BEA WebLogic Server™

WebLogic リソースのセキュリティ

BEA WebLogic Server バージョン 7.0
マニュアルの改訂 :2003 年 7 月 18 日

著作権

限定的権利条項

商標または登録商標

WebLogic リソースのセキュリティ

パート番号	マニュアルの改訂	ソフトウェアのバージョン
なし	2003 年 7 月 18 日	BEA WebLogic Server バージョン 7.0

目次

5. セキュリティ ポリシー

6. 例 : Administration Console を使用した URL (Web) リソースの保護

このマニュアルの内容

このマニュアルでは、さまざまなタイプの WebLogic リソースを紹介し、WebLogic Server を使用してそれらのリソースを保護するための情報を提供します。

このマニュアルの構成は次のとおりです。

- 第 1 章「WebLogic リソースの保護の概要」では、このマニュアルに関する予備的な情報 (対象読者など)、および WebLogic リソースの保護の概要を示します。また、手順を説明した節にすぐに進むユーザのために「主な手順」の節も設けてあります。
- 第 2 章「WebLogic リソースのタイプ」では、各タイプの WebLogic リソースと、一般的で複雑な WebLogic リソースの保護について説明します。
- 第 3 章「ユーザとグループ」では、ユーザおよびグループについて解説し、WebLogic Server のデフォルト グループに関する情報も示します。この節では、WebLogic Server Administration Console でユーザおよびグループを操作する方法についても、手順を追って説明します。
- 第 4 章「セキュリティ ロール」では、セキュリティ ロールについて解説し、WebLogic Server のデフォルト グローバル ロールに関する情報も示します。また、グローバル ロールとスコープ ロールの違いと、セキュリティ ロールのコンポーネントについても説明します。さらに、WebLogic Server Administration Console でグローバル ロールおよびスコープ ロールを操作する方法についても、手順を追って説明します。
- 第 5 章「セキュリティ ポリシー」では、セキュリティ ポリシーについて解説し、WebLogic Server のデフォルトのセキュリティ ポリシーに関する情報も示します。また、セキュリティ ポリシーのコンポーネントについて説明し、Administration Console でセキュリティ ポリシーを扱う方法についても、手順を追って説明します。
- 第 6 章「例 : Administration Console を使用した URL (Web) リソースの保護」では、Administration Console を使用して、さまざまな URL (Web) リソースを保護する方法を紹介します。

- 第 7 章「例 : エンタープライズ JavaBean (EJB) リソースの保護」では、Administration Console を使用して、さまざまなエンタープライズ JavaBean (EJB) リソースを保護する方法を紹介します。
- 第 8 章「例 : basicauth Web アプリケーションのセキュリティ コンフィグレーションのコピーと再初期化」では、既存のデプロイメント記述子から Web アプリケーションのセキュリティ コンフィグレーションをコピーし、Administration Console を使用してセキュリティ ポリシーを変更してから、デプロイメント記述子で指定していたセキュリティ コンフィグレーションに再初期化する手順について説明します。

対象読者

このマニュアルは、主にサーバ管理者を対象としています。**サーバ管理者**は、アプリケーション設計者と密接に連携しながら、サーバおよびサーバ上で動作するアプリケーションのセキュリティ方式の設計、潜在的なセキュリティ リスクの特定、およびセキュリティ上の問題を防止するコンフィグレーションの提案を行います。関連する責務として、重要なプロダクション システムの保守、セキュリティ レルムのコンフィグレーションと管理、サーバ リソースとアプリケーション リソースへの認証および認可方式の実装、セキュリティ機能のアップグレード、およびセキュリティ プロバイダのデータベースの保守などが含まれる場合もあります。サーバ管理者は、Web アプリケーションと EJB のセキュリティ、公開鍵セキュリティ、および SSL を含む、Java セキュリティ アーキテクチャについて深い知識を備えています。

このマニュアルは、WebLogic Server Administration Console を使用するサーバ管理者を対象としており、『WebLogic Security の管理』と合わせて、WebLogic Server デプロイメントのセキュリティを正しくコンフィグレーションするためのものです。

e-docs Web Site

BEA 製品のドキュメントは、BEA の Web サイトで入手できます。BEA のホームページで [製品のドキュメント] をクリックします。

このマニュアルの印刷方法

Web ブラウザの [ファイル | 印刷] オプションを使用すると、Web ブラウザからこのマニュアルを一度に 1 章ずつ印刷できます。

このマニュアルの PDF 版は、WebLogic Server の Web サイトで入手できます。PDF を Adobe Acrobat Reader で開くと、マニュアルの全体 (または一部分) を書籍の形式で印刷できます。PDF を表示するには、WebLogic Server ドキュメントのホーム ページを開き、[ドキュメントのダウンロード] をクリックして、印刷するマニュアルを選択します。

Adobe Acrobat Reader は Adobe の Web サイト (http://www.adobe.co.jp) で無料で入手できます。

関連情報

BEA の Web サイトでは、WebLogic Server の全マニュアルを提供しています。WebLogic リソースを保護するサーバ管理者の参考となる WebLogic Server マニュアルには、このマニュアルの他に以下のものがあります。

- 『WebLogic Security の管理』
- 『WebLogic Security プログラマーズ ガイド』の「Web アプリケーションのセキュリティ対策」、「エンタープライズ JavaBean (EJB) のセキュリティ対策」、および「Java セキュリティを使用しての WebLogic リソースの保護」
- 『管理者ガイド』の「システム管理操作の保護」
- 『WebLogic jCOM プログラマーズ ガイド』の「アクセス制御のコンフィグレーション」(COM リソース)
- 『WebLogic J2EE コネクタ アーキテクチャ』の「セキュリティ」(EIS リソース)
- 『WebLogic Web サービス プログラマーズ ガイド』の「セキュリティのコンフィグレーション」(Web サービス リソース)

他のセキュリティ関連のマニュアルについては、セキュリティのページに記載されています。

サポート情報

BEA のドキュメントに関するユーザからのフィードバックは弊社にとって非常に重要です。質問や意見などがあれば、電子メールで docsupport-jp@beasys.com までお送りください。寄せられた意見については、ドキュメントを作成および改訂する BEA の専門の担当者が直に目を通します。

電子メールのメッセージには、ご使用のソフトウェア名とバージョン名、およびマニュアルのタイトルと作成日付をお書き添えください。本バージョンの BEA WebLogic Server について不明な点がある場合、または BEA WebLogic Server のインストールおよび動作に問題がある場合は、BEA WebSupport (www.bea.com) を通じて BEA カスタマ サポートまでお問い合わせください。カスタマ サポートへの連絡方法については、製品パッケージに同梱されているカスタマ サポート カードにも記載されています。

カスタマ サポートでは以下の情報をお尋ねしますので、お問い合わせの際はあらかじめご用意ください。

- お名前、電子メール アドレス、電話番号、ファクス番号
- 会社の名前と住所
- お使いの機種とコード番号
- 製品の名前とバージョン
- 問題の状況と表示されるエラー メッセージの内容

表記規則

このマニュアルでは、全体を通して以下の表記規則が使用されています。

表記法	適用
〔Ctrl〕+〔Tab〕	複数のキーを同時に押すことを示す。
斜体	強調または書籍のタイトルを示す。

等幅テキスト	コード サンプル、コマンドとそのオプション、データ構造体とそのメンバー、データ型、ディレクトリ、およびファイル名とその拡張子を示す。等幅テキストはキーボードから入力するテキストも示す。
	例：
	`import java.util.Enumeration;`
	`chmod u+w *`
	`config/examples/applications`
	`.java`
	`config.xml`
	`float`

斜体の等幅テキスト	プレースホルダを示す。 例： `String CustomerName;`

大文字の等幅テキスト	デバイス名、環境変数、および論理演算子を示す。
	例：
	`LPT1`
	`BEA_HOME`
	`OR`

{ }	構文の中で複数の選択肢を示す。

表記法	適用
[ ]	構文の中で任意指定の項目を示す。 例: `java utils.MulticastTest -n name -a address [-p portnumber] [-t timeout] [-s send]`
|	構文の中で相互に排他的な選択肢を区切る。 例: `java weblogic.deploy [list|deploy|undeploy|update] password {application} {source}`
...	コマンドラインで以下のいずれかを示す。 ■ 引数を複数回繰り返すことができる ■ 任意指定の引数が省略されている ■ パラメータや値などの情報を追加入力できる
. . .	コード サンプルまたは構文で項目が省略されていることを示す。

1　WebLogic リソースの保護の概要

以下の節では、WebLogic リソースの保護についての概要を示します。

このガイドの対象読者

このマニュアルは、主にサーバ管理者を対象としています。**サーバ管理者**は、アプリケーション設計者と密接に連携しながら、サーバおよびサーバ上で動作するアプリケーションのセキュリティ方式の設計、潜在的なセキュリティ リスクの特定、およびセキュリティ上の問題を防止するセキュリティ コンフィグレーションの提案を行います。関連する責務として、重要なプロダクション システムの保守、セキュリティ レルムのコンフィグレーションと管理、サーバ リソースとアプリケーション リソースへの認証および認可方式の実装、セキュリティ機能のアップグレード、およびセキュリティ プロバイダのデータベースの保守などが含まれる場合もあります。サーバ管理者は、Web アプリケーションと EJB のセキュリティ、公開鍵セキュリティ、および SSL を含む、Java セキュリティ アーキテクチャについて深い知識を備えています。

このマニュアルは、WebLogic Server Administration Console を使用するサーバ管理者を対象としており、『WebLogic Security の管理』と合わせて、WebLogic Server デプロイメントのセキュリティを正しくコンフィグレーションするためのものです。

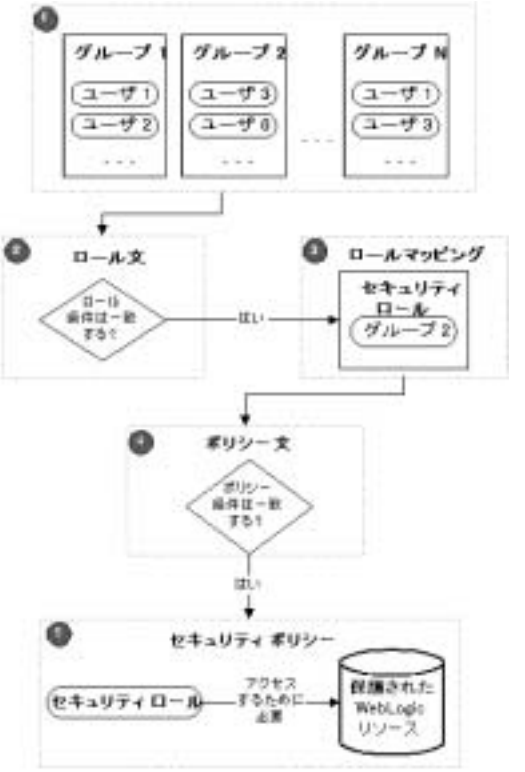
用語と概念

WebLogic Server のセキュリティには、理解しておく必要のある固有の用語や概念が多数あります。これらの用語と概念は WebLogic Server のセキュリティに関するマニュアルに登場しますが、用語については『WebLogic Security の紹介』の「用語」の節で、概念については「セキュリティの基礎概念」の節で定義されています。

WebLogic リソースの保護の概要

図 1-1 に、WebLogic リソースを保護するための全体的なプロセスを示し、その後簡単に説明します。

図 1-1　WebLogic リソースの保護



1. 管理者は、組織的な境界を表すグループにユーザを静的に割り当てます。同じユーザは複数のグループのメンバーになることができます。図 1-1 に、それぞれ 2 人のユーザで構成される 3 つのグループを示します。ユーザ 1 とユーザ 3 は複数のグループのメンバーです。

   注意： ユーザをグループに割り当てると、多数のユーザを扱う管理者の作業を効率化できます。

2. 管理者は、自社の確立されたビジネス手順に基づいてロール文を作成します。ロール文には、特定のグループに付与されるセキュリティ ロールを指定するロール条件が定義されます。

3. 実行時に、WebLogic Security サービスはロール条件とグループを比較して、そのグループにセキュリティ ロールを動的に付与するかどうかを決定します。このプロセスを、**ロール マッピング**と呼びます。図 1-1 では、グループ 2 だけにセキュリティ ロールを付与されています。

   注意： 個々のユーザにセキュリティ ロールを付与することもできますが、これは一般的ではありません。

4. 管理者は、自社の確立されたビジネス手順に基づいてポリシー文を作成します。ポリシー文には、特定のセキュリティ ロールに付与される保護対象 WebLogic リソースへのアクセス権を指定するポリシー条件が定義されます。

5. 実行時に、WebLogic Security サービスはセキュリティ ポリシーと WebLogic リソースを使用して、保護されているリソースへのアクセスを許可するかどうかを決定します。セキュリティ ロールを付与されたグループのメンバーであるユーザだけが、WebLogic リソースにアクセスできます。したがって、図 1-1 では、ユーザ 3 とユーザ 6 はグループ 2 のメンバーなので保護対象 WebLogic リソースにアクセスできます。

WebLogic リソースの保護：主な手順

WebLogic リソースを保護する主な手順は以下のとおりです。

1. 保護する WebLogic リソースを決定します。詳細については、「WebLogic リソースのタイプ」を参照してください。

2. URL (Web) またはエンタープライズ JavaBean (EJB) リソースを保護する場合は、次の手順に従います。

   a. 使用する方法を決定します。詳細については、2-7 ページの「URL リソースおよび EJB リソースを保護する方法」を参照してください。

   b. セキュリティ コンフィグレーションのオーバライドを避けるため、URL および EJB リソースの保護に関する重要事項に目を通しておきます。詳細については、2-10 ページの「URL リソースおよび EJB リソースを保護するための前提条件」を参照してください。

   c. WebLogic Server Administration Console を使用して URL または EJB リソースを保護する場合は、手順 3 の指示に従います。

d. デプロイメント記述子を使用して URL または EJB リソースを保護する場合は、『WebLogic Security プログラマーズ ガイド』の「Web アプリケーションでの宣言によるセキュリティの使用」と「EJB での宣言によるセキュリティの使用」をそれぞれ参照してください。

e. URL または EJB リソースの初回のデプロイ時に、既存のデプロイメント記述子からセキュリティ コンフィグレーションをコピーする場合や、URL または EJB リソースのセキュリティ コンフィグレーションを、デプロイメント記述子で指定された元の状態に再初期化する場合は、2-18 ページの「組み合わせた方法による URL および EJB リソースの保護」の指示に従います。

3. Administration Console を使用して WebLogic リソースを保護します。次の手順に従います。

a. セキュリティ ロールの付与の対象となるユーザおよびグループ (個人または個人の集合の表現) を作成します。手順については、3-2 ページの「ユーザの作成」と 3-7 ページの「グループの作成」を参照してください。

b. セキュリティ ロールを作成します。セキュリティ ロールは動的に計算される特権で、特定の条件に基づいてユーザまたはグループに付与され、WebLogic リソースへのアクセスを制限するために使用されます。手順については、「セキュリティ ロール」を参照してください。

注意： WebLogic リソースを保護するためには、多くのユーザを抱える管理者の作業を効率化できるので、(ユーザまたはグループよりも) セキュリティ ロールを作成および使用することをお勧めします。

c. セキュリティ ポリシー (WebLogic リソースとユーザ、グループ、またはセキュリティ ロールとの関連付け) を作成します。セキュリティ ポリシーは、だれが WebLogic リソースへのアクセスを許可されるのかを指定するものです。手順については、「セキュリティ ポリシー」を参照してください。

2 WebLogic リソースのタイプ

WebLogic リソースは、基底の WebLogic Server エンティティを表します。これらのエンティティは、セキュリティ ロールとセキュリティ ポリシーを使用して、権限のないアクセスから保護できます。

WebLogic リソースは階層化されています。このため、セキュリティ ロールとセキュリティ ポリシーは自由なレベルで定義できます。たとえば、エンタープライズ アプリケーション (EAR) 全体、複数の EJB を含むエンタープライズ JavaBean (EJB) JAR、その JAR 内の特定の EJB、その EJB 内の単一のメソッドなどに対してセキュリティ ロールとセキュリティ ポリシーを定義できます。

WebLogic Server のリソースについて以下の節で説明します。

管理リソース

管理リソースは、管理タスクの実行をユーザに許可する WebLogic リソースです。WebLogic Server Administration Console、weblogic.Admin ツール、および MBean API を保護する場合は、管理リソースを保護します。

管理リソースはスコープを制限されています。現在、WebLogic Server Administration Console を使用して、管理リソースに対するユーザ ロックアウト操作のみを保護することができます。この操作は WebLogic Server 6.x と互換性があり、セキュリティ要件を満たすユーザはロックされたアカウントのロックを解除できます。ユーザ ロックアウトの詳細については、『WebLogic Security の管理』の「ユーザ アカウントの保護」を参照してください。

アプリケーション リソース

アプリケーション リソースは、EAR (エンタープライズ アプリケーション アーカイブ) ファイルにパッケージ化されたエンタープライズ アプリケーションを表す WebLogic リソースです。その他の WebLogic リソースと違い、アプリケーション リソースの階層は格納するための仕組みであり、タイプ別の階層ではありません。エンタープライズ アプリケーション (たとえば EJB リソース、URL リソース、Web サービス リソース) を構成する複数の WebLogic リソースをまとめて保護する場合は、アプリケーション リソースを保護します。つまり、エンタープライズ アプリケーションを保護すると、そのアプリケーション内のすべての WebLogic リソースがセキュリティ コンフィグレーションを継承します。

