



Service Registry 3 2005Q4 管理ガイド

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

Part No: 819-3572
2005 年 10 月

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

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

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

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

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

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

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

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

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

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

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

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

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

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

「ATOK」は、株式会社ジャストシステムの登録商標です。

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

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

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

Unicode は、Unicode, Inc. の商標です。

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

原典: Service Registry 3 2005Q4 Administration Guide

Part No: 819-2684

Revision A



060119@13215



目次

はじめに 7

- 1 **Service Registry** の設定およびセットアップ 15
 - Service Registry の設定 15
 - ▼ 設定なしインストール後にカスタムプロパティを使って Service Registry を設定する方法 17
 - ▼ 管理ツールの使用を有効にする方法 18
 - Service Registry 用 Application Server ドメインの管理 19
 - ▼ Application Server 管理コンソールを使用する方法 20
 - ▼ レジストリ用 Application Server ドメインを停止および再起動する方法 20
 - ▼ レジストリ用ドメイン内の信頼できる証明書にルート証明書を追加する方法 21
 - 非デフォルト Application Server インストールに対する Service Registry の設定 22
 - ▼ install.properties ファイルを編集する方法 22
 - 非デフォルト Service Registry インストールに対する Service Registry の設定 23
 - ▼ install.properties ファイルを編集する方法 23
 - 外部 Web サイトへのアクセス許可 23
 - ▼ Web プロキシを指定する方法 23
 - 管理者の作成 24
 - ▼ 管理者を作成する方法 24
 - Web コンソールの設定 26
 - 定義済みクエリーの追加 26
 - ▼ 定義済みクエリーを追加する方法 26
 - デフォルトクエリーの変更 28

▼ デフォルトクエリーを変更する方法	28
Classification Scheme の非表示	28
▼ Classification Scheme を非表示にする方法	29
検索結果の表示の設定	29
▼ 検索結果の表示内の行数を設定する方法	30
▼ 検索結果領域の列を設定する方法	30
Service Registry の再インストール	31
▼ レジストリ用 Application Server ドメインを停止および削除する方法	32
▼ Service Registry データベースを再インストールする方法	32
データベースのバックアップおよび復元	32
▼ Apache Derby のドキュメントを見つける方法	33
2 管理ツールの使用	35
管理ツールについて	35
管理ツールの起動	36
バッチモード	36
対話型モード	36
管理ツールのコマンド行オプション	37
管理ツールの機能	39
権限	39
例外の表示	39
レジストリオブジェクトの特定	39
ロケールの名前指定への影響	40
大文字と小文字の区別	40
管理ツールコマンドの使用	41
add association	41
add user	43
cd	48
chown	49
cp	50
echo	52
help	52
lcd	53
ls	54
pwd	55
quit	56
rm	56
select	58

set 58
show 59
users 60

索引 63

はじめに

この『Service Registry 3 2005Q4 管理ガイド』では、Service Registry (「レジストリ」) のインストール後の設定方法やレジストリに付属する管理ツールの使用方法について説明します。また、レジストリデータベースのバックアップや復元など、その他の管理作業についても説明します。

対象読者

この管理ガイドの対象読者は、レジストリをインストール、アンインストール、および管理する必要がある方、およびレジストリのコンテンツを Web コンソールを使わずに一括作成する必要がある方です。

読者には、使用するオペレーティングシステム (Solaris™ オペレーティング環境または Linux) における UNIX® コマンドシェル環境の基本に習熟していることが期待されます。

お読みになる前に

このマニュアルをお読みになる前に、『Sun Java Enterprise System 2005Q4 Installation Guide for UNIX』の説明に従ってレジストリをインストールする必要があります。

Service Registry は、Java Web Services Developer Pack (<http://java.sun.com/webservices/jwsdp/>) の一部として入手できるほか、Sun Java Enterprise System のコンポーネントとしても入手できます。Sun Java Enterprise System は、特定のネットワーク上やインターネット環境に分散配置された

企業アプリケーションをサポートするソフトウェアインフラです。Service Registry を Java Enterprise System のコンポーネントとして購入した場合、<http://docs.sun.com/coll/1286.1>にあるシステムマニュアルを熟読されることをお勧めします。

管理作業の中には、次の各仕様の基本概念に習熟していることが要求されるものもあります。

- 『ebXML Registry Information Model Version 3.0』
- 『ebXML Registry Services and Protocols Version 3.0』

これらの仕様の最新の公開版を入手するには、OASIS Web サイト (<http://www.oasis-open.org/>)にアクセスし、ebXML RIM V3.0 と ebXML RS V3.0 に対するリンクをクリックします。

内容の紹介

このマニュアルの内容は次のとおりです。

第1章では、Service Registry のインストール後の設定方法やその他の管理作業の実行方法について説明します。

第2章では、管理ツールの使用方法について説明します。

Service Registry のマニュアルセット

Service Registry のマニュアルセットは <http://docs.sun.com/app/docs/coll/1314.1> から入手可能です。Service Registry について学ぶには、次の表に記載されたマニュアルを参照してください。

表 P-1 Service Registry のマニュアル

マニュアルタイトル	内容
『Service Registry 3 2005Q4 リリースノート』	既知の問題を含む、Service Registry の最新情報を記載しています。

表 P-1 Service Registry のマニュアル (続き)

マニュアルタイトル	内容
『Service Registry 3 2005Q4 管理ガイド』	Service Registry のインストール後の設定方法やレジストリに付属する管理ツールの使用方法について説明します。また、その他の管理作業の実行方法についても説明します。
『Service Registry 3 2005Q4 ユーザーズガイド』	Service Registry の Web コンソールによる Service Registry の検索方法やデータの発行方法について説明します。
『Service Registry 3 2005Q4 開発者ガイド』	JAXR (Java API for XML Registries) による Service Registry の検索方法やデータの発行方法について説明します。

関連マニュアル

Service Registry をインストールすると、それは Sun Java System Application Server に配備されます。Application Server の管理方法については、『Sun Java System Application Server Enterprise Edition 8.1 2005Q2 Administration Guide』を参照してください。

Java ES のマニュアルセットでは、配備計画やシステムのインストールについて説明しています。システムマニュアルの URL は、<http://docs.sun.com/coll/1286.1> です。Java ES の概要を把握するには、これらのマニュアルを次の表に記載されている順番で参照してください。

表 P-2 Java Enterprise System のマニュアル

マニュアルタイトル	内容
『Sun Java Enterprise System 2005Q4 Release Notes』	既知の問題を含む、Java ES の最新情報を記載しています。さらに、コンポーネントごとのリリースノートも用意されています。
『Sun Java Enterprise System 2005Q4 Documentation Roadmap』	Java ES に関するすべてのマニュアルについて説明します。システム全体に関するマニュアルと、個々のコンポーネントに関するマニュアルの両方を含みます。
『Sun Java Enterprise System 2005Q4 Technical Overview』	Java ES の技術的、概念的な基礎について概説します。コンポーネント、アーキテクチャー、プロセス、および機能について説明します。

表 P-2 Java Enterprise System のマニュアル (続き)

マニュアルタイトル	内容
『Sun Java Enterprise System 2005Q4 Deployment Planning Guide』	Java ES に基づく企業向け配備ソリューションの計画と設計について概説します。配備計画と配備設計に関する基本的な概念や原則について説明し、ソリューションのライフサイクルについて説明し、Java ES に基づくソリューションを計画する際に参考になる高レベルの例や方針を提供します。
『Sun Java Enterprise System 2005Q4 Installation Planning Guide』	Java ES 配備のハードウェア、オペレーティングシステム、ネットワークの各側面に対する実装仕様を作成する際に役立つ情報を提供します。コンポーネント間の依存関係など、インストール計画時や設計計画時に解決すべき問題について説明します。
『Sun Java Enterprise System 2005Q4 Installation Guide for UNIX』	Solaris オペレーティングシステムまたは Linux オペレーティングシステムへの Java ES のインストール手順について説明します。また、インストール後に各コンポーネントを設定する方法や各コンポーネントが正しく動作しているか確認する方法についても説明します。
『Sun Java Enterprise System 2005Q4 Installation Reference』	設定パラメータに関する追加情報や設計計画時に使用するワークシートを提供するほか、デフォルトのディレクトリやポート番号など、各種リファレンス情報の一覧表も提供します。
『Sun Java Enterprise System 2005Q1 Deployment Example Series: Evaluation Scenario』	Java ES を 1 つのシステムへインストールし、一連のコアサービス、共有サービス、およびネットワークサービスを確立し、確立したサービスにアクセス可能なユーザーアカウントを設定する方法について説明します。
『Sun Java Enterprise System 2005Q4 アップグレードガイド』	Solaris オペレーティングシステムまたは Linux オペレーティング環境における Java ES のアップグレード手順について説明します。
『Sun Java Enterprise System Glossary』	Java ES のマニュアルで使用されている用語を定義します。

Java ES とそのコンポーネントに関するすべてのマニュアルの URL は、<http://docs.sun.com/prod/entsys.05q4> です。

デフォルトのパスとファイル名

次の表は、このマニュアルで使用されているデフォルトのパス名とファイル名について説明したものです。

表 P-3 デフォルトのパスとファイル名

プレースホルダ	説明	デフォルト値
<i>ServiceRegistry-base</i>	Service Registry のベースインストールディレクトリを表します。	Solaris システムの場合: /opt/SUNWsoar Linux システムの場合: /opt/sun/SUNWsoar
<i>RegistryDomain-base</i>	Service Registry 用の Application Server ドメインや Service Registry データベースが格納されるディレクトリを表します。	Solaris システムの場合: /var/opt/SUNWsoar Linux システムの場合: /var/opt/sun/SUNWsoar

表記上の規則

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

表 P-4 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 machine_name% you have mail.
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	machine_name% su Password:
<i>aabbcc123</i>	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm <i>filename</i> と入力します。
『 』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。
「 」	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第 5 章「衝突の回避」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。

表 P-4 表記上の規則 (続き)

字体または記号	意味	例
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	sun% grep `^#define \ XV_VERSION_STRING`

コード例は次のように表示されます。

■ C シェル

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

■ C シェルのスーパーユーザー

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

■ Bourne シェルおよび Korn シェル

\$ **command y|n** [filename]

■ Bourne シェルおよび Korn シェルのスーパーユーザー

command y|n [filename]

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

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

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

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

コマンド例のシェルプロンプト

次の表は、デフォルトのシステムプロンプトとスーパーユーザープロンプトを示したものです。

表 P-5 シェルプロンプト

シェル	プロンプト
UNIX および Linux システムの C シェル	machine_name%

表 P-5 シェルプロンプト (続き)

シェル	プロンプト
UNIX および Linux システムの C シェルのスーパーユーザー	machine_name#
UNIX および Linux システムの Bourne シェルおよび Korn シェル	\$
UNIX および Linux システムの Bourne シェルおよび Korn シェルのスーパーユーザー	#
Microsoft Windows のコマンド行	C:\

