

System Administration Guide



Version 3.1

著作権情報

Copyright © 2004 Xpiori LLC
All rights reserved

法律上の特記事項

このマニュアルの情報は Xpiori LLC により提供されています。Xpiori LLC の知る限りにおいて、このマニュアルには情報処理テクノロジーの現在の状態に関する情報が収められていますが、このマニュアルおよびその記載情報は現状のまま提供されいかなる保証もされません。Xpiori LLC は、このマニュアルに記載されている情報および製品について、商品性、特定目的適合性、権利非侵害の保証を含め、かつ、それに限定されない一切の保証の責任を、明示的にも暗示的にも一切負うものではありません。このマニュアルの記載内容ならびに特定の環境に対して提案される用途は、詳細な技術的調査なしに信頼できることを保証するものではありません。このマニュアルの記載情報および製品に関する記述の使用またはそれらへの依存に起因する一切の損害について、Xpiori LLC はいかなる責任も負いません。製品および技術情報は、予告なく変更されることがあります。

特許

NeoCore™ の技術は以下の特許権により保護されています。

米国特許番号 5,742,611 (1998 年 4 月 21 日) - Client Server Network and Method of Operation - Christopher Lockton Brandin

米国特許番号 5,942,002 (1999 年 8 月 24 日) - Method and Apparatus for Generating a Transform - Christopher Lockton Brandin

米国特許番号 6,157,617 (2000 年 12 月 5 日) - Method and System of Network Packet Accounting - Christopher Lockton Brandin

米国特許番号 6,167,400 (2000 年 12 月 26 日) - Method of Performing a Sliding Window Search - Christopher Lockton Brandin

その他の米国特許および国際特許が出願中です。

商標

NeoCore™、NeoCore™ XML Information Management System (XMS)、Pattern-Based Associative Processing、および Digital Pattern Processing は Xpiori LLC 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。他のすべての商標は、それぞれ各社の商標です。

著作権

Copyright © 2004 Xpiori LLC. All rights reserved.

目次

概要	5
----	---

Xpiori NeoCore XML Information Management Systemバージョン 3.1	5
新機能	5
XMSアーキテクチャ概要	6
クライアント側の概要	7
サーバー側の概要	8

インストール	10
--------	----

インストール要件	10
ハードウェア要件	11
ソフトウェア	11
NeoCore XMSのインストール/アクセス/アンインストールの概要	12
Solarisでのインストールの準備 - NeoCore XMSをインストールする前に	12
Windowsでのインストールの準備 - NeoCore XMSをインストールする前に	13
NeoCore XMSのインストール	14
NeoCore XMSのアンインストール	20
Windows上のNeoCore XMSのアンインストール	20
Solaris上のNeoCore XMSのアンインストール	21
SolarisでのXMSの複数インスタンスのインストールおよび構成	22
XMSの開始	23
XMSの停止	24
NeoServerの状態の確認	26
NeoCore XMSの再インストール	26
リモートクライアントのインストール	27
NeoServerホストに対する追加NICの増設	27
安全なパスワードリセット	28
XMSAdminコマンド	29
NeoCore XMSへのアクセス	30

エラーメッセージ	32
----------	----

NeoServer管理コンソール	41
------------------	----

概要	41
コンソールの起動	41
コンソールパスワードの変更	42
XMSコンソールGUI	42
[Database Access] タブ	49

データベース管理 74

概要	74
NeoCore DB管理コマンド構文	74
コマンド構文:	74
XMLデータベースの管理	76
データベースの作成	77
空のデータベースの作成	78
データベースの削除	79
データベースのクリア	80
データベースのバックアップ	81
データベースの復元	82
データのエクスポート	83
データのインポート	86
データベースのインデックス再構築	89
データベースファイル再配置コマンド	91
NeoXMLUtilsコマンド	92
ホットバックアップ	94
Veritasを使用したSolarisボリューム管理	95
XMS Loader	100
手動（オフライン）ファイル拡張	102
ファイル拡張コマンド	103
領域自動拡張	104
SolarisとWindowsの領域自動拡張の比較	105
モニタリング	105
パフォーマンス改善のためのチューニング	116
NeoCore XMS構成変更の要件	120
ファイルの構成	120
NeoCore XMSインデックス	120
NeoDatabase.xmlファイルのパラメータの説明	121
NeoXDBRuntime.xmlファイルのパラメータの定義	136
Windowingのパフォーマンスに関する優先順位	151

アクセス制御とユーザー管理 156

アクセス制御（AC）のシステム図	156
アクセス制御（AC）のフィルタリング	156
アクセス制御の有効化	157
アクセス制御の呼び出し	157
NeoCore XMSのACルール of 動作	158
ルール構文	159

WEBDAV 161

WebDAVの概要	161
WebDAVの仕組み	161
WebDAVのクライアント	162

APIアクセス	162
XMS WebDAVへの接続	162
認証	163
許可	164
文書の格納、検索、管理	165
コレクション	165
プロパティの設定	166
ロック	166
バージョン管理	166
WebDAV XMLスキーマ	167
WEBDAVの参考資料	167
 用語集	 168

索引	173
----------	-----

概要

Xpiori NeoCore XML Information Management System バージョン 3.1

Xpiori NeoCore™ XML Information Management System（この資料では **XMS** と呼びます）は、情報管理のための高速で非常に柔軟なツールです。NeoCore XMS は Extensible Markup Language (XML) データとそのコンテキストの格納と検索を行います。XML データベースを使用すると、他のソリューションに伴う障害の多くを排除できます。

XMS には、**Server 管理コンソール**、**NeoServer**、および **API** が含まれます。

この『**System Administration Guide**』では、機能の概要を示し、NeoCore XMS のインストール、構成、使用、保守、さらにはデータ整合性の保持について詳しく解説します。

新機能

- **WebDAV** - WebDAV は、World Wide Web を読み取り専用の媒体から読み取り/書き込みが可能な媒体へと変換するための HTTP に対する一連のプロトコル拡張機能です。WebDAV は、コラボレーションの手段として幅広く採用されており、すでに Microsoft、Adobe、Altova、EMC、Oracle、IBM などの製品でサポートされています。XMS 用の Xpiori WebDAV インターフェースは、DAV プロトコルを実装することによって、XMS をバックエンドとして使用しながら、DAV サーバーに格納した XML に対するハイパフォーマンスアクセスを実現しています。
- **インポートとエクスポートの拡張** - XMS v3.1 では、インポートの実行時にもデータベースファイルの自動拡張機能が用意されているので、データベースのサイズを事前に構成する必要がありません。エクスポートも拡張され、より大きなファイルを生成できる設計になったので、エクスポートしたファイルの管理がシンプルになっています。
- **名前空間** - XMS では、名前空間を指定したタグと名前空間を指定しないタグを組み合わせたクエリをサポートするようになりました。
- **パフォーマンス** - XMS (Windows 版) と XMS (Linux 版) では、ディスクベース (Windowing) モードでの実行時に、メモリー管理の拡張機能により更新処理のパフォーマンスが改善されています。

お客様のコメントをお送りください

このガイドについてのコメントやご提案については、Mitsui NeoCore Center support@neocore.jp 宛メールにてお送りください。

対象読者

このマニュアルは、Windows や Solaris オペレーティングシステムに関する理解を前提として、NeoCore XMS のインストールと構成を担当するシステム管理者を対象としています。

関連資料

NeoCore XMS のアーキテクチャ、XML 操作に XPath と XQuery を使用方法、および Java/C++/COM/HTTP の API については、『*NeoCore XMS System Programming Guide*』を参照してください。

表記規則

このガイドで使用する表記規則は、以下のとおりです。

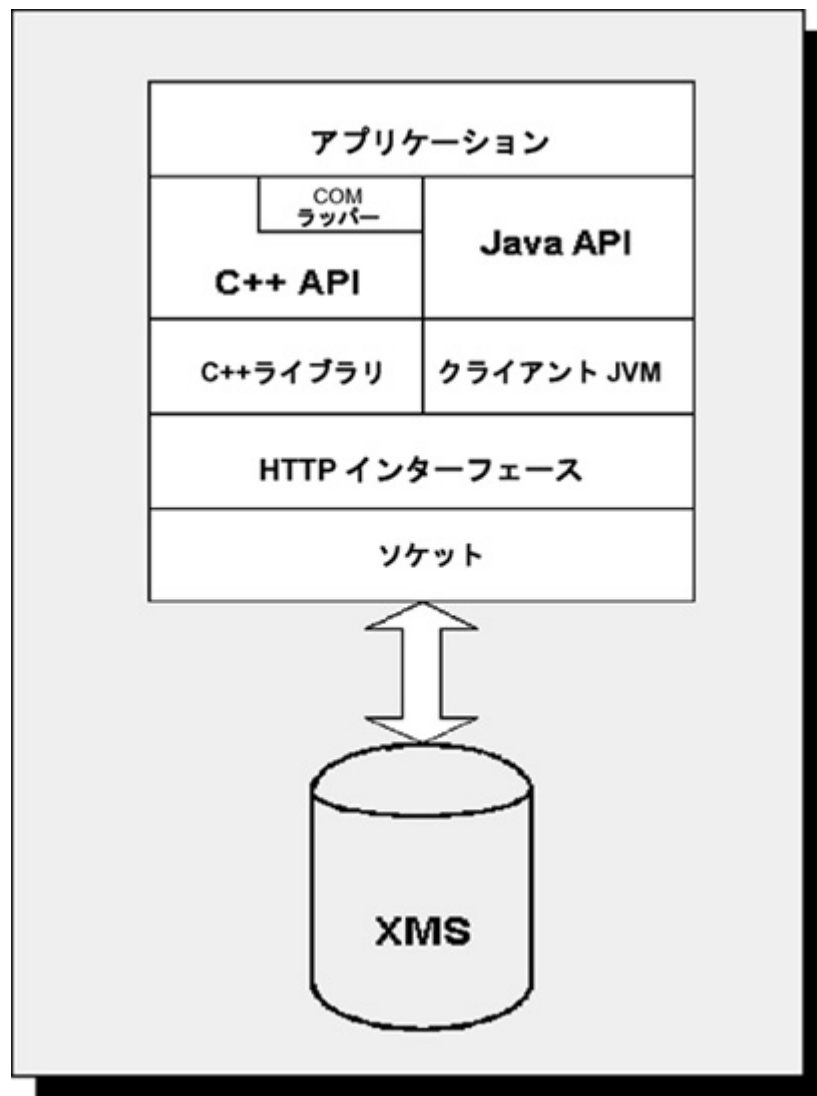
- 書籍、資料、セクションなどの表題は*斜体*で示します。
- コードサンプルや他のリテラル文字列は `courier` フォントで示します。
- ファイル名、コマンド、タブ、メニュー項目、ウィンドウラベルなどは**太字**で示します。

XMS アーキテクチャ概要

NeoCore XMS は、多種多様な情報要件やビジネス要件が絶えず変化していく状況に簡便かつ動的に対応できるハイパフォーマンスの XML データベース管理システムです。このシステムは、Xpriori 社の Digital Pattern Processing (DPP) エンジンを搭載しています。これはデータ内のパターンを認識し、そのデータパターンをフルデータストリングで検索する処理を高速化するための一連のアルゴリズムです (Xpriori 社では 6 件の特許を保持しており、その他に出願中の特許が 20 件あります)。

Xpriori 社は XML の拡張性と DPP コアテクノロジーを組み合わせ、NeoCore XML Information Management System (XMS) を開発しました。XMS はその柔軟なアーキテクチャコアによって、先進的なクエリーや多次元データ分析などの従来のリレーショナルデータベース管理システム (RDMS) の利点をすべて取り込みながら、従来のデータベースアーキテクチャに特有の構造上または処理上の制約や問題点を解消します。

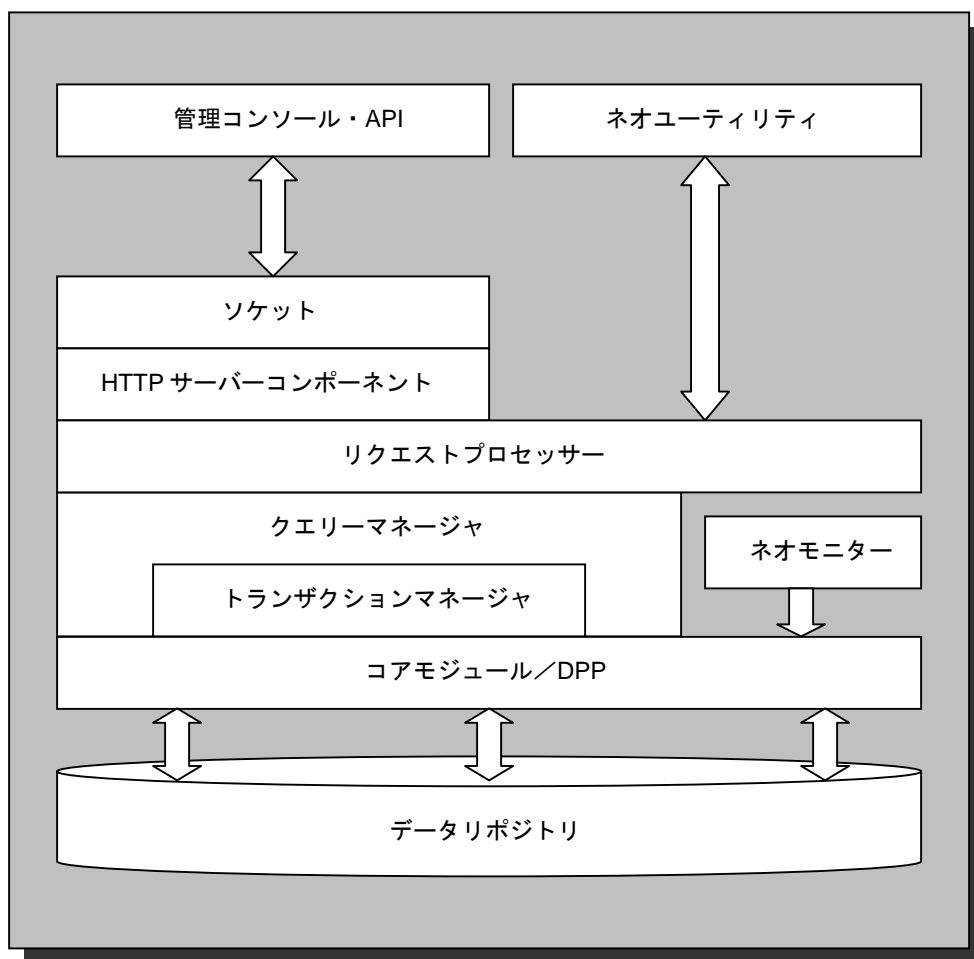
クライアント側の概要



NeoCore XMS のプロトコルレベルインターフェースは TCP 上の HTTP です。アプリケーションは、HTTP を直接生成するか、クライアント側の API によって HTTP を生成するかのいずれかになります。現時点では、Java と C++ の API が用意されています。さらに、C++ API には、Visual Basic などの Microsoft のプログラミング環境をサポートするための COM ラッパーがあります。

サーバー側の概要

以下の図に、XMS サーバーのアーキテクチャの概要を示します。サーバープロセスはサーバーの実行を制御し、サーバーコンポーネントマネージャによって実行対象の正しいコンポーネントをロードします。現在のところ、XMS は HTTP コンポーネントだけをサポートしています。HTTP サーバーコンポーネントは HTTP コマンドを処理し、該当するサブコンポーネントに送信します。



管理モジュールは、統計やサーバー管理などの管理コマンドの処理を担当します。この種のコマンドは、データベースのトランザクション処理に影響を与えません。すべての API からアクセスできますが、管理ポートを経由してアクセスする必要があります。

XMS モジュールは HTTP インターフェースからのコマンドを処理します。こうしたコマンドの大部分は、XMS 内で XML のクエリー、変更、格納を行うトランザクションコマンドです。通常のデータベースポート経由でも管理ポート経由でもアクセスできます。

XMS には、サーバー側の拡張処理を行うために、サーバー側の JVM が用意されています。現時点で、XSLT 変換エンジンはこの JVM で実行されます。他のアプリケーションを使用して、この組み込み環境でサーバー機能を拡張することも可能です。ただし、バージョン 2.7 から、この JVM へのインターフェースはエンドユーザーに公開されなくなっており、Xpriori 社がサーバー側の追加機能を展開するときに限って使用することになっています。将来のリリースでこのインターフェースが公開される可能性はあります。

クライアント要求を処理する XMS の層を XMS コマンドといいます。これには、通常のデータベースコマンドと管理コマンドが含まれていますが、サーバー側の拡張処理は含まれていません。その拡張処理は JNI 層が取り次ぎます。

サーバーコアは、クライアント要求を実行し、内部データ構造を操作し、アクセス制御を行い、トランザクションの整合性を維持します。

インストール



この章では、XMS のインストール関連の手順を説明します。

- インストール
- アンインストール
- Solaris での XMS の複数インスタンスのインストールおよび構成
- XMS の開始
- XMS の停止
- NeoServer の状態の確認
- NeoCore XMS の再インストール
- リモートクライアントのインストール
- NeoServer ホストに対する追加 NIC の増設
- 安全なパスワードリセット
- NeoCore XMS へのアクセス

インストール要件

NeoCore XMS は以下の環境で稼働します。

- Windows 2000 Advanced Server
- Windows 2000 Server
- Windows 2000 Professional
- Windows XP
- Windows Server 2003
- Sun Solaris 2.8 および 2.9 (2001 年 5 月の推奨パッチクラスタ適用済み)

ハードウェア要件

プラットフォーム	Windows 2000: Advanced Server、Server、 <i>Professional</i> Windows XP Windows Server 2003	Sun Solaris 2.8/2.9
プロセッサ	Pentium II 以上	Ultra 5 以上
RAM	256 MB 以上（512MB 以上推奨）	
ディスク領域	NeoCore XMS Server 用に 400 MB 以上	
配布媒体	CD-ROM	

注: メモリー要件は、予想データベースサイズに応じて増減する可能性があります。

ソフトウェア

NeoCore XMS は Java™ 2 SDK Standard Edition バージョン 1.4.1 との互換性があります。HTTP 経由で NeoServer にアクセスするための Java と C++ の API が用意されており、それぞれの資料も添付されています。

API の資料を読むには、以下のようにします。

1. NeoServer 管理コンソールを開始します。
2. [Documentation] タブを選択します。
3. C++ または Java の資料のリンクを選択します。

JRE については、Java™ 2 Runtime Environment Standard Edition バージョン 1.4.1 に関する Sun Microsystems の資料を参照してください。この資料は <http://java.sun.com/docs/index.html> からオンラインで入手できます。

NeoCore XMS のインストール/アクセス/アンインストールの概要

NeoCore XMS のインストールとアンインストールは、管理者権限を持つユーザー（Windows では Administrator、Solaris では root）が実行する必要があります。

Windows プラットフォームの場合、NeoServer はサービスとしてインストールされます。

インストールが完了した時点で、NeoCore XMS にアクセスできます。この章の「**NeoCore XMS へのアクセス**」を参照してください。

NeoCore XMS をアンインストールした場合、削除されるのは NeoCore XMS だけです。通常の操作中に作成したデータベース、ログ、トランザクションファイルは手動で削除する必要があります。

注：再インストールする場合は、まず既存の NeoCore XMS をアンインストールしてください。

注：Solaris 上でインストールやアンインストールを行うには、root アクセス権を持っている必要があります。

Solaris でのインストールの準備 - NeoCore XMS をインストールする前に

NeoCore XMS をインストールする前に、NeoCore XMS 用のグループ、ユーザー、およびアカウントを作成する必要があります。以下の指示に従って、インストールの準備をします。

neoadmin ユーザーの作成

1. root ユーザーとしてログインします。
2. `groupadd neocore` というコマンドで **neocore** というグループを作成します。
3. アカウント **neoadmin** を作成します。neoadmin のホームディレクトリは、NeoCore XMS のインストール先に設定する必要があります。

```
useradd -d <install directory> -g neocore neoadmin <passwd> neoadmin
```

ユーザー **neoadmin** は、デフォルトとして Korn シェルを使用します。以下が Korn シェルパスです。

```
/usr/bin/ksh.
```

X-Window 用の表示設定

NeoCore XMS インストーラは X-Window を使用してインストールウィザードを表示します。コンソールの位置で X-Window を開くための表示設定が必要です。Korn シェルで、次のコマンドを使用します。


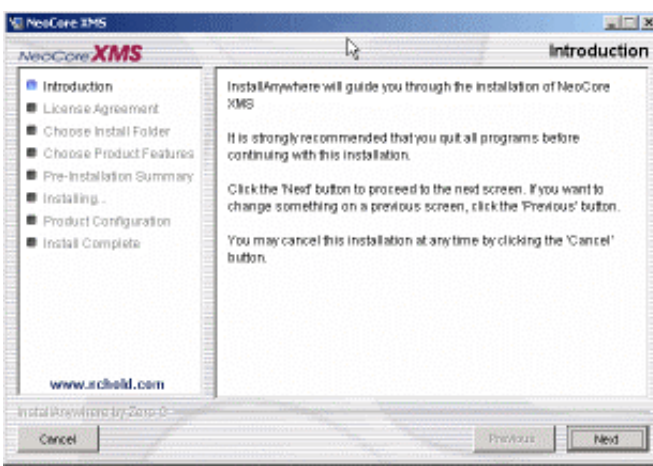
```
export DISPLAY=<ipaddress:0>
```

「ipaddress」はコンソールステーションの IP アドレスです。

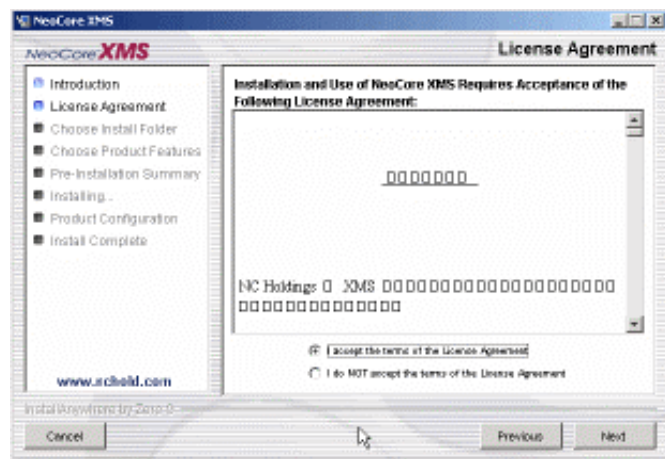
Windows でのインストールの準備 - NeoCore XMS をインストールする前に

NeoCore XMS をインストールする前に、インストールのためのユーザーがインストール先のマシンの管理者グループに属していることを確かめてください。また、NeoCore XMS インストーラが、ローカルコンピュータの管理者グループに属していることも確認してください。

NeoCore XMS のインストール

<p>NeoCore XMS のインストール (Windows および Solaris 2.8/2.9)</p> <p>WINDOWS: CD ドライブに XMS CD を挿入します。</p> <p>SOLARIS: CD ドライブに XMS CD を挿入します。sh(space) NeoCoreXMS<バージョン>.bin を実行します</p>	 <p>The image shows the NeoCore XMS Version 2.7 installation window. It features the NeoCore XMS logo at the top left, a 3D rendering of the CD case in the center, and four buttons on the right: 'Install NeoCore XMS', 'View Documentation', 'Browse CD Contents', and 'View Release Notes'. The text 'XML INFORMATION MANAGEMENT SYSTEM' is at the bottom.</p>
<p>各画面の左ペインには各インストールステップが一覧表示され、作業中の現在のステップが強調表示されます。右側のペインには、情報と指示が表示されます。</p>	 <p>The image shows the 'Introduction' screen of the NeoCore XMS installer. On the left is a list of steps: Introduction, License Agreement, Choose Install Folder, Choose Product Features, Pre-Installation Summary, Installing..., Product Configuration, and Install Complete. The 'Installing...' step is currently selected. The right pane contains text explaining that InstallAnywhere will guide the user through the installation, recommending that all programs be quit before continuing. It also provides instructions on how to proceed to the next screen, return to a previous screen, or cancel the installation. At the bottom, there are 'Cancel', 'Previous', and 'Next' buttons.</p> <p>注: NeoCore XMS の標準インストールではデフォルトのままにします。</p>
<p>[Introduction] を読んでから、[Next] を選択します。</p>	

[License Agreement] を読んでから、
[Next] を選択します。



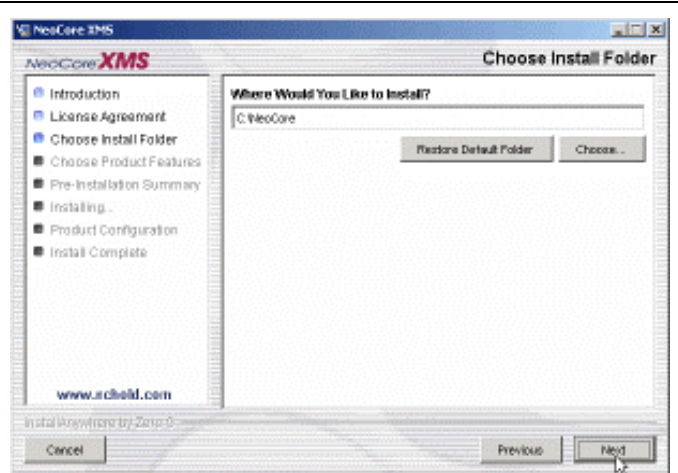
注: Solaris のインストールパスは、以下のインストール画面ショットに示される Windows パスとは少し違います。

注: Solaris にインストールする場合は、`export DISPLAY=<ip-address>:0` も実行する必要があります。`<ip-address>` は X-Window 表示をエクスポートする端末の IP アドレスです。UNIX は大文字小文字を区別するので、`SET` と `set` は同じではありません。また、X-Window は、NeoCore XMS のインストール先のホストではなく、クライアントボックスで実行します。

注: Windows のみ – インストーラの実行中に、このコンピュータには NeoCore が既にインストールされているというメッセージが表示される場合は、インストールの前に NeoCore XMS をアンインストールする必要があります。「NeoCore XMS のアンインストール」を参照してください。

インストールディレクトリを選択するか、デフォルトのままにしてから、[Next] を選択します。

XMS はどこでも望む位置にインストールできます。デフォルトは C:\NeoCore です。



注: Solaris 2.8/2.9 の場合、インストールのためのデフォルトディレクトリは/opt/NeoCore です。

注: Solaris 上に NeoCore XMS の複数インスタンスをインストールできます。その場合は、各インスタンスを別々のディレクトリにインストールする必要があります。なお、ディレクトリ名にスペースを入れることはできません。

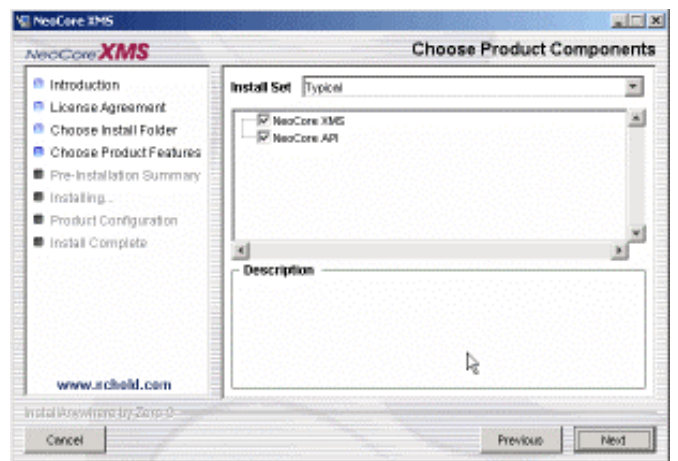
標準インストールには以下のコンポーネントが含まれます。

NeoCore XMS (NeoServer と NeoServer 管理コンソールが含まれます)。

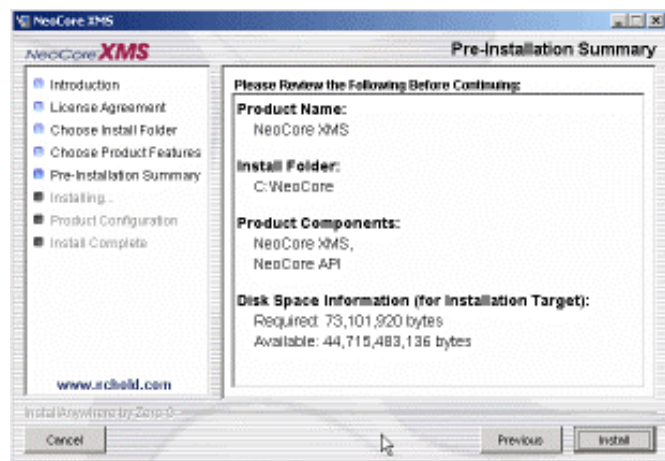
NeoCore API (アプリケーションが NeoServer にアクセスするためのアプリケーションプログラミングインターフェースです)。

カスタムインストールを選択することもできます。カスタムオプションを選択した場合は、どのコンポーネントをインストールするかを自分で選択できます。NeoCore API は通常は開発環境で使用するものなので、不要な場合もあります。

インストールするコンポーネントを選択した後、[Next] をクリックします。

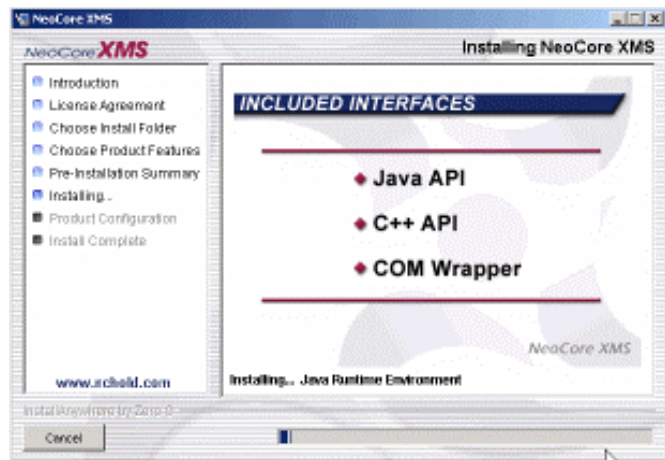


設定を確認してから、[Install] を選択します（前の画面に戻って設定を変更するには、[Previous] を使用します）。



注: Windows 用のデフォルトディレクトリは C:\NeoCore です。Solaris 用のデフォルトディレクトリは /opt/NeoCore です。

[Installing NeoCore XMS] 画面に、インストールの進捗状況が表示されます。完了までに数分かかります。



ドロップダウンリストからデフォルトロケールを選択します。

en – 英語

ja – 日本語

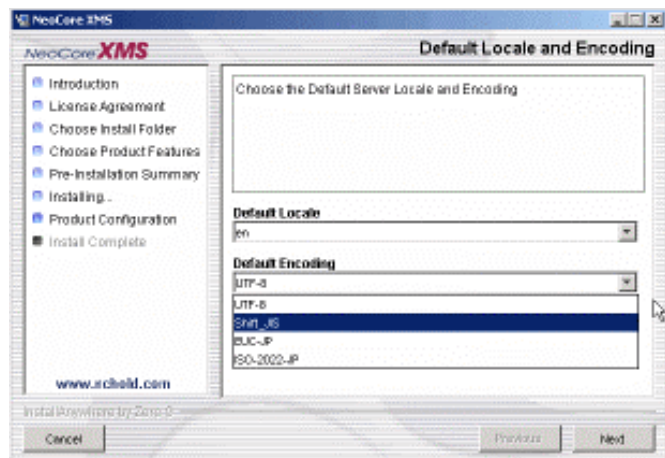
ドロップダウンリストからデフォルトエンコーディングを選択します。

UTF-8

Shift_JIS

EUC-JP

ISO-2022-JP

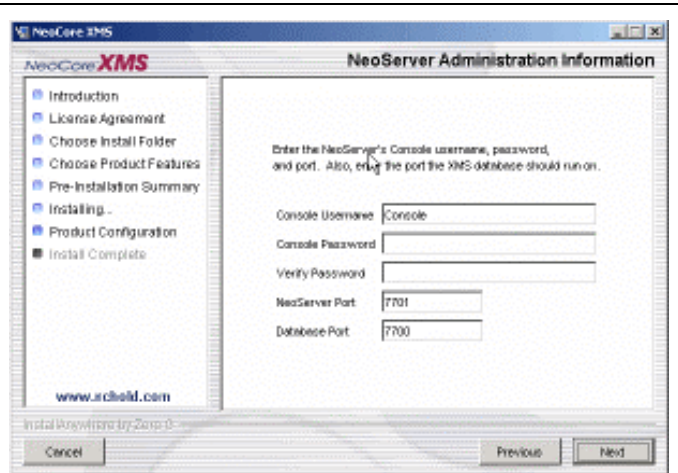


このウィンドウで、パスワード、NeoServer ポート、データベースポートを指定できます。

【**Console Username**】はデフォルトで Console になっています。デフォルトユーザー名を使用することも、別の名前を選択することもできます。希望するコンソールパスワードを入力してから、確認のために再度入力します。

【**NeoServer Port**】は NeoServer を管理するための管理コンソールが使用するポートで、デフォルトは 7701 です。【**Database port**】はデータベース内の情報にアクセスするために使用するポートで、デフォルトは 7700 です。

【**Next**】をクリックします。



注: すべてのユーザー名とパスワードには大文字小文字の区別があります。

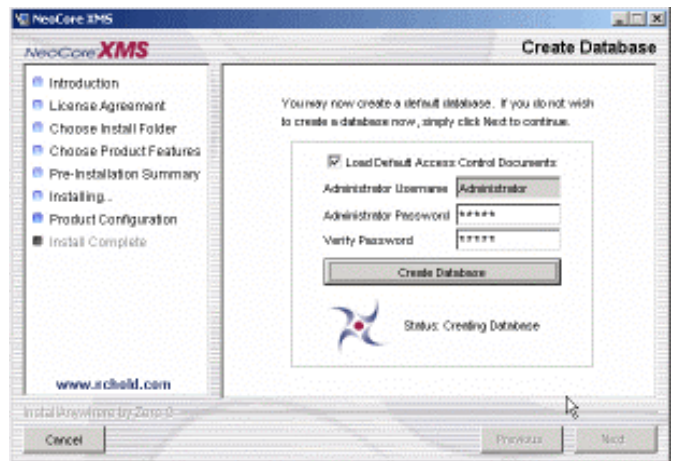
注: この画面で入力したユーザー名とパスワードは管理コンソールへのアクセス専用です。これは、データベース内の情報にアクセスするためにユーザーが入力するユーザー名とパスワードではありません。

次に、デフォルトデータベースを作成します。【**Load Default Access Control Documents**】を選択して、アクセス制御をオンにします。

データベース管理者のユーザー名は **Administrator** です。このユーザー名は変更できません。

パスワードを選択して、入力します。確認のために再度パスワードを入力してから、【**Create Database**】をクリックします。

すぐにデータベースを作成したくない場合は、【**Next**】をクリックします。



注: この時点でデータベースを作成したくないという場合があります。例えば、実際のデータベースに必要なサイズが、XMS に含まれているデフォルトデータベースのサイズとは合わない場合などです。このような場合は、データベースの作成前に **NeoDatabase.xml** ファイルをまず変更してください。

注: 最後のインストール画面では、デフォルトで NeoServer を開始します。データベースを作成していないければ、NeoServer は開始できません。

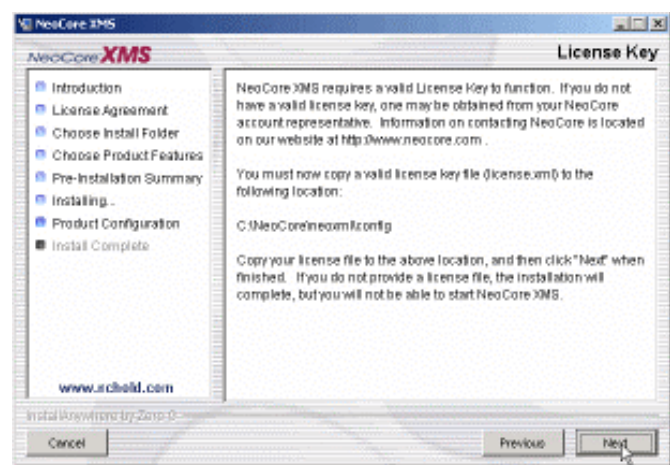
データベースの作成状況に関するメッセージが表示されます。インストールとライセンス設定が完了した時点で、データベース内の情報に完全にアクセスできます。

インストール完了後に XMS を実行するには、以下の場所に有効なライセンスキーファイル (license.xml) をコピーする必要があります。

{NeoCore インストールディレクトリ}
¥neoxml¥config

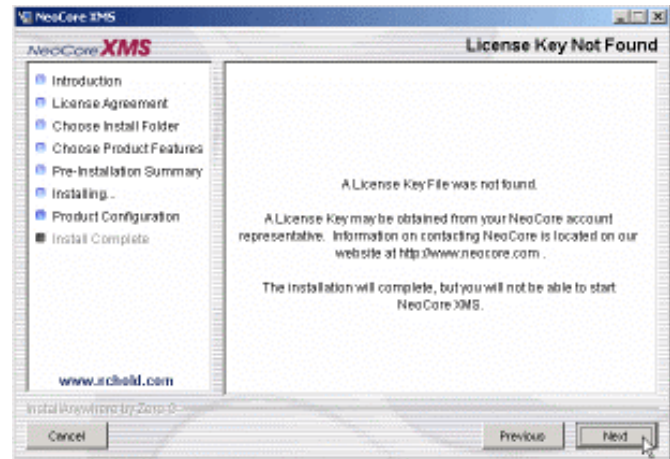
[Next] をクリックします。

有効なライセンスキーファイルがない状態で [Next] をクリックすると、以下のダイアログが表示されます。



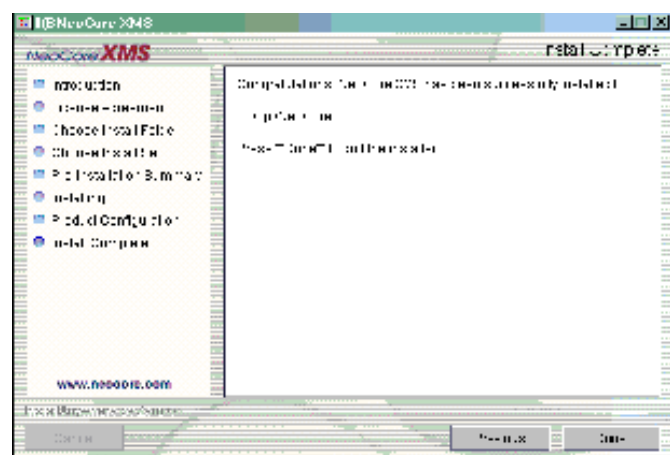
データベースを開始するために、NeoCore サポートセンターから有効なライセンスキーファイル入手する必要があります。Mitsui NeoCore Center にご連絡ください。

[Next] をクリックします。



Solaris 上でインストールしている場合は、この画面が表示されます。

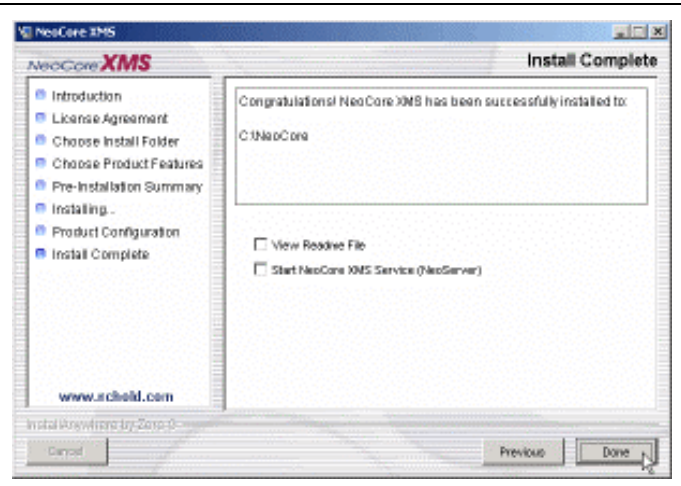
インストーラを終了するには、[Done] をクリックします。



Windows上でインストールしている場合は、この画面が表示されます。

該当するボックスをチェックすることにより、Readme ファイルの表示か、NeoServer の開始か、その両方を選択できます。

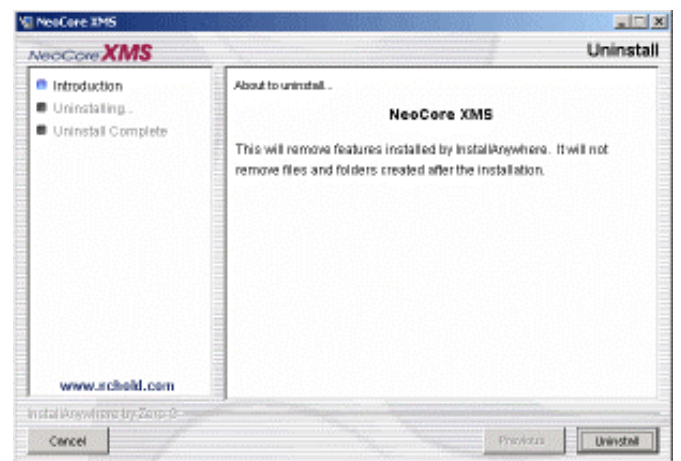
インストーラを終了するには、[Done] をクリックします。



NeoCore XMS のアンインストール

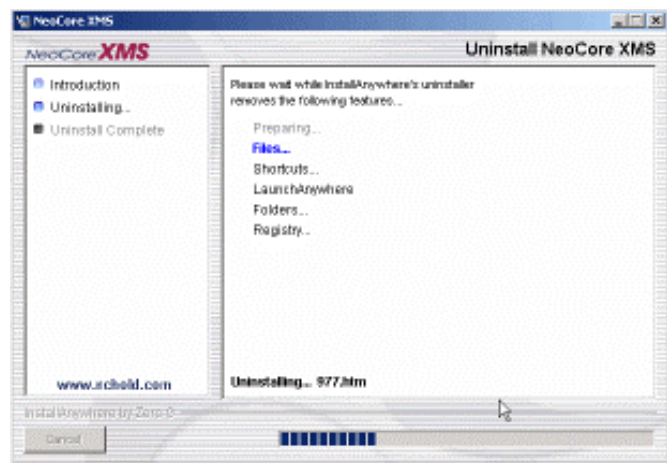
Windows 上の NeoCore XMS のアンインストール

[コントロールパネル] > [プログラムの追加と削除] を選択します。[NeoCore XMS] を選択します。[変更と削除] を選択します。[NeoServer Uninstall Introduction] 画面が表示されます。



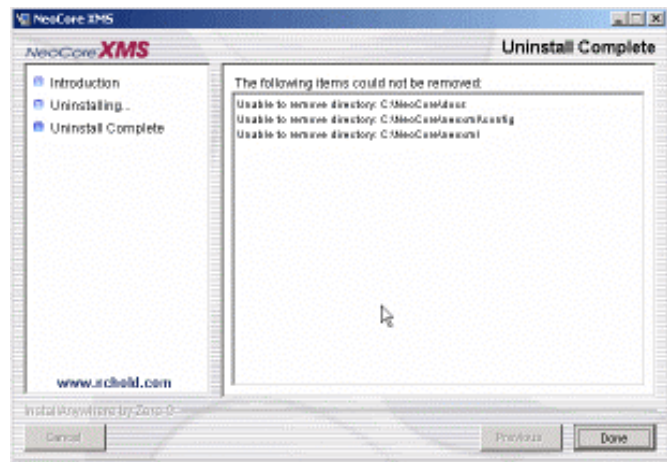
[Uninstall] を選択します。[NeoServer Uninstalling] 画面が表示されます。

この画面に、アンインストールの進捗状況が表示されます。



アンインストールが完了すると、[Uninstall Complete] 画面が表示されます。

アンインストーラを終了するには、[Done] をクリックします。



注: アンインストール時には、[サービス] パネルウィンドウを閉じてください。アンインストール時に [サービス] パネルが開いた状態になっていると、リブートするまでレジストリ情報の更新ができません。