エンタープライズ アプリケーション (EAR) を構成する WebLogic リソースを個別に保護することもできます。個々の WebLogic リソースのセキュリティ コンフィグレーションは、エンタープライズ アプリケーションから継承されたセキュリティ コンフィグレーションに優先します。

EIS (エンタープライズ情報システム) リソース

J2EE コネクタは、WebLogic Server などのアプリケーション サーバで EIS (エンタープライズ情報システム) に接続するために使用されるシステム レベルのソフトウェア ドライバです。BEA では、EIS ベンダおよびサードパーティ アプリケーション開発者が開発し、Sun Microsystems の J2EE プラットフォーム仕様、バージョン 1.3 に準拠しているアプリケーション サーバにデプロイ可能なコネクタをサポートしています。コネクタ (**リソース アダプタ**とも呼ばれる) には、Java、および必要に応じて EIS との対話に必要なネイティブ コンポーネントが含まれます。

EIS (エンタープライズ情報システム) リソースは、コネクタとして設計された WebLogic リソースです。EIS へのアクセスを保護するには、すべてのコネクタまたは個々のコネクタに対してセキュリティ ポリシーおよびセキュリティ ロールを作成します。

注意： EIS リソースの保護については、このマニュアルと『WebLogic J2EE コネクタ アーキテクチャ』の「セキュリティ」で取り上げています。

EIS リソースで使用する資格マップの作成手順については、『WebLogic Security の管理』の「エンタープライズ情報システムでのシングル サインオン」を参照してください。

COM リソース

WebLogic jCOM は、WebLogic Server でデプロイされる Java/J2EE オブジェクトと、Microsoft Office 製品ファミリ、Visual Basic オブジェクトおよび C++ オブジェクト、その他のコンポーネント オブジェクト モデル / 分散コンポーネント オブジェクト モデル (COM/DCOM 準拠) 環境で使用できる Microsoft ActiveX コンポーネントとの間で双方向アクセスを可能にするソフトウェア ブリッジです。

COM リソースは、Microsoft のフレームワークに従ってプログラム コンポーネント オブジェクトとして設計された WebLogic リソースです。BEA の双方向 COM-Java (jCOM) ブリッジ ツールを使用してアクセスする COM コンポーネントを保護するには、複数の COM クラスを含むパッケージまたは個々の COM クラスに対してセキュリティ ポリシーおよびセキュリティ ロールを作成します。

注意： COM リソースの保護については、このマニュアルと『WebLogic jCOM プログラマーズ ガイド』の「アクセス制御のコンフィグレーション」で取り上げています。

JDBC (Java Database Connectivity) リソース

JDBC (Java DataBase Connectivity) リソースは、JDBC 関連の WebLogic リソースです。JDBC アクセスを保護するには、接続プール全体、個々の接続プール、またはマルチプールに対してセキュリティ ポリシーおよびセキュリティ ロールを作成します。個々の接続プールを保護する場合、接続プールに対するすべての操作を保護する方法と、以下のいずれかの操作を指定する方法があります。

- *admin* - admin 操作では、`JDBCConnectionPoolRuntimeMBean` に対して、`clearStatementCache`、`destroy`、`disableDroppingUsers`、`disableFreezingUsers`、`enable`、`forceDestroy`、`forceShutdown`、`forceSuspend`、`getProperties`、`poolExists`、`resume`、`shutdown`、`shutdownHard`、`shutdownSoft`、および `suspend` メソッドを呼び出すことができます。
- *reserve* - アプリケーションは、接続プールが指すデータ ソースをルックアップしてから getConnection を呼び出すことで、接続プール内の接続を予約します。
- *shrink* - 接続プールを、予約済みの最大接続数または初期サイズに縮小します。
- *reset* - データベースとの物理接続をシャットダウンしてから接続を再確立して、データベース接続プールをリセットします。これにより、接続プール内

の各接続のステートメント キャッシュもクリアされます。正常に動作している接続プールのみをリセットできます。

注意： セキュリティ ポリシーでマルチプール内の接続プールへのアクセスを制御する場合、アクセスのチェックは、JDBC リソース階層の 2 つのレベルで実行されます (マルチプールのレベルで 1 回、個々の接続プールのレベルで 1 回)。こうした二重のチェックをすべてのタイプの WebLogic リソースで実行することで、セキュリティ レベルの高い方のセキュリティ ポリシーがアクセスを制御することになります。

JMS (Java Message Service) リソース

JMS (Java Messaging Service) リソースは、JMS 関連の WebLogic リソースです。JMS の送り先を保護するには、送り先全体 (JMS キューおよび JMS トピック) または JMS サーバの個々の送り先に対するセキュリティ ポリシーおよびセキュリティ ロールを作成します。JMS サーバの特定の送り先を保護する場合、送り先に対するすべての操作を保護する方法と、JMS キューに対する送信、検索、または受信操作を指定したり、JMS トピックに対する送信および受信操作を指定したりする方法があります。

JNDI (Java Naming and Directory Interface) リソース

JNDI は、LDAP (Lightweight Directory Access Protocol) や DNS (Domain Name System) など、既存のさまざまなネーミング サービスに対する共通インタフェースを提供します。これらのネーミング サービスは、名前をオブジェクトに関連付けて、名前でオブジェクトをルックアップできるようにするバインディングを保持します。JNDI を使用すると、分散アプリケーション内のコンポーネント同士が互いを見つけることができます。

JNDI は、特定のネーミング サービスまたはディレクトリ サービスの実装とは無関係に定義されています。JNDI では、さまざまな新しいサービスや既存のサービスにアクセスするための多くのメソッドを使用できます。このサポートでは、標準サービス プロバイダ インタフェース (SPI) 規約を使用して JNDI フレームワークに任意のサービス プロバイダ実装をプラグインできます。

JNDI (Java Naming and Directory Interface) リソースは、業界標準の JNDI SPI を使用して、異種エンタープライズ ネーミング サービスおよびディレクトリ サービスとの接続を可能にする WebLogic リソースです。JNDI ツリーへのアクセスを保護するには、JNDI ツリー全体またはツリーの個々のブランチに対するセキュリティ ポリシーおよびセキュリティ ロールを作成します。その際、すべての操作を保護することも、操作をルックアップ、変更、または一覧表示に限定することもできます。

サーバ リソース

多くのエンジニアリング チームでは、管理責任を複数のロールに分けています。アプリケーションまたはモジュールをデプロイするパーミッションは 1 人か 2 人のチーム メンバーにしか付与しないが、サーバ コンフィグレーションを参照するパーミッションはチーム メンバー全員に付与する、というような設定がプロジェクトごとに行われます。セキュリティ ポリシーで説明されているように、WebLogic Server では、管理者だけがセキュリティ ポリシーを使用して WebLogic リソースを保護できるようにすることで、このような責任の分担が可能になります。一般に、セキュリティ ポリシーはユーザまたはユーザ グループに特定のセキュリティ ロールが付与されているかどうかに基づます。

サーバ リソースは、WebLogic Server のインスタンスに関連する WebLogic リソースです。このタイプの WebLogic リソースには、サーバの起動、終了、ロック、およびロック解除が含まれているので、ほとんどの管理者が対話する WebLogic リソースとなります。その他の WebLogic リソースと同様に、サーバ リソースとその操作はセキュリティ ポリシーを使用して保護されます。ただし、サーバのコンフィグレーションは MBean を通じて公開されるので、管理者が MBean 操作にアクセスする方法に適用する保護措置もサーバ リソースに設定されます。

注意： サーバ リソースの保護については、このマニュアルと『管理者ガイド』の「システム管理操作の保護」で取り上げています。

URL (Web) リソースと EJB (エンタープライズ JavaBean) リソース

URL (Web) リソースは、Web アプリケーションに関連する WebLogic リソースです。Web アプリケーションを保護するには、WAR (Web アプリケーション アーカイブ) ファイル、または Web アプリケーションの個々のコンポーネント (サーブレットや JSP など) に対するセキュリティ ポリシーおよびセキュリティ ロールを作成します。**EJB (エンタープライズ JavaBean) リソース**は、EJB に関連する WebLogic リソースです。EJB を保護するには、EJB JAR、EJB JAR 内の個々の EJB、または EJB の個々のメソッドに対するセキュリティ ポリシーおよびセキュリティ ロールを作成します。

Java 2 Enterprise Edition (J2EE) プラットフォームは Web アプリケーションおよび EJB のセキュリティをデプロイメント記述子で標準化しているので、WebLogic Server はこの標準メカニズムをセキュリティ サービスに統合して、URL リソースおよび EJB リソースの保護に 2 通りの方法を選択できるようにしています。選択する方法によって、実行する手順が変わり、WebLogic Server Administration Console で別の事前設定が必要になります。詳細については、それぞれ 2-7 ページの「URL リソースおよび EJB リソースを保護する方法」と 2-10 ページの「URL リソースおよび EJB リソースを保護するための前提条件」を参照してください。

注意： このマニュアルで説明されている EJB リソースに関する手順は、メッセージ駆動型 Bean (MDB) にも適用されます。

URL リソースおよび EJB リソースを保護する方法

以下の節では、URL (Web) リソースおよび EJB (エンタープライズ JavaBean) リソースを保護するためのさまざまな方法を詳しく説明します。

- 2-8 ページの「デプロイメント記述子を使用する」
- 2-9 ページの「2 つの方法を組み合わせる」

WebLogic Server Administration Console を使用する

URL (Web) リソースおよびエンタープライズ JavaBean (EJB) リソースを保護する第 1 の方法は、WebLogic Server Administration Console を使用するものです。この方法の主な利点は、セキュリティ管理が統合されていることです。組織的なセキュリティ要件が変化した場合に、開発者が複数のデプロイメント記述子を変更する必要はなく、管理者が一元的なグラフィカル ユーザ インタフェースから、すべてのセキュリティ コンフィグレーションを変更できます。ユーザ、グループ、セキュリティ ロール、およびセキュリティ ポリシーは、すべて Administration Console を使用して定義できます。結果として、更新されたセキュリティ要件に基づく変更のプロセスが効率的になります。

Web サービス リソースを除き、すべてのタイプの WebLogic リソースをこの方法で保護できます。このため、このマニュアルで説明する WebLogic リソースの保護手順は、特に Administration Console のユーザに向けて書かれています。

デプロイメント記述子を使用する

URL リソースおよび EJB リソースを保護する第 2 の方法は、Java 2 Enterprise Edition (J2EE) のデプロイメント記述子と WebLogic デプロイメント記述子を使用するものです。この方法の主な利点は、URL と EJB に宣言的なセキュリティを追加するための一般的かつ標準的な方法であることです。多くの企業にとって一般的な方法でもあります。この方法の場合、ユーザとグループは WebLogic Server Administration Console で定義できますが、セキュリティ ロールとセキュリティ ポリシーは `web.xml`、`weblogic.xml`、`ejb-jar.xml`、および `weblogic-ejb-jar.xml` デプロイメント記述子で定義されます。

注意： この方法を使用する場合、デプロイメント記述子はテキスト エディタで編集することも、Administration Console で編集することもできます。Administration Console の使い方の詳細については、「Web アプリケー

ション デプロイメント記述子エディタ (war)」および『WebLogic エンタープライズ JavaBeans プログラマーズ ガイド』の「EJB デプロイメント記述子エディタの使用」を参照してください。

WebLogic Server 7.0 SP02 では、<global-role/> という特別なタグが導入されました。このタグを使用すると、セキュリティ ロール マッピングをデプロイメント記述子または Administration Console のどちらで定義するかをロールごとに指定できます。URL または EJB リソースでのこのタグの使用については、『WebLogic Security プログラマーズ ガイド』の「Web アプリケーションでの <global-role/> タグの使用」と「EJB での <global-role/> タグの使用」をそれぞれ参照してください。

この方法では、URL および EJB リソースのみを保護できます。デプロイメント記述子の使い方については、『WebLogic Security プログラマーズ ガイド』の「Web アプリケーションでの宣言によるセキュリティの使用」および「EJB での宣言によるセキュリティの使用」を参照してください。

2 つの方法を組み合わせる

現在デプロイメント記述子を使用して URL リソースと EJB リソースを保護している組織では、WebLogic Server Administration Console の統一されたセキュリティ管理機能を利用することもできます。このような場合、Web アプリケーションまたは EJB モジュールの初回のデプロイ時に、既存のデプロイメント記述子からセキュリティ コンフィグレーションをコピーするように Administration Console に指示することができます。セキュリティ コンフィグレーションをコピーすると、以降の更新では Administration Console を使用できます。また、この組み合わせた方法を使用して、URL および EJB リソースのセキュリティ コンフィグレーションを、デプロイメント記述子で指定されている状態に再初期化することもできます。

警告： 組み合わせた方法を使用する場合、URL (Web) および EJB リソースのセキュリティ コンフィグレーションはオーバライドされる可能性があります。したがって、組み合わせた方法を使用する場合は、その Web アプリケーションや EJB の適切なセキュリティ コンフィグレーションが用意されていることを十分に確認する必要があります。重要な情報については、2-10 ページの「URL リソースおよび EJB リソースを保護するための前提条件」を参照してください。

組み合わせた方法の詳細については、第 8 章「例 : basicauth Web アプリケーションのセキュリティ コンフィグレーションのコピーと再初期化」を参照してください。

URL リソースおよび EJB リソースを保護するための前提条件

WebLogic Server Administration Console またはデプロイメント記述子のいずれかを使用して URL および EJB リソースを保護する場合でも、WebLogic Server の重要な 2 つの設定について理解しておく必要があります。これらの設定を理解していないと、セキュリティ コンフィグレーションが不適切なものになったり、失われたりすることがあります。

URL または EJB リソースを保護する前に、以下の節を読んでおいてください。

- 2-10 ページの「fullyDelegateAuthorization フラグについて」
- 2-11 ページの「fullyDelegateAuthorization フラグの変更方法」
- 2-15 ページの「[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスについて」
- 2-16 ページの「[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスの設定変更」
- 2-16 ページの「2 つの設定の相互作用について」

fullyDelegateAuthorization フラグについて

パフォーマンスを制御するために、WebLogic Server Administration Console では WebLogic Security サービスがセキュリティ チェックを実行する方法を指定する必要があります。これを指定するには、WebLogic Server の起動時に設定するコマンドライン引数である `fullyDelegateAuthorization` フラグを使用します。

注意 : WebLogic Server 7.0 SP3 では、コマンドライン引数を使用する代わりに、WebLogic Server Administration Console でこの設定を指定できるようになりました。詳細については、2-14 ページの「[ロールとポリシーのチェック対象] 設定」を参照してください。

fullyDelegateAuthorization フラグの値が false の場合、WebLogic Security サービスは、関連付けられているデプロイメント記述子 (DD) でセキュリティが指定されている URL および EJB リソースに対してのみセキュリティ チェックを実行します。これはデフォルト設定です。

fullyDelegateAuthorization フラグの値が true の場合、WebLogic Security サービスは、すべての URL (Web) および EJB リソースに対して、そのデプロイメント記述子 (DD) にセキュリティ設定があるかどうかに関係なく、セキュリティ チェックを実行します。fullyDelegateAuthorization フラグの値を true に変更した場合は、Web アプリケーションまたは EJB モジュールが再デプロイされるときに WebLogic Security サービスが実行する処理を指定する必要があります。詳細については、2-15 ページの「[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスについて」を参照してください。

注意： fullyDelegateAuthorization フラグは、それが設定されている WebLogic Server インスタンス (サーバ) に対してのみ有効です。したがって、管理サーバと管理対象サーバの両方で fullyDelegateAuthorization フラグが同じように設定されていることを確認してください。

fullyDelegateAuthorization フラグの変更方法

fullyDelegateAuthorization フラグは以下の 3 つの方法のいずれかで設定できます。

- 次のように入力する。

  -Dweblogic.security.fullyDelegateAuthorization = *boolean_value*

  boolean_value には、WebLogic Server インスタンスを起動するたびにコマンドラインで true または false を指定します。

- startWLS スクリプト (*WL_HOME*\server\bin ディレクトリに格納) を編集して以下を追加する。

  -Dweblogic.security.fullyDelegateAuthorization = *boolean_value*

  boolean_value は true または false です。WebLogic Server インスタンスを起動するたびにフラグを設定する必要がなくなるため、こちらの方が効率的です。ただし、この設定はインストールされているすべての WebLogic Server ドメインに適用されます。

コード リスト 2-1 に、このフラグ (太字) を true に設定した startWLS ファイルの該当箇所を示します。

コード リスト 2-1 startWLS スクリプトのサンプル

```
@rem Start Server

@echo off
if "%ADMIN_URL%" == "" goto runAdmin
@echo on
"%JAVA_HOME%\bin\java" %JAVA_VM% %MEM_ARGS% %JAVA_OPTIONS%
-Dweblogic.Name=%SERVER_NAME% -Dbea.home="d:\bea"
-Dweblogic.management.username=%WLS_USER%
-Dweblogic.management.password=%WLS_PW%
-Dweblogic.management.server=%ADMIN_URL%
-Dweblogic.ProductionModeEnabled=%STARTMODE%
-Djava.security.policy="%WL_HOME%\server\lib\weblogic.policy"
-Dweblogic.security.fullyDelegateAuthorization=true
weblogic.Server
goto finish

:runAdmin
@echo on
"%JAVA_HOME%\bin\java" %JAVA_VM% %MEM_ARGS% %JAVA_OPTIONS%
-Dweblogic.Name=%SERVER_NAME% -Dbea.home="d:\bea"
-Dweblogic.management.username=%WLS_USER%
-Dweblogic.management.password=%WLS_PW%
-Dweblogic.ProductionModeEnabled=%STARTMODE%
-Djava.security.policy="%WL_HOME%\server\lib\weblogic.policy"
-Dweblogic.security.fullyDelegateAuthorization=true
weblogic.Server

:finish

ENDLOCAL
```