記号の表記規則

次の表では、このマニュアルで使用する記号について説明します。

表 P-6 記号の表記規則

記号	説明	例	意味
[]	省略可能な引数およびコマンドオプションを含みます。	ls [-l]	-l オプションは必須ではありません。
{ }	必須コマンドオプションの選択肢を含みます。	-d {y n}	-d オプションには、引数 y または引数 n を指定する必要があります。
\${ }	変数の参照を示します。	\${com.sun.javaRoot}	com.sun.javaRoot 変数の値を参照します。
-	同時に押すキーを連結します。	Control-A	Ctrl キーと A キーを同時に押します。
+	連続して押すキーを連結します。	Ctrl + A + N	Ctrl キーを押して離れたあと、後続のキーを押します。
→	グラフィカルユーザーインターフェイスでのメニュー項目の選択を示します。	「ファイル」→「新規」 →「テンプレート」	「ファイル」メニューから「新規」を選択します。「新規」サブメニューから「テンプレート」を選択します。

マニュアル、サポート、およびトレーニング

Sun のサービス	URL	内容
マニュアル	http://jp.sun.com/documentation/	PDF 文書および HTML 文書をダウンロードできます。
サポートおよび トレーニング	http://jp.sun.com/supporttraining/	技術サポート、パッチのダウンロード、および Sun のトレーニングコース情報を提供します。

第 1 章

Service Registry の設定およびセットアップ

この章では、Service Registry のインストール後の設定方法およびその他の管理作業の実行方法について説明します。

この章の内容は次のとおりです。

- 15 ページの「Service Registry の設定」
- 23 ページの「外部 Web サイトへのアクセス許可」
- 24 ページの「管理者の作成」
- 26 ページの「Web コンソールの設定」
- 31 ページの「Service Registry の再インストール」
- 32 ページの「データベースのバックアップおよび復元」

Service Registry の設定

レジストリのデフォルトのプロパティ設定を使って Service Registry のインストール後の設定を実行する方法については、『Sun Java Enterprise System 2005Q4 Installation Guide for UNIX』を参照してください。カスタムプロパティ設定を使用するには、設定を実行する前に、ファイル

`ServiceRegistry-base/install/install.properties` を編集します。

`ServiceRegistry-base` の場所は、Solaris オペレーティング環境の場合は `/opt/SUNWsoar`、Linux システムの場合は `/opt/sun/SUNWsoar` になります。

注 – Service Registry を設定する前に、Sun Java System Application Server (「Application Server」) をインストールおよび設定しておく必要があります。Service Registry の設定プロセスは、レジストリを Application Server ドメイン内にインストールします。

Application Server はデフォルトの場所にインストールすることをお勧めします。Application Server をデフォルト以外の場所にインストールした場合には、Service Registry を設定する前に 22 ページの「非デフォルト Application Server インストールに対する Service Registry の設定」の手順に従ってください。

install.properties ファイルには、一連の変更可能なプロパティ設定が含まれています。表 1-1 に記載されたプロパティは、インストールプロセスによって使用されるものです。各プロパティ名には、プレフィックス registry.install. (末尾はピリオド) が付加されます。これらのプロパティの大部分は、レジストリ用に作成された Application Server ドメインにデフォルト以外のポートを設定します。

表 1-1 Service Registry インストールプロパティ

プロパティ名	説明	デフォルトのプロパティ値
DomainName	Application Server ドメイン名	registry
ServerInstancePort	Application Server HTTP ポート。	6060
ServerInstanceSecurePort	Application Server HTTPS ポート。	6443
ServerJMSPort	Application Server Message Queue ポート	6484
ServerIIOPPort	Application Server IIOP ポート	6485
ServerIIOPSecurePort	Application Server IIOP セキュアポート	6486
ServerIIOPMutualAuthPort	Application Server IIOP 相互認証ポート	6487
AdministrationJMXPort	Application Server JMX ポート	6488
AdministrationPort	Application Server 管理サーバーポート	6489

表 1-1 Service Registry インストールプロパティ (続き)

プロパティ名	説明	デフォルトのプロパティ値
AdministratorUserID	Application Server 管理サーバーへのアクセス時に使用するユーザー名	admin
AdministratorPassword	Application Server 管理サーバーへのアクセス時に使用するパスワード	12345678
ApplicationServerKeystorePassword	Application Server キーストアへのアクセス時に使用するパスワード	12345678
RegistryServerKeystorePassword	Service Registry キーストアへのアクセス時に使用するパスワード	12345678

▼ 設定なしインストール後にカスタムプロパティを使って Service Registry を設定する方法

始める前に レジストリを設定するには、root としてログインするか、スーパーユーザーになる必要があります。

- 手順
1. ディレクトリ `ServiceRegistry-base/install` に移動します。
 2. ファイル `install.properties` 内の変更可能なプロパティを編集します。
セキュリティ上の理由により、このファイルの編集時にはパスワード値を変更しないことをお勧めします。代わりに、コマンド行でそれらの値を指定します。
 3. ファイルの編集が完了したら、次のコマンドを実行します (すべてを 1 行で入力)。
Solaris の場合: `/usr/sfw/bin/ant -f build-install.xml install`
Linux の場合: `/opt/sun/bin/ant --noconfig -f build-install.xml install`

ant コマンドが動作するには、`JAVA_HOME` 環境変数が設定されている必要があります。通常はこの変数を次の値に設定します。

```
/usr/jdk/entsys-j2se
```

変更後のパスワードをコマンド行で指定するには、コマンドのオプションを次のように指定します (すべてを 1 行で入力)。

```
/usr/sfw/bin/ant -f build-install.xml
-Dregistry.install.RegistryServerKeystorePassword=passwd1
-Dregistry.install.AdministratorPassword=passwd2
-Dregistry.install.ApplicationServerKeystorePassword=passwd3 install
```

レジストリ設定プロセスは、

`RegistryDomain-base/domains/${registry.install.DomainName}` に Application Server ドメインを作成します。デフォルトのドメイン名は、`registry` です。その後、設定プロセスは、そのドメインを起動し、レジストリを配備したあと、その実行中のドメインをそのままの状態にしておきます。

レジストリ設定プロセスは、レジストリデータベースとサーバーキーストアをディレクトリ `RegistryDomain-base/3.0` 内にインストールします。このディレクトリはレジストリをアンインストールしても削除されませんが、これは、そのデータベースを将来のリリースにおいても使用できるように保護するためです。このディレクトリを削除するかどうかや、削除する場合のタイミングは、管理者によって制御されます。

`RegistryDomain-base` の場所は、Solaris オペレーティング環境の場合は `/var/opt/SUNWsoar`、Linux システムの場合は `/var/opt/sun/SUNWsoar` になります。

4. `ant install` コマンドの出力にエラーが含まれていないか確認します。エラーが存在しない場合には、Web コンソールまたは管理ツールを使い始めることができます。

▼ 管理ツールの使用を有効にする方法

管理者のロールを持つユーザーに限定された管理ツール作業を実行するには、管理ツールマニフェストのクラスパスにある特定の JAR ファイルを追加することによってバグを回避する必要があります。

- 手順
1. 自分自身がまだ `ServiceRegistry-base/install` ディレクトリ内にいることを確認します。

2. 配備されたレジストリ内のファイル `soapprocessor.jar` を、レジストリの `lib` ディレクトリにコピーします。次のコマンドを実行します (すべてを 1 行で入力)。

```
cp
RegistryDomain-base/domains/registry/applications/j2ee-modules/soar/WEB-INF/lib/soapprocessor.jar
../lib
```

3. `admin.jar.manifest.fix` インストールターゲットを次のようにして実行します。

Solaris の場合: `/usr/sfw/bin/ant -f build-install.xml`
`admin.jar.manifest.fix`

Linux の場合: `/opt/sun/bin/ant --noconfig -f build-install.xml`
`admin.jar.manifest.fix`

Service Registry 用 Application Server ドメインの管理

Service Registry の設定プロセスはデフォルトで、registry という名前の Application Server ドメインを作成し、そこに Service Registry Web アプリケーションを配備します。このドメインは、RegistryDomain-base/domains/registry ディレクトリ内に存在します。

この場所は、Application Server ドメインのデフォルトの場所である、
/var/opt/SUNWappserver/domains (Solaris)、
/var/opt/sun/appserver/domains (Linux) とは異なります。

registry ドメインを管理するには、Application Server 管理コンソール(「管理コンソール」)を使用できます。管理コンソールを使えば、ドメインの起動と停止、サーバーログの表示など、各種の管理作業を行えます。詳細については、[20 ページ](#)の「Application Server 管理コンソールを使用する方法」を参照してください。

また、サーバーログを直接確認することもできます。このログはファイル RegistryDomain-base/domains/registry/logs/server.log 内に格納されています。

また、asadmin コマンドを使って registry ドメインを管理することも可能です。ただし、このドメインはデフォルトの場所に存在しないため、--domaindir オプションをサポートする asadmin コマンドを使用する際には、そのオプションを指定する必要があります。

registry ドメインのパスワードファイルは、ServiceRegistry-base/pw.txt です。registry ドメインを管理するには、このパスワードファイルを、asadmin コマンドの --passwordfile オプションの引数として指定します。

registry ドメインでは、デフォルトの Application Server ドメインである domain1 との間で衝突が発生しないように、一連の非デフォルトポートが使用されます。表 1-2 に、それらのポートの一覧と説明を示します。詳細については、『Sun Java System Application Server Enterprise Edition 8.1 2005Q2 Administration Guide』の「Ports in the Application Server」を参照してください。

表 1-2 Service Registry ドメインのデフォルトのポート

ポート値	説明
6060	HTTP ポート
6443	HTTPS over SSL
6484	Message Queue ポート
6485	IIOp ポート
6486	IIOp SSL ポート

表 1-2 Service Registry ドメインのデフォルトのポート (続き)

ポート値	説明
6487	IIOP 相互認証ポート
6488	JMX ポート
6489	Application Server ドメイン管理ポート

▼ Application Server 管理コンソールを使用する方法

- 手順
1. **Web** ブラウザで URL `https://hostname:6489/` にアクセスします。
hostname は、Application Server と Service Registry が動作しているシステムです。
 2. 提供された証明書を受け入れます。
 ログインページが表示されます。
 3. ログインページの「ユーザー名」フィールドに「**admin**」と入力します。
 4. 「パスワード」フィールドに **Application Server** 管理者パスワードを入力します。レジストリ設定時に **AdministratorPassword** プロパティに指定した値を使用します。デフォルトは **12345678** です。
 5. 「ログイン」をクリックします。

参照 管理コンソールの使用方法の詳細については、管理コンソールのオンラインヘルプまたは『Sun Java System Application Server Enterprise Edition 8.1 2005Q2 Administration Guide』を参照してください。

Service Registry のロギングレベルを変更するには、『Sun Java System Application Server Enterprise Edition 8.1 2005Q2 Administration Guide』の「To configure log levels」の手順に従います。「追加プロパティ」領域で指定すべきプロパティは、`org.apache.commons.logging.simplelog.log.org.freebxml.omar` です。

▼ レジストリ用 Application Server ドメインを停止および再起動する方法

レジストリの設定プロセスは、レジストリの配備先となる Application Server ドメインを起動します。特定の管理作業を実行し終わったら、そのドメインを停止および再起動する必要があります。そうした作業の例としては、23 ページの「外部 Web サイトへのアクセス許可」や 24 ページの「管理者の作成」が挙げられます。

ドメインを再起動する必要がある場合には、管理コンソールからユーザーにその旨が通知されます。管理コンソールを使えばその作業を実行できます。asadmin コマンドを使用している場合には、Ant タスクを使ってドメインを停止および再起動できます。

