13-14 dnssecond サーバー用 resolve.conf ファイル

```
;
; /etc/resolv.conf for dnssecond
;
domain          doc.com
nameserver      111.22.3.5
nameserver      123.45.6.1
```

## サンプル named.local ファイル

次に示すのは、2つのネットワーク上の2つの主サーバーで使われている named.local ファイルです。どちらのサーバーも同じファイルを持っています。

例 13-15 主サーバー用 named.local ファイルの例

```
; SOA rec
0.0.127.in-addr.arpa. IN SOA siriusdoc.com. sysop.centauri.doc.com. (
                                19970331 ; serial number
                                10800    ; refresh every 3 hours
                                10800    ; retry every 3 hours
                                604800   ; expire after a week
                                86400 )   ; TTL of 1 day

; Name Servers
0.0.127.in-addr.arpa. IN NS  sirius.doc.com.
0.0.127.in_addr.arpa IN NS  dnssecond.doc.com
1 IN PTR localhost.
```

## サンプル hosts ファイル

次に示すのは、2つのネットワーク上の2つの主サーバーで使われている db.doc ファイルと db.sales ファイルです。

例 13-16 dnsmastr サーバー用 db.doc ファイルの例

```
; SOA rec
doc.com. IN SOA sirius.doc.com. sysop.centauri.doc.com. (
                                19970332 ; serial number
                                10800    ; refresh every 3 hours
                                10800    ; retry every 3 hours
                                604800   ; expire after a week
                                86400 )   ; TTL of 1 day

; Name Servers
doc.com.          IN NS  sirius.doc.com.
sales.doc.com.   IN NS  altair.sales.doc.com.

; Addresses
localhost        IN A  127.0.0.1
sirius           IN A  123.45.6.1
rigel            IN A  123.45.6.112
antares         IN A  123.45.6.90
polaris         IN A  123.45.6.101
procyon         IN A  123.45.6.79
tauceti         IN A  123.45.6.69
altair.sales.doc.com. IN A  111.22.3.4

; aliases
dnsmastr        IN CNAME sirius.doc.com.
dnssecond.doc.com IN CNAME deneb.doc.com
```

例 13-17 dnssales サーバー用 db.sales ファイルの例

```
; SOA rec
sales.doc.com. IN SOA altair.sales.doc.com. sysop.polaris.doc.com. (
                        19970332 ; serial number
                        10800    ; refresh every 3 hours
                        10800    ; retry every 3 hours
                        604800   ; expire after a week
                        86400 )   ; TTL of 1 day

; Name Servers
doc.com.          IN NS  sirius.doc.com.
sales.doc.com.   IN NS  altair.sales.doc.com.

; Addresses
altair            IN A  111.22.3.4
localhost        IN A  127.0.0.1
sirius.doc.com.  IN A  123.45.6.1
luna             IN A  192.168.8.22
phoebus         IN A  192.168.8.24
deimos          IN A  192.168.8.25
ganymede        IN A  192.168.8.27
europa          IN A  192.168.8.28
callisto        IN A  192.168.8.29

;
; aliases
dnssales.sales.doc.com IN CNAME altair.sales.doc.com
```

## サンプル hosts.rev ファイル

次に示すのは、2つのネットワーク上の2つの主サーバーで使われている hosts.rev ファイルです。

例 13-18 dnsmastr サーバー用 hosts.rev ファイルの例

```
; SOA rec
6.45.123.in-addr.arpa. IN SOA sirius.doc.com. sysop.centauri.doc.com. (
                        19970331 ; serial number
                        10800    ; refresh every 3 hours
                        10800    ; retry every 3 hours
                        604800   ; expire after a week
                        86400 )   ; TTL of 1 day

; Name Servers
6.45.123.in-addr.arpa. IN NS  sirius.doc.com.
;Pointer records for 123.45.6
1                        IN  PTR sirius.doc.com.
112                     IN  PTR rigel.doc.com.
90                      IN  PTR antares.doc.com.
101                     IN  PTR polaris.doc.com.
79                      IN  PTR procyon.doc.com.
```