注意 : 太字の行は startWLS ファイルの main セクションと runAdmin セクションの両方に追加します。ただし、他の変更が行われたためスクリプトに両方のセクションが含まれていない場合は除きます。

- *WL_HOME*\user_projects*domain* ディレクトリに格納されている startWebLogic スクリプト (*domain* は作成した WebLogic Server ドメインの名前) を編集して、JAVA_OPTIONS セクションに次の行を含める。

  ```
  set JAVA_OPTIONS=...
  -Dweblogic.security.fullyDelegateAuthorization=true
  ```

この方法を使用すると、fullyDelegateAuthorization フラグを、インストールされているすべての WebLogic Server ドメインではなくドメインごとに設定できます。

コード リスト 2-2 に、このフラグ (太字) を true に設定した startWebLogic ファイルの該当箇所を示します。

コード リスト 2-2 startWebLogic スクリプトのサンプル

```
@rem Set JAVA_OPTIONS to the java flags you want to pass to the vm. i.e.:
@rem set JAVA_OPTIONS=-Dweblogic.attribute=value -Djava.attribute=value

set JAVA_OPTIONS=-Dweblogic.security.SSL.trustedCAKeyStore=C:\bea_sp02_7a\
     weblogic700\server\lib\cacerts
    -Dweblogic.security.fullyDelegateAuthorization=true
```

ノード マネージャの使用

ノード マネージャを使用して管理対象サーバを起動する場合、上記の起動スクリプトは使用されません。したがって、WebLogic Server Administration Console を使用して fullyDelegatedAuthorization フラグを設定する必要があります。

フラグを設定するには、サーバに対応する [コンフィグレーション | リモート スタート] タブをクリックし、[引数] フィールドに次のように指定します。

```
-Dweblogic.security.fullyDelegatedAuthorization=true
```

図 2-1 に、このフラグを true に設定した examplesServer ファイルの Administration Console の該当フィールドを示します。

図 2-1 examplesServer の [コンフィグレーション | リモート スタート] タブ

[適用] ボタンをクリックし、サーバを再起動して変更を保存してください。

[ロールとポリシーのチェック対象] 設定

WebLogic Server 7.0 SP3 より前は、WebLogic Security サービスがセキュリティ チェックを実行する方法を指定するには、2-11 ページの「fullyDelegateAuthorization フラグの変更方法」で説明されているように、`fullyDelegateAuthorzation` コマンドライン引数を使用する必要がありました。WebLogic Server 7.0 SP3 では、[ロールとポリシーのチェック対象] 設定が導入され、WebLogic Server Administration Console で同じ設定を指定できるようになりました。[ロールとポリシーのチェック対象] ドロップダウン メニューは、[セキュリティ | レルム] に続いて [*myrealm*] (または作成したセキュリティ レルムの名前) をクリックした後に表示される [一般] タブにあります。

注意 : [ロールとポリシーのチェック対象] 設定は、それが設定されている WebLogic Server ドメイン内のすべての WebLogic Server インスタンス (サーバ) に影響します。

[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスについて

`fullyDelegateAuthorization` フラグを `true` に設定して、WebLogic Security サービスによるセキュリティ チェックをすべての Web アプリケーションおよび EJB に対して実行する場合は、URL (Web) および EJB リソースを保護する方法も指定する必要があります (詳細については 2-7 ページの「URL リソースおよび EJB リソースを保護する方法」を参照)。方法を指定するには、[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスを使用します。

注意 : WebLogic Server 7.0 SP3 では、この設定は、WebLogic Server Administration Console の [デプロイメント記述子のセキュリティ動作] ドロップダウン メニューで行われるようになりました。[デプロイメント記述子からセキュリティを取得] 値は、[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスのチェックをはずすことと等しく、[デプロイメント記述子内のセキュリティ データを無視] 値は、[デプロイメント記述子内のセキュリティ データを無視] チェックボックスをチェックすることと同じです。

[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスは、以下のように設定する必要があります。

- WebLogic Server Administration Console のみを使用して URL と EJB リソースを保護するには、1-4 ページの「WebLogic リソースの保護 : 主な手順」の手順 3a から 3c に従って、[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスをチェックする。
- デプロイメント記述子 (`ejb-jar.xml`、`weblogic-ejb-jar.xml`、`web.xml`、および `weblogic.xml` ファイル) のみを使用して URL および EJB リソースを保護するには、[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスのチェックをはずす。『WebLogic Security プログラマーズ ガイド』の「Web アプリケーションでの宣言によるセキュリティの使用」および「EJB での宣言によるセキュリティの使用」を参照してください。

警告 : [デプロイメント記述子内のセキュリティ データを無視] チェック ボックスの値の切り替えは危険であり、セキュリティ コンフィグレーションが不適切になったり失われたりする場合があります。この設定を切り替える必要がある場合は (特に 2-9 ページの「2 つの方法を組み合わせる」

で説明する状態の場合)、2-18 ページの「組み合わせた方法による URL および EJB リソースの保護」の手順に注意深く従ってください。

[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスの設定変更

[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスの値を変更するには、次の手順に従います。

1. WebLogic Server Administration Console の左側のナビゲーション ツリーを使用して、[セキュリティ | レルム] を展開します。
2. このオプションを設定するセキュリティ レルムの名前 (たとえば `myrealm`) をクリックします。
3. [一般] タブで、[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスをクリックして、チェックするかチェックをはずします。
4. [適用] をクリックして変更を保存します。

2 つの設定の相互作用について

表 2-1 に、`fullyDelegateAuthorization` と [デプロイメント記述子内のセキュリティ データを無視] の設定値を組み合わせて WebLogic Security サービスの動作を指定する方法を示します。

表 2-1 2 つの設定の相互作用

セキュリティ チェックの実行対象	URL (Web) および EJB リソースのセキュリティの設定	fullyDelegateAuthorization の設定	[デプロイメント記述子内のセキュリティ データを無視] の設定
すべての URL (Web) および EJB リソース	Administration Console のみを使用する	true	チェックする

表 2-1 2 つの設定の相互作用 (続き)

セキュリティチェックの実行対象	URL (Web) および EJB リソースのセキュリティの設定	fullyDelegateAuthorization の設定	[デプロイメント記述子内のセキュリティデータを無視] の設定
すべての URL (Web) および EJB リソース	Web アプリケーションまたは EJB モジュールのデプロイ時に、コンフィグレーション済みの認可プロバイダおよびロール マッピング プロバイダのデータベースにデプロイメント記述子のセキュリティ データをコピーするか、または再初期化してから、その他の方法を使用する。 **注意 :** セキュリティ データは、Web アプリケーションまたは EJB モジュールをデプロイするたびにコピー / 再初期化される。	true	チェックをはずす
デプロイメント記述子で指定されている URI および EJB メソッドのみ (デフォルト コンフィグレーション)	デプロイメント記述子のみを使用する	false	チェックをはずす

組み合わせた方法による URL および EJB リソースの保護

2-7 ページの「URL リソースおよび EJB リソースを保護する方法」で説明されているように、WebLogic Server Administration Console による方法と J2EE/WebLogic デプロイメント記述子による方法を組み合わせて使用できます。これには通常 2 つの理由があります。

- Web アプリケーションと EJB モジュールの初期デプロイ時に、セキュリティ コンフィグレーションをデプロイメント記述子からコンフィグレーション済みの認可プロバイダとロール マッピング プロバイダのデータベースにコピーするため。このプロセスを行うと、Administration Console を使用してセキュリティ ロールとセキュリティ ポリシーを変更できます。
- Web リソースと EJB リソースのセキュリティ コンフィグレーションを、デプロイメント記述子で指定された元の状態に再初期化するため。

注意 : 組み合わせた方法を他の目的で使用することはお勧めしません。以下の節に進む前に、2-10 ページの「URL リソースおよび EJB リソースを保護するための前提条件」に目を通してください。

以下の節では、組み合わせた方法を使用して URL および EJB リソースを保護する手順について説明します。

- 2-18 ページの「セキュリティ コンフィグレーションのコピー」
- 2-26 ページの「セキュリティ コンフィグレーションの再初期化」

注意 : これらのタスクを実行する前に、第 8 章「例 : basicauth Web アプリケーションのセキュリティ コンフィグレーションのコピーと再初期化」に目を通しておいてください。

セキュリティ コンフィグレーションのコピー

この手順は、現在 J2EE および WebLogic デプロイメント記述子を使用して URL (Web) およびエンタープライズ JavaBean (EJB) リソースを保護しており、今後は WebLogic Server Administration Console だけを使用する予定の管理者を対象としています。セキュリティ コンフィグレーションをデプロイメント記述子と Administration Console の両方で管理することはお勧めしません。

警告 : 組み合わせた方法を使用する場合、URL (Web) および EJB リソースのセキュリティ コンフィグレーションはオーバライドされる可能性があります。したがって、適切なセキュリティ コンフィグレーションが用意されていることを十分に確認する必要があります。データの消失を防ぎ、URL および EJB リソースが適切に保護されるように、以下の手順には慎重に従ってください。

URL または EJB リソースのセキュリティ コンフィグレーションをコピーして、以後 WebLogic Server Administration Console で変更を行うようにするには、次の手順に従います。

- 2-19 ページの「手順 1 : 事前設定を変更してリソースをデプロイする」
- 2-20 ページの「手順 2 : コピーしたセキュリティ ポリシーを検証する (省略可能)」
- 2-22 ページの「手順 3 : コピーしたセキュリティ ロールを検証する (省略可能)」
- 2-25 ページの「手順 4 : [デプロイメント記述子内のセキュリティ データを無視] の設定を元に戻す」
- 2-26 ページの「手順 5 : Administration Console を使用してセキュリティ ロールとセキュリティ ポリシーを変更する (省略可能)」

手順 1 : 事前設定を変更してリソースをデプロイする

1. 2-11 ページの「fullyDelegateAuthorization フラグの変更方法」の指示に従って、`fullyDelegateAuthorization` フラグを `true` に設定します。

   注意 : この設定の意味 : すべての URL (Web) および EJB リソースに対して WebLogic Security サービスによるセキュリティ チェックを実行するよう WebLogic Server に指示します。詳細については、2-10 ページの「fullyDelegateAuthorization フラグについて」を参照してください。

   `fullyDelegateAuthorization` フラグが既に `true` に設定されている場合は、そのまま手順 2 に進みます。

2. サーバを起動し、WebLogic Server Administration Console にサイン インします。詳細については、『管理者ガイド』の「WebLogic Server の起動と停止」を参照してください。

サーバが起動すると、fullyDelegateAuthorization フラグがコンソールに表示されます。

3. Administration Console の左側のナビゲーション ツリーを使用して、[セキュリティ | レルム] を展開します。
4. セキュリティ レルムの名前 (たとえば myrealm) をクリックします。
5. [一般] タブで、[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスのチェックをはずします (チェック ボックスは、デフォルト設定のまま、チェックされていない状態になっている場合があります)。

   注意 : この設定の意味 : リソースをデプロイするたびに、URL (Web) および EJB リソースのセキュリティをデプロイメント記述子からコンフィグレーション済みの認可プロバイダとロール マッピング プロバイダのデータベースにコピーするよう WebLogic Server に指示します。詳細については、2-15 ページの「[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスについて」を参照してください。

6. [適用] をクリックして変更を保存します。
7. 手順 1 で fullyDelegateAuthorization フラグを true に設定する必要があった場合 (つまり、目的の値が設定されていなかった場合)、サーバを再起動します。詳細については、『管理者ガイド』の「WebLogic Server の起動と停止」を参照してください。

   手順 1 で fullyDelegateAuthorization フラグの値を変更していない場合は、**サーバを再起動しないで**手順 8 に進みます。

8. 適切なサーバを対象として、セキュリティ コンフィグレーションをコピーする Web アプリケーションまたは EJB モジュールをデプロイします。

   注意 : Web アプリケーションをデプロイする手順については、『WebLogic Server アプリケーションの開発』の「Administration Console を使用した J2EE アプリケーションのデプロイ」を参照してください。

手順 2 : コピーしたセキュリティ ポリシーを検証する (省略可能)

コピーされたセキュリティ ポリシーを検証するには、表 2-2 で該当するカラムの手順に従ってください。

表 2-2 コピーされたセキュリティ ポリシーのリソース別検証

手順	URL (Web) リソース	EJB リソース
1	Web アプリケーションの web.xml デプロイメント記述子を開き、<url-pattern> および <http-method> 要素の内容と、<auth-constraint> 要素の <role-name> 下位要素の内容を記録する。	EJB の ejb-jar.xml デプロイメント記述子を開き、<method-permission> 要素の内容、特に <role-name>、<ejb-name>、および <method-name> 下位要素の内容を記録する。 **注意 :** このデプロイメント記述子で <unchecked /> 要素が使用されている場合 (通常は <role-name> 要素)、そのメソッドに対するセキュリティ チェックは実行されないので、そのメソッドのセキュリティ データはコピーされない。
2	Administration Console の左側のナビゲーション ツリーを使用して、デプロイ済み Web アプリケーション モジュールの名前を右クリックする。	Administration Console の左側のナビゲーション ツリーを使用して、デプロイ済み EJB モジュールの名前を右クリックする。
3	メニューから [ポリシーを定義] オプションを選択する。	メニューから [個別の Bean のポリシーとロールを定義] オプションを選択する。 JAR ファイル内のすべての EJB を示すテーブルが表示される。
4	[URL パターン] テキスト フィールドに、手順 1 で記録した <url-pattern> 要素の内容に対応する URL パターンを入力する。[ポリシーを定義] ボタンをクリックして次に進む。	手順 1 で記録した <ejb-name> 要素に対応する EJB の [ポリシーを定義] リンクをクリックする。

表 2-2 コピーされたセキュリティ ポリシーのリソース別検証 (続き)

手順	URL (Web) リソース	EJB リソース
5	表示されたポリシー エディタ ページの [Methods] ドロップダウン メニューから、手順 1 で記録した <http-method> 要素の内容に対応するメソッドを選択する。 [ポリシー条件] リスト ボックスの [呼び出し側に許可するロールは] 条件が強調表示される。[ポリシー文] リスト ボックスの内容は、手順 1 で記録した該当する <role-name> 要素の内容に対応している。	表示されたポリシー エディタ ページの [Methods] ドロップダウン メニューから、手順 1 で記録した <method-name> 要素の内容に対応するメソッドを選択する。 [ポリシー条件] リスト ボックスの [呼び出し側に許可するロールは] 条件が強調表示される。[ポリシー文] リスト ボックスの内容は、手順 1 で記録した該当する <role-name> 要素の内容に対応している。
6	複数のセキュリティ ポリシーを検証するには、手順 1 から 5 を繰り返す。	複数のセキュリティ ポリシーを検証するには、手順 1 から 5 を繰り返す。

手順 3 : コピーしたセキュリティ ロールを検証する (省略可能)

コピーされたセキュリティ ロールを検証するには、表 2-3 で該当するカラムの手順に従ってください。

表 2-3 コピーされたセキュリティ ロールのリソース別検証

手順	URL (Web) リソース	EJB リソース
1	Web アプリケーションの weblogic.xml デプロイメント記述子を開き、<security-role-assignment> 要素の内容、特に <role-name> および <principal-name> 下位要素の内容を記録する。 **注意 :** このデプロイメント記述子で Web アプリケーションに対する <global-role/> 要素を使用した場合、スコープ ロールは定義されないので、Web アプリケーションのスコープ ロールはコピーされない。	EJB の weblogic-ejb-jar.xml デプロイメント記述子を開き、<security-role-assignment> 要素の内容、特に <role-name> および <principal-name> 下位要素の内容を記録する。 **注意 :** このデプロイメント記述子で EJB に対する <global-role/> 要素を使用した場合、スコープ ロールは定義されないので、EJB のスコープ ロールはコピーされない。

表 2-3 コピーされたセキュリティ ロールのリソース別検証

手順	URL (Web) リソース	EJB リソース
2	Administration Console の左側のナビゲーション ツリーを使用して、デプロイ済み Web アプリケーション モジュールの名前を右クリックする。	Administration Console の左側のナビゲーション ツリーを使用して、デプロイ済み EJB モジュールの名前を右クリックする。
3	メニューから [ロールを定義] オプションを選択する。 [一般] タブが表示される。	メニューから [ロールを定義] オプションを選択する。 [ロールの選択] ページには、WebLogic ロール マッピング プロバイダのデータベースで現在定義されているこの EJB のすべてのスコープ ロール (デプロイメント記述子の `<role-name>` 要素から取得したものを含む) が表示される。
4	[URL パターン] テキスト フィールドに `/*` と入力してから、[ロールを定義] ボタンをクリックして次に進む。 **注意 :** URL パターン `/*` により、デプロイメント記述子から取得したセキュリティ ロールは常にコンフィグレーション済みのロール マッピング プロバイダのデータベースにスコープ ロールとしてコピーされる。 [ロールの選択] ページには、WebLogic ロール マッピング プロバイダのデータベースで現在定義されているこの Web アプリケーションのすべてのスコープ ロール (デプロイメント記述子の `<role-name>` 要素から取得したものを含む) が表示される。	スコープ ロールの名前のリンクをクリックする。