手順 1. **Service Registry** インストールディレクトリに移動します。

```
cd ServiceRegistry-base/install
```

2. 次のコマンドを実行します (すべてを 1 行で入力)。

```
Solaris の場合: /usr/sfw/bin/ant -f build-install.xml  
appserver.domain.bounce
```

```
Linux の場合: /opt/sun/bin/ant --noconfig -f build-install.xml  
appserver.domain.bounce
```

このターゲットは、ドメインを停止したあと、ドメインを再起動します。

build-install.xml ファイルには、レジストリ用ドメインの停止と起動を別個に行うための Ant ターゲットも含まれています。このドメインを停止するには、Ant ターゲット `appserver.domain.stop` を使用します。このドメインを起動するには、Ant ターゲット `appserver.domain.start` を使用します。

▼ レジストリ用ドメイン内の信頼できる証明書にルート証明書を追加する方法

この作業を行うと、Application Server registry ドメイン内の信頼できる証明書リストが拡張されます。

この作業は、ユーザーがサードパーティーの証明書を使用し、そのサードパーティーのルート証明書発行局 (CA) の証明書が Application Server の `truststore` 内にまだ存在しない場合にのみ実行します。レジストリで発行された証明書のみを使用する場合は、この作業を実行しないでください。

手順 1. サポートする必要があるルート証明書をダウンロードします。ルート証明書を提供するサイトを次に示します。

- <http://www.entrust.net/developer/>
- http://www.geotrust.com/resources/root_certificates/
- <http://www.thawte.com/roots/>
- <http://www.verisign.com/support/roots.html>

2. 必要に応じて、**unzip** コマンドを使って、ダウンロードしたアーカイブから **.cer** ファイルを取り出します。

注 - ファイルによっては、サフィックスが **.der** になっています。

3. **.cer** ファイルをディレクトリ `ServiceRegistry-base/install/cacerts` にコピーします。

4. ディレクトリ `ServiceRegistry-base/install` に移動します。

5. 次のコマンドを実行します (すべてを 1 行で入力)。

Solaris の場合: `/usr/sfw/bin/ant -f build-install.xml
install.cacerts`

Linux の場合: `/opt/sun/bin/ant --noconfig -f build-install.xml
install.cacerts`

このコマンドは、ディレクトリ `ServiceRegistry-base/install/cacerts` 内で見つかったすべての証明書を、Application Server ドメインの `truststore` 内にインストールします。

`list.cacerts` ターゲットを使えば、証明書が正しくインストールされたか確認できます。

6. 20 ページの「レジストリ用 Application Server ドメインを停止および再起動する方法」の手順に従います。

非デフォルト Application Server インストールに対する Service Registry の設定

Application Server のデフォルトのインストール場所は、Solaris オペレーティング環境の場合は `/opt/SUNWappserver/appserver`、Linux システムの場合は `/opt/sun/appserver` です。Application Server を別の場所にインストールした場合、Service Registry を設定する前に `install.properties` を編集する必要があります。

▼ `install.properties` ファイルを編集する方法

- 手順
1. `ServiceRegistry-base/install` ディレクトリで、ファイル `install.properties` をテキストエディタで開きます。
 2. プロパティ `appserver.root.dir` のコメントアウトされた定義を見つけます。
 3. コメント文字 (`#`) を削除し、そのプロパティ定義を Application Server の実際の場所置き換えます。
 4. `install.properties` ファイルを保存して閉じます。

次の手順 続いて、15 ページの「Service Registry の設定」の手順に従います。

非デフォルト Service Registry インストールに対する Service Registry の設定

Service Registry のデフォルトのインストール場所は、Solaris オペレーティング環境の場合は `opt/SUNWsoar`、Linux システムの場合は `/opt/sun/SUNWsoar` です。Service Registry を別の場所にインストールした場合、Service Registry を設定する前に `install.properties` を編集する必要があります。

▼ `install.properties` ファイルを編集する方法

- 手順
1. `ServiceRegistry-base/install` ディレクトリで、ファイル `install.properties` をテキストエディタで開きます。
 2. プロパティ `soar.sdk.home` および `soar.server.home` のコメントアウトされた定義を見つけます。
 3. 各プロパティのコメント文字 (`#`) を削除し、プロパティ定義を **Service Registry** の実際の場所で置き換えます。
 4. `install.properties` ファイルを保存して閉じます。

次の手順 続いて、15 ページの「[Service Registry の設定](#)」の手順に従います。

外部 Web サイトへのアクセス許可

任意のレジストリオブジェクト内に格納できる `ExternalLink` オブジェクトは、そのレジストリオブジェクトに関連付けられた URL を指定します。ユーザーが `ExternalLink` オブジェクトを作成できるためには Service Registry が URL を検証できる必要がありますが、それには外部の Web サイトにアクセスできる必要があります。レジストリがファイアウォールの背後に配備されている場合には、こうしたアクセスを可能にするプロキシを設定する必要があります。

プロキシ設定を行うには、Service Registry が配備されている Application Server インスタンスの Java 仮想マシン (JVM) オプションとして、Web プロキシのホストとポートを指定する必要があります。

▼ Web プロキシを指定する方法

- 手順
1. 20 ページの「[Application Server 管理コンソールを使用する方法](#)」の説明に従って **Application Server** 管理コンソールにログインします。

2. 「設定」ノードを開きます。
3. サーバーノード **server-config (Admin Config)** を開きます。
4. 「JVM 設定」をクリックします。
5. 「JVM オプション」タブをクリックします。
6. 「JVM オプションを追加」をクリックします。
7. テキストフィールドに次のように入力します(すべてを1行で入力)。
`-Dhttp.proxyHost=hostname.domainname -Dhttp.proxyPort=8080`
ポートの値は通常 8080 ですが、実際のポートの値がこれとは異なっている場合には、その正しい値を指定します。
8. 「保存」をクリックします。
9. 20 ページの「レジストリ用 **Application Server** ドメインを停止および再起動する方法」の手順に従います。

管理者の作成

Service Registry 管理ツールで行う作業の中には、管理者として登録されているユーザーのみが行えるものもあります。さらに、管理者には、ほかのユーザーが送信したオブジェクトに対するライフサイクル変更 (承認など) の実施を要求されることもあります。

また、管理者は、デフォルトのアクセス制御ポリシー (ACP) を変更することもできます。ただし、ACP の記述は現時点では手動による処理であり、OASIS の XACML (eXtensible Access Control Markup Language) の知識が必要になります。詳細については、『ebXML RIM 3.0』の第9章「Access Control Information Model」、特に9.7.6 節から 9.7.8 節にかけてのサンプルを参照してください。ebXML RIM 3.0 仕様を入手する方法については、7 ページの「お読みになる前に」を参照してください。

▼ 管理者を作成する方法

自分自身を管理者として登録するには、次の手順に従います。

- 手順 1. 『**Service Registry 3 2005Q4 ユーザーズガイド**』の「ユーザーアカウントの作成」の説明に従ってユーザー登録を行います。
ダウンロードした証明書のパス名を記録しておきます。証明書ファイルのデフォルト名は、generated-key.p12 です。

2. 次のようにして **User** オブジェクトの一意の識別子を取得します。
 - a. **Web** コンソールを使って、**User** に設定したオブジェクト型で基本クエリーを実行します。
 - b. 「詳細」リンクをクリックして、レジストリによって作成された **User** オブジェクトを表示します。
 - c. 「一意の識別子」フィールドの値を記録します。
3. ホームディレクトリの次の場所に証明書をコピーします。なお、必要に応じてディレクトリを作成します。

```
$HOME/soar/3.0/jaxr-ebxml/security
```
4. ディレクトリ
`RegistryDomain-base/domains/registry/applications/j2ee-modules/soar/WEB-INF/cla`
に移動します。
5. ファイル **omar.properties** をテキストエディタで開きます。
6. プロパティー **omar.security.authorization.registryAdministrators** の定義を見つけます。
7. そのプロパティー定義を編集し、縦棒 (|) と手順 2 で記録した論理識別子の文字列を追加します。

プロパティー定義はすべて 1 行で入力する必要があります。また、定義内に空白が含まれてはいけません。編集後のプロパティー定義は、次のようになります (すべて 1 行に入る)。

```
omar.security.authorization.registryAdministrators=  
urn:freebxml:registry:predefinedusers:registryoperator|  
urn:uuid:77f5c196-79de-4286-8483-8d80def3583b
```
8. **omar.properties** ファイルを保存して閉じます。
9. 20 ページの「レジストリ用 **Application Server** ドメインを停止および再起動する方法」の手順に従います。

次の手順 別の管理者を追加する場合、**omar.properties** ファイルを編集する必要はありません。管理者になると、管理ツール、**Web** コンソールのいずれかを使ってユーザーを追加し、それらのユーザーを **Web** コンソールを使って管理者として分類することができます。

Web コンソールの設定

管理者は、設定ファイルを編集することにより、Web コンソールの表示方法の一部をカスタマイズできます。ここでは、次の作業について説明します。

- 26 ページの「定義済みクエリーの追加」
- 28 ページの「デフォルトクエリーの変更」
- 28 ページの「Classification Scheme の非表示」
- 29 ページの「検索結果の表示の設定」

Web Console の使用方法の詳細については、『Service Registry 3 2005Q4 ユーザーズガイド』を参照してください。

定義済みクエリーの追加

Service Registry には定義済みクエリーがいくつか含まれていますが、これらは、Web コンソール上の「検索」フォームの「定義済みクエリーを選択」ドロップダウンリスト内に表示されます。管理者は、対象レジストリインストールに固有の新しいクエリーをそのドロップダウンリストに追加できます。

▼ 定義済みクエリーを追加する方法

- 手順
1. Web コンソールを使って **AdhocQuery** オブジェクトをレジストリに発行します。クエリーに指定した名前と説明は、定義済みクエリーのドロップダウンリストに表示されます。クエリーの SQL 文には、ユーザーが入力するデータのプレースホルダを指定します。次のように、対になった一重引用符でデータを囲みます。

```
select * from registryobject where id = '$lid'
```

2. **AdhocQuery** オブジェクトの一意の識別子と SQL 文内のプレースホルダを記録します。
3. ディレクトリ *RegistryDomain-base/3.0/jaxr-ebxml* に移動します。
4. ファイル **registry-browser-config.xml** をテキストエディタで開きます。
5. 次の形式のエントリを **registry-browser-config.xml** ファイルに追加します。SQL 文内の各プレースホルダに対応する **Parameter** 要素を指定します。

```
<Query>
  <AdhocQueryRef id="unique_identifier"/>
  <Parameter parameterName="$placeholder_name" datatype="string">
    <rim:Name>
      <rim:LocalizedString xml:lang="en" charset="UTF-8"
```

```

        value="parameter_name_in_en_locale" />
        <rim:LocalizedString xml:lang="fr" charset="UTF-8"
        value="parameter_name_in_fr_locale" />
    </rim:Name>
    <rim:Description>
        <rim:LocalizedString xml:lang="en" charset="UTF-8"
        value="parameter_description_in_en_locale" />
        <rim:LocalizedString xml:lang="fr" charset="UTF-8"
        value="parameter_description_in_fr_locale" />
    </rim:Description>
</Parameter>
...
</Query>