(続く)

```
69                               IN PTR tauceti.doc.com.
```

### 例 13-19 dnssales サーバー用 hosts.rev ファイルの例

```
; SOA rec
3.22.111.in-addr.arpa. IN SOA altair.sales.doc.com. sysop.polaris.doc.com. (
                                19970331      ; serial number
                                10800         ; refresh every 3 hours
                                10800         ; retry every 3 hours
                                604800        ; expire after a week
                                86400 )       ; TTL of 1 day
; Name Servers
3.22.111.in-addr.arpa. IN NS  altair.sales.doc.com.
;Pointer records for 111.22.3
22          IN  PTR  luna
23          IN  PTR  deneb
24          IN  PTR  phoebus
25          IN  PTR  deimos
26          IN  PTR  altair
27          IN  PTR  ganymede
28          IN  PTR  europa
29          IN  PTR  callisto
```

## サンプル named.ca ファイル

次に示すのは、2つのネットワーク上の2つの副サーバーのどちらにも格納される named.ca ファイルです。

### 例 13-20 named.ca ファイルの例

```
;
; formerly NS1.ISI.EDU
.          3600000      NS      B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET. 3600000      A      128.9.0.107
;
; formerly C.PSI.NET
.          3600000      NS      C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET. 3600000      A      192.33.4.12
;
; formerly TERP.UMD.EDU
```

(続く)

```

.                3600000    NS    D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET. 3600000    A    128.8.10.90
;
; formerly NS.NASA.GOV
;.                3600000    NS    E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET. 3600000    A    192.203.230.10
;
; formerly NS.ISC.ORG
.                3600000    NS    F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET. 3600000    A    192.5.5.241
;
; formerly NS.NIC.DDN.MIL
.                3600000    NS    G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET. 3600000    A    192.112.36.4
;
; formerly AOS.ARL.ARMY.MIL
.                3600000    NS    H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET. 3600000    A    128.63.2.53
;
; formerly NIC.NORDU.NET
.                3600000    NS    I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET. 3600000    A    192.36.148.17
;
; temporarily housed at NSI (InterNIC)
.                3600000    NS    J.ROOT-SERVERS.NET.
J.ROOT-SERVERS.NET. 3600000    A    198.41.0.10
;
; temporarily housed at NSI (InterNIC)
.                3600000    NS    K.ROOT-SERVERS.NET.
K.ROOT-SERVERS.NET. 3600000    A    198.41.0.11
;
; temporarily housed at ISI (IANA)
.                3600000    NS    L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET. 3600000    A    198.32.64.12
;
; temporarily housed at ISI (IANA)
.                3600000    NS    M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET. 3600000    A    198.32.65.12
; End of File

```



# 索引

## A

adjunct ファイル 200  
aliases ファイル 60, 85, 199  
.attr マップ 232

## B

BIND (Berkeley Internet Name Domain) 31

## C

Can't find メッセージ 268  
can't initialize address メッセージ 268  
chkey 115  
compatserver.log ファイル 142  
ctx\_dir ディレクトリ 225, 227, 228, 233, 234  
.ctx マップ 232

## D

data.dict ファイル 38, 104, 141  
db.ADDR ファイル 31  
db.cache ファイル 31  
db.cache ファイル (DNS) 31  
db.domain ファイル 31  
dbm ファイル 176  
defaultdomain ファイル 195  
.dict ファイル 38  
DIR ディレクトリ 199  
dir ディレクトリオブジェクト 111  
.dir ファイル 176  
DNS 30, 240  
    Can't find メッセージ 268  
    can't initialize address メッセージ 268

NIS 213  
non-authoritative メッセージ 269  
Non-existent domain メッセージ 268  
resolv.conf ファイル 242  
RFC1535 240  
Solaris 上の 240  
switch ファイル 28  
偽名ログイン 240  
逆クエリー 240  
キャッシュオンリーサーバー 256, 259  
クライアントの設定 240, 241  
サーバー機能の指定 255  
サーバーの初期設定 267  
サーバーの設定 245  
サーバーの設定 (非インターネット)  
    ト) 272  
サブドメインの設定 269  
サブドメインの設定 (単一ゾーン) 270  
サブドメインの設定 (複数ゾーン) 271  
資源レコード 260  
主マスターサーバー 255, 256  
使用できるようにする方法 28  
スイッチファイル 243  
設定の確認 267  
ゾーン逆変換 257  
データファイルの設定 260  
テストプログラム 240  
デフォルトドメイン名 240, 250, 251  
ドメイン名 250  
ドメイン名の末尾のドット 251  
バージョン 240  
バックアップファイル 258  
ファイル名 247