表 2-3 コピーされたセキュリティ ロールのリソース別検証

手順	URL (Web) リソース	EJB リソース
5	スコープ ロールの名前のリンクをクリックする。	[条件] タブをクリックする。 [ロール文] リスト ボックスに、デプロイメント記述子の対応する <principal-name> 要素の内容に基づいたロール文が表示される。 **注意：** プリンシパルはユーザまたはグループなので、[ロール文] リスト ボックスには、<principal-name> 要素の内容を [呼び出し側のユーザ名は] ロール条件で使用した式と、この要素の内容を [呼び出し側をメンバとするグループは] ロール条件で使用した式の 2 つが or 文を挟んで表示される。Administration Console では、デプロイメント記述子で使用されるユーザまたはグループが存在することを前提にしている。存在していない場合は、それらを作成する必要がある。

表 2-3 コピーされたセキュリティ ロールのリソース別検証

手順	URL (Web) リソース	EJB リソース
6	[条件] タブをクリックする。 [ロール文] リスト ボックスに、デプロイメント記述子の対応する <principal-name> 要素の内容に基づいたロール文が表示される。 **注意 :** プリンシパルはユーザまたはグループなので、[ロール文] リスト ボックスには、<principal-name> 要素の内容を [呼び出し側のユーザ名は] ロール条件で使用した式と、この要素の内容を [呼び出し側をメンバとするグループは] ロール条件で使用した式の 2 つが or 文を挟んで表示される。Administration Console では、デプロイメント記述子で使用されるユーザまたはグループが存在することを前提にしている。存在していない場合は、それらを作成する必要がある。	複数のスコープ ロールを検証するには、手順 1 から 5 を繰り返す。
7	複数のスコープ ロールを検証するには、手順 1 から 6 を繰り返す。	--

手順 4 : [デプロイメント記述子内のセキュリティ データを無視] の設定を元に戻す

警告 : この手順は必須です。この設定を元に戻さないと、Web アプリケーションおよび EJB モジュールを再デプロイした場合に、セキュリティ コンフィグレーションの整合性が失われる可能性があります。このため、サーバを再起動する前に必ずこの手順を実行してください。

1. Administration Console の左側のナビゲーション ツリーを使用して、[セキュリティ | レルム] を展開します。
2. セキュリティ レルムの名前 (たとえば myrealm) をクリックします。

3. [一般] タブで、[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスをクリックします (つまり、ボックスにチェック マークを入れます)。

   注意 : この設定の意味 : Administration Console を使用して、Web アプリケーションおよび EJB リソースのセキュリティを設定するように WebLogic Server に指示します。詳細については、2-15 ページの「[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスについて」を参照してください。

4. [適用] をクリックして変更を保存します。

手順 5 : Administration Console を使用してセキュリティ ロールとセキュリティ ポリシーを変更する (省略可能)

4-18 ページの「グローバル ロールの変更」と 5-21 ページの「セキュリティ ポリシーの変更」に示す手順に従って、URL (Web) リソースのセキュリティ ロールおよびセキュリティ ポリシーを変更します。

セキュリティ コンフィグレーションの再初期化

URL (Web) および EJB リソースのセキュリティ コンフィグレーションをデプロイメント記述子に指定されている元の状態に再初期化するには、次の手順に従います。

- 2-26 ページの「手順 1 : 事前設定を変更して WebLogic リソースを再デプロイする」
- 2-28 ページの「手順 2 : 再初期化したセキュリティ ポリシーとセキュリティ ロールを検証する (省略可能)」
- 2-28 ページの「手順 3 : [デプロイメント記述子内のセキュリティ データを無視] の設定を元に戻す」
- 2-29 ページの「手順 4 : Administration Console を使用してセキュリティ ロールとセキュリティ ポリシーを変更する (省略可能)」

手順 1 : 事前設定を変更して WebLogic リソースを再デプロイする

1. 2-11 ページの「fullyDelegateAuthorization フラグの変更方法」の指示に従って、`fullyDelegateAuthorization` フラグを `true` に設定します。

注意 : この設定の意味 : すべての URL (Web) および EJB リソースに対して WebLogic Security サービスによるセキュリティ チェックを実行するよう WebLogic Server に指示します。詳細については、2-10 ページの「fullyDelegateAuthorization フラグについて」を参照してください。

`fullyDelegateAuthorization` フラグが既に `true` に設定されている場合は、そのまま手順 2 に進みます。

2. サーバを起動し、WebLogic Server Administration Console にサイン インします。詳細については、『管理者ガイド』の「WebLogic Server の起動と停止」を参照してください。

   サーバが起動すると、`fullyDelegateAuthorization` フラグがコンソールに表示されます。

3. Administration Console の左側のナビゲーション ツリーを使用して、[セキュリティ | レルム] を展開します。

4. セキュリティ レルムの名前 (たとえば `myrealm`) をクリックします。

5. [一般] タブで、[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスのチェックをはずします。

   注意 : この設定の意味 : リソースをデプロイするたびに、URL (Web) および EJB リソースのセキュリティをデプロイメント記述子からコンフィグレーション済みの認可プロバイダとロール マッピング プロバイダのデータベースにコピーするよう WebLogic Server に指示します。詳細については、2-15 ページの「[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスについて」を参照してください。

6. [適用] をクリックして変更を保存します。

7. Administration Console の左側のナビゲーション ツリーを使用して、[デプロイメント] を展開してから次のいずれかをクリックします。

   - URL (Web) リソースの場合は [Web アプリケーション]
   - エンタープライズ JavaBean (EJB) リソースの場合は [EJB]

8. Web アプリケーションまたは EJB の名前をクリックします。

   すべての Web アプリケーションまたは EJB を示すテーブルが右ペインに表示されます。

9. セキュリティ コンフィグレーションを再初期化する Web アプリケーションまたは EJB と同じ行にあるごみ箱アイコンをクリックします。
10. [はい] をクリックしてから [続行] リンクをクリックして、Web アプリケーションまたは EJB を削除します。

    削除した Web アプリケーションまたは EJB はテーブルに表示されなくなります。
11. 適切なサーバを対象として、セキュリティ コンフィグレーションを再初期化する Web アプリケーションまたは EJB を再デプロイします。

    注意 : Web アプリケーションおよび EJB をデプロイする手順については、『WebLogic Server アプリケーションの開発』の「デプロイメント ツールおよび手順」を参照してください。

手順 2 : 再初期化したセキュリティ ポリシーとセキュリティ ロールを検証する (省略可能)

再初期化されたセキュリティ ポリシーおよびセキュリティ ロールを検証するには、表 2-2 または 表 2-3 で該当するカラムの手順に従ってください。

手順 3 : [デプロイメント記述子内のセキュリティ データを無視] の設定を元に戻す

警告 : この手順は必須です。この設定を元に戻さないと、Web アプリケーションおよび EJB を再デプロイした場合に、セキュリティ コンフィグレーションの整合性が失われる可能性があります。このため、サーバを再起動する**前に**必ずこの手順を実行してください。

1. Administration Console の左側のナビゲーション ツリーを使用して、[セキュリティ | レルム] を展開します。
2. セキュリティ レルムの名前 (たとえば `myrealm`) をクリックします。
3. [一般] タブで、[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスをクリックします (つまり、ボックスにチェック マークを入れます)。

    注意 : この設定の意味 : Administration Console を使用して、Web アプリケーションおよび EJB リソースのセキュリティを設定するように WebLogic Server に指示します。詳細については、2-15 ページの「[デ

プロイメント記述子内のセキュリティ データを無視] チェック ボックスについて」を参照してください。

4. [適用] をクリックして変更を保存します。

手順 4 : Administration Console を使用してセキュリティ ロールとセキュリティ ポリシーを変更する (省略可能)

4-18 ページの「グローバル ロールの変更」と 5-21 ページの「セキュリティ ポリシーの変更」に示す手順に従って、URL (Web) または EJB リソースのセキュリティ ロールおよびセキュリティ ポリシーを変更します。

Web サービス リソース

一般に、WebLogic Web サービスは、`web-services.xml` というデプロイメント記述子が追加された特別な Web アプリケーションを含むエンタープライズ アプリケーションとしてパッケージ化されます。Web サービスが Java クラスを実装している場合、Web アプリケーションの WAR ファイルには Java クラス ファイルが含まれます。Web サービスがステートレス セッション EJB を実装している場合、エンタープライズ アプリケーションの EAR ファイルには対応する EJB JAR ファイルが含まれます。

注意 : Web サービスは、1 つの Java クラスだけを実装している場合、スタンドアロン Web アプリケーションの WAR ファイルとしてもパッケージ化できます。ただし、Web サービスをこのようにパッケージ化することは一般的でなく、通常は、EAR ファイルとしてパッケージ化します。

Web サービス リソースは、Web サービスに関連する WebLogic リソースです。Web サービスを保護するには、以下を対象としたセキュリティ ポリシーおよびセキュリティ ロールを作成します。

- Web サービス全体
- Web サービスの操作の一部
- Web サービスの URL
- Web サービスを実装するステートレス セッション EJB

- ステートレス セッション EJB のメソッドの一部
- WSDL および Web サービスのホーム ページ

注意： WebLogic Web サービスの保護の詳細については、『WebLogic Web サービス プログラマーズ ガイド』の「セキュリティのコンフィグレーション」を参照してください。

3　ユーザとグループ

ユーザとは、認証が可能なエンティティです。ユーザは、個人または Java クライアントなどのソフトウェア エンティティでもかまいません。各ユーザには、セキュリティ レルムの中で固有の ID が与えられます。セキュリティ管理を効率化するために、ユーザをグループに追加するようにしてください。**グループ**は、通常、企業の同じ部門に所属しているなどの共通点を持つユーザの集合です。

以下の節では、ユーザの詳細について説明します。

以下の節では、グループの詳細について説明します。

ユーザの作成

注意： この節の手順は、WebLogic 認証プロバイダだけに適用されます。デフォルト セキュリティ コンフィグレーションをカスタマイズしてカスタム認証プロバイダを使用している場合、ユーザを作成するにはそのセキュリティ プロバイダの管理ツールを使用する必要があります。

WebLogic 認証プロバイダにアップグレードする場合、既存のユーザを WebLogic 認証プロバイダのデータベースに自動的にロードする方法は存在しません。このリリースの WebLogic Server では、既存のユーザは手動で追加します。既存のユーザが多い場合は、レルム アダプタ認証プロバイダの使用を検討してください。レルム アダプタ認証プロバイダの詳細については、『WebLogic Security の管理』の「レルム アダプタ認証プロバイダのコンフィグレーション」を参照してください。

新しいユーザを作成するには、次の手順に従います。

1. WebLogic Server Administration Console の左ペインで、[セキュリティ | レルム] を展開します。
2. ユーザを作成するセキュリティ レルム (たとえば `myrealm`) を展開します。
3. [ユーザ] をクリックします。

   ユーザが定義されている場合は、定義済みユーザのテーブルが右ペインに表示されます。
4. [新しいユーザのコンフィグレーション] リンクをクリックして、[ユーザの作成] ページを表示します。

   注意： 複数の WebLogic 認証プロバイダがセキュリティ レルムでコンフィグレーションされている場合、新しいユーザの情報をどの WebLogic 認証プロバイダのデータベースに格納するかを選択する必要があります。
5. [一般] タブで、ユーザの名前を [名前] フィールドに入力します。

   注意： スペース、カンマ、ハイフン、\t、< >、#、|、&、~、?、( )、{ }、* を使用しないでください。ユーザ名では大文字 / 小文字を区別します。

6. 必要な場合は、[記述] フィールドにユーザの説明 (フルネームなど) を入力します。
7. [パスワード] フィールドにユーザのパスワードを入力します。

   注意： WebLogic 認証プロバイダで定義されるユーザのパスワードは、最短で 8 文字です。プロダクション環境では、`weblogic/weblogic` というユーザ名とパスワードの組み合わせを使用しないでください。

8. [パスワードの確認] フィールドにユーザのパスワードを再度入力します。
9. [適用] をクリックして変更を保存します。

ユーザのグループへの追加

グループを使用すると多数のユーザを同時に管理できるため、ユーザをグループに追加することをお勧めします。一般的にその方が各ユーザを個別に管理するよりも効率的です。

注意： この節の手順は、3-7 ページの「グループの作成」で説明するようにグループを既に作成してあるか、3-5 ページの「デフォルト グループ」で説明するようにデフォルト グループを使用することを想定しています。

ユーザをグループに追加するには、次の手順に従います。

1. WebLogic Server Administration Console の左ペインで、[セキュリティ | レルム] を展開します。
2. ユーザをグループに追加するセキュリティ レルム (たとえば `myrealm`) を展開します。
3. [ユーザ] をクリックします。

   ユーザが定義されている場合は、定義済みユーザのテーブルが右ペインに表示されます。

4. グループに追加するユーザの名前のリンクをクリックします。

   注意： ユーザ数が多い場合は、[フィルタ条件] フィールドを使用して、検索条件に一致するユーザのみを検索して表示します。[フィルタ条件] フィールドでは、アスタリスク (*) をワイルドカード文字として使用できます。

5. [グループ] タブをクリックします。

   指定可能なすべてのグループが [指定できるグループ] リスト ボックスに表示されます。[現在のグループ] リスト ボックスには、ユーザが所属するすべてのグループが表示されます。

6. [指定できるグループ] リスト ボックスで、グループ名を強調表示します。
7. 強調表示された矢印をクリックして、[指定できるグループ] リスト ボックスから [現在のグループ] リスト ボックスにグループを移動します。
8. 必要な場合は、手順 6 と 7 を繰り返して、複数のグループにユーザを追加します。
9. [適用] をクリックして変更を保存します。

ユーザの変更

既存のユーザを変更するには、次の手順に従います。

1. WebLogic Server Administration Console の左ペインで、[セキュリティ | レルム] を展開します。
2. ユーザを変更するセキュリティ レルム (たとえば `myrealm`) を展開します。
3. [ユーザ] をクリックします。

   定義済みユーザのテーブルが右ペインに表示されます。

4. 変更するユーザの名前のリンクをクリックします。

   注意 : ユーザ数が多い場合は、[フィルタ条件] フィールドを使用して、検索条件に一致するユーザのみを検索して表示します。[フィルタ条件] フィールドでは、アスタリスク (*) をワイルドカード文字として使用できます。

5. ユーザの説明とパスワードを変更するには [一般] タブを、1 つまたは複数のグループでユーザのメンバーシップを変更するには [グループ] タブを使用します (それぞれの手順については、3-2 ページの「ユーザの作成」と 3-3 ページの「ユーザのグループへの追加」参照)。

注意： いずれのタブでも、必ず [適用] ボタンをクリックして変更を保存してください。

ユーザの削除

既存のユーザを削除するには、次の手順に従います。

1. WebLogic Server Administration Console の左ペインで、[セキュリティ | レルム] を展開します。
2. ユーザを削除するセキュリティ レルム (たとえば `myrealm`) を展開します。
3. [ユーザ] をクリックします。

   定義済みユーザのテーブルが右ペインに表示されます。
4. 削除するユーザと同じ行にあるごみ箱アイコンをクリックします。

   注意： ユーザ数が多い場合は、[フィルタ条件] フィールドを使用して、検索条件に一致するユーザのみを検索して表示します。[フィルタ条件] フィールドでは、アスタリスク (*) をワイルドカード文字として使用できます。
5. [はい] をクリックして削除を確認します。
6. [続行] をクリックします。

   [ユーザを選択] ページのテーブルには該当するユーザが表示されなくなります。

デフォルト グループ

WebLogic Server では、表 3-1 に示すグループがデフォルトで定義されています。

表 3-1 デフォルト グループ

グループ名	メンバーシップ
users	(たとえば Web ページからの) ログイン時にユーザが自身の身元を示す場合、ユーザはこのグループのメンバーとなる。 **注意：** users グループには <anonymous> ユーザを除くすべてのユーザが含まれる。<anonymous> ユーザの詳細については、『WebLogic Server 7.0 へのアップグレード』の「ゲスト ユーザ」を参照してください。
everyone	ログイン時にユーザが自身の身元を示すかどうかに関係なく、ユーザはこのグループのメンバーとなる。 **注意：** everyone グループには users グループが含まれる (つまり、ネストされる)。
Administrators	デフォルトでは、グループにはインストール プロセスの一部として入力されたユーザ情報、および system ユーザ (WebLogic Server インスタンスが互換性セキュリティを実行している場合) が含まれる。Administrators グループに割り当てられているユーザは、デフォルトで Admin セキュリティ ロールが付与されている。
Deployers	デフォルトでは、このグループは空。Deployers グループに割り当てられているユーザは、デフォルトで Deployer セキュリティ ロールが付与されている。
Operators	デフォルトでは、このグループは空。Operators グループに割り当てられているユーザは、デフォルトで Operator セキュリティ ロールが付与されている。
Monitors	デフォルトでは、このグループは空。Monitors グループに割り当てられているユーザは、デフォルトで Monitor セキュリティ ロールが付与されている。