```

unique_identifier は、AdhocQuery オブジェクトの一意の識別子です。

各パラメータの *parameterName* 属性は、クエリーの SQL 文内のプレースホルダから取り込む必要があります。

datatype 属性には、次のいずれかの値を指定できます。

- *string*: パラメータが検索フォーム内でテキストフィールドとして表示されず。
- *taxonomyElement*: パラメータが検索フォーム内でドロップダウンリストとして表示されます。*taxonomyElement* データ型を指定すると、Name 要素および Description 要素のあとに次のような SlotList 要素が追加されます。

```

<rim:SlotList>
  <rim:Slot name="domain">
    <rim:ValueList>
      <rim:Value>
        classification_scheme_or_concept_id
      </rim:Value>
    </rim:ValueList>
  </rim:Slot>
</rim:SlotList>

```

classification_scheme_or_concept_id は、Concept (または Subconcept) をドロップダウンリストに表示する必要がある Classification Scheme または Concept の一意の識別子です。該当する Classification Scheme がレジストリにまだ存在しない場合は、発行する必要があります。

スロット名は "domain" にする必要があります。

- *boolean*: パラメータが検索フォーム内でチェックボックスとして表示されず。

datatype が *string* または *boolean* である場合は、*defaultValue* 属性を Parameter 要素に追加して、検索フォームに表示されるデフォルト値を指定することもできます。

サポートするすべてのロケールのために、各パラメータの名前と説明に対してローカライズされた文字列値を指定します。検索フォーム内のパラメータのラベルには、現在のロケールの *parameter_name* が表示されます。

registry-browser-config.xml ファイル内の既存エントリを参考として使用します。

6. **registry-browser-config.xml** ファイルを保存して閉じます。
7. 20 ページの「レジストリ用 **Application Server** ドメインを停止および再起動する方法」の手順に従います。

デフォルトクエリーの変更

「定義済みクエリーを選択」ドロップダウンリストにデフォルトとして表示されるクエリーは、ユーザーが名前、説明、および分類でレジストリオブジェクトを検索できる基本クエリーです。

管理者は、このデフォルトを実際の環境に適したクエリーに変更できます。たとえば、デフォルトクエリーを 26 ページの「定義済みクエリーの追加」の手順に従ってレジストリに追加した新しい定義済みクエリーにすることもできます。この変更を行うには、設定ファイルのプロパティを編集します。

▼ デフォルトクエリーを変更する方法

- 手順
1. ディレクトリ
`RegistryDomain-base/domains/registry/applications/j2ee-modules/soar/WEB-INF/classes` に移動します。
 2. ファイル **jaxr-ebxml.properties** をテキストエディタで開きます。
 3. プロパティ **jaxr-ebxml.thin.defaultQueryPanel** の定義を見つけます。デフォルトでは、このプロパティは次のようにコメントアウトされています。

```
#jaxr-ebxml.thin.defaultQueryPanel=
```
 4. コメント文字 (#) を削除します。
 5. 次の例のように、デフォルトにするクエリーの論理識別子を指定することにより、このプロパティの値を設定します。

```
jaxr-ebxml.thin.defaultQueryPanel=urn:oasis:names:tc:ebxml-regrep:query:MyQuery
```
 6. **jaxr-ebxml.properties** ファイルを保存して閉じます。
 7. 20 ページの「レジストリ用 **Application Server** ドメインを停止および再起動する方法」の手順に従います。

Classification Scheme の非表示

Classification scheme のツリー構造は、Web コンソールの次の領域に表示されます。

- 検索フォーム領域 (基本クエリーを選択した場合)

- 「探索」メニュー領域
- ある種のレジストリオブジェクトの Concept を選択する必要がある場合に表示される「ClassificationScheme/Concept セレクタ」ウィンドウ

管理者は、Service Registry のユーザーが Classification Scheme を使用できないようにする場合に、その Classification Scheme を非表示にできます。Classification Scheme を非表示にするには、設定ファイル内にプロパティを定義します。

▼ Classification Scheme を非表示にする方法

- 手順
1. ディレクトリ `RegistryDomain-base/domains/registry/applications/j2ee-modules/soar/WEB-INF/cla` に移動します。
 2. ファイル `jaxr-ebxml.properties` をテキストエディタで開きます。
 3. 次の構文を使ってプロパティ `jaxr-ebxml.registryBrowser.ConceptsTreeModel.hiddenSchemesList` を設定します。プロパティ定義はすべて 1 行で入力する必要があります。また、定義内に空白が含まれてはいけません。


```
jaxr-ebxml.registryBrowser.ConceptsTreeModel.hiddenSchemesList=
class_scheme_id1|class_scheme_id2|...
```

非表示にする各 Classification Scheme の論理識別子を指定します。複数の識別子を指定する場合は、次の例のように縦棒 (|) を使って識別子を区切ります。

```
jaxr-ebxml.registryBrowser.ConceptsTreeModel.hiddenSchemesList=
urn:oasis:names:tc:ebxml-regrep:classificationScheme:StatusType|
urn:oasis:names:tc:ebxml-regrep:profile:ws:classificationScheme:BindingType
```
 4. `jaxr-ebxml.properties` ファイルを保存して閉じます。
 5. 20 ページの「レジストリ用 Application Server ドメインを停止および再起動する方法」の手順に従います。

検索結果の表示の設定

Web コンソールには、デフォルトでは 1 つのクエリーに対して一度に 10 件の検索結果が表示されます。10 件を超える検索結果が返された場合、ユーザーは追加の検索ページを表示できます。管理者は、各ページに表示される検索結果の数を変更できます。

Web コンソールの検索結果領域には、デフォルトでは特定の列が表示されます。オブジェクトごとに、オブジェクトの型、名前、説明、バージョン、およびバージョンコメントが表示されます。一部のオブジェクト型には、非デフォルトの表示が設定されています。たとえば、ServiceBinding オブジェクトでは、バージョン情報の代わりにエンドポイントが表示に含まれます。管理者は、選択したオブジェクト型について非デフォルトのデータを表示するように設定データを追加できます。

これらの作業を実行するには、設定ファイルを編集します。

▼ 検索結果の表示内の行数を設定する方法

- 手順
1. ディレクトリ `RegistryDomain-base/domains/registry/applications/j2ee-modules/soar/WEB-INF/classes` に移動します。
 2. ファイル `jaxr-ebxml.properties` をテキストエディタで開きます。
 3. 次のようなプロパティ `omar.client.thinbrowser.numSearchResults` の定義を見つけます。

```
omar.client.thinbrowser.numSearchResults=10
```
 4. 値として設定されている「10」を希望する値に変更します。
 5. `jaxr-ebxml.properties` ファイルを保存して閉じます。
 6. 20 ページの「レジストリ用 **Application Server** ドメインを停止および再起動する方法」の手順に従います。

▼ 検索結果領域の列を設定する方法

- 手順
1. ディレクトリ `RegistryDomain-base/3.0/jaxr-ebxml` に移動します。
 2. ファイル `registry-browser-config.xml` をテキストエディタで開きます。
 3. 次の形式のエントリを `registry-browser-config.xml` ファイルに追加します。
この例では、Service オブジェクトについて非デフォルトの表示を設定しています。

```
<ObjectTypeConfig
  className="org.freeebxml.omar.client.xml.registry.infomodel.ServiceImpl"
  id="urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Service">
  <SearchResultsConfig>
    <SearchResultsColumn columnClass="java.lang.Object"
      columnHeader="Object Type" columnWidth="25" editable="false"
      method="getObjectType"/>
    <SearchResultsColumn columnClass="java.lang.Object"
      columnHeader="Name" columnWidth="25" editable="true" method="getName"/>
    <SearchResultsColumn columnClass="java.lang.Object"
      columnHeader="Description" columnWidth="30" editable="true"
      method="getDescription"/>
    <SearchResultsColumn columnClass="java.lang.Object"
      columnHeader="Status" columnWidth="15" method="getStatusAsString"/>
  </SearchResultsConfig>
</ObjectTypeConfig>
```

```
<SearchResultsColumn columnClass="java.lang.Object"
    columnHeader="Version" columnWidth="5" method="getVersionName"/>
</SearchResultsConfig>
</ObjectTypeConfig>
```

registry-browser-config.xml には、ObjectTypeConfig 要素のための構文が用意されています。ファイル内の既存の要素を例として使用します。これらの要素は、レジストリオブジェクトについてデフォルトの表示を設定し、ExternalLink オブジェクト、ExtrinsicObject オブジェクト、および ServiceBinding オブジェクトについて非デフォルトの表示を設定しています。

ほとんどの場合、ebXML Registry Information Model Version 3.0 仕様のクラス属性からメソッド名を推測できます (詳細については、7 ページの「お読みになる前に」を参照)。getStatusAsString メソッドは、RegistryObjectImpl 実装クラスで見つかります。(ただし、Service Registry の今回のリリースに API のドキュメントは含まれていません。)

4. **registry-browser-config.xml** ファイルを保存して閉じます。
5. 20 ページの「レジストリ用 **Application Server** ドメインを停止および再起動する方法」の手順に従います。
6. 再構成の結果を確認するため、**Web** コンソールの「検索」メニューまたは「探索」メニューを使って、変更した列のオブジェクトを表示します。

Service Registry の再インストール

Service Registry のアンインストールと再インストールを行う必要がある場合、再インストールを実行する前に次の作業を実行します。

- 保存すべきデータがレジストリデータベースに含まれている場合、32 ページの「データベースのバックアップおよび復元」の説明に従ってデータベースのバックアップを取ります。
- レジストリ用 Application Server ドメインを停止したあと、このドメインを削除します。このドメインを削除しなかった場合、再インストールされたレジストリのインストール後設定が失敗します。

Service Registry データベースを再インストールする必要がある場合 (たとえば、データベースが破損した場合など) は、32 ページの「Service Registry データベースを再インストールする方法」の手順に従います。データベースを再インストールする前にアンインストールする必要はありません。

▼ レジストリ用 Application Server ドメインを停止および削除する方法

手順 1. ディレクトリ `ServiceRegistry-base/install` に移動します。

2. 次のコマンドを実行します (すべてを 1 行で入力)。

Solaris の場合: `/usr/sfw/bin/ant -f build-install.xml
appserver.domain.stop`

Linux の場合: `/opt/sun/bin/ant --noconfig -f build-install.xml
appserver.domain.stop`

3. 次のコマンドを実行します (すべてを 1 行で入力)。

Solaris の場合: `/usr/sfw/bin/ant -f build-install.xml
appserver.domain.delete`

Linux の場合: `/opt/sun/bin/ant --noconfig -f build-install.xml
appserver.domain.delete`

▼ Service Registry データベースを再インストールする方法

手順 1. ディレクトリ `ServiceRegistry-base/install` に移動します。

2. 次のコマンドを実行します (すべてを 1 行で入力)。

Solaris の場合: `/usr/sfw/bin/ant -f build-install.xml install.db`

Linux の場合: `/opt/sun/bin/ant --noconfig -f build-install.xml
install.db`

データベースのバックアップおよび復元

このレジストリは Apache Derby データベースを使用しています。このデータベースは、デフォルトでディレクトリ `RegistryDomain-base/3.0/data/registry/soar/` にあります。