Solaris 上の NeoCore XMS のアンインストール

Solaris から NeoCore XMS をアンインストールするには、root で以下のコマンドを入力します。

```
/opt/NeoCore/UninstallerData/Uninstall_NeoCoreXMS
```

この例は、NeoCore XMS がデフォルトディレクトリ /opt/NeoCore にインストールされている場合です。

NeoCore XMS を別のディレクトリにインストールした場合のコマンドは、

<インストール先のディレクトリ>/UninstallerData/Uninstall_NeoCoreXMS になります。

Solaris での XMS の複数インスタンスのインストールおよび構成

このセクションでは、NeoCoreXMS の複数インスタンスを 1 つの Solaris システムで実行するために必要なインストールとセットアップについて詳しく説明します。また、複数インスタンス間の競合を軽減するために必要な変更と構成設定についても詳しく取り上げます。

複数インスタンスのインストールは、複数データベースにアクセスするために使用します。

注: システム管理者は以下の手順を実行する必要があります。

1. NeoCore XMS の最初のインスタンスをインストールします。

- XMS の旧バージョンがインストールされていない状態でインストールを実行する場合は、「**インストール**」の指示を使用します。
- インストール時に、インストールディレクトリ、NeoServer ポート、データベースポートを記録しておいてください。デフォルトインストールディレクトリ以外のディレクトリにインストールすることを推奨します。

2. このインストールによって、スクリプト `/etc/init.d/neocore` とリンク `/etc/rc3.d/S99neocore` が作成されます。これらのリンクは `/etc/init.d/neocore` を指しており、これにより NeoCore XMS インスタンスの自動的な開始と停止が可能になります。スクリプト `/etc/init.d/neocore` はこのインスタンス専用で、最後にインストールした NeoCoreXMS を指します（インストール先ディレクトリを使用）。

システムの起動時や停止時にシステムの整合性を保つために、このスクリプトと関連リンクの名前を変更してください。

3. 2 番目の NeoCore XMS インスタンスをインストールします。

- 旧バージョンの XMS の 2 番目のインスタンスがインストールされていない状態でインストールを実行する場合は、「**インストール**」の指示を使用します。
- NeoServer ポートとデータベースポートの値を一意的値に変更します。「`netstat -a`」コマンドを実行すると、現在動作中のポートが一覧表示されます。

4. このインスタンスが最後のインスタンスでない場合は、スクリプト `/etc/init.d/neocore` とリンク `/etc/rc3.d/S99neocore` の名前を変更します。

5. さらに NeoCore XMS のインスタンスをインストールする場合は、手順 3 と 4 を繰り返します。

注: NeoCore XMS のインスタンスを別個に制御する必要がない場合は、すべての NeoCore XMS インスタンスの起動と終了を同時に実行できるように、スクリプト `/etc/init.d/neocore` を編集してください。

注: 1 つの NeoCore XMS インスタンス専用の管理ポートに接続した場合は、その NeoCore XMS インスタンス固有の管理機能しか実行できません。データベースポートについても同じことが言えます。複数の NeoCore XMS インスタンスを制御するグローバル制御やデータベース機能は存在しません

（ただし、複数の NeoCore XMS インスタンスを制御するようにスクリプト/etc/init.d/neocore を変更すれば、複数のインスタンスの停止や起動を同時に実行できます）。

注: NeoCore XMS の各インスタンスの起動と停止のためのコマンドは、/etc/init.d/neocore の名前の変更内容に依存します。例えば、インスタンス 1 のスクリプト名を/etc/init.d/neocore1 に変更した場合、起動コマンドと停止コマンドはそれぞれ、「/etc/init.d/neocore1 start」と「/etc/init.d/neocore1 stop」になります。

XMS の開始

Windows

注: NeoServer 管理コンソールや NeoServer にアクセスする前に、サーバーを開始しておく必要があります。

推奨する方法: コマンドラインから、C:¥NeoCore¥neoxml¥bin ディレクトリで次のコマンドを入力します。

```
NeoServerCmd start
```

別の開始方法:

〔スタート〕 > 〔設定〕 > 〔コントロールパネル〕 > 〔管理ツール〕 > 〔サービス〕 を選択します。

NeoServer を選択します。

右クリックして 〔開始〕 を選択します。

注意: コマンドラインコマンドの **net start** や **net stop** は使用しないでください。ロールバックとインデックス再構築が必要になる場合があります。

注意: C:¥NeoCore¥neoxml¥bin¥ディレクトリで「neoserver」と入力して **NeoServer** を開始することはしないでください。

Solaris 2.8/2.9

Solaris 2.8/2.9 で NeoServer を開始するための望ましい方法は、/etc/init.d/neocore スクリプトを使用することです。このファイルは、NeoServer を正しく稼働できるようにインストール時に変更されます。

root または neoadmin としてログインして、以下のように入力します。

```
/etc/init.d/neocore start
```

別の開始方法:

NeoServer [-d] [-r <RootDir>]

例: NeoServer -d -r /usr/local/NeoCore

注: NeoServer 実行可能ファイルに rootdir 引数を渡さないと、デフォルトで /opt/NeoCore が使用されます。このデフォルトディレクトリ（またはその log ディレクトリや config ディレクトリ）が存在しない場合は、NeoServer を開始できません。NeoServer は neoadmin として実行する必要があります。それで、XMS が /usr/local/NeoCore にインストールされている場合は、「NeoServer -d -r /usr/local/NeoCore」を実行してください。-d オプションは、デーモンとして（端末セッションと結び付けずに）このコマンドを実行するためのオプションです。

XMS の停止

Windows

推奨する方法: コマンドラインから、C:\%NeoCore%\neoxml%\bin ディレクトリで次のコマンドを入力します。

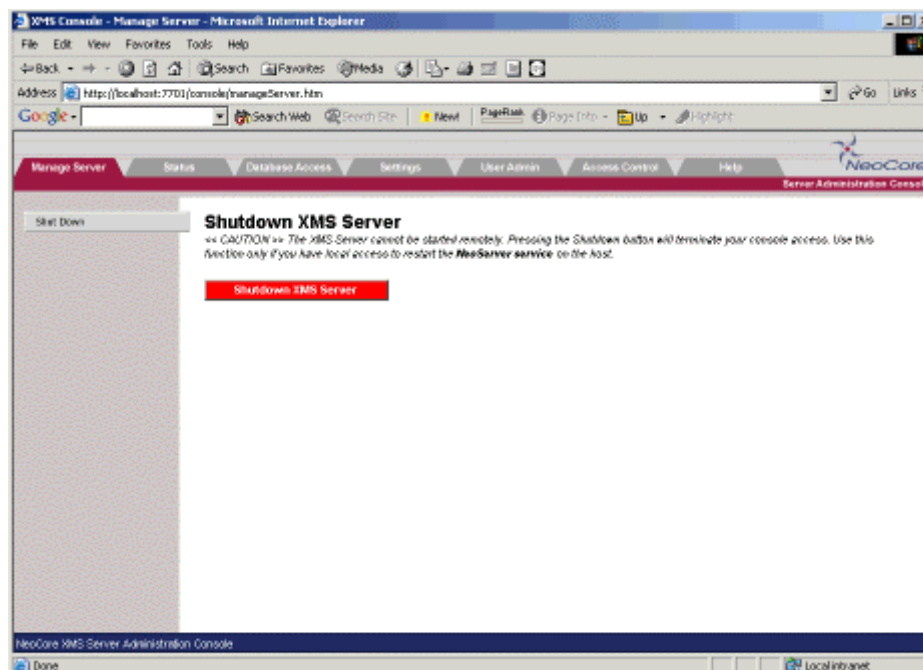
NeoServerCmd stop

あるいは、以下のようにします。

[**Server Administration Console**] で [**Manage Server**] タブを選択します。

[**Shut Down**] ボタンを選択します。

[**Shutdown XMS Server**] ボタンを選択します。



注意: コマンドラインコマンドの **net start** や **net stop** は使用しないでください。ロールバックとインデックス再構築が必要になる場合があります。

Solaris 2.8/2.9

Solaris 2.8/2.9 で NeoServer を停止するための望ましい方法は、`/etc/init.d/neocore` スクリプトを使用することです。

root としてログインし、以下のように入力します。

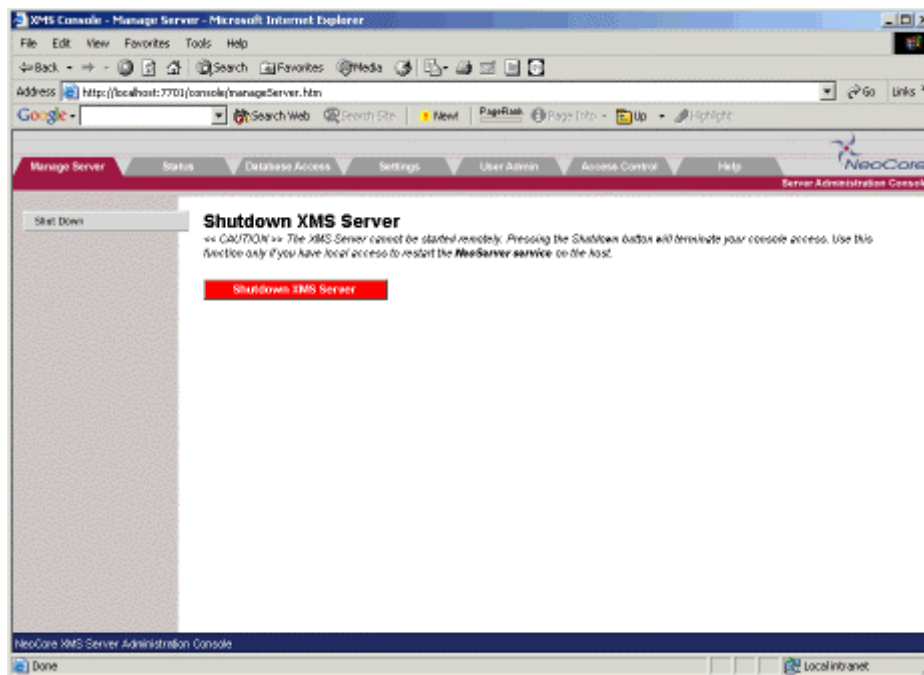
```
/etc/init.d/neocore stop
```

あるいは、以下のようにします。

[**Server Administration Console**] で [**Manage Server**] タブを選択します。

[**Shut Down**] ボタンを選択します。

[**Shutdown XMS Server**] ボタンを選択します。



別の方法:

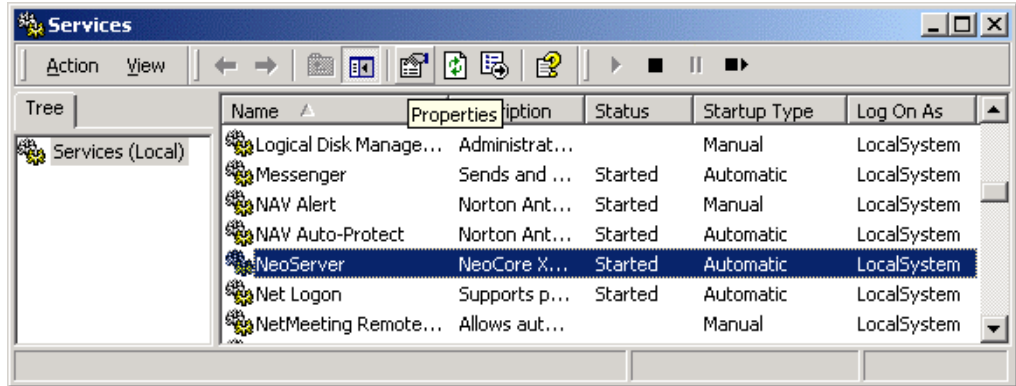
```
kill -HUP [pid or %job-id]
```

注: 「`kill -n [pid or %job-id]`」は使用しないでください。停止時 NeoServer がトランザクションを休止していたり、インデックス再構築を行っていたりする場合があります。

NeoServer の状態の確認

NeoServer の状態を確認するには、[スタート] > [設定] > [コントロールパネル] > [管理ツール] > [サービス] を選択します。

一覧に NeoServer が表示されているはずです。ファイルはアルファベット順に表示されています。[状態] 欄に [開始] と表示されているか、停止しているか、空白になっています。



NeoCore XMS の再インストール

Windows

1. NeoCore XMS の Windows 用アンインストール手順を実行します。
2. NeoCore XMS の Windows 用インストール手順を実行します。

Solaris 2.8/2.9

1. NeoCore XMS の Solaris 2.8/2.9 用アンインストール手順を実行します。
2. NeoCore XMS の Solaris 2.8/2.9 用インストール手順を実行します。

リモートクライアントのインストール

リモートマシン上で NeoCore XMS コンソールを使用するには、ローカルマシン上の Web ブラウザにアクセスできなければならない、NeoCore XMS ソフトウェアをリモートマシン上にインストールしておく必要があります。

NeoServer ホストに対する追加 NIC の増設

注: 複数の NIC（ネットワークインターフェースカード）を備えたホストで NeoServer を稼働する場合は、NeoServer.xml ファイルを変更する必要があります。ホストが複数の IP アドレスを持つ状況としては、1 つの IP リンクではネットワークトラフィックが低下するとか、クライアントが特定のネットワークリンクに制限されている、などの状況が考えられます。

NeoServer.xml の既存の IP 要素の後に以下を挿入します。

```
<IP NeoID="Title">
  <Port>7702</Port>
  <Address NeoID="NIC2">192.255.255.1</Address>
  <Alias NeoID="Console">/console</Alias>
  <Alias NeoID="XMS Query">/neoquery</Alias>
  <Alias NeoID="XMS Administration">/neoadmin</Alias>
  <Alias NeoID="Documentation">/docs</Alias>
</IP>
```

- **IP NeoID** は接続の名前で、通常は目的や機能を示す名前にします。例えば、データベース用なら「Title」を **XMS** という名前で置き換え、管理コンソール用なら「Title」を **Console** という名前で置き換えるといった具合です。この名前は一意である必要があります。
- **Port** は接続ポートです。アドレスとポートの組み合わせは一意である必要があります。上記の 7702 の代わりに一意のポート番号を入力できます。
- **Address NeoID** は NIC カードの名前です。名前とアドレスの組み合わせは一意である必要があります。例えば、**NIC2** はアドレス **192.255.255.1** を参照し、そのアドレスとペアになっています。上記の NIC2 の代わりに一意の名前を入力できます。
- **Address** は NIC カードの IP アドレスです。上記の IP アドレス 192.255.255.1 の代わりに、その NIC カードの一意のアドレスを入力する必要があります。
- **Alias** は接続の使用法を定義するオプションフィールドです。以下に挙げる 4 つのいずれかまたはすべてを使用できます（どれも使用しない場合、その接続は使用不可になります）。以下の中から選択できます。
 - **Console** – コンソールログインとコンソール機能
 - **XMS Query** – データベーストランザクション（クエリーなど）
 - **XMS Administration** – データベース管理機能（データベースログイン、アクセス制御など）
 - **Documentation** – 資料へのアクセス

コンソール接続には 4 つの Alias すべてを使用することを推奨します。データベースユーザー接続には Console 以外のすべてを使用することを推奨します。

注: 他の IP 要素のデフォルト (<Address NeoID="NIC0">default</Address>) を各 NIC の特定の IP アドレスで置き換えることを推奨します。これを置き換えておかないと、ipconfig の変更 (または他のシステム変更) で NeoServer が特定の IP アドレスに応答しない可能性があります。

例: これは、既存の NeoServer に **Console1** を追加して、**Console** 要素のデフォルトを特定の IP アドレスに変更したものです。NeoServer.xml ファイルの一部だけを示します。

```
</IP>
  <IP NeoID="Console">
    <Port>7701</Port>
    <Address NeoID="NIC0">20.20.20.1</Address>
    <Alias NeoID="Console">/console</Alias>
    <Alias NeoID="XMS Query">/neoquery</Alias>
    <Alias NeoID="XMS Administration">/neoadmin</Alias>
    <Alias NeoID="Documentation">/docs</Alias>
  </IP>
  <IP NeoID="Console1">
    <Port>7702</Port>
    <Address NeoID="NIC1">192.255.255.1</Address>
    <Alias NeoID="Console">/console</Alias>
    <Alias NeoID="XMS Query">/neoquery</Alias>
    <Alias NeoID="XMS Administration">/neoadmin</Alias>
    <Alias NeoID="Documentation">/docs</Alias>
  </IP>
</Network>
```

注: 変更を有効にするには、NeoServer をいったん停止してから開始し直す必要があります。以下の「NeoServer の開始」を参照してください。

注: 各 NIC に一意のポートを割り当てる必要はありません。ホストのデフォルト IP アドレスを指定するときに、「localhost」または「default」という語をすべて大文字かすべて小文字のいずれかで使用することもできます。これらの語はどちらを使用しても同じ意味になります。例えば、次のようにします。

```
<IP NeoID="Console">
  <Port>7701</Port>
  <Address NeoID="NIC1">localhost</Address>
  <Address NeoID="NIC0">20.20.20.1</Address>
  <Address NeoID="NIC1">192.255.255.1</Address>
  <Alias NeoID="Console">/console</Alias>
  <Alias NeoID="XMS Query">/neoquery</Alias>
  <Alias NeoID="XMS Administration">/neoadmin</Alias>
  <Alias NeoID="Documentation">/docs</Alias>
</IP>
```

安全なパスワードリセット

コンソールパスワードと XMS データベース管理パスワードの両方を安全にリセットするために、XMSAdmin というコマンドがあります。

XMSAdmin コマンド

XMSAdmin コマンドは、Solaris と Windows のどちらの場合でもシェルコマンドラインから開始します。実行方法とパラメータはどちらのプラットフォームでも同じです。このコマンドは XMS インストールの存在するローカルマシンで実行し、そのインストールだけに影響を与えます。このコマンドを使用して、別のマシンのパスワードをリセットすることはできません。このコマンドを実行するユーザーは、オペレーティングシステムレベルの認証を必要とします。Solaris の場合、このコマンドを実行するユーザーは「neocore」グループのメンバーである必要があります。

XMSAdmin のパラメータ

現在、XMSAdmin が実行し、パラメータとして渡すサブコマンドには、**XMSPassword** と **ConsolePassword** の 2 つがあります。これらのコマンド実行の構文は、以下のとおりです。

ConsolePassword

XMSAdmin ConsolePassword [root_directory]

このコマンドは、管理コンソールユーザーのパスワードをリセットします。新しいパスワードを有効にするには NeoServer を再起動する必要があります。このコマンドは、**コンソール管理パスワード**だけに適用されます。このコマンドで**コンソール管理者のユーザー名**を変更することも可能です。XMS がデフォルト位置（つまり Windows では「C:\NeoCore」、Solaris では「/opt/NeoCore」）以外の位置にインストールされている場合は、オプションの「root_directory」パラメータを指定する必要があります。このコマンドを実行したときの標準的な表示内容は、以下のとおりです。

```
$ XMSAdmin ConsolePassword
NeoCore XML Management System (XMS) Administration Version: 3.0
Console password reset utility.
```

```
Please, enter new console user name : System
New password : *****
Confirm New password : *****
The console password has been reset.
```

XMSPassword

XMSAdmin XMSPassword [port]

このコマンドは、データベースのユーザーパスワードをリセットします。このコマンドの本来の意図は管理者のパスワードをリセットすることですが、どのユーザーのパスワードでもリセットできます。このコマンドを実行するには、NeoServer が稼働している必要があります。この XMS インストールで 7700 以外のポートを使用している場合は、任意指定の port パラメータを指定する必要があります。このコマンドを実行したときの標準的な表示内容は、以下のとおりです。

```
$ XMSAdmin XMSPassword
NeoCore XML Management System (XMS) Administration Version: 3.1
XMS password reset utility.
```

```
Please, enter user name : Administrator
New password : *****
Confirm New password : *****
<success/>
```

注: このコマンド実行からの応答は、NeoServer からの XML 応答です。その XML は解析されず、stdout にそのまま渡されます。

NeoCore XMS へのアクセス

インストールが完了した時点で、ユーザーはコンソール経由で NeoCore XMS にアクセスできます。アプリケーションから NeoServer にアクセスするには、アプリケーションプログラミングインターフェース (API) を使用する必要があります。API の詳細については、『**NeoCore XMS System Programming Guide**』を参照してください。

NeoCore XMS を開始する前に

- ソフトウェアが正しくインストールされている必要があります
- データベースが作成されている必要があります
- データベースの構成ファイルが/NeoCore/neoxml/config に存在する必要があります
- Windows の場合、データベースは、必要なら Windowing を使用して、仮想メモリーに収まっている必要があります。

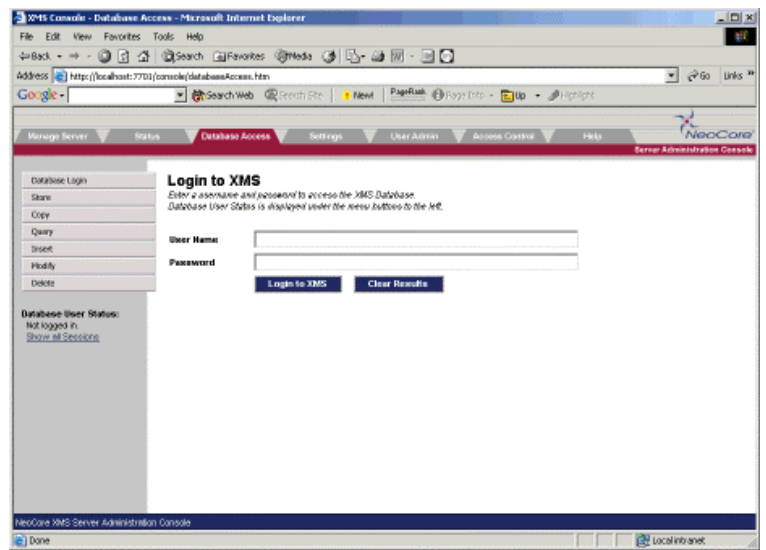
注: NeoServer 管理コンソールや NeoServer にアクセスする前に、サーバーを開始しておく必要があります。

XMS にアクセスするには、デスクトップ上の **[Launch Console]** アイコンをダブルクリックするか、**[スタート] > [プログラム] > [NeoCore XMS] > [Launch Console]** を選択します。

NeoServer 管理コンソールにログインするには、インストール時に定義したデフォルトの「Console」でログインする必要があります。



XMS データベースにログインするには、まず **[Database Access]** タブでログインする必要があります。



エラーメッセージ

以下の表に、クライアント側の例外とそれぞれの例外番号を示します。2 つ目の表では、クライアント側の例外とサーバー側を対応付け、原因と対応策を一覧にしています。

クライアント側の例外	例外番号
NeoInvalidArgumentException	1
NeoAuthorizationFailedException	2
NeoCommunicationException	3
NeoInvalidSessionException	4
NeoMalformedURLException	5
NeoMalformedXMLException	6
NeoMalformedXPathException	7
NeoServerException	8
NeoTransactionDeadlockException	9
NeoTransactionException	10
NeoTransactionRollbackException	11
NeoTransactionSessionException	12
NeoTransactionTimeoutException	13
NeoDestroyedResultChunkException	14
NeoQueryResultExceededMaxException	15
NeoACException	16
NeoACGroupExistsException	17
NeoACRuleExistsException	18
NeoACUserExistsException	19
NeoACUsersExistInGroupException	20
NeoACGroupDoesNotExistException	21
NeoACAssociatedRulesExistException	22
NeoCursorOutOfRangeException	23

クライアント側の例外	例外番号
NeoManualLoadRequiredException	24
NeoClientException	25

例外 番号	例外名	原因	対応策
1	CANNOT_RESOLVE_NS	名前空間が正しく設定されていません。	正しい名前空間を使用します。
1	INVALID_COMMAND	XMS Server に無効なコマンドが送信されました。	コマンドを確認します。
1	MISSING_DATA	コマンドに必要なデータがありません。	必要なデータがすべて含まれたコマンドを再入力します。 modifyXML の変更後データがない等
1	MISSING_PARAM	コマンドにパラメータが不足しています。	必要なパラメータがすべて含まれたコマンドを再入力します。 modifyXML の XPATH 式がない等
1	MODIFY_NODE_NOT_FOUND	変更するノードが存在しません。	ノードへの XPath を確認します。
1	XPATH_EXPR_NO_DOC	文書が存在しません。XPath 式で文書が指定されていません。Copy コマンド実行の際に、文書単位 (MetaData 含む) でデータを取得しないクエリ ("/ND/A/B" など) による指定を行った場合、もしくは該当する文書が存在しない場合です。	XPath が正しいことと、文書が格納されていることを確認します。

1	UNK_ISOL_LEVEL	分離レベルが無効です。	正しい分離レベルを再入力します。有効な分離レベルは、Read_uncommitted、Read_committed、およびRepeatable_read です。
4	INVALID_SID	現在、データベースにログインしていません。 ログインしていないか、セッションがタイムアウトになりました。セッションタイムアウト値は、NeoXDBRuntime.xml の<Sessions><Timeout>で指定します。	前のセッションでNeoServerと切断された可能性があります。 もう一度データベースにログインします。
6	MALFORMED_XML	整形形式のXMLではありません。	XML データを修正します。
7	MALFORMED_XPATH	XPath のシンタックスが正しくありません。	XPath シンタックスを修正します。
8	AC_NOT_ACTIVATED	AC 文書がないか、壊れています。	以前のエクスポートから文書をインポートするか、バックアップから復元します。
8	INTERNAL_ERROR	エラーの詳細については、サーバー ログを確認してください。	詳細をサーバー ログで確認します。
8	CANNOT_ALLOC_BUFFER_SPACE	RAMが不足しており、設定されたバッファを操作できません。クエリ実行時にあらかじめ定義された BufferPool を確保する際に、起動中他のプロセスによるメモリ利用などで、所定の領域が確保できない場合に発生するエラーです。 【補足】NeoServer 起動時には MemoryBufferPool で指定された Buffer をメモリ上に確保するためにメモリの空き領域チェックを行っておりますが、定義されたすべての Buffer を起動直後に展開するわけではありません。	NeoServer.xml (NeoServer.xml の MemoryBufferPool で定義されている Buffer) でバッファの容量を小さくするか、RAM を拡張します。

8	CANNOT_GET_SESSION	NeoServer が動作していません。	<p>(Solaris の場合のみ、 中断状態のプロセスが存在します。 NeoServer のプロセスをすべて強制終了してください)</p> <p>NeoServer を起動します。</p>
8	COULD_NOT_GET_BUFFER	十分なバッファが設定されていません。 NeoServer 内部で Buffer 取得に関するファイル IO エラーが生じた場合に出力されます。	NeoServer.xml で容量の大きいバッファを設定します。
8	UTIL_NEC_COULD_INIT_DB_STARTUP	データベースが壊れている可能性があります。	バックアップからデータベースを復元します。
8	DICT_OR_INX_FULL	<p>Database Dictionary/Index であらかじめ割り当てられた Index は、すべて使用されています。 データのストア時に、Database Dictionary/Index ※ の Status が Full になった状態で発生します。</p> <p>※ : NeoDatabase.xml の以下のサイズです。</p> <ul style="list-style-type: none"> ・ <DataDictionary><Size> ・ <DataIndex><Core><Size> ・ <TagDictionary><Size> ・ <TagIndex><Core><Size> ・ <TagPlusDataIndex><Core><Size> 	<p>ステータスを確認し、一杯になっているファイルを特定します。</p> <p>NeoDatabase.xml でファイル サイズを拡張し、データベースを再作成します。</p>
8	DB_FULL	<p>ステータスを確認し、一杯になっているファイルを特定します。 NeoXMLUtils import コマンド実行時に、データベース領域が足りなくなった場合に出力されます。</p>	<p>NeoDatabase.xml でファイル サイズを拡張し、データベースを再構築します。</p> <p>詳細をサーバー ログで確認します。</p>

8	UTIL_NEC_DB_MAP_FULL	データベースに使用する Database Map File のフットプリントが不足しています。	ステータスを確認し、一杯になっているファイルを特定します。 NeoDatabase.xml でファイル サイズを拡張し、データベースを再構築します。
1	UTIL_NEC_CANNOT_INSERT_AT_ND	root レベルに挿入しようとしています。	/ND より下に挿入されるよう、insert 文の XPath を修正します。
8	FULL_INX_MUST_BE_ENABLED	フルインデックスの機能が有効化されていません。	NeoDatabase.xml で Index Mode を変更し、データベースにインデックスを再構築します。
8	INX_FAILURE	NeoCore XMS Server のインデックスで内部エラーが発生しました。詳細はサーバー ログを参照してください。	インデックスを再構築します（reindex が機能しない場合は、バックアップからデータベースを復元します）。
8	INVALID_ADMIN_COMMAND	無効なコマンドです。	コマンドを修正します。
8	INVALID_MAP_FILE	NeoCore XMS Server の内部エラーが発生しました。 存在しないマップ ファイルへのアクセスがありました。	詳細をサーバー ログで確認します。
8	INVALID_QUERY	無効なクエリーです。	クエリーを修正します。
8	LOCK_EXCP	トランザクションの処理中に XMS Server の内部エラーが発生しました。	例外の種類を調べます。 詳細をサーバー ログで確認します。
1	NO_ACTION_TO_TAKE	コマンドが入力されていません。	コマンドを入力します。
8	OP_FAILED	内部操作で、XMS Server の内部エラーが発生しました。	詳細をサーバー ログで確認します。

8	RES_SET_TOO_LARGE	コマンドの戻り値が XmlBufferSize を超えています。	NeoServer.xml で XmlBufferSize を拡張し（サイズの拡張または適切なサイズの BufferPool の作成が必要な場合があります）、NeoServer を再起動します。
8	STORE_FAILED	保存が完了していません。	STORE で領域不足が発生した時に発生。1 件も文書が格納されなかった時等。サーバー ログで詳細を確認し、修正します。
9	TXN_DEADLOCK	同一のデータに複数のトランザクションが試行されました。	トランザクションを終了し、再試行します。
10	TXN_NOT_ACTIVE	プロセスにトランザクションがない状態で、トランザクションのコミットまたはロールバックが試行されました。	新しいトランザクションを開始し、コマンドを再実行します。
10	TXN_NOT_STARTED	プロセスにトランザクションがない状態で、トランザクションのコミットまたはロールバックが試行されました。	新しいトランザクションを開始し、コマンドを再実行します。
10	TXNS_NOT_ENABLED	NeoCore XMS Server でトランザクションが有効化されていません。	インストールを確認します。
11	TXN_ROLLBACK	Transaction_rollback コマンドまたはタイムアウトにより、トランザクションがロールバックされました。	トランザクションを開始し、コマンドを再実行するか、ロールバック処理を行う場合のロジックをアプリケーションに追加します。
13	TXN_TIMEOUT	トランザクションがタイムアウトになりました。	NeoXDBRuntime.xml で MaxDuration を増やします。
8	XMS_APP_ERROR	XMS Server の内部エラーが発生しました。	詳細をサーバー ログで確認します。

1	QUERY_MATCHED_MULT_DOCS	一度にコピーできる文書は 1 つだけです。	文書を 1 つだけ指定するよう、コピーの設定を変更します。 (/ND[MetaData/DocID=NN]ではなく、 /ND[MetaData/DocID=NN and MetaData/CopyNumber=X] とします)
8	DB_NOT_INITIALIZED	データベースを開けません。	サーバー ログで詳細を確認し、必要に応じて修正します。
2	AUTH_FAILED	データベースへのログインに失敗しました。	無効なユーザー名、パスワード、またはアカウントが入力された可能性があります。
1	INVALID_LOCK_TYPE	無効なロック タイプが指定されました。	次の有効なロック タイプを指定します。 Shared update exclusive
12	SESSION_NOT_ACTIVE	セッションの SID は無効です。	一旦ログアウトし、データベースにログインし直します。
1	NODE_MISMATCH	変更対象の XPath に、変更すべき XML がありません。	変更する XPath または XML を修正します。
11	NODE_MISMATCH_TXN_RB	変更処理でノードの不一致があったため、トランザクションがロールバックされました。	変更する XPath または XML を修正します。
8	AC_PROC_EXCP	アクセス管理ルール処理中に、NeoCore XMS Server の内部エラーが発生しました。	詳細をサーバー ログで確認します。
1	NEC_ISO_LVL_NOT_SUPPORTED	サポートされていないシリアル化可能な分離レベルが使用されました。	有効なレベルは、 Read_uncommitted、 Read_committed、および Repeatable_read です。

14	RES_CHUNK_NOT_AVAILABLE	チャンクはすでに転送または破棄されています。	破棄される前にチャンクをフェッチするか、破棄を 2 回行わないようにしてください。 カーソルを使用した時のみ。 カーソルを使用し Destory 済の結果にアクセスした場合。
8	CANNOT_ALLOC_NEORESULT	neoreresult を使用できません。	NeoServer がビジー状態でないときに再試行します。 カーソルを使用した時のみ。 カーソルを使用し、複数クライアントからのアクセスでリソースが不足。
8	SES_REACHED_CAPACITY	設定されている NeoXDBRuntime.xml の最大値を超えました。	config パラメータの値を増やすか、アプリケーションを再設定し、未使用の NeoResult をできる限り早急に解放します。
15	RES_EXCEEDS_MAX	結果セットの要素が多すぎます。	config パラメータの値を増やします。
8	RES_REQ_MORE_CHUNKS	設定されている最大値を超えました。	config パラメータの値を増やします。 カーソルを使用した場合のみ。 <ResultChunking>の<MaxNeoChunksPerNeoResult>の値が不足。
1	NRP_CONTENTS_INVALID	NeoResult Profile に無効なデータが含まれています。	NeoResult Profile コンストラクタの引数を確認します。
1	INVALID_TRACELEVEL	無効なトレース レベルが指定されました。	有効なトレース レベルを指定します。
8	LIC_INVALID	入力された XMS ライセンスは無効です。	有効なライセンスを取得します。

8	LIC_UNINITIALIZED	XMS ライセンスが初期化されていません。	詳細をサーバー ログで確認します。
---	-------------------	-----------------------	-------------------


NeoServer 管理コンソール

概要

NeoServer 管理コンソールからは、NeoServer の管理機能にアクセスできます。また、データベースとその情報にも限定的にアクセスできます。このコンソールはインターネットブラウザ上で稼働します。推奨するのは Internet Explorer 5.0 以降か Netscape 7.0 以降です。このコンソールを使用するには、ブラウザの Cookie と JavaScript がブラウザで有効になっていて、ネットワークでも許可されている必要があります。

コンソールの起動

コンソールを起動するには、以下のようにします。

<p>[スタート] > [プログラム] > [NeoCore XMS] で [Launch Console] を選択します。もしくは、デスクトップで [Launch Console] アイコンをダブルクリックします。</p> <p>(この方法は、NeoServer がインストールされているマシンの管理コンソールにリモートでアクセスするときに使用します)</p>	
<p>この方法は、NeoServer がインストールされているマシンで管理コンソールにアクセスするときに使用します。</p> <p>Web ブラウザを開始し、以下のアドレスを選択します。</p> <p><code>http://<servername>:port/console</code></p>	

例:

```
http://localhost:7701/console
http://172.21.3.5:7701/console
```

コンソールを開始すると、ログイン画面が表示されます。インストール時に選択した管理コンソールのユーザー名とパスワードを使用します。

注: NeoCore XMS は 2 つのポートを使用します。ポート 7700 はデータベース用のデフォルトで、ポート 7701 は管理コンソール用のデフォルトです。これらのポートはインストール時に設定しますが、設定ファイル **NeoServer.xml** で変更することも可能です。

コンソールパスワードの変更

管理コンソールのユーザー名とパスワードは、Windows の場合、 {NeoCore インストールディレクトリ}¥neoxml¥bin ディレクトリ、Solaris の場合、 {NeoCore インストールディレクトリ}/neoxml/bin ディレクトリで変更できます。

Windows:

```
NeoServer -RootDir=<root-dir> -AuthString=<username>:<password>
```

例:

```
NeoServer -RootDir=C:¥NeoCore -AuthString=console:web
```

Solaris:

```
GenAdminPwd <username>:<password> <root-dir>
```

例:

```
GenAdminPwd console:web /opt/NeoCore
```

変更を有効にするには、NeoServer を再起動する必要があります。

XMS コンソール GUI

NeoServer 管理コンソールにアクセスすると、バージョン情報とアクセスに関する画面が表示されます。以下のタブがあります。

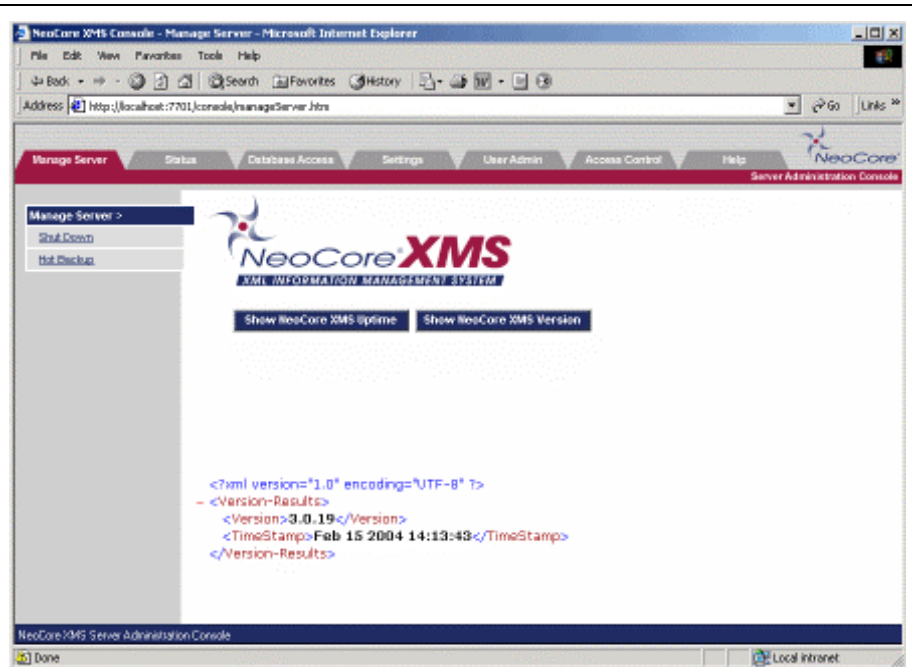
- Manage Server (NeoServer の停止、バージョンと実行時間の表示)
- Status (NeoServer とデータベースの状態)
- Database Access (データベースへのアクセス)
- Settings (NeoServer の設定の表示)
- User Admin (ユーザーとグループの追加、変更、削除)
- Access Control (ユーザーアカウント、グループアカウント、管理ルールの管理)
- Help (サポート窓口、資料、オンラインヘルプへのアクセス)

Manage Server

「**Manage Server**」タブでは、NeoServer の実行時間とバージョンの表示、データベースと NeoServer の停止、ホットバックアップの実行を行います。

「**Show XMS Uptime**」ボタンを選択して、NeoServer の実行時間を表示できます。

「**Show XMS Version**」ボタンを選択して、NeoServer のバージョンを表示できます。



コンソールからの XMS Server の停止

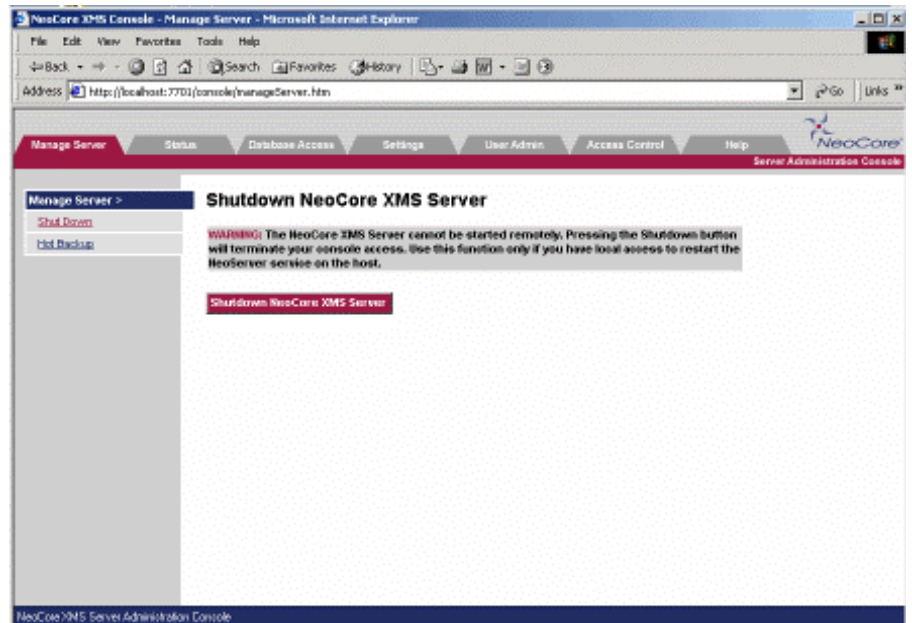
注意: XMS Server をコンソールから開始することはできません。

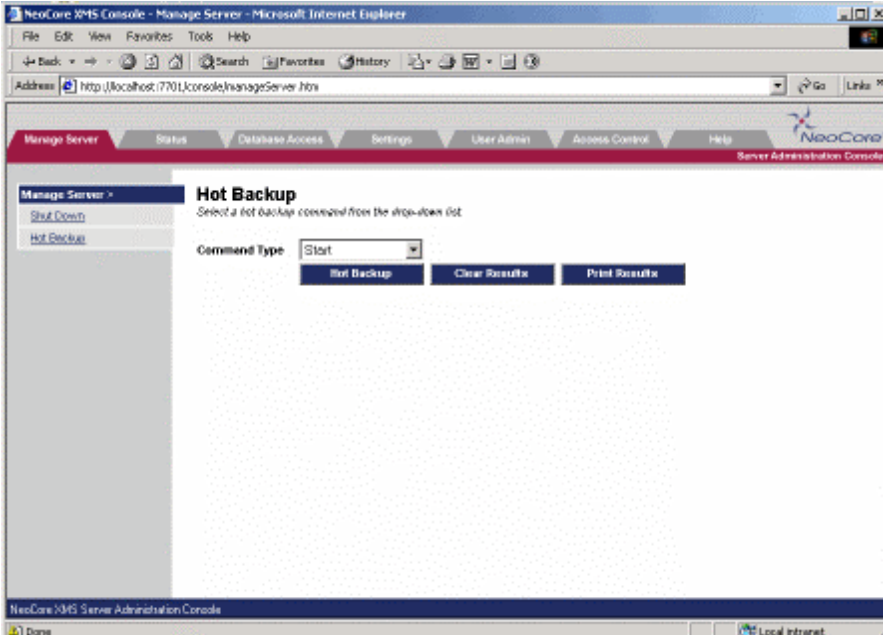
「**Shutdown**」ボタンを押すと、コンソールアクセスが停止します。この機能は、ホストの NeoServer サービスを再起動するためのローカルアクセス権を持っている場合にのみ使用してください。