注意： デフォルト セキュリティ ロールの詳細については、4-5 ページの「デフォルト グローバル ロール」を参照してください。

3-7 ページの「グループの作成」で説明するとおり、独自のグループを作成してデフォルト グループに追加することもできます。

グループの作成

注意： この節の手順は、WebLogic 認証プロバイダだけに適用されます。デフォルト セキュリティ コンフィグレーションをカスタマイズしてカスタム認証プロバイダを使用している場合、グループを作成するにはそのセキュリティ プロバイダの管理ツールを使用する必要があります。

WebLogic 認証プロバイダにアップグレードする場合、既存のグループを WebLogic 認証プロバイダのデータベースに自動的にロードする方法は存在しません。このリリースの WebLogic Server では、既存のグループは手動で追加します。既存のグループが多い場合は、レルム アダプタ認証プロバイダの使用を検討してください。レルム アダプタ認証プロバイダの詳細については、『WebLogic Security の管理』の「レルム アダプタ認証プロバイダのコンフィグレーション」を参照してください。

新しいグループを作成するには、次の手順に従います。

1. WebLogic Server Administration Console の左ペインで、[セキュリティ | レルム] を展開します。
2. グループを作成するセキュリティ レルム (たとえば `myrealm`) を展開します。
3. [グループ] をクリックします。

   グループが定義されている場合は、定義済みグループのテーブルが右ペインに表示されます。
4. [新しいグループのコンフィグレーション] リンクをクリックして、[グループの作成] ページを表示します。

   注意： 複数の WebLogic 認証プロバイダがセキュリティ レルムでコンフィグレーションされている場合、新しいグループの情報をどの WebLogic 認証プロバイダのデータベースに格納するかを選択する必要があります。
5. [一般] タブで、グループの名前を [名前] フィールドに入力します。

注意： スペース、カンマ、ハイフン、\t、<>、#、|、&、~、?、( )、{ }、* を使用しないでください。グループ名では大文字 / 小文字を区別します。BEA の命名規約では、グループ名は複数形です。

6. 必要な場合は、[記述] フィールドにグループの説明を入力します。
7. [適用] をクリックして変更を保存します。

グループのネスト

必要な場合は、グループを他のグループにネストすることができます。

注意： この節の手順は、3-7 ページの「グループの作成」で説明したようにグループを既に作成してあるか、3-5 ページの「デフォルト グループ」で説明したようにデフォルト グループを使用することを想定しています。

グループを別のグループにネストするには、次の手順に従います。

1. WebLogic Server Administration Console の左ペインで、[セキュリティ | レルム] を展開します。
2. グループをネストするセキュリティ レルム (たとえば `myrealm`) を展開します。
3. [グループ] をクリックします。

   定義済みグループのテーブルが右ペインに表示されます。
4. 別のグループにネストするグループの名前のリンクをクリックします。

   注意： グループ数が多い場合は、[フィルタ条件] フィールドを使用して、検索条件に一致するグループのみを検索して表示します。[フィルタ条件] フィールドでは、アスタリスク (*) をワイルドカード文字として使用できます。
5. [メンバシップ] タブをクリックします。

   指定可能なすべてのグループが [指定できるグループ] リスト ボックスに表示されます。[現在のグループ] リスト ボックスには、そのグループがネストされているすべてのグループが表示されます。
6. [指定できるグループ] リスト ボックスで、グループ名を強調表示します。

7. 強調表示された矢印をクリックして、[指定できるグループ] リスト ボックスから [現在のグループ] リスト ボックスにグループを移動します。
8. 必要な場合は、手順 6 と 7 を繰り返して、複数のグループにグループをネストします。
9. [適用] をクリックして変更を保存します。

グループの変更

既存のグループを変更するには、次の手順に従います。

1. WebLogic Server Administration Console の左ペインで、[セキュリティ | レルム] を展開します。
2. グループを変更するセキュリティ レルム (たとえば `myrealm`) を展開します。
3. [グループ] をクリックします。

   定義済みグループのテーブルが右ペインに表示されます。
4. 変更するグループの名前のリンクをクリックします。

   注意： グループ数が多い場合は、[フィルタ条件] フィールドを使用して、検索条件に一致するグループのみを検索して表示します。[フィルタ条件] フィールドでは、アスタリスク (*) をワイルドカード文字として使用できます。
5. グループの説明を変更するには [一般] タブを、1 つまたは複数の他のグループでグループのメンバーシップを変更するには [メンバシップ] タブを使用します (それぞれの手順については、3-7 ページの「グループの作成」と 3-8 ページの「グループのネスト」を参照)。

   注意： いずれのタブでも、必ず [適用] ボタンをクリックして変更を保存してください。

グループの削除

既存のグループを削除するには、次の手順に従います。

1. WebLogic Server Administration Console の左ペインで、[セキュリティ | レルム] を展開します。
2. グループを削除するセキュリティ レルムの名前 (`myrealm` など) を展開します。
3. [グループ] をクリックします。

   定義済みグループのテーブルが右ペインに表示されます。
4. 削除するグループと同じ行にあるごみ箱アイコンをクリックします。

   注意 : グループ数が多い場合は、[フィルタ条件] フィールドを使用して、検索条件に一致するグループのみを検索して表示します。[フィルタ条件] フィールドでは、アスタリスク (*) をワイルドカード文字として使用できます。
5. [はい] をクリックして削除を確認します。
6. [続行] リンクをクリックします。

   [グループを選択] ページのテーブルには該当するグループが表示されなくなります。

4　セキュリティ ロール

セキュリティ ロールは、特定の条件に基づいてユーザまたはグループに付与される特権です。グループと同様、セキュリティ ロールを使用すると、複数のユーザによる WebLogic リソースへのアクセスを一度に制限できます。ただし、セキュリティ ロールには以下のような特長があります。

- ユーザ名、グループ メンバーシップ、または時刻などの条件に基づいて動的に計算されてユーザまたはグループに付与される。
- (常に WebLogic Server ドメイン全体を対象とするグループとは異なり) WebLogic Server ドメインの単一のアプリケーションに属する特定の WebLogic リソースを対象にできる。

セキュリティ ロールをユーザまたはグループに付与すると、そのセキュリティ ロールを付与されている限り、そのユーザまたはグループには定義されたアクセス特権が与えられます。たとえば、管理者が `AppAdmin` というセキュリティ ロールを定義するとします。このロールは、ある Web アプリケーションのリソースに対する書き込みアクセス権を持っています。この場合、`AppAdmin` セキュリティ ロールを付与されたすべてのユーザまたはグループは、そのリソースに対して書き込みアクセス権を持つことになります。複数のユーザまたはグループに単一のセキュリティ ロールを付与することができます (ユーザとグループの詳細については、第 3 章「ユーザとグループ」を参照)。

注意： WebLogic Server 6.x では、セキュリティ ロールは Web アプリケーションと エンタープライズ JavaBean (EJB) だけに適用されました。このバージョンの WebLogic Server では、セキュリティ ロールは、定義されているすべての WebLogic リソースに対して適用されます。詳細については、第 2 章「WebLogic リソースのタイプ」を参照してください。

以下の節では、セキュリティ ロールについて詳しく説明します。

- 4-12 ページの「デフォルト グループの関連付け」
- 4-13 ページの「セキュリティ ロールの構成要素：ロール条件、式、およびロール文」
- 4-15 ページの「グローバル ロールの操作」
- 4-19 ページの「スコープ ロールの操作」

動的ロール マッピング

実行時に、WebLogic Security サービスはロール条件とユーザまたはグループを比較して、そのユーザまたはグループにセキュリティ ロールを動的に付与するかどうかを決定します。このプロセスを**ロール マッピング**と言います。ロール マッピングは、WebLogic Security サービスが保護対象の WebLogic リソースに対するアクセス決定を下す直前に発生します。

注意： ロール条件とアクセス決定の詳細については、4-13 ページの「セキュリティ ロールの構成要素：ロール条件、式、およびロール文」および『WebLogic Security サービスの開発』の「アクセス決定」を参照してください。

セキュリティ ロールの動的マッピングには、ビジネス ルールまたはリクエストのコンテキストに基づいてユーザまたはグループにセキュリティ ロールを付与できるという、非常に重要な利点があります。たとえば、本来の管理者が不在の間だけユーザに `Manager` セキュリティ ロールを割り当てるといったことができます。このセキュリティ ロールを動的に付与することで、そうした一時的な措置のためにアプリケーションを変更したり再デプロイしたりする必要はなくなります。一時的に管理者となるユーザに特権を割り当てる期間を指定するだけでかまいません。さらに、本当の管理者が戻ってきたときに、特別に付与した一時的な特権を忘れずに取り消す必要もありません。なお、ユーザを一時的に管理者グループに追加した場合には、その必要があります。

セキュリティ ロールのタイプ：グローバル ロールとスコープ ロール

WebLogic Server には、グローバル ロールとスコープ ロールの 2 種類のセキュリティ ロールがあります。セキュリティ レルム内にデプロイされるすべての WebLogic リソース (つまり WebLogic Server ドメイン全体) に適用されるセキュリティ ロールは、**グローバル ロール**と呼ばれます。セキュリティ レルム内にデプロイされる WebLogic リソースの特定のインスタンス (EJB のメソッドや JNDI ツリーのブランチなど) に適用されるセキュリティ ロールは、**スコープ ロール**と呼ばれます。WebLogic リソースのセキュリティ ポリシーを作成するために複数のロール (グローバルまたはスコープ) を使用できます。詳細については、第 5 章「セキュリティ ポリシー」を参照してください。

WebLogic リソースを保護するために (ユーザまたはグループよりも) セキュリティ ロールを作成して使用することを強くお勧めしますが、必ずしも特定のタイプのセキュリティ ロールを使用する必要はありません。WebLogic リソースを保護するためにそのまま使用できる複数のデフォルト グローバル ロールが用意されています (4-5 ページの「デフォルト グローバル ロール」を参照)。これらを必要としない場合、スコープ ロールを使用する必要ありません。スコープ ロールは柔軟性を高めるためのもので、高度なユーザ向けの特別な機能です。

Administration Console でのセキュリティ ロールの作成方法

セキュリティ ロールを作成するための WebLogic Server Administration Console の使い方は、グローバル ロールとセキュリティ ロールのどちらを作成するかによって異なります。

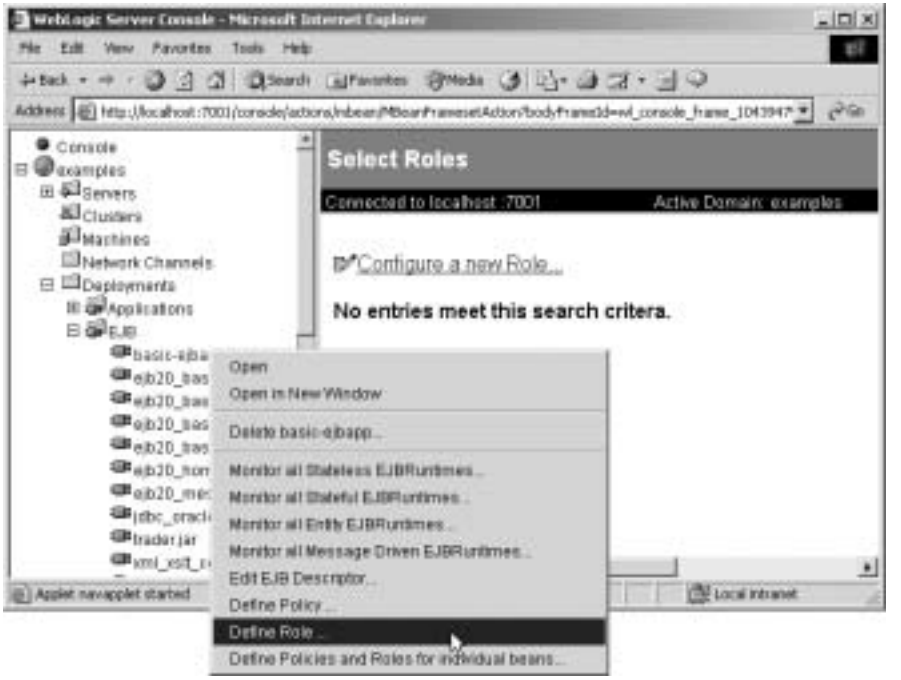
グローバル ロールはセキュリティ レルム内のすべての WebLogic リソースに適用されるため、セキュリティ レルム レベルで作成します。Administration Console を使用して、[セキュリティ | レルム] に続いて `myrealm` (または作成し

たセキュリティ レルムの名前) を展開します。次に、[ロール] をクリックしてグローバル ロールの作成ページを表示します。このナビゲーション パスを図 4-1 の左側に、表示されるページを右側に示します。

図 4-1　グローバル ロールの作成

スコープ ロールはセキュリティ レルム内の特定の WebLogic リソースだけに適用されるため、WebLogic リソース レベルで作成します。スコープ ロールを作成可能なデプロイ済みコンポーネント (Web アプリケーションや EJB など) の場合、Administration Console のナビゲーション ツリーでそのコンポーネントを右クリックすると、[ロールを定義] オプションが表示されます。次に、[ロールを定義 ...] をクリックしてスコープ ロールの作成ページを表示します。このナビゲーション パス (`basic-ejbapp` を WebLogic リソースとして使用) を図 4-2 の左側に、表示されるページを右側に示します。

図 4-2　スコープ ロールの作成



デフォルト グローバル ロール

WebLogic Server では、表 4-1 に示すグローバル ロールがデフォルトで定義されています。この表では、これらのセキュリティ ロールのユーザまたはグループに付与される特権についても説明します。

注意： デフォルト グローバル ロールは、ほとんどのタイプの WebLogic リソースを保護するデフォルト セキュリティ ポリシーで使用されます。また、デフォルト グローバル ロールを使用すると、MBean として公開されるサーバ リソースのセキュリティを強化できます。詳細については、第 5 章「セキュリティ ポリシー」および『管理者ガイド』の「システム管理操作の保護」を参照してください。

表 4-1 デフォルト グローバル ロールと特権

グローバル ロール	特権
Anonymous	すべてのユーザ (everyone グループ) にこのグローバル ロールが付与される。 **注意：** このグローバル ロールは利便性のために用意されており、weblogic.xml および weblogic-ejb-jar.xml デプロイメント記述子で指定できる。
Admin	■ サーバ コンフィグレーション (暗号化された属性の暗号化された値を含む) を表示する。 ■ サーバ コンフィグレーション全体を変更する。 ■ エンタープライズ アプリケーションをデプロイし、クラス、Web アプリケーション、EJB、J2EE コネクタ、および Web サービス コンポーネントを起動 / 停止する。必要に応じて、デプロイメント記述子を編集する。 ■ デフォルトでは、サーバを起動、再開、および停止する。
Deployer	■ サーバ コンフィグレーション (暗号化された属性は除く) を表示する。 ■ エンタープライズ アプリケーションをデプロイし、クラス、Web アプリケーション、EJB、J2EE コネクタ、および Web サービス コンポーネントを起動 / 停止する。必要に応じて、デプロイメント記述子を編集する。
Operator	■ サーバ コンフィグレーション (暗号化された属性は除く) を表示する。 ■ デフォルトでは、サーバを起動、再開、および停止する。
Monitor	サーバ コンフィグレーション (暗号化された属性は除く) を表示する。 **注意：** このセキュリティ ロールは実際には、Administration Console、weblogic.Admin ユーティリティ、および MBean API に対する読み取り専用アクセスを提供する。

注意： WebLogic Server MBean を直接操作するために、グローバル ロールおよび特権について表 4-1 よりも詳しい情報が必要な場合は、4-7 ページの「保護されている MBean の属性および操作」を参照してください。

4-15 ページの「グローバル ロールの作成」および 4-19 ページの「スコープ ロールの作成」で説明するように、独自のセキュリティ ロール (グローバルまたはスコープ) を作成して、デフォルト グローバル ロールに追加することができます。

保護されている MBean の属性および操作

表 4-2 に、Admin デフォルト グローバル ロールを付与されたユーザまたはグループがさまざまな WebLogic Server MBean に関して与えられる一定の特権を示します。つまり、Admin デフォルト グローバル ロールを付与されたユーザまたはグループは、表 4-2 に示した MBean 属性にアクセスするパーミッションを持ちます。

注意： Admin デフォルト グローバル ロールを付与されたユーザまたはグループには、表 4-3 と 表 4-5 に示された特権も与えられます。

表 4-2 Admin デフォルト グローバル ロールの MBean 特権

MBean	属性
BridgeDestinationCommonMBean	UserPassword
BridgeDestinationMBean	UserPassword
JDBCConnectionPoolMBean	Password、XAPassword
JDBCDataSourceFactoryMBean	Password
JMSBridgeDestinationMBean	UserPassword
NetworkChannelMBean	DefaultIIOPPassword
NodeManagerMBean	CertificatePassword
SecurityConfigurationMBean	Credential、EncryptedSecretKey、Salt
SecurityMBean	Salt、EncryptedSecretKey

表 4-2 Admin デフォルト グローバル ロールの MBean 特権 (続き)

MBean	属性
ServerMBean	SystemPassword、DefaultIIOPPassword、DefaultTGIOPPassword、CustomIdentityKeyStorePassPhrase、CustomTrustKeyStorePassPhrase、JavaStandardTrustKeyStorePassPhrase
ServerStartMBean	Password
SSLMBean	ServerPrivateKeyPassPhrase
WLECConnectionPoolMBean	UserPassword、ApplicationPassword