データベースのバックアップ方法および復元方法については、Apache Derby のドキュメントを参照してください。

▼ Apache Derby のドキュメントを見つける方法

- 手順
1. Web ブラウザで、Apache Derby の Web サイト (<http://db.apache.org/derby/>)にアクセスします。
 2. 「Manuals」タブをクリックします。
 3. 「10.0 Manuals」をクリックします。
 4. 「Server & Admin Guide」を探します。
 5. データベースのバックアップや復元に関する項目を探します。

第 2 章

管理ツールの使用

この章では、Service Registry の管理ツールの使用方法について説明します。

この章の内容は次のとおりです。

- 35 ページの「管理ツールについて」
- 36 ページの「管理ツールの起動」
- 39 ページの「管理ツールの機能」
- 41 ページの「管理ツールコマンドの使用」

管理ツールについて

Service Registry 管理ツールは、レジストリへの関連付けの追加やレジストリからのオブジェクトの削除といった一般的な管理作業を行うための、単純なコマンド行インタフェースを提供します。

このツールは次の 2 つのモードのいずれかで動作します。

- バッチモードでは、ツールのコマンド行で 1 つまたは複数のコマンドを指定します。
- 対話型モードでは、ツールの対話型シェル内でコマンドを入力します。

ls や rm などのいくつかのコマンドは、ファイルやフォルダの操作に用いられる有名な UNIX[®] コマンドの名前と動作の両方を模倣しています。その他のコマンドについては、対応する UNIX コマンドはありません。

管理ツールの起動

管理ツールを起動するには、`admin-tool.jar` ファイルを実行します。

```
java -jar ServiceRegistry-base/lib/admin-tool.jar [options]...
```

`ServiceRegistry-base` の場所は、Solaris オペレーティング環境では `/opt/SUNWsoar`、Linux システムでは `/opt/sun/SUNWsoar` です。

ツールの起動時に表示される警告は無視してください。重要なものではありません。

管理ツールを終了するには、`quit` コマンドを使用します。

管理者だけに使用が限定されているコマンドを実行するには、コマンド行から次のように、証明書ファイル (つまりキーストア) の種類、場所、およびパスワードを指定する必要があります (すべてを 1 行で入力)。

```
java -Djaxr-ebxml.security.storetype=PKCS12 \  
-Djaxr-ebxml.security.keystore=security/filename.p12 \  
-Djaxr-ebxml.security.storepass=password -jar admin-tool.jar
```

ここで、`filename` は、証明書ファイルのルート名 (デフォルトでは `generated-key`)、`password` は、証明書作成時にこのファイルの保護用として指定されたパスワードです。場所 `security/filename.p12` は、ディレクトリ `$HOME/soar/3.0/jaxr-ebxml` からの相対パスです。

入力の手間を省くには、このコマンドを実行するスクリプトを作成します。

バッチモード

管理ツールをバッチモードで実行するには、管理ツール起動時のコマンド行で `-command` オプションを指定します。

たとえば、次のコマンドは、`ls` コマンドを実行します。

```
java -jar ServiceRegistry-base/lib/admin-tool.jar -command "ls *.html"
```

管理ツールは、コマンドとその応答を画面上にエコーし、コマンドの実行が完了すると処理を終了します。

シェルにとって特別な意味を持つ文字は、必ず正しくエスケープするようにしてください。

対話型モード

管理ツールを対話型モードで実行するには、コマンド行で `-command` 以外の任意のオプションを指定して管理ツールシェルを起動します。または、次の例のように、オプションを指定せずに管理ツールシェルを起動します。

```
java -jar ServiceRegistry-base/lib/admin-tool.jar
```

管理ツールは、次のプロンプトを表示し、ユーザーからの入力を待ちます。

```
admin>
```

管理ツールのコマンド行オプション

管理ツールは、37 ページの「形式」にリストされ、37 ページの「オプション」で説明されているコマンド行オプションを認識します。

形式

```
[-alias alias] [-command commands] [-debug] [-help]  
[-keypass keypass] [-localdir localdir] [-locale locale] [-registry url]  
[-root locator [-create]] [-sqlselect SQL_statement] [-verbose]
```

オプション

- | | |
|----------|---|
| -alias | キーストア内のユーザーの証明書にアクセスする際に使用するエイリアス。ユーザー登録時に使用したエイリアスを指定します。 |
| -command | 対話型シェルからコマンドを取得しないで実行する管理ツールコマンドシーケンス。複数のコマンドを区切るには、セミコロン (;) を使用します。 <i>commands</i> に quit コマンドを含める必要はありません。コマンド区切り文字でないセミコロンを使用する必要がある場合は、セミコロンの前に円記号を付けます。

\;

管理ツールを実行しているシェルが、円記号をもう 1 つの円記号でエスケープすることを要求する可能性があります。

\; |
| -create | -root オプションで指定された RegistryPackage オブジェクトとそのすべての親 RegistryPackage オブジェクトを、必要に応じて作成します。このオプションが有効になるのは、管理ツールの実行ユーザーがオブジェクト作成権限を持っている場合だけです。 |

-debug	デバッグ時に役立つ追加情報を出力します。
-help	これらのオプションの一覧を表示します。
-keypass	キーストア内のユーザーの証明書にアクセスする際に使用するパスワード。ユーザー登録時に使用したパスワードを指定します。
-localdir	ローカルファイルシステム内のベースディレクトリ。ローカルファイルシステム内のファイルに関するコマンドの場合に指定します。
-locale	エラーメッセージや状態メッセージ用のリソースバンドルを選択する際に使用するロケール (en、fr など)。デフォルトは、Java 仮想マシン (JVM) によって決定されます。
-registry	接続先となる ebXML レジストリの URL。デフォルトは、 <code>http://localhost:6060/soar/registry/soap</code> です。
-root	リポジトリを RegistryPackage オブジェクトのツリーとして扱うコマンドにおいてベースとして使用される RegistryPackage のロケータ (<code>/registry/userData</code> など)。なお、こうしたツリー内の各オブジェクトには、ほかの RegistryObject オブジェクトや RegistryPackage オブジェクトが含まれます。デフォルトは、すべてのユーザーのデータ用に定義された RegistryPackage である、 <code>/registry/userData</code> です。
-sqlselect	レジストリオブジェクトを選択するための <i>SQL_statement</i> を実行します。この文は、 <code>select</code> で始まる完全な SQL 文であるべきです。この SQL 文は、引用符で囲む必要がありますが、末尾にセミコロンを付ける必要はありません。このオプションを指定したあとで、引数なしの <code>select</code> コマンドを使用した場合、 <i>SQL_statement</i> 以外の引数を持つ <code>select</code> コマンドを使用するまでその <i>SQL_statement</i> が繰り返し実行されます。
-v -verbose	状態メッセージの冗長出力を指定します。

注 - `-help` オプションの出力には、このリリースでサポートされていないオプションが 2 つ表示されます。-`class` と -`property` です。

管理ツールの機能

ここでは、管理ツールの次の特徴について説明します。

- 39 ページの「権限」
- 39 ページの「例外の表示」
- 39 ページの「レジストリオブジェクトの特定」
- 40 ページの「ロケールの名前指定への影響」
- 40 ページの「大文字と小文字の区別」

権限

管理ツール使用時に実行可能なアクションは、ツール起動時に指定されたキーエイリアスとパスワードに対応するユーザーに許可されているアクションだけです。コマンドの中には、`chown` のように、管理者のロールを持つユーザーだけが実行できるものもあります。詳細については、[24 ページの「管理者の作成」](#)を参照してください。

例外の表示

管理ツールを使えば、コマンド失敗時に長いスタックトレースが表示されるのを回避できます。

コマンドが失敗した場合、管理ツールは、スタックトレースの 1 行目を出力したあと、次のメッセージを表示します。

```
An error occurred when executing the function. Use the show exception command to view messages.
```

より詳しい情報が必要な場合には、続けて `show exception` コマンドを実行して、完全なスタックトレースを確認します。

`show exception` コマンドは常に、直前のコマンドに対するスタックトレースを表示します。

レジストリオブジェクトの特定

レジストリオブジェクトの特定方法としてもっともよく使用されるのは、名前による特定です。ただし、通常は `registry` ルートから `RegistryPackage` までのパスによって `RegistryPackage` オブジェクトを特定します。たとえば、`/registry/userData` は、`userData RegistryPackage` へのパスです。

一部の名前一致では、ワイルドカードがサポートされています。1文字に一致させるには疑問符(?)を使用します。0個以上の文字に一致させるにはアスタリスク(*)を使用します。

cd や chown など、一部のコマンドでは、URN (Uniform Resource Name) によるオブジェクトの特定がサポートされています。ただし、その場合、先頭の urn: まで含めて指定する必要があります。たとえば、
urn:uuid:2702f889-3ced-4d49-82d1-e4cd846cb9e4 は有効な URN です。

chown コマンドでは別の方法もサポートされています。その方法では、先行する users コマンドで表示された一覧内の特定の User を、% number を使って参照します。

一部のコマンドでは、空白を含む名前を入力できます。その場合、名前全体を二重引用符で囲むか、名前に含まれる各空白の直前に円記号を付けます。

select コマンドでは、SQL のワイルドカードがサポートされています。複数の文字に一致させるにはパーセント記号(%)を、1文字に一致させるにはアンダースコア(_)を、それぞれ使用します。

ロケールの名前指定への影響

RegistryObject (または RegistryPackage) は、異なるロケールに関連付けられた複数の名前を持つことができます。

ユーザーが指定したパスやオブジェクト名は、現在のロケールだけに基いて評価されます。複数の名前を持つレジストリオブジェクトを名前で選択しようとした場合、ユーザーが入力した名前的一致対象となるのは、そのレジストリオブジェクトの名前の1つの選択肢 (現在のロケールにもっとも近いロケールを持つ選択肢) のみです。そのレジストリオブジェクトの複数の名前すべてが一致対象になるわけではありません。

たとえば、現在の RegistryPackage が1つのメンバーオブジェクトを持ち、そのメンバーオブジェクトが2つの名前を持ち、それぞれの名前が異なるロケールに関連付けられているとします。具体的には、red が en (英語) ロケールに、rouge が fr (フランス語) ロケールに、それぞれ関連付けられています。現在のロケールが en である場合、コマンド ls rouge を実行してもそのメンバーオブジェクトは表示されませんが、ロケールが fr (またはその派生の1つ) である場合には、そのコマンドによってそのメンバーオブジェクトが表示されます。

大文字と小文字の区別

管理ツールが認識するコマンド名やリテラルパラメータでは、大文字、小文字の区別はありません。たとえば、ls、Ls、LS はすべて同等です。

値が指定されたオプションは、そのオプションを使用するコードにそのまま渡されません。

管理ツールコマンドの使用

次では、利用可能なコマンドについて説明します。各コマンドの形式、オプション説明、およびオペランド説明は、次の文字体裁規約に従って記述されています。

- 斜体は、コマンド実行時に実際の値に置き換えるべき、オプション引数またはオペランドを示します。
- オプションまたはオペランドの選択肢は、大括弧 ({ }) で囲まれます。ユーザーは、この括弧内のオプションまたはオペランドのいずれか 1 つを選択する必要があります。複数のオプションまたはオペランドは、縦棒 (|) で区切られます。
- 省略可能な 1 つ以上のオプションまたはオペランドは、角括弧 ([]) で囲まれます。
- 1 つのオプションまたはオペランドに続く省略記号 (...) は、その引数またはオペランドを繰り返し指定できることを示します。