副マスターサーバー 255, 258  
ユーティリティスクリプト 240  
リゾルバ 241  
ルートドメインサーバー 256  
例 273  
ローカルループバック 257  
domainname 100, 128, 129, 195, 211, 214  
最後のドット 100  
DOM変数 211, 212

## E

/etc/.rootkey ファイル 109, 126  
/etc/defaultdomain ファイル 195  
/etc/hosts 58, 69, 80, 214, 229, 234  
/etc/hosts ファイル 225, 232, 233  
/etc/init.d/rpc 74, 75, 78, 140  
/etc/mail/aliases ファイル 199  
/etc/mail ディレクトリ 199  
/etc/named.boot ファイル 253  
/etc/nodename ファイル 195  
/etc/nsswitch.conf 26, 27, 52, 100, 101  
/etc/nsswitch.files 22  
/etc/nsswitch.files 22  
/etc/nsswitch.nis 22  
/etc/nsswitch.nisplus 22  
/etc/passwd 52  
/etc/passwd ファイル 31, 99, 229, 232, 233  
nisaddent コマンド 174  
/etc/resolv.conf 99  
/etc/resolv.conf ファイル 213, 242, 243  
/etc/shadow ファイル  
nisaddent コマンド 174  
/etc/users 225

## F

fncreate 226, 227, 229, 231, 232  
FNS 224  
NIS+ サービスの準備 226  
NIS+ の下での作成 230  
NIS の下での作成 231  
NIS の準備 228  
グローバルな作成 229  
条件 224  
設定の準備 226  
ファイルネーミングの下での FNS コンテ  
キストの作成 232

ファイルを使用した名前空間の準備 229  
複製 233  
複製 (NIS) 234  
複製 (NIS+) 233  
複製 (ファイルを使用) 235  
fns\_マップ 232  
fn ファイル 225

## G

groups\_dir  
nissetupでの作成 107, 158  
groups\_dir ディレクトリ 227

## H

/hostname ディレクトリ 141  
hosts.byname マップ 229  
hosts.org\_dir テーブル 227, 231  
hosts.rev ファイル 246, 250, 265, 278  
サブドメイン 271  
サブドメイン (単一ゾーン) 270  
設定 265  
複数のゾーン 271  
例 265, 278, 279  
hosts (DNSファイル) 249  
named.boot ファイルでの指定 256  
いくつかのゾーン 249  
サブドメイン 271  
設定 264  
複数のゾーン 271  
例 264, 277, 278  
hosts ファイル 60, 69, 85, 214, 225, 229, 232 -  
234, 246, 257, 265  
hostsファイル 69

## I

in.named 247  
in.named ファイル 259, 260  
\$INCLUDE ファイル 250  
IPv6  
使用できるようにする 32  
スイッチファイル 32

