COMPAQ

WHITE PAPER

technical level

BASIC ADVANCED

1999年8月

作成： Micky BAlladelli

Senior Consultant

Compaq Services
Applied Microsoft Technologies Group

Compaq Computer Corporation

コンパック コンピュータ株式会社

Active Directoryの複製

Windows 2000の最も重要な機能の1つが、Active Directoryです。Active Directoryは、拡張性のあるプログラム可能なディレクトリ サービスであり、オペレーティング システムに組み込まれています。

新しいActive Directory対応アプリケーションは、オブジェクトと構成 データをディレクトリに格納でき、Windows 2000で標準化された複製メカニズムとトポロジを使って、この情報をエンタープライズの全サーバへ複製します。最初に登場するActive Directory対応アプリケーションは、コードネーム“Platinum”と呼ばれるMicrosoft Exchange Serverの次のリリースです。Platinumでは、アプリケーションは、Active Directoryのメッセージング ユーザ、ルーティング トポロジ、アドミニストレーション、サーバなどに関する情報を格納することによって、複製を活用します。Windows 2000とアプリケーションの導入を成功させるには、管理者やシステム設計者がActive Directoryの複製を理解する必要があります。

本書では、Windows 2000のActive Directoryの複製機能を詳細に説明し、導入過程で考慮すべき設計上のオプションや機能を紹介します。

注：この資料は米国の資料を翻訳したものです。
そのため、日本語環境とは異なる点もございます。

ご注意

本書の内容は、将来、予告なしに変更されることがあります。

コンパック コンピュータ株式会社は、コンパック製品保証書で保証する内容以外には、一切の保証はいたしません。本書の内容が、保証期間の延長や保証内容の拡張を意味することは一切ありません。

本書は、製品について、検査の結果を保証するものではありません。また、ここで検査または記述された構成が唯一の解決策とは限りません。この検査は、製品の品質や正当性を確定するものでも、国またはその機関が定める必要条件に従っていることを保証するものでもありません。

本書に掲載されている製品は、それぞれ各社の商標または登録商標として承認されています。

Compaq、Contura、Deskpro、Fastart、Compaq Insight Manager、LTE、PageMarq、Systempro、Systempro/LT、ProLiant、TwinTray、LicensePaq、QVision、SLT、ProLinea、SmartStart、NetFlex、DirectPlus、QuickFind、RemotePaq、BackPaq、TechPaq、SpeedPaq、QuickBack、PaqFaxは、米国Compaq Computer Corporationの登録商標です。

Aero、Concerto、QuickChoice、ProSignia、Systempro/XL、Net1、SilentCool、LTE Elite、Presario、SmartStation、MiniStation、Vocalyst、PageMate、SoftPaq、FirstPaq、SolutionPaq、EasyPoint、EZ Help、MaxLight、MultiLock、QuickBlank、QuickLock、TriFlex Architecture and UltraView、CompaqCare、Innovateロゴは、Compaq Computer Corporationの商標またはサービス マークです。

本書に掲載されている製品は、それぞれ各社の商標または登録商標として承認されています。

Microsoft、Windows、Windows NT、Windows NT Advanced Server、SQL Server for Windows NTは、Microsoft Corporationの商標または登録商標です。

1999年8月1日

第1版

ご意見は、直接次のアドレスへお送りください。micky.balladelli@compaq.com.

本書のレビューをサポートしていただいたTony Redmond氏に感謝します。

また、空港のラウンジで、多くの時間を使って技術的内容についての理解を助けていただいたDung Hoang Khac氏にも感謝します。

本書のベースとなったWindows 2000 Academyの資料のレビューをしていただいたMicrosoftのAndreas Luther氏にも感謝します。

目次

はじめに

本書では、Active Directoryの複製について、様々な角度から説明します。Windows 2000で使用する複製テクノロジに関する用語を紹介し、Windows 2000サーバでデータを複製する方法を説明します。また、複製トポロジがWindows 2000サイト内やサイト間で作成される方法についても説明します。最後に、複製トポロジの設計やヒントを紹介します。

Windows 2000の複製は、Windows NTから大幅に変更されています。Windows NTでは、複製はプライマリ ドメイン コントローラ（PDC）から、バックアップ ドメイン コントローラ（BDC）として機能するドメイン内の他のコンピュータへ、情報を単にコピーするだけでした。Windows 2000では、マルチ マスタのフル メッシュ レプリケーション メカニズムを採用し、組織内にある任意のコントローラを変更することが可能になり、変更内容を他のすべてのドメイン コントローラへコピーすることができます。複製は、組織がどのような方法で分散されている場合でも、安全で信頼性の高い方法でデータをコピーし、ドメイン コントローラを最短時間で更新するメカニズムです。

Windows 2000の設計目的の1つが、大規模エンタープライズのニーズへの対応です。Windows NT複製は、非常に単純なものであり、スケーラビリティの問題には対応できませんでしたが、Windows 2000はこれに対応しました。Active Directoryの拡張性と、サポート可能なオブジェクトの数は、Windows NTで採用していたシングル マスタ レプリケーション モデルを大きく上回ります。ドメインをグループ化してグローバル ネームスペースを構成する環境で、多数のオブジェクトをサポートできるように、新しい複製ルールを定義する必要があります。

本書は、Windows 2000の基本概念を理解している方を対象としています。基本概念については、http://www.compaq.com/activeanswersに掲載されているホワイト ペーパ『*Active Directory – A technical overview*』を参照してください。

複製の基本知識

複製とは

Windows 2000ドメイン コントローラ（DC）は、ドメインに属するすべてのオブジェクトの複製を格納し、このオブジェクトに対してリード/ライト アクセス権限を持っています。アドミニストレータは、ドメイン内の任意のDCを使って、管理作業を行います。このような管理作業は、オブジェクトのステートや値を変更するので、他のDCにレプリケートする必要があります。複製は、オブジェクト更新をDC間で伝播するプロセスです。

複製の目的は、すべてのコントローラを更新し、Active Directoryデータベースのコピーを一貫性のある状態にすることにあります。ディレクトリ同期プロシージャなど、大量のレコードの追加や更新を短時間で行うプログラムがディレクトリを変更する場合は、大量の複製オペレーションをトリガする可能性はありますが、オブジェクトが変更されたからといって、すぐに複製がトリガされるわけではありません。複製がトリガされるのは、ある程度時間が経過してからであり、すべての変更内容を収集し、この内容を他のコントローラへ提示します。したがって、他のコントローラからの複製が途中になっている場合もあるので、いずれのコントローラでも、Active Directoryは常に緩やかな一貫性が維持された状態にあるといえます。最終的には、変更内容が到着し、DCは他のコントローラと同期します。ただし、アドミニストレータが管理作業を行う場合、ユーザや他のアドミニストレータが他のDCで同じ内容の操作を行っているかどうかはわかりません。このような場合、複製の競合が発生するので、解決する必要があります。競合の解決方法については、後の節で説明します。

ドメイン コントローラ

Windows NT 4.0は、*シングル マスタ レプリケーション* モデルを採用しています。

このモデルでは、プライマリ ドメイン コントローラ（PDC）だけがデータベースへのリード/ライト アクセスが可能なので、操作はPDC上でのみ実行されます。PDCで実行された内容は、バックアップ ドメイン コントローラ（BDC）へ複製されます。BDCは、読み出し専用のデータベース コピーを持ち、ドメイン情報の複製をユーザに提供します。ユーザは遠くにあるDCへ低速なリンクを介して接続する必要がなくなるので、認証が速くなります。

Windows NT 4.0では、複製は大きな単位で実行されます。ユーザやアドミニストレータがパスワードを変更すると、属性を含むユーザ オブジェクト全体を、ドメイン内のすべてのBDCへ複製しなくてはなりません。複製トポロジは単純で、同じドメイン内のすべてのBDCとPDCのリンクで構成されています。

Windows 2000インフラストラクチャでは、各ドメイン コントローラ（DC）がActive Directoryデータベースのコピーを持っており、ドメイン内の全オブジェクトへのリード/ライト アクセス権を持ちます。DC上で操作が行われた場合、これはドメイン内にある他のすべてのDCへ複製されます。ドメインがリンクされてフォレストを構成している場合、各ドメインからのデータを他のドメインへ複製する必要があり、これによってフォレストのビューを形成します。Windows 2000では、*マルチ マスタ レプリケーション* モデルによって、ドメイン内やドメイン間のデータ コピーをサポートします。また、実際に変更されたデータのみを複製するので、Active Directory複製は最適化されています。つまり、ユーザがパスワードを更新した場合、オブジェクト全体ではなく、更新されたパスワードだけが他のコントローラへ複製されます。

ドメイン コントローラは、複製プロセスの開始と実行を行います。各DCは、他のDCの複製パートナーとなります。複製は、必ずDC間で行われます。メンバ サーバは、複製プロセスで役割を果たします。

内部的には、DCは、他のDC、つまり複製パートナーを、GUID（Globally Unique Identifier）を使って参照します。GUIDは、オブジェクトの認識に使用する一意の値です。GUIDは、アプリケーション インタフェースを一意に認識することを目的に、MicrosoftのCOMが採用したものです。GUIDは、定義上、一意の数値です。したがって、GUIDは、同じシステム上で同時に生成されても、必ず一意になるようなアルゴリズムを使って生成されます。

Windows 2000では、次の2つのGUIDをDCの参照に使用します。

- サーバGUID：複製パートナーはこのGUIDを使ってDCを参照します。各DCのGUIDは、DNSデータベースに保存されているので、これによって複製パートナーは使用可能なDCを参照します。
- データベースGUID：まず、データベースGUIDはサーバGUIDと同様に、複製コールの間、データベースの認識に使用されます。リストアでは、他のDCがデータベースGUIDを使用し、データベースがリストアされ、DCのステートが変更されていることを認識します。

システムの名前が変更された場合でも、GUIDの値は変わらないので、名前よりも信頼性が高くなります。したがって、名前の変更による影響を受けません。したがって、Active Directoryの一部からオブジェクトを移動する場合、オブジェクトを削除してから再作成する必要がなくなります。

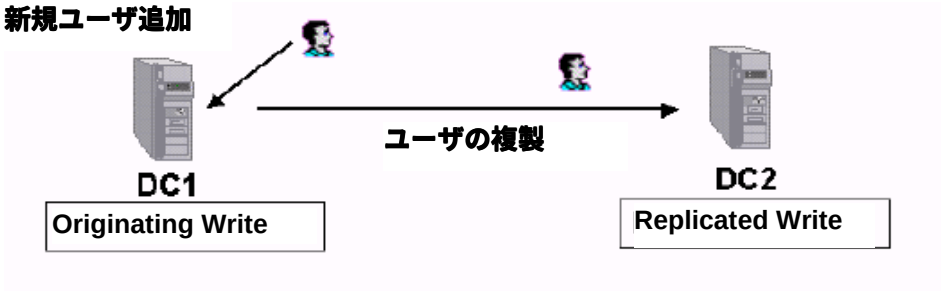


図1：Originating Write とReplicated Write

複製

DC間の複製をトリガするオペレーションには、様々なものがあります。アクセス許可に従って、アドミニストレータやユーザは、Active Directoryに格納されているオブジェクトの操作を行います。

次のようなオペレーションが複製をトリガします。

- オブジェクトの生成 – 新規ユーザの追加など、データベースに新しいオブジェクトを作成します。
- オブジェクトの操作 – ユーザ パスワードの変更など、オブジェクト属性を変更します。
- オブジェクトの移動 – オブジェクトのコンテナを変更します。Active Directory内でオブジェクトを整理するために使用するのがOU（組織単位）であり、一般的に会社内の部門と同じ名称を使い、“Sales”や“Marketing”などがあります。“Sales”というOUから“Marketing” OUへの移動がオブジェクトの移動の例であり、移動の結果、企業内で再組織化が行われます。
- オブジェクトの削除 – データベースからオブジェクトを削除します。オブジェクトは、実際に削除されるのではなく、tombstoneへ転送されます。

tombstoneは、オブジェクトの状態であり、オブジェクトは削除済みでアクセス不能であることを示すフラグです。たとえば、データベースに多数のオブジェクトを作成してからオブジェクトを削除しても、データベースのサイズは小さくなりません。これは、データベース内のオブジェクトはすべてtombstoneへ転送されるからです。tombstoneは、元のオブジェクトと同じサイズです。tombstoneには、ドメイン全体に対して、オブジェクトが削除されたことを通知する役割があります。tombstoneは60日間保存され、60日経過すると、オブジェクトはデータベースから削除されます。ただし、Active Directoryの次の属性で、tombstoneの保存期間を変更できます。

cn=DirectoryServices,cn=WindowsNT,cn=Services,cn=configuration,
dc=ドメインName with the name tombstonelifetime .

更新タイプ

書き込みオペレーションは、この操作がDC上でローカルに実行されたのか、リモートDCで実行されてからローカルに複製されたのかによって異なります。図1に示すように、更新タイプには次の2つがあります。

- *Originating Write*は、同じシステム上で最初に実行された操作を指します。これは、DC上で実行される4種類の操作（追加、変更、移動、削除）の1つです。この操作は、現在のDC上で開始された操作であるために、この名称が付いています。たとえば、現在のDCにユーザを作成した場合、これはOriginating Writeとなります。

- *Replicated Write*は、他のシステム上で実行された操作がローカルに複製されたものです。他のDCが開始した操作から発行されたデータベース変更が、現在のDCに複製されたものです。DC1上にユーザを作成し、これがDC2へ複製された場合、DC2データベースの変更をReplicated Writeと呼びます。

名前付けコンテキスト

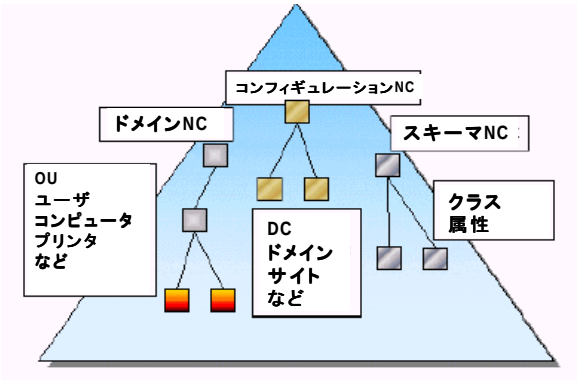
NC（名前付けコンテキスト）は、Active Directoryに格納されているオブジェクトのツリーです。NCには、次の3つがあります。

- 構成 NC – ドメイン、DC、その他構成タイプ オブジェクトなど、Active Directoryの構造を示す全オブジェクトを含みます。
- スキーマ NC – オブジェクトを定義するクラスや属性、Active Directoryに格納されている属性などをすべて含みます。
- ドメイン NC – ユーザ、グループ、OU、コンピュータなど、Active Directory内のその他オブジェクトをすべて含みます。

ドメインは、Active Directoryのパーティションの役割をしますが、複製の境界にはなりません。複製の境界は、名前付けコンテキストが定義し、複製の範囲を定義します。複製の境界は、DC上で実行するマネジメント オペレーションを同じドメイン上のDCへ複製するか、フォレスト内の他のDCにまで複製するか、を示します。つまり、名前付けコンテキストは、Active Directoryの変更を組織内のどこまで複製するのかを定義します。

名前付けコンテキストは、Windows 2000で新しく採用された概念です。Windows NTドメインでの複製は、ドメインの範囲でしかなく、このような概念は存在しませんでした。

図2：名前付けコンテキスト



NCの複製の2つのスコープ：フォレスト ワイドとドメイン ワイド

構成とスキーマは、フォレスト全体で1つしかなく、ユニットを構成します。つまり、NCは、フォレストを構成する各ドメイン内の全DCに対して複製する必要があります。したがって、構成 NCとスキーマ NCは、フォレスト ワイドのスコープということになります。