上記以外はすべて、コマンド実行時に含める必要のあるリテラルテキストです。

add association

Association オブジェクトをレジストリに追加します。

形式

```
add association -type association-type sourceURN targetURN
```

説明

add association コマンドは、指定されたタイプの Association オブジェクトをレジストリに追加します。次のタイプのいずれかを使用できます。

- AccessControlPolicyFor
- AffiliatedWith (Subconcept として EmployeeOf と MemberOf を持つ)
- Contains
- ContentManagementServiceFor

- EquivalentTo
- Extends
- ExternallyLinks
- HasFederationMember
- HasMember
- Implements
- InstanceOf
- InvocationControlFileFor (Subconcept として CatalogingControlFileFor と ValidationControlFileFor を持つ)
- OffersService
- OwnerOf
- RelatedTo
- Replaces
- ResponsibleFor
- SubmitterOf
- Supersedes
- Uses

オプション

`-type` Association オブジェクトのタイプ。

オペランド

`sourceURN` ソースオブジェクトの URN。

`targetURN` ターゲットオブジェクトの URN。

例

次のコマンド (すべてを 1 行で入力) は、指定された URN を持つ 2 つのオブジェクト間に RelatedTo 関係を作成します。

```
admin> add association -type RelatedTo
urn:uuid:ab80d8f7-3bea-4467-ad26-d04a40045446
urn:uuid:7a54bbca-2131-4a49-8ecc-e7b4ac86c4fd
```

add user

特定のユーザーをレジストリに追加します。

形式

```
add user [-edit] [- load filename] [-firstname string]  
[-lastname string] [-middleName string] - alias string - keypass string  
[-post1.type string] [-post1.city string] [-post1.country string]  
[- post1.postalcode string] [-post1.stateOrProvince string]  
[-post1.street string] [-post2.streetNumber string ]  
[-post2.type string] [-post2.city string] [- post2.country string]  
[-post2.postalcode string] [-post2.stateOrProvince string]  
[-post2.street string] [-post2.streetNumber string ]  
[-post3.type string] [-post3.city string] [- post3.country string]  
[-post3.postalcode string] [-post3.stateOrProvince string]  
[-post3.street string] [-post3.streetNumber string ]  
[-telephone1.type string] [-telephone1.areaCode string]  
[-telephone1.countryCode string] [-telephone1.extension string]  
[-telephone1.number string] [- telephone1.URL string]  
[-telephone2.type string] [-telephone2.areaCode string]  
[-telephone2.countryCode string] [-telephone2.extension string]  
[-telephone2.number string] [-telephone2.URL string ]  
[-telephone3.type string] [-telephone3.areaCode string]  
[-telephone3.countryCode string] [-telephone3.extension string]  
[-telephone3.number string] [- telephone3.URL string]  
[-email1.type string] [-email1.address string] [-email2.type string]  
[-email12address string] [- email3.type string]  
[-email3.address string]
```

説明

add user コマンドは、User オブジェクトを追加します。User オブジェクトには通常、PostalAddress、TelephoneNumber、EmailAddress の各オブジェクトが少なくとも1つずつ含まれます。

ユーザーに関する情報を指定するには、それらの情報をコマンド行から直接入力するか、あるいは -load オプションを使ってそれらの情報を含む Java プロパティファイル指定します。情報オプションと -load オプションは、コマンド行に指定された順番で評価されます。たとえば、いくつかのプロパティをコマンド行から指定し、その他のプロパティをプロパティファイルから読み込んだあとで、そのプロパティファイルの情報を後続のコマンド行オプションで上書きする、といったことも可能です。

新規ユーザーごとにアドレス、電話番号、および電子メールアドレスを最大3つずつ指定できます。それ以上必要な場合も、あとで Web コンソールや JAXR を使って追加できます。

住所、電話番号、電子メールアドレスのいずれかを指定する際には、`-emailType OfficeEmail` のようにタイプの値を指定する必要があります。

コマンド行では、すべてのユーザーで必要となる基本情報のいくつかに対して短形式オプション (`-fn` など) が使用できるようになっています。ただし、そうした情報をプロパティファイルで指定する場合には、長形式を使用する必要があります。たとえば、ユーザーの1つ目の電子メールアドレスをコマンド行から指定する場合には、`-email1.address`、`-emailAddress`、`-email` のいずれかを使用できますが、1つ目の電子メールアドレスをプロパティファイルで指定する場合には、`email1.address=` を使用する必要があります。ユーザーの2つ目の電子メールアドレスに対するオプションは1つしか用意されていないため、コマンド行では `-email2.address` を、プロパティファイル内では `email2.address=` を、それぞれ使用する必要があります。

`-edit` オプションを指定した場合、新しいユーザーの情報を編集できるように、管理ツールによってエディタが起動されます。詳細については、オプション説明を参照してください。

注 - `-load` の読み込み対象または `-edit` の編集対象となるプロパティファイルでは、その他のすべての Java プロパティファイルと同じく、ISO-8859-1 文字が使用されます。ISO-8859-1 に含まれない文字をプロパティファイル内で表現する方法の詳細については、`java.util.Properties.load(InputStream)` のドキュメントを参照してください。

オプション

`-edit`

これを指定すると、新しいユーザーの情報を編集できるように、管理ツールによってエディタが起動されます。ツールによってエディタが起動されるのは、ほかのコマンド行パラメータが評価されたあとです。したがって、コマンド行またはプロパティファイル内に指定された情報の評価結果に基づいて編集作業が開始されます。編集プログラムが正常に終了しないと、コマンドは続行されません。管理ツールは、`set editor` コマンド (58 ページの「`set`」を参照) で指定されたエディタを起動します。これはデフォルトでは `vi` エディタになっています。

注 - このリリースでは、`-edit` は、`emacsclient` および `NetBeans™` コマンド `bin/runide.sh --open` と組み合わせて使用できますが、動作は保証されておらず、`vi` と組み合わせた場合に正常に動作するかどうかは、明らかになっていません。

`-load`
ユーザーのプロパティが格納された Java プロパティファイルを指定します。
そのプロパティ名は、`add user` コマンドの長形式オプション (`lastName`、`post1.type` など) と同一です。

`-fn | -firstName`
ユーザーの名を指定します。

`-ln | -lastName`
ユーザーの姓 (名字) を指定します。姓は必須です。コマンド行、プロパティファイル内のいずれかで指定する必要があります。

`-mn | -middleName`
ユーザーのミドルネームを指定します。

`-alias`
キーストア内のユーザーの証明書にアクセスする際に使用するエイリアス。このオプションは必須です。エイリアスの文字長は 3 文字以上である必要があります。

`-keypass`
キーストア内のユーザーの証明書にアクセスする際に使用するパスワード。このオプションは必須です。パスワードの文字長は 6 文字以上である必要があります。

`-postalType | -post1.type`
1 つ目の `PostalAddress` のタイプ。このタイプは必須です。コマンド行、プロパティファイル内のいずれかで指定する必要があります。値は任意の文字列 (`Office`、`Home` など) です。

`-city | -post1.city`
1 つ目の `PostalAddress` の市。

`-country | -post1.country`
1 つ目の `PostalAddress` の国。

`-postalCode | -postcode | - zip | -post1.postalcode`
1 つ目の `PostalAddress` の郵便番号。

`-stateOrProvince | -state | - province | -post1.stateOrProvince`
1 つ目の `PostalAddress` の州または都道府県。

`-street | -post1.street`
1 つ目の `PostalAddress` の町名。

`-streetNumber | -number | - -post1.streetNumber`
1 つ目の `PostalAddress` の番地。

`-post2.type`
2 つ目の `PostalAddress` のタイプ。2 つ目の `PostalAddress` を指定する場合、そのタイプは必須です。コマンド行、プロパティファイル内のいずれかで指定する必要があります。値は任意の文字列 (`Office`、`Home` など) です。

`-post2.city`
2 つ目の `PostalAddress` の市。

`-post2.country`
2 つ目の `PostalAddress` の国。

- post2.postalcode
2つ目の PostalAddress の郵便番号。
- post2.stateOrProvince
2つ目の PostalAddress の州または都道府県。
- post2.street
2つ目の PostalAddress の町名。
- post2.streetNumber
2つ目の PostalAddress の番地。
- post3.type
3つ目の PostalAddress のタイプ。3つ目の PostalAddress を指定する場合、そのタイプは必須です。コマンド行、プロパティファイル内のいずれかで指定する必要があります。値は任意の文字列 (Office、Home など) です。
- post3.city
3つ目の PostalAddress の市。
- post3.country
3つ目の PostalAddress の国。
- post3.postalcode
3つ目の PostalAddress の郵便番号。
- post3.stateOrProvince
3つ目の PostalAddress の州または都道府県。
- post3.street
3つ目の PostalAddress の町名。
- post3.streetNumber
3つ目の PostalAddress の番地。
- phoneType | -telephone1.type
1つ目の TelephoneNumber のタイプ。このタイプは必須です。コマンド行、プロパティファイル内のいずれかで指定する必要があります。値は任意の文字列ですが、次の既知のタイプのいずれかを指定できます。Beeper、FAX、HomePhone、MobilePhone、または OfficePhone。
- areaCode | -telephone1.areaCode
1つ目の TelephoneNumber の市外局番。
- countryCode | -telephone1.countryCode
1つ目の TelephoneNumber の国番号。
- extension | -telephone1.extension
1つ目の TelephoneNumber の内線。
- number | -telephone1.number
1つ目の TelephoneNumber の電話番号サフィックス。国番号や市外局番は含みません。この番号は必須です。コマンド行、プロパティファイル内のいずれかで指定する必要があります。

-URL | -telephone1.URL
1 目目の TelephoneNumber の URL (この番号に電子的に電話をかけることのできる URL)。

-telephone2.type
2 目目の TelephoneNumber のタイプ。2 目目の TelephoneNumber を指定する場合、そのタイプは必須です。コマンド行、プロパティファイル内のいずれかで指定する必要があります。値は任意の文字列ですが、次の既知のタイプのいずれかを指定できます。Beeper、FAX、HomePhone、MobilePhone、または OfficePhone。

-telephone2.areaCode
2 目目の TelephoneNumber の市外局番。

-telephone2.countryCode
2 目目の TelephoneNumber の国番号。

-telephone2.extension
2 目目の TelephoneNumber の内線。

-telephone2.number
2 目目の TelephoneNumber の電話番号サフィックス。国番号や市外局番は含みません。2 目目の TelephoneNumber を指定する場合、その番号は必須です。コマンド行、プロパティファイル内のいずれかで指定する必要があります。

-telephone2.URL
2 目目の TelephoneNumber の URL (この番号に電子的に電話をかけることのできる URL)。

-telephone3.type
3 目目の TelephoneNumber のタイプ。3 目目の TelephoneNumber を指定する場合、そのタイプは必須です。コマンド行、プロパティファイル内のいずれかで指定する必要があります。値は任意の文字列ですが、次の既知のタイプのいずれかを指定できます。Beeper、FAX、HomePhone、MobilePhone、または OfficePhone。