「**Manage Server**」タブを選択します。

「**Shut Down**」ボタンを選択します。

「**Shutdown XMS Server**」ボタンを選択します。

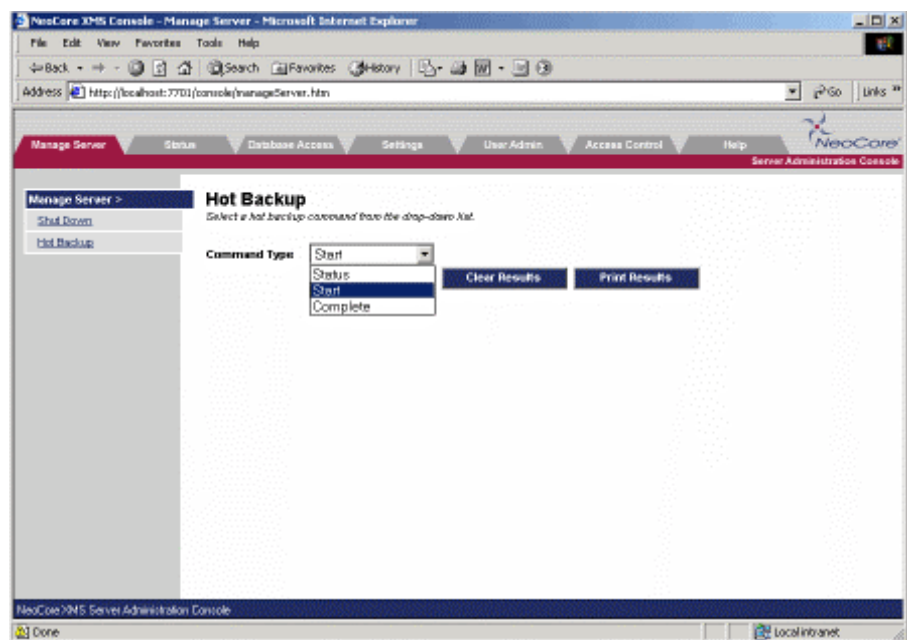


	<p>サーバーの再起動</p> <p>Windows（ローカル）：[コントロールパネル] > [管理ツール] > [コンピュータの管理] > [サービスとアプリケーション] > [サービス] を選択します。NeoServer を選択し、右クリックして [開始] を選択します。</p> <p>Solaris（ローカル）:rootとしてログインし、以下のように入力します。</p> <pre>/etc/init.d/neocore start</pre> <p>別の開始方法:</p> <pre>NeoServer [-d] [-r <RootDir>]</pre> <p>例:</p> <pre>NeoServer -d -r /usr/local/NeoCore</pre> <p>リモート: [スタート] > [設定] > [コントロールパネル] > [管理ツール] > [コンピュータの管理] > [操作] メニュー> [別のコンピュータへ接続] > [サービスとアプリケーション] > [サービス] を選択し、NeoServer を開始します。</p>
<p>この画面では、ホットバックアップコマンドオプションをドロップダウンリストから選択できます。</p> <p>結果のクリアと結果の印刷を行うこともできます。</p>	

Start – このコマンドで、ホットバックアッププロセスを開始します。

Status – このコマンドで、NeoServer が認識している限りのホットバックアップ手順の現在の状態を取得できます。例えば、手動バックアップの場合は、バックアップが保留中であることを示す状態だけが表示されます。

Complete – このコマンドで、ホットバックアップ手順を終了します。管理者がミラーボリュームの手動バックアップを行う場合は、このコマンドで HotBackupManager にバックアップが完了したことを通知できます。



Status

「**Status**」タブには、以下のオプションがあります。

Server Status

(NeoServer の状態の表示)

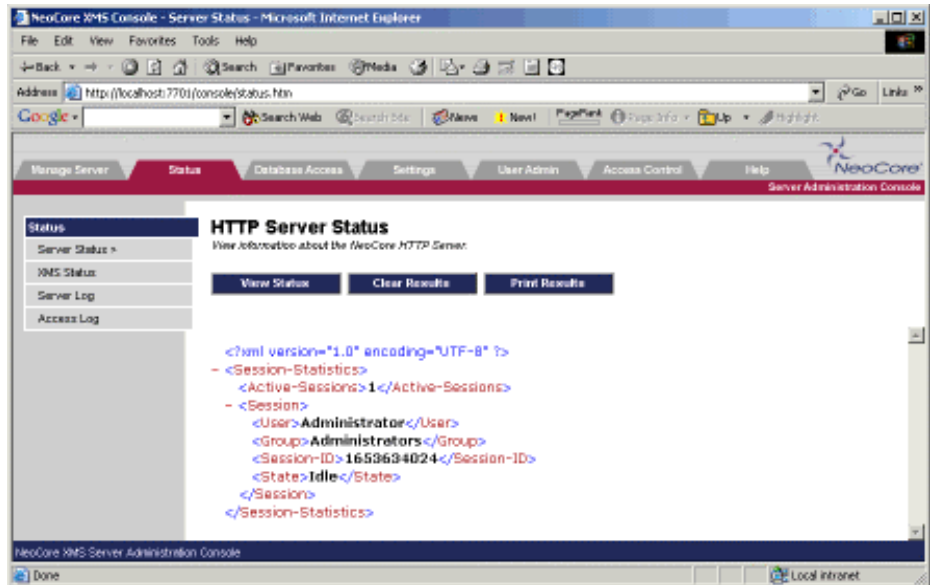
XMS Status (データベースの状態の表示)

Server Log (現在のデータベースログファイルの表示)

Access Log

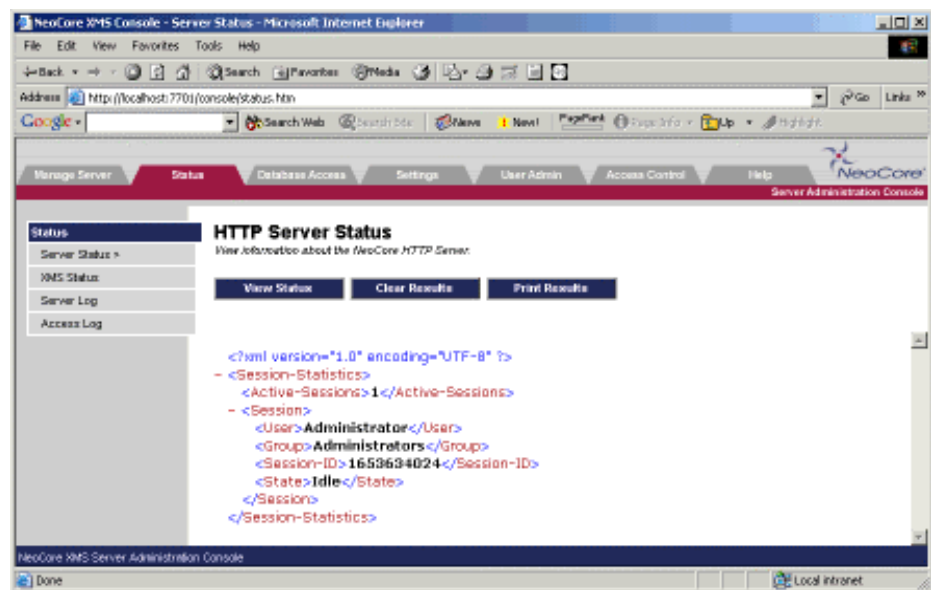
(HTTP アクセスログファイルの表示 [オンになっている場合])

状態とログの表示には、共通の機能があります。「**View Status**」や「**View Log**」は要求された情報を取得します。「**Clear Results**」は表示されている情報をクリアします。情報のソースを変更するわけではありません。「**Print Results**」で結果を印刷できます。



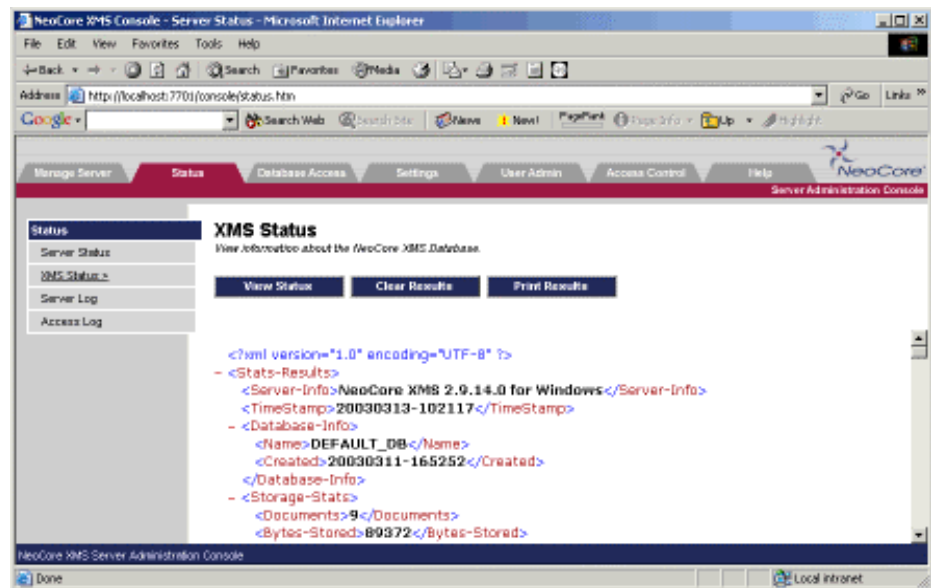
HTTP Server Status

NeoCore HTTP サーバーに関する情報を表示します。[View Status] ボタンをクリックすると、HTTP のサーバーの状態が取得されます。



XMS Status

データベースの状態に関する情報を表示します。[View Status] ボタンをクリックすると、データベースの状態が取得されます。



サーバーログの表示

NeoServer のログを表示します。[View Log] ボタンをクリックすると、現在開いているログファイルが表示されます。ログは24時間ごとに切り換えられるので、前のファイルを見るには、ファイルをテキストエディタで開く必要があります。サーバーログファイルは以下の場所にあります。

C:\<NEOCORE HOME>\%neoxml%\log

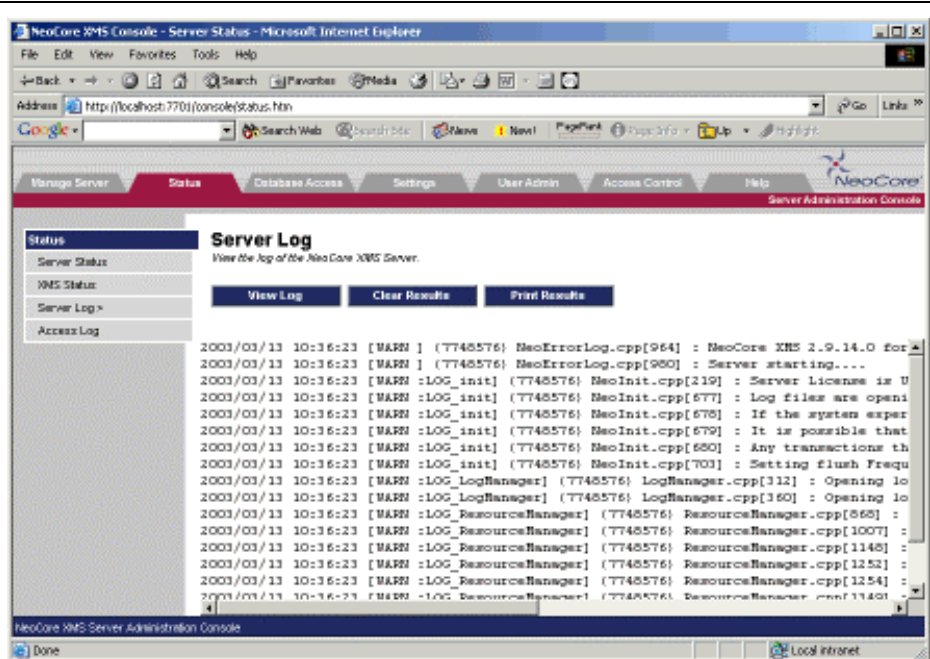
「C:」は NeoCore XMS のインストール先のドライブ、<NEOCORE HOME> は NeoServer のインストールディレクトリです。

サーバーログの例:

```
c:\NeoCore\%neoxml%\log
e:\%bob%\neoxml%\log
```

サーバーログファイルは以下の場所にあります。

```
{NeoCore インストールディレクトリ}
%neoxml%\log
(Windows)
{NeoCore インストールディレクトリ}
/neoxml/log
(Solaris)
```

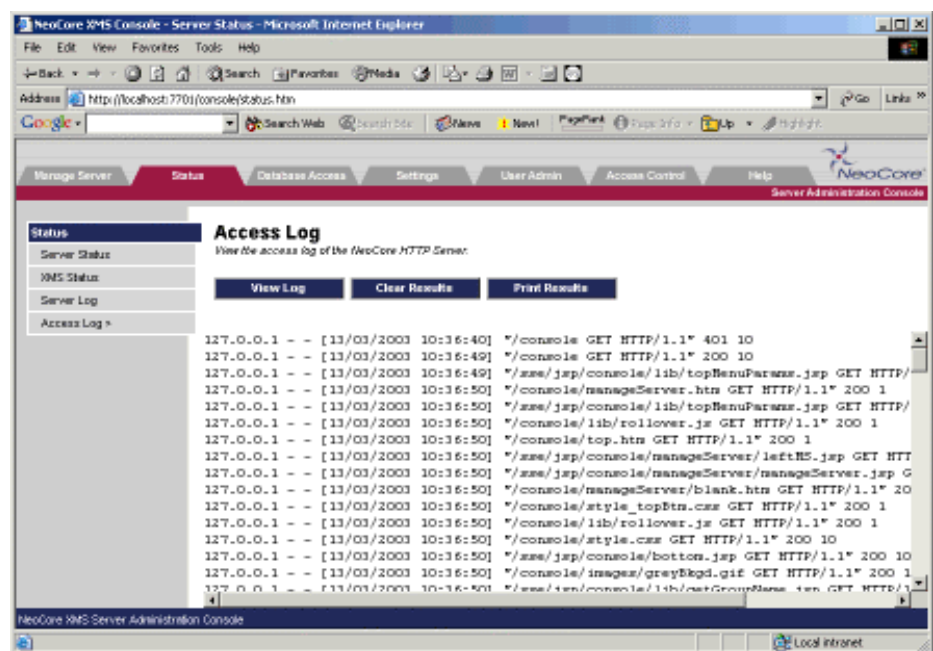


アクセスログの表示

NeoCore HTTP サーバーのアクセスログを表示します。

アクセスログが利用できない場合は、以下のようなメッセージが表示されます。

"The Access Log is currently unavailable. To enable this feature, set the value of the AccessLog tag in NeoServer.xml to ON and restart the server. To view the current server settings, click on `localhost:7701/console/settings.htm`. If you feel this is in error, see your Administrator."



注: アクセスログには、NeoCore HTTP サーバーとの間でやり取りしたメッセージが記録されます。大量のトランザクションがあるデータベースの場合は、大きなログファイルが作成されます。このため、デフォルトではこのログはオフになっています。このログは、トラブルシューティングするときだけにオンにしてください。

Database Access

[Database Access] タブ

このタブでは、データベースにログインして、以下の機能を使用できます。

Store (XML 文書の格納)

Copy (XML 文書のコピー)

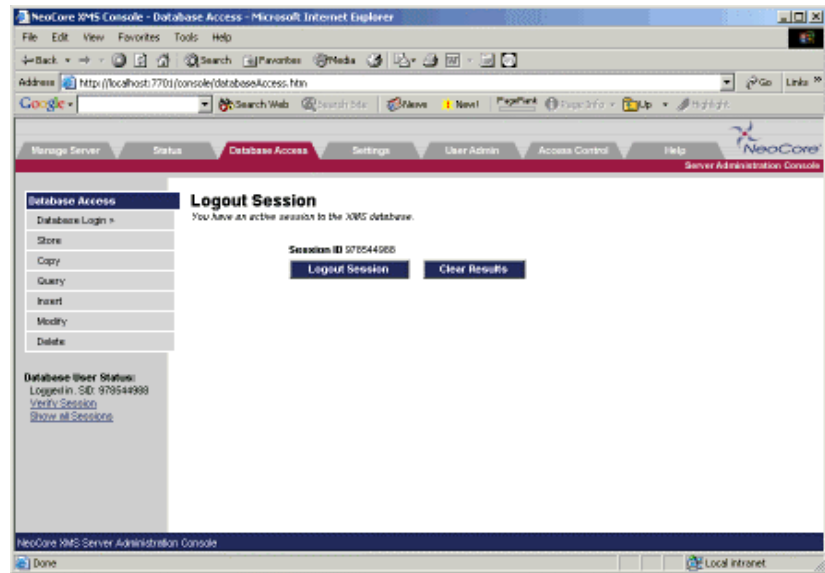
Query (XPath を使用したサーバーのクエリー)

Insert (既存の文書への XML の挿入)

Modify (XML 文書内の要素または属性の変更)

Delete (データベース内の XML の削除)

注: これらのコマンドを使用するための詳細については、このガイドの「**アクセス制御とユーザー管理**」を参照してください。



XMS へのログイン

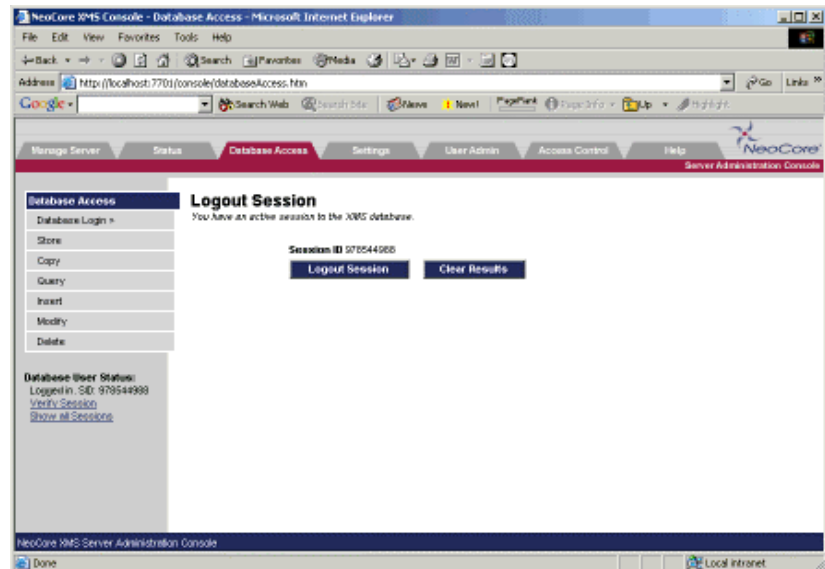
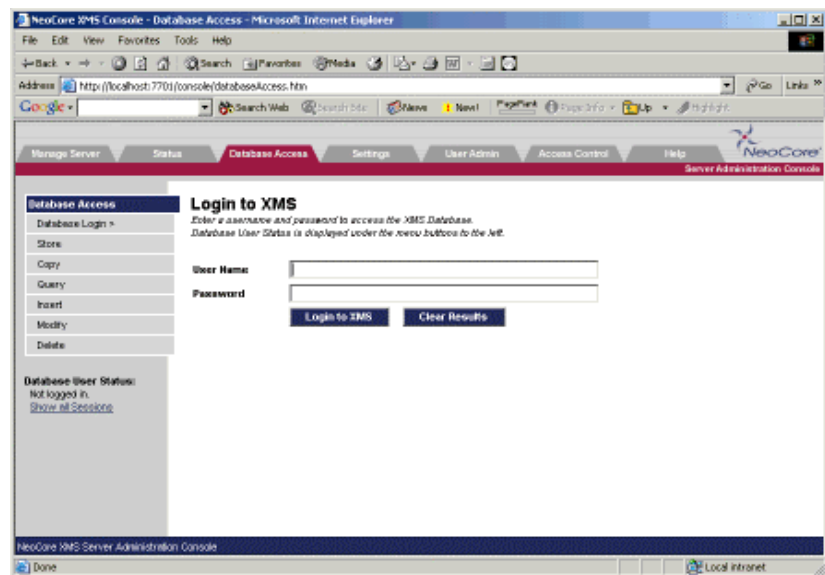
データベースにアクセスするためのユーザー名とパスワードを入力します。

ログイン後、セッション ID が表示されます。この ID は NeoServer の実行時間内でこのインスタンスに固有の ID です。これは管理上の問題を追跡するのに便利です。ログイン後には、**[Verify Session]** リンクと **[Show all Sessions]** リンクや、他の管理ツールも表示されます。

XMS からのログアウト

[Logout Session] を選択して、セッションを終了します。

[Clear Results] を選択すれば、結果をすべてクリアできます。



XML の格納

XMS に格納する XML ファイルを入力します（オプションとして、スキーマ参照とプリフィックスファイルも入力できます）。

文書を格納するときに、以下の格納用パラメータに注意してください。

PrefixFile – オプション – 文書のメタデータに追加するプリフィックスファイルの位置パラメータ

SchemaFile – オプション

同じルートタグを持った文書が複数ある場合は、それらを 1 つのファイルに入れて格納できます。その場合、ファイル内の各文書は別々に格納されます。



XMLFile – 必須

XML を格納するための詳細については、「**NeoCore XMS トランザクションとコマンド**」を参照してください。

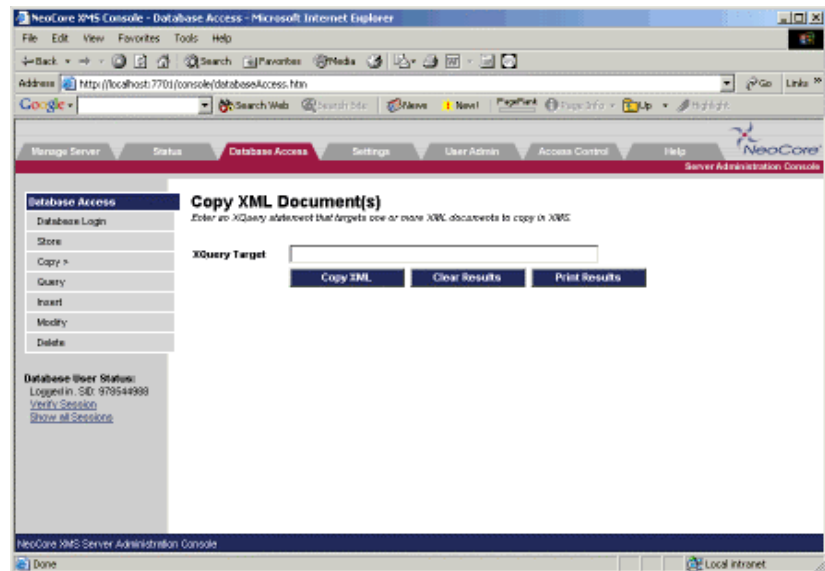
注: 多数のファイルを格納する（一括して読み込む）場合は、**XMS Loader** を使用してください。

XML 文書のコピー

コピートランザクションでは、文書のコピーを NeoCore XMS に格納します。文書をコピーすると、新しいコピーには、前のコピーよりも 1 つ大きい値の **CopyNumber** タグが **MetaData** 内に割り当てられます。この番号によって、格納されているコピーを区別できます。文書のコピーを多数格納した場合でも、**DocID** と **SourceFile** は同じままです。

XMS 内に XML 文書をコピーするための XPath ターゲットを入力します。この XPath ターゲットは文書レベルである必要があります。

XML をコピーするための詳細については、「**NeoCore XMS トランザクションとコマンド**」を参照してください。

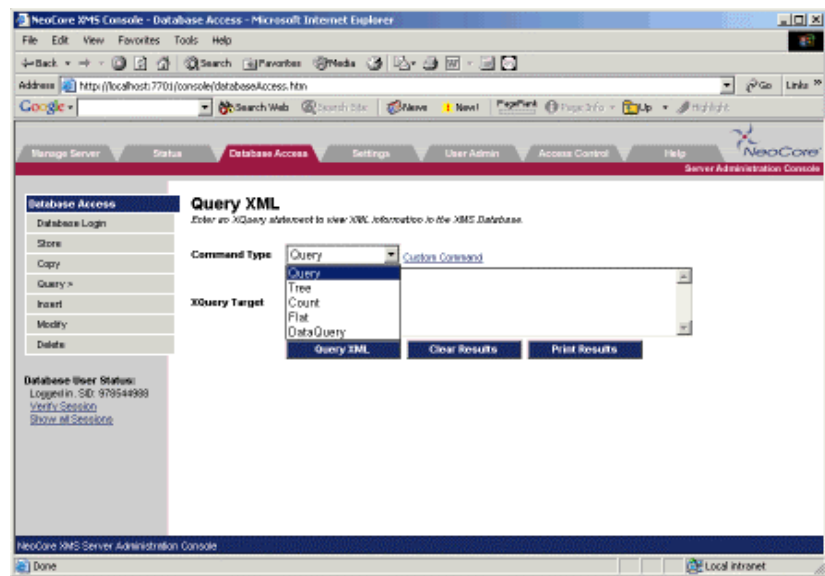


XML のクエリー

XMS データベース内の XML 情報を表示するための XPath ターゲットを入力します。

[**Command Type**] には 5 つの選択肢があります。つまり、[**Query**]、[**Tree**]、[**Count**]、[**Flat**]、[**DataQuery**] の 5 つです。[**Custom Command**] を入力することもできます。XPath ターゲットも入力する必要があります。

XML のクエリーを実行するための詳細については、「**NeoCore XMS トランザクションとコマンド**」を参照してください。

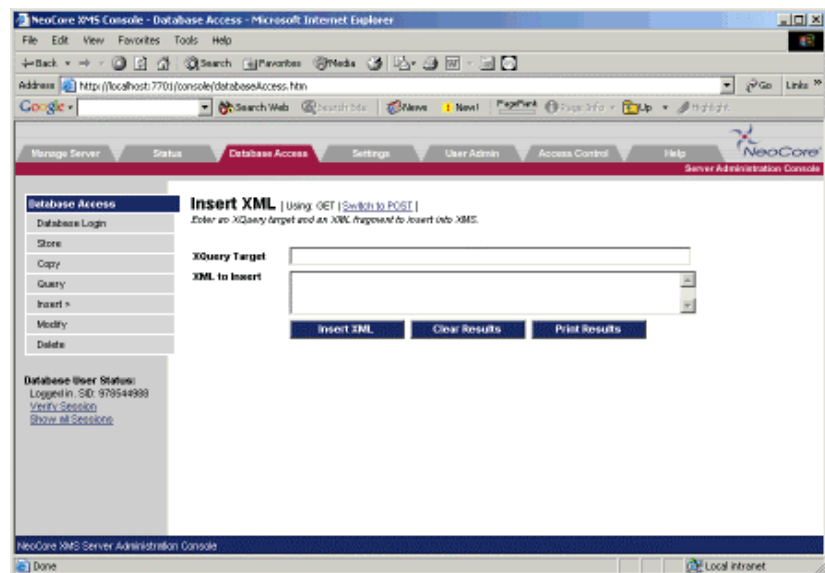


XML の挿入

挿入トランザクションでは、データストア内の既存の XML に XML を追加します。XPath 式を使用して、ターゲットノードを指定します（挿入はそのターゲットの後に行われます）。

XPath ターゲットと、XMS に挿入する XML フラグメントを入力します。

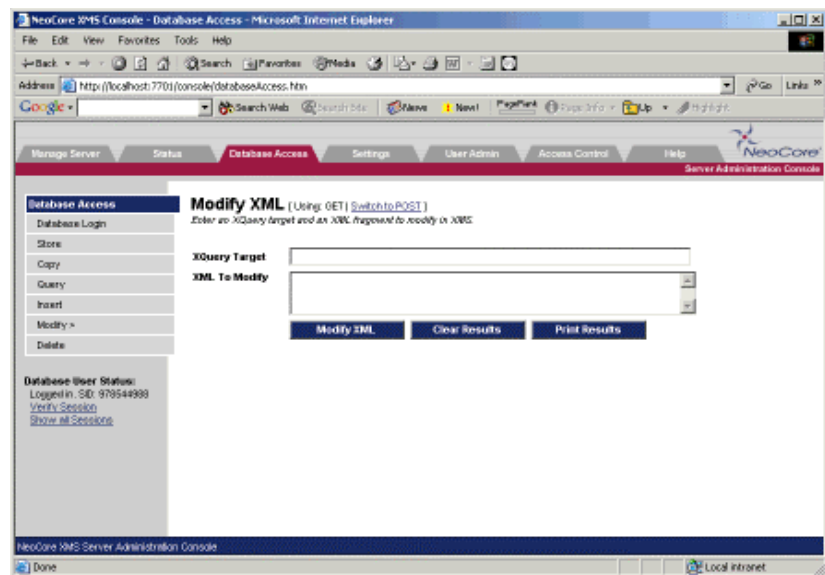
XML を挿入するための詳細については、「**NeoCore XMS トランザクションとコマンド**」を参照してください。



XML の変更

変更トランザクションでは、XML の既存データを変更します。タグ構造（メタデータ）はそのまま残ります。XPath ターゲットと、XMS 内で変更する XML フラグメントを入力します。

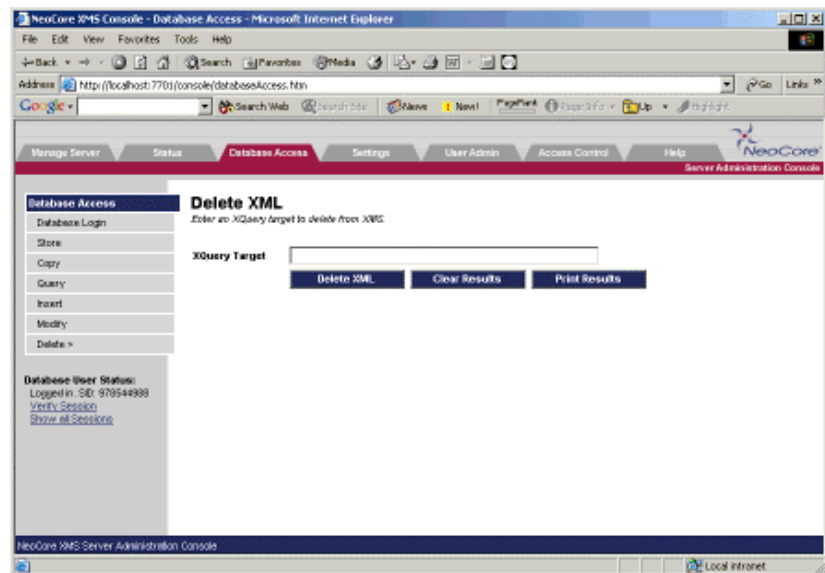
XML を変更するための詳細については、「**NeoCore XMS トランザクションとコマンド**」を参照してください。



XML の削除

削除トランザクションでは、データストアから XML を削除します。削除対象を指定する XPath 式を入力します。

XML を削除するための詳細については、「**NeoCore XMS トランザクションとコマンド**」を参照してください。



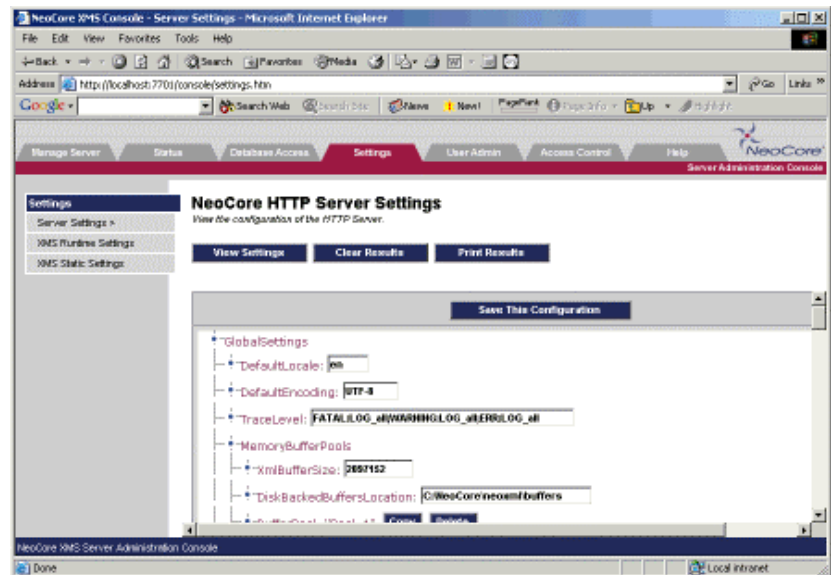
Settings

【Settings】タブには、以下のオプションがあります。

Server Settings

XMS Runtime Settings

XMS Static Settings



これらの設定画面では、設定を表示できるだけでなく、設定ファイルをエディタ等から直接編集しないで設定内容を変更することもできます。すべての設定変更は、【Settings】タブから直接行うことを推奨します。

注：構成ファイルパラメータのオンライン変更もサポートされています。

ただし、変更した値はリアルタイムで動的に反映されるわけではないので、変更した値を使用するには、管理者が NeoServer をいったん停止してから再起動する必要があります。値の検証は現在行われていないので、不正な値のために NeoServer が起動しなくなり、問題のある構成ファイルを手動で修正することが必要になる場合もあります。

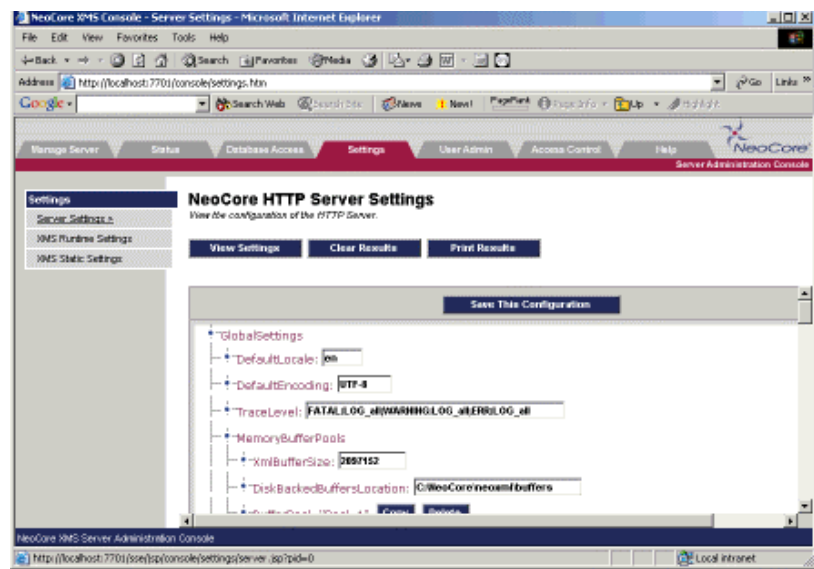
注：設定の変更を有効にするには、NeoServer を再起動する必要があります。

一部の設定ノードには【Copy】ボタンと【Delete】ボタンがあります。【Copy】ボタンは、ディープコピー、つまり親とそのすべての下位項目のコピーを行います。例えば、【Copy】ボタンを使用して、バッファプールを追加したり、別の NIC のためのサポートを追加できます。【Delete】ボタンも、親とそのすべての下位項目を削除します。

【Copy】ボタンと【Delete】ボタンは、指定したノードだけに変更を加えます。【Copy】や【Delete】の前に他のノードを変更した場合は、まず【Save This Configuration】を実行してください。

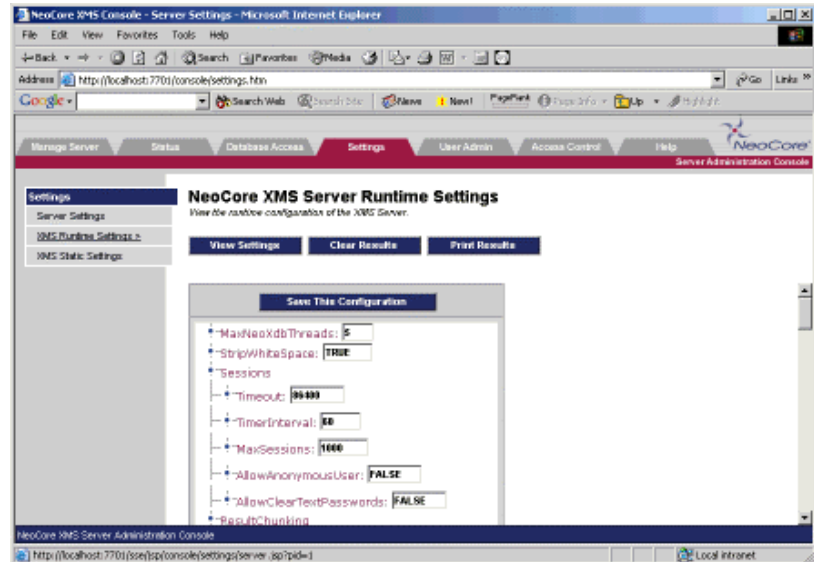
NeoCore HTTP Server Settings

これらはこのコンソールと HTTP インターフェースの設定です。



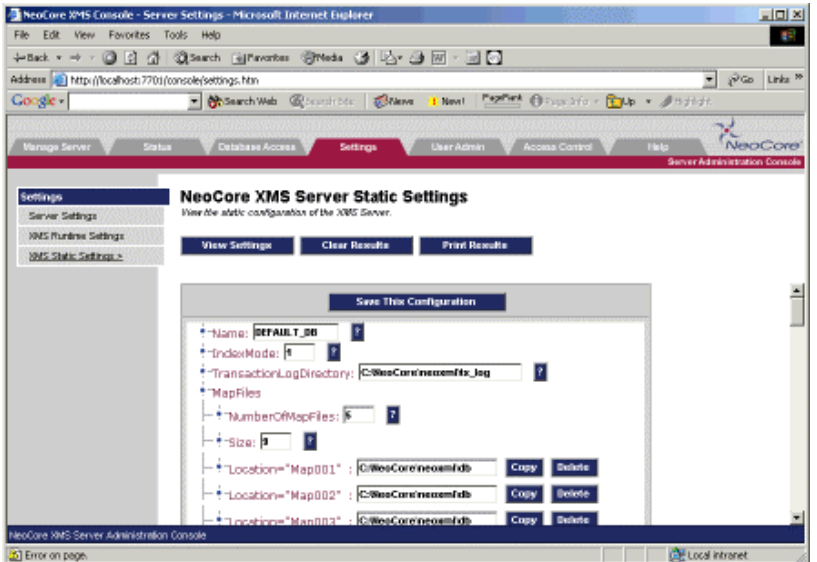
NeoCore XMS Server Runtime Settings

ランタイム設定です。これらの設定はいつでも更新できます。ただし、NeoServer を再起動するまでは有効になりません。



NeoCore XMS Server Static Settings

これらは、データベースの作成、データベースの再構築、インデックスの再構築のときだけに適用される設定です。



User Admin

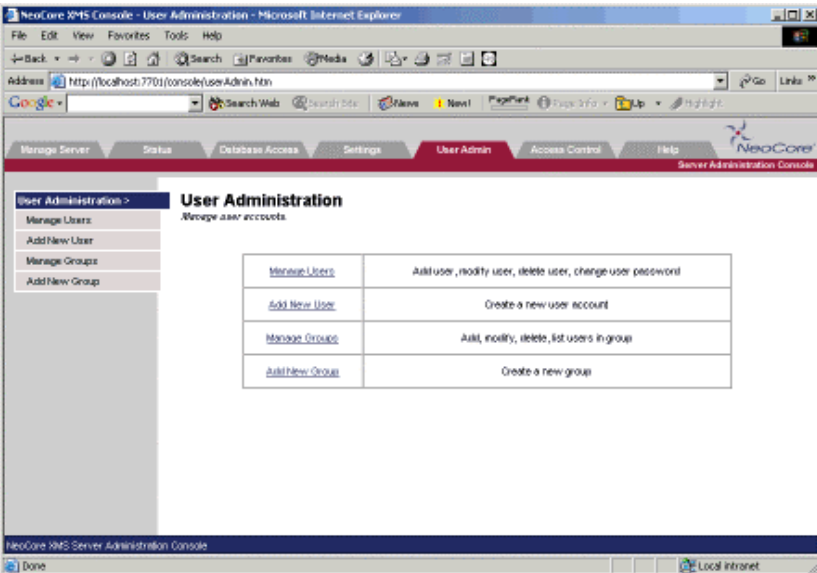
[User Admin] タブには、以下のオプションがあります。

Manage Users

Add New User

Manage Groups

Add New Group



[Manage Users] を選択すると、
[User Administration] ダイア
ログが表示されます。

このダイアログから以下の作業
を行えます。

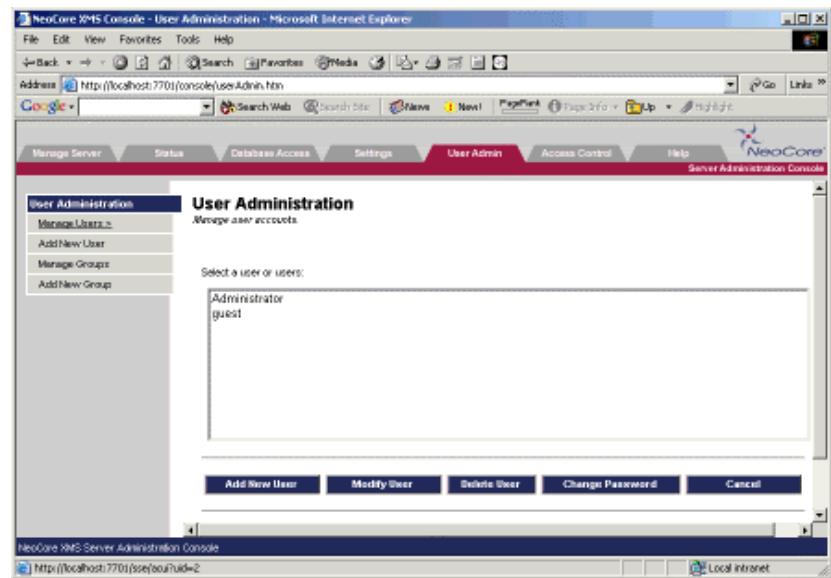
新規ユーザーの追加

ユーザーの変更

ユーザーの削除

パスワードの変更

[Cancel] での終了



[Add New User] を選択するか、[User Administration] ダイアログで [Add New User] オプションを選択すると、[Add a new user] ダイアログが表示されます。

新規ユーザーを追加するには、以下のようにします。

1. 以下のフィールドに入力します。

User Name

Password

Confirm Password

2. [Group] ドロップダウンリストからグループを選択します。

3. [First Name] と [Last Name] を入力します。

4. [Create User] を選択します。

このダイアログを終了するには、[Cancel] を選択します。

入力した値を消去するには、[Clear] を選択します。

エラーのためにすべてをやり直すには、[Clear Results] を選択します。

The screenshot shows a web browser window titled "NeoCore VMS Console - User Administration - Microsoft Internet Explorer". The address bar shows "http://localhost:7701/console/UserAdmin.htm". The browser has several tabs open, including "Search Web", "Search Site", "Wave", "New1", "PlayPoint", "Page Info", and "Links". The main content area displays the "User Administration" section with a sidebar menu containing "Manage Users", "Add New User", "Manage Groups", and "Add New Group". The "Add New User" option is selected, and the "Add a new user" dialog is shown. The dialog has the following fields: "User Name", "Password", "Confirm Password", "Group" (a dropdown menu), "First Name", and "Last Name". At the bottom of the dialog are four buttons: "Create User", "Cancel", "Clear", and "Clear Results". The status bar at the bottom of the browser window shows "Done" and "Local intranet".