ドメイン オブジェクトは、現在所属しているドメイン内でのみ複製を行います。ドメインNCは、ドメイン ワイドのスコープです。つまり、ドメインに格納されている情報は、1つのNCに限られ、ドメイン内で複製されます。ただし、これには例外があり、ドメインNCからGCへの複製は、フォレスト ワイドのスコープで行われます。GCは複数のドメインで使用するので、ドメインNCにはフォレスト ワイドのスコープが必要となるためです。ドメインNCと他のドメイン内にあるGCとを結ぶ複製トポロジについては、後で説明します。

名前付けコンテキストは、Active Directoryの全オブジェクトを含み、Active Directory内にあります。各NCの親はRootDSEです。RootDSEは、仮想コンテナです。これは、ドメイン内の全オブジェクトを含みます。また、Active Directoryの中でも、親オブジェクトを持たない唯一のオブジェクトです。RootDSEオブジェクトは、複製されることもスペースを持つこともありません。つまり、ドメイン内に格納されているNCの仮想的な親です。RootDSEは、Active Directoryの階層構造の最も上に位置します。

USN（Update Sequence Number）

DCは、Active Directoryに格納されているオブジェクト上で実行されたオペレーションを記録するために、シーケンス番号を持っています。この番号はUSN（Update Sequence Number）と呼ばれます。

USNは、DC上で実行したオペレーションに関連付けられています。オペレーションを実行するたびに、シーケンス番号は1だけ大きくなり、オブジェクト内で更新されます。オペレーションは、成功する場合も失敗する場合もありますが、USN番号は、更新されたままになります。これは、LSASSプロセス（Active Directoryを実行しているプロセスの名前）がマルチ スレッドであり、複数のスレッドがデータベースに影響を与えている場合があるためです。オペレーションにはそれぞれUSNが必要です。USNは、自動的にインクリメントされ、実行されたオペレーションに対して割り当てられます。このトランザクションによって、USNは必ず1つのオペレーションに対応していることになります。

USNは、64ビットの数値であり、1から始まってシーケンシャルに増加するので、64ビットがオーバーフローするまでに、USNは20桁になります。オーバーフローが発生すると、複製プロセスは、ドメイン内の他のDCからの変更を、USNの1からすべてリクエストします。この結果、複製障害が発生します。USNのオーバーフローは、理論上は発生する可能性はありますが、オブジェクトの変更が20桁に達するには非常に長い時間がかかるので、現実的ではありません。各DCはドメイン内にある他のDCとは別に、独自のUSNを持っています。つまり、各DCのUSNは、他のDCとは関係なく変更されます。

ここまでの説明から、各オブジェクトは1つのUSNを持ち、これを使ってオブジェクトの変更を記録していることになりますが、Active Directoryのオブジェクトは、それぞれが2種類のUSNを持っています。1つ目のUSNは*usnCreated*と呼ばれ、オブジェクトの作成時に設定されます。2つ目のUSNは*usnChanged*と呼ばれ、オブジェクトが変更されるたびに値は更新されます。オブジェクトの属性も、それぞれ2つのUSN番号を持っています。1つ目は、属性が変更された時のオペレーションに関連付けられています。2つ目は、送信側のDCで実行されたライト オペレーションに関連付けられています。属性にUSN番号を持たせている理由は、Windows 2000では属性レベルの複製をサポートするためです。オブジェクト レベルのUSNがないと、分散インフラストラクチャでオブジェクト レベルの複製を実行した場合、トラッキングが不可能になります。

複製プロセス

この節では、USNの役割と、複製プロセスでのOriginating writeとReplicated writeの比較を詳しく説明します。

オブジェクトの作成

ドメインDC1の中に、DCが1つあると仮定します。このDC上では、多数のオペレーションが実行され、現在のUSNは110です。ユーザを作成すると、このオペレーションには、次のUSN番号である111が自動的に割り当てられます。ユーザ作成は失敗する可能性がありますが、USN番号はこのオペレーションに対して割り当てられたままになります。次に実行するオペレーションには、USN番号として112が割り当てられます。

作成オペレーションが完了した後、DC1のActive Directoryデータベースのテーブルから、次の内容がわかります。

- オブジェクトは作成されたばかりなので、このオブジェクトの*usnCreated*と*usnChanged*の属性には、同じUSNが割り当てられています。
- 各属性（ここでは、Name、Address、Phone、Passwordの4つのみとします）には、受け取った値が設定されています。
- 各属性には、現在のUSN（111）が設定されています。
- これは新規ユーザであり、各属性の値を1度設定しているので、バージョン番号は1に設定されます。バージョン番号は、競合を回避するために使用します。これについては、後で説明します。
- 現在のタイムスタンプが保存されます。タイムスタンプは、複製の競合の回避でも使用します。
- Originating WriteのDC GUIDが保存されます。DC1上にユーザを作成したので、DC1が送信側DCであり、GUIDがテーブルに保存されます。送信元のUSN（ここでは111）もテーブルに保存されます。

属性	値	USN	バージョン	タイムスタンプ	Originating DC GUID	Originating USN
Name	Joe	111	1	*TS*	*DC1 GUID*	111
Address	Compaq Valbonne	111	1	*TS*	*DC1 GUID*	111
Phone	+33 4 92 95 1111	111	1	*TS*	*DC1 GUID*	111
Password	********	111	1	*TS*	*DC1 GUID*	111

次のDCへの複製

次に、このユーザを、DC2という名前のDCへ複製します。DC2は、DC1の複製であり、同じドメイン内にあります。DC2には専用のUSNシーケンスがあり、現在の値は520です。したがって、DC2上では、Originating writeやReplicated writeなど、520個のオペレーションがこれまでに実行されたことになります。複製によって、新しいユーザ オブジェクトが追加されると、USNの値は521になります。

オブジェクトの複製が完了した後、DC2のテーブルから次の内容がわかります。

- これは、新しいオブジェクトの作成なので、オブジェクトUSNである*usnCreated*と*usnChanged*は、DCの現在のUSNである521を受け取ります。
- オブジェクトの属性の値は、複製されます。
- 複製オペレーションのUSN（521）は、各属性に保存されます。
- 複製オペレーションの時間は、タイムスタンプに保存されます。
- Ooriginating WriteのDB GUIDと、Originating WriteのUSNが保存されます。このオペレーションは、最初にDC1で実行されたものなので、保存されるのはDC1のGUIDとUSN番号です。

属性	値	USN	バージョン	タイムスタンプ	Originating DC GUID	Originating USN
Name	Joe	521	1	*TS*	*DC1 GUID*	111
Address	Compaq Valbonne	521	1	*TS*	*DC1 GUID*	111
Phone	+33 4 92 95 1111	521	1	*TS*	*DC1 GUID*	111
Password	********	521	1	*TS*	*DC1 GUID*	111

2番目のDC上でのオブジェクト操作

次に、ユーザ属性の1つであるAddressを変更します。

このオペレーションによって、DC2のUSNの値は521から522になります。ユーザの住所は変更されます。オペレーションを実行すると、テーブルから次の内容がわかります。

- *usnChanged*は、このオペレーションに関連したUSNで更新されます。
- Addressの値は、変更されています。
- このオペレーションのUSN（522）は、Address属性で保存されます。
- バージョン番号は1つ大きくなります。バージョン番号の役割については、後で説明します。
- タイムスタンプが保存されます。
- このオペレーションが最初に実行されたのはDC2なので、これがOriginating Writeになります。DC2データベースのGUIDと送信元のUSN（522）が保存されます。

属性	値	USN	バージョン	タイムスタンプ	Originating DC GUID	Originating USN
Name	Joe	521	1	*TS*	*DC1 GUID*	111
Address	**Compaq Sophia Antipolis**	**522**	**2**	***TS***	***DC2 GUID***	**522**
Phone	+33 4 92 95 1111	521	1	*TS*	*DC1 GUID*	111
Password	********	521	1	*TS*	*DC1 GUID*	111

DC1への複製

マルチ マスタ レプリケーション モデルでは、すべての複製の一貫性を維持するために、DC2で実行したオペレーションをDC1へ複製する必要があります。データを複製すると、DC1のUSNは111から112になります。DC1のユーザのテーブルから、次の内容がわかります。

- *usnChanged*プロパティは、このオペレーションのUSNである112で更新されます。*usnCreated*プロパティは変更されないという点に注意してください。
- Address属性の値は、複製されます。
- USNは、属性レベルで保存されます。
- バージョン番号は2になります。
- タイムスタンプが保存されます。
- このオペレーションが最初に実行されたのはDC2なので、これはReplicated Writeになります。送信元であるDCのDC GUIDとUSN（522）が保存されます。

属性	値	USN	バージョン	タイムスタンプ	Originating DC GUID	Originating USN
Name	Joe	111	1	*TS*	*DC1 GUID*	111
Address	**Compaq Sophia Antipolis**	**112**	**2**	***TS***	***DC2 GUID***	**522**
Phone	+33 4 92 95 1111	111	1	*TS*	*DC1 GUID*	111
Password	********	111	1	*TS*	*DC1 GUID*	111

Originating WriteとReplicated Writeを区別することや、属性レベルでのUSNの保存は、次のような理由で、非常に重要な意味を持ちます。

- Windows 2000複製は、属性レベルで実行される場合、すべての属性を変更してしまうオブジェクト作成と属性変更を区別する必要があります。区別することにより、オブジェクト全体ではなく、属性の値を複製することが可能になります。
- 送信元のUSNは、伝播の抑制、つまり、DCが更新済みかどうかを検出する機能に使用します。次の節では、このメカニズムを詳しく説明します。

伝播の抑制

伝播の抑制とは、複製が、複製パートナーで既に実行済みかどうかを検出する機能です。この機能には、同じDCを同じ内容で2回複製してしまうことを防ぐ目的があります。

これは、Active Directory複製の非常に重要な機能です。マルチ マスタ レプリケーションでは、DCは複数の複製パートナーを持ち、各パートナーは様々なパスを使ってターゲットDCへデータを複製するので、伝播の抑制は、重要な役割を果たします。ターゲットDCへ最初に到着した複製オペレーションがActive Directoryを更新します。したがって、複製プロセスは実行済みであることをすべてのDCが認識することによって、トラフィックの量を減らし、不要なオペレーションをなくすことができます。

伝播の抑制と複製がトリガされる方法を理解するために、Active Directoryの2つのテーブル、高水準ベクタ テーブルと最終更新ベクタ テーブルについて説明します。

高水準ベクタ テーブル

*高水準*ベクタは、各DC上に格納されているテーブルです。このテーブルには、全複製パートナーの一覧とUSNの最大値が格納されています。高水準ベクタは、複製パートナーである他のDCで実行された変更を検出するために使用します。また、このテーブルから、まだ複製されていない変更内容がないか検出し、これをリクエストすることもできます。たとえば、1つのドメイン内にDCが2つ（DC1とDC2）ある場合、この2つのシステムが複製パートナーとなります。DC1は、DC2から最大のUSNを受け取り、高水準ベクタ テーブルへ保存します。ここでは、このUSNが100に設定されているとします。DC2では、USNは現在120です。つまり、20個のオペレーションが未処理の状態であり、DC1へ複製する必要があることを意味します。このテーブルがないと、DC2は、すべてのオペレーションを複製するしかなく、帯域幅と処理時間が無駄になります。

複製パートナーは、通知メッセージを介して、最大のUSN番号を交換します。6時間以内に更新が行われていない場合、複製パートナーは、最大のUSNをお互いに交換します。DCがオフラインになっていた後、複製パートナーとの接続が復帰したばかりで、通知をいくつか受け取っていない時に、このようなメッセージが交換されます。

たとえば、1つのドメイン内にDC1、DC2、DC3、DC4という4つのドメイン コントローラがあるとします。DCはそれぞれ2つの複製パートナーを持ち、リングを形成しています。図3で示したDC4の高水準ベクタ テーブルを見ると、USNの最大値など、2つの複製パートナーのデータが保存されています。DC4は、パートナーDCが持っている実際のUSNよりもテーブル内のUSNの方が小さいことを検出すると、テーブルは最新の状態ではないことがわかるので、パートナーからデータを要求するために、複製サイクルを開始する必要があると判断します。

最終更新ベクタ テーブル

エンタープライズWindows 2000ネットワークでは、DCが多数存在します。複製は複数のパスで実行されるので、オブジェクトを何度も更新してしまう危険があります。

Active Directoryは、不要な複製を防ぐため、*最終更新ベクタ*を使用します。これは、ステート ベクタ テーブルとも呼ばれます。このテーブルには、複製パートナーの一覧と、各パートナーからのOriginating WriteのUSNの最大値が格納されています。各DCは専用の最終更新ベクタ テーブルを持ち、これを複製パートナーへ送ることによって、パートナーは、既に最新の内容になっている複製不要なデータを除外します。複製パートナーは、最終更新ベクタ テーブルのUSNと、自分の高水準ベクタ テーブルを比較して、更新が必要な属性があるかどうかを判断します。すべてのDCは、どの複製パートナーが最新の状態かがわかっているので、不要な複製を行わないで済むため、このオペレーションには伝播の抑制機能を採用します。

図3に示した4つのコントローラを含むドメインの例を見ると、最終更新ベクタ テーブルから、Originating Writeが実行されたのはDC1とDC2のみであることがわかるので、DC1とDC2のGUIDは、DC4の最終更新ベクタ テーブルに保存されます。

H

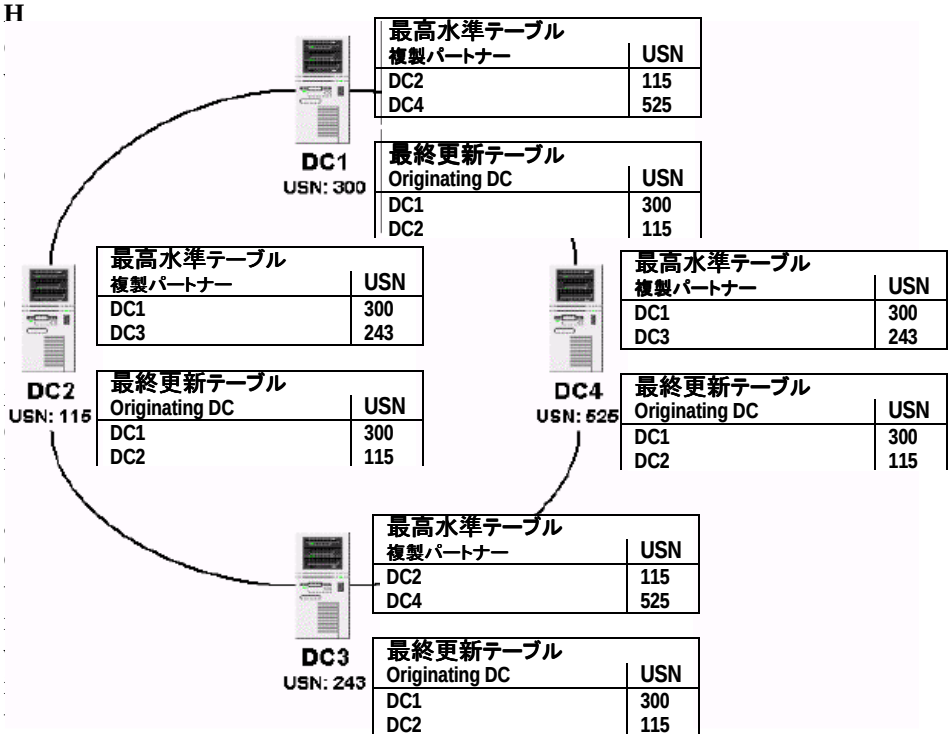


図3：高水準ベクタ テーブルと最終更新ベクタ テーブル

y

複製のトリガと抑制のメカニズム

高水準ベクタ テーブルと最終更新ベクタ テーブル複製を使った、複製のトリガと抑制のメカニズムを理解することは重要です。DCは、複数の複製パートナーを持っています。したがって、複数のDCが、同じ複製パートナーと同じデータを複製し、オペレーションが何回も複製される可能性があります。これを防ぐために、Windows 2000複製は、抑制機能をサポートします。複製の抑制機能とは、DCで既に実行された複製を検出するものであり、マルチ マスタ レプリケーション モデルでは重要な役割を果たします。この機能がないと、DCが構成する複製トポロジでは、データがDC間を何度も行き来し、膨大なネットワーク トラフィックが生成されてしまい、複製メッセージのデータ処理のために貴重なCPUサイクルを無駄に使うことになり