## K

keylogin 127

keyserv 101

## L

LOCALDOMAIN 251

.log ファイル 38, 175

## M

make 183, 211

makedbm 180, 209, 211

Makefile ファイル 196, 199, 200, 209, 211

例 200

mapname.dir ファイル 209

mapname.pag ファイル 209

multihomed NIS+ 複製サーバー

/etc/hosts 80

multihomed NIS+ ルートマスターサーバー

/etc/hosts 58

## N

named.boot ファイル 247, 253

キャッシュオンリーサーバー 259

サーバー機能 255

主マスターサーバー 256

設定 (サーバー) 252

ゾーン逆変換 257

ディレクトリの要素 255

バックアップファイル 258

非インターネット 272 - 273

副マスターサーバー 258

例 254, 275

ローカルループバック 257

named.ca ファイル 246, 249, 256, 263, 273

インターネットのバージョン 261

非インターネット 272

非インターネットバージョン 263

ルートサーバーの設定 261

例 277, 279

例 (インターネットバージョン) 262

例 (非インターネットバージョン) 263

named.local ファイル 246, 250, 257, 267

設定 266

例 266

named.root ファイル 261

ndbm 199

+ netgroup 31

Network Informatim Serice 31

Network Informatim Serice Plus 31

NIS 193

DNS 213

FNS の複製 234

FNS 用 NIS の準備 228

Makefile の準備 199

NIS+ から情報を転送する 182

NIS デーモンの起動 212

NIS の下での FNS の作成 231

ypinit 210

開始 212

クライアントの設定 216

コマンド行からのスタート 212

自動スタート 212

マスターサーバーの設定 210

スレーブサーバーでの開始 216

スレーブサーバーの設定 213, 214

設定手順 195

設定の計画 194

設定のための準備 194, 196

ソースファイル 196, 198

停止 213

デフォルトスイッチファイル 23

ドメイン名 194

パスワードデータ 196, 197

複数のドメイン 211

マップ 217

NIS+ 35

FNS の下での作成 230

FNS の複製 233

FNS 用の NIS+ サービスの準備 226

NIS に情報を転送する 182

構成 46

構成方法 39

サブネット 143

セキュリティ 36

設計 36

設定スクリプト 31, 41, 49, 51

設定の前に 36, 37

テーブル 31

デフォルトスイッチファイル 22

ドメイン名 38

名前空間サンプル 46, 48

パフォーマンス 142

ホスト名 38

リモートサイト 143

ルートサーバー 38

- NIS+ グループ
  - 管理グループ 115
- NIS+ テーブル
  - NIS マップからの生成 175, 177
  - オプションの追加 166
  - スワップ空間 116, 162
  - 生成のオプション 166
  - 生成方法 165
  - 設定 (コマンド) 165
  - 設定 (スクリプト) 59, 61
  - 置換 166
  - ファイルから生成する 167, 169
  - マージ 167
- NIS+ テーブルの生成 31
- NIS+ デーモン 31
- nisaddcred 108, 124, 161
  - DES 資格の追加 114
  - LOCAL 資格の追加 114
  - 管理者の資格の追加 115
- nisaddent 115, 168, 171 - 173, 178 - 180, 183
- nisbackup 147
- nis\_cachemgr 31, 140
- niscat 67, 111, 112, 122, 146, 172
- nischmod 160, 186
- nisclient 68, 71, 90, 91
- nisclient スクリプト 41, 42, 67
  - DNS 43
  - multihomed NIS+ 複製サーバー 80
  - クライアントの追加 70
  - サブドメインクライアント 89 - 91
  - 前提条件 68, 71
  - ユーザーの初期設定 71, 72, 91, 92
- NIS\_COLD\_START ファイル 113, 131, 133, 134
- NIS\_GROUP
  - 設定 157
- nisgrpadm 43, 65, 110, 116, 159, 160, 162
- nisinit 103, 104, 131, 133, 135
- nisls 228
- nismkdir 103, 146, 157, 228, 234
- nisping 147, 148, 150, 151, 181, 234
  - チェックポイント 175, 181
  - チェックポイントの設定 228, 231
  - テーブルの作成 174
  - 複製サーバーの設定 148, 149
- nispopulate スクリプト 41, 42, 59, 61
  - multihomed NIS+ 複製サーバー 80
- multihomed NIS+ ルートマスターサーバー 58
  - 管理グループ 65
  - セキュリティ 60, 85
  - 前提条件 59
  - ドメインの追加 84 - 87
  - 入力ファイル 60
  - 必要なスワップ領域 67
  - ファイルから生成 87
  - ファイルの準備 60, 85
  - マップから生成 87
- nisrestore 148
  - 複製サーバーの設定 146
- nisserver 81, 88
- nisserver スクリプト 41, 42, 52, 76
  - DNS と NIS+ サーバー 74
  - multihomed NIS+ 複製サーバー 79, 80
  - multihomed NIS+ ルートマスターサーバー 57
  - NIS+ サーバー 75
  - NIS+ 複製サーバー 72, 74
  - NIS+ ルート複製サーバー 75 - 77
  - サブドメイン 81, 82
  - サブドメイン複製サーバー 88
  - 前提条件 52
  - デフォルトのセキュリティレベル 54
  - ドメインの追加 84
  - ルートサーバー 51, 53, 56
- nissetup 55, 96
  - automounter テーブル 173
  - groups\_dir の作成 107, 158
  - org\_dir の作成 107, 158
  - passwd テーブル 174
  - テーブルの作成 107, 158
- NIS\_SHARED\_DIRCACHE ファイル 31, 112
- nistbladm 186
  - 管理者の資格の追加 115
- NIS 互換モード 54
  - 設定 (コマンド) 96, 154
- nodename ファイル 195
- non-authoritative メッセージ 269
- Non-existent domain メッセージ 268
- nscd 125
- nscd デーモン 27, 29, 101
- nslookup 268, 269
- nsswitch.conf で DNS を使用できるようにする方法 28