「**User Administration**」ダイアログで「**Modify User**」オプションを選択すると、「**Modify User**」ダイアログが表示されます。

最初にユーザーを選択する必要があります。

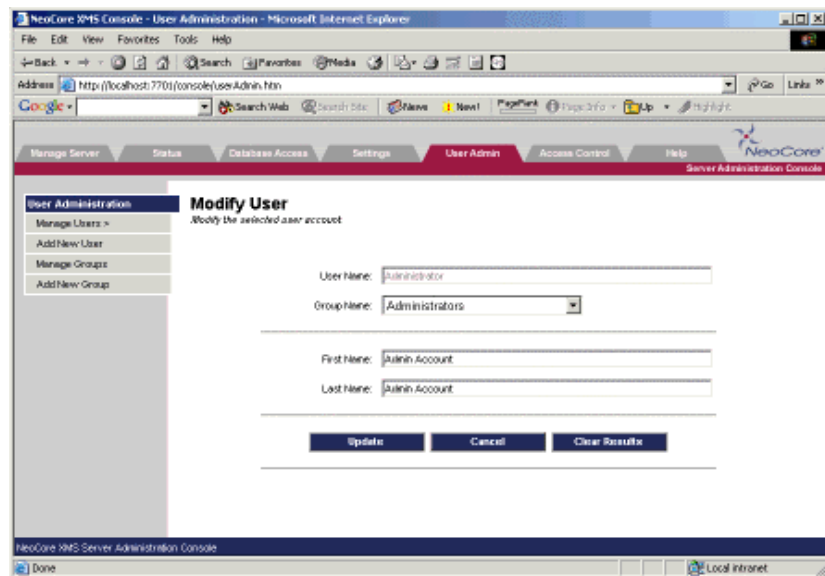
このユーザーが属するグループを変更するには、グループのプルダウンリストから対象のグループを選択します。

「**First Name**」や「**Last Name**」の値を変更することもできます。

変更を行うには、「**Update**」をクリックします。

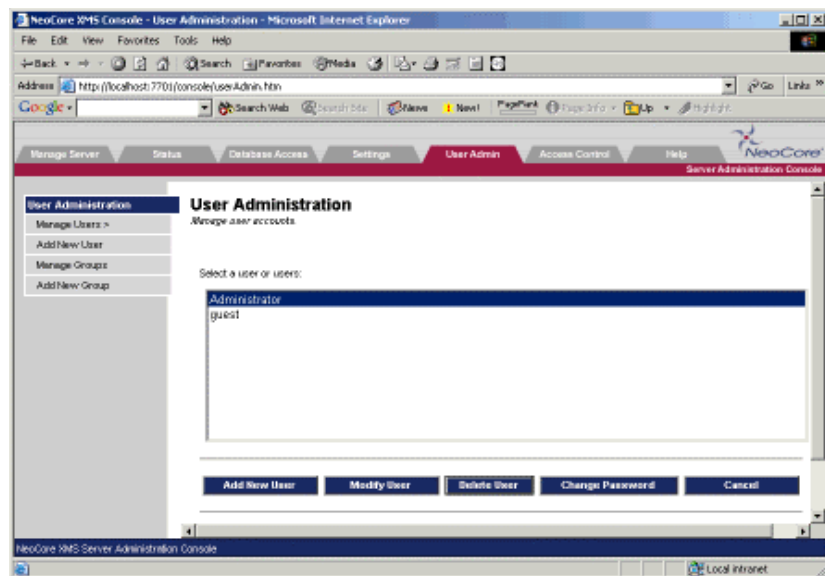
このダイアログを終了するには、「**Cancel**」を選択します。

エラーのためにすべてをやり直すには、「**Clear Results**」を選択します。



「**User Administration**」ダイアログで「**Delete User**」オプションを選択すると、ダイアログが表示されます。

削除するには「**OK**」、中止するには「**Cancel**」をクリックします。



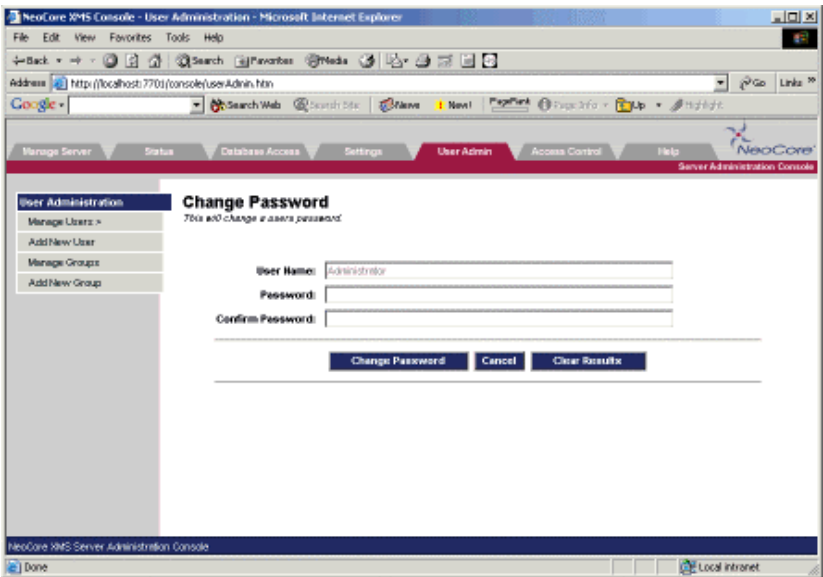
「**User Administration**」ダイアログで「**Change Password**」オプションを選択すると、「**Change Password**」ダイアログが表示されます。

パスワードを変更するには、以下のようにします。

1. 新しいパスワードを入力します。
2. 新しいパスワードを「**Confirm Password**」ボックスにもう一度入力します。
3. 「**Change Password**」をクリックします。

このダイアログを終了するには、「**Cancel**」を選択します。

エラーのためにすべてをやり直すには、「**Clear Results**」を選択します。



「**User Admin**」タブで「**Manage Groups**」を選択すると、「**Group Administration**」ダイアログが表示されます。このダイアログから以下の作業を行えます。

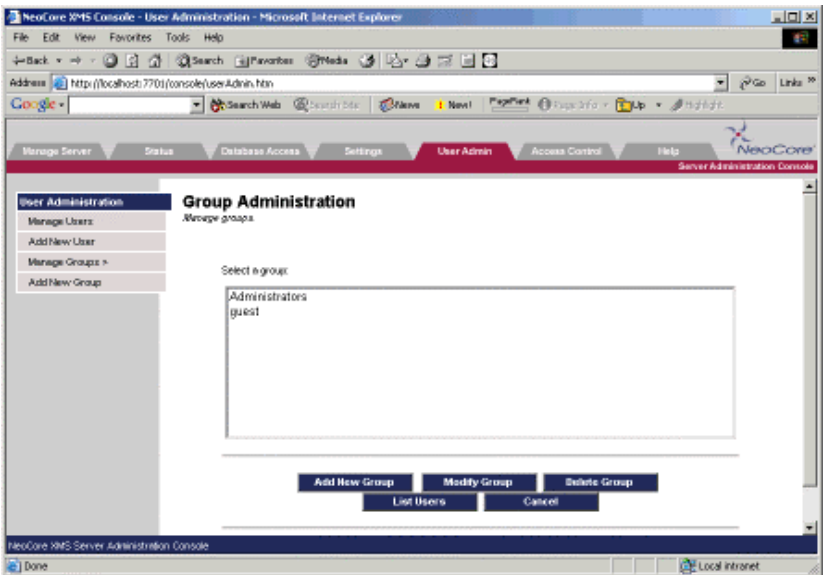
新規グループの追加

グループの変更

グループの削除

ユーザー一覧の表示

キャンセル

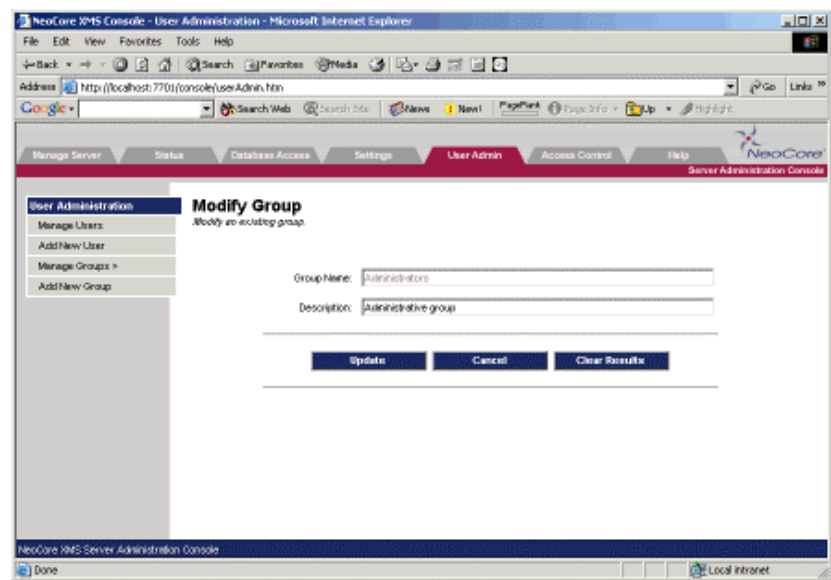


グループの変更

グループの説明を変更するには、新しい説明を「**Description**」ボックスに入力します。

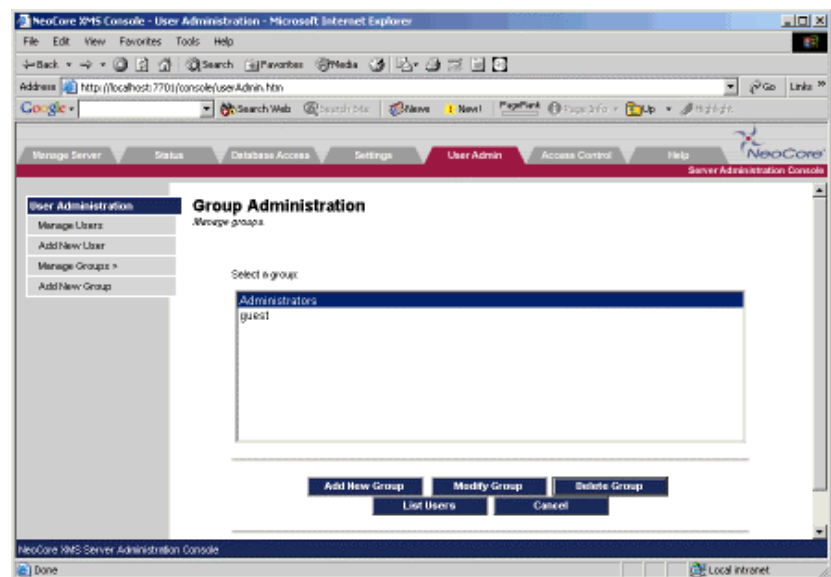
このダイアログを終了するには、「**Cancel**」を選択します。

エラーのためにすべてをやり直すには、「**Clear Results**」を選択します。



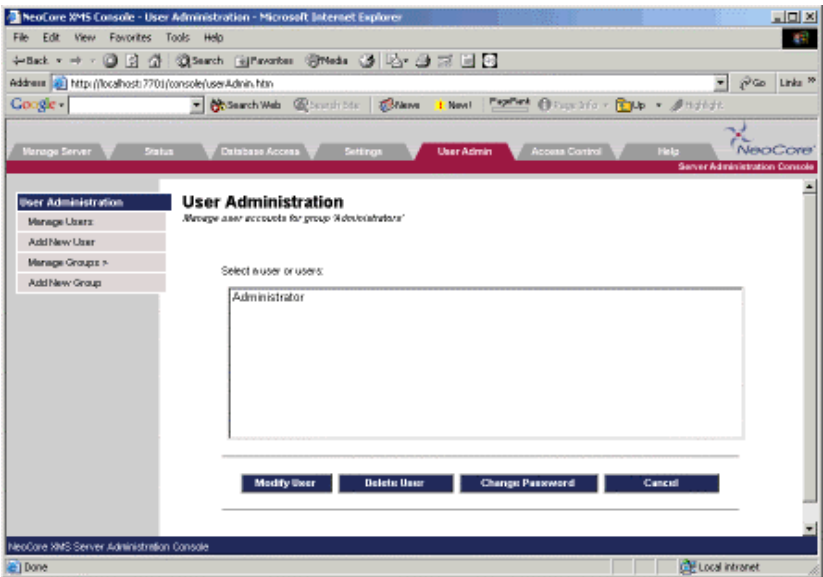
「**Group Administration**」ダイアログで「**Delete Group**」オプションを選択すると、「**Delete**」ダイアログが表示されます。

削除するには「**OK**」、中止するには「**Cancel**」をクリックします。



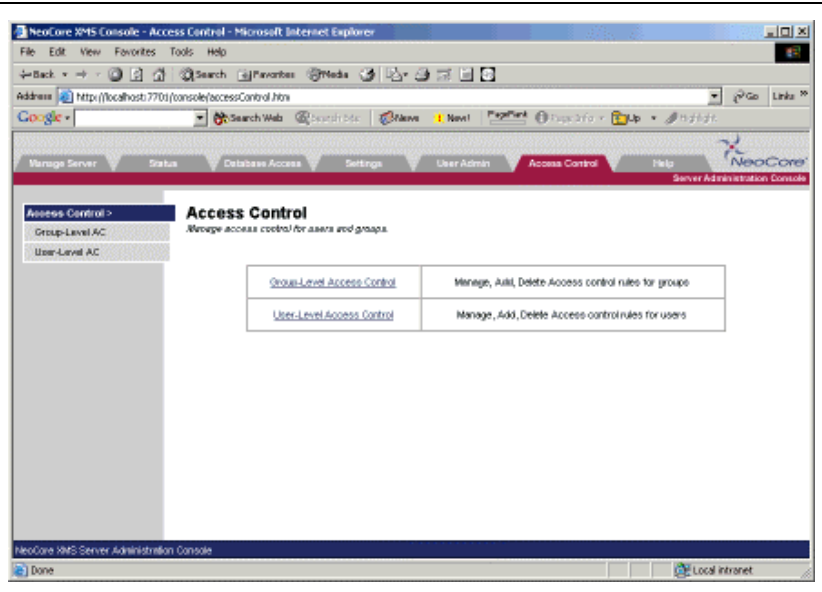
[Group Administration] ダイアログで [List Users] オプションを選択すると、[User Administration] ダイアログが表示されます。このダイアログには、現在のすべてのユーザーが一覧表示されます。

ユーザーの変更、削除、パスワードの変更を行えます。



Access Control

[Access Control] タブでは、ユーザーとグループのどちらのアクセス制御を行うかを選択できます。



「**Group-Level AC**」を選択すると、**「Group Level Access Control」**ダイアログが表示されます。

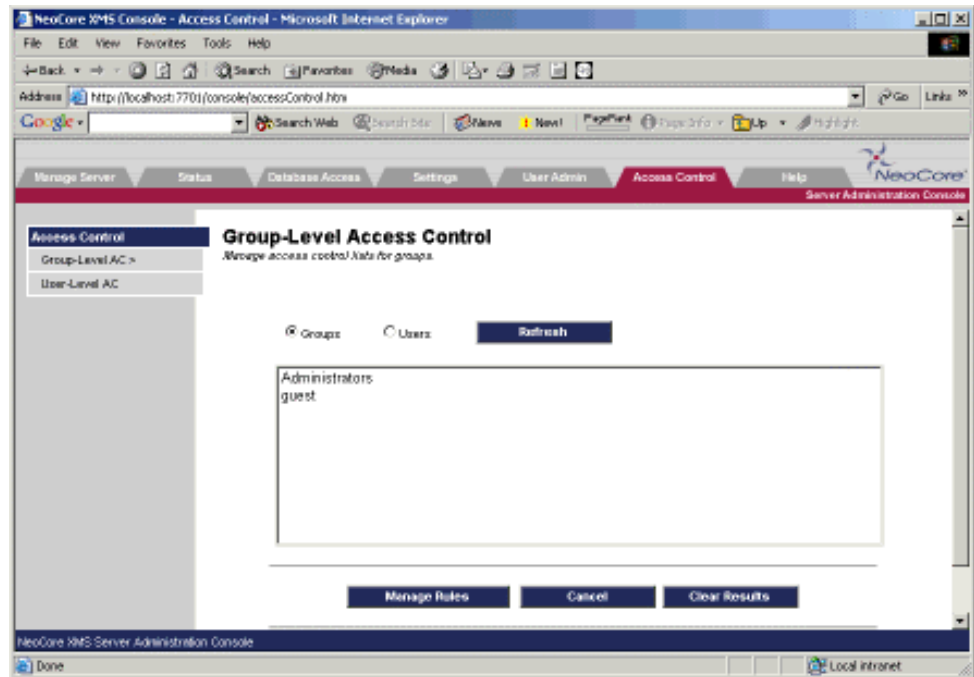
中央のペインに現在のグループが表示されます。

「**Refresh**」は現在のグループ一覧を更新します。

アクセス制御ルールを管理するには、グループを選択してから「**Manage Rules**」をクリックします。

このダイアログを終了するには、「**Cancel**」を選択します。

エラーのためにすべてをやり直すには、「**Clear Results**」を選択します。



「**Manage Access Control Rules**」ダイアログには、選択したグループと関連付けられたルールがすべて表示されます。

ルールごとに、
「**XPath**」、
「**Command**」、
「**Condition**」が表示されます。

このダイアログで
ルールの表示と削除を実行できます。

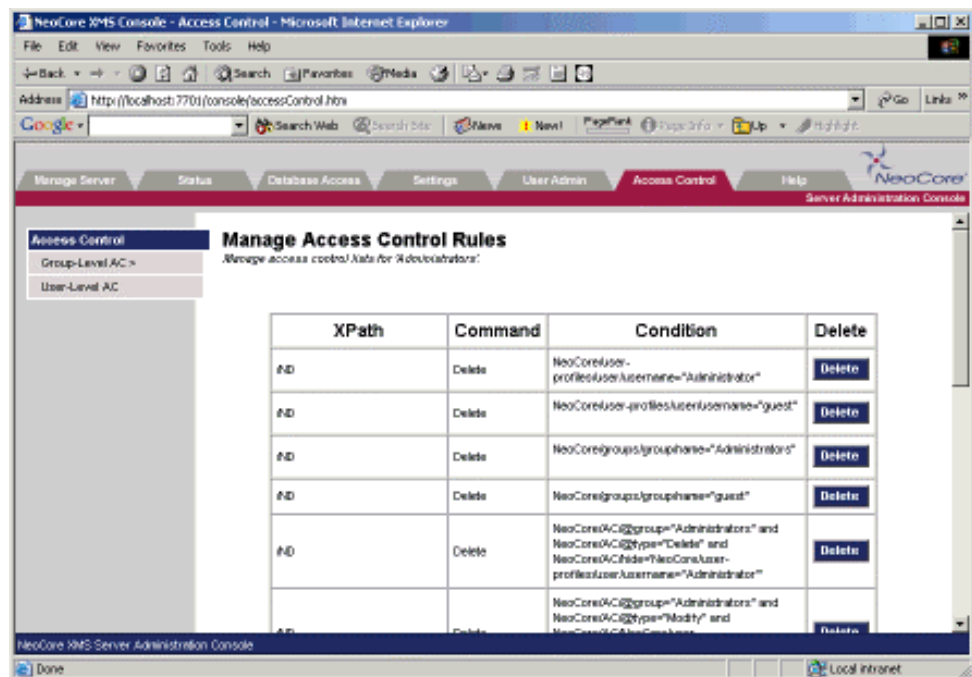
ルールを削除するには、該当する行の
「**Delete**」ボタンをクリックします。

注: いくつかのデフォルトルールは削除できません。

「**Add New Rule**」を選択すると、
「**Add New Access Control Rule**」ダイアログが表示されます。

このダイアログを終了するには、
「**Cancel**」を選択します。

エラーのためにすべてをやり直すには、
「**Clear Results**」を選択します。



XPath ターゲットを追加するために

[Add New Rule] を選択した後、中央のペインの

[Selected XPath] を使用して、ルールを追加する XPath ターゲットを選択します。そのためには、ツリー内で対象のノードをクリックします。[+] をクリックすると階層ツリーが展開し、[-] をクリックすると縮小します。

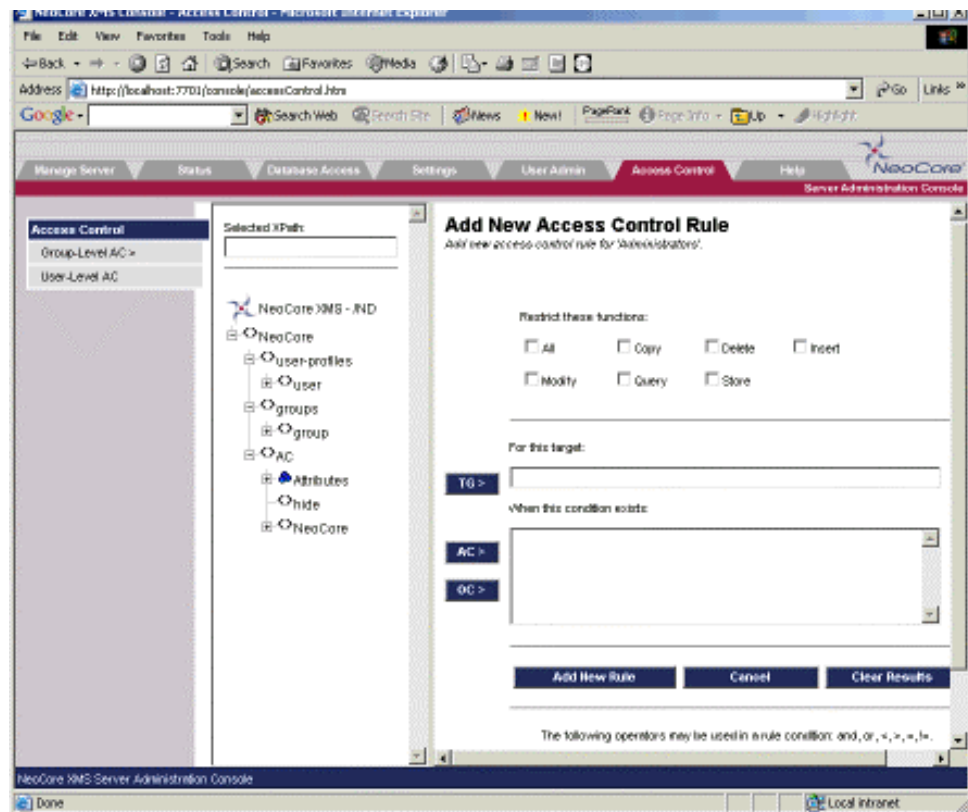
どのグループも、最初はすべてのコマンドにアクセスできる状態になっています。

XPath を選択した後、[Modify]、[Copy]、[Query]、[Delete]、[Store]、[Insert]、[All] を使用して、アクセスを制限する機能を選択します。

前に選択したターゲットパスを設定するには、[TG] をクリックします。（これは条件を適用する XPath ノードです。）

選択した XPath を条件に追加するには、[AC] を選択します。

条件を上書きするには、[OC] を選択します。



例: Xpath Target : /ND/CATALOG/CD で PRICE が\$9.00 を超える場合は、このグループの[Modify]と[Store]の機能に対するアクセスを制限します。

例: Target=/ND/CATALOG/CD Condition=PRICE > 9.00

演算子として、
AND、**OR**、**>**（より大）、**<**（より小）、**=**（等しい）、**!=**（等しくない）を使用できます。

このダイアログを終了するには、
[**Cancel**] を選択します。

エラーのためにすべてをやり直すには、[**Clear Results**] を選択します。

ユーザーのアクセス制御を選択すると、[**User-Level Access Control**] ダイアログが表示されます。

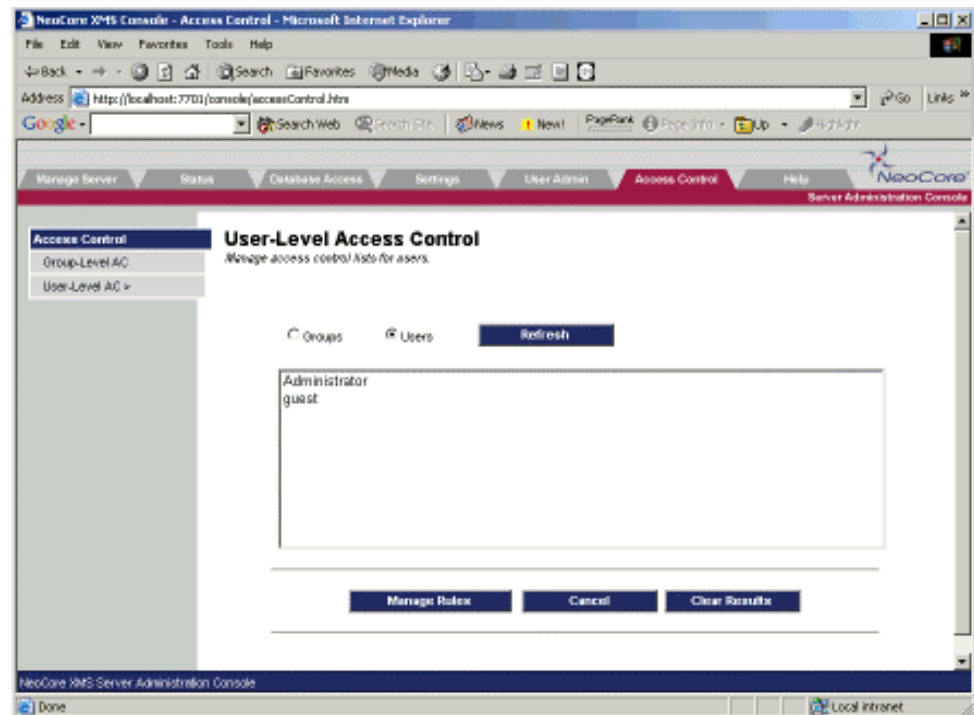
中央のペインに現在のユーザーが表示されます。

[**Refresh**] は現在のユーザー一覧を更新します。

アクセス制御ルールを管理するには、**ユーザー**を選択してから [**Manage Rules**] をクリックします。

このダイアログを終了するには、
[**Cancel**] を選択します。

エラーのためにすべてをやり直すには、[**Clear Results**] を選択します。



このダイアログには、選択したユーザーと関連付けられたルールがすべて表示されます。

ルールごとに、
[XPath]、
[Command]、
[Condition]が表示されます。

このダイアログでルールの表示と削除を実行できます。

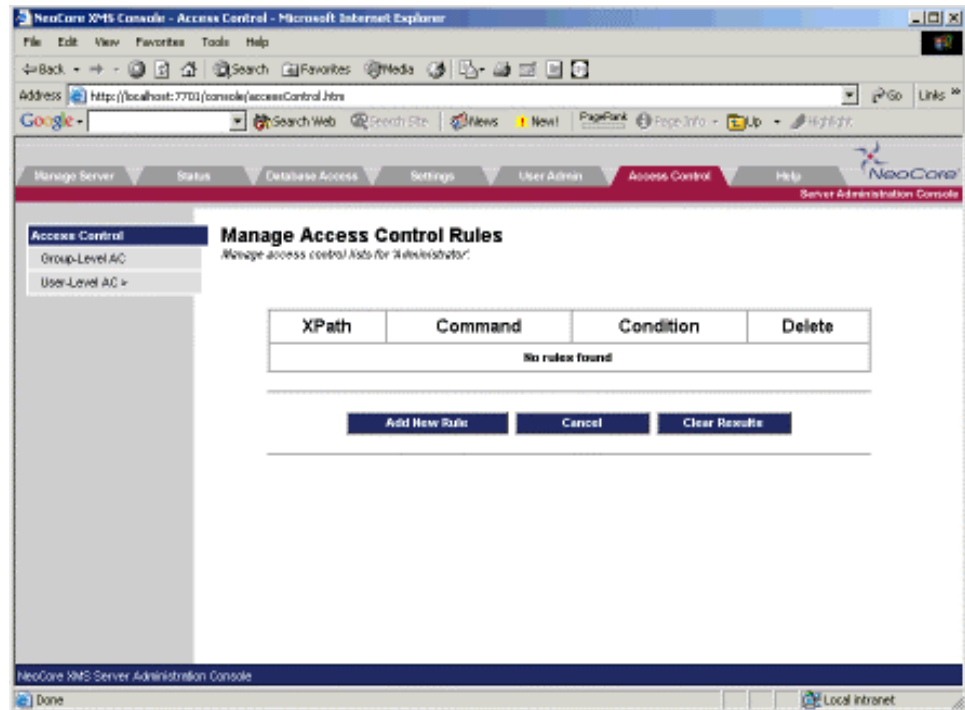
ルールを削除するには、該当する行の
[Delete] ボタンをクリックします。

注: いくつかのデフォルトルールは削除できません。

[Add New Rule] を選択すると、
[Add New Access Control Rule] ダイアログが表示されます。

このダイアログを終了するには、
[Cancel] を選択します。

エラーのためにすべてをやり直すには、
[Clear Results] を選択します。



注: [Copy] コマンドは文書全体を選択するため、ルールのコピーは、選択した XPath として/ND をターゲットにしている場合にのみ有効です。

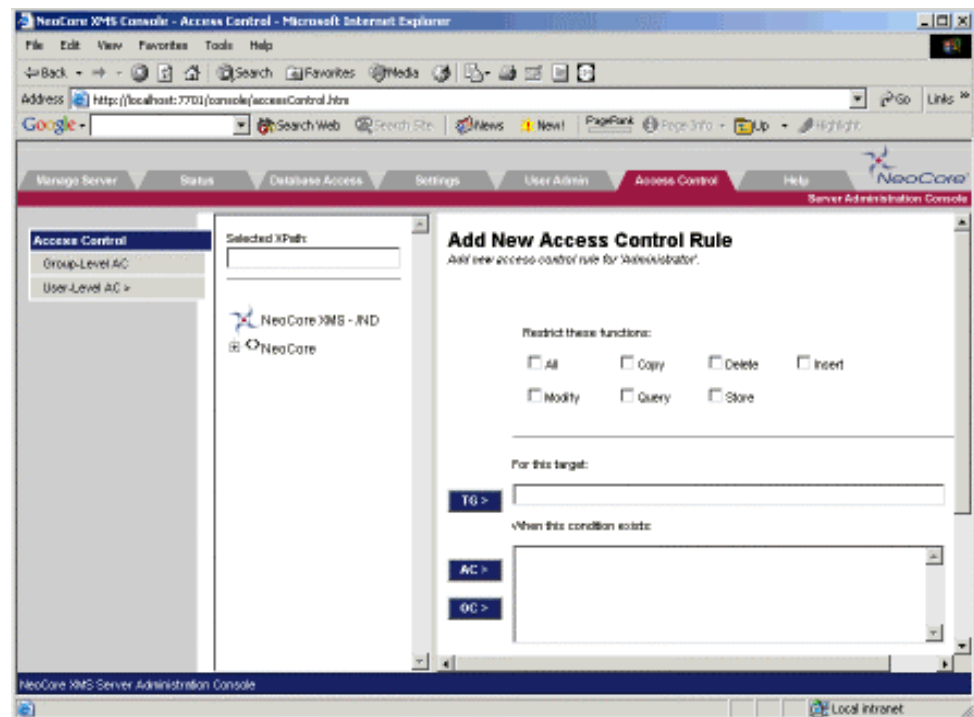
中央のペインの
[**Selected XPath**]
を使用して、ルール
を追加する XPath
ターゲットを選択
します。そのため
には、ツリー内で
対象のノードを
クリックします。
[+] をクリック
すると階層ツリー
が展開し、[-]
をクリックすると
縮小します。

どのユーザーも、
最初はすべての
コマンドにアクセ
スできる状態に
なっています。

XPath を選択し
た後、[**Modify**]
、[**Copy**]、[**Query**]
、[**Delete**]、
[**Store**]、
[**Insert**]、[**All**]
を使用して、アク
セスを制限する
機能を選択しま
す。

前に選択したタ
ーゲットパスを
設定するには、
[**TG**] をクリッ
クします。
(これは条件を
適用する XPath
ノードです。)

選択した XPath
を条件に追加す
るには、[**AC**]
を選択します。



例：Xpath Target : **/ND/CATALOG/CD** で **PRICE** が**\$9.00** を超える場合は、このグループの[**Modify**]と[**Store**]の機能に対するアクセスを制限します。

例：Target=**/ND/CATALOG/CD** Condition=**PRICE > 9.00**

注：データベースインスタンス全体にルールを適用するには、ルート XPath **/ND** をターゲットにします。

注：文書全体にルールを適用するには、**/ND** をターゲットにして、条件として文書のルートを選択します。

演算子として、
AND、**OR**、**>**（より大）、**<**（より小）、**=**（等しい）、**!=**（等しくない）を使用できます。

このダイアログを終了するには、
[**Cancel**] を選択します。

エラーのためにすべてをやり直すには、
[**Clear Results**] を選択します。

Help

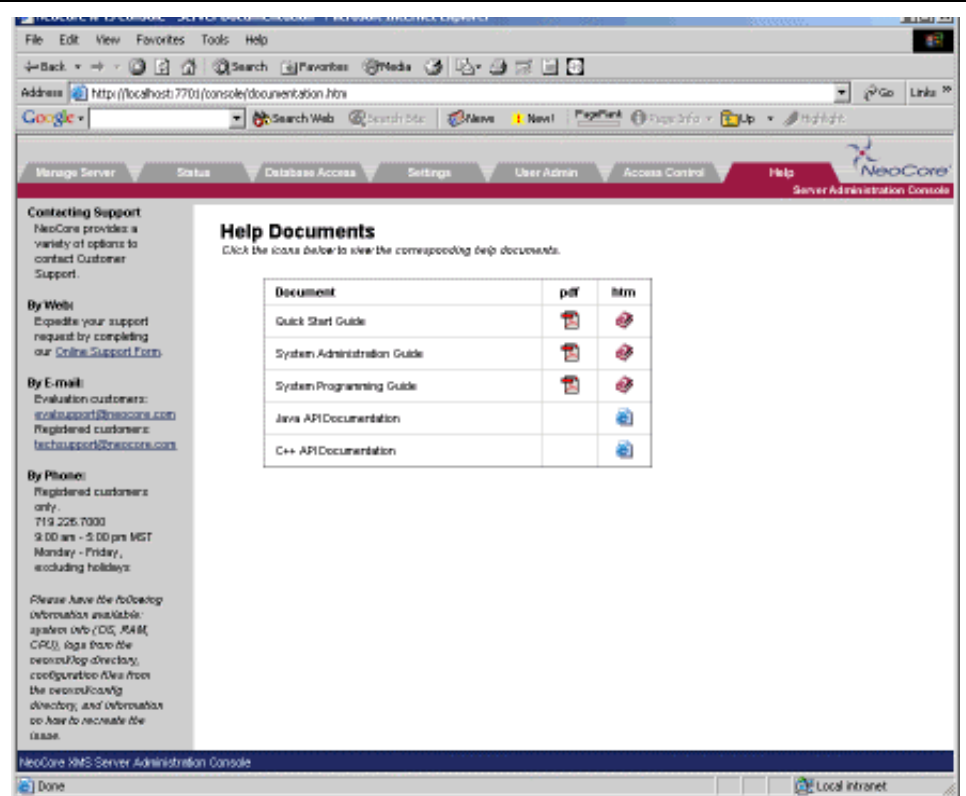
「Help」タブには、NeoCore XMSの使用に役立つ情報が用意されています。このページは 2 つの情報欄に分かれています。

1. Contacting Support

オンラインサポート
フォーム

E メール

電話



2. Help Documents

ここでは、オンライン HTML ヘルプを使用するか、以下の資料の現行の.pdf ファイルをダウンロードできます。

Quick Start Guide

System
Administration Guide

System
Programming Guide

また、Java API と C++ API のブラウザベースの資料を開くこともできます。

データベース管理

概要

NeoCore XMS のデータベース管理タスクには、構成作業、多様な管理コマンドで実行する「ハウスキューピング」、パフォーマンスのためのチューニングが含まれます。この章で取り上げるタスクを実行するには、**Administrator** ユーザー名とパスワードでログインする必要があります。

注：管理タスクはすべて NeoCore XMS の停止を必要とします。

NeoCore DB 管理コマンド構文

すべての NeoCore XMS データベース管理コマンドは、以下に示すコアコマンド構文に従っています。

すべての NeoCore XMS データベース管理作業には、以下のユーティリティを使用します。

- **NeoXMLUtils.exe** <---> (Microsoft Windows 用)
- **NeoXMLUtils** <---> (Sun Solaris 用)

注：Solaris プラットフォームで NeoXMLUtils を使用するには、**neocore** グループのメンバーである必要があります。

注：Microsoft Windows ではパスを指定するときの区切り文字として円記号 (¥) を使用し、Solaris ではスラッシュ (/) を使用します。

コマンド構文:

NeoXMLUtils command config-dir command options

左から順に、最初のものが NeoXMLUtils プログラム、次の **command** が実行する機能、次の **config directory** がデータベース構成ファイルの位置、最後の **command options** が使用する command パラメータに固有のオプションです。

Windows の例:

```
NeoXMLUtils CreatedB C:¥NeoCore¥neoxml¥config
NeoXMLUtils Export C:¥NeoCore¥neoxml¥config C:¥NeoCore¥neoxml¥export
```

Solaris の例:

```
NeoXMLUtils CreateDB \opt\NeoCore\neoxml\config
NeoXMLUtils CreateDB \opt\NeoCore\neoxml\config .\CreateDBtrace
NeoXMLUtils CreateDB \opt\NeoCore\neoxml\config .\CreateDBtrace INFO:LOG_all
```

(この場合は、デフォルトの FATAL、WARNING、ERROR のメッセージに加えて、INFO レベルのメッセージがトレースファイルに出力されます。)

構成ディレクトリの位置

構成ディレクトリのデフォルトは、以下のとおりです。

- Microsoft Windows – デフォルト設定は必須ではありません。デフォルト構成ディレクトリは、X:¥NeoCore¥neoxml¥config (X は NeoCore XMS のインストール先のドライブ) です。
- Sun Solaris – デフォルト設定のみを使用します。構成ディレクトリは、/opt/NeoCore/neoxml/config (XMS がデフォルトの /opt/NeoCore にインストールされている場合) です。

XML データベースの管理

XMS データベースの管理では、以下の操作を行います。

- データベースの作成
- 空のデータベースの作成
- データベースの削除
- データベースのクリア
- データベースのバックアップ
- データベースの復元
- データのエクスポート
- データのインポート
- インデックス再構築
- データベースの再構築
- 管理者の復元
- データベースファイル再配置コマンド:
 - MoveDataDictionary
 - MoveTagDictionary
 - MoveTagIndex
 - MoveTagIndexDups
 - MoveDataIndex
 - MoveDataIndexDups
 - MoveTagPlusDataIndex
 - MoveTagPlusDataIndexDups

データベースの作成

CreateDB の構文は、以下のとおりです。

```
NeoXMLUtils CreateDB config-dir (trace-file (trace-level))
```

CreateDB のパラメータを以下の表にまとめます。

パラメータ	説明
config-dir	これは NeoServer 構成ファイルの格納先のパスです。 config-dir パラメータは必須です。
trace-file	<p>データベース作成中にトレースファイルを出力したい場合は、その名前をここに指定します。例えば、次のようにします。</p> <pre>.%createdbtrace.log</pre> <p>デフォルトでは、CreateDB コマンドは log ディレクトリに NeoXMLUtils.log というログファイルを作成します。</p> <p>trace-level を指定しない限り、trace-file パラメータは必須ではありません。このパラメータのデフォルトは、以下のとおりです。</p> <pre>c:%neocore%neoxml%log%NeoXMLUtils.log</pre> <p>(Windows の場合)</p> <pre>/opt/NeoCore/neoxml/log/NeoXMLUtils.log</pre> <p>(Solaris の場合)</p>
trace-level	<p>トレースファイルを出力する場合、どのトレースレベルを使用するかを指定できます。トレースレベルを指定しないと、以下のデフォルト設定でトレースが行われます。</p> <pre>FATAL:LOG_all;WARN:LOG_all;ERR:LOG_all</pre>

有効な **CreateDB** コマンドの例を以下に示します。

- `NeoXMLUtils CreateDB C: ¥neocore¥neoxml¥config`
- `NeoXMLUtils CreateDB .. ¥config`

データベースが既に存在する場合、**CreateDB** コマンドはエラーを報告します。

デフォルトのアクセス制御を作成するかどうかを選択するためのプロンプトが表示されます。

注: 初期データの回復: Xpiori 社では、新規の NeoCore XMS データベースを (CreateDB コマンドで) 作成した直後に BackupDB を実行することを推奨します。こうすることにより、最初に予定されているバックアップより前のトランザクションを元に戻すことができます。

空のデータベースの作成

CreateEmptyDB の完全な構文は、以下のとおりです。

```
NeoXMLUtils CreateEmptyDB config-dir (trace-file (trace-level))
```

CreateEmptyDB のパラメータを以下の表にまとめます。

パラメータ	説明
config-dir	これは NeoServer 構成ファイルの格納先のパスです。 config-dir パラメータは必須です。
trace-file	<p>空のデータベース作成中にトレースファイルを出力したい場合は、その名前をここに指定します。例えば、次のようにします。</p> <pre>¥createemptydbtrace.log</pre> <p>デフォルトでは、CreateEmptyDB コマンドは log ディレクトリに NeoXMLUtils.log というログファイルを作成します。</p> <p>trace-level を指定しない限り、trace-file パラメータは必須ではありません。</p> <p>このパラメータのデフォルトは、以下のとおりです。</p> <pre>c:¥neocore¥neoxml¥log¥NeoXMLUtils.log</pre> <p>(Windows の場合)</p> <pre>/opt/NeoCore/neoxml/log/NeoXMLUtils.log</pre> <p>(Solaris の場合)</p>
trace-level	<p>トレースファイルを出力する場合、どのトレースレベルを使用するかを指定できます。トレースレベルを指定しないと、以下のデフォルト設定でトレースが行われます。</p> <pre>FATAL:LOG_all;WARN:LOG_all;ERR:LOG_all</pre>

有効な **CreateEmptyDB** コマンドの例を以下に示します。

- NeoXMLUtils CreateEmptyDB C: %neocore%\neoxml%\config
- NeoXMLUtils CreateEmptyDB .. %config

データベースが既に存在する場合、**CreateEmptyDB** コマンドはエラーを報告します。

注: 空のデータベースを作成すると、ログインできなくなります。**CreateEmptyDB** コマンドは、ユーザー/AC 文書の完全インポートを実行する前にのみ使用してください。このコマンドでは、前にエクスポートしたユーザー文書や AC 文書をインポートするための空のデータベースを作成できます。

データベースの削除

DeleteDB の完全な構文は、以下のとおりです。

```
NeoXMLUtils DeleteDB config-dir (trace-file (trace-level))
```

注: DeleteDB コマンドはデータベースファイルとトランザクションログを削除します。

DeleteDB のパラメータを以下の表にまとめます。

パラメータ	説明
config-dir	これは NeoServer 構成ファイルの格納先のパスです。 config-dir パラメータは必須です。
trace-file	<p>データベース削除中にトレースファイルを出力したい場合は、その名前をここに指定します。例えば、次のようにします。<code>.%deletedbtrace.log</code></p> <p>trace-level を指定しない限り、trace-file パラメータは必須ではありません。</p> <p>このパラメータのデフォルトは、以下のとおりです。</p> <pre>c:%neocore%\neoxml%\log%\NeoXMLUtils.log (Windows の場合) /opt/NeoCore/neoxml/log/NeoXMLUtils.log (Solaris の場合)</pre>
trace-level	<p>トレースファイルを出力する場合、どのトレースレベルを使用するかを指定できます。トレースレベルを指定しないと、以下のデフォルト設定でトレースが行われます。</p> <pre>FATAL:LOG_all;WARN:LOG_all;ERR:LOG_all</pre>

パスワードの指定を求めるプロンプトが表示されたら、管理者パスワードを入力する必要があります。

データベースのクリア

ClearDB の完全な構文は、以下のとおりです。

```
NeoXMLUtils ClearDB config-dir (trace-file (trace-level))
```

ClearDB のパラメータを以下の表にまとめます。

パラメータ	説明
config-dir	これは NeoServer 構成ファイルの格納先のパスです。 config-dir パラメータは必須です。
trace-file	データベースのクリア中にトレースファイルを出力したい場合は、その名前をここに指定します。例えば、次のようにします。 <code>./%cleardbtrace.log</code> trace-level を指定しない限り、 trace-file パラメータは必須ではありません。 このパラメータのデフォルトは、以下のとおりです。 <code>c:%neocore%neoxml%log%NeoXMLUtils.log</code> (Windows の場合) <code>/opt/NeoCore/neoxml/log/NeoXMLUtils.log</code> (Solaris の場合)
trace-level	トレースファイルを出力する場合、どのトレースレベルを使用するかを指定できます。トレースレベルを指定しないと、以下のデフォルト設定でトレースが行われます。 <code>FATAL:LOG_all;WARN:LOG_all;ERR:LOG_all</code>

オプション:

ClearDB コマンド実行時に、何を保存するのかを確認するためのプロンプトが表示されます。

- a. 自分で作成したものも含め、すべてのユーザー、グループ、AC ルールを保存するには、**Yes** と答えます。
- b. 新規データベースで作成されたデフォルトのユーザー、グループ、AC ルールを保存するには、**No** と答えます。