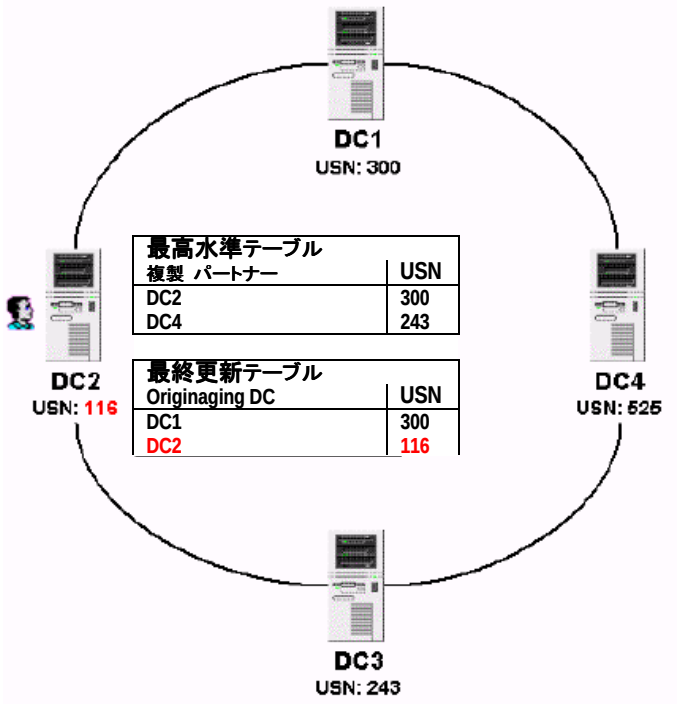
ます。

高水準ベクタ テーブルと最終更新ベクタ テーブルを使った複製のメカニズムを説明するために、例を使って、ドメイン内でのデータ複製のプロセスと各DCの役割を考えます。ここでも、DC1、DC2、DC3、DC4の4つのDCを持つドメインの例を使います。DCの高水準ベクタ テーブルから、どのコントローラが複製パートナーであるかがわかります。図3を見ると、DCは、それぞれ隣にあるDCを複製パートナーとして持っていることがわかります。また、テーブルの内容から、各DCが複製パートナーと完全に同期した状態になっているかどうかがわかります。たとえば、DC4の高水準ベクタ テーブルからは、DC1とDC3が複製パートナーであり、現在同期した状態であることがわかります。これは、テーブル内のUSN番号が、DC1とDC3の現在のUSN番号と同じだからです。

DCが完全に同期した状態であれば、すべてのDCの最終更新ベクタ テーブルは同じになります。DC4の最終更新ベクタ テーブルを見ると、DC1とDC2がOriginating Writeを実行したことがわかります。つまり、アドミニストレータまたはユーザがDC1とDC2でタスクを実行し、複製をトリガしたことを意味します。では次に、DC2でユーザ オブジェクトを作成した場合を考えてみましょう。

ステップ1：ユーザの作成と、隣のパートナーに対する複製

図4：DC2にユーザを作成



ユーザをDC2上に作成します。このオペレーションでの、DC2のUSN番号は、前のUSN（115）に1を加えた116になります。

一定の時間が経過した後（これについては後で説明します）、DC2は、複製パートナーに対して、変更が発生したことを通知し、最大のUSN番号を送ります。DC2の複製パートナーは、DC1とDC3です。この2つのDCは、高水準テーブルに保存されているDC2のUSN番号をチェックし、変更が発生したことを認識します。DC1とDC3は、複製サイクルをトリガし、DC2からの情報をリクエスト（プル）します。では、ここで、最初のコントローラであるDC1が実行したオペレーションから考えます。

更新をリクエストするために、DC1は次の情報をDC2へ送ります。

- 複製 オペレーションは、それぞれが名前付けコンテキスト（NC）に関連付けらているので、DC1は、必要なNCを指定する必要があります。ここでは、作成されたユーザの複製を行うため、ドメインNCに含まれています。
- このNCの最大USN – これに基づいて、DC2は変更内容を複製する必要があるかどうかを判断します。DC1が、最大のUSNが115であることをDC2に通知すると、DC2は、USN 116に相当するデータを提供しなければならないことがわかります。
- 複製されたオブジェクトの数と値 – この値は、デフォルトで100に設定されていますが、レジストリで変更できます。この値の目的は、ネットワークの混雑を防ぎ、複製パートナーで大量の変更が発生しないようにすることにあります。複製は、100個のオブジェクト単位で実行されます。100個のオブジェクトが転送され、値を処理します。一定の時間が経過したら、次の100個が転送されます。
- 最終更新ベクタ テーブル ― このテーブルから、DC2は、DC1へ複製済みのOriginating Writeを認識できます。このテーブルの役割については、後で詳しく説明します。

上記の情報から、DC2は、複製が必要なオペレーションはどれかを正確に判断できます。データは、DC1からDC2へ提供され、新しいオブジェクトがActive DirectoryのDC1上で作成されます。その結果、図5に示すように、DC1のUSNは300から301となります。

ステップ2：2番目のパートナーに対する複製

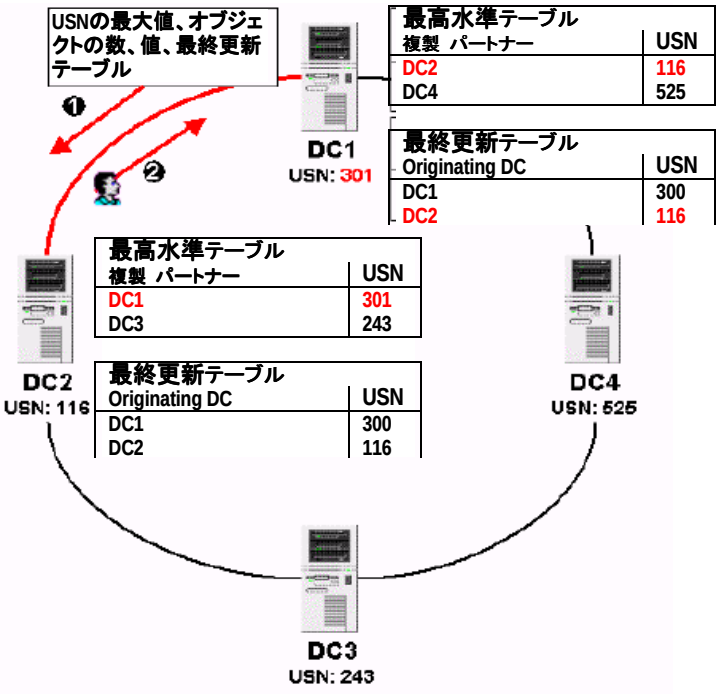
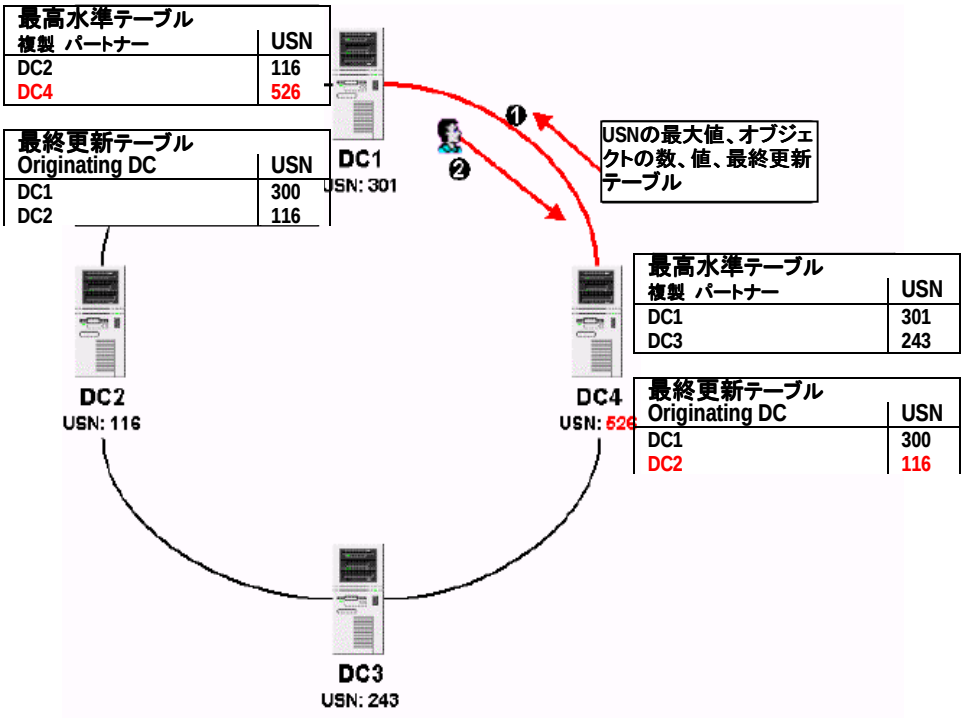


図5：ユーザ オブジェクトをDC1へ複製

DC1は、複製パートナーとして、DC2とDC4の2つを持っています。DC2からはユーザを複製しています。DC1は、情報が複製されたことをDC4に通知します。また、DC1は、USN番号がインクリメントされたことをDC4に通知します。

図6：ユーザをDC4へ複製



DC1のUSNが変更されたことがDC4に通知されると、DC4は通知されたUSNの値と自分の最終更新テーブルの値を比較し、複製が必要であることを認識します。DC4は、複製を開始するために必要な情報をDC1に送信し、DC1からの情報をリクエストします。

上記のようなオペレーションから、Windows 2000の複製はマルチ ティアで実行されることがわかります。つまり、DCは相互にコンタクトし、これが他のDCへのコンタクトへとつながります。このメカニズムでは、分散型のインプリメンテーションで、一度作成したデータをマルチ ティア ツリーのように波及していくことができるので、非常に重要な役割を果たします。Windows NTの複製モデルは非常に単純で、すべての情報をPDCから複製します。したがって、NT4で必要な帯域幅は、ドメイン内のBDCの数によって決まります。これに対して、Windows 2000では、DC間で複製しなければならない情報量によって必要な帯域幅が決まります。帯域幅は、Windows 2000ドメインで使用する複製トポロジに大きく依存します。これについては、後で詳しく説明します。

DC4のテーブル（図6）から、次の内容がわかります。

- 最終更新ベクタ テーブルは、Originating Writeの情報で更新されています。ユーザ オブジェクトは、DC2上で作成されているので、DC2のGUIDと最大のUSN番号が保存されています。
- 高水準ベクタ テーブルは、DC4の複製パートナーであるDC1の情報で更新されています。DC1の最大USNが保存されています。

ステップ3：複数の複製パートナー

これまでの操作で、DC2で作成された新しいオブジェクトは、DC1へ複製されました。次に、DC4への複製を行います。

DC2は、もう1つDC3という複製パートナーを持っています。DC2が、パートナーに対してUSNが変更したということを通知すると、すべての複製パートナーは、DC2の最大USNを受け取り、同様の処理を行います。DC1と同様に、DC3は、変更内容を複製する必要があることを検出して、DC2との複製を開始します。

図7は、DC2からDC3への複製の結果を示しています。

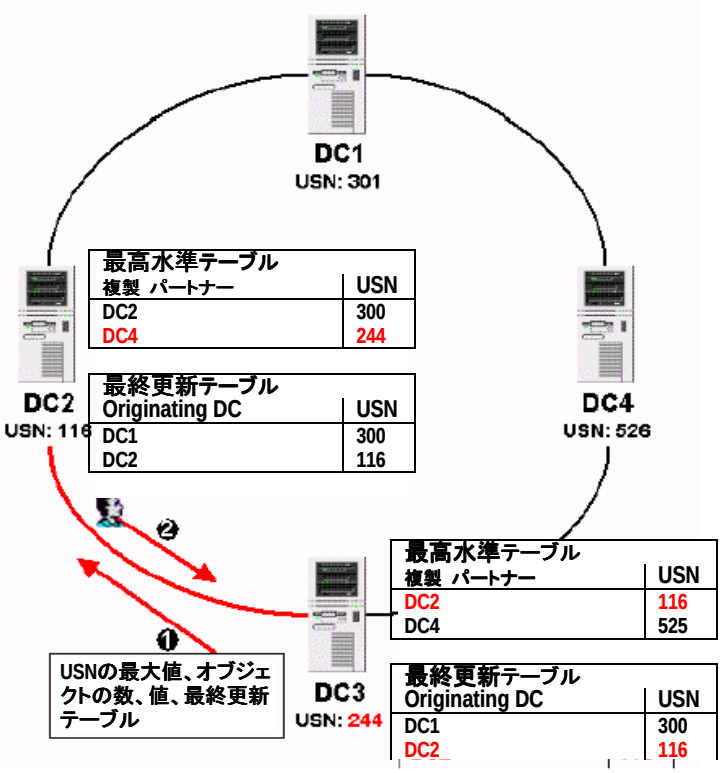


図7：DC2からDC3へのレプリケート

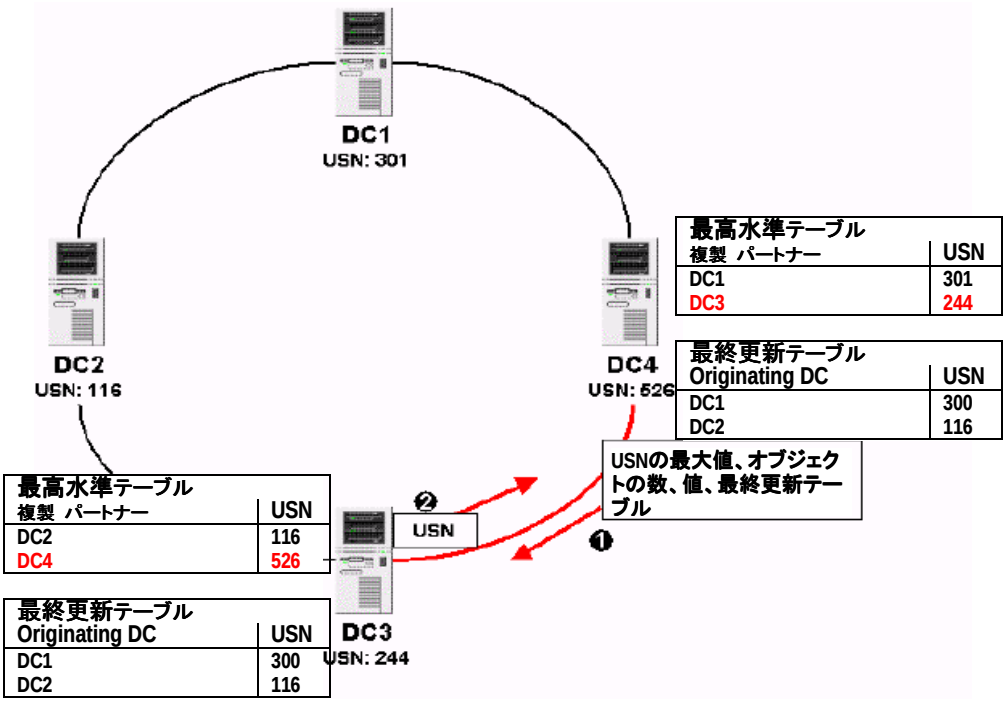


図8：伝播の抑制

ステップ4：伝播の抑制

このステップは、伝播の抑制機能のしくみを理解し、全体の概念を把握する上で、鍵となる部分です。

DC3は、DC2とDC4の2つの複製パートナーを持っています。ユーザオブジェクトは、DC2からDC3へ複製されています。DC4は、DC3の複製パートナーであるため、DC3はデータをDC4へ複製しようとしています。しかし、DC4は、新規ユーザに関するデータをDC1から既に複製しているので、最新の状態になっています。ただし、DC3は、DC4が最新の状態であることを認識していないので、伝播プロセスが抑制されない限りは、データを複製しようとします。