nsswitch.confで IPv6 を使用できるようにする 32  
nsswitch.conf ファイル 21, 26, 27, 29, 30, 52, 100, 101, 121, 194, 195, 216, 244  
DNS 用の修正 243  
NIS+ 用デフォルトファイル 22  
NIS 用デフォルトファイル 23  
インストール 25  
デフォルトのファイル 22  
テンプレート 22  
ファイルの選択 25  
ローカルファイル用デフォルトファイル 24  
nsswitch.confファイル  
ローカルファイル用デフォルトファイル 25  
nsswitch.files ファイル 22  
nsswitch.nisplus ファイル 22  
nsswitch.nis ファイル 22

## O

org\_dir  
nissetupでの作成 107, 158  
org\_dir ディレクトリ 225, 227  
org\_ ディレクトリオブジェクト 111

## P

.pag ファイル 176  
passwd.adjunct ファイル 209  
passwd.org\_dir テーブル 231  
passwd テーブル 31  
passwd ファイル 31, 52, 60, 85, 99, 196, 229, 232, 233  
passwd マップ 31, 197  
printer.conf.byname マップ 229  
publickey ファイル 173, 180  
publickey マップ 179  
PWDIR 197  
/PWDIR/shadow ファイル 209  
/PWDR/security/passwd.adjunct 209

## R

rcp 214  
resolv.conf ファイル 99, 213, 242, 243, 251

設定 251  
デフォルトドメイン名 250  
例 242, 251, 274 - 276  
RFC1535 240  
root.cache ファイル (DNS) 31  
.rootkey ファイル 126  
root.object ファイル 103  
root ディレクトリオブジェクト 111  
rpc.nisd 43, 54, 73, 76, 78, 82, 88, 104, 140, 147, 148  
実行するための前提条件 73  
rpc.nisd\_resolv デーモン 138  
rpc.nisd デーモン 43, 57, 105  
起動 113  
rpc.nisd ファイル 138  
rpc ファイル 74, 75, 78, 140  
EMULYP="-Y -B" オプション 75  
EMULYP="-Y" オプション 74, 78

## S

Secure RPC netname 114  
shadow ファイル 31, 60, 85, 196, 209

## T

TMPDIR 62  
/tmp ディレクトリ 134  
trans.log ファイル 38, 104, 140 - 142

## U

uname 195  
user.byname マップ 229  
users ファイル 225  
/usr/lib/netsvc/yp/ypstart 212  
/usr/lib/netsvc/yp/ypstart スクリプト 197, 213  
/usr/lib/nis/nisupdkeys 111  
/usr/lib/nis ディレクトリ 167, 170, 175, 177

## V

/var/fn ディレクトリ 229, 232  
/var/fn ファイル 225  
/var/named/hosts.rev 250  
/var/named/named.ca ファイル 249  
/var/nis/data.dict 38, 104

/var/nis/data ディレクトリ 38, 103, 104,  
140 - 142  
/var/nis/root.object 103  
/var/nis/trans.log 38, 104, 140, 142  
/var/nis ディレクトリ 38, 74, 102, 104, 125,  
141  
/var/yp/Makefile 211  
/var/yp ディレクトリ 176, 196, 199, 209, 214

## W

WARNING: password differs from login  
password 109

## X

XFN 31  
xfr ファイル 167

## Y

yp 216, 235  
ypbind 212, 216, 217, 235  
ypcat 30  
ypinit 199, 210 - 212, 214 - 216, 234  
ypserv 212, 213, 216  
ypstart 212  
ypstop 213