コマンドの結果の要約: このように、最終的にはデータの削除と、オプション (a) かオプション (b) のいずれかが行われます。

データベースをクリアするタイミング

データベースのクリアは、以下の条件のいずれかに該当するときに行います。

- 自分で作成したユーザー、グループ、AC ルール以外のすべての文書をデータベースから削除しなければならない場合。
- 定期的なバッチ処理やテスト手順をサポートするなどの理由でデータベースに一時データが含まれている場合。
- NeoServer を特別な目的で使用する場合。例えば、高速ワークフローエンジンや、他のデータベースのデータキャッシュなどが考えられます。

データベースのバックアップ

BackupDB の完全な構文は、以下のとおりです。

```
NeoXMLUtils BackupDB config-dir target-dir (trace-file (trace-level))
```

BackupDB のパラメータを以下の表にまとめます。

パラメータ	説明
config-dir	これは NeoServer 構成ファイルの格納先のパスです。 config-dir パラメータは必須です。
target-dir	これはバックアップファイルの格納先のパスです。 このディレクトリは存在しないか空である必要があります。 target-dir パラメータは必須です。
trace-file	データベースのバックアップ中にトレースファイルを出力したい場合は、その名前をここに指定します。例えば、次のようにします。 .\$backupdbtrace.log trace-level を指定しない限り、 trace-file パラメータは必須ではありません。 このパラメータのデフォルトは、以下のとおりです。 c:¥neocore¥neoxml¥log¥NeoXMLUtils.log (Windows の場合) /opt/NeoCore/neoxml/log/NeoXMLUtils.log (Solaris の場合)
trace-level	トレースファイルを出力する場合、どのトレースレベルを使用するかを指定できます。トレースレベルを指定しないと、以下のデフォルト設定でトレースが行われます。 FATAL:LOG_all;WARN:LOG_all;ERR:LOG_all

このコマンドはトランザクションログを適用してから、すべてのデータベースファイルを対象ディレクトリにバックアップします。

データベースのバックアップを行う理由

データベースのバックアップによって、以下のことが可能になります。

- 操作が失敗した場合に、データベースを既知の状態に戻せます。
- オペレーティングシステムなどの障害があった場合に、データベースを既知の状態に戻せます。
- インデックス再構築またはデータベース再構築の作業が失敗した場合に、データの消失を防げます。

注：インデックス再構築作業では、既存のデータベースのコピーがバックアップディレクトリに必ず作成されますが、**ClearDB** 操作の前にはデータベースを取り外し可能メディアにコピーするようにしてください。**ClearDB** はデータを永久的に削除してしまうからです。データベースファイルは非常に大きいので、古いバックアップを削除せずにたくさん格納しておくと、大量のディスクスペースを占有してしまいます。以前のバックアップを記録として取っておく必要がある場合は、別のディスクドライブか取り外し可能メディアにバックアップしてください。定期的にバックアップディレクトリをチェックし、その中のすべてのファイルが必要かどうか、削除したり別の場所に移動することが可能かどうかを判断してください。

データベースの復元

注：システムがクラッシュしたら、**RestoreDB** を実行してトランザクションログを適用します。

RestoreDB の完全な構文は、以下のとおりです。

```
NeoXMLUtils RestoreDB config-dir source-dir (trace-file (trace-level))
```

RestoreDB のパラメータを以下の表にまとめます。

パラメータ	説明
config-dir	これは NeoServer 構成ファイルの格納先のパスです。 config-dir パラメータは必須です。
source-dir	これはデータベースファイルのバックアップ先のソースディレクトリです。 source-dir パラメータは必須です。

trace-file	<p>データベースの復元中にトレースファイルを出力したい場合は、その名前をここに指定します。例えば、次のようにします。</p> <pre>.\$restoredbtrace.log</pre> <p>trace-level を指定しない限り、trace-file パラメータは必須ではありません。</p> <p>このパラメータのデフォルトは、以下のとおりです。</p> <pre>c:\$neocore\$neoxml\$log\$NeoXMLUtils.log</pre> <p>(Windows の場合)</p> <pre>/opt/NeoCore/neoxml/log/NeoXMLUtils.log</pre> <p>(Solaris の場合)</p>
trace-level	<p>トレースファイルを出力する場合、どのトレースレベルを使用するかを指定できます。トレースレベルを指定しないと、以下のデフォルト設定でトレースが行われます。</p> <pre>FATAL:LOG_all;WARN:LOG_all;ERR:LOG_all</pre>

トランザクションログを適用するかどうかを確認するためのプロンプトが表示されます。「Yes」と答えると、前回の BackupDB 以降のトランザクションすべてが復元後のデータベースファイルに適用されます。これにより、データベースは最新の状態になります。

このコマンドは、ソースディレクトリからデータベースファイルをそのあるべき場所にコピーします（その「あるべき場所」は、構成ファイルから読み込んで判断します）。トランザクションログを適用しなかった場合、トランザクションログを `\tx_log\backup` ディレクトリに退避し、`\tx_log` 以下から削除します。

データのエクスポート

旧バージョンの XMS では、個々の文書ファイルに対してエクスポートを行っていました。各文書には、XMS に格納されている文書の NeoCore MetaData と XML の内容が組み込まれます。この方法の場合は、XMS の内容をエクスポートするときに多数の文書が作成されるという問題が発生します。その結果、オペレーティングシステムのファイル管理機能に支障を来すこともありました。

エクスポートに以下の新機能が用意されました。

1. XMS の内容を巨大なマルチ文書にエクスポートできるようになりました。この文書の理論上のデフォルトサイズは 25 MB です。エクスポートでは、1つのエクスポートファイルが1つの文書の境界で終結します。NeoCore MetaData は、巨大なマルチ文書内の各サブ文書に組み込まれます。
2. エクスポート処理のステータスファイルが作成され、Export ディレクトリに格納されます。ステータスファイルの名前は、「ExportCfgStats.cst」です。このステータスファイルには、各マップファイルに基づくエクスポート処理の追跡情報が記録され、以下のデータが書き込まれます。
 - a) エクスポートされた文書の合計数
 - b) エクスポートされた合計バイト数
 - c) 文書の平均サイズ

3. エクスポート処理は、いったん停止してから再開することもできます（Windows では、<ESC> キーで停止できます）。エクスポートを再開するには、オプションで"-n"フラグを使用せずにエクスポートを行います。

警告: エクスポートを再開できるのは、NeoCore XMS が変更されていない場合に限られます。エクスポートをいったん停止してから、NeoServer を起動し、新しい文書の格納や削除など、データベースの内容に変更を加えた後にエクスポートを再開すると、エクスポートファイルに破損が生じる可能性があります。Export はこの状態をチェックしません。

4. エクスポート操作を制御するために、エクスポートの構成ファイルを作成することもできます。その構成ファイルは、「ExportCfgStats.cst」という名前にしなければならず、Export コマンドが参照する基本ディレクトリに配置しなければなりません。構成ファイルのサンプルを以下に示します。構成ファイルに追加できる主な項目は、以下のとおりです。
 - a) Target_Export_FileSize_MB。ターゲットのエクスポートファイルのサイズを設定します。有効なサイズは、1 から 1000 です。この設定によって、エクスポートファイルの理論上のサイズは N MB になります。このパラメータを指定しない場合のデフォルトのサイズは、25 MB です。
 - b) マップファイルごとの Export ディレクトリ。各 NeoCore XMS マップファイルを別々のディレクトリにエクスポートできます。各マップファイルのためのセクションを以下のように追加してください。

```
<Export_Cfg_Stats>
  <Target_Export_FileSize_MB>
    25
  </Target_Export_FileSize_MB>
  <MapFile_1>
    <XPT_Location>
      c:\%neocore%\export\%MapFile1_dir%
    </XPT_Location>
  </MapFile_1>
  <MapFile_2>
    <XPT_Location>
      c:\%neocore%\export\% MapFile2_dir%
    </XPT_Location>
  </MapFile_2>

  ...

</Export_Cfg_Stats>
```

Export の完全な構文は、以下の通りです。

```
NeoXMLUtils Export <config_dir> <export_dir> (<options>)
```

Export のパラメータを以下の表にまとめます。

パラメータ	説明
config-dir	これは NeoServer 構成ファイルの格納先のパスです。 config-dir パラメータは必須です。
export-dir	これはデータのエクスポート先のディレクトリです。 export-dir に neocore サブディレクトリが作成されます。 export-dir パラメータは必須です。
options	Export は新たなオプションをサポートします。 オプションの種類は以下の通りです。 -n 新しいエクスポートを強制的に行う -d 指定したエクスポートディレクトリにエクスポートファイル、XML ファイルが既に存在する場合は削除する -s ファイルサイズ:エクスポートファイルのサイズを MB で指定する -f エクスポート構成ファイル -l エクスポートログファイルのディレクトリ（設定されない場合はデフォルトのログディレクトリに出力されます）

このコマンドを実行すると、NeoCore XMS 内の全文書が指定の **export-dir** に書き出されます。さらに、すべての文書に **.XPT** というファイル拡張子が付きます。これにより、**.XPT** ファイルを逆にデータベースにインポートする際に、メタデータが付加されなくなります（メタデータは既に存在するからです）。

例:

```
1_102.xpt
2_103.xpt
2_104.xpt
3_105.xpt
```

注: FAT32 ドライブフォーマットを使用している場合、大量の文書をエクスポートすると、FAT32 ディスク特性の制限のためにエクスポートが失敗する可能性があります。

注: エクスポート操作では、エクスポート先ディレクトリの別個のサブディレクトリにすべてのユーザー文書、グループ文書、AC 文書が格納されます。これにより、ユーザーは後ほど対象データベースにこれらの文書をインポートするかどうかを選択できます。

データベースのエクスポートを行う理由

データベースのエクスポートによって、以下のことが可能になります。

- 他のシステムで使用するために、または別のデータベースへの入力のために、データベース全体をテキストとして出力できます。

バックアップ機能があるのに、なぜエクスポートが必要なのでしょう。エクスポートは、データベースのサイズ変更前に既存のデータと文書を保存しておいてから、元の（エクスポートした）データをインポートするようなときに適しています。

注: データファイルは大変大きい場合もあるので、古いエクスポートファイルを削除せずに何回もエクスポートを行うと、大量のディスクスペースを占有してしまう可能性があります。以前のエクスポートファイルを記録として取っておく必要がある場合は、別のディスクドライブか取り外し可能メディアにバックアップする必要があります。

データのインポート

インポートに以下の新機能が用意されました。

1. マルチ文書のエクスポートファイル（.xpt）をインポートします。
2. エクスポートの構成/ステータスファイル「ExportCfgStats.cst」が存在する場合にそのファイルを読み込み、その情報に基づいてエクスポートファイルの場所を特定します。
3. ステータスファイル「ImportCfgStats.cst」を作成し、インポートの状態とステータスの情報を記録します。これにより、インポートをいったん停止してから再開することが可能になります（Windows では、<ESC>キーで停止できます）。インポート処理を再開する場合は、同じ Import コマンドを使用してください。
4. インポートするファイル（filename）ごとに「filename.log」を作成します。このステータスファイルには、インポート対象ファイル内の次の XML 文書の開始点の追跡情報が記録されます。インポート処理が何かの理由で停止した場合でも、その開始点からそのファイルのインポートを続行できます。
5. インポート時に XMS の自動拡張を実行します。
6. -r オプション（Rebuild フラグ）を使用する場合は、各文書の DocID と CopyNumber が維持され、また NeoCore サブディレクトリ以下の管理者情報もインポートします。また、NeoCore-r オプションは CreateEmptyDB コマンドで新しい XMS データベースを作成した後に使用できません。-r オプションを使用するには、「ExportCfgStats.cst」が存在していなければなりません。
7. インポートによってすべてのファイルを完全にインポートできなかった場合インポートを再開することができます。ファイルまたは XMS の問題を解決してから、Import を再度使用してインポート処理を完了できます。

この Import コマンドは、以前の Import コマンドと同じように機能します。Import コマンドは、XMS データベースが作成されていて、管理情報が既にロードされていることを前提としています（この前提を満たすのは、基本的には CreateDB コマンドです）。指定のディレクトリにある.xpt または.xml のファイルがすべて XMS にインポートされます。インポートユーティリティは、<import_dir>にエクスポートの構成ファイル「ExportCfgStats.cst」が存在するかどうかをまずチェックします。そのファイルが存在する場合、インポートユーティリティは、その「ExportCfgStats.cst」の内容を読み込み、ディレクトリ情報と、インポートする必要のあるファイルと文書の数を確認します。

注： Import コマンドは、NEOCORE サブディレクトリにある NeoCore 管理情報をインポートしません。「ExportCfgStats.cst」が存在しない場合、インポートユーティリティは、<import_dir>を読み込み、インポートする必要のある.xpt と.xml のファイルの数と合計サイズを判別します。

インポート対象を判別できた時点で、インポート処理が始まります。つまり、<import_dir>に存在するファイルをロードし始めます。大きなマルチ文書は、小さな（約 100K の）トランザクション単位でロードします。各トランザクションをそれぞれ 1 つのマップファイルに格納し、トランザクションごとに新しいマップファイルを選択します。このようにして、XML を各マップファイルに均等に配分します。最初の段階では、約 10 MB の XML をロードしようとします。それからいったん停止し、XMS のステータスファイルをチェックして、各ファイルがどれほど満杯になっているのかを確認します。その最初のロードの情報と次の段階でロードする残りの XML の量の情報に基づき、各 XMS ファイルを必要に応じて拡張します。XMS のファイルを拡張してから、インポートを続行します。今回は、さらに大量の XML をロードしようとします。その大量の XML をロードした時点で、各 XMS ファイルを拡張する必要があるかどうかを再びチェックします。すべての XML データをロードするまで、必要なファイルを拡張し、インポートを続行するという処理をさらに続けていきます。

インポートでは、.xpt 文書内の DocID 番号を一定の数に基づいてシフトします。このように値を変更する必要があるのは、NeoCore 管理文書の数、Export によって.xpt ファイルが作成された時点とは異なっている可能性があるからです。

-r オプション（Rebuild フラグ）：

このコマンドは、基本的に Import と同じように機能します。ただし、CreateEmptyDB コマンドで NeoCore XMS を作成した状態を前提としている点が大きく異なります。CreateEmptyDB コマンドは、NeoCore 管理文書のない空のデータベースを作成します。NeoCore 管理情報は、直前のエクスポートで作成された文書からインポートされます。このコマンドは、XMS を新しいバージョンに移行するときに使用します。つまり、まず旧バージョンの XMS に対してエクスポートを実行し、次に NeoCore XMS を新しいバージョンにアップグレードし、CreateEmptyDB でデータベースを作成してから、Import_Rebuild を実行するという流れになります。NeoCore XMS が空でない場合、Import_Rebuild は失敗します。このコマンドの使用時には、DocID と CopyNo は変更されません。

インポートの再開

インポート処理は、中断コマンドやエラーの発生などにより、.xpt 文書と.xml 文書を全て完全にロードできなかった場合に再開することができます（再開する場合、-n オプションは使用しないでください）。たとえば、バッファのサイズが十分でないために一部のデータ項目を収容できず、そのために.xpt 文書と.xml 文書の一部または全部をインポートできなかった、というような状況が考えられます。そのようなときに、インポートは、ロードできるファイルだけをロードします。全てをロードしないでインポートが終了した場合は、ロードできなかった文書や、ロードできなかった文書内の特定部分のロードを再開できます。

Import の完全な構文は、以下のとおりです。

```
NeoXMLUtils Import <config-dir> <import-dir> (<options>)
```

Import のパラメータを以下の表にまとめます。

パラメータ	説明
config-dir	これは NeoServer 構成ファイルの格納先のパスです。 config-dir パラメータは必須です。
import-dir	これはデータのインポート元のディレクトリです。 import-dir は、 .XPT または .XML のファイル拡張子がついたファイルが保存されている場所です。 import-dir パラメータは必須です。
Options	Import は新たなオプションをサポートします。 オプションの種類は以下の通りです。 -n 新規 Import を強制的に行う -r Rebuild フラグ。 -k インポート中のトランザクションログを削除します。-r オプションと一緒に設定します。 -q 新しい DocID に変更し、CopyNumber をすべて 1 に設定します。 -f Import 構成ファイル -l ログファイルが生成されるディレクトリ（設定されない場合はデフォルトのログディレクトリに出力されます）

このコマンドを実行すると、**.XPT** ファイルと **.XML** ファイルが NeoCore XMS データベースに格納されます。**.XML** ファイルと **.XPT** ファイルは一緒に読み込むことができます。**.XML** ファイルのエンコードは UTF-8 フォーマットである必要があります。

注: **.XPT** ファイルは、このデータベースによって認識される特殊なフォーマットであり、ファイルの格納時に異なる処理が行われます。**.XML** ファイルの拡張子を **.XPT** に変更したり、**.XPT** ファイルの拡張子を **.XML** に変更したりすると、インポート時に文書が壊れてしまう可能性があります。**.XPT** ファイルはメタデータを既に持っていることが前提になっているので、手動による **.XPT** ファイルの作成は絶対にしないでください。データベースを壊してしまう可能性があります。逆に **.XML** ファイルはメタデータを持っていないことが前提になっています。**.XML** ファイルは、メタデータが付加されてからデータベースにインポートされます。

データベースのインポートを行う理由

データのインポートによって、以下のことが可能になります。

- 文書のあるデータベースから別のデータベースにコピーできます。
- 新しい **.XML** ファイルをバッチモードで格納できます。

データベースのインデックス再構築

ReIndex の完全な構文は、以下のとおりです。

```
NeoXMLUtils ReIndex <config-dir> <backup-dir> (<trace-file> (<trace-level>))
```

ReIndex のパラメータを以下の表にまとめます。

パラメータ	説明
config-dir	これは NeoServer 構成ファイルの格納先のパスです。 config-dir パラメータは必須です。
backup-dir	インデックスファイルのバックアップコピーを格納するディレクトリです。インデックス再構築操作が要求されると、インデックスファイルは自動的に MMDDYYYY_HHMMSS というフォルダ名のバックアップディレクトリにコピーされます。 backup-dir パラメータは必須です。
Options	Import は新たなオプションをサポートします。 オプションの種類は以下の通りです。 -n 新規 Import を強制的に行う -r Rebuild フラグ。 -k インポート中のトランザクションログを削除します。-r オプションと一緒に設定します。 -q 新しい DocID に変更し、CopyNumber をすべて 1 に設定します。 -f Import 構成ファイル -l ログファイルが生成されるディレクトリ（設定されない場合はデフォルトのログディレクトリに出力されます）

注：インデックス再構築作業を頻繁に行うと、バックアップディレクトリが大きくなり、ディスクスペースが足りなくなってしまう可能性があります。時々、バックアップディレクトリをチェックして、不要なファイルを削除するようにしてください。

インデックス再構築を行う理由

インデックス再構築は、以下の場合に行います。

- インデックスモードを最小モードから完全モードへ、または完全モードから最小モードに切り替えたとき。インデックスモードの構成については、「**NeoDatabase.xml ファイルの構成**」で説明します。
- インデックスサイズを増減したいとき。後述の「**NeoDatabase.xml ファイルの構成**」を参照してください。

このコマンドは、以下の2つの処理を行います。

- **NeoDatabase.xml** を編集してインデックスファイルのサイズを変更した後に、インデックスファイルを再作成します。
- ディクショナリファイルとマップファイルを処理して、インデックスファイルに値を設定します。

注: このプロセスで、インデックスファイルはバックアップディレクトリにコピーされます。

データベースの再構築コマンド Rebuild は 3.1 では廃止となりました。

データベース再構築の際には以下の手順で行います。

1. Export
2. CreateEmptyDB
3. Import (-r オプション必須)

データベースの再構築を行う理由

データベースの再構築を行うのは、以下の処理を行いたい場合です。

1. データベースの物理サイズを増減したいとき。後述の「**NeoXDBRuntime.xml ファイルの構成**」と「**NeoDatabase.xml ファイルの構成**」を参照してください。
2. 削除した XML データによって使用されていた記憶域を回復したいとき。

データベースファイル再配置コマンド

以下の NeoXMLUtils コマンドは、データベースファイルを1つのディレクトリから別のディレクトリに移動し、その新しい位置に合わせてデータベース管理構造を更新します。

MoveDataDictionary	データディクショナリファイルを移動する構文: NeoXMLUtils MoveDataDictionary <config-dir> <new-location> <<trace-file> <<trace-level>>
MoveTagDictionary	タグディクショナリファイルを移動する構文: NeoXMLUtils MoveTagDictionary <config-dir> <new-location> <<trace-file> <<trace-level>>
MoveTagIndex	タグインデックスファイルを移動する構文: NeoXMLUtils MoveTagIndex <config-dir> <new-location> <<trace-file> <<trace-level>>
MoveTagIndexDups	タグインデックス重複ファイルを移動する構文: NeoXMLUtils MoveTagIndexDups <config-dir> <new-location> <<trace-file> <<trace-level>>
MoveDataIndex	データインデックスファイルを移動する構文: NeoXMLUtils MoveDataIndex <config-dir> <new-location> <<trace-file> <<trace-level>>
MoveDataIndexDups	データインデックス重複ファイルを移動する構文: NeoXMLUtils MoveDataIndexDups <config-dir> <new-location> <<trace-file> <<trace-level>>
MoveTagPlusDataIndex	タグ+データインデックスファイルを移動する構文: NeoXMLUtils MoveTagPlusDataIndex <config-dir> <new-location> <<trace-file> <<trace-level>>
MoveTagPlusDataIndexDups	タグ+データインデックスファイルを移動する構文: NeoXMLUtils MoveTagPlusDataIndexDups <config-dir> <new-location> <<trace-file> <<trace-level>>

NeoXMLUtils コマンド**NeoXMLUtils コマンド**

処理内容	コマンド
データベースを構成ファイルから作成	NeoXMLUtils CreateDB <config-dir> (<trace-directory> (<trace-level>))
空のデータベースを構成ファイルから作成	NeoXMLUtils CreateEmptyDB <config-dir> (<trace-directory> (<trace-level>))
構成ファイルからデータベースをクリア	NeoXMLUtils ClearDB <config-dir> (<trace-directory> (<trace-level>))
構成ファイルで記述したデータベースを削除	NeoXMLUtils DeleteDB <config-dir> (<trace-directory> (<trace-level>))
データベースのエクスポート	NeoXMLUtils Export <config-dir> <export-dir> (<trace-directory> (<trace-level>))
データベースへインポート	NeoXMLUtils Import <config-dir> <import-dir> (<trace-directory> (<trace-level>))
インデックスファイルの再構築	NeoXMLUtils ReIndex <config-dir> <backup-dir> (<trace-directory> (<trace-level>))
データベースファイルのバックアップ	NeoXMLUtils BackupDB <config-dir> <target-dir> (<trace-directory> (<trace-level>))
バックアップしたデータベースファイルの復元	NeoXMLUtils RestoreDB <config-dir> <source-dir> (<trace-directory> (<trace-level>))
データベース管理ファイルをデコンパイルして、データベースプロパティファイルに反映	NeoXMLUtils Decompile <config_dir> <output_config_file> (<trace-directory> (<trace-level>))
マップファイルの移動	NeoXMLUtils MoveMapFile <config-dir> <map-file-no> <new-location> (<trace-directory> (<trace-level>))
データディクショナリファイルの移動	NeoXMLUtils MoveDataDictionary <config-dir> <new-location> (<trace-directory> (<trace-level>))
タグディクショナリファイルの移動	NeoXMLUtils MoveTagDictionary <config-dir> <new-location> (<trace-directory> (<trace-level>))
データディクショナリ相互参照ファイルの移動	NeoXMLUtils MoveCrossReference <config-dir> <new-location> (<trace-directory> (<trace-level>))

処理内容	コマンド
タグインデックスファイルの移動	NeoXMLUtils MoveTagIndex <config-dir> <new-location> (<trace-directory> <trace-level>))
タグインデックス MTMem ファイルの移動	NeoXMLUtils MoveTagIndexMTMem <config-dir> <new-location> (<trace-directory> <trace-level>))
タグインデックス DTMem ファイルの移動	NeoXMLUtils MoveTagIndexDTMem <config-dir> <dtmem-file-no> <new-location> (<trace-directory> <trace-level>))
データインデックスファイルの移動	NeoXMLUtils MoveDataIndex <config-dir> <new-location> (<trace-directory> <trace-level>))
データインデックス MTMem ファイルの移動	NeoXMLUtils MoveDataIndexMTMem <config-dir> <new-location> (<trace-directory> <trace-level>))
データインデックス DTMem ファイルの移動	NeoXMLUtils MoveDataIndexDTMem <config-dir> <dtmem-file-no> <new-location> (<trace-directory> (<trace-level>))
タグおよびデータインデックスファイルの移動	NeoXMLUtils MoveTagPlusDataIndex <config-dir> <new-location> (<trace-directory> <trace-level>))
タグおよびデータインデックス MTMem ファイルの移動	NeoXMLUtils MoveTagPlusDataIndexMTMem <config-dir> <new-location> (<trace-directory> <trace-level>))
タグおよびデータインデックス DTMem ファイルの移動	NeoXMLUtils MoveTagPlusDataIndexDTMem <config-dir> <dtmem-file-no> <new-location> <trace-directory> (<trace-level>))
管理ファイルの移動	NeoXMLUtils MoveAdminFile <config-dir> <new-location> (<trace-directory> <trace-level>))

ホットバックアップ

- **Host Backup** - これは、コマンドラインインターフェースまたは管理コンソール内からホットバックアップを開始する機能です。具体的には、ミラーボリュームのメンテナンス機能を活用し、1つ以上のミラーボリューム上に構成された、データ格納ディスクのコピーを作成します。このコピーはスナップショットと呼ばれ、データ格納ディスクのある時点の複製です。スナップショットを作成しておけば、任意の方法で格納されたデータをバックアップすることができます。ホットバックアップ実行時には、スナップショットを作成するためにミラーボリュームを分割する短い間、トランザクションは中断されます。ミラーボリュームの分割が完了すると、トランザクションは再開されます。また、バックアッププロセスは、通常取得されるトランザクションログとは別に、トランザクションログを取得します。

ホットバックアップは、以下のような場合に、格納されているデータの保護に役立ちます。

- ユーザーによる破損
- アプリケーションプログラム障害
- 情報管理システム障害
- オペレーティングシステム障害
- コンピュータシステムハードウェア障害
- ディスク制御障害
- ディスクドライブ障害

さらに、ホットバックアップには以下の特色があります。

- NeoServer の停止および再起動を必要とせず、バックアップを完了できます。
- ミラーボリュームからデータ格納ディスクを分割し、バックアップディスクスペースまたはストレージ装置にコピーできるスナップショットを作成します。
- ミラーボリュームとは別のディスク上に、トランザクションログのミラーを作成できます。この機能により、プライマリディスクに障害が発生した場合でも、ホットバックアッププロセスによって、完了したトランザクションの復元が可能になります。
- 代替ホットバックアップデバイスを「追加可能」モジュールとしてサポートしています。この機能により、サードパーティのフォールトトレラントソリューションを追加することが可能になります。
- ローカルホストおよびリモートホストから管理スクリプトを実行するための、コマンドラインインターフェースを提供しています。

管理コンソールの Manage Server ページに、ホットバックアップメニュー項目が追加されています。

Veritas を使用した Solaris ボリューム管理

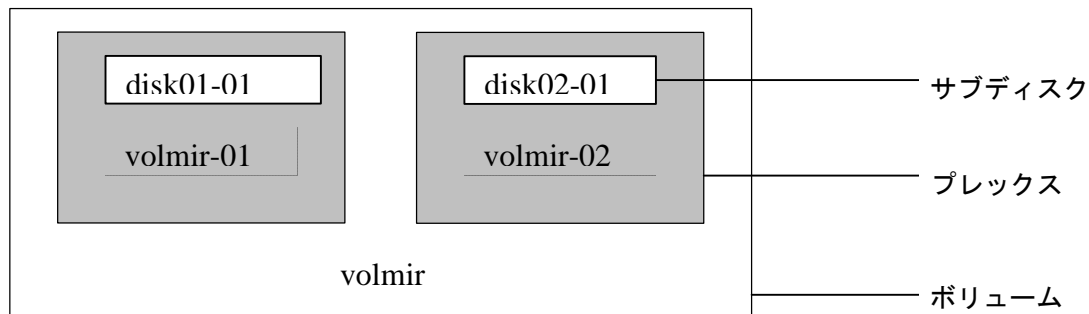
Solaris プラットフォーム上の XMS でホットバックアップ用のディスクボリュームを管理するための推奨ユーティリティは、Veritas Volume Manager 3.2 以降です。最大限の保護のために、XMS データのミラーリングとトランザクションログミラーリングを組み合わせてデータの冗長性を確保できます。

このセクションでは、ホットバックアップを実行してデータを確実に復元するために XMS が使用するボリュームをミラーリングする手順を説明します。

1. Veritas Volume Manager をインストールします。
2. XMS インストール先ディスクをミラーボリュームとするか、既にインストールされている XMS のデータ格納先をミラーボリュームとしてデータベースを作成します。
3. ホットバックアップ中ミラーボリュームは分割され、データベースファイルをミラードライブの外にコピーできるようにするため、ミラーボリュームが再構成されます。トランザクションログは、ホットバックアップ中 2 か所に取得されます。

Veritas ソリューションの概要

Veritas のミラーボリュームを以下の図に示します。ミラーボリュームには 2 つのプレックスを作成します。



ホットバックアップ構成

ホットバックアップの構成作業は、NeoXDBRuntime.xml 構成ファイルに定義される以下の構成要素等によって行います。

- XMS がホットバックアップ中に重複してトランザクションログを書き出す必要があるかどうかを示す要素
- 2 次トランザクションログの格納場所を示す要素
- バックアップする XMS データを格納するミラーボリュームの名前と、ミラーを構成するプレックスの名前を示す要素

操作

ホットバックアップの実行時に XMS データをミラーボリュームのスナップショットからコピーできるようにするため、XMS データベースファイルおよびトランザクションログはミラーボリュームに格納します。ホットバックアップ中はミラーリングが停止しますが、どの時点で障害が起きてもそこまでの回復が確実にできるようにするため、トランザクションログは 2 か所書き出されます。スナップショットからデータがバックアップされた時点で、そのミラーボリュームは再接続されます。

XMS インストール前の初期ミラーの構築

望ましいのは、ミラーボリュームの作成を行ってから XMS をインストールするという手順です。ディスクを（ボリュームの作成によって）Veritas 制御下に置いた後に、どのサブディスクを使用して、ミラーを構成するプレックスを作成するかを計画できます。2 台の異なるコントローラのディスクを使用して、可能な限り冗長性の高い構成にすることを推奨します。

ミラーボリュームを構築するには、`vxmake` コマンドまたは `vxassist` コマンドを使用し、次の手順に従う必要があります。まず、`vxmake` コマンドを使用してサブディスクを作成します。次に、`vxmake` コマンドを使用してプレックスとサブディスクを関連付けます。最後に、`vxmake` コマンドを使用してミラーボリュームを作成し、少なくとも 2 つのプレックスを関連付けます。その後、`vxvol start` コマンドを使用してボリュームを開始します。

`vxassist` コマンドを使用してミラーを構築することもできます。このコマンドは必要なプレックスの作成も行います。例:

```
vxassist make volume_name size layout=mirror
```

ミラーボリュームを作成した後、そのミラーボリュームにマウントするファイルシステムを作成します。そのためには、UNIX コマンド `newfs` と `mount` を使用します。このボリュームへのパスは、`/dev/vx/dsk/volume_name` のようになります。

ボリューム管理ソフトウェアのインストールと構成を行う前に、Veritas のユーザーガイドを参照してください。

XMS インストール後の初期ミラーの構築

この手順は推奨できません。ミラーリングを実行したいシステムで、既に XMS が稼働している場合は、XMS データをバックアップしてから、「XMS インストール前の初期ミラーの構築」にある指示に従うことをお勧めします。

XMS が既にインストールされている場合、まず、その XMS インストール先またはデータ格納先のディスクを Veritas Volume Manager の制御下に置く必要があります。Veritas の制御下に置くディスクにはカプセル化が必要です。カプセル化を行わない場合、そのディスクは初期化され、データはすべて破壊されてしまいます。

Veritas の資料にあるとおり、カプセル化が失敗する可能性を減らすために、ディスクに関して以下のことを確認してください。

- ディスクの先頭または最後に、どのパーティションにも属さない空き領域が 1MB 以上存在すること
- ディスクに 2 つの空きパーティションが存在すること
- ディスク全体を表す s2 スライスが存在すること

Vxdiskadm コマンドの実行後に、オペレーティングシステムのリブートが必要です。

上記の条件が満たされている場合は、vxdiskadm コマンドを使用して、『Administration Guide』の指示に従ってディスクをカプセル化できます。

ディスクを Veritas 制御下に置いたら、ミラーボリュームを構成するサブディスクを作成できます。別々のコントローラのサブディスクを使用して、可能な限り冗長性の高いディスク構成にすることを推奨します。ミラーボリュームを構築するには、vxmake コマンドまたは vxassist コマンドを使用し、次の手順に従う必要があります。まず、vxmake コマンドを使用してサブディスクを作成します。次に、vxmake コマンドを使用してプレックスとサブディスクを関連付けます。最後に、vxmake コマンドを使用して XMS のインストール先のボリュームに新しいプレックスを追加することにより、ミラーボリュームを作成します。

ホットバックアップ用のミラーの中断と再構築

ミラーボリューム上で XMS が稼働するようになれば、ホットバックアップを実行できます。バックアップは、ミラーリングを停止して、ミラーボリュームのプレックスの 1 つからデータを取得することによって行います。具体的には、まずミラープレックスが、XMS データを持つミラーボリュームから切り離されます。このプレックスに基づいて一時ボリュームが作成され、ファイルシステムとしてマウントされます。バックアップはそこから実行されます。その後、新しいファイルシステムがアンマウントされ、一時ボリュームが停止し、そのプレックスが一時ボリュームから切り離されます。さらに、そのプレックスがミラーボリュームと関連付けられ、ミラーリングが再開します。最後に、一時ボリュームが削除されます。

以下に、一連のコマンドと手順を示します。ここで取り上げるコマンドでは、volmir がミラーボリュームの名前、volmir-02 が volmir の（2 番目の）ミラープレックスの名前、xms_tempvol が一時ボリュームの名前、/backup/xms が一時ボリュームがマウントされ、バックアップの対象となるファイルシステムのマウントポイントの名前です。

1. vxplex dis volmir-02
2. vxmake -U gen vol xms_tempvol plex=volmir-02
3. vxvol start xms_tempvol
4. fsck -y /dev/vx/rdisk/xms_tempvol
5. mount /dev/vx/dsk/tempvol /backup/xms
6. /backup/xms のバックアップを実行します。
7. umount /backup/xms
8. vxvol stop xms_tempvol
9. vxplex dis volmir-02
10. vxplex att volmir volmir-02
11. vxedit rm xms_tempvol

Veritas のための neoadmin 管理者権限の付与

UNIX マシンの root 管理者は、Veritas Volume Manager コマンドを実行する権限をユーザー「neoadmin」に与える必要があります。役割ベースのアクセス制御（RBAC）機能を構成して、Veritas コマンドを実行するための権限を neoadmin ユーザーに付与できます。

スーパーユーザーとなり、以下を実行します。

- 役割の実行属性を追加します。詳細については、exec_attr（4）を参照してください。この作業では、/etc/security/exec_attr に以下の行を追加します。

```
Veritas:suser:cmd:::/usr/sbin/vxplex:uid=0
```

```
Veritas:suser:cmd:::/usr/sbin/vxmake:uid=0
```

```
Veritas:suser:cmd:::/usr/sbin/vxvol:uid=0
```

```
Veritas:suser:cmd:::/usr/sbin/vxedit:uid=0
```

```
Veritas:suser:cmd:::/usr/sbin/fsck:uid=0
```

```
Veritas:suser:cmd:::/usr/sbin/mount:uid=0
```

```
Veritas:suser:cmd:::/usr/sbin/umount:uid=0
```

- 役割の名前を追加します。詳細については、prof_attr（4）を参照してください。この作業では、/etc/security/prof_attr ファイルに以下の行を追加します。

```
Veritas:::Allow neoadmin to use Veritas commands for hot backup:
```

- ユーザーに役割を割り当てます。詳細については、user_attr（4）を参照してください。この作業では、/etc/user_attr ファイルに以下の行を追加します。

```
neoadmin::::profiles=Veritas
```

- nscd デーモンを再起動します。

```
/etc/init.d/nscd stop
```

```
/etc/init.d/nscd start
```

- Veritas コマンドのパスを neoadmin ユーザーの PATH 環境変数に設定します。例えば、Veritas がデフォルトの位置である /usr/sbin にインストールされている場合は、neoadmin ユーザーの定義ファイル.profile を更新して、/usr/sbin ディレクトリを PATH 環境数に設定します。

Neoadmin ユーザーは、Veritas コマンドを /usr/bin/pfksh シェルから実行する権限を持つようになります。XMS に用意されているスクリプトを使用する場合は、実行シェルが指定されているので、neoadmin ユーザーのデフォルトシェルを変更する必要はありません。Neoadmin ユーザーのデフォルトシェルを変更すると、neoadmin による XMS の管理作業が正常に行えなくなる場合がありますので、ご注意ください。

ユーザーによる手動バックアップ

<BackupDestination>が指定されていない場合、XMS はミラーボリュームのスナップショットの作成だけを行います。この時点で、ユーザーは手動で、スナップショットを一時マウントポイントにマウントし、バックアップ装置へのバックアップを行い、一時マウントポイントのアンマウントを実行する必要があります。

一時領域のマウントと一時ボリュームのクリーンアップのためのスクリプトが XMS のインストール先の `neoxml/bin` ディレクトリに用意されています。また、ユーザーが手動で Veritas コマンドを実行することもできます。

スクリプト:`verReleaseSnapshot.sh`

使用法:`verMountSnapshot.sh mirrored_plex temp_volname temp_mount_dir`

説明:一時マウントポイントをアンマウントし、一時ボリュームを停止し、ミラープレックスを一時ボリュームから切断し、一時ボリュームを削除します。

ロギング:すべての出力メッセージは/tmp/neoadmin.hotbackup.log に書き出されます。

実行される Veritas コマンドと UNIX コマンド:

```
umount temp_mount_dir

vxvol stop temp_volname

vxvol dis mirrored_plex

vxedit rm temp_volname
```

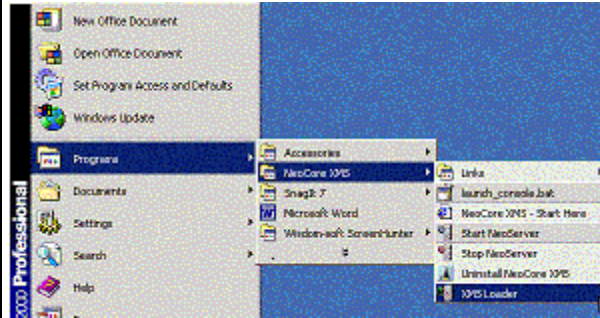
管理コンソールのホットバックアップコマンド

機能	コマンド
このコマンドで、ホットバックアッププロセスを開始します。	Start
このコマンドで、NeoServer が認識している限りのホットバックアップ手順の現在の状態を取得します。例えば、手動バックアップの場合は、バックアップが保留中であることを示す状態だけが表示されます。	Status
このコマンドで、ホットバックアップ手順を終了します。管理者がミラーボリュームの手動バックアップを行う場合は、このコマンドで HotBackupManager にバックアップが完了したことを通知できます。	Complete

XMS Loader

XMS Loader は、稼働中の XMS サーバーに対して、**任意のエンコーディングで任意の量の XML** をロードするためのウィザードです。これは、データベースをロードするための XMS インポート機能の代わりに用意されたツールであり、NeoServer ではなくクライアントとして稼働します。XMS Loader の非常に重要な機能は、**再試行**です。つまり、ロードが失敗したときでも再試行が可能です。

XMS Loader を開始するには、**[Programs]**、**[NeoCore XMS]**、**[XMS Loader]** を順にクリックします。



最初の **[Load XMS Wizard]** 画面が表示されます。

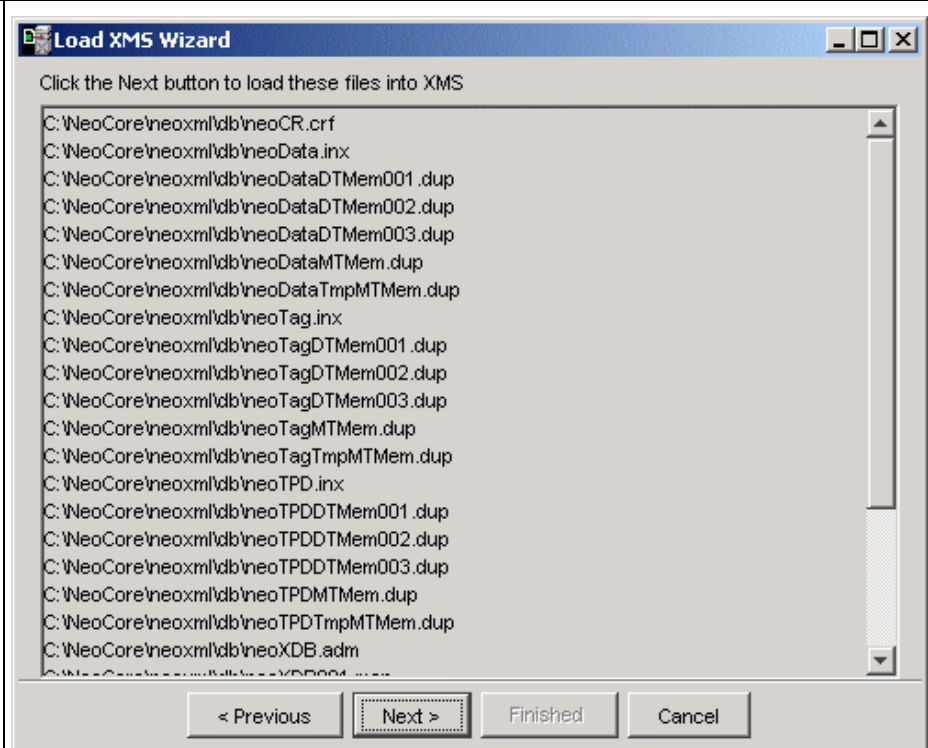
[Machine Name]
(通常は localhost)、
[Machine Port]
(通常は 7700)、
[User Name]、
[Password]、**[Locale]**、
[Character Set] を入力します。接続情報を保存するボックスをクリックしてから、ロード対象のファイルを配置するディレクトリを選択します。

フィールドを完成させたら、**[Next>]** をクリックします。

 A screenshot of the 'Load XMS Wizard' dialog box. The title bar says 'Load XMS Wizard'. The main area is titled 'Enter the connection information for NeoCore XMS'. It contains several input fields: 'Machine Name' (localhost), 'Machine Port' (7700), 'User Name' (Administrator), 'Password' (masked with asterisks), 'Locale' (en), and 'Character Set' (UTF-8). Below these fields is a checkbox labeled 'Click here to save connection information'. At the bottom, there is a section titled 'Choose the directory that contains the files to be loaded' with a text box containing 'C:\NeoCore\neoxml\ddb' and a folder selection icon. At the very bottom are four buttons: '< Previous', 'Next >', 'Finished', and 'Cancel'.