複製プロセスの1過程として、DC4はパラメータを送信し、これによってDC3は、複製するデータの内容を正確に把握します。このパラメータの1つに最終更新ベクタ テーブルがあり、テーブルにはドメイン内のOriginating Writeが含まれています。

DC3は、DC4が既に更新済みであることを最終更新ベクタ テーブルによって把握できるので、伝播抑制機能は、このテーブルによって制御されます。最終更新ベクタには、Originating Write DCのGUIDと、このオペレーションのUSNが格納されており、DC3は、このテーブルの内容から、ユーザが最初に作成されたのはDC2であり、DC4には別のパスを介して複製済みであることがわかります。DC3が最終更新ベクタを受け取ると、テーブルの内容から、DC2上で発生した変更内容がUSN 116まではDC4に複製されていることがわかります。DC2の最大USNは、116がDC3に通知されています。両方のUSNが一致し、DC3はDC4へ複製する必要がないことがわかるので、伝播を抑制します。DC3は、自分の最大USN番号の送信だけを行います。DC4は高水準ベクタ テーブルを更新し、複製パートナーと同様、最新の状態になります。

ここで紹介しているのは、DCが数個しかない簡単な例です。エンタープライズのWindows 2000インフラストラクチャや何百ものDCが存在する環境を想像してみてください。DCが複数の複製パートナーを持つモデルでは、同じ情報の複製を防ぎ、ネットワーク帯域幅や時間を節約するためにも、伝播の抑制は重要な機能となります。

サイト

Windows 2000サイトは、接続性に優れたIPサブネットの集合です。最もわかりやすいのは、サイトをLANにたとえて考えてみてください。同じサイトに属する全システム同士が物理的に近い位置にあり、理想的には、LANクラスのネットワーク接続性のメリットを活用できるのであれば、サイトはローカル性を反映します。つまり、Windows 2000サイトの各サーバは、512 K/bps以上のリンクで接続してください。本格的なLANタイプの帯域幅（10Mb/s以上）が望ましいと考えれます。サイト内にあるサブネットに求められるLANクラスの接続性は、サイト間 複製の実行方法に大きく依存します。

ローカル性の概念は、ワークステーションにまで拡張されます。ワークステーションは、同じサイト内のドメイン コントローラへ常に接続しようとするためです。Active Directoryは、サイトやIPサブネットに関する情報を含み、ワークステーションのIPアドレスとサイト定義を比較することによって、ワークステーションとサイトの関連付けを行います。サーバをドメインへ追加すると、Active Directoryは、サーバのIPアドレスをチェックし、最適なサイトへ配置しようとします。

Windows 2000サーバが新しいドメインを作成する場合、Active Directoryはサイトの*Default-First- Site-Name*を生成し、ここにDCを配置します。このドメインに属すすべてのDCは、デフォルト サイトに追加されます。新しいサイトが明示的に作成されるまでは、システムはデフォルト サイトに追加されます。

図9：サイト

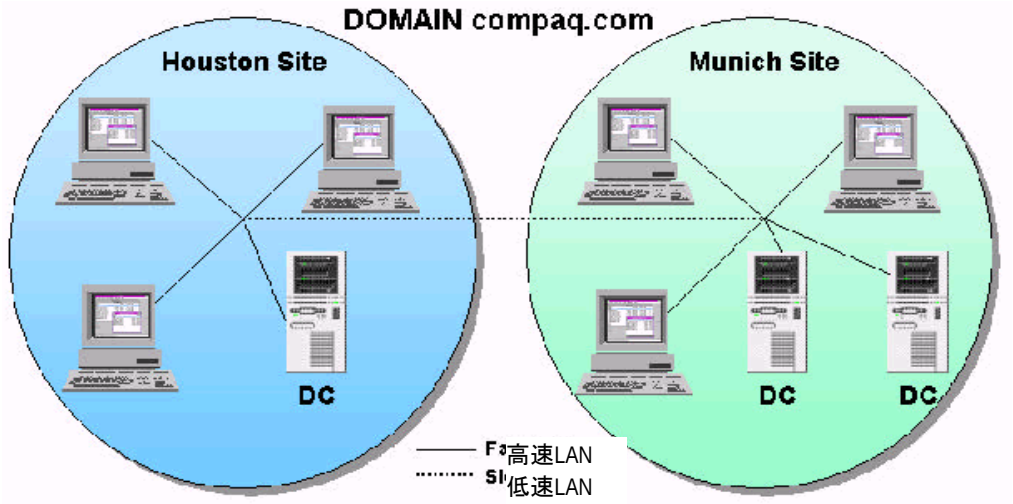


図9のように、ドメインは、複数サイトにまたがる場合があります。サイトは、これが属すドメインとは独立しているので、複数のドメインが1つのサイトに属す場合があります。サイトは、構成 NCに格納されているオブジェクトなので、サイト情報はフォレスト内の全DCへ複製されます。

サイトには、次のような役割があります。

1. ワークステーションのログオンで、最も近いDCを決定します。この方法については、次の節で説明します。
2. Active Directory複製で、ルートやサイト間のデータ転送を最適化します。これについては、後で説明します。

この2つの役割を果たすために、ドメイン内の全DCは、全サイトに関する情報を持っています。

複製トランスポート

複製は、同じサイト内のDC間と、異なるサイトにあるDC間で発生します。サイト内 複製は、標準Windows 2000 RPC（Remote Procedure Call）で実行されます。ただし、異なるサイトにあるドメイン間の複製には、異なるトランスポートを使用します。Active Directoryは、各サイト内のドメイン コントローラ、各サイト間の接続性、複製情報を取得する際の最適なルートなどに関する情報を保持しています。サイト間のネットワーク リンクには、大きなばらつきがあります。様々なタイプの接続をサポートするために、Active Directoryは、主に2つの複製トランスポートをサポートしています。

1. DS-RPC（Directory Services RPC）
2. ISM- SMTP（Inter-Site Messaging - Simple Mail Transport Protocol）

一般的な前提として、複製はサイト内で迅速に処理される必要があり、サイト内の全コントローラに渡るディレクトリ内では、不整合に対する耐性が弱くなります。このため、サイト内複製は、常にRPCベースとなります。サイト間複製は、RPCを使用する場合と、ドメイン コントローラ間で送信されるSMTPメッセージという形式をとる場合があります。メッセージ ベースの複製が非同期なのに対して、RPC複製は、必ず同期です。サイト内複製は、スケジュールに従った実行はできません。各ドメイン コントローラは、更新が完了した後、更新の通知を複製パートナーへ送信します。通常、これは5分おきに実行されます。この間隔は、システム レジストリで設定します。変更がなく、複製が発生していない場合でも、ドメイン コントローラは、同期が保たれていることを確認するために、最新のUSNの詳細を6時間おきにやりとりします。DCは、更新の通知を受け取ると、変更が発生したDCへ接続し、複製を開始します。サイト内複製では、データは圧縮されません。

サイト間複製では、複製トランスポートのタイプに関係なく、データが50KBよりも大きい場合には、データを圧縮します。データは、元の10～15%に圧縮されてから、送信されます。データの圧縮と伸長にはCPUサイクルはが必要ですが、ネットワーク送信するデータのサイズが小さいことによるメリットは、CPUサイクルで必要なオーバーヘッドよりも大きくなります。サイト間複製は、設定した時間に実行するようにスケジュールできます。

サイト間の接続性は、サイト内には劣りますが、データ圧縮とスケジューリングを利用できるというメリットがあります。スケジューリングによって、複製を強制的に実行できます。複製トポロジは、通常の場合、自動生成されます。

サイト内複製とサイト間複製の比較

	サイト内	サイト間
トランスポート	*RPC*	*RPCまたはSMTP(1)*
トポロジ	*リング*	*分散ツリー*
スケジュール	*頻度*	*スケジュール*
複製モデル	*Notify and Pull*	*Store and Forward*
圧縮	*なし*	*50KBより大で実行*

サイト間複製とサイト内複製には、いくつかの相違点があります。

次に、サイト内複製モデルの特徴を示します。

- トランスポートは、IP上のRPCのみ
- 生成された複製トポロジは、ドメイン コントローラ間のリング
- 複製の間隔は、デフォルトで5分おき
- 複製は、通知と要求モデル
- サイトの帯域幅が使用可能なので、データは圧縮しない

サイト間複製モデルには、次のような特徴があります。

- トランスポートは、RCPまたはSMTP。ただし、構成とスキーマ名前付けコンテ

キストの複製や、DCからGCへの複製では、SMTPトランスポートのみ

- 生成されるトポロジは、分散ツリーであり、ループを生成しない
- スケジュールによって、サイト間のデータ 複製を制御。複製は、デフォルトで3時間おき、最短で15分おき
- データが50KBを超える場合は圧縮

複製トポロジ

複製トポロジは、ドメイン コントローラ間で情報を複製する方法を示すマップです。トポロジは、各DC上で稼動するサーバが生成し、維持します。

DCは、複製トポロジに従って、他の複製パートナーを探します。たとえば、DCがシャットダウンによって使用不能になった場合、複製パートナーが、別ルートで複製を実行します。

名前付けコンテキスト スコープごとに、フォレスト ワイドとドメイン ワイドの複製トポロジがあります。つまり、スキーマと構成 NCは、フォレスト ワイド スコープを持つため、同じトポロジを共有します。フォレスト内の各ドメインは、ドメイン ワイド スコープを持つので、独自のトポロジを生成します。

たとえば、フォレスト内に3つのドメインがある場合、4つの複製トポロジが生成されます。

- ドメインNCに1つずつ
- 構成 NCとスキーマ NCについて1つずつ

知識整合性チェッカ（KCC）

知識整合性チェッカ（KCC）は、各DC上で実行されるサービスです。KCCには、次のような役割があります。

- DC間に接続オブジェクトを生成することによって、複製トポロジを生成および最適化
- 複製データの整合性チェックと、複製 パートナーに対する複製のトリガ

デフォルトで、KCCは15分おきに実行されます。アドミニストレータは、［Active Directory サイトとサービス］スナップ インによって、KCCを手動で実行します。次のレジストリ設定によって、トポロジのチェック間隔を変更できます。

```
HKLM¥System¥CurrentControlSet¥Services¥NTDS¥Parameters¥Repl
topology update period (secs)
```

接続オブジェクト

接続オブジェクトは、DC間で情報を複製するために使用する認証パイプです。接続オブジェクトのプロパティには、複製パートナーや、これが属すサイトなどが含まれます。プロパティには、接続オブジェクトが使用するトランスポート（RPCまたはSMTP）、複製に使用するスケジュールなども定義されています。

接続オブジェクトは、複製パートナーからの一方向の接続です。つまり、サイト内のDC間には、必ず2つの接続オブジェクトが存在することになります。接続オブジェクトは、通常KCCが生成しますが、アドミニストレータも手動で生成できます。アドミニストレータは、必要に応じて、特定の接続オブジェクトで複製を強制的に実行できます。

複製を強制実行するには、［Active Directory サイトとサービス］スナップ インの接

続オブジェクトを右クリックし、［今すぐ複製］ボタンをクリックします。

複製の強制には、Replication Monitorと呼ばれるリソース キット ツールを使用する方法もあります。このツールは、名前付けコンテキスト全体を同期します。Replication Monitorは、選択した名前付けコンテキストを含むすべてのDC間で、複製を実行します。

図10の接続オブジェクトは、KCCが生成したものです。また、このDCはGCであり、構成とスキーマ NCと部分ドメイン*cpqcorp.net*を複製していることがわかります。

サイト内複製トポロジの生成

Windows 2000ドメインを新しく作成し、ドメイン内にはDCが1つだけある場合、複製は不要なので、何も起こりません。2つ目のDCがネットワークへ追加されると、KCCは2つのDC間で複製トポロジを生成します。

2つのコントローラは同じサイト内にあるので、図11のように、双方向の接続が生成されます。この接続オブジェクトは、2つのDC間の相互認証チャネルであり、これを介して情報の複製が可能です。複製されたオブジェクトは、これが属する名前付けコンテキストに依存します。2つのDCが最初に接続する時、すべてのNCが複製されます。つまり、最初のDCが属している構成、スキーマ、ドメインが複製されます。異なるドメインのDC間では、構成とスキーマ NCだけが複製されます。DCがグローバル カタログでもある場合は、3つのNCも複製されます。ただし、グローバル カタログに複製されるのは、ドメインNCのサブセットだけです。

同じサイトにDCが追加されると、各DC上で実行されているKCCサービスは、自動的に必要な接続オブジェクトを生成し、複製パスが重複しないようにします。コネクションの数を最適化するために、各DC上のKCCは、サイト内のDC間のホップが3台以内になるように、必要とするコネクションの数を計算します。

複製のリング

ここでは、1つのドメインcompaq.com、1つのドメイン コントローラDC1を持つサイトを例にあげます。

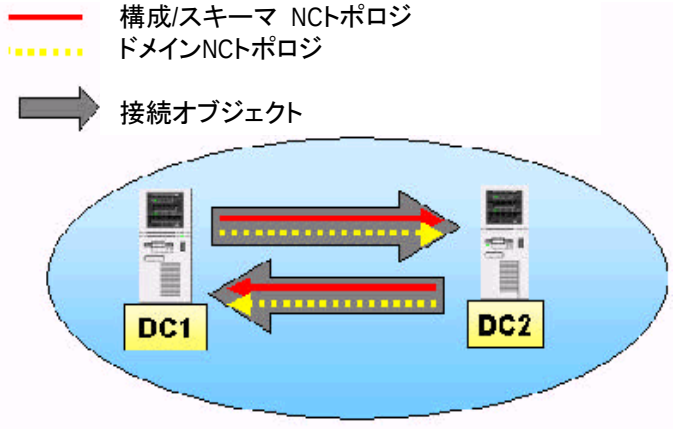


図11：2つのDC間の接続オブジェクト

2つ目のサーバであるDC2上でDCPROMOを実行します。DCPROMOを使って、DC2をcompaq.comドメインの複製として設定します。

DC2上のKCCは、DCPROMOプロセスで、接続オブジェクトを生成し、各スコープ（フォレスト ワイドとドメイン ワイド）に1つずつ、合計2つの複製トポロジを生成します。図12は、上のラインは構成 NCとスキーマ NCの複製トポロジ、下のラインはドメインNCの複製トポロジを示しています。DC2は、構成、スキーマ、ドメインの各オブジェクトをDC1から複製します。DCPROMOプロセスが完了すると、DC1はDC2から接続オブジェクトを生成し、DC2で実行されたオペレーションがあれば、これを複製します。