## あ

アクセス権 43

## い

インストール ション 31  
インターネット  
DNS 246  
IP 登録 246  
named.ca ファイル 261  
ルートドメイン 52

## か

管理グループ  
作成 109, 156

## き

偽名ログイン 240  
逆クエリー 240  
キャッシュファイル (DNS) 31  
キャッシュマネージャ 112  
起動 112

## く

クライアント  
NIS+ 初期設定 (スクリプト) 72  
NIS+ 設定 (コマンド) 119, 123  
NIS+ 設定 (スクリプト) 67, 68, 70, 89 - 91  
NIS+ の初期設定 (コマンド) 126, 129  
NIS+ ユーザーの初期設定 (スクリ  
プト) 71, 91, 92  
NIS の設定 216  
コールドスタートによる初期化  
(NIS+) 133, 134  
ブロードキャストによる初期設定  
(NIS+) 130, 131  
ホスト名による初期化 (NIS+) 131, 132  
グループ 29, 31  
NIS+ グループ 31

## こ

公開鍵  
更新 111  
構成 31  
コールドスタートによる初期化 (NIS+) 133,  
134

## さ

サーバー 31  
NIS+ 設定 (スクリプト) 72, 74, 75  
NIS+ 複製の設定 (スクリプト) 72  
NIS+ ルート複製サーバーの設定 (スクリ  
プト) 75 - 77  
NIS スレーブの設定 213, 214  
NIS の準備 196  
multihomed NIS+ 複製サーバーの設定  
(スクリプト) 79  
multihomed NIS+ ルートマスターの設定  
(スクリプト) 57

- サブドメイン複製サーバーの設定 (スクリプト) 88
- サブネット 75
- スレーブサーバーでの NIS の開始 216
- 設定 (NIS+ コマンド) 137, 139
- 複製サーバーの設定と nisrestore 146
- 複製の設定 (NIS+ コマンド) 142, 144, 145
- 複製の設定と nisping 148, 149
- リモートサイト 76
- ルートサーバー 31
- サブドメイン
  - 設定 (スクリプト) 81, 82
- サブネット
  - NIS+ 143
  - NIS+ ルート複製サーバー 75

## し

- 資格
  - LOCAL 資格の追加 113
  - WARNING: password differs from login password 109
  - 管理者の資格の追加 115
  - クライアント 124
  - 作成 108, 160
- 資源レコード (DNS) 260
- 承認 36

## す

- スイッチ 31
- スイッチファイル 29, 31
- スクリプト (NIS+) 31
- スワップ空間
  - NIS+ テーブル 116, 162

## せ

- セキュリティ 31
  - NIS 196, 197
  - passwd column へのアクセスの制限 183, 185
  - S O フラグ 105
  - 資格 31
  - 承認 36
  - デフォルトのレベル (スクリプト) 54
  - 認証 36
  - レベル 43

## 設定

- DNS サーバーの初期設定 267
- DNS サーバー (非インターネット) 272
- DNS サブドメイン 269
- DNS サブドメイン (単一ゾーン) 270
- DNS サブドメイン (複数ゾーン) 271
- DNS データファイル 260
- DNS と NIS+ サーバー 74
- DNS の確認 267
- DNS のクライアント 240, 241
- DNS のサーバー 245
- DNS のファイル名 247
- DNS の例 273
- FNS 224
- FNS サービスの複製 233
- FNS 条件 224
- FNS のグローバルな作成 229
- FNS の準備 226
- FNS ファイルを使用した名前空間の準備 229
- FNS 用 NIS サービスの準備 228
- FNS 用の NIS+ サービスの準備 226
- multihomed NIS+ 複製サーバー (スクリプト) 79
- multihomed NIS+ ルートマスターサーバー (スクリプト) 57
- named.boot ファイル 247
- NIS 193, 195
- NIS+ 46
- NIS+ クライアント (コマンド) 119, 123
- NIS+ クライアント (スクリプト) 67, 68, 70
- NIS+ クライアントの初期設定 126, 129
- NIS+ サーバー (スクリプト) 75
- NIS+ スクリプト 41
- NIS+ 設定スクリプト 49, 51
- NIS+ テーブル (コマンド) 165
- NIS+ テーブル (スクリプト) 59, 61
- NIS+ の下での FNS コンテキストの作成 230
- NIS+ の準備 36, 37
- NIS+ 複製サーバー (スクリプト) 72, 74, 88
- NIS+ ユーザーの初期設定 (スクリプト) 71, 72, 91, 92
- NIS+ ルート複製サーバー (スクリプト) 75 - 77