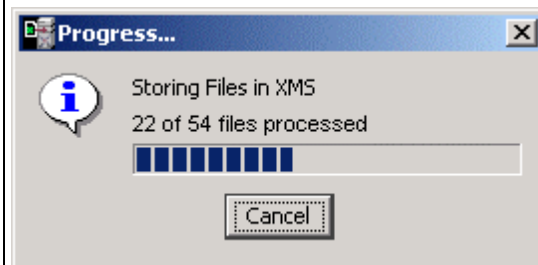
2 番目の **[Load XMS Wizard]** 画面が表示されます。前の画面で選択したディレクトリ内に存在するファイルが表示されます。**[Next>]** ボタンをクリックしてこれらのファイルを XMS にロードします。

最初の **[Load XMS Wizard]** 画面に戻るには、**[<Previous]** をクリックします。ロード処理をキャンセルするには、**[Cancel]** をクリックします。



進捗状況ダイアログに、ロードされたファイルの数が表示されます。

ロード処理を停止するには、**[Cancel]** をクリックします。



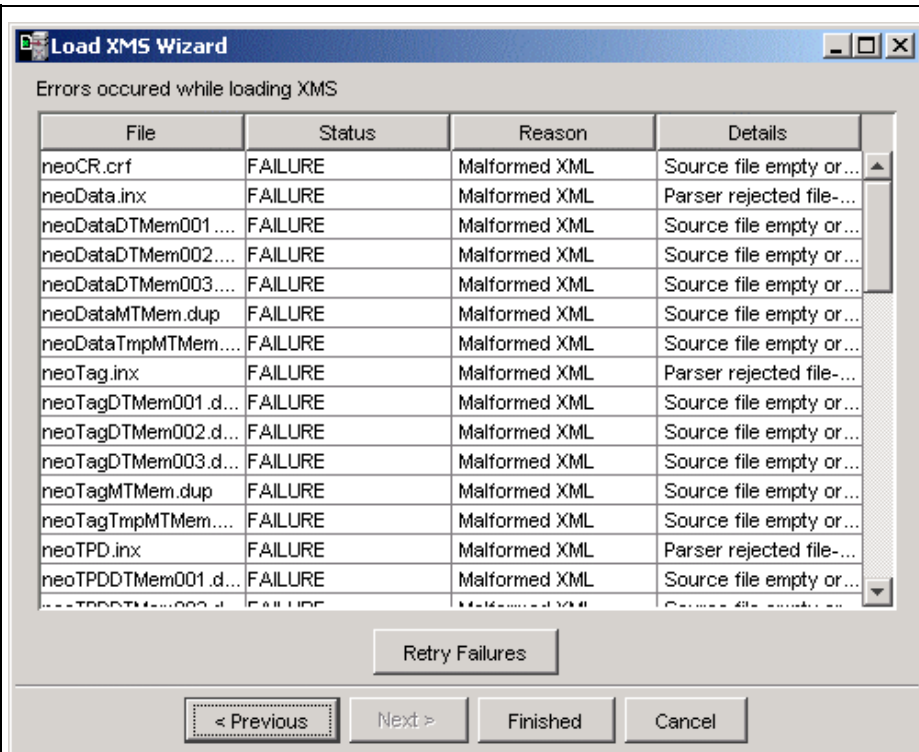
ロード処理が完了すると、3番目の **[Load XMS Wizard]** 画面が表示されます。この画面には、ロード処理の結果が表示されます。

ロード処理を再試行するには、**[Retry Failures]** をクリックします。

ロード処理をキャンセルするには、**[Cancel]** をクリックします。

前の **[Load XMS Wizard]** 画面に戻るには、**[<Previous]** をクリックします。

完了したら、**[Finished]** をクリックします。



手動（オフライン）ファイル拡張

NeoDatabase.xml ファイルには、最初にデータベースを作成するときのためのパラメータが用意されており、ファイルのサイズもその1つです。ファイルサイズは固定されており、しきい値に達すると、データベースにそれ以上文書を格納できなくなります。

そこで、ファイルを手動で拡張するためのコマンドが NeoXMLUtils に用意されており、以下のファイルタイプに対応しています。

- データディクショナリ
- クロスリファレンス
- タグディクショナリ
- マップファイル（すべてが同じサイズに拡張される）
- コアインデックス
- DTMem
- MTMem

ファイル拡張コマンド

機能	コマンド
すべてのマップファイルのサイズを大きくします。	NeoXMLUtils GrowMapFile <config-dir> <MB to grow> (<trace-directory> (<trace-level>))
データディクショナリのサイズを大きくします。	NeoXMLUtils GrowDataDictionary <config-dir> <MB to grow> (<trace-directory> (<trace-level>))
タグディクショナリのサイズを大きくします。	NeoXMLUtils GrowTagDictionary <config-dir> <MB to grow> (<trace-directory> (<trace-level>))
データディクショナリのクロスリファレンスファイルのサイズを大きくします。	NeoXMLUtils GrowCrossReference <config-dir> <MB to grow> (<trace-directory> (<trace-level>))
すべてのタグインデックス DTMem ファイルのサイズを大きくします。	NeoXMLUtils GrowTagIndexDTMem <config-dir> <MB to grow> (<trace-directory> (<trace-level>))
すべてのデータインデックス DTMem ファイルのサイズを大きくします。	NeoXMLUtils GrowDataIndexDTMem <config-dir> <MB to grow> (<trace-directory> (<trace-level>))
すべてのタグ+データインデックス DTMem ファイルのサイズを大きくします。	NeoXMLUtils GrowTagPlusDataIndexDTMem <config-dir> <MB to grow> (<trace-directory> (<trace-level>))
タグインデックス MTMem のサイズを大きくします。	NeoXMLUtils GrowTagIndexMTMem <config-dir> <MB to grow> (<trace-directory> (<trace-level>))
データインデックス MTMem のサイズを大きくします。	NeoXMLUtils GrowDataIndexMTMem <config-dir> <MB to grow> (<trace-directory> (<trace-level>))

タグ+データインデックスMTMemのサイズを大きくします。	NeoXMLUtils GrowTagPlusDataIndexMTMem <config-dir> <MB to grow> (<trace-directory> (<trace-level>)) <MB to grow>には、ファイルサイズをどれほど大きくするのかを MB 単位で指定します。
タグインデックスTmpMTMem のサイズを大きくします。	NeoXMLUtils GrowTagIndexTmpMTMem <config-dir> <amt to grow> (<trace-directory> (<trace-level>))
データインデックスTmpMTMem のサイズを大きくします。	NeoXMLUtils GrowDataIndexTmpMTMem <config-dir> <amt to grow> (<trace-directory> (<trace-level>))
タグ+データインデックスTmpMTMemのサイズを大きくします。	NeoXMLUtils GrowTagPlusDataIndexTmpMTMem <config-dir> <amt to grow> (<trace-directory> (<trace-level>)) <amt to grow>には、TmpMTMem ファイルをどれほど拡張するのかを 194K の倍数で指定します。
タグインデックスコアのサイズを大きくします。	NeoXMLUtils GrowTagIndex <config-dir> <quanta-increment> (<trace-directory> (<trace-level>))
データインデックスコアのサイズを大きくします。	NeoXMLUtils GrowDataIndex <config-dir> <quanta-increment> (<trace-directory> (<trace-level>))
タグ+データインデックスコアのサイズを大きくします。	NeoXMLUtils GrowTagPlusDataIndex <config-dir> <quanta-increment> (<trace-directory> (<trace-level>)) <quanta-increment>には、コアインデックスに追加するカンタム数を 2 の乗数で指定します。

領域自動拡張

XMS には、XML 情報を格納するファイルがしきい値 (AlarmLevel) に達したときに、そのファイルを拡張するための機能が用意されています。データベースの作成時には、NeoDatabase.xml 構成ファイルに指定されているサイズに基づいてデータベース構造のための領域が静的に割り振られます。時間の経過とともにデータベース構造がしきい値に達すると、XMS は、新しい情報を収容するためにファイルを自動的に拡張します。

XMS ファイルのしきい値レベルと拡張幅は、NeoXDBRuntime.xml ファイルで設定します。領域自動拡張パラメータの詳細については、「**NeoXDBRuntime.xml ファイルのパラメータの定義**」を参照してください。

Solaris と Windows の領域自動拡張の比較

Solaris と Windows では、領域自動拡張の構成作業を基本的に同じ方法で行いますが、重大な違いが 1 つあります。Windows では、XMS ファイルの Windowing を設定する必要があります。自動拡張の対象になるのは、Windowing されたファイルだけです。XMS での Windowing は、NeoXDBRuntime.xml 構成ファイルの Windowing タグに基づいて実行されます。

64 ビット Solaris オペレーティングシステムの場合、XMS ファイルの Windowing は必要ありません。XMS でサポートされているどんなサイズのファイルでも、64 ビットアーキテクチャによるアドレッシングが可能だからです。ところが、32 ビットの Windows オペレーティングシステムの場合は違います。メモリー内で 32 ビットアーキテクチャによるアドレッシングが不可能なサイズにまでファイルが拡張する可能性があります。そのため、XMS ファイルの Windowing が必要になります。また、「Windowing のスライド処理」という言い方もあります。

モニタリング

このセクションでは、以下のトピックについて説明します。

- NeoServer 統計の表示と解説
- 情報取得のためのログファイルの使用

サーバー統計

NeoServer 統計情報とは、NeoServer とそのデータベースの状態のスナップショットを提供するための一群の測定値であり、構造化 XML 文書の形で用意されます。測定値には、以下の 5 つの主要なカテゴリがあります。

- **<Storage-Stats>** – データリポジトリの状態を記述した格納統計です。
- **<Access-Stats>** – 対応済みのユーザー要求の数を示したアクセス統計です。
- **<Admin-Stats>** – 特定の内部状態の変化を記述した管理統計です。
- **<Buffer-Stats>** – 合計バッファ数とバッファの可用性を記述したバッファ統計です。
- **<Transaction-Stats>** – 未解決のトランザクション、使用可能なロック、現在のロック要求と保留のロック要求、一般的なトランザクション統計を示したトランザクション統計です。

NeoServer の状態は絶えず変化するため、測定値は表示するたびに少しずつ変わっていきます。ただし、格納単位の合計物理サイズを示す統計は変わりません。

NeoServer 統計情報の表示

NeoServer 統計を表示するには、以下のようになります。

1. コンソールに移動します。
2. [Server Status] タブを開きます。

3. [View Status] を選択します。サーバー統計が XML フォーマットで表示されます。

<Storage-Stats>について

NeoCore XMS には、データディクショナリ、タグディクショナリ、マップファイル、データインデックス、タグインデックスなどの名前が付いた、独立した永続格納単位がいくつかあります。

格納統計には、各格納単位の事前定義の物理サイズ（バイト単位）、各格納単位の使用率、現在使用中のバイト数、エントリの数が表示されます（「エントリ」の定義は NeoCoreXMS 独自のものです）。

現在のサイズは常に物理サイズよりも小さくなり、この差が、新しいエントリに使用できる領域となります。その差が小さい場合、格納単位は物理領域を使い切ってしまうおそれがあります。格納測定値を確認すれば、格納条件の低下を検出してすぐにアクションを実行できるので、このことには大きな価値があります。さらに、格納統計からは、XML 文書の数とリポジトリ内に格納されているバイト数も判別できます。

それぞれの**格納統計タグ**を以下の表にまとめます。その中でも特に注目する必要があるのは、データベースのデータストアの使用率（%）です。

タグ	説明	例
<Documents>	リポジトリ内に格納されている文書の数。	<pre><Storage-Stats> <Documents>453</Documents></pre>
<Bytes-Stored>	リポジトリ内に格納されている合計バイト数。	<pre><Bytes-Stored>5168759</Bytes-Stored></pre>
<Data-Dictionary-Size>	<ul style="list-style-type: none"> Percent Full: 使用率 Physical: 物理サイズ Current: 現在サイズ Entries: エントリ数 リポジトリ内のすべてのデータを含んだファイルについての情報。	<pre><Data-Dictionary-Size> <Percent-Full>20.00</Percent-Full> <Physical>10485760</Physical> <Current>2166797</Current> <Entries>23530</Entries> </Data-Dictionary-Size></pre>

<Tag-Dictionary-Size>	<ul style="list-style-type: none"> ▪ Percent Full: 使用率 ▪ Physical: 物理サイズ ▪ Current: 現在サイズ ▪ Entries: エントリ数 <p>リポジトリ内のすべての XML タグを含んだファイルについての情報。</p>	<pre> <Tag-Dictionary-Size> <Percent-Full>0.00</Percent-Full> <Physical>1048576</Physical> <Current>5790</Current> <Entries>239</Entries> </Tag-Dictionary-Size> . . . </pre>
<Map-Files> <ul style="list-style-type: none"> ▪ Number Of Map Files: マップファイル数 ▪ Max Entries Per Map File: マップファイルあたりの最大エントリ数 ▪ Map High Water Mark: マップ上限値（マップファイルがこれ以上格納を受け付けない限界値であり、デフォルトでは約 90%） ▪ Map XXX Size: マップファイル XXX サイズ ▪ Documents: 文書数 ▪ Inserted Jump Entries: 挿入済みのジャンプエントリ数 	<ul style="list-style-type: none"> ▪ Percent Full: 使用率 ▪ Physical: 物理サイズ ▪ Current: 現在サイズ ▪ Entries: エントリ数 <p>リポジトリ内で使用可能なマップファイルの数と、それぞれの使用可能なサイズと領域。</p>	<pre> <Number-Of-Map-Files>4</Number-Of-Map-Files> <Max-Entries-Per-Map-File>393216</Max-Entries-Per-Map-File> <Map-High-Water-Mark>353893</Map-High-Water-Mark> <Map-1-Size> <Percent-Full> 4.54</Percent-Full> <Physical>9437184</Physical> <Current>428376</Current> <Entries>17849</Entries> <Documents>147</Documents> <Inserted-Jump-Entries>5</Inserted-Jump-Entries> </Map-1-Size> . . . </Map-Files> . . . </pre>

<p><Index-Data-Size></p> <ul style="list-style-type: none"> ▪ Allocated Quanta: 割り振り済みのカンタム ▪ Core Index: コアインデックス ▪ Total Quanta: 合計カンタム ▪ Allocated: 割り振り済み ▪ Duplicate Trees: 重複ツリー ▪ Total Quanta: 合計カンタム ▪ Allocated: 割り振り済み 	<ul style="list-style-type: none"> ▪ Percent Full: 使用率 <p>データインデックスを含んだファイルについての情報。</p>	<pre> <Index-Data-Size> <Allocated-Quanta>23530</Allocated-Quanta> <Core-Index> <Total-Quanta>262144</Total-Quanta> <Allocated>23530</Allocated> <Percent-Full>8.98</Percent-Full> </Core-Index> <Duplicate-Trees> <Total-Quanta>1008</Total-Quanta> <Allocated>0</Allocated> <Percent-Full>0.00</Percent-Full> . . . </Duplicate-Trees> </Index-Data-Size> . . . </pre>
---	--	--

<p><Index-Tag-Size></p> <ul style="list-style-type: none"> ▪ Allocated Quanta: 割り振り済みのカンタム ▪ Core Index: コアインデックス ▪ Total Quanta: 合計カンタム ▪ Allocated: 割り振り済み ▪ Duplicate Trees: 重複ツリー ▪ Total Quanta: 合計カンタム ▪ Allocated: 割り振り済み 	<ul style="list-style-type: none"> ▪ Percent Full: 使用率 <p>XML タグインデックスを含んだファイルについての情報。</p>	<pre> <Index-Tag-Size> <Allocated-Quanta>45556</Allocated-Quanta> <Core-Index> <Total-Quanta>131072</Total-Quanta> <Allocated>239</Allocated> <Percent-Full>0.18</Percent-Full> </Core-Index> <Duplicate-Trees> <Total-Quanta>2097136</Total-Quanta> <Allocated>126800</Allocated> <Percent-Full>6.05</Percent-Full> . . . </Duplicate-Trees> </Index-Tag-Size> . . . </pre>
--	---	--

<Index-TagPlusData-Size> <ul style="list-style-type: none"> Allocated Quanta: 割り振り済みのカンタム Core Index: コアインデックス Total Quanta: 合計カンタム Allocated: 割り振り済み Duplicate Trees: 重複ツリー Total Quanta: 合計カンタム Allocated: 割り振り済み 	<ul style="list-style-type: none"> Percent Full: 使用率 タグ+データインデックスを含んだファイルについての情報。	<pre> <Allocated-Quanta>69582</Allocated-Quanta> <Core-Index> <Total-Quanta>131072</Total-Quanta> <Allocated>26328</Allocated> <Percent-Full>20.09</Percent-Full> </Core-Index> <Duplicate-Trees> <Total-Quanta>1048560</Total-Quanta> <Allocated>215072</Allocated> <Percent-Full>20.51</Percent-Full> . . . </Duplicate-Trees> </Index-TagPlusData-Size> </Storage-Stats> </pre>
---	--	--

NeoCore XMS サーバーは、クエリー、カウント、変更、削除など、さまざまなデータアクセス操作をサポートしています。アクセス統計には、データストアの開始またはデータベースのクリア以降のそれぞれの操作の実行回数が表示されます。

それぞれのアクセス統計タグを以下の表にまとめます。

タグ	説明	例
<Data-Queries>	実行したクエリーの数。この表の下にある注を参照してください。	<pre> <Access-Stats> <Data-Queries>182528</Data-Queries> </pre>

<Count-Queries>	実行した count クエリーの数。一般に、 count クエリーは、ターゲットセット内の要素そのものではなく、要素の数を返します。この表の下にある注を参照してください。	<pre> <Count-Queries>223</Count-Queries> <Stat-Requests>12</Stat-Requests> <Deletes>342</Deletes> <Inserts>1241</Inserts> <Modifies>56</Modifies> <Copies>2</Copies> </Access-Stats> </pre>
<Stat-Requests>	統計を表示した回数。	
<Deletes>	削除コマンドを呼び出した回数。	
<Inserts>	挿入コマンドを呼び出した回数。	
<Modifies>	変更コマンドを呼び出した回数。	
<Copies>	文書をコピーした回数。	

注：<Data-Queries>と<Count-Queries>の統計には、アクセス制御から生成された内部アクションのカウン트가含まれている可能性があります。

<Admin-Stats>について

管理統計には、NeoServer 内の管理機能についての測定値が示されます。

注：ほかにも当社独自の意味を持った管理統計がいくつかありますが、NeoServer の状態の全体像を把握するにはその種の統計も記録しておくことが重要です。

それぞれの管理統計タグを以下の表にまとめます。

タグ	説明	例
<Inserted-Jump-Entries>	挿入済みのジャンプエントリ数。	<code><Inserted-Jump-Entries>21</Inserted-Jump-Entries></code>
<Index-Rebuilds>	インデックスを再構築した回数。	<code><Index-Rebuilds>0</Index-Rebuilds></code>
<Store-Rebuilds>	データベースを再構築した回数。	<code><Store-Rebuilds>0</Store-Rebuilds></code>
<Next-DocID>	リポジトリ内に次に格納する文書に割り当てられる番号。	<code><Next-DocID>455</Next-DocID></code>

<Buffer-Stats>について

バッファ統計には、NeoServer 内で使用されているメモリープールとディスク上に設けられた退避用のバッファプールの両方の測定値が示されます。したがって、ユーザーは共有プールから使用されている短期のバッファ領域を確認できます。

それぞれのバッファ統計タグを以下の表にまとめます。

タグ	説明	例
<Memory-Pool>	メモリープールの合計サイズ、合計バッファ数、使用中のバッファ数、空きバッファ数。	<pre> <Buffer-Stats> <Memory-Pool> <Total-Pool-Size>74972160</Total-Pool-Size> <Number-Of-Buffers>4620</Number-Of-Buffers> <Buffers-In-Use>7</Buffers-In-Use> <Free-Buffers>4613</Free-Buffers> <Low-Buffer-Count>4609</Low-Buffer-Count> <Requests-For-Buffer>186</Requests-For-Buffer> <Number-Of-Buffer-Pools>10</Number-Of-Buffer-Pools> . . . </pre>
<Disk-Backed-Pool>	ディスク上に設けられたバックアップ用のプールの合計サイズ、合計バッファ数、使用中のバッファ数、空きバッファツリー数。	

<Transaction-Stats>について

トランザクション統計には、現在のトランザクションと完了したトランザクションについての測定値が示されます。したがって、ユーザーはサーバートラフィックの評価や、データの変更やクエリーに対して付与されているロックの数の確認を行えます。

それぞれのトランザクション統計タグを以下の表にまとめます。

タグ	説明	例
<TransactionManager-Stats>	開始したトランザクションの数、コミットしたトランザクションの数、ロールバックしたトランザクションの数についての情報。	<pre><TransactionManager-Stats> <NumberTransactionsStarted>2</NumberTransactionsStarted> <NumberTransactionsCommitted>1</NumberTransactionsCommitted> <NumberTransactionsRolledback>0</NumberTransactionsRolledback> </TransactionManager-Stats> <LockManager-Stats> . . . </LockManager-Stats> <LogManager-Stats> . . . </LogManager-Stats> </Transaction-Stats></pre>
<LockManager-Stats>	使用または要求されている各種のロックの数についての情報。ロックタイプの情報は、組み込みタグインデックス、データインデックス、タグ+データインデックス、マップファイルについて示されます。	

<LogManager-Stats>	トランザクションログに報告されている情報（注: ユーザーはトランザクションログを読めません。トランザクションログの内容は、サーバー内で内部的に使用されます）。記録される情報としては、開始したトランザクションの数、ロールバックしたトランザクションの数、コミットしたトランザクション数、サーバーインデックスに組み込まれたエントリの数などがあります。	
--------------------	--	--

情報取得のためのログファイルの使用

NeoCore XMS には、参照用に 2 つのログファイルが用意されています。

NeoCore XML 情報ログファイル

名前	位置	機能
Server.log	NeoCore/neoxml/log	<p>NeoServer の初期設定と NeoServer に対する要求についての情報が書き込まれます。トレースレベルの設定に応じて、各エントリの情報の詳細レベルが変わります。</p> <p>ログファイルの形式は、以下のとおりです。</p> <p>Server.log</p> <p>NeoServer が起動するたびに、新しいログが開きます。ログファイルは、午前 0 時にロールオーバーします。</p> <p>古いログは、Server[timestamp]log とリネームされて保存されます。</p>

パフォーマンス改善のためのチューニング

チューニングとは、パフォーマンスを改善するために構成設定を最適化する作業のことです。

注: ファイルシステムの領域が満杯になっても、エラーが表示されない場合もあります。したがって、ディスク領域が満杯にならないようにモニタリングを行う必要があります。そうしなければ、予測できない結果が生じるおそれがあります。

データベースには、データベースの最大サイズやパフォーマンスを制御するための設定が数多く用意されています。ここでは、チューニングを以下の3種類に分けることにします。

1. ディスク入出力の最適化（ファイルパス）
2. メモリーの最適化
3. データタイプの最適化

ディスク入出力の最適化（ファイルパス）

ディスクパフォーマンスを最適化するために、トランザクションログ、ページファイル、インデックス、マップファイル、ディスク上に設ける退避用のバッファにはそれぞれ別個のハードディスクを使用することを推奨します。一般に、ディスクが多ければ多いほど、ディスクのパフォーマンスは良くなります。

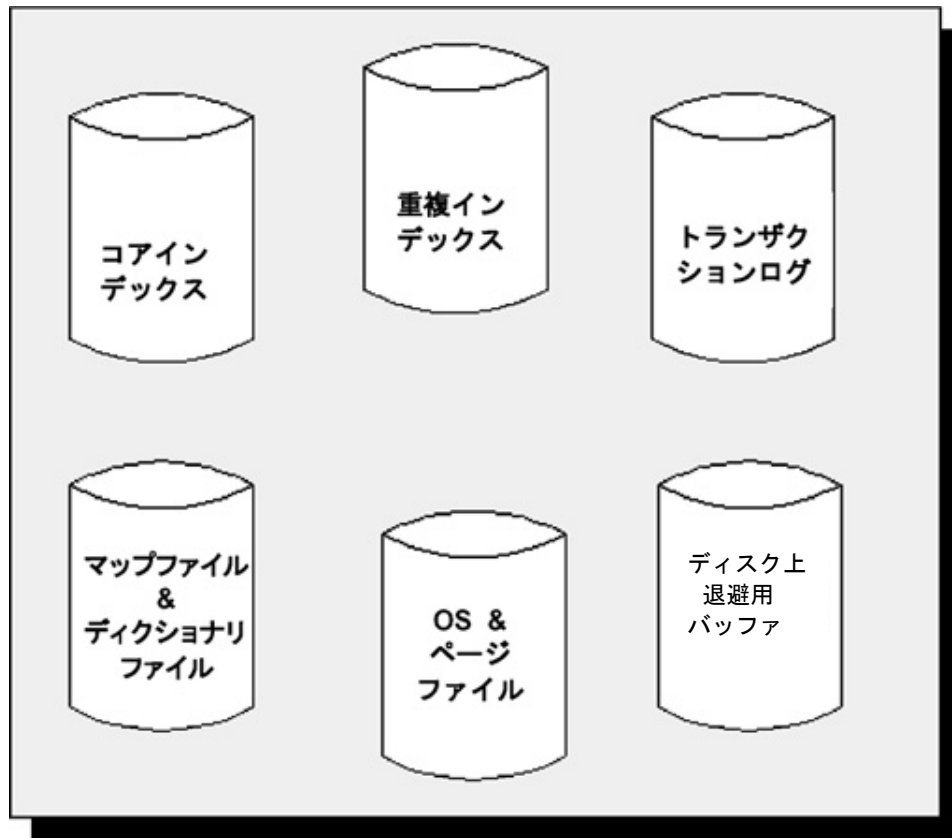
第1 優先事項: データベースファイルを OS ページファイルとは別のディスクに置きます。

第2 優先事項: トランザクションログを専用のディスクに置きます。

第3 優先事項: タグ重複インデックス、タグ+データ重複インデックス、各種コアインデックスをそれぞれ別個のディスクに分けます。

注: これらのファイルの位置は、**NeoXDBRuntime.xml** ファイルと **NeoDatabase.xml** ファイルで確認できます。

以下の図に示すのは、必ずしも最適な最小構成ではありませんが、最小構成の良い例です。



ディスク入出力の最適化の例

メモリーの最適化

NeoCore XMS Windows 版の Windowing:

Windowing のサイズ設定は、NeoXDBRuntime.xml ファイルで調整できます。

注: Windowing は、Solaris では使用しません。

NeoCore XMS Solaris 版の Windowing (使用不可):

Solaris では、NeoServer が 64 ビット処理に対応しているので、Windowing の必要はありません。つまり、データベース全体を NeoServer の処理領域 (理論上は 18 EB) のメモリーに簡単にマップして、OS レベルでファイルのページングを管理できるということです。

データベースのフットプリントが使用可能な処理アドレス領域 (約 1.25 ギガバイト [GB]) を超過する場合は、データベースファイルの Windowing を実行する必要がありますが、その際には以下のようにしてください。

Windowing を実行する場合でも、Windowing をできるだけ少なくするために、以下の順序でファイルを選択します。

1. マップファイルの Windowing
2. データディクショナリファイルの Windowing
3. 重複ファイルの Windowing
4. コアインデックスの Windowing

コアインデックス:	重複ファイル:
タグ	タグ
タグ+データ	タグ+データ
データ	データ

- 一般に、Windowing のサイズが小さければ小さいほど、一度にフラッシュしなければならないメモリーが少なくて済むため、パフォーマンスは良くなります。ただし、それと引き換えに 1 つの問題があります。つまり、Windowing のサイズが小さければ小さいほど、データや位置を見つけるためにより多くの「スライド処理」が必要になるということです。

メモリーバッファプールのサイズ

- スレッドのコンテキストが多ければ多いほど、より多くのバッファが必要になります。また、重複データが多ければ多いほど、より大きなバッファが必要になります。メモリーバッファプールのサイズには、バッファのサイズと数の両方が関係しています。

注: XmlBufferSize パラメータは、バッファプール内の最大バッファのサイズ以下である必要があります。また、バッファプール内の 1 つのバッファのサイズと正確に一致している必要もあります。さらに大きなバッファプール設定が必要であれば、[**Server Administration Console**] の [**Settings**] タブの [**NeoCore HTTP Server Settings**] ウィンドウで値を増やせます。

その一例を以下の表に示します。XML バッファサイズは 2097152 (NeoServer.xml 内の最大バッファのサイズ) 以下でなければいけません。

バッファプールサイズの例

XML バッファプール数	バッファプールのサイズ
1	デフォルト
2	512
3	1024
4	2048
5	8192
6	65536
7	131072
8	512000
9	1024000
10	2097152

データタイプの最適化

データベースファイルのスライスは、格納するデータに合わせてできるだけ小さくすることをお勧めします（そうすれば、メモリー使用とディスクアクセス速度の制限の中でパフォーマンスを最適化できます）。格納する文書に合わせてファイルのサイズを増やさなければならない場合もあります。

注：データディクショナリのサイズと最小限必要なマップファイルの数との間には関連があります。最小限必要なマップファイルの数を計算するには、データディクショナリのサイズを一番近い 4 GB の倍数に丸め、その結果を 4 GB で除算します。例えば、データディクショナリが 7 GB であれば、少なくとも 2 つのマップファイルが必要です。最小限必要な数よりも多くのマップファイルを用意してもかまいません。この情報は、**NeoDatabase.xml** ファイルで設定できます。

データタイプを最適化するには、格納する文書の構造に基づく 4 種類のデータベース構造について検討する必要があります。

- 多数の重複タグを含むデータベース（つまり、同じタグを多数含む文書）。すべての重複タグのカンタムが十分な大きさであることを確認してください。
- 大量の重複データを含むデータベース（データインデックス構築がオンの場合のみ）。例えば、Colorado 州 Colorado Springs 在住の 100,000 人分のエントリを含む電話帳に、1000 人の Smith と 1000 人の Jones のエントリが存在するような場合です。このような場合は、データインデックスのサイズを十分な大きさに増やします。多数のタグ+データ重複を生み出しているタグ重複とデータ重複の大きな交差部分に注意してください。
- 大量の非重複データを含むデータベース。データディクショナリのサイズを増やします。

- 多数の非重複タグを含むデータベース。タグディクショナリのサイズを増やし、重複カンタムを小さくし、タグインデックスコアのサイズを増やします。

NeoCore XMS 構成変更の要件

NeoDatabase.xml ファイルのパラメータを変更した場合は、データベースの再構築を行う必要があります。データベースの再構築については、「データベースの再構築」を参照してください。

NeoServer.xml ファイルのパラメータを変更した場合は、データベースの再起動を行う必要があります。ただし、NeoServer > GlobalSettings > TraceLevel のみ再起動は必要ありません。

NeoXDBRuntime.xml ファイルのパラメータを変更した場合は、データベースの再起動を行う必要があります。

ただし、Runtime > Admin の値を変更した場合は、データベースの再構築を行う必要があります。データベースの再構築については、「データベースの再構築」を参照してください。

ファイルの構成

このセクションでは、以下のファイルの構成について取り上げます。

- NeoServer.xml
- NeoXDBRuntime.xml
- NeoDatabase.xml

NeoDatabase.xml ファイルの構成

NeoCore XMS インデックス

NeoCore XMS インデックスは、XML ノード群を高速で検索するためのデータ構造です。NeoCore XMS には、インデックスを収容する 3 つの格納単位があります。

1. タグ（オンリー）
2. タグ+データ
3. データ（オンリー）

これらの 3 つのインデックスはいずれも各データベースで必須の部分です。「タグ」と「タグ+データ」には常に値が取り込まれます。「データ」は、インデックスモードがデータオンリークエリーモードに設定されていない限り、最小限の値だけが取り込まれます。

データオンリークエリーモードは、処理オーバーヘッドが大きくなるため現在のバージョンでは推奨されません。

インデックスのサイズの変更

NeoDatabase.xml ファイルを編集して、インデックスのサイズを変更できます。インデックスサイズの変更は、インデックス再構築を完了して NeoCore XMS を起動した後に、サーバー統計ページに示されます（コンソールの **[Server Status]** タブを選択してください）。

注：インデックスのサイズを減らす係数が大きすぎてサイズ変更が失敗し、インデックス再構築が失敗してしまった場合は、**NeoDatabase.xml** でインデックスサイズを増やしてから、ReIndex コマンドを再度実行してください。

NeoDatabase.xml ファイルのパラメータの説明

名前	説明
Name	デフォルト:DEFAULT_DB
IndexMode	デフォルト: 1 推奨設定:1 データ専用クエリーを実行する場合は、0 に変更します（現在のバージョンでは推奨されません）。
TransactionLogDirectory	トランザクションログの場所を指定します。 デフォルト値: Windows: {NeoCore インストールディレクトリ}¥neoxml¥db Solaris: {NeoCore インストールディレクトリ}/neoxml/db
MapFiles	マップファイル（ファイル名:neoXDBXXX.map）の設定を行います。 マップファイルには、ストアする XML の物理的な構造情報が格納されます。 作成されたファイルのうち、1 つはデフラグメンテーションのためのスペアファイルとして使用されます。

MapFiles - NumberOfMapFiles	作成するマップファイルの数を指定します。 デフォルト: 5 推奨設定: データストアのサイズによって異なります。
MapFiles - Size	デフォルト値: Enterprise 版: 21 Developer 版: 9 推奨設定: データストアのサイズに比例して使用領域が増加します。 サイズは 3 の倍数である必要があります。
MapFiles - Location	マップファイルの場所を指定します。このエントリは、構成するマップファイルごとに繰り返されます。例えば、2 つのマップファイルを構成する場合、パラメータ名は、Location NeolD="Map 001"、Location NeolD="Map 002" のようになります。 デフォルト値: Windows: {NeoCore インストールディレクトリ}¥neoxml¥db Solaris: {NeoCore インストールディレクトリ}/neoxml/db 推奨設定: 小規模なデータベースの場合は、ファイルをデフォルトの位置のままにします。大規模なデータベースの場合や、パフォーマンスが重要な中規模のデータベースの場合は、ファイルを別々のディスクに置きます（「 パフォーマンス改善のためのチューニング 」を参照してください）。
DataDictionary	データディクショナリ（ファイル名: neoXDBD.dct）に関する設定を行います。 データディクショナリには、ユニークなデータ値と属性地が格納されます。

DataDictionary - Size	<p>データディクショナリのサイズを指定します。</p> <p>推奨設定:ユニークなデータ量に比例して使用領域が増加します。</p> <p>デフォルト値:</p> <p>Enterprise 版:50</p> <p>Developer 版:10</p>
DataDictionary - Location	<p>データディクショナリの場所を指定します。</p> <p>デフォルト値:</p> <p>Windows:</p> <p>{NeoCore インストールディレクトリ}¥neoxml¥db</p> <p>Solaris:</p> <p>{NeoCore インストールディレクトリ}/neoxml/db</p> <p>推奨設定:小規模なデータベースの場合は、ファイルをデフォルトの位置のままにします。大規模なデータベースの場合や、パフォーマンスが重要な中規模のデータベースの場合は、ファイルを別々のディスクに置きます（「パフォーマンス改善のためのチューニング」を参照してください）。</p>
CrossReference	<p>クロスリファレンス（ファイル名 :neoCR.crf）に関する設定を行います。</p> <p>クロスリファレンスには、DataDictionary に格納される各エントリに関連したエントリが格納されます。</p>
CrossReference - Size	<p>クロスリファレンスのサイズを指定します。</p> <p>デフォルト値:3</p> <p>推奨設定:ユニークなデータ量に比例して必要な使用領域が増加します。</p>
CrossReference - Location	<p>クロスリファレンスの場所を指定します。</p> <p>デフォルト値:</p> <p>Windows:</p> <p>{NeoCore インストールディレクトリ}¥neoxml¥db</p> <p>Solaris:</p> <p>{NeoCore インストールディレクトリ}/neoxml/db</p>

NoDTMems	タグ、データ、タグ+データすべての DTMem の数を指定します。
DataIndex	データ部のインデックスです。
DataIndex - Core	データコアインデックス（ファイル名： neoData.inx ）に関する設定です。 データコアインデックスには、ユニークなデータのインデックスが格納されます。
DataIndex - Core - Size	データコアインデックスのエントリ数を 2 の 2 乗の指数として指定します。この数値を「 カンタム 」ともいいます。各データインデックスエントリは 12 バイトです。 デフォルト値: Enterprise 版:20 Developer 版:18 (2 の 18 乗 = 2^{18} = 262144 個のデータインデックスエントリを格納) ディスク領域の計算:1 エントリごとに 12 バイト: 2^{18} (デフォルト) * 12 バイト = 3MB 推奨設定: ユニークなデータ量に比例して、使用領域が増加します。 注: カンタムの値を 1 つ増やすと、インデックスの容量は 2 倍になります。したがって、19 (2^{19}) のカンタムのエントリ数は、18 (2^{18}) のカンタムのエントリ数の 2 倍になります。
DataIndex - Core - Location	データコアインデックスの場所を指定します。 デフォルト値: Windows: {NeoCore インストールディレクトリ}¥neoxml¥db Solaris: {NeoCore インストールディレクトリ}/neoxml/db 推奨設定: 小規模なデータベースの場合は、ファイルをデフォルトの位置のままにします。大規模なデータベースの場合や、パフォーマンスが重要な中規模のデータベースの場合は、ファイルを別々のディスクに置きます（「 パフォーマンス改善のためのチューニング 」を参照してください）。

DataIndex - MTMem	<p>データ重複インデックス（ファイル名：neoDataMTMem.dup）に関する設定を行います。</p> <p>データ重複インデックスには、重複するデータのインデックスが格納されます。</p>
DataIndex - MTMem - Size	<p>データ重複インデックスのファイルサイズをメガバイト単位で指定します。</p> <p>デフォルト: 1</p> <p>推奨設定:データの重複インデックスは、データオンリークエリー機能(推奨されていません)が ON でない場合、デフォルト値で十分です。</p>
DataIndex - MTMem - Location	<p>データ重複インデックス(neoDataMTMem.dup)の場所を指定します。</p> <p>デフォルト値:</p> <p>Windows:</p> <p>{NeoCore インストールディレクトリ}\%neoxml%\db</p> <p>Solaris:</p> <p>{NeoCore インストールディレクトリ}/neoxml/db</p> <p>推奨設定:小規模なデータベースの場合は、ファイルをデフォルトの位置のままにします。大規模なデータベースの場合や、パフォーマンスが重要な中規模のデータベースの場合は、ファイルを別々のディスクに置きます（「パフォーマンス改善のためのチューニング」を参照してください）。</p>
DataIndex - MTMem - NoMapBlocks	<p>注:非常に大きなデータをストアする場合以外は、この値を 4 のままにすることを推奨します。</p>
DataIndex - TmpMTMem	<p>データ重複インデックス（ファイル名：neoDataTmpMTMem.dup）に関する設定を行います。</p>
DataIndex - TmpMTMem - Size	<p>データ重複インデックス（neoDataTmpMTMem.dup）のファイルサイズを指定します。</p> <p>デフォルト値: 1</p> <p>推奨設定:データの重複インデックスは、データオンリークエリー機能(推奨されていません)が ON でない場合、デフォルト値で十分です。</p>

DataIndex - DTMem	データ重複インデックス（ファイル名: neoDataDTMemXXX.dup ）に関する設定を行います。
DataIndex - DTMem - Size	データ重複インデックス（ neoDataDTMemXXX.dup ）のファイルサイズを指定します。 デフォルト値: 1 推奨設定: データの重複インデックスは、データオンリークエリー機能(推奨されていません)が ON でない場合、デフォルト値で十分です。
DataIndex - DTMem - Location	データ重複インデックス（ neoDataDTMemXXX.dup ）の場所を指定します。 デフォルト値: Windows: {NeoCore インストールディレクトリ}¥neoxml¥db Solaris: {NeoCore インストールディレクトリ}/neoxml/db
DataIndex - DTMem - ArraySizeBits	デフォルト値: 4
TagDictionary	タグディクショナリ（ファイル名: neoXDBT.dct ）に関する設定を行います。 ユニークな flattened タグと、タグの順列が格納されます。
TagDictionary - Size	タグディクショナリのサイズをメガバイト単位で指定します。 デフォルト値: 1 推奨設定: ユニークなタグ名の数に比例して使用領域が増加します。タグ構造が一般的なデータを格納する場合は増加しません。

TagDictionary - Location	<p>タグディクショナリの場所を指定します。</p> <p>デフォルト値:</p> <p>Windows:</p> <pre>{NeoCore インストールディレクトリ}¥neoxml¥db</pre> <p>Solaris:</p> <pre>{NeoCore インストールディレクトリ}/neoxml/db</pre> <p>推奨設定:小規模なデータベースの場合は、ファイルをデフォルトの位置のままにします。大規模なデータベースの場合や、パフォーマンスが重要な中規模のデータベースの場合は、ファイルを別々のディスクに置きます（「パフォーマンス改善のためのチューニング」を参照してください）。</p>
TagIndex	<p>タグ部のインデックスです。</p>
TagIndex - Core	<p>タグコアインデックス（ファイル名: neoTag.inx）に関する設定を行います。</p> <p>タグコアインデックスには、ユニークなタグのインデックスが格納されます。</p>
TagIndex - Core - Size	<p>タグコアインデックスのエントリ数を 2 の 2 乗の指数として指定します。この数値を「カンタム」ともいいます。各データインデックスエントリは 12 バイトです。デフォルト値:17</p> <p>(2 の 17 乗 = 2^{17} = 131072 個のデータインデックスエントリを格納)</p> <p>ディスク領域の計算:エントリごとに 2^{17} (デフォルト) * 16 バイト = 2MB</p> <p>推奨設定:ユニークなタグ名の数に比例して使用領域が増加します。タグ名、タグ構造が一様なデータを格納する場合は増加しません。</p> <p>注:カンタムの値を 1 つ増やすと、インデックスの容量は 2 倍になります。したがって、18 (2^{18}) のカンタムのエントリ数は、17 (2^{17}) のカンタムのエントリ数の 2 倍になります。</p>

TagIndex - Core - Location	<p>タグコアインデックスの場所を指定します。</p> <p>デフォルト値:</p> <p>Windows:</p> <p>{NeoCore インストールディレクトリ}¥neoxml¥db</p> <p>Solaris:</p> <p>{NeoCore インストールディレクトリ}/neoxml/db</p> <p>推奨設定:小規模なデータベースの場合は、ファイルをデフォルトの位置のままにします。大規模なデータベースの場合や、パフォーマンスが重要な中規模のデータベースの場合は、ファイルを別々のディスクに置きます（「パフォーマンス改善のためのチューニング」を参照してください）。</p>
TagIndex - MTMem	<p>タグ重複インデックス（ファイル名：neoTagMTMem.dup）に関する設定を行います。</p> <p>タグ重複インデックスには、重複するタグのインデックスが格納されます。</p>
TagIndex - MTMem - Size	<p>タグ重複インデックス（neoTagMTMem.dup）のファイルサイズをメガバイト単位で指定します。</p> <p>デフォルト:2</p> <p>推奨設定:タグ名、タグ構造がー様なデータストアの場合は増加しません。タグ名の重複が多いデータストアの場合は、サイズを増やす必要があります。必要な値は、ストアするデータの特性によって異なります。</p>
TagIndex - MTMem - Location	<p>タグ重複インデックス（neoTagMTMem.dup）の場所を指定します。</p> <p>デフォルト値:</p> <p>Windows:</p> <p>{NeoCore インストールディレクトリ}¥neoxml¥db</p> <p>Solaris:</p> <p>{NeoCore インストールディレクトリ}/neoxml/db</p> <p>推奨設定:小規模なデータベースの場合は、ファイルをデフォルトの位置のままにします。大規模なデータベースの場合や、パフォーマンスが重要な中規模のデータベースの場合は、ファイルを別々のディスクに置きます（「パフォーマンス改善のためのチューニング」を参照してください）。</p>

TagIndex - MTMem - NoMapBlocks	デフォルト値: 4
TagIndex - TmpMTMem	タグ重複インデックス (neoTagTmpMTMem.dup) に関する設定を行います。
TagIndex - TmpMTMem - Size	タグ重複インデックス (neoTagTmpMTMem.dup) のサイズを指定します。 デフォルト値: 2 推奨設定 : 通常、サイズを増やす必要はありません。
TagIndex - DTMem	タグ重複インデックス (ファイル名: neoTagDTMemXXX.dup) に関する設定を行います。 タグ重複インデックスには、重複するタグのインデックスが格納されます。1 つはデフラグメンテーションのためのスペアファイルとして使用されます。
TagIndex - DTMem - Size	タグ重複インデックス (neoTagDTMemXXX.dup) のサイズを指定します。 デフォルト値: 10 推奨設定 : 重複するタグの数に比例して使用領域が増加します。
TagIndex - DTMem - Location	タグ重複インデックス (neoTagDTMemXXX.dup) の場所を指定します。 デフォルト値: デフォルト値: Windows: {NeoCore インストールディレクトリ}¥neoxml¥db Solaris: {NeoCore インストールディレクトリ}/neoxml/db