次に、3つ目のサーバDC3をブートし、DCPROMOを実行してcompaq.comドメインへ複製として追加します。

図12では、DC3がDC2へ接続した後、ドメインに追加されています。ここで問題になるのは、新しいコントローラが追加された時、ドメインの中の最初のコントローラとの最初の接続は、なぜ自動的に確立されないのか、という点です。サーバをドメイン コントローラへ昇格するには、DCPROMOユーティリティを使います。プロモーション プロセスの間に、指定のコントローラにコンタクトしてドメイン情報を受け取るかどうかを決定します。

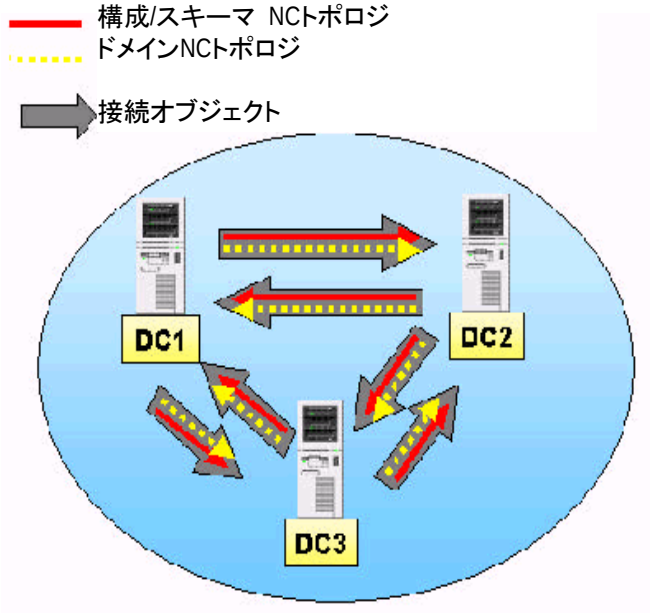


図12：3つのDC間で作成されたトポロジ

DCPROMOは、DC3上で実行されると、DNSが、compaq.comドメインに属すドメイン コントローラのリストを返すかどうかチェックします。Compaq.comドメインのSRVレコードを受け取ると、DNSはDC1とDC2に応答します。DC3は、DC1とDC2に対してブロードキャストし、どちらが現在使用可能なのかを検出します。この例では、DC2が先に応答することとします。

DC3は、DC2にコンタクトし、スキーマ/構成とドメイン名前付けコンテキストに必要な2つのトポロジ用に、接続オブジェクトを生成します。複製が実行され、DC3がドメインに加わります。

次に、DC2は、DC3からの更新情報を使って、自分の複製トポロジを生成し、新しいDCがドメインへ追加されたことをDC1へ通知します。DC1への通知は、構成 NCの複製によって実行されます。この構成 NCには、新しいDCに関する情報が含まれています。DC1とDC3は、相互の複製に必要となる接続オブジェクトを生成します。

図13は、4つ目のDC4がドメインに追加された様子を示しています。DC2とDC3は、DC4への接続オブジェクトを生成します。

各DC上で実行されるKCCサービスは、複製トポロジを常に最適化しようとします。3ホップ以内でDCにアクセスできない場合は、接続オブジェクトが新しく生成されます。たとえば、7つのDCが同じサイト内にある場合、DCは、他のDCへ3ホップ以内ではアクセスできないので、3ホップ以内でアクセスできるように、各コントローラへの接続オブジェクトが生成されます。

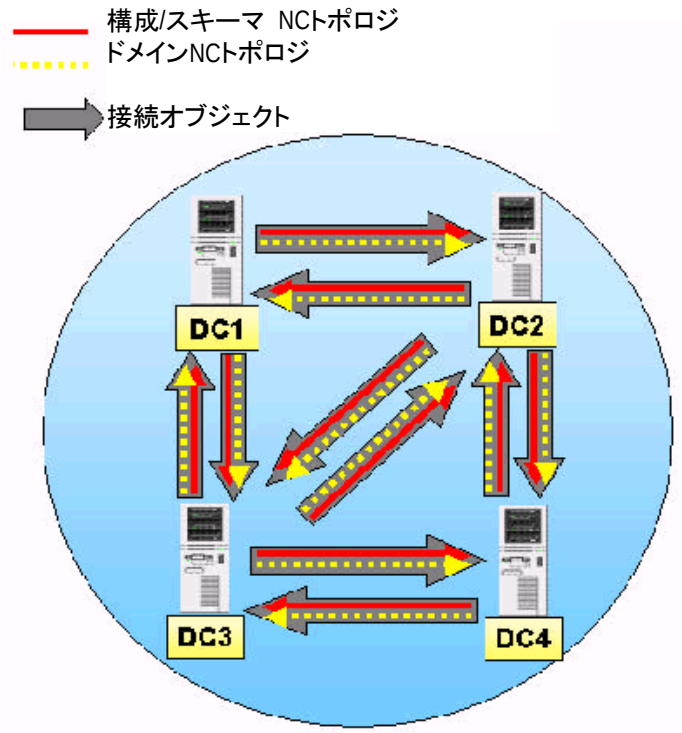


図13：サイトにDCが4つある場合のリダンダント接続

この例では（図13）、DCは4つのみなので、DCはどのDCにも3ホップ以内でアクセスできます。KCCは、十分な複製パスが既に存在しているので、DC3とDC2の間に接続オブジェクトは不要だと判断します。KCCは、サイト内に複製パスのリングを常に作ろうとします。DC3とDC2のKCCは、他に有効なルートが存在するので、お互いをポイントする接続オブジェクトを削除します。この結果、複製トポロジは2つになり、リングを形成します。

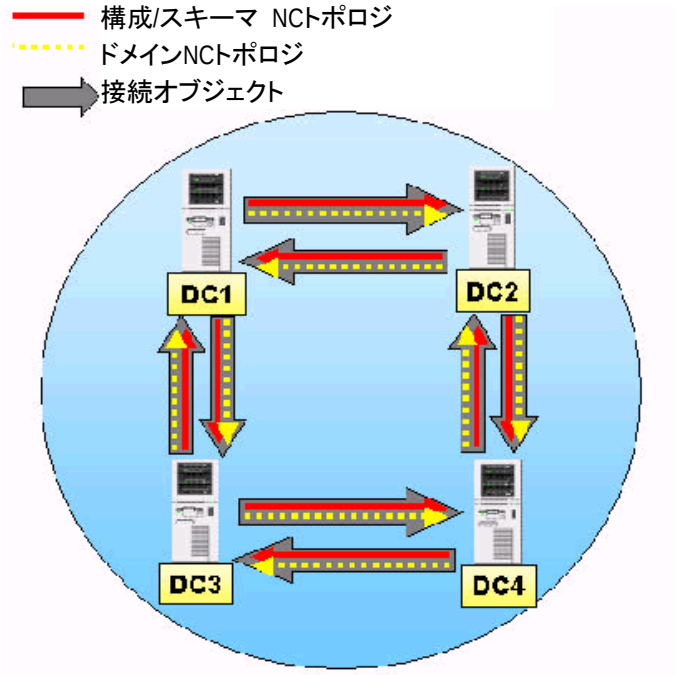


図14：接続オブジェクトの最適化

DCが2つの場合には、一方向に1つずつ、最大2つの接続オブジェクトが生成されます。複数の複製トポロジは、接続オブジェクトを共有してレプリケーションを行います。

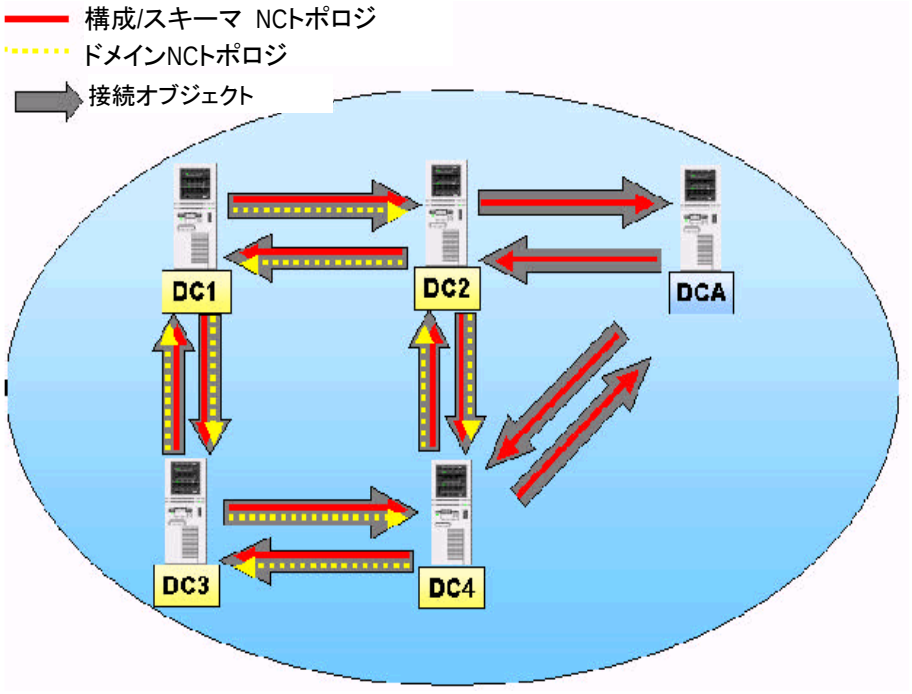


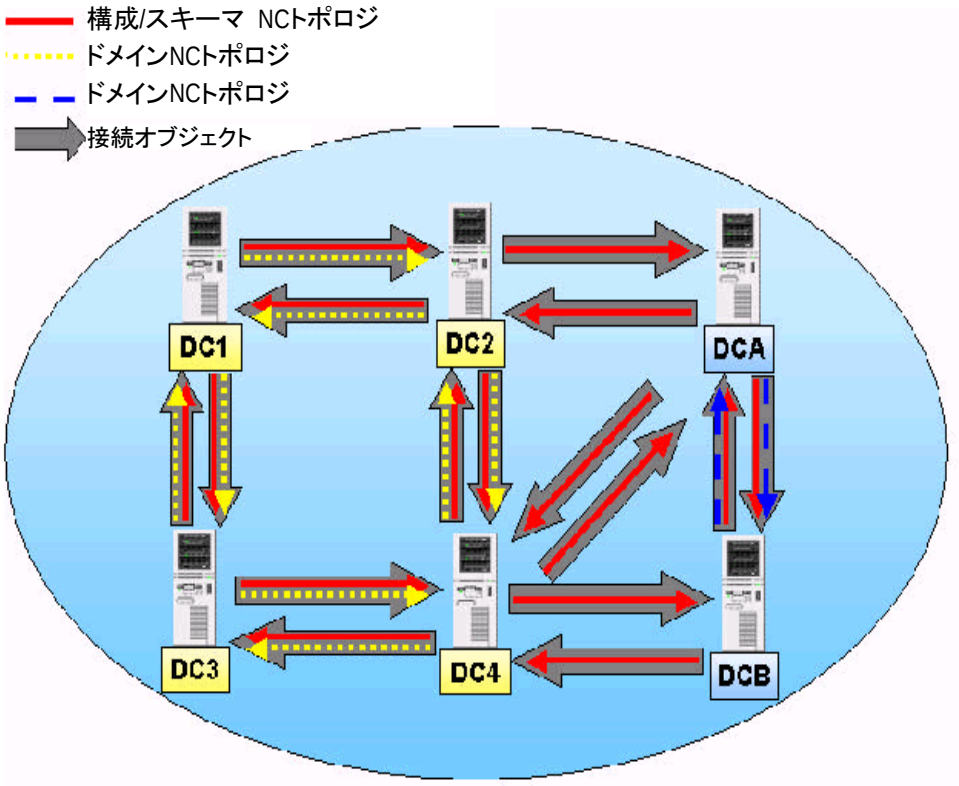
図15：トポロジにドメインを追加

次に、図15に示すように、compaq.comドメインの子ドメインを作成し、sales.compaq.comという名前を付けます。compaq.comドメインに属する他のDCと同じサイトに、子ドメインのDC（DCA）をインストールします。これまで説明した通り、新しいコントローラ（DCA）はDNSに問い合わせし、compaq.comドメインに属す全DCを検索し、ブロードキャストによって、DCからの応答を待ちます。ここでは、Compaq.comドメインのDC2が、最も速く応答したとします。

DCPROMOプロセスでは、DCAは、構成とスキーマ NCの両方に接続オブジェクトを生成します。compaq.comのドメイン NC接続は作成されない点に注意してください。これは、ドメイン NCは、自分のドメイン内でしか複製されないからです。DCAは、compaq.comドメインに属さないので、フォレスト ワイド トポロジの接続オブジェクトだけが生成されます。DCAがフォレストに追加され、DC2は、他の全DCに対して、DCを持つ新しい子ドメインが同じサイト内に存在することを通知します。他のDC、厳密に言うとDC上で実行されているKCCは、状況を判断し、DC4は、DCAとの接続オブジェクトを作成することによって、ループを閉じます。DCAは、同様にDC4との接続オブジェクトを生成します。

ここで、例を更に複雑にするために、子ドメインsales.compaq.comへDCBを追加します（図16）。DCPROMOプロセスの間、DCBは、DCAへアクセスし、sales.compaq.comドメインNCと、構成とスキーマ NCを複製します。DCAは、フォレスト内の他のDCへ、新しいDC（DCB）がドメインに追加されたことを通知します。KCCは、状況を判断し、トポロジを最適化します。DC4からDCBの接続オブジェクトが生成されます。

図16：複製を追加



最適化プロセスでは、DCAとDC4は、相互間の接続オブジェクトを削除します。これは、他のルート上にリングが既に存在するからです。最適化の結果、3つのリングを持つ2つのドメインが作成されます。

- compaq.comの Domian NCが1つ
- sales.compaq.comのドメイン NCが1つ
- フォレスト全体のスキーマ/構成 NCが1つ

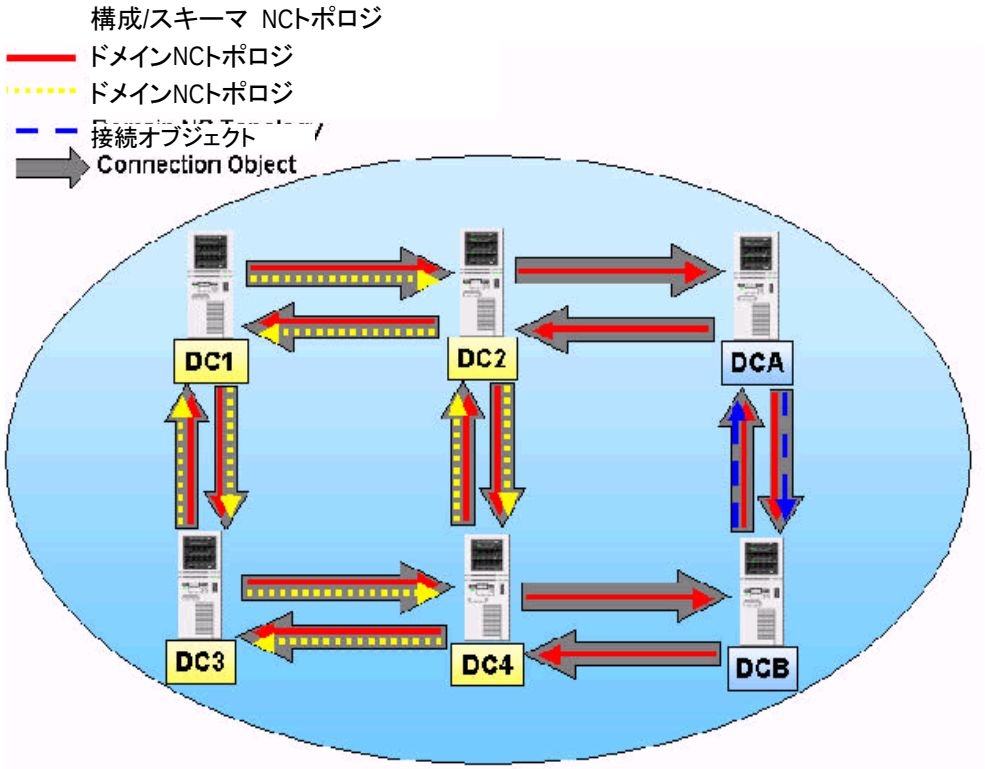


図17：ドメイン間の接続オブジェクトの最適化

図17は、すべてのドメイン コントローラをフォレストに追加した後に生成された複製のリングを示しています。実線の矢印は、構成とスキーマ NCトポロジ リング、点線とダッシュの矢印は、2つのドメインNCを示します。