注意 : 表 4-2 に示されている MBean は、すべて `weblogic.management.configuration` パッケージに入っています。WebLogic Server をコンフィグレーションするための MBean の詳細については、『管理者ガイド』の「システム管理のインフラストラクチャ」を参照してください。

表 4-3 に、`Admin` または `Deployer` デフォルト グローバル ロールを付与されたユーザまたはグループがさまざまな WebLogic Server MBean に関して与えられる一定の特権を示します。つまり、`Admin` または `Deployer` デフォルト グローバル ロールを付与されたユーザまたはグループは、表 4-3 に示した MBean 操作にアクセスするパーミッションを持ちます。

表 4-3 Admin または Deployer デフォルト グローバル ロールの特権

MBean	操作
Application、ApplicationConfig	すべて
ConnectorComponent、ConnectorComponentConfig	すべて
DeployerRuntime、DeploymentTaskRuntime	すべて
EJBComponent、EJBComponentConfig	すべて

表 4-3 Admin または Deployer デフォルト グローバル ロールの特権 (続き)

MBean	操作
WebAppComponent、WebAppComponentConfig	すべて
WebServiceComponent、WebServiceComponentConfig	すべて
WebServer、WebServerConfig	すべて
JDBCConnectionPool、JDBCConnectionPoolConfig	すべて
JDBCDataSourceFactory、JDBCDataSourceFactoryConfig	すべて
JDBCMultiPool、JDBCMultipoolConfig	すべて
JDBCDataSource、JDBCDataSourceConfig	すべて
JDBCTxDataSource、JDBCTxDataSourceConfig	すべて
JDBCPoolComponent、JDBCPoolComponentConfig	すべて
JMSBridgeDestination、JMSBridgeDestinationConfig	すべて
JMSConnectionConsumer、JMSConnectionConsumerConfig	すべて
JMSConnectionFactory、JMSConnectionFactoryConfig	すべて
JMSDestination、JMSDestinationConfig	すべて
JMSDistributedDestination、JMSDistributedDestinationConfig	すべて
JMSDistributedDestinationMember、JMSDistributedDestinationMemberConfig	すべて

表 4-3 Admin または Deployer デフォルト グローバル ロールの特権 (続き)

MBean	操作
JMSDistributedTopic、JMSDistributedTopicConfig	すべて
JMSDistributedTopicMember、JMSDistributedTopicMemberConfig	すべて
JMSDistributedQueue、JMSDistributedQueueConfig	すべて
JMSDistributedQueueMember、JMSDistributedQueueMemberConfig	すべて
JMSFileStore、JMSFileStoreConfig	すべて
JMSDestinationKey、JMSDestinationKeyConfig	すべて
JMSServer、JMSServerConfig	すべて
JMSStore、JMSStoreConfig	すべて
JMSSessionPool、JMSSessionPoolConfig	すべて
JMSTemplate、JMSTemplateConfig	すべて
JMSQueue、JMSQueueConfig	すべて
JMSTopic、JMSTopicConfig	すべて
JMSJDBCStore、JMSJDBCStoreConfig	すべて
WTCServer、WTCServerConfig	すべて
WTCBridgeGlobal、WTCBridgeGlobalConfig	すべて
WTCResources、WTCResourcesConfig	すべて
WTCExport、WTCExportConfig	すべて
WTCImport、WTCImportConfig	すべて

表 4-3 Admin または Deployer デフォルト グローバル ロールの特権 (続き)

MBean	操作
WTCLocalTuxDom、WTCLocalTuxDomConfig	すべて
WTCRemoteTuxDom、WTCRemoteTuxDomConfig	すべて
WTCPassword、WTCPasswordConfig	すべて
WTCtBridgeGlobal、WTCtBridgeGlobalConfig	すべて
WTCtBridgeRedirect、WTCtBridgeRedirectConfig	すべて
EJBDescriptor、ConnectorDescriptor、WebDescriptor	すべて
Server	addDeployment、lookupServerLifeCycleRuntime、lookupServerRuntime、removeDeployment、sendNotification
ServerConfig	addDeployment、lookupServerLifeCycleRuntime、removeDeployment、sendNotification

表 4-4 に、`Admin` または `Monitor` デフォルト グローバル ロールを付与されたユーザまたはグループがさまざまな WebLogic Server MBean に関して与えられる一定の特権を示します。つまり、`Admin` または `Monitor` デフォルト グローバル ロールを付与されたユーザまたはグループは、表 4-4 に示した MBean 操作にアクセスするパーミッションを持ちます。

表 4-4 Admin または Monitor デフォルト グローバル ロールの特権

MBean	操作
Machine	lookupNodeManagerRuntime
NodeManagerRuntime	getStateForAll、register

表 4-4 Admin または Monitor デフォルト グローバル ロールの特権 (続き)

MBean	操作
Server	lookupServerLifeCycleRuntime、lookupServerRuntime

表 4-5 に、Admin または Operator デフォルト グローバル ロールを付与されたユーザまたはグループがさまざまな WebLogic Server MBean に関して与えられる一定の特権を示します。つまり、Admin または Operator デフォルト グローバル ロールを付与されたユーザまたはグループは、表 4-5 に示した MBean 操作にアクセスするパーミッションを持ちます。

表 4-5 Admin または Operator デフォルト グローバル ロールの特権

MBean	操作
ServerLifeCycleRuntime	すべて
ServerLifeCycleTaskRuntime	すべて
ServerStart	すべて
Server	ExpectedToRun、lookupServerLifeCycleRuntime、lookupServerRuntime、sendNotification、start、suspend
ServerConfig	ExpectedToRun、lookupServerLifeCycleRuntime、sendNotification
ServerRuntime	forceShutdown、resume、shutdown、start、stop

デフォルト グループの関連付け

デフォルトでは、WebLogic Server は 4 つのデフォルト グループに 4 つのグローバル ロールを付与します。これらのグループのいずれかにユーザを追加すると、そのユーザにはグローバル ロールが自動的に付与されます。このデフォルト グループの関連付けを表 4-6 に示します。

表 4-6 デフォルト グループの関連付け

グループ	関連付けられるグローバル ロール
Administrators	Admin
Deployers	Deployer
Operators	Operator
Monitors	Monitors

セキュリティ ロールの構成要素：ロール条件、式、およびロール文

ロール条件は、セキュリティ ロール (グローバルまたはスコープ) をユーザまたはグループに付与する条件です。このリリースの WebLogic Server で使用できるロール条件は以下のとおりです。

- [呼び出し側のユーザ名は] ― ユーザ名に基づいてセキュリティ ロールの条件を作成します。たとえば、ユーザ John だけが BankTeller セキュリティ ロールを付与されるという条件を作成できます。
- [呼び出し側をメンバとするグループは] ― グループに基づいてセキュリティ ロールの条件を作成します。たとえば、グループ FullTimeBankEmployees に属するユーザだけが BankTeller セキュリティ ロールを付与されるという条件を作成できます。セキュリティ管理がより効率的になるため、このロール条件をお勧めします。
- [アクセス可能な時間帯は] ― 指定した時間に基づいてセキュリティ ロールの条件を作成します。たとえば、銀行の営業時間中にだけ BankTeller セキュリティ ロールをユーザに付与するという条件を作成できます。

  [アクセス可能な時間帯は] ロール条件を使用する場合、他のロール条件を追加してユーザをさらに制限しないかぎり、セキュリティ ロールは指定した時間中にすべてのユーザに付与されます。

これらのロール条件に対して特定の情報 (実際のユーザ名、グループ、開始 / 終了時間など) を指定したものは**式**と呼ばれます。WebLogic Server Administration Console に表示される式の例を図 4-3 に示します。

図 4-3　式の例

この式の例では、1 行目がロール条件、2 行目が条件に対して指定した特定の情報 (この場合は、`FullTimeBankEmployees` というグループ) です。

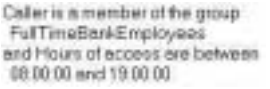
ロール文は、セキュリティ ロールが付与される条件を定義した式の集合です。したがって、作成するセキュリティ ロールの主要部分となります。複数の式を使用できるため、企業のセキュリティ要件に合わせて複雑なセキュリティ ロールを作成できます。式の間の `and` と `or` の使い方、および式の順序も重要な機能です。

- `and` は、セキュリティ ロールが付与されるにはすべての式が true でなければならないことを指定するために使用します。
- `or` は、セキュリティ ロールが付与されるには少なくとも 1 つの式が true でなければならないことを指定するために使用します。

注意 : ユーザまたはグループにセキュリティ ロールが付与されるには、ロール文全体が true でなければなりません。ロール文の中では、制約が厳しい式ほど後に指定します。WebLogic Server では、ロール文中の式は左から右に評価されます。

Administration Console に表示されるロール文の例を図 4-4 に示します。

図 4-4　ロール文の例



このロール文の例には 2 つの式があります。1 行目と 2 行目は [呼び出し側をメンバとするグループは] ロール条件に基づく式、3 行目と 4 行目は [アクセス可能な時間帯は] ロール条件に基づく別の式です。

グローバル ロールの操作

以下の節では、グローバル ロールの操作手順を説明します。

注意： この節では、グローバル ロールを作成、変更、および削除する方法を説明します。スコープ ロールは常に WebLogic リソースを対象としているので、スコープ ロールを作成、変更、および削除する手順は 4-19 ページの「スコープ ロールの操作」で説明されています。

グローバル ロールの作成

注意： セキュリティ ロールを作成する前に 4-3 ページの「Administration Console でのセキュリティ ロールの作成方法」を見直しておくことをお勧めします。サーバ リソースを保護するためのグローバル ロールを作成する場合は、「システム管理操作の保護」の「一貫性のあるセキュリティ方式の維持」で説明されているアドバイスに従ってください。

新しいグローバル ロールを作成するには、次の手順に従います。

1. WebLogic Server Administration Console の左ペインで、[セキュリティ | レルム] を展開します。
2. グローバル ロールを作成するセキュリティ レルム (たとえば `myrealm`) を展開します。
3. [ロール] をクリックして [ロールの選択] ページを表示します。

   グローバル ロールが定義されている場合は、定義済みグローバル ロールのテーブルが右ペインに表示されます。
4. [新しい Role のコンフィグレーション] をクリックします。

   注意： 複数の WebLogic ロール マッピング プロバイダがセキュリティ レルムでコンフィグレーションされている場合、新しいグローバル ロール

の情報をどの WebLogic ロール マッピング プロバイダのデータベースに格納するかを選択する必要があります。

5. [一般] タブで、グローバル ロールの名前を [名前] フィールドに入力します。

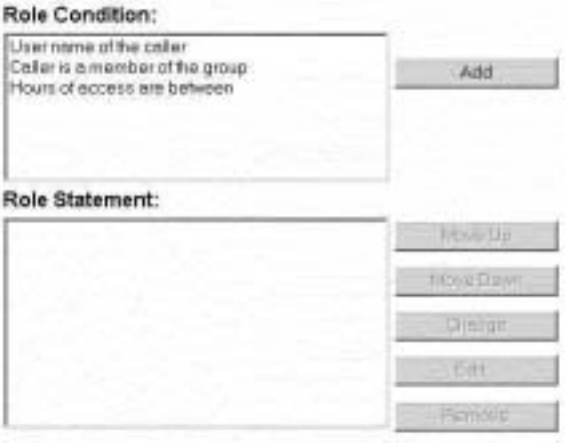
   注意 : スペース、カンマ、ハイフン、\t、<>、#、|、&、~、?、( )、{ } を使用しないでください。セキュリティ ロール名では大文字 / 小文字を区別します。BEA の命名規約では、セキュリティ ロール名は単数形で、先頭の文字は大文字です。

   セキュリティ ロール名の適切な構文は、Extensible Markup Language (XML) 勧告で `Nmtoken` に関して定義されているとおりです。

6. [適用] をクリックして変更を保存します。

7. [条件] タブをクリックして、ロール エディタ ページを表示します (図 4-5 参照)。

図 4-5　ロール エディタ ページ



8. [ロール条件] リスト ボックスで、いずれかの条件をクリックします。さまざまなロール条件の詳細については、4-13 ページの「セキュリティ ロールの構成要素 : ロール条件、式、およびロール文」を参照してください。

   注意 : [呼び出し側をメンバとするグループは] 条件を使用して式を作成することをお勧めします。グループを使用してセキュリティ ロールを作

成すると、セキュリティ ロールはそのグループのすべてのメンバー (つまり複数のユーザ) に付与されます。

9. [追加] をクリックしてカスタマイズ ウィンドウを表示します。
10. [アクセス可能な時間帯は] 条件を選択した場合は、[時間制約] ウィンドウを使用して開始時刻と終了時刻を選択し、[OK] ボタンをクリックします。ウィンドウが閉じて、[ロール文] リスト ボックスに式が表示されます。

    他の条件のいずれかを選択した場合は、次の手順に従います。

    a. [ユーザ] または [グループ] ウィンドウを使用してユーザまたはグループの名前を入力し、[追加] をクリックします。リスト ボックスに式が表示されます。

       注意： 複数のユーザまたはグループを追加するには、この手順を複数回繰り返します。

    b. 必要に応じて、リスト ボックスの右側にあるボタンを使用して式を変更します。

       [上へ移動] および [下へ移動] をクリックすると、強調表示されたユーザ名またはグループ名の順序が変更されます。[変更] をクリックすると、式の間にある強調表示された and 文と or 文が切り替わります。[削除] をクリックすると、強調表示されたユーザ名またはグループ名が削除されます。

    c. [OK] をクリックして、ロール文に式を追加します。ウィンドウが閉じて、[ロール文] リスト ボックスに式が表示されます。

11. 必要な場合は、手順 8 から 10 を繰り返して、別のロール条件に基づいて式を追加します。
12. 必要に応じて、[ロール文] リスト ボックスの右側にあるボタンを使用して式を変更します。
    - [上へ移動] および [下へ移動] をクリックすると、強調表示された式の順序が変更されます。
    - [変更] をクリックすると、式の間にある強調表示された and 文と or 文が切り替わります。
    - [編集] をクリックすると、強調表示された式のカスタマイズ ウィンドウが再び開き、式を変更できます。
    - [削除] をクリックすると、選択した式が削除されます。

13. [ロール文] リスト ボックスのすべての式が正しい場合は、[適用] をクリックします。

    注意： [リセット] をクリックして、ロール エディタ ページを最初にロードしたときの状態に戻す (つまり、変更をすべて元に戻す) こともできます。

グローバル ロールの変更

グローバル ロールを変更する手順は、新しいグローバル ロールの作成手順とほとんど同じです。次の手順に従います。

1. WebLogic Server Administration Console の左ペインで、[セキュリティ | レルム] を展開します。
2. グローバル ロールを変更するセキュリティ レルム (たとえば `myrealm`) を展開します。
3. [ロール] をクリックします。

   定義済みグローバル ロールのテーブルが右ペインに表示されます。
4. 変更するグローバル ロールをテーブルから選択します。
5. [条件] タブを選択して、ロール エディタ ページを表示します。
6. 4-15 ページの「グローバル ロールの作成」の手順 8 から 12 を参考にして、変更を加えます。
7. [適用] をクリックして変更を保存します。

グローバル ロールの削除

グローバル ロールを削除するには、次の手順に従います。

1. WebLogic Server Administration Console の左ペインで、[セキュリティ | レルム] を展開します。
2. グローバル ロールを削除するセキュリティ レルム (たとえば `myrealm`) を展開します。

3. [ロール] をクリックします。

   定義済みグローバル ロールのテーブルが右ペインに表示されます。

4. 削除するグローバル ロールと同じ行にあるごみ箱アイコンをクリックします。

5. [はい] をクリックして削除を確認します。

6. [続行] をクリックします。

   [ロールを選択] ページのテーブルには該当するグローバル ロールが表示されなくなります。

スコープ ロールの操作

以下の節では、さまざまなタイプの WebLogic リソースに対するスコープ ロールの操作手順を説明します。

- 4-19 ページの「スコープ ロールの作成」
- 4-30 ページの「スコープ ロールの変更」
- 4-31 ページの「スコープ ロールの削除」

スコープ ロールの作成

WebLogic リソースのスコープ ロールを作成するには、次の手順に従います。

- 4-20 ページの「手順 1 : WebLogic リソースを選択する」
- 4-28 ページの「手順 2 : スコープ ロールを作成する」
- 4-28 ページの「手順 3 : ロール条件を作成する」

注意 : スコープ ロールの操作手順は、WebLogic リソースごとに若干異なります。この手順で示した WebLogic リソースのタイプごとの違いに注意して、適切な手順に従ってください。詳細については、第 2 章「WebLogic リソースのタイプ」を参照してください。

手順 1 : WebLogic リソースを選択する

該当する節の手順に従って WebLogic リソースのタイプを選択します。

- 4-20 ページの「管理リソース」
- 4-20 ページの「アプリケーション リソース」
- 4-21 ページの「COM リソース」
- 4-22 ページの「EIS リソース」
- 4-22 ページの「EJB リソース」
- 4-23 ページの「JDBC リソース」
- 4-24 ページの「JMS リソース」
- 4-25 ページの「JNDI リソース」
- 4-26 ページの「サーバ リソース」
- 4-27 ページの「URL リソース」

管理リソース

WebLogic Server Administration Console の左ペインで、WebLogic Server ドメインの名前 (たとえば `examples`) を右クリックし、[ロールを定義] を選択して、[ロールの選択] ページを表示します。