-telephone3.areaCode
3 目目の TelephoneNumber の市外局番。

-telephone3.countryCode
3 目目の TelephoneNumber の国番号。

-telephone3.extension
3 目目の TelephoneNumber の内線。

-telephone3.number
3 目目の TelephoneNumber の電話番号サフィックス。国番号や市外局番は含みません。3 目目の TelephoneNumber を指定する場合、その番号は必須です。コマンド行、プロパティファイル内のいずれかで指定する必要があります。

-telephone3.URL
3 目目の TelephoneNumber の URL (この番号に電子的に電話をかけることのできる URL)。

- emailType | -email1.type
1つ目の EmailAddress のタイプ。このタイプは必須です。コマンド行、プロパティファイル内のいずれかで指定する必要があります。値は任意の文字列ですが、次の既知のタイプのいずれかを指定できます。HomeEmail または OfficeEmail。
- emailAddress | -email | -email1.address
1つ目の電子メールアドレス。1つ目の電子メールアドレスは必須です。
- email2.type
2つ目の EmailAddress のタイプ。2つ目の EmailAddress を指定する場合、そのタイプは必須です。コマンド行、プロパティファイル内のいずれかで指定する必要があります。値は任意の文字列ですが、次の既知のタイプのいずれかを指定できます。HomeEmail または OfficeEmail。
- email2.address
2つ目の電子メールアドレス。
- email3.type
3つ目の EmailAddress のタイプ。3つ目の EmailAddress を指定する場合、そのタイプは必須です。コマンド行、プロパティファイル内のいずれかで指定する必要があります。値は任意の文字列ですが、次の既知のタイプのいずれかを指定できます。HomeEmail または OfficeEmail。
- email3.address
3つ目の電子メールアドレス。

例

次のコマンドは、ユーザーのホームディレクトリ内のファイル JaneSmith.properties から User のプロパティを読み込みます。

```
admin> add user -load ~/JaneSmith.properties
```

次のコマンド (すべてを 1 行で入力) では、User オブジェクトの作成に最低限必要なプロパティを指定しています。

```
admin> add user -ln Smith -postaltype Office -country US  
-phonetype Office -number 333-3333 -emailtype OfficeEmail  
-emailaddress JaneSmith@JaneSmith.com -alias 123 -keypass 123456
```

cd

RegistryPackage の場所を変更します。

形式

```
cd { locator | URN }
```


説明

`cd` コマンドは、指定されたパスまたは指定された URN を持つ `RegistryPackage` にディレクトリ (メタファー) を変更します。

このコマンドは、(現在のロケールに対して) 同じパスを持つ `RegistryPackage` オブジェクトが複数存在する場合には、指定された URN に変更します。

オペランド

locator レジストリのルートからリポジトリ内の特定のオブジェクトまでの、複数のレジストリオブジェクト名から成るパス。各オブジェクト名の前にはスラッシュ (/) を付けます。

たとえば、`registry RegistryPackage` (これ自身はどの `RegistryPackage` のメンバーにもなっていない) のメンバーになっている `userData RegistryPackage` のロケータは `/registry/userData` であり、その `userData RegistryPackage` のメンバーになっている `folder1 RegistryPackage` のロケータは `/registry/userData/folder1` です。

URN `RegistryPackage` の URN。これは、`urn:` で始まる URN である必要があります。

例

次のコマンドは、URN `urn:uuid:92d3fd01-a929-4eba-a5b4-a3f036733017` を持つ `RegistryPackage` にディレクトリを変更します。

```
admin> cd urn:uuid:92d3fd01-a929-4eba-a5b4-a3f036733017
```

次のコマンドは、場所 `/registry/userData/myData` にディレクトリを変更します。

```
admin> cd /registry/userData/myData
```

chown

特定の `RegistryObject` の所有者を変更します。

形式

```
chown { URN | %index }
```

説明

`chown` コマンドは、先行する `select` コマンドで選択されたオブジェクトの所有者を、URN として指定されたユーザー、または先行する `users` コマンドで表示された一覧内の URN への参照として指定されたユーザーに変更します。

このコマンドを正常に実行できるのは、管理者のロールを持つユーザーだけです。

オペランド

URN URN によって指定された User オブジェクト。

%index 先行する `users` コマンドで表示された一覧内に含まれる User オブジェクトの URN への番号参照。

例

次のコマンドは、選択されたオブジェクトの所有者を、URN `urn:uuid:26aa17e6-d669-4775-bfe8-a3a484d3e079` で指定されるユーザーに変更します。

```
admin> chown urn:uuid:26aa17e6-d669-4775-bfe8-a3a484d3e079
```

次のコマンドは、選択されたオブジェクトの所有者を、先行する `users` コマンドで番号 2 に対応しているユーザーに変更します。

```
admin> chown %2
```

cp

ファイルとフォルダをレジストリにコピーします。

形式

```
cp [-owner] {URN | % index} [] [- exclude pattern]... [-include pattern]...  
pattern...
```

説明

`cp` コマンドは、フォルダおよびファイルをそれぞれ `RegistryPackage` オブジェクトおよび `ExtrinsicObject` オブジェクトとして、レジストリ内にコピーします。

ファイルやフォルダのコピー元となるローカルファイルシステム上のローカルディレクトリはデフォルトで、管理ツールの起動元である現在のディレクトリになります。-localdir オプションを使えば、管理ツール起動時にローカルディレクトリを変更できます。また、lcd コマンドを使えば、管理ツール起動後にローカルディスクを変更できます。show localdir コマンドを使えば、現在のローカルディレクトリの絶対パスを取得できます。

このコマンドは再帰的です。つまり、ユーザーが特定のディレクトリを指定すると、このコマンドはそのディレクトリの下にあるすべてのファイルとフォルダをコピーします。

オプション

-owner 引数の URN または %index に指定されたユーザーに、コピー先のレジストリオブジェクトの所有者を設定します。これらの引数については、chown コマンドの説明を参照してください。自分以外の所有者を指定するには、管理者のロールを持っている必要があります。

-exclude 指定されたパターンを名前に含むファイル以外のすべてのファイルをコピーします。ここで、*pattern* は、リテラル文字と、特殊文字のアスタリスク (*) (0 個以上の文字を表現) または疑問符 (?) (1 文字を表現) で構成されるパターンです。

このオプションは複数回指定可能です。

-include 指定されたパターンを名前に含むファイルをすべてコピーします。ここで、*pattern* は、リテラル文字と、特殊文字のアスタリスク (*) (0 個以上の文字を表現) または疑問符 (?) (1 文字を表現) で構成されるパターンです。

このオプションは複数回指定可能です。

オペランド

pattern コピー対象のファイルまたはフォルダ。リテラル文字と、特殊文字のアスタリスク (*) (0 個以上の文字を表現) または疑問符 (?) (1 文字を表現) で構成されるパターンを使って指定します。*pattern* は複数個指定可能です。

例

次のコマンドは、ディレクトリ mydir をレジストリにコピーし、その所有者として、先行する users コマンドの番号 4 に対応するユーザーを設定します。

```
admin> cp -owner %4 mydir
```

次のコマンドは、ディレクトリ `mydir` をレジストリにコピーします。ただし、文字列 `.z`、`.c` のいずれかで終わるファイルやディレクトリは除外されます。

```
admin> cp mydir -exclude *.z -exclude *.c
```

echo

特定の文字列をエコーします。

形式

```
echo string
```

説明

`echo` コマンドは、指定された *string* を出力先にエコーします。このコマンドは、管理ツールをバッチモードで実行する場合に `-command` オプション内で指定すると、非常に便利です。

オペランド

string 文字シーケンス。

例

次のコマンドは、「date」と `ls` コマンドの結果をログファイルに出力します。

```
java -jar admin-tool.jar -command "echo "date"; ls" > admin.log
```

help

コマンドに関する情報を表示します。

形式

```
help [command_name]
```

説明

`help` コマンドは、利用可能なコマンドまたは指定された特定のコマンドに関する情報を表示します。

`add` や `show` など、サブコマンドを持つコマンドが指定された場合、`help` コマンドはそのサブコマンドに関する情報を表示します。

引数が指定されなかった場合、`help` コマンドは、すべてのコマンドの使用法に関する情報を表示します。

オペランド

`command_name` 特定の管理ツールコマンドの名前。

例

次のコマンドは、すべてのコマンドの使用法に関する情報を表示します。

```
admin> help
```

次のコマンドは、`lcd` コマンドの使用法に関する情報を表示します。

```
admin> help lcd
```

次のコマンドは、`add` サブコマンドの使用法に関する情報を表示します。

```
admin> help add
```

lcd

ローカルファイルシステム上の現在のディレクトリを変更します。

形式

```
lcd [path_name]
```

説明

`lcd` コマンドは、ローカルファイルシステム上の現在のローカルディレクトリを変更します。

引数が指定されなかった場合、`lcd` コマンドは、現在のディレクトリをユーザーのデフォルトホームディレクトリに変更します。

オペランド

path_name ディレクトリ名。絶対パス、相対パスのいずれを使ってもかまいません。

例

次のコマンドは、現在のローカルディレクトリを `/usr/share` ディレクトリに変更します。

```
admin> lcd /usr/share
```

次のコマンドは、ローカルファイルシステム上の現在のローカルディレクトリをユーザーのデフォルトホームディレクトリに変更します。

```
admin> lcd
```

ls

現在の `RegistryPackage` 内のオブジェクトを一覧表示します。

形式

```
ls { pattern | URN } []
```

説明

引数が指定されなかった場合、`ls` コマンドは、現在の `RegistryPackage` 内のオブジェクトを一覧表示します。*pattern* または *URN* が指定された場合は、現在の `RegistryPackage` 内のオブジェクトのうち、(現在のローカルにおける) 名前または一意の識別子が *pattern* または *URN* に一致するものを一覧表示します。

オペランド

pattern リテラル文字と、特殊文字のアスタリスク (*) (0 個以上の文字を表現) または疑問符 (?) (1 文字を表現) で構成されるパターン。*pattern* は複数個指定可能です。

URN urn:uuid:4a6741e7-4be1-4cfb-960a-e5520356c4fdなどの、urn:で始まるURN。URNは複数個指定可能です。このURNは、オブジェクトの論理識別子ではなく、一意の識別子である必要があります。

例

次のコマンドは、現在のRegistryPackage内のすべてのオブジェクトを一覧表示します。

```
admin> ls
```

次のコマンドは、パターンurn:bird:poultry:chickenに一致する名前を持つかurn:bird:poultry:chickenをIDに持つすべてのオブジェクトを一覧表示します。

```
admin> ls urn:bird:poultry:chicken
```

次のコマンドは、パターン*bird*に一致する名前を持つすべてのオブジェクトを一覧表示します(*bird*が有効なIDである場合、このコマンドは、*bird*をIDに持つオブジェクトも一覧表示する)。

```
admin> ls *bird*
```

次のコマンドは、パターン*bird*に一致する名前を持つか、パターンurn:bird:poultry:chickenに一致する名前を持つか、urn:bird:poultry:chickenをIDに持つような、すべてのオブジェクトを一覧表示します。

```
admin> ls *bird* urn:bird:poultry:chicken
```

pwd

現在のRegistryPackageへのパスを表示します。

形式

pwd

説明

pwdコマンドは、現在のRegistryPackageへの1つまたは複数のパスを、現在のロケールにもっともふさわしい名前を使って表示します。また、そのパスに対するロケールも表示します。