グローバル カタログ複製

グローバル カタログ（GC）は、特殊な形のドメイン コントローラであり、フォレスト内の全ドメインからのオブジェクト情報を、読み出し専用のサブセットとして持っています。グローバル カタログを構成するための複製とは、ドメインNCがフォレスト内で複製されることを意味します。前の節の例では、DC4とDC1がGCであるとすると、KCCは、compaq.comとsales.compaq.comの複製トポロジを拡張してGCへアクセスできるようにし、ドメイン情報をドメイン全体に複製します。フォレスト内でのGCへの複製は、ドメインNCの情報をDC間で複製するのと同じ方法で行いますが、NCがフォレスト ワイド スコープを持つ点が異なります。GCへの複製は、SMTPを使って実行します。これは、リンクが低速な場合に最適です。

複製では、DCとGCには次のような関係があります。

- ドメイン コントローラは、3つの名前付けコンテキストを複製するので、DCの複製は、名前付けコンテキストの合計となります。次の式は、ドメイン コントローラを含む複製は、3つの名前付けコンテキストの変化によって影響を与えられることを示しています。

$$R_{dc} = NC_{domain} + NC_{schema} + NC_{config}$$

- グローバル カタログ複製は、通常のDCの複製と同じです。これは、GCは、自分のドメイン内ではDCだからです。各GCは、フォレスト内の他の全ドメインに対して、partial ドメイン NCを複製します。partial ドメイン NCは、ドメインNC内の全オブジェクトを含みますが、各オブジェクトの属性のサブセットのみをカバーします。属性をGCへ複製するかどうかは、publishプロパティで指定

します。これは、スキーマ マスタで設定されます。プロパティを設定すると、属性はすべてのGCへ提供されます。したがって、GCの複製とDCの複製との相違点は、次の式で示されます。

$$R_{gc} = R_{dc} + NC_{partial\ domain} * (n - 1)$$

n = フォレスト内のドメインの数

フォレストにドメインが1つしかない場合は、すべてのDCをGCとして設定しても、トラフィックは増加しません。これは、ドメインの各コントローラは、いずれにしても全オブジェクトの属性をフルセットで複製するためです。

サイトにGCが存在する場合、サイト内のすべてのDCはGCとして構成できます。GCや別サイトのDCから情報を複製するブリッジ ヘッド サーバが必ず1つ存在するので、サイト間複製を実行するGCは1つのみとなります。同じサイトにある他のコントローラは、サイト内複製を使用します。サイトは接続性に優れたサブネットで構成されているため、同じサイト内の全GCとのGC複製は、必ずRPCで実行されます。

DCは、スキーマ/構成データを、他のGCから複製できません。これは、GCに含まれているのは、他のドメインからの読み出し専用複製であり、オブジェクトに対する権限はないからです。アドミニストレータは、GCで読み出し専用複製を変更することはできないので、DCは、GCを使って複製トポロジを生成し、データを複製することはできません。

サイト トポロジ

サイト トポロジ設計の基礎

Windows 2000では、サイト トポロジは、物理ネットワークの上にある論理モデルです。Windows 2000は、ネットワークとネットワーク特性に関する情報をOSFPから取得する際、ネットワーク ルータへアクセスしません。つまり、Windows 2000は、ネットワークの可用性や複製のコストなどに関する情報については、物理ネットワークを検出するのではなく、サイト トポロジを使用します。KCCは、サイト トポロジを使用して、DC間の複製に最適なパスを生成します。

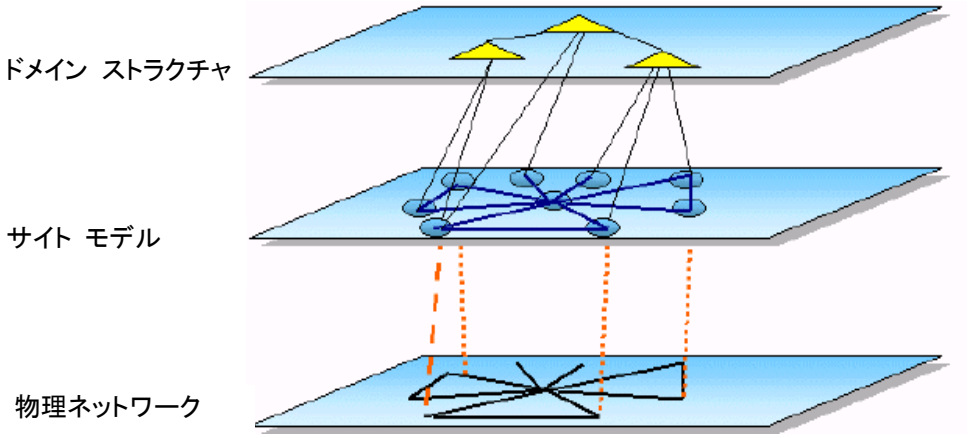


図18：サイト トポロジ

Windows 2000インフラストラクチャの設計者は、基礎となるネットワーク トポロジを正確に反映したサイト トポロジをモデル化しなければなりません。図18に示すように、Windows 2000サイトの設計では、次の3つの層を考慮する必要があります。

- 最下層にあるのは物理ネットワークです。
- 中央にある層がWindows 2000サイト トポロジであり、物理ネットワークを反映します。
- Windows 2000ドメイン ストラクチャは、一番上の層です。ドメインは、サイト トポロジによって複製を行うので、モデルに従ったデザインが必要になります。複製の制限が理由で、ドメインを分割する場合や、逆に物理ネットワークが複製に制限を与える場合もあります。

ドメイン ストラクチャの設計とサイト トポロジの設計を区別することが重要です。ドメインはオブジェクトを含むのに対して、サイトはユーザ コミュニティの位置を意味します。ドメインは、サイトに複製を配置することによって、サイト内にマッピングされます。1つのサイトは、ドメインの一部ではなく、ドメイン全体を含みます。サイトはDCを含み、ドメインの複製全体です。GCをサイトに配置すれば、サイトはActive Directory全体を使用できるようになります。サイトにDCやGCが含まれない場合、サイト内に使用可能なオブジェクトが存在しないので、このサイトは使用できません。

サイト トポロジは、複製に必要な帯域幅という点で、ドメイン ストラクチャに影響を与えます。Active Directoryはスケーラビリティに優れているため、ほとんどの場合、エンタープライズ内にドメインは1つで十分です。ただし、通常の場合、様々な理由でドメインを分割する必要があります。Active Directory複製で使用するネットワーク帯域幅も、このような理由の1つです。Active Directoryのオブジェクトの数

や、変更の頻度によっては、大量のトラフィックが生成されます。サイト トポロジを微調整することによって、複製に必要なネットワーク トラフィックを軽減できます。

サイト リンク

サイト間複製には、明示的なサイト リンクを作成する必要があります。サイト リンクとは、名前からもわかるように、2つのサイトを接続するものです。サイト リンクは、2つのサイト間のネットワーク コミュニケーションです。

サイト リンクには、コストが指定されています。コストは、サイト間でのデータ複製が、どの程度簡単にできるかを評価するものです。複数のサイトをサイト リンクで接続する場合、KCCは、コストとサイト リンクの可用性に基づいて、DC間で複製を行う際、どのような接続オブジェクトを生成するべきかを判断します。

KCCは、異なるサイトにあるDC間で、分散ツリーの接続オブジェクトを生成します。分散ツリーは、ループを構成しません。サイト間の複製は、ストア フォワード モデルで実行します。これに対して、サイト内複製はプル モデルを採用しています。

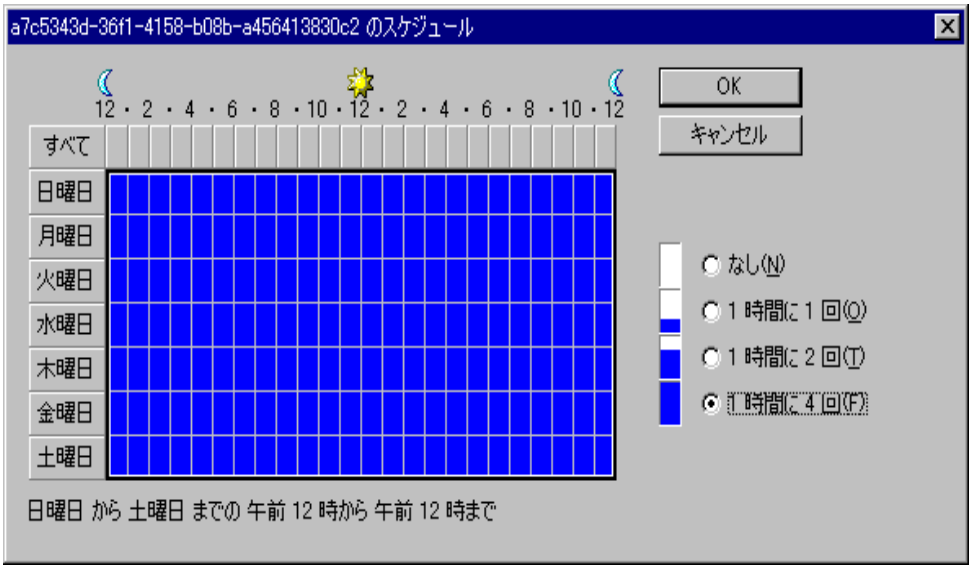
図19：サイト リンクのプロパティ

サイト内複製とサイト間複製に異なるモデルを採用する点は、Exchange Server 5.5のスキーマに類似しています。Exchange Server 5.5では、サイト内ではプル モデル、サイト間ではプッシュ モデルを採用します。サイトは、接続性に優れたIPサブネットの集合であり、サイト リンクは、サイト間の論理的接続です。サイト リンクは、ネットワークに近いものであり、サイト間接続に似た特徴を持っているとみなされます。つまり、サイト リンクはWANリンクに相当します。

接続オブジェクトとサイト リンクの違いを理解してください。オブジェクトは、ドメインコントローラを接続するものであり、KCCが作成するか、アドミニストレータが手動で作成します。サイト リンクは、アドミニストレータが作成し、KCCはこれを元に、サイト間の複製に必要なコストやネットワークの可用性を決定します。図19は、サイト リンクのプロパティを示します。画面左側に、使用可能なサイトのリストが表示されます。アドミニストレータは、このサイト リンクを使ってどのサイトを接続するかを選択できます。図19では、2つのサイトを選択できます。

サイト リンクには、コストとスケジュールが指定されており、KCCはこれに基づいて複製トポロジの生成や最適化を行います。図20は、サイト リンクのスケジュールを示します。

図20：複製 スケジュール



コスト ファクタは、1～32767の範囲の数値であり、サイト間の複製にかかるコストを示します。デフォルト値は100であり、これが最小値となります。ネットワークの接続性が向上するほど、リンク間の複製は簡単になります。KCCは、コスト ファクタを使って、複製トポロジの作成や最適化を行います。図19では、ValbonneとSeattleの2つのサイトが表示されています。複製のコストは100で、複製の頻度は15分おきに設定されています。

図21は、KCCがサイト間の複製コストをどのように計算するのかを示しています。基本的には、2つの遠隔サイト間で、各サイト リンクのコストを合計し、この合計コストを使って最適な（最もコストの低い）ルートを決定します。

この例では、NYサイトからChicagoサイトまでのコストが1で、Chicagoサイトから Atlantaまでのコストが8だとすると、DC1からDC3までの複製のコストは9となり、 DC1からDC4へのコストは10となります。

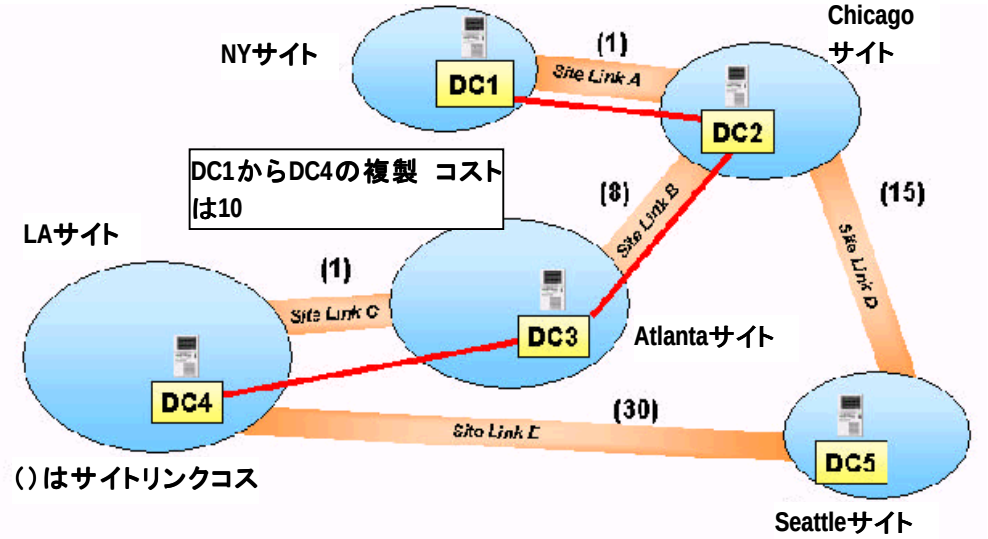


図21：コストのレプリケート

サイト リンクには、複数のサイトを追加できます。つまり、同じコストとスケジュールを共有するサイトが、サイト リンク内に複数存在することが可能です。KCCは、このようなサイトを同じように扱い、リソース オブジェクトの作成や管理を行います。

ブリッジヘッド サーバ

ブリッジヘッド サーバ（BHS）とは、他のサイトのDCとの複製を実行するDCです。サイト内のDCは、すべてがBHSとは限りません。各サイトには、ISTG（Inter-Site Topology Generator）の機能を持つDCが1つ存在します。ISTGは、サイト内のBHSを検出し、これに対して接続オブジェクトをチェック、作成します。ISTG機能は、他のFSMO機能のように、移動できません。アドミニストレータは、どのDCにISTG機能を持たせるかを選択することはできません。サイト内にあるすべてのDCが、この機能を実行できます。

どのDCにISTG機能を持たせるかを決定するアルゴリズムでは、各DCが、構成 NCで定義されているサイト内のドメイン コントローラのリストをチェックし、現在使用不能なドメイン コントローラを削除してから、残りをGUID順に並べます。次に、DCは、昇順のリストの最初のドメイン コントローラを選択します。ドメイン コントローラは、同じアルゴリズムに従っているので、ISTG機能を持つDCは1つに決まります。