スコープ ロールが定義されている場合は、定義済みスコープ ロールのテーブルが右ペインに表示されます。

アプリケーション リソース

1. WebLogic Server Administration Console の左ペインで、[デプロイメント | アプリケーション] を展開します。

   注意 : 必要に応じて、スコープ ロールを作成するエンタープライズ アプリケーション (EAR) を展開して、別のタイプの WebLogic リソースを表示します。

2. エンタープライズ アプリケーション (EAR) の名前を右クリックし、[ロールを定義] を選択して、[ロールの選択] ページを表示します。

スコープ ロールが定義されている場合は、定義済みスコープ ロールのテーブルが右ペインに表示されます。

COM リソース

EJB クラス (`ejb20.basic.beanManaged.*` など) のパッケージに COM クライアントからアクセスする場合は、次の手順に従ってスコープ ロールを作成します。

1. WebLogic Server Administration Console の左ペインで、[デプロイメント | EJB] を展開します。

   [EJB] ノードを展開すると、デプロイ済みの EJB JAR が表示されます。

2. パッケージにアクセスするための EJB を格納している EJB JAR の名前を右クリックし、[個別の Bean のポリシーとロールを定義] を選択して、EJB のリストを表示します。

3. パッケージにアクセスするための EJB と同じ行の [JCOM ロールの定義] リンクをクリックします。

   [一般] タブの [COM クラス] フィールドには、スコープ ロールの対象とするパッケージの名前が表示されます。

   注意 : [COM クラス] フィールドの値は、jCOM ブリッジを介して COM に公開される Java クラスまたはパッケージの名前です。

4. [ロールを定義] ボタンをクリックして、[ロールの選択] ページを表示します。

   スコープ ロールが定義されている場合は、定義済みスコープ ロールのテーブルが右ペインに表示されます。

Java クラス (`java.util.*` など) のパッケージまたは個々のクラス (`java.util.Collection` など) に COM クライアントからアクセスする場合は、次の手順に従ってスコープ ロールを作成します。

1. WebLogic Server Administration Console の左ペインで、[サービス] を展開します。

2. [JCOM] ノードを右クリックして、[ロールを定義] を選択します。

3. [一般] タブの [COM クラス] フィールドに、スコープ ロールの対象とする Java クラスまたはパッケージの名前を入力します。

注意： [COM クラス] フィールドに入力する値は、jCOM ブリッジを介して COM に公開される Java クラスまたはパッケージの名前です。

4. [ロールを定義] ボタンをクリックして、[ロールの選択] ページを表示します。

   スコープ ロールが定義されている場合は、定義済みスコープ ロールのテーブルが右ペインに表示されます。

EIS リソース

1. WebLogic Server Administration Console の左ペインで、[デプロイメント] を展開します。

   [デプロイメント] ノードを展開すると、デプロイ可能な WebLogic リソースのタイプが表示されます。

2. スコープ ロールの対象とする EIS リソースのレベルで右クリックし、[ロールを定義] を選択して、[ロールの選択] ページを表示します。

   すべてのコネクタを対象とするスコープ ロールを作成するには、[コネクタ] を右クリックします。特定のコネクタを対象とするスコープ ロールを作成するには、[コネクタ] を展開してからコネクタの名前を右クリックします。図 4-6 では、例として `basic-connector` を使用して、クリックする位置を示します。

図 4-6　Administration Console ナビゲーション ツリーのデプロイメント部分

スコープ ロールが定義されている場合は、定義済みスコープ ロールのテーブルが右ペインに表示されます。

EJB リソース

注意： ここで説明する手順は、メッセージ駆動型 Bean (MDB) にも当てはまります。

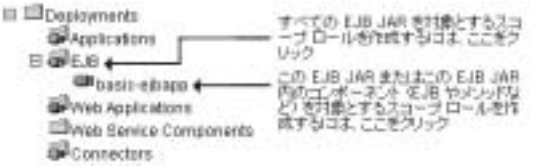
1. WebLogic Server Administration Console の左ペインで、[デプロイメント] を展開します。

   [デプロイメント] ノードを展開すると、デプロイ可能な WebLogic リソースのタイプが表示されます。

2. 作成するスコープ ロールの対象となる EJB リソースのレベルで右クリックします。

   すべての EJB JAR を対象とするスコープ ロールを作成するには、[EJB] を右クリックします。特定の EJB JAR、または JAR 内の EJB を対象とするスコープ ロールを作成するには、[EJB] を展開してから EJB JAR の名前を右クリックします。図 4-7 では、例として `basic-ejbapp` JAR を使用して、クリックする位置を示します。

図 4-7 Administration Console ナビゲーション ツリーのデプロイメント部分



3. すべての EJB JAR または特定の EJB JAR (つまり、JAR 内のすべての EJB) を対象とするスコープ ロールを作成する場合は、[ロールを定義] を選択して [ロールの選択] ページを表示します。

   EJB JAR 内の特定の EJB を対象とするスコープ ロールを作成する場合は、次の手順に従います。

   a. [個別の Bean のポリシーとロールを定義] を選択して、EJB のリストを表示します。

   b. 作成するスコープ ロールの対象となる EJB と同じ行の [ロールを定義] リンクをクリックします。

   スコープ ロールが定義されている場合は、定義済みスコープ ロールのテーブルが右ペインに表示されます。

JDBC リソース

1. WebLogic Server Administration Console の左ペインで、[サービス | JDBC] を展開します。

[JDBC] を展開すると、さまざまな JDBC コンポーネント (接続プール、マルチプール、およびデータ ソース) に対応するノードが表示されます。

2. スコープ ロールの対象とする JDBC リソースのレベルで右クリックし、[ロールを定義] を選択して、[ロールの選択] ページを表示します。

   すべての接続プールを対象とするスコープ ロールを作成するには、[接続プール] を右クリックします。特定の接続プールを対象とするスコープ ロールを作成するには、[接続プール] を展開してから接続プールの名前を右クリックします。個々のマルチプールを対象とするスコープ ロールを作成するには、[マルチプール] を展開してからマルチプールの名前を右クリックします。

   注意 : すべてのマルチプールを対象とするスコープ ロールを作成することはできません。

   図 4-8 では、例として接続プールとマルチプールを使用して、クリックする位置を示します。

図 4-8　Administration Console ナビゲーション ツリーのサービス部分

スコープ ロールが定義されている場合は、定義済みスコープ ロールのテーブルが右ペインに表示されます。

JMS リソース

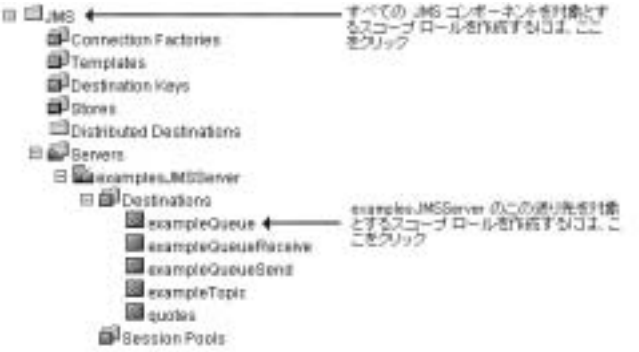
1. WebLogic Server Administration Console の左ペインで、[サービス | JMS] を展開します。

   [JMS] を展開すると、さまざまな JMS コンポーネント (接続ファクトリ、テンプレート、送り先キーなど) に対応するノードが表示されます。

2. スコープ ロールの対象とする JMS リソースのレベルで右クリックし、[ロールを定義] を選択して、[ロールの選択] ページを表示します。

   すべての JMS コンポーネントを対象とするスコープ ロールを作成するには、[JMS] を右クリックします。JMS サーバ上の特定の送り先を対象とするスコープ ロールを作成するには、[サーバ | JMS サーバ | 送り先] ノードを展開してから送り先の名前を右クリックします。図 4-9 では、例として `examplesJMSServer` を使用して、クリックする位置を示します。

図 4-9 Administration Console ナビゲーション ツリーのサービス部分



   スコープ ロールが定義されている場合は、定義済みスコープ ロールのテーブルが右ペインに表示されます。

JNDI リソース

1. WebLogic Server Administration Console の左ペインで、[サーバ] を展開します。

   [サーバ] ノードを展開すると、現在の WebLogic Server ドメインで利用可能なサーバが表示されます。

2. 作成するスコープ ロールの対象となる JNDI リソースを含むサーバの名前 (たとえば `myserver`) を右クリックします。

3. メニューから [JNDI ツリーを見る] オプションを選択します。

   新しい Administration Console ウィンドウに、このサーバの JNDI ツリーが表示されます。

4. Administration Console ウィンドウで、スコープ ロールの対象とする JNDI ツリーのレベルで右クリックし、[ロールを定義] を選択して、[ロールの選択] ページを表示します。

   オブジェクトのグループを対象とするスコープ ロールを作成するには、オブジェクト タイプを表すノードを右クリックします。特定のオブジェクトを対象とするスコープ ロールを作成するには、オブジェクトを表すノードを展開してからオブジェクトの名前を右クリックします。

   図 4-10 では、例として examplesServer JNDI ツリーを使用して、クリックする位置を示します。

図 4-10　examplesServer JNDI ツリーを示す新しい Administration Console ウィンドウ

   スコープ ロールが定義されている場合は、定義済みスコープ ロールのテーブルが右ペインに表示されます。

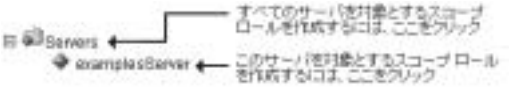
サーバ リソース

WebLogic Server Administration Console の左ペインで、[サーバ] を展開します。

1. [サーバ] ノードを展開すると、スコープ ロールを作成可能な別のサーバ リソースが表示されます。
2. スコープ ロールの対象とするサーバ リソースのレベルで右クリックし、[ロールを定義] を選択して、[ロールの選択] ページを表示します。

   すべてのサーバを対象とするスコープ ロールを作成するには、[サーバ] を右クリックします。特定のサーバを対象とするスコープ ロールを作成するには、[サーバ] を展開してからサーバの名前を右クリックします。図 4-11 では、例として `examplesServer` を使用して、クリックする位置を示します。

図 4-11 Administration Console ナビゲーション ツリーのサーバ部分



スコープ ロールが定義されている場合は、定義済みスコープ ロールのテーブルが右ペインに表示されます。

URL リソース

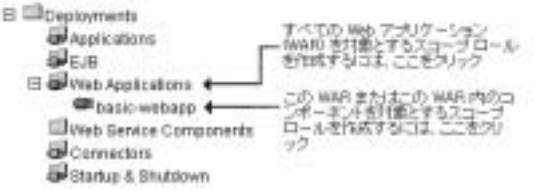
1. WebLogic Server Administration Console の左ペインで、[デプロイメント] を展開します。

   [デプロイメント] ノードを展開すると、デプロイ可能な WebLogic リソースのタイプが表示されます。

2. 作成するスコープ ロールの対象となる URL (Web) リソースのレベルで右クリックします。

   すべての Web アプリケーション (WAR) を対象とするスコープ ロールを作成するには、[Web アプリケーション] を右クリックします。特定の WAR、または WAR 内のコンポーネント (たとえば特定のサーブレットや JSP) を対象とするスコープ ロールを作成するには、[Web アプリケーション] を展開してから Web アプリケーション (WAR) の名前を右クリックします。図 4-12 では、例として `basic-webapp` WAR を使用して、クリックする位置を示します。

図 4-12 Administration Console ナビゲーション ツリーのデプロイメント部分



3. すべての Web アプリケーション (WAR) を対象とするスコープ ロールを作成する場合は、[ロールを定義] を選択して [ロールの選択] ページを表示します。

   特定の WAR または WAR 内のコンポーネントを対象とするスコープ ロールを作成する場合は、次の手順に従います。

a. [ロールを定義] を選択して [一般] タブを表示します。

b. テキスト フィールドに URL パターンを入力します。

URL パターンは、Web アプリケーションに含まれる特定のコンポーネントのパスです。または、/* を使用して、Web アプリケーション内のすべてのコンポーネント (サーブレット、JSP など) にスコープ ロールを関連付けることができます。

c. [ロールを定義] ボタンをクリックして、[ロールの選択] ページを表示します。

スコープ ロールが定義されている場合は、定義済みスコープ ロールのテーブルが右ペインに表示されます。

手順 2 : スコープ ロールを作成する

1. [新しい Role のコンフィグレーション] リンクをクリックします。

注意 : 複数の WebLogic ロール マッピング プロバイダがセキュリティ レルムでコンフィグレーションされている場合、新しいスコープ ロールの情報をどの WebLogic ロール マッピング プロバイダのデータベースに格納するかを選択する必要があります。

2. [一般] タブで、スコープ ロールの名前を [名前] フィールドに入力します。

注意 : スペース、カンマ、ハイフン、\t、< >、#、|、&、~、?、( )、{ } を使用しないでください。セキュリティ ロール名では大文字 / 小文字を区別します。BEA の命名規約では、セキュリティ ロール名は単数形で、先頭の文字は大文字です。

セキュリティ ロール名の適切な構文は、Extensible Markup Language (XML) 勧告で `Nmtoken` に関して定義されているとおりです。

警告 : グローバル ロールと同名のスコープ ロールを作成する場合、スコープ ロールがグローバル ロールに優先します。

3. [適用] をクリックして変更を保存します。

手順 3 : ロール条件を作成する

1. [条件] タブを選択して、ロール エディタ ページを表示します (図 4-13 を参照)。

図 4-13　ロール エディタ ページ

2. [ロール条件] リスト ボックスで、いずれかの条件をクリックします。さまざまなロール条件の詳細については、4-13 ページの「セキュリティ ロールの構成要素 : ロール条件、式、およびロール文」を参照してください。

   注意 : 可能であれば、[呼び出し側をメンバとするグループは] 条件を使用して式を作成することをお勧めします。グループを使用してセキュリティ ロールを作成すると、セキュリティ ロールはそのグループのすべてのメンバー (つまり複数のユーザ) に付与されます。

   JMS サブシステムはセキュリティ チェックを 1 回しか実行せず、[アクセス可能な時間帯は] 条件ではその後のセキュリティ チェックが必要になるので、JMS リソースを対象としてスコープ ロールを作成する場合は [アクセス可能な時間帯は] 条件を使用しないでください。

3. [追加] をクリックしてカスタマイズ ウィンドウを表示します。

4. [アクセス可能な時間帯は] 条件を選択した場合は、[時間制約] ウィンドウを使用して開始時刻と終了時刻を選択し、[OK] をクリックします。ウィンドウが閉じて、[ロール文] リスト ボックスに式が表示されます。

   他の条件のいずれかを選択した場合は、次の手順に従います。

   a. [ユーザ] または [グループ] ウィンドウでユーザまたはグループの名前を入力し、[追加] をクリックします。リスト ボックスに式が表示されます。

注意： 複数のユーザまたはグループを追加するには、この手順を複数回繰り返します。

b. 必要に応じて、リスト ボックスの右側にあるボタンを使用して式を変更します。

[上へ移動] および [下へ移動] をクリックすると、強調表示されたユーザ名またはグループ名の順序が変更されます。[変更] をクリックすると、式の間にある強調表示された and 文と or 文が切り替わります。[削除] をクリックすると、強調表示されたユーザ名またはグループ名が削除されます。

c. [OK] をクリックして、ロール文に式を追加します。ウィンドウが閉じて、[ロール文] リスト ボックスに式が表示されます。

5. 必要な場合は、手順 2 から 4 を繰り返して、別のロール条件に基づいて式を追加します。

6. 必要に応じて、[ロール文] リスト ボックスの右側にあるボタンを使用して式を変更します。

- [上へ移動] および [下へ移動] をクリックすると、強調表示された式の順序が変更されます。
- [変更] をクリックすると、式の間にある強調表示された and 文と or 文が切り替わります。
- [編集] をクリックすると、強調表示された式のカスタマイズ ウィンドウが再び開き、式を変更できます。
- [削除] をクリックすると、選択した式が削除されます。

7. [ロール文] リスト ボックスのすべての式が正しい場合は、[適用] をクリックします。

注意： [リセット] をクリックして、ロール エディタ ページを最初にロードしたときの状態に戻す (つまり、変更をすべて元に戻す) こともできます。

スコープ ロールの変更

WebLogic リソースのスコープ ロールを変更するには、次の手順に従います。

1. 4-20 ページの「手順 1 : WebLogic リソースを選択する」で説明されているように、該当する WebLogic リソースの [ロールの選択] ページに移動します。

   右ペインに、WebLogic リソースを対象とするすべてのスコープ ロールを示すテーブルが表示されます。

2. 変更するスコープ ロールをテーブルから選択します。
3. [条件] タブを選択します。
4. 4-28 ページの「手順 3 : ロール条件を作成する」 手順を参考にして変更を加えます。
5. [適用] をクリックして変更を保存します。

スコープ ロールの削除

WebLogic リソースのスコープ ロールを削除するには、次の手順に従います。

1. 4-20 ページの「手順 1 : WebLogic リソースを選択する」で説明されているように、該当する WebLogic リソースの [ロールの選択] ページに移動します。

   右ペインに、WebLogic リソースを対象とするすべてのスコープ ロールを示すテーブルが表示されます。

2. 削除するスコープ ロールと同じ行にあるごみ箱アイコンをクリックします。
3. [はい] をクリックして削除を確認します。
4. [続行] をクリックします。