TagIndex - DTMem - ArraySizeBits	デフォルト値: 6
TagPlusDataIndex	タグ+データインデックスは、タグ+データ部のインデックスです。
TagPlusDataIndex - Core	タグ+データコアインデックス（ファイル名： neoTPD.inx ）に関する設定を行います。 タグ+データコアインデックスには、ユニークなタグ+データ部のインデックスが格納されます。
TagPlusDataIndex - Core - Size	タグ+データコアインデックス（ neoTPD.inx ）内のエントリ数を 2 の 2 乗の指数として指定します。この数値を「カンタム」ともいいます。各データインデックスエントリは 12 バイトです。 デフォルト:20（つまり、2 の 17 乗 = 2^{17} = 131072 個のデータインデックスエントリ） 推奨設定: ユニークな、タグ+データ部の数に使用領域が比例します。 注: カンタムの値を 1 つ増やすと、インデックスの容量は 2 倍になります。したがって、19（ 2^{18} ）のカンタムのエントリ数は、18（ 2^{17} ）のカンタムのエントリ数の 2 倍になります。
TagPlusDataIndex - Core - Location	タグ+データコアインデックス（ neoTPD.inx ）の場所を指定します。 デフォルト: デフォルト値: Windows: {NeoCore インストールディレクトリ}¥neoxml¥db Solaris: {NeoCore インストールディレクトリ}/neoxml/db 推奨設定: 小規模なデータベースの場合は、ファイルをデフォルトの位置のままにします。大規模なデータベースの場合や、パフォーマンスが重要な中規模のデータベースの場合は、ファイルを別々のディスクに置きます（「 パフォーマンス改善のためのチューニング 」を参照してください）。

TagPlusDataIndex - MTMem	<p>タグ+データ重複インデックス（ファイル名：neoTPDMTMem.dup）に関する設定を行います。</p> <p>タグ+データ重複インデックスには、重複するタグ+データのインデックスが格納されます。</p>
TagPlusDataIndex - MTMem - Size	<p>タグ+データ重複インデックス（neoTPDMTMem.dup）のファイルサイズをメガバイト単位で指定します。</p> <p>デフォルト: 10</p> <p>推奨設定: 重複するタグ+データの数に比例して使用領域が増加します。</p>
TagPlusDataIndex - MTMem - Location	<p>タグ+データ重複インデックス（neoTPDMTMem.dup）の場所を指定します。</p> <p>デフォルト値:</p> <p>Windows:</p> <p>{NeoCore インストールディレクトリ}¥neoxml¥db</p> <p>Solaris:</p> <p>{NeoCore インストールディレクトリ}/neoxml/db</p> <p>推奨設定: 小規模なデータベースの場合は、ファイルをデフォルトの位置のままにします。大規模なデータベースの場合や、パフォーマンスが重要な中規模のデータベースの場合は、ファイルを別々のディスクに置きます（「パフォーマンス改善のためのチューニング」を参照してください）。</p>
TagPlusDataIndex - MTMem - NoMapBlocks	<p>デフォルト値: 4</p>
TagPlusDataIndex - TMPMTMem	<p>タグ+データ重複インデックス（ファイル名：neoTPDTmpMTMem.dup）に関する設定を行います</p>
TagPlusDataIndex - TMPMTMem - Size	<p>タグ+データ重複インデックス（neoTPDTmpMTMem.dup）のサイズを指定します。</p> <p>デフォルト値: 2</p> <p>推奨設定: 通常は、サイズを増やす必要はありません。</p>

TagPlusDataIndex - DTMem	<p>タグ+データ重複インデックス（ファイル名：neoTPDDTMemXXX.dup）に関する設定を行います。</p> <p>タグ+データ重複インデックスには、重複するタグ+データのインデックスが格納されます。1つはデフラグメンテーションのためのスペアファイルとして使用されます。</p>
TagPlusDataIndex - DTMem - Size	<p>タグ+データ重複インデックス（neoTPDDTMemXXX.dup）のサイズを指定します。</p> <p>デフォルト値: 10</p> <p>推奨設定：重複するタグ+データの数に比例して使用領域が増加します。</p>
TagPlusDataIndex - DTMem - Location	<p>タグ+データ重複インデックス（neoTPDDTMemXXX.dup）の場所を指定します。</p> <p>デフォルト値:</p> <p>Windows:</p> <p>{NeoCore インストールディレクトリ}¥neoxml¥db</p> <p>Solaris:</p> <p>{NeoCore インストールディレクトリ}/neoxml/db</p> <p>推奨設定:小規模なデータベースの場合は、ファイルをデフォルトの位置のままにします。大規模なデータベースの場合や、パフォーマンスが重要な中規模のデータベースの場合は、ファイルを別々のディスクに置きます（「パフォーマンス改善のためのチューニング」を参照してください）。</p>
TagPlusDataIndex -DTMem - ArraySizeBits	<p>デフォルト値: 4</p>

NeoServer.XML ファイルの構成

名前	説明
DefaultLocale	NeoServer のデフォルトロケールを指定します。 設定値:en, ja
DefaultEncoding	NeoServer のデフォルトエンコードを指定します。 設定値:UTF-8, Shift_JIS, EUC-JP, ISO-2022-JP
PreserveRoundtripEncoding	マルチエンコーディングサポートを設定します。 設定値:TRUE (マルチ) または FALSE (UTF-8 のみ)
TraceLevel	すべてのデータベース機能ログのレベルを指定します。 Mitsui NeoCore Center では、必要な場合、デフォルト (FATAL、WARNING、ERR) 以外のトラブルシュート用のレベルをご提供することがあります。
MemoryBufferPools	NeoServer の起動時に、各スレッドは 2 つの最大バッファと 1 つの空きバッファを使用します。NeoServer のスレッドは、別の構成ファイル (以下のパラメタ) で構成します。 NeoXDBRuntime.xml Runtime > MaxNeoXDBThreads
MemoryBufferPool - XmlBufferSize	最大 XML 戻りセットのサイズをバイト単位で指定します。 最大バッファサイズ以下の値である必要があります。
MemoryBufferPools - DiskBackedBuffersLocation	カーソル/チャンククエリーで使用する、ディスク上のバックアップ用バッファの場所を指定します。
MemoryBufferPools - BufferPool	事前に割り振られているメモリーバッファプールの情報を指定します。 NeoServer の起動時には、仮想メモリーが使われます。 メモリー使用量:サイズ x 数 重複してその他の割り振りを作成する場合は、NeoID 属性値を Pool_11、Pool_12 などと増やしていきます。

MemoryBufferPools - BufferPool - Size	各バッファのサイズをバイト単位で指定します。
MemoryBufferPools - BufferPool - Count	プール内のバッファ数を指定します。
Component	NeoCore 付属コンポーネントの構成を指定します。
Component="JVM" - Javahome	JVM のホームディレクトリを指定します。 この値は編集しないでください。
Component="JVM" - VmOptions	JVM パラメータを指定するときに使用するオプション。 この値は編集しないでください。
Component = "HTTP Server"	これは、API や Web コンソールによってすべての HTTP 呼び出しを処理する組み込み HTTP サーバーの構成です。
Component = "HTTP Server" - Network	これは、HTTP サーバーのネットワーク関連パラメータの構成です。
Component = "HTTP Server" - Network - MaxConnections	MaxConnections パラメータでは、NeoServer に対する並行ソケット接続の最大数を指定します。 デフォルト値の 256 を推奨します。
Component = "HTTP Server" - Network - IP	HTTP サーバーの TCP/IP バインドを指定します。 以下の 2 つの IP モジュールを構成します。 NeolD="XMS" (データベースアクセス用) NeolD="Console" (Web コンソールアクセス用)

Component = "HTTP Server" - Network - IP - Address	IP モジュールに割り当てる IP アドレス。 NIC に対するすべてのトラフィックのデフォルトになります。 マシンに複数の NIC がある場合は、重複する可能性があります。 明示的な IP アドレスを使用してください。 NeoID 属性値を NIC1、NIC2 などと増分してください。
Component = "HTTP Server" - NetWork - IP - Alias	このモジュールの URL パスを指定します。
Component = "HTTP Server" - Threads - ThreadPoolSize	プールされる HTTP スレッドの数を設定します。
Component = "HTTP Server" - Threads - ThreadStackSize	スレッド毎のスタックサイズを設定します。
Component = "HTTP Server" - Module	起動時にロードする複数の NeoServer モジュールを指定します。 この値は変更しないでください。 Loadlib - NeoServer モジュールのライブラリをロードします。 「NeoXMS」は、コアデータベースモジュールです。 「NeoJVM」は、JVM モジュールです。
Component = "HTTP Server" - AccessLog	値: 「ON」または「OFF」。デフォルトは「OFF」。 アクセスログ（すべての HTTP 要求のログ）のオン/オフを切り替えます。

NeoXDBRuntime.xml

NeoXDBRuntime.xml ファイルについての全般的な注釈:

- 一般に、コアインデックスの Windowing 設定では、ファイル全体を 1 つのウィンドウとして扱うのが望ましいと言えます。コアインデックスのデータの内部構造は複雑なので、Windowing のサイズを指定すると、ファイルアクセスのたびにスライドが行われる結果になってしまうからです。コアインデックスには、タグインデックス、データインデックス、タグ+データインデックスの 3 種類があります。

NeoXDBRuntime.xml ファイルのパラメータの定義

MaxNeoXdbThreads	<p>コンテキストプールで作成される NeoXdbThreadContext オブジェクトの最大数。この値によって、並行して実行できるサーバースレッドの数を制限します。</p> <p>デフォルト: 5</p> <p>推奨設定: システムによって異なります。メモリー、プロセッサ数などのシステムリソースに基づいて調整できます。</p>
StripWhiteSpace	<p>TRUE に設定すると、データ部前後の空白や空白が取り除かれます。</p> <p>例:</p> <p>ストア前のデータ:<Sample> Space </Sample></p> <p>ストア後のデータ<Sample>Space</Space></p> <p>FALSE に設定すると、データ部前後の空白や空白がそのまま残ります。</p>
Sessions	<p>NeoServer のセッションについての設定です。</p>
Sessions - Timeout	<p>不在セッションがタイムアウトになるまでの時間の長さ。値は秒単位で指定します。</p> <p>デフォルト: 86400 (24 時間)</p> <p>推奨設定: アプリケーションのニーズによって異なります。</p>

Sessions - TimerInterval	<p>サーバーがセッションのタイムアウトをポーリングする頻度。値は秒単位で指定します。これは、トランザクションのタイムアウトをポーリングする頻度にもなります。</p> <p>デフォルト: 60</p> <p>推奨設定: デフォルトと同じです。</p>
Sessions - MaxSessions	<p>同時セッションの最大数。これは、サーバーで同時に発生できるイベントの数です。</p> <p>デフォルト: 1000</p> <p>推奨設定: アプリケーションによって異なります。</p>
Sessions - AllowAnonymousUser	<p>ログインせずに、URL 経由などの方法でコマンドを実行できるようにします。</p> <p>デフォルト: FALSE</p> <p>推奨設定: アプリケーションによって異なります。</p>
Sessions - AllowClearTextPasswords	<p>TRUE: クリアテキストのパスワードを使用して、POST メソッドと GET メソッドの両方を使用できます。</p> <p>FALSE: POST メソッドと MDS エンコードパスワードだけを使用できます。</p> <p>デフォルト: FALSE</p>
ResultChunking	<p>カーソル、チャンキングに関する設定です。</p>
ResultChunking - MaxNeoResultsPerSession	<p>デフォルト: 2</p>
ResultChunking - MaxNeoChunksPerNeoResult	<p>デフォルト: 1000</p>
ResultChunking - MaxNeoResults	<p>デフォルト: 25</p>

Result Chunking - MaxFitBufferSize	デフォルト: 1024000
QuerySize	<p>QuerySize を増やせば、ステートメント、式ノード、変数の許可数が増えます（これらはすべて相互に関連しているため、QuerySize を 200 に設定すると、この 3 つのサイズはすべて 2 倍になります）。QuerySize を 100 増やせば、スレッドあたり約 16K だけメモリー割り振り量が増えます。</p> <p>デフォルト: 100</p>
Admin - Location	<p>管理ファイルの場所を指定します。</p> <p>デフォルト:</p> <p>{NeoCore インストールディレクトリ}¥neoxml¥db(Windows)</p> <p>{NeoCore インストールディレクトリ}/neoxml/db(Solaris)</p>
Transactions	トランザクションに関する設定です。
Transactions - MaxLogSize	<p>新しいトランザクションログを作成することになる前に、トランザクションログをどこまで拡大できるかをバイト単位で指定します。</p> <p>デフォルト:10,485,760 バイト（10 MB）</p> <p>推奨設定:1 つのトランザクションログファイルに含めるデータ量によって異なります。</p>

<p>Transactions</p> <p>- LockHeadPoolSize</p>	<p>付与できるロックの最大数を指定します。この値は、1 つのリソースに対応します。この場合のリソースとは、インデックス、文書、マップファイルのことです。付与可能なロックの数は、要求可能なロックの数よりも多くなります。複数のロック要求が、付与されている同じロックを使用することがあるからです。</p> <p>例えば、データベース内に 1000 個の文書があり、1000 個の文書すべてに対してクエリーを実行する場合は、トランザクションロックに 1000 を使用することになります。これは、クエリーごとの値です。複数のクエリーを実行する場合は、クエリーごとにサイズが異なります。</p> <p>デフォルト: 100,000</p> <p>推奨設定: 使用可能なシステムメモリーの量と、NeoCore XMS でのトランザクション実行の程度によって異なります。大きな値にすればするほど、より多くのシステムメモリーが必要になります。</p> <p>注: LockHeadPoolSize の設定値が大きすぎると、システムリソースを過剰に使用し、サーバーのメモリ使用率を増やす可能性があります。</p>
<p>Transactions</p> <p>- LockRequestPoolSize</p>	<p>要求できるロックの最大数を指定します。要求可能なロック要求の数は、付与可能なロックの数よりも少なくなります。複数のロック要求が、付与されている同じロックを使用することがあるからです。</p> <p>ロック（トランザクション）を要求すると、LockRequestPoolSize のサイズを入手することになります。例えば、システムに 1000 個の文書があり、2 つのトランザクションで同時に 1 つのクエリーを実行する場合は、LockHeadPoolSize には 1000 を、LockRequestPoolSize には 2000 を使用することになります。</p> <p>LockRequestPoolSize は、LockHeadPoolSize よりもメモリーを使用しません。</p> <p>デフォルト: 200,000</p> <p>推奨設定: 使用可能なシステムメモリーの量と、NeoCore XMS でのトランザクション実行の程度によって異なります。大きな値にすればするほど、より多くのシステムメモリーが必要になります。</p>

<p>Transactions</p> <p>- MaxDuration</p>	<p>トランザクションの完了に費やせる最大時間（つまり、トランザクションの一部がタイムアウトになったりロールバックされたりするまでの時間）を指定します。</p> <p>デフォルト:300 秒</p> <p>推奨設定:大規模なトランザクションや大規模な挿入操作や変更操作を実行しない小さな文書のシステムの場合は、この制限を減らせます。非常に大きな（メガバイト級の）文書を格納して、アプリケーションで複雑なトランザクションロジックを実行するような場合は、この制限を増やせます。</p>
<p>Transactions</p> <p>- InactivityDuration</p>	<p>トランザクションをアイドル状態（アクションが実行されない状態）にしておける最大時間（つまり、強制ロールバックが行われるまでの時間）を指定します。</p> <p>デフォルト:30 秒</p> <p>推奨設定:トランザクション内の操作と操作の間に一時停止を入れる場合は、この値を増やします。</p>
<p>Transactions</p> <p>- LockTimeout</p>	<p>トランザクションがロックの付与を待機する最大時間を指定します。</p> <p>デフォルト:10 秒</p> <p>推奨設定:多数の大規模なトランザクションを同時に実行する場合は、この制限を増やせます。</p>
<p>Transactions</p> <p>- MaxLockRequests</p>	<p>グローバルロックが要求される前のロック要求の最大数を指定します。</p> <p>デフォルト: 90000</p> <p>推奨設定:デフォルトと同じです。</p>
<p>Transactions</p> <p>- CheckpointInterval</p>	<p>チェックポイント間の間隔（秒単位）です。</p> <p>デフォルト:300</p>
<p>Transactions</p> <p>- LogWriteThrough</p>	<p>トランザクションログの更新を RAM にバッファリングするかしないかを指定します。TRUE の場合、リアルタイムでハードディスクに書き込まれます（FlushFrequency=0 と同意）。</p> <p>デフォルト:FALSE</p>

Transactions - FlushFrequency	データベースが RAM からハードディスクに書き込まれる間隔を秒単位で指定します。 デフォルト: 60 推奨設定: デフォルトと同じです。
HotBackup	ホットバックアップに関する設定です。
HotBackup - MirrorTxLog	トランザクションログのミラーリングを実行するかどうかを指定します。 デフォルト: FALSE
HotBackup - TxLogMirrorLocation	トランザクションログミラーのロケーションディレクトリを指定します。メインのトランザクションログとは別のディレクトリにする必要があります。 トランザクションログのミラーリングを実行する場合は、この構成項目で場所を指定します。実用面からして、別の物理ドライブ上のディレクトリを指定することが望まれます。 デフォルト: MIRROR_LOCATION
HotBackup - Device	バックアップに使用するホットバックアップ装置のドライバ。
HotBackup - DeviceDefinition	
HotBackup - DeviceDefinition - Library	使用するドライバファイル名を指定します。 Solaris (Veritas 使用) の場合: libxmsverhb.so Windows の場合: xmswmhb
HotBackup - DeviceDefinition - BackupDestination	バックアップ先のディレクトリを指定します。省略された場合、手動でファイルコピーを行います。
HotBackup - DeviceDefinition - VolMirrorOn	ミラーボリュームが使用可能かを TRUE/FALSE で指定します。

HotBackup - DeviceDefinition - TempVolName	ホットバックアップ実行時に作成される一時ボリューム名を指定します。
HotBackup - DeviceDefinition - TempMountDir	一時ボリュームをマウントするマウントポイントを指定します。
HotBackup - DeviceDefinition - Drive	ドライブ指定子。
HotBackup - DeviceDefinition - Drive - Letter	ソースドライブ名
HotBackup - DeviceDefinition - Drive - MirrorDisk	ミラーリングを実行するディスクドライブ番号。
HotBackup - DeviceDefinition - Drive - MirrorLetter	切断したミラーをマウントするために使用する空きドライブ名。
HotBackup - Timeouts	タイムアウトには3つのレベルがあります。

HotBackup - Timeouts - LevelOne	トランザクションマネージャがトランザクションの生成を許可する時間を、秒単位で指定します。指定された時間の間、新規トランザクションの生成が許可されます。 デフォルト: 600
HotBackup - Timeouts - LevelTwo	トランザクションマネージャが新規トランザクションの生成を待ち合わせる時間を、秒単位で指定します。新規に生成されたトランザクションは、継続中のトランザクションと同様に処理されます。 デフォルト: 1800
HotBackup - Timeouts - LevelThree	新規トランザクションの生成を待ち合わせる時間の総計を、秒単位で指定します。指定された時間の間、LevelOne・LevelTwoのタイムアウト監視を繰り返します。LevelThreeのタイムアウトが発生した場合、ホットバックアップ処理は中断されます。 デフォルト: 7200
<p>Solaris に関する注意事項:</p> <p>現バージョンの XMS では、ホットバックアップ管理のために Veritas のみが使用できます。そのため、NeoXDBRuntime.xml 構成ファイルの更新の<HotBackup>下の<Device>要素の値を「Veritas」（引用符なし）に変更します。例えば、次のようにします。</p> <pre><Device>Veritas</Device></pre> <p>以下に示すのは、Solaris で Veritas 装置を使用するためのデフォルト構成です。</p> <pre><DeviceDefinition NeoID="Veritas"> <Library>libxmsverhb.so</Library> <BackupDestination>BACKUP_DESTINATION</BackupDestination> <VolMirrorOn>true</VolMirrorOn> <TempVolName>xms_tempvol</TempVolName> <TempMountDir>/xms_tempdir</TempMountDir> <NumVolumes>1</NumVolumes> <Volume NeoID="1"> <MirrorVolName>volmir</MirrorVolName> <MirrorPlex>volmir-02</MirrorPlex> </Volume> </DeviceDefinition></pre>	

上記の構成については、それぞれの稼働システムに合わせてユーザーが更新しなければならない要素があります。必要な変更内容とデフォルトの動作を以下にまとめます。

要素: <Library>

必須変更/任意変更/変更不可:変更不可

デフォルト（指定がない場合）:なし

説明:ホットバックアップを処理するためにロードする必要のあるライブラリを XMS に対して指定します。

要素: <BackupDestination>

必須変更/任意変更/変更不可:必須変更

デフォルト（指定がない場合）:なし

説明:XMS ファイルの最終的なバックアップ先パス名またはブランクを指定します。パス名を指定する場合は、絶対パス名にすること、パスが存在すること、neoadmin にそのパスに対する書き込み許可があることが条件になります。例えば、ユーザーが UNIX システムにバックアップ領域として /backup 使用している場合、xms 用のディレクトリを追加してパス名に /backup/xms を指定します。ファイルは、構成ファイルに指定されている一時マウント領域からコピーされます。

<BackupDestination>をブランクにする場合は、バックアップ時にユーザーの介入が必要になります。ユーザーは、ホットバックアップを開始してから、XMS が提供するスクリプトまたは Veritas コマンドを使用し、一時ミラーボリュームからバックアップ装置へのファイルコピーを手動で行う必要があります。

要素: <VolMirrorOn>

必須変更/任意変更/変更不可:任意変更

デフォルト（指定がない場合）:TRUE

説明:XMS データが入っているボリュームのミラーリングを常に行うための構成を Veritas で行っているということを XMS に対して指定します。そのための設定は、ユーザーが XMS の起動前に Veritas コマンドを実行して行います。FALSE に設定するのは、ミラーリングを実行するボリュームが作成されていない、または、有効になっていない場合です。XMS では、ホットバックアップの前にミラーリングが正常に稼働している必要があります。

要素: <TempVolName>

必須変更/任意変更/変更不可:任意変更

デフォルト（指定がない場合）:xms_tempvol

説明:ミラーから切断されたミラープレックス（スナップショット）を一時的にマウントするために使用するボリューム名を指定します。Veritas の下で既に構成されているボリュームと同じ名前を指定することはできません。

要素: <TempMountDir>

必須変更/任意変更/変更不可:任意変更

デフォルト（指定がない場合）:/tmp/xms_tempdir

説明:一時ボリュームにアクセスしてスナップショットをコピーするとき、ファイルシステムをマウントするコピー元のマウントポイントを指定します。そのディレクトリが存在しない場合は作成されます（neoadmin ユーザーの権限で作成できる必要があります）。

要素: <NumVolumes>

必須変更/任意変更/変更不可:必ず 1 以上にする

デフォルト（指定がない場合）:なし

説明:XMS データが入っているミラーボリュームの数。この値は、Veritas のインストールと構成を行うときに設定します。

要素: <Volume NeolD="1">

必須変更/任意変更/変更不可:必須変更

デフォルト（指定がない場合）:なし

説明:ミラーボリュームごとに 1 つ設定します。NeolD は 1 から始まります。

要素: <MirrorVolName>

必須変更/任意変更/変更不可:必須変更

デフォルト（指定がない場合）:なし

説明:Veritas のインストールと構成を行ったときに指定したミラーボリューム名を指定します。

要素: <MirrorPlex>

必須変更/任意変更/変更不可:必須変更

デフォルト（指定がない場合）:なし

説明:Veritas のインストールと構成を行ったときに設定したミラーボリュームを構成する 2 つのミラープレックスのどちらかの名称を指定します。

Windowing	<p>Windowing に関する設定です。</p> <p>32bit 環境の OS (Windows/Linux) で、DB ファイルが物理メモリに展開できないサイズまで拡張された場合に必要になります。また、Windowing が設定されていないファイルは自動拡張機能が無効となり、ファイルサイズの自動拡張は行われません。</p> <p>Solaris では OS が 64bit 処理に対応しているので、Windowing の必要はありません。</p>
Windowing - DataIndex	データインデックスに対する Windowing のサイズをシステムメモリの粒度の倍数で指定します。現在のバージョンでは、0 (OFF) または 2 (ON) が推奨される設定です。
Windowing - TagIndex	タグインデックスに対する Windowing のサイズをシステムメモリの粒度の倍数で指定します。現在のバージョンでは、0 (OFF) または 2 (ON) が推奨される設定です。
Windowing - TagPlusDataIndex	タグ+データインデックスに対する Windowing のサイズをシステムメモリの粒度の倍数で指定します。現在のバージョンでは、0 (OFF) または 2 (ON) が推奨される設定です。
Windowing - MapFile	<p>マップファイルに対する Windowing のサイズをシステムメモリの粒度の倍数で指定します。現在のバージョンでは、0 (OFF) または 6 (ON) が推奨される設定です。</p> <p>注:MapFile のみ、ON の設定値が他のファイルとは異なりますのでご注意ください。</p>
Windowing - DataDictionary	データディクショナリに対する Windowing のサイズをシステムメモリの粒度の倍数で指定します。現在のバージョンでは、0 (OFF) または 2 (ON) が推奨される設定です。
Windowing - CrossReference	クロスリファレンスに対する Windowing のサイズをシステムメモリの粒度の倍数で指定します。現在のバージョンでは、0 (OFF) または 2 (ON) が推奨される設定です。
Windowing - TagDictionary	タグディクショナリに対する Windowing のサイズをシステムメモリの粒度の倍数で指定します。現在のバージョンでは、0 (OFF) または 2 (ON) が推奨される設定です。
FileMaintenance	ファイルメンテナンス処理のパラメータ
FileMaintenance - Autogrowth	<p>XMS が必要に応じてファイルを自動的に拡張できるようにします。</p> <p>デフォルト値 : ON</p>

FileMaintenance - ScheduledStartTime	<p>ファイルメンテナンスのスケジュールを設定する時間（時単位と分単位）。</p> <p>デフォルト:午前 2 時</p> <p>Hour（時単位）:2</p> <p>Minute（分単位）:0</p>
FileMaintenance - MapStoreHighWaterMark	<p>90</p> <p>マップファイルの使用率がこの値を超えると、新たなデータのストアをエラーとして受け付けなくなります。</p>
FileMaintenance - HighAlarmLevel	<p>パーセントで指定します。例えば、90 に設定した場合は、ファイルが 90%の使用率になった時点ですぐに拡張が開始します。パラメータごとに別々の値を設定できます。</p>
FileMaintenance - LowAlarmLevel	<p>パーセントで指定します。例えば、75 に設定した場合は、ファイルが 75%の使用率を超えた後の次のメンテナンススケジュールに合わせて拡張が実行されます。パラメータごとに別々の値を設定できます。</p>
FileMaintenance - FragmentationAlarmLevel	<p>データディクショナリファイル内のフラグメンテーションの許容最大量を指定するアラームパーセンテージ。フラグメンテーションのパーセンテージがアラームパーセンテージを超え、ファイルの使用率が LowAlarmLevel に達すると、Monitor プロセスがサーバーログにエラーメッセージを書き込みます。</p> <p>フラグメンテーション領域はデータストアの際自動的に再利用されますが、フラグメンテーション領域を解消したい場合は、データベースの再構築を行ってください。</p>
FileMaintenance - AutogrowthAmount	<p>各ファイルを拡張する量（メガバイト単位）。</p>
FileMaintenance - MaxFileSize	<p>最大ファイルサイズ（メガバイト単位）。</p>

Windows に関する注意事項:

バックアップの宛先

Windows の場合、ユーザーが BackupDestination を指定すると、ホットバックアップ機能を開始するコマンドを実行するだけで、完全なホットバックアップがユーザーの介入なしで実行されます。

ユーザーが BackupDestination を指定しない場合、ホットバックアップ機能は、ミラーの作成や中断を行うだけで、ユーザーが手動でバックアップを実行するのを待機します。ユーザーが COMPLETE アクションを実行した時点で、ホットバックアップ手順は完了します。

サーバーOS

ホットバックアップを実行するには、Windows の「サーバー」バージョン（つまり、Windows 2000 Server、Windows 2000 Advanced Server、Windows XP、Windows 2003 Server）を実行している必要があります。

Solaris に関する注意事項:

NeoXDBRuntime.xml 構成ファイルの更新によって、XMS ではホットバックアップ管理のために Veritas を使用できるようになっています。そのためには、<HotBackup>の下の<Device>要素の値を「Veritas」（引用符なし）に変更します。例えば、次のようにします。

```
<Device>Veritas</Device>
```

以下に示すのは、Solaris で Veritas 装置を使用するためのデフォルト構成です。

```
<DeviceDefinition NeoID="Veritas">
  <Library>libxmsverhb.so</Library>
  <BackupDestination>BACKUP_DESTINATION</BackupDestination>
  <VolMirrorOn>true</VolMirrorOn>
  <TempVolName>xms_tempvol</TempVolName>
  <TempMountDir>/xms_tempdir</TempMountDir>
  <NumVolumes>1</NumVolumes>
  <Volume NeoID="1">
    <MirrorVolName>volmir</MirrorVolName>
    <MirrorPlex>volmir-02</MirrorPlex>
  </Volume>
</DeviceDefinition>
```

上記の構成については、それぞれの稼働システムに合わせて顧客側で更新しなければならない要素があります。必要な変更内容とデフォルトの動作を以下にまとめます。

要素: <Library>

必須変更/任意変更/変更不可:変更不可

デフォルト（指定がない場合）:なし

説明:ホットバックアップを処理するためにロードする必要のあるライブラリを XMS に対して指定します。

要素: <BackupDestination>

必須変更/任意変更/変更不可:必須変更

デフォルト（指定がない場合）:なし

説明:XMS ファイルの最終的なバックアップ先。これは、パス名にするかブランクにします。パス名にする場合は、絶対パス名にすること、パスが存在すること、neoadmin にそのパスに対する書き込み許可があることが条件になります。例えば、顧客側で UNIX システムにバックアップ領域 (/backup など) を既に設定している場合は、xms 用のディレクトリを追加してパス名を /backup/xms にできます。

<BackupDestination>をブランクにする場合は、バックアップ時にユーザーの介入が必要になります。ユーザーは、ホットバックアップを開始してから、XMS の画面指示に従ってファイルを手動でバックアップ装置にコピーする必要があります。ファイルは、構成ファイルに指定されている一時マウント領域からコピーされます。

要素: <VolMirrorOn>

必須変更/任意変更/変更不可:任意変更

デフォルト（指定がない場合）:TRUE

説明:XMS データが入っているボリュームのミラーリングを常に行うための構成を Veritas で行っているということを XMS に対して指定します。そのための設定は、顧客側が XMS の起動前に Veritas コマンドを実行して行います。FALSE に設定するのは、顧客側がミラーリングを実行するボリュームを既に設定しているものの、まだオンにしていない場合です。XMS では、ホットバックアップの前にミラーリングをオンにする必要があります。

要素: <TempVolName>

必須変更/任意変更/変更不可:任意変更

デフォルト（指定がない場合）:xms_tempvol

説明:ミラーから切断されたミラープレックス（スナップショット）を一時的にマウントするために使用するボリュームの名前。Veritas の下で既に構成されているボリュームと同じ名前を指定することはできません。

要素: <TempMountDir>

必須変更/任意変更/変更不可:任意変更

デフォルト（指定がない場合）:/tmp/xms_tempdir

説明:一時ボリュームにアクセスしてスナップショットをコピーするときのコピー元のマウントポイント（ファイルシステム）。そのディレクトリが存在しない場合は作成されます（neoadmin にその作成権限が必要です）。

要素: <NumVolumes>

必須変更/任意変更/変更不可:必ず 1 以上にする

デフォルト（指定がない場合）:なし

説明:XMS データが入っているミラーボリュームの数。この値は、Veritas のインストールと構成を行うときに設定します。

要素: <Volume NeolD="1">

必須変更/任意変更/変更不可:必須変更

デフォルト（指定がない場合）:なし

説明:ミラーボリュームごとに 1 つ設定します。NeolD は 1 から始まります。

要素: <MirrorVolName>

必須変更/任意変更/変更不可:必須変更

デフォルト（指定がない場合）:なし

説明:Veritas のインストールと構成を行ったときに指定したミラーボリュームの名前。

要素: <MirrorPlex>

必須変更/任意変更/変更不可:必須変更

デフォルト（指定がない場合）:なし

説明:Veritas のインストールと構成を行ったときに設定したミラーボリュームを構成するミラープレックスの名前。

Windowing のパフォーマンスに関する優先順位

NeoCore XMS Windows 版の Windowing:最適なパフォーマンスに到達するまで、以下の優先順位に従い、1 つずつ Windowing を試してください。Windowing は、Microsoft Windows 環境でのみ有効です。

NeoCore XMS Solaris 版の Windowing:Solaris では、サーバーが 64 ビット処理に対応しているので、Windowing の必要はありません。つまり、データベース全体をサーバーの処理領域（理論上は 18 EB）のメモリーに簡単にマップして、OS レベルでファイルのページングを管理できるということです。

インデックスファイルの Windowing の順序は、以下のとおりです。

マップファイル - （最小値 = 6）

データディクショナリ - （最小値 = 2）

重複インデックス（3 種類すべて） - （最小値 = 2）


タグ+データインデックス - （最小値 = 2）


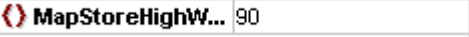
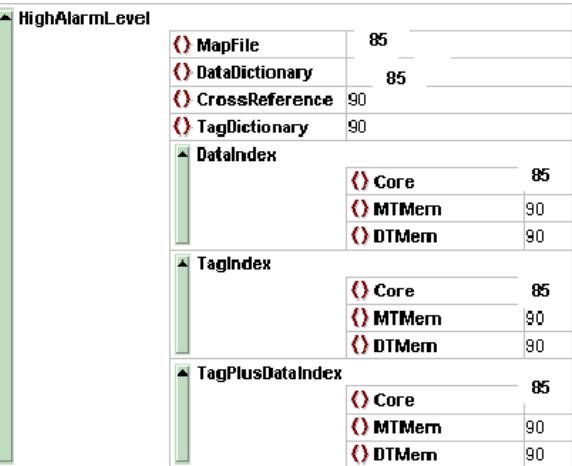
データインデックス - （最小値 = 2）

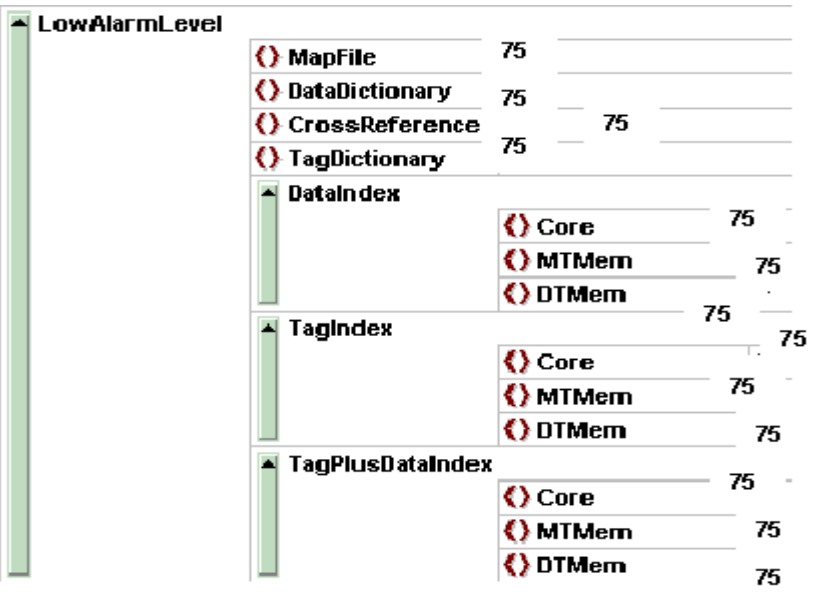
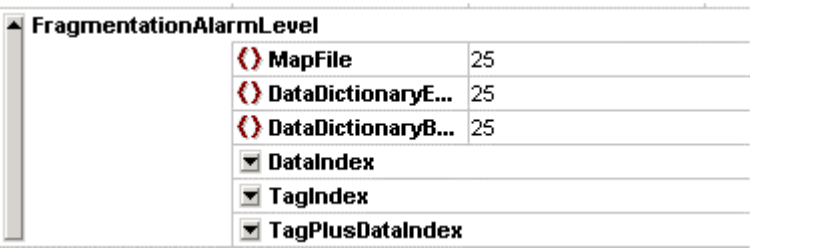
タグインデックスとタグディクショナリ - （最小値 = 2、ただし、通常は Windowing を実行しない）

Windowing	
Windowing - DataIndex	Core 2
説明:DataIndex パラメータでは、データインデックスに対する Windowing のサイズをシステムメモリーの粒度の倍数で指定します。	MTMem 2
	TmpMTMem 0
	DTMem 2
Windowing - TagIndex	Core 2
説明:TagIndex パラメータでは、タグインデックスに対する Windowing のサイズをシステムメモリーの粒度の倍数で指定します。	MTMem 2
	TmpMTMem 0
	DTMem 2
Windowing - TagPlusDataIndex	Core 2
説明:TagPlusData パラメータでは、タグ+データインデックスに対する Windowing のサイズをシステムメモリーの粒度の倍数で指定します。	MTMem 2
	TmpMTMem 0
	DTMem 2

Windowing - MapFile 説明:MapFile パラメータでは、マップファイルに対する Windowing のサイズをシステムメモリーの粒度の倍数で指定します。	MapFile 2
Windowing - DataDictionary 説明:DataDictionary パラメータでは、データディクショナリに対する Windowing のサイズをシステムメモリーの粒度の倍数で指定します。	DataDictionary 2
Windowing - CrossReference 説明:CrossReference パラメータでは、クロスリファレンスに対する Windowing のサイズをシステムメモリーの粒度の倍数で指定します。	CrossReference 2
Windowing - TagDictionary 説明:TagDictionary パラメータでは、タグディクショナリに対する Windowing のサイズをシステムメモリーの粒度の倍数で指定します。	TagDictionary 2

File Maintenance ファイルメンテナンス処理のパラメータ:	
Autogrowth XMS が必要に応じてファイルを自動的に拡張できるようにします。	ON

<p>File Maintenance</p> <p>ScheduledStartTime</p> <p>ファイルメンテナンスのスケジュールを設定する時間（時単位と分単位）。</p> <p>時: 2</p> <p>分: 0</p>	
<p>File Maintenance</p> <p>MapStoreHighWaterMark</p> <p>90</p> <p>マップファイルの使用率がこの値を超えることはありません。</p>	
<p>File Maintenance</p> <p>HighAlarmLevel</p> <p>HighAlarmLevel の各パラメータは、パーセントで指定します。例えば、90 に設定した場合は、ファイルが 90% の使用率になった時点ですぐに拡張が始まります。パラメータごとに別々の値を設定できます。</p>	

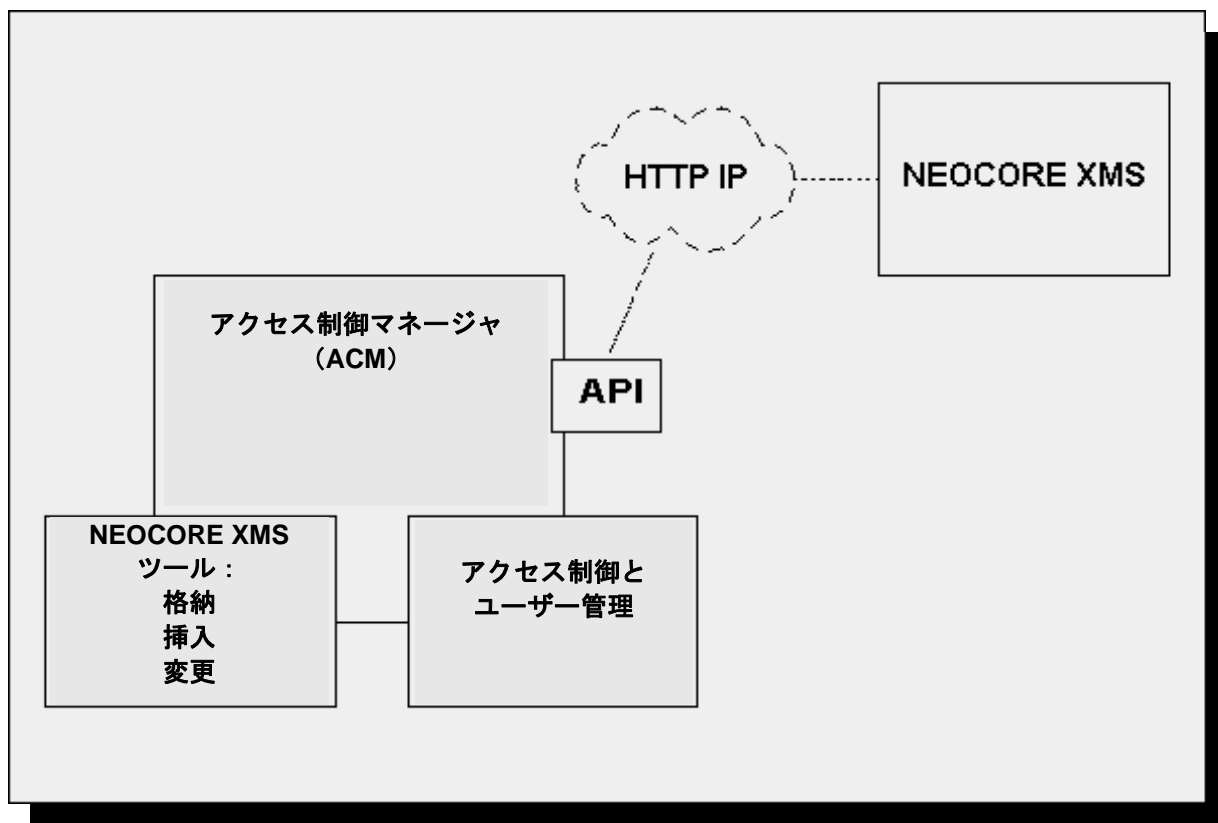
<p>File Maintenance</p> <p>LowAlarmLevel</p> <p>LowAlarmLevel の各パラメータは、パーセントで指定します。例えば、75 に設定した場合は、ファイルが 75%の使用率を超えた後の次のメンテナンススケジュールに合わせて拡張が実行されます。パラメータごとに別々の値を設定できます。</p>	 <p>LowAlarmLevel</p> <table border="1"> <tr><td>MapFile</td><td>75</td></tr> <tr><td>DataDictionary</td><td>75</td></tr> <tr><td>CrossReference</td><td>75</td></tr> <tr><td>TagDictionary</td><td>75</td></tr> </table> <p>DataIndex</p> <table border="1"> <tr><td>Core</td><td>75</td></tr> <tr><td>MTMem</td><td>75</td></tr> <tr><td>DTMem</td><td>75</td></tr> </table> <p>TagIndex</p> <table border="1"> <tr><td>Core</td><td>75</td></tr> <tr><td>MTMem</td><td>75</td></tr> <tr><td>DTMem</td><td>75</td></tr> </table> <p>TagPlusDataIndex</p> <table border="1"> <tr><td>Core</td><td>75</td></tr> <tr><td>MTMem</td><td>75</td></tr> <tr><td>DTMem</td><td>75</td></tr> </table>	MapFile	75	DataDictionary	75	CrossReference	75	TagDictionary	75	Core	75	MTMem	75	DTMem	75	Core	75	MTMem	75	DTMem	75	Core	75	MTMem	75	DTMem	75
MapFile	75																										
DataDictionary	75																										
CrossReference	75																										
TagDictionary	75																										
Core	75																										
MTMem	75																										
DTMem	75																										
Core	75																										
MTMem	75																										
DTMem	75																										
Core	75																										
MTMem	75																										
DTMem	75																										
<p>File Maintenance</p> <p>FragmentationAlarmLevel</p> <p>データディクショナリファイル内のフラグメンテーションの許容最大量を指定するアラームパーセンテージ。フラグメンテーションのパーセンテージがアラームパーセンテージを超え、ファイルが低位上限値に達すると、モニターはオフラインのデフラグが必要であることをログに書き込みます。</p>	 <p>FragmentationAlarmLevel</p> <table border="1"> <tr><td>MapFile</td><td>25</td></tr> <tr><td>DataDictionaryE...</td><td>25</td></tr> <tr><td>DataDictionaryB...</td><td>25</td></tr> </table> <p>DataIndex</p> <p>TagIndex</p> <p>TagPlusDataIndex</p>	MapFile	25	DataDictionaryE...	25	DataDictionaryB...	25																				
MapFile	25																										
DataDictionaryE...	25																										
DataDictionaryB...	25																										