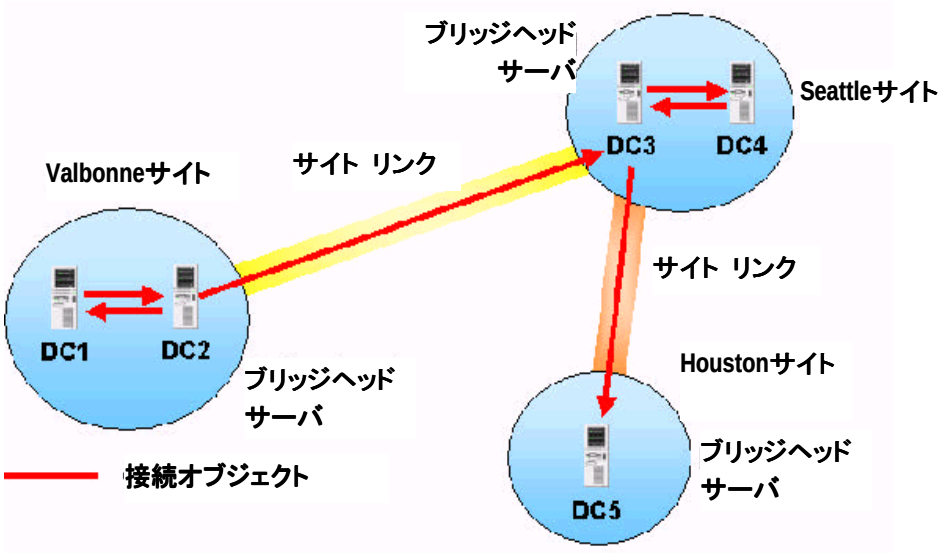
サイトのトランスポート1つあたり、ブリッジヘッド サーバは1つです。したがって、IPあたりブリッジヘッド サーバ1つ、SMTPあたりブリッジヘッド サーバ 1つを指定できます。図22は、ドメイン内のドメイン コントローラのプロパティを示しています。この画面は、[Active Directory サイトとサービス]スナップインのデフォルトサイトとサーバ フォルダで、DCを右クリックすると表示されます。

図22：優先的なブリッジヘッド サーバ

このDCのトランスポートとして、IPやSMTPよりもRPCを選択したということは、ブリッジヘッド サーバとして優先されます。

図23は、ブリッジヘッド サーバを示しています。サイトは、Valbonne、Seattle、Houstonの3つです。Valbonneサイトには、DC1とDC2の2つのDCがあります。この2つのDCは、相互に複製のリングを構成しています。DC2は、サイトのブリッジヘッド サーバです。つまり、アドミニストレータは、このサーバ上のトランスポートをサイト間複製と定義したことを意味します。

図23：ブリッジヘッド サーバ



Seattleサイトには、DC3とDC4の2つのコントローラがあり、似た構成となっています。DC3は、ブリッジヘッド サーバです。接続オブジェクトは、DC2とDC3間にしかない点に注意してください。この2台のサーバのうち1台が使用不能になると、Seattleサイトへの複製は不可能になります。だたし、サイト リンクは推移的であるため、Houstonサイトへの複製には問題ありません。つまり、ブリッジヘッド サーバがSeattleサイトで稼動中のDCを検出できなければ、DC2とDC5間に接続オブジェクトが作成されます。次の節では、このメカニズムを更に詳細に説明します。

サイト リンクのインプリメント

図24は、4つのサイトで2つのドメインを構成した様子です。compaq.comドメインに属すコントローラは濃い色、sales.compaq.comドメインに属すコントローラは薄い色で示しています。

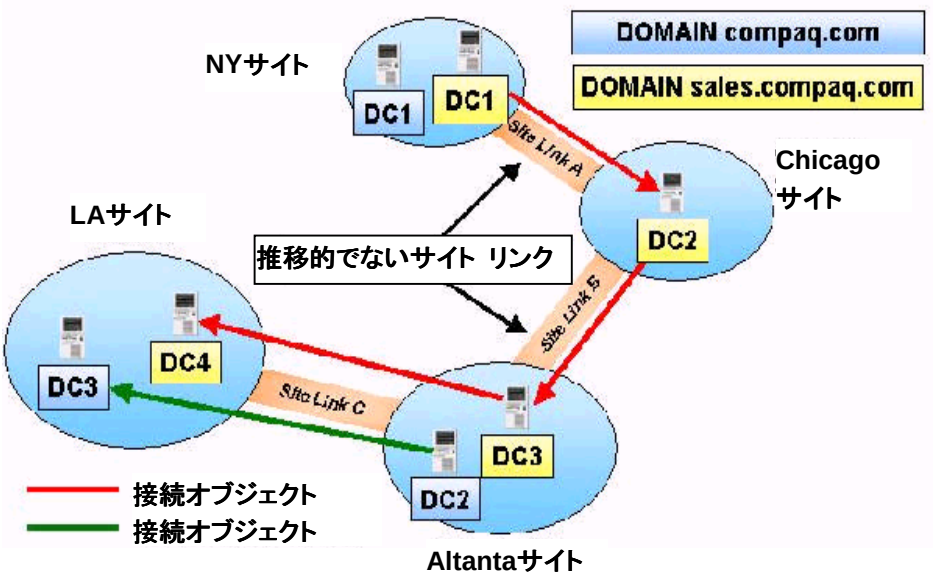


図24：サイト リンクのインプリメント

サイト リンクがサイト間で使用可能になると、様々なKCCが接続オブジェクトを作成します。複製は、sales.compaq.comドメインで実行されます。各サイトでは、DCが1つ使用可能になり、分散ツリー トポロジが生成されます。compaq.comドメインをみると、複製が実行されるのは、DC2とDC3間のみです。これには、次のような理由があります。

- compaq.comのDC2サーバは、複製情報を他のドメインへ転送できません。
- NYとLAサイト間には、直接のサイトリンクはありません。NYとChicago間、ChicagoとAtlanta間のサイト リンクは、推移的ではありません。この例では、compaq.comからは、DC1は同じドメイン内のDC2へアクセスできません。

複製トポロジは、次の2つの方法で修正できます。

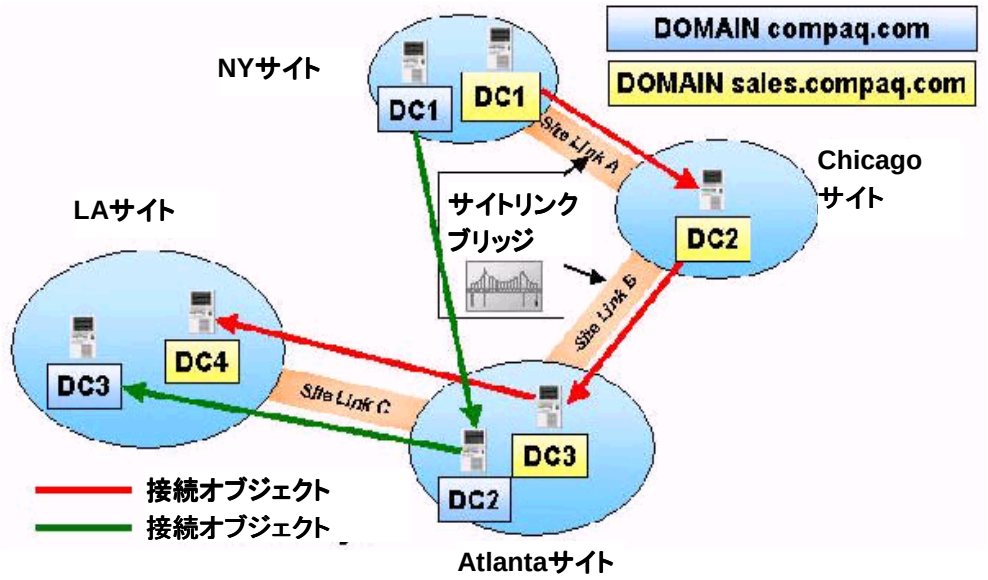
- サイト リンクの設定を変更し、推移的にします。
- サイト リンクAとサイト リンクB間に、明示的なサイト リンクを作成します。

次の節では、サイト リンク ブリッジの詳細を説明します。

サイト リンク ブリッジ

サイト リンク ブリッジは、2つのサイト リンクを接続し、ブリッジを構成します。サイト リンク ブリッジは、2つのネットワークを接続するルータに相当します。複製の観点からいえば、このネットワークはサイト リンクとなります。サイト リンク ブリッジは、サイト リンクを接続し、明示的なサイト リンクを持たない2つのサイト間で、推移的で論理的なリンクを構成します。

図25：サイト リンク ブリッジ



前の節で使用した例では、サイトが4つありました。NYサイトには、Atlantaリンクへの明示的なサイト リンクはありません。サイト リンク ブリッジでこの2つのサイトを接続することにより、NYサイトのDCは、AltantaサイトのDCに対する接続オブジェクトを生成し、複製を実行できます。

Windows 2000では、サイト リンクは推移性を持っています。つまり、ほとんどの場合、サイト リンク ブリッジは必要ありません。サイト リンクの推移性を無効にするには、複製トランスポートのプロパティを選択します。推移性は、すべてのサイト リンクに適用されるので、オフに設定すると、推移性がなくなり、明示的なサイト リンク ブリッジを作成しなければなりません。

Windows 2000では、サイト リンクの推移性はデフォルトでオンになっています。Windows 2000を使用する中小規模のネットワークでは、KCCが使用可能な任意のサイトでDCを検出しても問題はなく、むしろ推奨されているからです。これに対して、大規模ネットワークでは、オフにすることをお勧めします。複製トポロジは、明示的なサイト リンク ブリッジで微調整し、サイト間の接続オブジェクトの作成を管理する必要があります。本書の後の節では、例を挙げて説明します。

図26は、IP複製トランスポートのプロパティを示しています。アドミニストレータは、［*サイトリンクをすべてブリッジ*］チェック ボックスをクリックし、サイト リンクの推移性をオンにします。他のトランスポートについても、同様に設定できます。

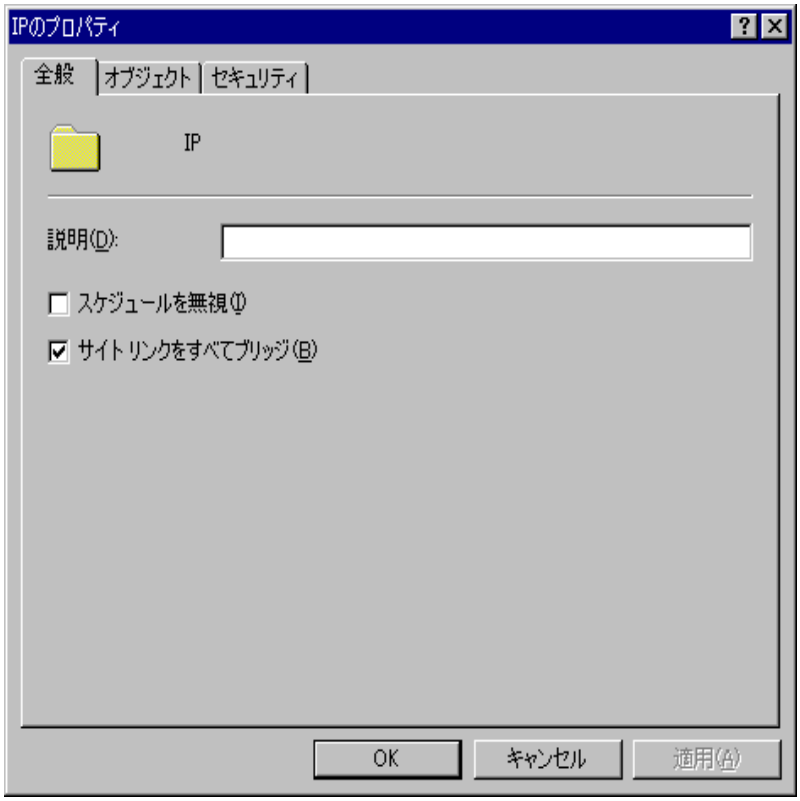


図26：IPトランスポートのプロパティ

複製トポロジの作成

複製トポロジの作成は、次の3つの基本的なアプローチによって行います。

- 1つ目のオプションとして、KCCをオフにして、DC間の接続オブジェクトを手動で作成します。トポロジを完全に管理することができますが、サーバがダウンすると、KCCは別ルートを検索しないので、アドミニストレータは代わりの接続オブジェクトを作成しなくてはなりません。大規模な組織では、明示的な接続オブジェクトの管理やドメイン コントローラのモニタにかかるコストが非常に高くなるので、このソリューションは現実的ではありません。
- 2つ目のオプションとして、KCCにすべての処理を任せます。アドミニストレータは、サイトと、サイトを含むサイト リンクを作成します。これは、完全なメッシュ ネットワークと非常に似たものになります。複製のコストと可用性は、すべてのサイトで一定になります。問題点は、作成される複製トポロジは、基礎となるネットワーク トポロジを必ずしも反映しているわけではなく、経済的ではない点です。
- 3つ目のオプションは、ネットワーク情報をKCCに提供し、サイトや、コストとスケジュールを指定したサイト リンクを作成します。このオプションの目的は、物理ネットワークの制約を考慮し、ピーク時に複製が発生しないようにします。このオプションでは、アドミニストレータは、複製の微調整や最適化が可能なので、最も柔軟性が高く、強力なアプローチだといえます。このオプションでは、サイト リンクの推移性をオフにし、推移性を持たせるためにサイト リンク ブリッジを作成して、状況によってリダンダントなサイト リンクを作成します。このオプションでは、サイト トポロジを慎重に設計する必要があります。

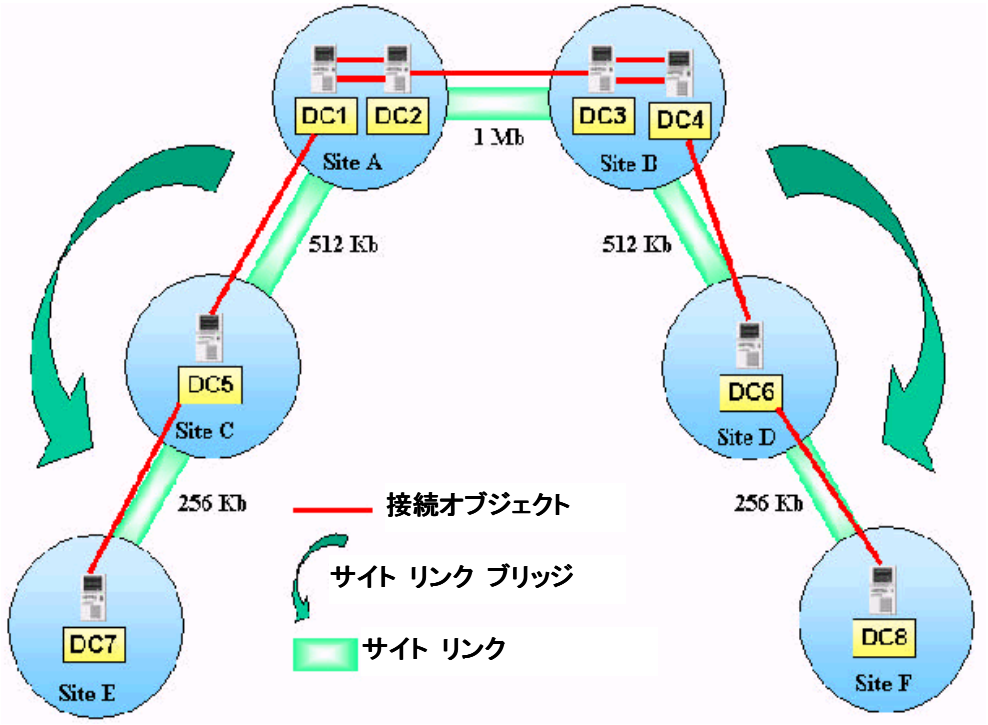
ネットワーク トポロジとサイトの設計

この節では、各種のネットワーク トポロジを紹介し、様々なサイト設計を分析します。また、リダンダント トポロジや、サイト リンク ブリッジがサイト設計で持つ役割についても説明します。