- NIS makefile 199
- NIS クライアント 216
- NIS スレーブサーバー 213
- NIS 設定の開始 212
- NIS 設定の計画 194
- NIS 設定のための準備 194, 196
- NIS の下での FNS コンテキストの作成 231
- NIS のスレーブサーバー 214
- NIS のマスターサーバー 210
- PATH 変数 53
- resolv.conf ファイル 242
- 管理グループ (スクリプト) 65
- コールドスタートによる初期化 (NIS+) 133, 134
- サーバー (NIS+ コマンド) 137, 139
- サブドメインクライアント (スクリプト) 89 - 91
- サブドメインの設定 (スクリプト) 81, 82
- スイッチファイル 22
- スワップ領域 (スクリプト) 67
- デフォルトのセキュリティレベル 54
- ドメイン (NIS+ コマンド) 153, 156
- ドメインの追加 84 - 87
- ドメイン名 60
- ファイルネーミングの下での FNS コンテキストの作成 232
- 複数の NIS ドメイン 211
- 複製サーバー (NIS+ コマンド) 142, 144, 145
- 複製の設定と nisping 148, 149
- 複製の設定と nisrestore 146
- ブロードキャストによる初期設定 (NIS+) 130, 131
- ホスト名 60
- ホスト名による初期化 (NIS+) 131, 132
- リゾルバ 241
- ルートサーバー (スクリプト) 51 - 53, 56
- ルートドメイン (NIS+ コマンド) 95, 96, 99
- ルートドメイン (NIS コマンド) 95
- 設定スクリプト (NIS+) 31

## そ

- ゾーン (DNS) 31

## ち

- チェックポイント 31
- 置換 (NIS+ テーブル) 166

## つ

- 追加 (NIS+ テーブル) 166

## て

### デーモン

- NIS デーモンの起動 212
- nscd 27
- nscd デーモン 101
- rpc.nisd\_resolv デーモン 138
- rpc.nisd デーモン 43, 57, 105, 113
- テストプログラム 240
- デフォルトドメイン名 240

## と

- 独自のテーブル 43

### ドメイン

- NIS 194
- 管理グループ 156
- 設定 (NIS+ コマンド) 153, 156
- 設定 (スクリプト) 84 - 87
- 名前 38, 53, 60
- 名前 (DNS) 250
- 複数の NIS ドメイン 211
- ホストのドメインの変更 127, 128
- ルート 31
- ルートドメインの設定 51
- ドメイン名システム 31

## な

### 名前空間

- FNS のグローバルな作成 229
- FNS 用の名前空間の準備 226
- NIS+ の準備 37

## に

- 認証 36



## ね

### ネーミング

相互運用性 21

調整 21

ネームサービス 31

ネームサービススイッチ 31

## は

パスワードデータ 29, 31

NIS 196, 197

passwd column へのアクセスの制限 183, 185

バックアップファイル (DNS) 258

パフォーマンス

NIS+ 142

## ひ

表記上の規則

## ふ

ブートファイル (DNS) 31

フェデレーテッド・ネーミング・サービス 31

ブロードキャストによる初期設定 (NIS+) 130, 131

## ほ

ホスト (データ)

名前 53

名前の中のドット 38

ホスト (マシン)

ドメインの変更 127, 128

名前 38, 53

ホスト名による初期設定 (NIS+) 131, 132

## ま

マージ (NIS+ テーブル) 167

マシン 31

マップ 31

## ゆ

ユーザー

初期設定 (NIS+) 71, 91, 92

ユーティリティスクリプト 240

## り

リゾルバ 241

リモートサイト

NIS+ 143

NIS+ ルート複製サーバー 76

## る

ルートサーバー 31

設定 (NIS+ コマンド) 95, 96, 99

設定 (スクリプト) 51 - 53, 56

ルートドメイン

NIS 互換モード 96, 154

インターネットルートドメイン 52

設定 (NIS+ コマンド) 95, 96, 99

設定 (スクリプト) 52, 53

名前 52

## ろ

ローカルファイル 31

デフォルトスイッチファイル 24

ローカルループバック 257

## わ

ワークステーション 31

## ロ

ローカルファイル

デフォルトスイッチファイル 25