<p>AutogrowthAmount -</p> <p>各ファイルを拡張する量（メガバイト単位）。</p> <p>MapFile パラメータでは、3 の倍数を指定します。</p> <p>MapFile、DataDictionary、CrossReference、TagDictionary、MTMem、DataIndex、TagIndex、TagPlusDataIndex、DTMem については、前の Windowing のセクションを参照してください。</p>	<pre> <AutoGrowthAmount> <MapFile>6</MapFile> <DataDictionary>4</DataDictionary> <CrossReference>2</CrossReference> <TagDictionary>2</TagDictionary> <MTMem> <DataIndex>2</DataIndex> <TagIndex>2</TagIndex> <TagPlusDataIndex>2</TagPlusDataIndex> </MTMem> <DTMem> <DataIndex>5</DataIndex> <TagIndex>5</TagIndex> <TagPlusDataIndex>5</TagPlusDataIndex> </DTMem> </AutoGrowthAmount> </pre>
<p>MaxFileSize -</p> <p>最大ファイルサイズ（メガバイト単位）。</p> <p>MapFile、DataDictionary、CrossReference、TagDictionary、MTMem、DataIndex、TagIndex、TagPlusDataIndex、DTMem については、前の Windowing のセクションを参照してください。</p>	<pre> <MaxFileSize> <MapFile>100</MapFile> <DataDictionary>100</DataDictionary> <CrossReference>50</CrossReference> <TagDictionary>20</TagDictionary> <Core> <DataIndex>100</DataIndex> <TagIndex>100</TagIndex> <TagPlusDataIndex>100</TagPlusDataIndex> </Core> <MTMem> <DataIndex>25</DataIndex> <TagIndex>25</TagIndex> <TagPlusDataIndex>25</TagPlusDataIndex> </MTMem> <DTMem> <DataIndex>100</DataIndex> <TagIndex>100</TagIndex> <TagPlusDataIndex>100</TagPlusDataIndex> </DTMem> </MaxFileSize> </pre>

アクセス制御とユーザー管理

NeoCore XMS でアクセス制御（AC）を使用すると、XML 情報へのアクセスをノードレベルで自由自在に制御できます。そのためには、ルールを使用します。すべてのノードを対象とする単純なルールもあれば、該当文書内の情報に基づく条件値に合致するノードだけを対象とする複雑なルールもあります。

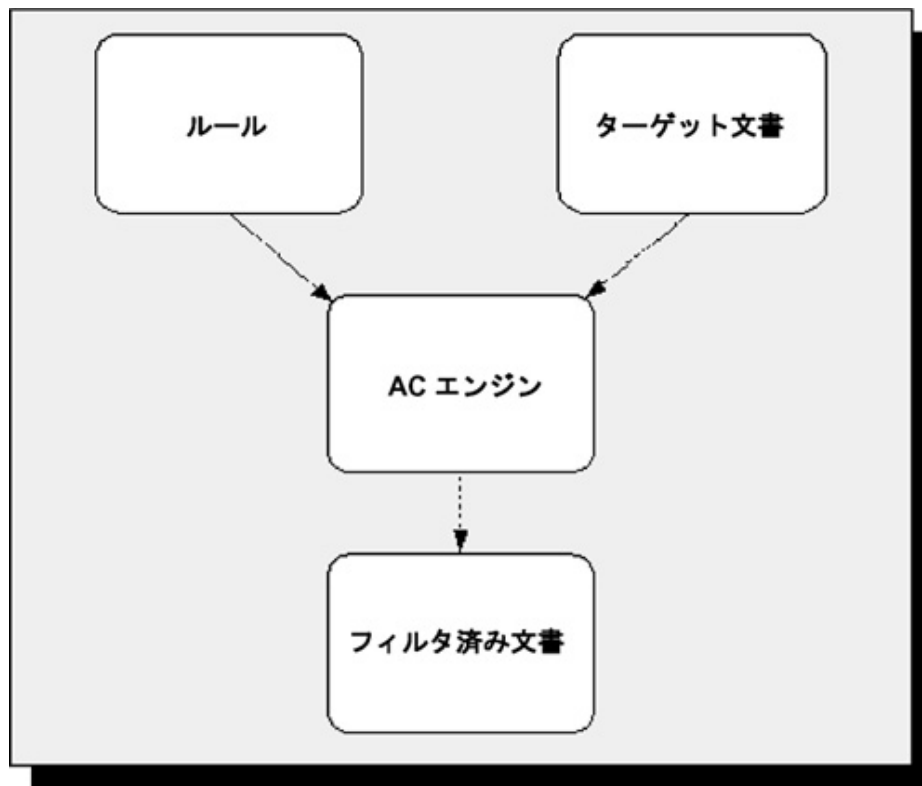
アクセス制御（AC）のシステム図



アクセス制御の概略図

アクセス制御（AC）のフィルタリング

NeoCore XMS のアクセス制御とは、制限項目を対象のユーザーやグループから見えないようにするために情報をフィルタリングするメカニズムのことを指します。隠された項目はデータベース内に存在しますが、ユーザーの観点からはその項目が存在しないかのように処理されるため、その種の項目はクエリーの結果にも現れません。隠す項目を決定するため、AC エンジンには制御文書（ルール）に基づくパターンマッチングを使用します。



ACエンジン

アクセス制御の有効化

アクセス制御は、データベースの作成時に有効化されます。デフォルトのアクセス制御文書を作成することにした場合は、**DefaultAC.xml** ファイルにデフォルトのルールが含まれています。アクセス制御を変更するには、この文書を編集します。編集の方法は、格納済みの他の文書の場合と同じです。

アクセス制御の呼び出し

アクセス制御ルールは、ユーザー、グループ、クエリータイプに基づいて適用されます。1 つのクエリータイプに適用することも、すべてのクエリータイプに適用することもできます。

ユーザー/グループの AC を正しく呼び出すには、ユーザーまたはアプリケーションが NeoCore XMS インスタンスにログインしており、有効なセッション ID (SID) を持っていることが前提になります。クエリーごとにこの SID を NeoServer に渡す必要があるからです。SID は NULL でもかまいません。この場合、AC は anonymous ログインを想定して、guest ルールを適用します。ただし、**NeoXDBRuntime.xml** の **AllowAnonymousUser** パラメータが **TRUE** に設定されていることが条件になります。

AC を呼び出すと、SID に基づいて、対応するユーザーセッションが検索されます。そのセッションがタイムアウトになっている場合は、エラーが返されます。そうでない場合は、ユーザー、グループ、クエリータイプに基づいて、ユーザーのルール設定を取り出すためのクエリーが生成されます。そのクエリーに該

当するルールだけが実行されます。同じノードについてグループルールとユーザールールの両方が存在する場合は、ユーザールールのほうが優先し、グループルールは無視されます。

このプロセスでは、**Administrator** ユーザーと **guest** ユーザーが必ず作成され、それぞれに関連するグループも作成されます。また、**Administrator** アカountのパスワードの作成が必要になります。

NeoCore XMS の AC ルールの動作

一般に、AC ルールは、タグ構造や、ルールの XPath と現在のクエリーの XPath の深さに従って、文書やフラグメントに適用されます。この場合の深さとは、XPath の収束的な深さのことです。ルールが該当するかどうかを判別するために、ルールの XPath とクエリーの XPath がそれぞれの深さに従って比較されます。

以下のルールは、QUERY、INSERT、MODIFY、DELETE、COPYの各操作に適用されます。詳細については、「NeoCore XMS トランザクションとコマンド」²を参照してください。

1. クエリーの深さまでクエリーとルールが一致する場合は、そのルールが適用されます。
2. ルールの深さがクエリーの深さよりも深い場合は、結果の XML 文書でルールに該当する要素が隠されます。
3. クエリーの深さがルールの深さより深く、ルールの深さまでルールとクエリーが一致する場合は、クエリーの結果全体が隠されます。
4. ルールの深さまでルールとクエリーの XPath が一致しない場合、そのルールは適用されません。
5. show コマンドに述部のないノードは無効です。
6. 述部のある show コマンドを使用したノードの場合は、述部の基準を満たすノードを除き、そのタイプの全ノードが隠されます。
7. hide コマンドに述部のないノードの場合は、そのタイプの全ノードが隠されます。
8. 述部のある hide コマンドを使用したノードの場合は、述部の基準を満たす全ノードが隠されます。

NeoCore XMS の場合、STORE コマンドは文書のルートノードに基づきます。STORE に対する AC ルールの動作は、以下のとおりです。

1. 文書のルートレベルで適用されるルールがない場合は、すべての文書タイプを格納できます。
2. 特定文書タイプに対して hide ルールを設定すると、その文書タイプをルートレベルで格納できなくなります。
3. 特定文書タイプに対して show ルールを設定すると、その文書タイプだけを格納できるようになります。特定ユーザーに対する show ルールがある場合は、そのユーザーが格納を許可されている文書タイプごとに show ルールが必要になります。
4. 特定文書タイプに該当する show ルールがない場合は、hide ルールで隠されていないすべての文書タイプを格納できます。
5. NeoCore XMS の STORE では、ルールの述部は無視されます。
6. NeoCore XMS の STORE では、ルートノードよりも深いルールは無視されます。

2

ルール構文

ルールは、XPath 述部か完全修飾 XPath ステートメントのいずれかの形式にします。以下のようにルールを記述してください。

```
hide ルールを作成するための構文は、以下のとおりです。  
/ND/CATALOG/CD/PRICE>"8.00"
```

例えば、次のようにします。

```
タイトルなし  
<PreserveRoundtripEncoding>FALSE</PreserveRoundtripEncoding>
```

TRUE または FALSE（デフォルトは TRUE）

クエリで文書を検索するときに、Unicode 以外のエンコードを指定した文書内の特定の文字を別の（同等の）文字にマッピングできるかどうかを指定します。

このオプションを TRUE にした場合、Unicode 文書内に私用領域（E000 から F8FF）の文字が含まれていれば、その文字を別の文字に変換できます。これらの文字の変換を禁止するには、このオプションを FALSE に設定します。

NeoCore XMS の場合、XPath ステートメントに対するルールは、ルール適用対象ノードのすべての子ノードにも適用されます。完全修飾 XPath 式には、独自の XPath 述部を含めることができます。この述部は、XPath のルールに適合していなければならない、前後を括弧で囲む必要があります。

XPath の詳細については、『NeoCore XMS System Programming Guide』の「基本ロケーションパス」を参照してください。

AC エンジンの仕組みを理解するため、書籍を記述した文書を例として取り上げましょう。ユーザー **guest** が「**Space Sciences**」カテゴリの書籍以外は見られないように制限する制御文書（ルール）を作成します。

AC ルールの例

XML 文書:

```
<Book>
<Title>Oceans of Europa</Title>
<Category>Space Sciences</Category>
<Year>1998</Year>
<Pub-Info>
  <Author>
    <First>Roger</First>
    <Last>Wu</Last>
  </Author>
  <Publisher>
    <Name>Acme Media</Name>
    <Phone>404-717-5555</Phone>
  </Publisher>
</Pub-Info>
<Subject>Discussion on the potential for liquid water and subsequent
  life forms on Europa.</Subject>
</Book>
```

制御文書:

```
<NeoCore>
  <AC user="guest" type="All">
    <Book>
      <hide>Category!="Space Sciences"</hide>
    </Book>
  </AC>
</NeoCore>
```

この要件を満たす書籍がデータベース内に 1 冊しかない場合、クエリー/ND/Book は次の結果を返します。

```
<Query Results>
<Book>
  <Title>Oceans of Europa</Title>
  <Category>Space Sciences</Category>
  <Year>1998</Year>
  <Subject>Discussion on the potential for liquid water and
    subsequent life forms on Europa.</Subject>
</Book>
</Query Results>
```

WEBDAV

WebDAV の概要

WebDAV とは、「Web-based Distributed Authoring and Versioning」の略語です。HTTP プロトコルに対する一連の拡張機能であり、リモートの Web サーバー上のファイルを編集したり管理したりする作業を複数のユーザーが共同で行うことが可能になります。要するに、Web を読み取り専用のメディアから読み取り/書き込み可能なメディアに変換するための機能です。

WebDAV は、オープンスタンダードに基づいているので、Web ブラウザと同じようにプラットフォームやソフトウェアを選びません。

NeoCore XMS の WebDAV 実装は、Apache の Jakarta オープンソースプロジェクトである「Slide」に由来しています。Slide の詳細については、<http://jakarta.apache.org/slide/index.html> を参照してください。Slide は Java サーブレットであり、通常は Apache の Tomcat がホストになります。しかし、XMS では、Slide のホストとして独自のサーブレットエンジンを用意しており、インストールと構成の作業を簡略化し、パフォーマンスを改善し、システム要件を緩和しています。

WebDAV の格納と検索以外の重要な機能としては、認証、コレクション、Delta V バージョン管理、チェックイン/チェックアウト、ユーザープロパティ、許可、文書ロックなどがあります。これらの機能については、後から詳しく説明します。また、WebDAV のドラフト 12 ではアクセス制御プロトコルが実装されています。さらに、Slide 文書だけでなくあらゆる XMS 文書について、XMS のユニークなパターンベースのアクセス制御を使用することも可能です。WebDAV バインドと DASL (WebDAV 検索) は現在実装されていません。

WebDAV の仕組み

読み取り/書き込み許可を持つユーザーは、インターネットなどのネットワークに存在する共有サーバー上の WebDAV フォルダに自分のデスクトップを接続できます。WebDAV は、HTTP プロトコルの拡張機能なので、サーバーがどのオペレーティングシステムを実行しているかは問題になりません。ユーザーは、通常のドライブやフォルダに存在するファイルにアクセスする場合とまったく同じ要領で WebDAV フォルダのファイルにアクセスできます。

WebDAV を使用すれば、電子メールでファイルをやり取りする必要がなくなります。電子メールでのやり取りは、時間がかかり、エラーが起きやすく、安全面で問題があり、追跡管理が困難です。チェックイン/チェックアウトやバージョン管理の機能もありませんし、多数の文書をやり取りする場合は、すぐに管理不能な状態になってしまいます。VPN 経由での作業にも限界があります。基本的なファイルアクセス機能しかなく、特別なソフトウェアをインストールしていないデスクトップから文書にアクセスすることは不可能です。その点、WebDAV 対応のオペレーティングシステムとアプリケーションを使用すれば、場所やプラットフォームに関係なく、メンバーが散在するチームの共同作業とコンテンツ管理が可能になります。

WebDAV のクライアント

WebDAV プロトコルをサポートするプログラムが増えています。他の多くのプロトコルの場合と同じように、WebDAV の場合も、フルサポートがなければ仕様準拠にならないわけではありません。したがって、機能性はクライアントによって異なります。Windows は、Windows エクスプローラまたは Internet Explorer によって WebDAV サーバーへの接続をネイティブでサポートしています。Mac OS/X にも同様の機能があります。実際の操作は、他のフォルダにアクセスする操作とまったく同じであり、要は、ローカルのハードディスクに文書を格納していないというだけの違いです。文書の格納場所は WebDAV サーバーであり、そのサーバーは世界のどの場所に置いておかまいません。

Microsoft Office、Adobe Designer、Altova の xmlspy、OpenOffice、Documentum などのツールも、程度の差こそあれ、いずれも WebDAV をサポートしています。さらに、オープンソースの DAV Explorer や Independent DAV など、WebDAV 専用のクライアントもあります。一般に、これらのクライアントのほうが機能は豊富です。

注: Windows XP には、WebDAV 用の「リダイレクタ」が組み込まれています。要するに、あらゆる Windows プログラムが WebDAV 対応のサーバーとの間で文書の保存や検索を自動的に実行できるようにするための機能です。しかし、このリダイレクタにはいくつかのバグがあり、XMS をはじめとするほとんどの WebDAV サーバーに対して正しく機能しません。つまり、ファイルを指定して開くことはできても、保存ができません。この問題を回避するには、Windows Explorer を使用してファイルをローカルディレクトリに保存してから、XMS の WebDAV フォルダにコピーしてください。

API アクセス

API から WebDAV 機能にアクセスする場合は、<http://jakarta.apache.org/slide/index.html> に用意されている Apache Jakarta の Slide クライアント API を使用できます。他の言語のための API もインターネット経由で入手できる可能性があります。

XMS WebDAV への接続

XMS WebDAV のための主な URL は、<http://localhost:7701/slide> です（localhost は、作業対象のマシンまたは WebDAV サーブレットのホストであるマシンの IP またはマシン名です）。この URL は、Internet Explorer などのほとんどの WebDAV クライアントから入力できます。

Windows エクスプローラからの接続

「スタート」メニューから「マイネットワーク」を選択し、「ネットワークプレースを追加する」を選択し、「次へ」をクリックし、「別のネットワークの場所を選択」を選択し、URL として **<http://localhost:7701/slide>** を入力し（localhost は、作業対象のマシンまたは WebDAV サーブレットのホストであるマシンの IP またはマシン名）、「次へ」をクリックし、XMS WebDAV などの接続名を入力し、「次へ」をクリックし、「完了」をクリックします。XMS WebDAV ストアのフォルダが開きます。

Windows と MacOS に接続するための詳細や、コマンドラインクライアントを使用するための詳細については、**<http://jakarta.apache.org/slide/>** を参照してください。

認証

認証がオン（インストール後のデフォルト設定）になっている場合、WebDAV サーバーでは、ユーザー名/パスワードの入力が必要です。Slide のための構成項目は、c:¥NeoCore¥xms¥Domain.xml ファイルにあります。

Slide には「役割」という概念があるのに対し、XMS にはその概念がありません。したがって、XMS の通常のメカニズムでは Slide のユーザーを認証できません。Domain.xml ファイルでユーザーを作成し、保守する必要があるのはそのためです。このファイルはクリアテキストなので、ユーザー名/パスワードを安全に保管するためにファイルを保護するようにしてください。

以下にユーザーエントリのサンプルを示します。新しいユーザーを追加するときには、このサンプルをコピー/編集できます。

```
<objectnode classname="org.apache.slide.structure.SubjectNode"
  uri="/users/john">
<revision>
<property namespace="http://jakarta.apache.org/slide/"
  name="password">john</property>
</revision>
```

許可

特定のリソースやコレクションに対するアクセス制御を設定することを「許可」といいます。

Apache Jakarta Slide Web サイトからの抜粋（日本語訳は本書）

WebDAV によるプリンシパルリソース（ユーザー、役割/グループ）の作成と保守

プリンシパルリソース（ユーザー、役割/グループ）の作成と保守は、WebDAV/ACL 仕様の守備範囲を超えていますが、Slide では、SlideRealm（`wrappers.catalina.SlideRealm`）がアクティブになっている限り、MKCOL メソッドの独自の拡張機能によってその作成と保守が可能になっています。

以下のいずれかのパラメータで指定した位置に MKCOL で作成したリソースは、自動的にプリンシパルになります（つまり、`DAV:resourcetype` プロパティには、自動的に `DAV:principal` 要素が組み込まれます）。

`userspath` (xpath: `/slide/namespace/configuration/userspath`)

`groupspath` (xpath: `/slide/namespace/configuration/groupspath`)

`rolespath` (xpath: `/slide/namespace/configuration/rolespath`)

パスワードまたはユーザーを設定するには、`password` プロパティを設定して PROPPATCH 要求を実行します（このプロパティの名前空間は、`http://jakarta.apache.org/slide/` です）。

ユーザーを役割/グループに関連付けるには、`DAV:group-member-set` プロパティを設定して PROPPATCH 要求を実行します。

注：ユーザーを役割/グループに追加するには、`DAV:group-member-set` プロパティの値として、追加するユーザーだけでなく、役割/グループの全メンバーのリストを指定する必要があります。

文書の格納、検索、管理

文書またはファイルを XMS WebDAV リポジトリに格納すると、サーブレットは、Base64 エンコードを使用してその文書を各「チャンク」にエンコードして、XML 文書の中にカプセル化します。これは SQL の BLOB に似ていますが、XMS の場合はさらに XML の豊富なメタデータが組み込まれます。バージョン、プロパティ、許可などの情報はすべて、正しくモデル化された XML に格納され、通常の XMS メカニズムによるアクセスが可能になります。

格納したファイルに XML、XSD、XSLT のいずれかの拡張子が付いている場合は、XMS にも「ネイティブに」格納されるので、通常の XMS インターフェースによってその内容全体にアクセスできるようになります。これは非常に便利です。例えば、XML 注文フォルダ全体を WebDAV フォルダにドラッグアンドドロップすれば、XMS を使用するアプリケーション（Insight など）からすぐにそのフォルダを利用できるようになります。つまり、その注文に含まれているメタデータやデータに対してクエリーを実行できるということです。

ただし、注意が必要です。XMS のクエリーメカニズムによって XML 文書の内容を変更した場合、その変更はネイティブの XML 文書だけに反映され、元のエンコード済みの WebDAV ファイルには反映されません。つまり、エンコード済みの文書とネイティブの文書は同期しないということです。ただし、通常の使用法モデルでは、基本的にそのような同期は必要ありません。この動作の唯一の例外として、XML 文書である WebDAV 文書を削除した場合には、ネイティブの文書も削除されます。

制限事項:どんなサイズのファイルでも保存できますが、検索するファイルについては、Java 仮想マシンのメモリーヒープ内に収まる必要があります。これが XMS の HTTP サポートに伴う制限事項です。インストール時のデフォルトでは、8~10 MB よりも大きなファイルの検索は制限される場合があります。C:\#NeoCore\#neoxml\#config\#NeoServer.xml ファイルに定義されている最大ヒープサイズを増やすには、以下の要素を変更します。

```
<Option NeoID="MaxHeapSize" cloneable="values">-Xmx64M</Option>
```

例えば、ヒープサイズを-Xmx128M などに増やしてください。ただし、それでも制限は残ります。つまり、システムで設定されている仮想 RAM の量から他のすべてのシステムオーバーヘッドを差し引いた量を超えることはできません。この制限事項は、将来のリリースの NeoCore XMS で除去されます。

コレクション

WebDAV では、フォルダのことをコレクションといいます。コレクションの作成方法は、通常のファイルフォルダの作成方法と同じです。また、ファイルやフォルダの移動、コピー、削除も、ローカルファイルシステムの場合と同様です。WebDAV ストア（データベース）を最初に作成したときに、いくつかのコレクションが自動的に作成されます。そのうちの 1 つは、バージョン履歴フォルダです。このフォルダにアクセスすれば、旧バージョンのファイルや文書を取り出せます。

プロパティの設定

プロパティとは、基本的にシステムまたはユーザーが作成する名前/値のペアです。プロパティを設定するには、この機能をサポートするクライアント（Independent DAV や DAVExplorer など）が必要です。作成できるプロパティの数は無限です。Author、Department、Classification など、自分の思い通りのプロパティを作成できます。XMS では、プロパティの検索も可能です（WebDAV では、検索用の拡張機能である DASL の開発を進めています。NeoCore XMS は、現時点で DASL をサポートしていませんが、同等の機能を XQuery のメカニズムで実行できます）。

「catalog.xml」の DocID を抽出するためのクエリーの一例は、次のようになります。

```
/ND[SlideDoc//Property/Name='displayname' and  
SlideDoc//Property/Value='catalog.xml']/MetaData/DocID
```

すべての WebDAV 文書のユーザープロパティ「Author」を抽出するためのクエリーの一例は、次のようになります。

```
/ND/SlideDoc/Version/Property[Name='Author']
```

ロック

クライアントがロックをサポートしていれば、明示的にロックを実行できます。排他ロックと共有ロックが可能です。文書にロックを設定すると、他のユーザーはその文書に対する変更を実行できませんが、読み取り専用モードで文書を検索することはできます。HTTP は、疎結合のセッションレスのメカニズムであり、クライアントが接続を解除することもしばしばあるので、ロックを作成するときにはタイムアウト値を設定できます。その場合は、ロックの期限が切れた時点で、リソースは解放されます。

バージョン管理

バージョン管理は、特定のリソースに対して明示的に設定する必要があります。バージョン管理を設定しておくと、文書をチェックアウトし（その時点で文書はロックされます）、変更を加えてから WebDAV サーバーに戻した時点で、新しいバージョンが自動的に作成されます。この強力な機能をまだサポートしていないクライアントも少なくありませんが、無償で入手できる DAVExplorer では、文書のバージョン管理、バージョン履歴の表示、システム内に格納されている任意のバージョンの検索が可能です。

バージョン管理の詳細については、<http://jakarta.apache.org/slide/howto-deltav.html> を参照してください。

WebDAV XML スキーマ

Xpiori の WebDAV 実装の背後に正式なスキーマは存在していません。以下に示すのは、許可やプロパティの多くを削除したサンプル文書です。この文書を見れば、現時点の基礎構造をある程度把握できますが、この構造は将来のリリースで変更される可能性もあります。

もちろん、構造の変更ではなく、構造の追加を行うようにして、プロパティの検索や操作のために顧客側で作成したクエリーをそのまま使用できるように最大限努力するつもりですが、それでも、XML の検索や操作を直接実行する代わりに、基礎構造を隠すために WebDAV クライアント API によって検索や操作を実行することをお勧めします。

```
<SlideDoc>
  <Common>
    <URI>/roles</URI>
    <Binding>
      <Name>roles</Name>
      <ParentURI>/</ParentURI>
    </Binding>
    <Permission>
      <SubjectURI>unauthenticated</SubjectURI>
      <ObjectURI>/roles</ObjectURI>
      <ActionURI>all</ActionURI>
      <isInheritable>true</isInheritable>
      <isNegative>true</isNegative>
      <Succession>0</Succession>
    </Permission>
    <ClassName>org.apache.slide.structure.SubjectNode</ClassName>
    <URIID>115</URIID>
  </Common>
  <Version>
    <Revision>1.0</Revision>
    <Property>
      <Namespace>DAV:</Namespace>
      <Name>creationdate</Name>
      <Value>2004-07-28T17:11:50Z</Value>
      <Type></Type>
      <isProtected>true</isProtected>
    </Property>
    <Branch>main</Branch>
  </Version>
</SlideDoc>
```

WEBDAV の参考資料

WebDAV は、HTTP に対する複雑で強力な拡張機能です。このガイドでは、用意されている機能の一部をごく簡単に挙げただけであり、包括的な解説をしているわけではありません。詳細については、<http://www.webdav.org> を参照してください。

用語集

関連付け: 関連付けは、コアインデックスの検索によって返される値です。NeoServer では、関連付けの値は、マップオフセットの一覧へのポインタとして解釈されます。またはそれ自体がマップオフセットとなることも可能です。

領域自動拡張: データベースの稼働中に、要求を処理しながら、NeoServer ファイルのサイズを拡張する機能。

バッファ: データベースの初期化時に、メモリーの一部が選択され、さまざまな長さのチャンク群に分割されます。メモリーのこの部分のことをバッファプールといいます。各プールは同じサイズのチャンクに対応します。例えば、サイズが 1K のチャンクが集まったプールが 1 つ、2K のチャンクが集まったプールが 1 つ、といった具合になります。1 つのチャンクがバッファであり、処理の必要に応じてスレッドを実行することによって、バッファを割り当てます。バッファが自身のプールに戻されることを確認するのはスレッドの役目です。

CDATA: マークアップなど、XML パーサーによって無視される文字データや、インデックススペースの検索を必要としない文字データを埋め込む方法。

チェックポイント: データベース内の復元ポイントを取り込むログエントリ。チェックポイントでは、すべての未解決トランザクションのトランザクション ID がトランザクションログに記録されます。データファイルはすべてディスクにフラッシュされます。起動時には、最後のチェックポイントに含まれる情報に基づいてログが適用されます。システムは、実行時構成ファイルに定義されている間隔でチェックポイントを設定します。

コアインデックス: コアインデックスは、NeoData の構造に基づいています。各コアインデックスは、固有の文字列を表します。XMS のコアインデックスには、タグとフラット形式のタグ、データのみ、タグ+データの組み合わせという 3 種類があります。

count クエリー: データを検索して、ターゲットに一致するノードの数を検出し、ターゲットセット内のノードの数を返します。

countnotacid クエリー: ターゲットセット内のノードの数を返します。countnotacid は、クエリー内で count コマンドの代わりに使用できます。

クロスリファレンスファイル: クロスリファレンスファイルは、データディクショナリに対するポインタのファイルベースの配列です。マップエントリがデータポインタを含む場合、そのポインタはクロスリファレンスファイルのオフセットをポイントし、そのオフセットはデータディクショナリをポイントします。2.8 よりも前のリリースでは、マップエントリはデータディクショナリを直接ポイントしていました。クロスリファレンスファイルは、データディクショナリの回復を目的として導入されました。クロスリファレンスを使用すると、データディクショナリで変更しなければならないポインタは 1 つだけで済みます。

データディクショナリ: データモデル内のデータオブジェクトまたは項目の記述の集まり。これらを参照するプログラマや他のユーザーにとって便利な機能です。

dataquery コマンド: コンテキストのデータを抽出できるだけでなく、XPath のワイルドカード検索の場合よりも詳細な情報を取得できます。

ディクショナリファイル: 格納されている XML 文書に現れる固有の文字列を内容とするファイル。XMS には、固有のフラット形式のタグを含むディクショナリファイルと、固有のデータ項目を含むディクショナリファイルがあります。

DPP: Digital Pattern Processing (エンジン)。

DTArray: DupTree Array (重複ツリー配列)。SubTree は、DTArray で構成されます。DTArray は固定サイズで、SubTree 構築時に割り当てられる最小メモリーブロックを表します。DTArray の内容は、他の DTArray への関連付けまたは参照インデックスのいずれかになる要素です。DTArray 内の要素の数は、16、32、64、128、256 のいずれかです。要素の数は、データベース構成変数です。各種インデックス (タグ、データ、タグ+データ) では、さまざまなサイズの DTArray を使用できます。1 つのインデックスタイプ内の DTArray はすべて同じサイズになります。

DTMem: Duplicate Tree Memory (重複ツリーメモリー)。これはサブツリー配列の割り当て元になるメモリー領域です。

重複インデックス: 重複インデックスは、固有の文字列が出現するたびにエントリを格納します。エントリは、その文字列を含むマップオフセットまたはその文字列を参照するマップオフセットに相当します。

flat クエリ: データを検索して、ターゲットに一致するノードを検出し、ターゲットノードの各出現部分に関するフラット形式の行を返します。

FLWR: FOR-LET-WHERE-RETURN XQuery キーワードの略語。FLWR 式は、SQL の SELECT columnlist FROM tablename WHERE criteria ステートメントに相当します。

アイコン: コア DPP 処理の基本単位。任意のデータ項目をマッピングできる固定フィールド数量です。アイコンジェネレータ (基本的には、特定の関連する多項式を持つ線形シフトレジスタ) によって生成されます。

挿入: データベースからターゲット要素のセットを選択し、指定の XML フラグメントをセット内の各ノードの後に挿入します。挿入は兄弟レベルでのみ行われます。つまり、挿入される各フラグメントはターゲットセット内のノードの兄弟になります。ターゲットセットのカーディナリティに基づいて、1 度の挿入操作で、同じフラグメントをデータベースに複数回挿入できます。

ロック: ロックとは、リソースを予約することです。ロックの種類には、共有ロック、更新ロック、書き込みロックがあります。共有ロックは、読み取りロックともいい、同じリソースに対して別の共有ロックを設定できます。更新ロックは更新目的での読み取りを意味します。更新ロックを保持するリソースに対して、共有ロックを設定できますが、更新ロックや書き込みロックは設定できません。書き込みロックは、排他ロックともいい、その同じリソースに対してその他のロックを許可しません。

マップエントリ: マップエントリとは、マップファイルにあるフラット形式の行の格納済みのイメージです。マップエントリは固定長です。マップエントリには、行のタグ部分を指すポインタ、文書全体のコンテキストにおける行の構造を説明する一連の数値、および、次のいずれかが含まれます。行のデータ部分のクロスリファレンスエントリを指すポインタ、または、行のデータ部分の文字列値 (長さが 6 バイト以下の場合)、または、行の数値の BCD 表現 (長さが 12 バイト以下の数値の場合)。

マップファイル: 固定長のエントリを含むファイル。各エントリは、XML のフラット形式の行を表します。

メタデータ: 別のファイルまたはデータベースの属性、構造、処理、変更に関する情報を格納するファイルまたはデータベースのことをいいます。データの内容や品質など、データのあらゆる特性を記述できます。

変更: 変更操作では、要素や属性に関連付けられたデータ値を変更します。矛盾が生じないように変更するには、要素にデータ値が関連付けられている必要があります。テキストデータを持たないノードを変更すると、動作しない場合があります。挿入と同様に、変更は一連のターゲットノード上で操作されます。そのため、ターゲットセットのカーディナリティに基づいて、同じ変更がデータベースで複数回実行されることもあります。

MTArray: Multi-Tree Array（マルチツリー配列）。MTBase オブジェクトは、MTArray で構成されます。MTArray は固定サイズで、MTBase 構築時に割り当てられる最小メモリーブロックを表します。MTArray は、4 つまたは 8 つの MapBlock で構成されます。この MapBlock 数は、データベース構成変数です。インデックスタイプ（タグ、データ、タグ+データ）ごとに、異なったサイズ（MapBlock が 4 つまたは 8 つ）の MTArray を使用できます。1 つのインデックスタイプの中では、すべての MTArray が同じサイズになります。

MTBase: Multi-Tree Base（マルチツリーベース）オブジェクト。ベースツリーオブジェクトは、コアインデックスから生成されます。MTBase オブジェクトは、MTArray で構成され、レベル 1 またはレベル 2 のツリーとして存在します。各リーフ MTArray は、MapBlock で構成されます。各 MapBlock は、SubTree への関連付けまたは参照のいずれかを格納し、特定の MapFile を参照します。

MTMem: Multi-Tree Memory（マルチツリーメモリー）。これは、マルチツリーベース配列の割り当て元になるメモリー領域です。

相互排除: 厳格な用法ではありませんが、データ構造や構造体の一部に対する非トランザクションロックのことをいいます。このロックは、その構造体の更新が完了するとすぐに解除されます。また、重要なセクションに対するロックを指す場合もあります。

プレフィックスファイル: プレフィックスファイルは、XML 文書と共に格納するように指定できます。プレフィックスファイルは、XML 文書に追加されます。

カンタム: コアインデックスの基本単位。一連の確認ビット、一連のフラグ、1 つの関連付けを含む、固定長フィールドです。

セッション: NeoServer の起動時にプール内で作成されるオブジェクト。ユーザーがログインするか、anonymous 要求を行うと、セッションがプールから割り当てられます。前者の場合、ユーザーがログアウトするか、タイムアウトになるまでセッションは継続します。後者の場合、セッションは、要求の存続期間が終わるか、セッションやトランザクションがタイムアウトになるまで継続します。

格納: NeoCore XMS に新しい XML 文書を生成します。文書を格納すると、格納タイムスタンプ、固有の文書 ID、変更タイムスタンプ（格納タイムスタンプと同じ内容）などのメタデータがその文書に関連付けられます。また任意で、ソースファイル名、スキーマ参照、プレフィックスファイル名もメタデータに格納できます。

トランザクション: データベースの作業単位。1 つのトランザクションに含まれるすべてのデータベース要求は、すべて成功するか、すべて失敗するかのどちらかになります。XMS には、明示的トランザクションと暗黙のトランザクションの 2 種類があります。明示的トランザクションは、トランザクションの開始と終了をユーザーが明示的に行うものです。暗黙のトランザクションは、明示的な開始に関連付けられてい

ないコマンドに割り当てられます。したがって、暗黙のトランザクションでは、1 つのデータベースコマンドが 1 つのトランザクションに対応しています。

Tree コマンド: アプリケーション内から所定のノードの構造を検出できます。このコマンドは、アプリケーション側が、渡された XML の内容を知らない場合に使用できます。

Tree クエリ: データを検索して、ターゲットよりも 1 レベル下のノードを検出します。

ウィンドウ: メモリーマップファイル上のビュー。ウィンドウでは、ファイル全体をメモリーにマッピングできます。ファイルが大きすぎて入りきらない場合は、ファイルの一部をメモリーにマッピングできます。後者の場合、現在マッピングされているファイルを超えた部分にアクセスすると、参照先の新しい領域を含むファイルの部分をマッピングするために、ウィンドウが「スライド」されます。

XML: Extensible Markup Language。

XML 名前空間: 要素群の固有の範囲 ID としての役割を果たす URI によって定義します。

XMS: NeoCore™ XML Information Management System。

XMS 挿入: 1 つ以上の文書に構造体を追加するために使用します。SQL データベースに列を追加するのと同様です。

Xpath: XML 文書内の項目を検出して処理するための方法を記述する言語。Xpath では、文書の論理構造や階層構造のパスに基づくアドレス指定構文を使用します。

XSLT: Extensible Stylesheet Language for Transformations。

索引

「

「..」の複数インスタンス • 62

1

1つの大きな文書の格納と多数の小さな文書の格納との比較 • 23

A

APIの互換性 • 116, 119

API接続管理 • 132

avg • 95

C

C++ API (アプリケーションプログラミングインターフェース) • 3, 119

C++ APIコードサンプル • 120

C++ APIの資料 • 119

C++フェーズ 1

 C++アプリケーションサンプルソースコード • 121

C++フェーズ 2

 サンプルアプリケーションのコンパイル • 124

C++フェーズ 3

 C++コンパイル済みの実行可能画面 • 128

ceiling • 85

COM API (アプリケーションプログラミングインターフェース) • 3, 130

COMオブジェクトによる NeoCore XMS データベースの操作 • 167

COMコンポーネントの登録 • 130

COMに関する付録 • 132

COMの使用法に関する Visual Basic (VB) サンプルコード • 180

concat • 86

contains • 87

count • 95

Count クエリーと Countnotacid クエリー • 115

count コマンドの使用 • 38

D

distinct-nodes • 93

distinct-values • 94

document • 96

double • 84

E

ends-with • 87

F

floor • 85

FLWR 式 • 80

H

histogram • 94

HTTP API • 187

HTTP API (アプリケーションプログラミングインターフェース) • 3, 187

I

integer • 84

IQueryResult • 168

IResultProfile • 178

J

Java API (アプリケーションプログラミングインターフェース) • 3, 116

Java APIの資料 • 116

Java API メソッド呼び出しの例 • 116

Java フェーズ 1

 Java アプリケーションサンプルソースコード • 117

L

last • 97

local-name • 92

lower-case • 91

M

max • 96

min • 96

N

NeoCore API COM の手動登録 • 131

NeoCore XMS でのデータの効率的な扱い方 • 21

NeoCore XMS へのアクセス • 3

NeoServer の開始と停止 • 4

NeoXDBRuntime.xml ファイルとの関連 • 102

normalize-space • 90

not • 91

number • 93

P

position • 97

R

replace (\$srcval, \$mapString, \$transString) • 91
root • 92
round • 86

S

Section 属性の値の変更 • 32
starts-with • 87
string • 86
string-length • 89
substring (\$sourceString, \$startingLoc, \$length) • 88
substring-after • 89
substring-before • 89
sum • 96

T

Tree クエリー • 115
tree コマンドの使用 • 37

U

upper-case • 90

X

XML コンストラクタ • 81
XML の格納 • 105
XML の削除 • 109
XML の挿入 • 110
XML の変更 • 113
XML 格納処理 • 105
XML 文書のコピー • 113
XML 文書のための格納パラメータ • 105
XMS クエリー • 54
XMS コマンド • 114
XMS での XPath ターゲットの使用 • 55
XMS トランザクションとコマンド • 101
XMS のカーソル機能とチャンク機能 • 167
XMS の構成 • 25
XMS 管理コマンド • 155
XMS 規則 • 55
XPath 述部の使用 • 56
Xpiori NeoCore XML Information Management System
– 概要 • 1
XQuery • 64
XQuery キーワードリスト • 99
XQuery に関する注意事項 • 99
XQuery 関数サポート • 84

XQuery 言語サポート • 65

あ

アーキテクチャ • 5
アクセス制御 • 160

い

位置クエリー • 56

え

エラーメッセージ • 9
エンコードの動作 • 205

お

お客様のコメントをお送りください • 2
お断り • 8

か

概要 • 54, 64
格納例外メッセージ • 108
括弧の付いた式 • 67
関数呼び出し • 67
関連資料 • 2

き

基本式 • 65
基本ロケーションパス • 54
兄弟レベルの要素の挿入 • 26

く

クエリー最適化の一般的なルール • 44
クエリー調整の目標 • 44
クエリー内のノード順序 • 24

こ

コメント • 68

さ

サーバー側の XSLT 拡張機能 • 202
サイズの限界 • 100
算術式 • 78

し

シーケンスの結合 • 75
シーケンスの作成 • 77

シーケンス式 • 75
軸 - XPath 軸の明示的な構文 (axis::name) • 69
子孫パスとワイルドカードパスの比較 • 49
自動登録 • 130
修飾パスと子孫パスの比較 • 47
修飾パスとワイルドカードパスの比較 • 48
修飾文字列とワイルドカード文字列の比較 • 50
述部 • 74
述部と WHERE 句の比較 • 46
述部の構文 • 58
述部のタイプ • 59
出力サイズの抑制 • 53
手動登録 • 130
初期データの回復 • 43
新機能 • 1

す

数値比較と文字列比較の比較 • 51
スキーマの格納 • 106
スタンドアロン要素を使用すると記憶域オーバーヘッドが大きくなるという問題 • 34
ステップ • 69
スペース • 99

せ

セッション管理 • 134

そ

挿入と格納の違い • 25
ソート式 • 81
属性と要素の使い分け • 33
属性の誤用 • 34

た

大規模なデータベースの再起動 • 41
対象読者 • 2
多言語対応の注意点 • 119
他の文書の参照 • 30
単一文書の更新 • 21

ち

著作権情報 • 2

て

定数を使用した数値比較 • 52
データの更新 • 21
データベースコマンド • 135
データ型変換や文字列操作をソートキーに組み込む方法 • 83

データ構造 • 17
データ専用クエリー • 114

と

トランザクション • 101
トランザクション間の分離レベル • 103
トランザクションコマンド • 150
トランザクションのネストの禁止 • 40

な

名前空間 • 31
名前空間を使用した XML の格納とクエリー • 107

ね

ネストされたセット
外側を小さく • 45

の

ノード(.)の内容のソート • 82
ノードテスト • 72
ノード不一致エラーを回避するための一致順序の変更 • 29

は

パス式 • 69

ひ

比較式 • 79
表記規則 • 2

ふ

複数の挿入 • 111
複数文書の格納 • 23
複数文書の更新 • 22
プレフィックスファイルの格納 • 107
文書のコピーの変更 • 42

へ

変数 • 66

ま

マルチ文書ファイル格納のトランザクション動作 • 109

め

メタデータによるソート • 83

も

文字データ（CDATA）の使用・30

ゆ

ユーティリティクラス分離・42

よ

用語集・207

要素ノードの使用・32

要素を使用した設計・33

り

リテラル・66

れ

例・203

ろ

ロック・103

論理式・80