ツリー トポロジ

ツリー トポロジは、複数の層を持つサイトで構成されます。図27は、代表的なツリー トポロジであり、最初の層には、1つ以上の中央サイトがあって、これが2番目の層のサイトへ接続しています。第2層のサイトは、第3層などへと接続します。一般的に、下の層になるほど、帯域幅は低くなります。ツリー トポロジの目的は、中央管理にあります。複製された情報は、1つの層から別の層へと転送され、第1層を除いて、同じレベルの層ではやりとりされません。

図27：ツリー トポロジ



ツリー トポロジでは、異なる層のサイト間にサイト リンクを作成します。サイト リンクの推移性はオフにします。

サイト リンク ブリッジは、異なるレベルのサイト リンク間で作成します。これは、中間層のサイトに、複製した情報の格納や転送ができるDCが存在しない場合、KCCが次のレベルのサイトにアクセスするためです。

リング トポロジ

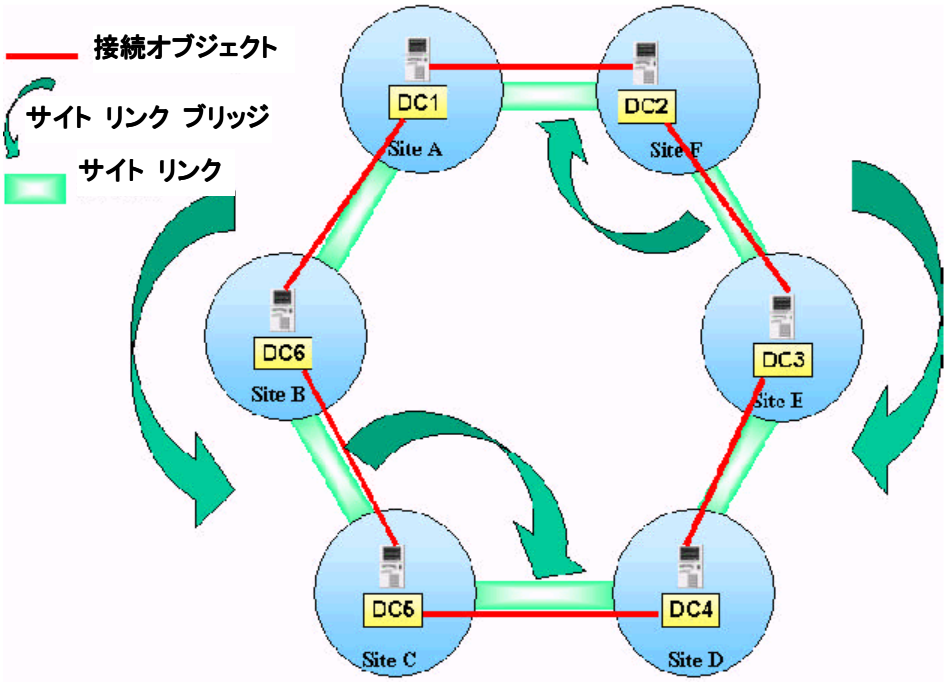
リング トポロジは、図のように、サイトの相互接続によって、ループやリングを構成します。リング トポロジの利点は、異なるサイトのDCが、複製した情報を格納し、隣のサイトへ転送できるという点です。情報は、リング上を同時に転送されます。

リング トポロジでは、サイト リンクの推移性をオフにします。推移性がオンの場合、あるサイトのDCが使用不能になると、KCCは別サイトにあるDCを検索してしまうので、長期的にはKCCはメッシュ トポロジを形成し、管理が困難になります。

サイト リンクの推移性をオフにし、3サイトごとに明示的なサイト リンク ブリッジを作成することをお勧めします。1つのサイトが使用不能になったら、KCCはサイトリンク ブリッジを使用し、次のサイトのDCに接続します。この方法では、リング トポロジを維持できるので、強力なアプローチだといえます。

さらに、サイト リンク ブリッジ間を完全に二重化し、リング トポロジを維持するために、3サイトごとに共通のサイトを1つ持たせます。これによって、KCCは、サイト リンク ブリッジのサイト内で接続オブジェクトを作成し、メッシュ トポロジになるのを防ぐことができます。

図28：リング トポロジ



メッシュ トポロジ

メッシュ トポロジは、すべてのサイトを相互接続します。たとえば、サイトが4つある場合、各サイトに、他のサイトへの明示的なサイト リンクが存在します。この場合、各サイトは、3つのサイト リンクのメンバとなります。必要なサイト リンクの数は、N*（N-1）で計算できます（Nは接続サイトの数）。このトポロジは、スケーラビリティや管理性に優れているとはいえません。サイト リンクの推移性を有効にすれば、サイト接続に必要なサイト リンクを減らすことができます。推移性によって、KCCは任意のサイトにあるDCへの接続オブジェクトを作成できます。

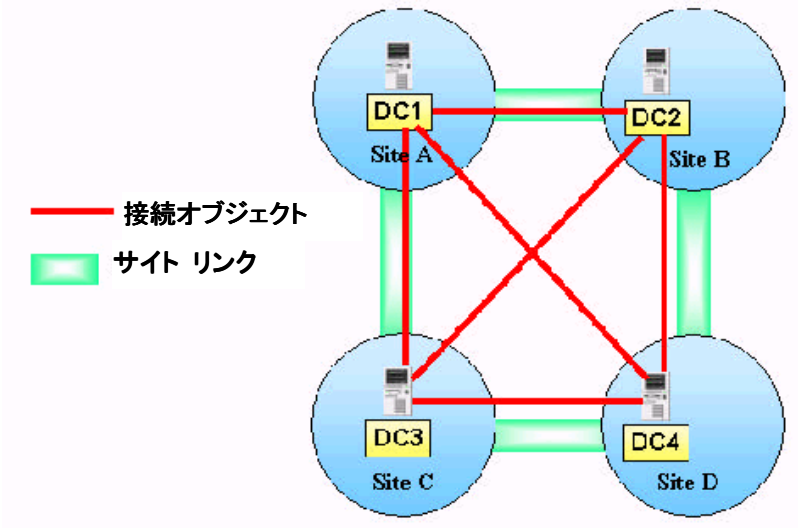


図29：メッシュ トポロジ

複製の課題

緊急複製

各種Active Directoryオブジェクトの複製には優先順位を割り当てることはできませんが、優先的に複製が行われるオブジェクトがあります。たとえば、アカウントが無効になり、ロックアウトされた場合、セキュリティ上の問題があるので、この変更内容はできるだけ迅速にすべてのDCへ複製する必要があります。

また、緊急複製は、RIDプールの割り当てにも適用されます。RIDプールがないと、新しくセキュリティ プリンシパルが作成されても、DCはSIDを割り当てられないからです。

緊急複製は、同じサイト内のDC間でしかトリガされません。緊急複製は、通知によって開始されます。サイト間ではStore and Forward複製モデルが採用されるので、通知はなく、緊急複製は行われません。

競合

マルチ マスタ レプリケーション モデルでは、同じ属性を2つのDCが同時に更新しようとする可能性があります。このような場合、両方のDCでUSNがインクリメントされます。複製が発生すると、変更されたデータは複製パートナーへ送信され、複製の競合が発生します。属性は何回か更新されているので、どの更新内容が有効であり、競合が解決された後の属性の値を判断する必要があります。

Active Directoryは、2つの複製オペレーションのバージョン番号を比較し、大きい方を選択します。バージョン番号が同じ場合は、タイムスタンプを比較し、新しい方を選択し、これよりも古いものは破棄します。ドメイン内のすべてのDCの時間は同期しています。同時に2つのDCでオペレーションが開始された場合、DCのGUIDをチェックし、新しく作成されたGUIDを選択します。ただし、このような状況は非常にまれであり、一般的には、最後に書き込みを行った方を選択します。

また、1つのDC上のOUにオブジェクトを作成する場合にも、競合が発生します。たとえば、アドミニストレータが異なるDC上で全く同じオブジェクトを削除する場合があります。削除されたオブジェクトは、削除OUに作成されます。このような競合は、どのように解決すればよいでしょうか。OUでの削除オペレーションは、ドメイン全体に複製され、オブジェクトはLost & Foundコンテナへ移動します。このコンテナは、［Active Directory ユーザとコンピュータ］スナップインのアドバンス機能で

表示できます。

さらに、あるDCで削除したコンテナに、オブジェクトを作成する場合にも、競合が発生します。たとえば、DC1のOUにユーザを作成し、同時にDC2上でOUを削除します。2つのドメイン コントローラ上で、RDN（relative distinguished name）を持つユーザ オブジェクトを2種類作成した場合、オブジェクト作成名の競合が発生します。この場合も、タイムスタンプとDC GUIDを比較して、競合は解決されます。

多数のDCが存在するネットワークで競合が発生した場合、2つのDCが、競合している変更内容をすべてのDCへ複製します。各DCは、競合している複製リクエストを受信し、競合は各DCが解決します。すべてのDCが同じアルゴリズムに従って競合を解決し、競合の結果をイベント ログに記録します。

SMTP複製プロセス

Windows 2000には、基本的なSMTP機能が含まれ、これによって、サーバはトランスポート経由で相互に情報を送信できます。複製という観点では、SMTPは、スキーマと構成 NCの複製や、ドメイン NCをグローバル カタログへ複製するためのトランスポートとして使用できます。ISM（Inter-Site Messaging ）と呼ばれる層は、非同期トランスポートのロードに使用します。トランスポートは、2つの複製パートナー間の接続オブジェクトのプロパティとして定義されています。接続オブジェクトは、KCCまたはアドミニストレータが作成し、サイト リンクで定義されているトランスポートに基づきます。接続オブジェクトは、サイト リンクで定義されているトランスポート以外は使用できません。

Windows 2000では、トランスポートはロード可能なDLLとして提供されています。Windows 2000で提供されている非同期トランスポートは、SMTPだけですが、各ベンダは独自のトランスポートDLLを開発し、Active Directoryの次のロケーションで提供しています。

`CN=Inter-Site Transports,CN=Sites,CN=configuration,DC=domain-`*`name`*

SMTPトランスポートDLLは、CDO（Collaborative Data Objects）を使用します。CDOは、SMTPメッセージの作成と送信に使用するCOMインタフェースを集めたものです。Exchange “Platinum”をWindows 2000サーバにインストールすると、CDOライブラリが更新され、メール ボックスやサーバ間のSMTPトランスポートが有効になり、パーソナル間メッセージ送信が可能になります。DCでSMTPトランスポートを使用したい場合は、CDOライブラリをアクティブにするために、IISをインストールする必要があります。

SMTPトランスポートは、SMTPを使用するすべてのDCに対して、証明書を発行する必要があります。SMTPは、この証明書を使って、2つの複製パートナー間の相互認証を行います。証明書には、DCに割り当てられた公開キーが含まれています。Certificate Serverは、プライベート キーと公開キーをペアで発行します。

プライベート キーは、キーのオーナがメッセージの署名に使用します。メッセージの受け手は、証明書に格納された公開キーを使って、メッセージの認証を行います。プライベート キーと公開キーには、数学的な関連性があります。キーはオーナが持っているので、メッセージの受信先は、信頼関係のあるCertificate Serverに対して、公開キーを検証します。公開キー インフラストラクチャの詳細は、ホワイト ペーパ『Security Fundamentals as implemented in Windows 2000』（Jan De Clercq著）を参照してください。

2つのDCは、複製が必要だと判断すると、証明書を使って相互認証を行います。証明書はグローバル カタログに格納されるので、別のドメインのDCも必要な証明書を取得できます。

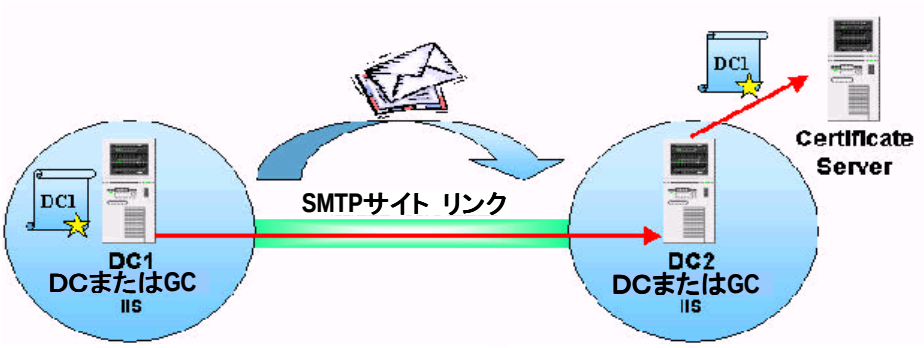
送信元のDCは、送信先となるDCに送るデータを準備します。データが50KBを超える場合は、圧縮し、ISMレイヤにSMTPトランスポートDLLのロードを要求します。

このDLLは、DCの公開キーを使って、圧縮データに署名し、CDOを使ってメッセージを作成し、ターゲットDCへ送信します。

ターゲットDCは、メッセージを受け取り、Certificate Serverを使って署名をチェックします。つまり、Certificate Serverに対して、証明書の公開キーは、送り手のDCに発行されたものかどうかをチェックします。チェックが完了したら、ターゲットDCは、メッセージを処理し、データはローカルに複製されます。

SMTPは、拡張された低速ネットワーク リンクでの動作を考慮して設計されています。したがって、このトランスポートは、低速で予測不能なネットワーク リンク上での複製に使用できます。実際、公衆インターネット リンクを使ったイントラネット間のメッセージ転送が可能です。

図30：SMTPサイト リンク



①. DCがデータを圧縮

②. ISMへデータを渡す

③. ISMはCDOをロード

④. CDOはSMTPトランスポートをロード

⑤. SMTPはDCプライベート キーを使ってデータに署名、メッセージを送信

⑥. SMTPサーバがメッセージを受信

⑦. SMTPがCertificateサーバへアクセスし、証明書の公開キーの認証をチェック

⑧. 問題ない場合は、ISMへデータを渡す

⑨. SIMはDCへデータを渡し、処理

⑩. SMTPはDCプライベート キーを使って

図30： SMTP複製を使ったDCによる処理

結論

大規模エンタープライズ向けのWindows 2000ベース インフラストラクチャの構築において、慎重に調整を行った複製トポロジは、重要な役割を果たします。Active Directoryは、次のようなアプリケーションで既に使用されています。

- Windows 2200で提供されているDNSバージョンは、データの格納にActive Directoryを使用し、他のDNSが更新内容を受信したことを確認するために、複製メカニズムを使います。
- Microsoft Exchange Serverの“Platinum”バージョンは、Active Directoryを使って、メールボックスやサーバに関する情報の格納を行います。
- その他のMicrosoft BackOfficeアプリケーションは、Windows 2000のアップグレード後に、Active Directoryを使用する予定です。

また、カスタマ ソリューションでも、データとActive Directoryを統合することが可能です。Active Directory対応アプリケーションは、スキーマを変更して独自のオブジェクト クラスを追加できます。このクラスのインスタンスは、既存の複製トポロジを使って、複製を行います。

Windows 2000や多くのアプリケーションがActive Directoryを使用しているため、複製トポロジを慎重に設計し、膨大な情報に対応できる基礎作りをする必要があります。Active Directory導入のプランニングは、Windows 2000の導入プロジェクトにおいて、重要な位置を占めています。