例

```
admin> pwd  
(en_US) /registry/userData
```

quit

管理ツールを終了します。

形式

quit

説明

quit コマンドは管理ツールを終了します。

例

```
admin> quit
```

rm

特定の RegistryPackage からオブジェクトを削除します。

形式

```
rm [-d] [-r] { pattern | URN } []
```

説明

rm コマンドは、現在の RegistryPackage のメンバーオブジェクトのうち、(現在のロケールにおける) 名前が *pattern* または *URN* で指定されたパターンに一致するものを削除します。

一致する RegistryObject が複数の RegistryPackage オブジェクトのメンバーになっている場合、このコマンドは、現在の RegistryPackage とそのオブジェクトとの間の関連付けだけを削除します。そのオブジェクトがレジストリから削除されるのは、ほかの RegistryObject との関連付けが1つも存在しない場合だけです。

一致するメンバーオブジェクト自身がほかのオブジェクトを含む RegistryPackage である場合には、-r、-d のいずれかのオプションが指定されない限り、そのオブジェクトが削除されたり、現在の RegistryPackage とそのメンバー RegistryPackage との間の関連付けが削除されたりすることはありません。

-d オプションと -r オプションが両方とも指定された場合は -d オプションが再帰的に適用されますが、これは、-r の選択対象となるすべてのオブジェクト (とそれらの関連付け) が、ほかの関連付けの有無にかかわらず削除されるようにするためです。

オプション

- d 現在の RegistryPackage と指定された RegistryPackage との間の関連付けを削除します。指定された RegistryPackage が削除されるのは、残りの関連付けが自身のメンバーオブジェクトに対するものだけである場合に限りです。削除された RegistryPackage のメンバーオブジェクトのうち、ほかの HasMember 関連付けのターゲットとして固定されていないものは、レジストリのルートの子としてアクセスできるようになります。
- r 指定された RegistryPackage オブジェクトとその配下にあるすべてのオブジェクトを削除します (ただし、ほかの関連付けを持つオブジェクトが存在する場合を除く)。

オペランド

pattern リテラル文字と、特殊文字のアスタリスク (*) (0 個以上の文字を表現) または疑問符 (?) (1 文字を表現) で構成されるパターン。 *pattern* は複数個指定可能です。

URN urn:uuid:4a6741e7-4be1-4cfb-960a-e5520356c4fd などの、urn: で始まる URN。 *URN* は複数個指定可能です。

例

次のコマンドは、文字列 "stat" を含むすべての RegistryPackage オブジェクトと、それらの配下にあるすべてのオブジェクトを削除します。

```
admin> rm -r *stat*
```

select

SQL の select 文を実行します。

形式

select [SQL]

説明

select コマンドは、コマンド全体を1つの SQL クエリーとして評価することで、指定されたオブジェクトを選択および一覧表示します。引数が指定されなかった場合、このコマンドは、先行する select コマンドまたは `-sqlselect` オプションによって選択されたすべてのオブジェクトを一覧表示します。

オペランド

SQL SQL の select 文 (ただし、先頭の select はすでにコマンド名として存在しているので含めない)。

例

次のコマンドは、レジストリ内のすべての ClassificationScheme オブジェクトを一覧表示します。

```
admin> select s.* from ClassificationScheme s
```

set

特定のプロパティの値を設定します。

形式

set *property value*

説明

`set` コマンドは、管理ツールシェルの特定のプロパティの値を設定します。

ツールがサポートするプロパティと値は、次のとおりです。

```
set debug {true | on | yes | false | off | no}
```

デバッグメッセージの出力を有効または無効にします。

```
set editor string
```

対話型エディタ起動時に管理ツールが使用するコマンドを設定します。UNIX および Linux システムにおけるデフォルト値は `/bin/vi` です。

```
set verbose { true | on | yes | false | off | no }
```

コマンド実行時における冗長メッセージの出力を有効または無効にします。

オペランド

property 次のプロパティのいずれかです。debug、editor、verbose。

value 指定されたプロパティのサポートされている値。詳細については、「説明」を参照してください。

例

次のコマンドは、エディタをデフォルトの `/bin/vi` ではなく、`/usr/bin/vi` に設定します。

```
admin> set editor /usr/bin/vi
```

次のコマンドは、デバッグをオンにします。

```
admin> set debug true
```

次のコマンドは、冗長な出力をオフにします。

```
admin> set verbose off
```

show

特定のプロパティの値を表示します。

形式

show [*property*]

説明

show コマンドは、管理ツールシェルの特定のプロパティの値を表示します。

引数が指定されなかった場合、このコマンドはすべてのプロパティの値を表示します。

このコマンドがサポートするプロパティは、次のとおりです。

debug	デバッグ出力が有効になっているかどうか。
editor	対話型エディタ起動時に管理ツールが使用するエディタ。
exception	直前に実行されたコマンドで発生した例外スタックトレース (存在する場合のみ)。
localdir	ローカルファイルシステム上の現在のディレクトリ。このプロパティを設定するには <code>lcd</code> コマンドを使用します。詳細については、53 ページの「 <code>lcd</code> 」を参照してください。
locale	現在のロケール。
verbose	冗長出力が有効になっているかどうか。

オペランド

property 現在の値を表示すべきプロパティ。プロパティ `exception` と `locale` については、表示は可能ですが、`set` コマンドによる値の設定は行えません。

例

次のコマンドは、直前のコマンドで発生した例外を表示します。

```
admin> show exception
```

users

現在の User オブジェクトを一覧表示します。

形式

users

説明

`users` コマンドは、現在レジストリ内に存在している `User` オブジェクトを一覧表示します。

出力の書式は次のとおりです。

```
%index: URN lastname, firstname middlename
```

出力内の `index` は数値ですが、`chown` コマンドまたは `cp` コマンドを実行する際にこの数値をパーセント記号 (%) も含めて使用すれば、特定のユーザーを参照することができます。 `lastname`、`firstname`、および `middlename` は、ユーザーの姓、名、およびミドルネームです。

例

次のコマンドは、現在のユーザーを表示します。

```
admin> users
%0: urn:freebxml:registry:predefinedusers:registryoperator Operator, Registry
%1: urn:freebxml:registry:predefinedusers:registryguest Guest, Registry
%2: urn:freebxml:registry:predefinedusers:farrukh Najmi, Farrukh Salahudin
%3: urn:freebxml:registry:predefinedusers:nikola Stojanovic, Nikola
%4: urn:uuid:799cc524-b7cd-4e51-8b34-d93b79ac52de User, Test
%5: urn:uuid:85428d8e-1bd5-473b-a8c8-b9d595f82728 Parker, Miles
```


索引

A

add association コマンド, 41-42
add user コマンド, 43-48
AdhocQuery オブジェクト, Web コンソールへの追加, 26-31
-alias コマンド行オプション, 37
Application Server 管理コンソール, 20
Application Server ドメイン
truststore へのルート証明書の追加, 21-22
管理, 19-21
停止および再起動, 20-21

C

cd コマンド, 48-49
chown コマンド, 49-50
-command コマンド行オプション, 37
cp コマンド, 50-52
-create コマンド行オプション, 37

D

-debug コマンド行オプション, 37
debug プロパティ
値の表示, 59-60
設定, 58-59

E

echo コマンド, 52

editor プロパティ

値の表示, 59-60

設定, 58-59

exception プロパティ, 値の表示, 59-60

ExternalLink オブジェクト, URL の検証を可能にする, 23-24

H

help コマンド, 52-53
-help コマンド行オプション, 38

K

-keypass コマンド行オプション, 38

L

lcd コマンド, 53-54
-localdir コマンド行オプション, 38
-locale コマンド行オプション, 38
locale プロパティ, 値の表示, 59-60
ls コマンド, 54-55

P

pwd コマンド, 55-56

Q

quit コマンド, 56

R

RegistryPackage オブジェクト

作成, 37

内容の一覧表示, 54-55

パスの表示, 55-56

メンバーオブジェクトの削除, 56-57

RegistryPackage からのオブジェクトの削除, 56-57

RegistryPackage の場所, 変更, 48-49

-registry コマンド行オプション, 38

registry ドメイン

truststore へのルート証明書の追加, 21-22
管理, 19-21

停止および再起動, 20-21

rm コマンド, 56-57

-root コマンド行オプション, 38

S

select コマンド, 58

ServiceBinding オブジェクト, URL の検証を可能にする, 23-24

Service Registry の設定, 15-23

Service Registry のデータベース, バックアップ
および復元, 32-33

set コマンド, 58-59

show コマンド, 59-60

-sqlselect コマンド行オプション, 38

SQL select 文, 実行, 38

SQL 文, 実行, 58

U

users, 一覧表示, 60-61

users コマンド, 60-61

V

-verbose コマンド行オプション, 38

verbose プロパティ

値の表示, 59-60

設定, 58-59

-v コマンド行オプション, 38

W

Web コンソール, 定義済みクエリーの追加, 26-31

い

インストールプロパティ, 16-17

お

大文字と小文字の区別, 40-41

か

外部 web サイト, アクセス許可, 23-24
管理コンソール, Application Server, 20

管理者, 作成, 24-25

管理者の作成, 24-25

管理ツール

概要, 35

起動, 36-39

コマンド行オプション, 37-39

使用の有効化, 18

停止, 56

管理ツールの起動, 36-39

管理ツールの終了, 56

管理ツールの停止, 56

関連付け, レジストリへの追加, 41-42

け

現在のディレクトリ, 変更, 53-54

こ

コマンド

- add association, 41-42
- add user, 43-48
- cd, 48-49
- chown, 49-50
- cp, 50-52
- echo, 52
- help, 52-53
- lcd, 53-54
- ls, 54-55
- pwd, 55-56
- quit, 56
- rm, 56-57
- select, 58
- set, 58-59
- show, 59-60
- users, 60-61

コマンド行オプション, 37-39

- alias, 37
- command, 37
- create, 37
- debug, 37
- help, 38
- keypass, 38
- localdir, 38
- locale, 38
- registry, 38
- root, 38
- sqlselect, 38
- v, 38
- verbose, 38

た

対話型モード, 36-37

て

定義済みクエリー, Web コンソールへの追加, 26-31
ディレクトリ, 変更, 53-54

は

バッチモード, 36

ひ

表示, プロパティー値, 59-60

ふ

ファイルシステム、ローカル
現在のディレクトリの変更, 53-54
ベースディレクトリ, 38
ファイルとフォルダ, レジストリへのコピー, 50-52
プロキシのホストとポート, 設定, 23-24
プロパティー値
設定, 58-59
表示, 59-60
プロパティー値の設定, 58-59

ほ

ポート, Service Registry のデフォルト, 19-20

ゆ

ユーザー, レジストリへの追加, 43-48
ユーザー名, コマンド行での指定, 37

よ

用語集, リンク先, 10

る

ルート証明書, registry ドメインの truststore への追加, 21-22

れ

例外, 表示, 39
レジストリオブジェクト
一覧表示, 54-55
所有者の変更, 49-50
特定, 39-40

レジストリへのファイルとフォルダのコピー
 , 50-52

ろ

ロケール, 名前指定への影響, 40

わ

ワイルドカード, 使用, 39-40