   [ロールを選択] ページのテーブルには該当するスコープ ロールが表示されなくなります。

5　セキュリティ ポリシー

セキュリティ ポリシーは、権限のないアクセスから WebLogic リソースを保護するための、WebLogic リソースと 1 つまたは複数のユーザ、グループ、セキュリティ ロールとの関連付けです。

注意： セキュリティ ポリシーは、以前のリリースの WebLogic Server で WebLogic リソースを保護するために使用していたアクセス制御リスト (ACL) とパーミッションに代わるものです。

以下の節では、セキュリティ ポリシーの詳細について説明します。

セキュリティ ポリシーの粒度と継承

セキュリティ ポリシーは常に WebLogic リソースを対象としますが、WebLogic リソースは階層化されているので、自由なレベルで定義できます。たとえば、エンタープライズ アプリケーション (EAR) 全体、複数の EJB を含む EJB (エンタープライズ JavaBean) JAR、その JAR 内の特定の EJB、その EJB 内の単一のメソッドなどに対してセキュリティ ポリシーを定義できます。

あるタイプの WebLogic リソース (たとえば EJB リソース) に対してセキュリティ ポリシーを作成すると、その WebLogic リソースの新しいインスタンスはすべてそのセキュリティ ポリシーを継承します (WebLogic リソースのタイプの

詳細については第 2 章「WebLogic リソースのタイプ」を参照)。このようにセキュリティ ポリシーを継承すると、複数の WebLogic リソースを効率的に保護できます。WebLogic Server は、デフォルト セキュリティ ポリシーで各 WebLogic リソース タイプを保護しています。デフォルト セキュリティ ポリシーは、その WebLogic リソースのすべてのインスタンスによって継承されます。詳細については、5-3 ページの「デフォルト セキュリティ ポリシー」を参照してください。

WebLogic リソースの特定のインスタンスに対して作成されたセキュリティ ポリシーは、その WebLogic リソース タイプに割り当てられているセキュリティ ポリシーをオーバライドします。つまり、特定の EJB に対してセキュリティ ポリシーを作成すると、このセキュリティ ポリシーが使用され、EJB リソース タイプに対して作成したセキュリティ ポリシーは使用されません。

セキュリティ ポリシーの格納および使用の前提条件

セキュリティ ポリシーは、デフォルト (アクティブな) セキュリティ レルムにコンフィグレーションされている認可プロバイダのセキュリティ プロバイダ データベースに格納されます。デフォルトでは、WebLogic 認可プロバイダがコンフィグレーションされ、セキュリティ ポリシーは組み込み LDAP サーバに格納されます。

ユーザまたはグループを使用してセキュリティ ポリシーを作成する場合、そのユーザまたはグループは、デフォルト セキュリティ レルムでコンフィグレーション済みの認証プロバイダのセキュリティ プロバイダ データベースで定義されている必要があります。セキュリティ ロールを使用してセキュリティ ポリシーを作成する場合、そのセキュリティ ロール (グローバルまたはスコープ) は、デフォルト セキュリティ レルムでコンフィグレーション済みのロール マッピング プロバイダのセキュリティ プロバイダ データベースで定義されている必要があります。デフォルトでは、WebLogic 認証プロバイダと WebLogic ロール マッピング プロバイダがコンフィグレーションされており、これらのセキュリティ プロバイダのデータベース (および組み込み LDAP サーバ) にはデフォルト グループとデフォルト グローバル ロールが格納されています。

注意： WebLogic 認証、認可、およびロール マッピング プロバイダの詳細については、『WebLogic Security の紹介』の「WebLogic セキュリティ プロバイダ」を参照してください。

デフォルト セキュリティ ポリシー

WebLogic Server では、表 5-1 に示すセキュリティ ポリシーがデフォルトで定義されています。これらのセキュリティ ポリシーは、第 2 章「WebLogic リソースのタイプ」で説明している WebLogic リソースの各タイプごとに定義されており、デフォルト グローバル ロールとデフォルト グループに基づいています。

表 5-1 WebLogic リソースのデフォルト セキュリティ ポリシー

WebLogic リソース	セキュリティ ポリシー
管理リソース	デフォルト グローバル ロール：`Admin`
アプリケーション リソース	なし
COM リソース	なし
EIS リソース	デフォルト グループ：`Everyone`
EJB リソース	デフォルト グループ：`Everyone`
JDBC リソース	デフォルト グループ：`Everyone`
JNDI リソース	デフォルト グループ：`Everyone`
JMS リソース	デフォルト グループ：`Everyone`
サーバ リソース	デフォルト グローバル ロール： ■ `Admin` ■ `Operator`
URL リソース (以前の Web リソース (非推奨))	デフォルト グループ：`Everyone`

警告： 制限を強化するために管理リソースおよびサーバ リソースのデフォルト セキュリティ ポリシーを変更しないでください。既存のセキュリティ ロールの中には、削除すると WebLogic Server の機能に悪影響を与えるものがあります。ただし、新しいセキュリティ ポリシーを追加するなどして、デフォルト セキュリティ ポリシーをより包括的にすることはできます。

注意： 表 5-1 に示した WebLogic リソースの詳細については、第 2 章「WebLogic リソースのタイプ」を参照してください。

5-7 ページの「セキュリティ ポリシーの操作」で説明するとおり、独自のグループを作成してデフォルト セキュリティ ポリシーに追加することもできます。

保護されたパブリック インタフェース

WebLogic Server Administration Console、`weblogic.Admin` コマンド、および MBean API は、デフォルト セキュリティ ポリシーを使用して保護され、これらは表 4-1 および表 4-6 で説明されているデフォルト グローバル ロールおよびデフォルト グループに基づいています。したがって、Administration Console を使用するには、ユーザがこれらのデフォルト グループに属しているか、またはこれらのグローバル ロールのいずれかを付与されている必要があります。また、MBean との対話が必要な管理操作は、『管理者ガイド』の「システム管理操作の保護」で説明されている MBean の保護措置によって保護されています。したがって、以下の保護されたパブリック インタフェースと対話するには、両方のセキュリティ方式を満たす必要があります。

- WebLogic Server Administration Console - WebLogic Security サービスは、特定のユーザがログインしようとしたときに、そのユーザが Administration Console にアクセスできるかどうかを確認します。アクセス権を持たない操作をユーザが呼び出そうとした場合、「アクセスが拒否されました」というエラーが表示されます。

  このパブリック インタフェースの使用方法については、Administration Console オンライン ヘルプを参照してください。

- weblogic.Admin コマンド - WebLogic Security サービスは、ユーザがコマンドを実行しようとしたときに、そのコマンドを実行するパーミッションをユーザが持っているかどうかを確認します。ユーザがアクセス権を持たない操作を呼び出そうとした場合、WebLogic Server は

weblogic.management.NoAccessRuntimeException を送出します。開発者はこの例外をプログラムで明示的に捕捉することができます。この例外はサーバのログ ファイルに送信されますが、例外を標準出力に送信するよう、サーバをコンフィグレーションすることもできます。

このパブリック インタフェースの使い方については、4-7 ページの「保護されている MBean の属性および操作」および「WebLogic Server コマンドライン インタフェース リファレンス」の「weblogic.Admin Command-Line Reference」を参照してください。

注意： weblogic.Admin コマンドは、MBean API (後述) との対話を抽象化する便利なユーティリティです。したがって、weblogic.Admin コマンドを使用して管理タスクを実行するために、MBean API を使用して独自のコードを記述することもできます。

- MBean API - WebLogic Security サービスは、ユーザが MBean に対する操作を実行しようとしたときに、API にアクセスするパーミッションをユーザが持っているかどうかを確認します。ユーザがアクセス権を持たない操作を呼び出そうとした場合、WebLogic Server は weblogic.management.NoAccessRuntimeException を送出します。開発者はこの例外をプログラムで明示的に捕捉することができます。この例外はサーバのログ ファイルに送信されますが、例外を標準出力に送信するよう、サーバをコンフィグレーションすることもできます。

  この API の使用方法については、4-7 ページの「保護されている MBean の属性および操作」および『WebLogic JMX Service プログラマーズ ガイド』を参照してください。

セキュリティ ポリシーの構成要素：ポリシー条件、式、およびポリシー文

ポリシー条件とは、セキュリティ ポリシーを作成する際の条件です。このリリースの WebLogic Server で使用できるポリシー条件は以下のとおりです。

- [呼び出し側のユーザ名は] ― ユーザ名に基づいてセキュリティ ポリシーの条件を作成します。たとえば、ユーザ John だけが Deposit EJB にアクセスできるという条件を作成できます。

- [呼び出し側をメンバとするグループは] ― グループに基づいてセキュリティ ポリシーの条件を作成します。グループを使用してセキュリティ ポリシーを作成すると、セキュリティ ポリシーはそのグループのすべてのメンバーに割り当てられます。たとえば、グループ `FullTimeBankEmployees` に属するユーザだけが `Deposit` EJB にアクセスできるという条件を作成できます。
- [呼び出し側に許可するロールは] ― セキュリティ ロールに基づいてセキュリティ ポリシーの条件を作成します。たとえば、`BankTeller` セキュリティ ロールのユーザまたはグループだけが `Deposit` EJB にアクセスできるという条件を作成できます。
- [アクセス可能な時間帯は] ― 指定した時間に基づいてセキュリティ ポリシーの条件を作成します。たとえば、`BankTeller` セキュリティ ロールは銀行の営業時間中にだけ `Deposit` EJB にアクセスできるというセキュリティ ポリシーを作成できます。

これらのポリシー条件に対して特定の情報 (実際のユーザ名、グループ、セキュリティ ロール、開始 / 終了時間など) を指定したものは**式**と呼ばれます。WebLogic Server Administration Console に表示される式の例を図 5-1 に示します。

図 5-1　式の例

この式の例では、1 行目がポリシー条件、2 行目が条件に対して指定した特定の情報 (この場合は、`FullTimeBankEmployees` というグループ) です。

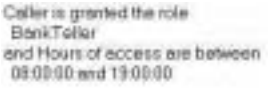
ポリシー文は、誰に WebLogic リソースへのアクセス権が付与されるかを定義する式の集合です。したがって、作成するセキュリティ ポリシーの主要部分となります。複数の式を使用できるため、企業のセキュリティ要件に合わせて複雑なセキュリティ ポリシーを作成できます。式の間の `and` と `or` の使い方、および式の順序も重要な機能です。

- `and` は、セキュリティ ポリシーが適用されるにはすべての式が true でなければならないことを指定するために使用します。
- `or` は、セキュリティ ポリシーが適用されるには少なくとも 1 つの式が true でなければならないことを指定するために使用します。

注意： セキュリティ ポリシーが適用されるには、ポリシー文全体が true でなければなりません。ポリシー文の中では、制約が厳しい式ほど後に指定します。WebLogic Server では、ポリシー文中の式は左から右に評価されます。

Administration Console に表示されるポリシー文の例を図 5-2 に示します。

図 5-2 ポリシー文の例



このポリシー文の例には 2 つの式があります。1 行目と 2 行目は [呼び出し側に許可するロールは] ポリシー条件に基づく式、3 行目と 4 行目は [アクセス可能な時間帯は] ポリシー条件に基づく別の式です。

セキュリティ ポリシーの操作

以下の節では、さまざまなタイプの WebLogic リソースに対するセキュリティ ポリシーの操作手順を説明します。

- 5-8 ページの「セキュリティ ポリシーの作成」
- 5-21 ページの「セキュリティ ポリシーの変更」
- 5-22 ページの「セキュリティ ポリシーの削除」

セキュリティ ポリシーの作成

注意： セキュリティ ポリシーの操作手順は、WebLogic リソースごとに若干異なります。この手順で示した WebLogic リソースのタイプごとの違いに注意して、適切な手順に従ってください。詳細については、第 2 章「WebLogic リソースのタイプ」を参照してください。

WebLogic Server のこのリリースでは、作成したセキュリティ ポリシーを常に追跡する必要があります。現在のところ、WebLogic Server Administration Console で作成済みのセキュリティ ポリシーのリストを表示するメカニズムは存在しません。

WebLogic リソースを対象とするセキュリティ ポリシーを作成するには、次の手順に従います。

- 5-8 ページの「手順 1 : WebLogic リソースを選択する」
- 5-20 ページの「手順 2 : ポリシー条件を作成する」

手順 1 : WebLogic リソースを選択する

該当する節の手順に従って WebLogic リソースのタイプを選択します。

- 5-9 ページの「管理リソース」
- 5-9 ページの「アプリケーション リソース」
- 5-10 ページの「COM リソース」
- 5-12 ページの「EIS リソース」
- 5-12 ページの「EJB リソース」
- 5-14 ページの「JDBC リソース」
- 5-15 ページの「JMS リソース」
- 5-16 ページの「JNDI リソース」
- 5-17 ページの「サーバ リソース」
- 5-18 ページの「URL リソース」

管理リソース

WebLogic Server Administration Console の左ペインで、WebLogic Server ドメインの名前 (たとえば `examples`) を右クリックし、[ポリシーを定義] を選択して、ポリシー エディタ ページ (5-10 ページの図 5-3「ポリシー エディタ ページ」を参照) を表示します。

注意 : このバージョンの WebLogic Server では、`unlockuser` メソッドのみを保護できます。ユーザ ロックアウトの詳細については、『WebLogic Security の管理』の「ユーザ アカウントの保護」を参照してください。

[呼び出し側に許可するロールは `: Admin`] ポリシー文は、選択した管理リソースのタイプに関連付けられているデフォルト セキュリティ ポリシーから継承されています。5-20 ページの「手順 2 : ポリシー条件を作成する」では、このデフォルト セキュリティ ポリシーをオーバライドします。詳細については、5-3 ページの「デフォルト セキュリティ ポリシー」および 5-1 ページの「セキュリティ ポリシーの粒度と継承」を参照してください。

アプリケーション リソース

1. WebLogic Server Administration Console の左ペインで、[デプロイメント | アプリケーション] を展開します。

   注意 : 必要に応じて、スコープ ロールを作成するエンタープライズ アプリケーション (EAR) を展開して、別のタイプの WebLogic リソースを表示します。

2. エンタープライズ アプリケーション (EAR) の名前を右クリックし、[ポリシーを定義] を選択して、ポリシー エディタ ページ (図 5-3 参照) を表示します。

図 5-3　ポリシー エディタ ページ

注意： アプリケーション リソースを対象とするデフォルト ポリシー文はありません (詳細については 5-3 ページの「デフォルト セキュリティ ポリシー」 を参照)。

COM リソース

COM クライアントからアクセスする EJB クラス (`ejb20.basic.beanManaged.*` など) のパッケージを対象とするセキュリティ ポリシーを作成する場合は、次の手順に従います。

1. WebLogic Server Administration Console の左ペインで、[デプロイメント | EJB] を展開します。

   [EJB] ノードを展開すると、デプロイ済みの EJB JAR が表示されます。

2. パッケージにアクセスするための EJB を格納している EJB JAR の名前を右クリックし、[個別の Bean のポリシーとロールを定義] を選択して、EJB のリストを表示します。

3. パッケージにアクセスするための EJB と同じ行の [JCOM ポリシーを定義] リンクをクリックします。

   [一般] タブの [COM クラス] フィールドには、セキュリティ ポリシーの対象とするパッケージの名前が表示されます。

   注意： [COM クラス] フィールドの値は、jCOM ブリッジを介して COM に公開される Java クラスまたはパッケージの名前です。

4. [ポリシーを定義] ボタンをクリックして、ポリシー エディタ ページ (5-10 ページの図 5-3「ポリシー エディタ ページ」参照) を表示します。

注意： COM クライアントからアクセスする EJB クラスのパッケージを対象とするセキュリティ ポリシーを作成し、[呼び出し側に許可するロールは] 条件でスコープ ロールを使用する場合は、EJB クラスのパッケージに関連付けられているスコープ ロールを使用してください (4-21 ページの「COM リソース」を参照)。

Java クラス (`java.util.*` など) または COM クライアントからアクセスする個々のクラス (`java.util.Collection` など) のパッケージを対象とするセキュリティ ポリシーを作成する場合は、次の手順に従います。

1. WebLogic Server Administration Console の左ペインで、[サービス] を展開します。

2. [JCOM] ノードを右クリックして、[ポリシーを定義] を選択します。

3. [一般] タブの [COM クラス] フィールドに、保護する Java クラスまたはパッケージの名前を入力してから、[ポリシーを定義] ボタンをクリックしてポリシー エディタ ページ (5-10 ページの図 5-3「ポリシー エディタ ページ」参照) を表示します。

   注意： [COM クラス] フィールドに入力する値は、jCOM ブリッジを介して COM に公開される Java クラスまたはパッケージの名前です。

   COM リソースを対象とするデフォルト ポリシー文はありません (詳細については 5-3 ページの「デフォルト セキュリティ ポリシー」を参照)。

EIS リソース

1. WebLogic Server Administration Console の左ペインで、[デプロイメント] を展開します。

   [デプロイメント] ノードを展開すると、デプロイ可能な WebLogic リソースのタイプが表示されます。

2. セキュリティ ポリシーの対象とする EIS リソースのレベルで右クリックし、[ポリシーを定義] を選択して、ポリシー エディタ ページ (5-10 ページの図 5-3「ポリシー エディタ ページ」参照) を表示します。

   1 つのセキュリティ ポリシーですべてのコネクタを保護するには、[コネクタ] を右クリックします。特定のコネクタを保護するには、[コネクタ] を展開してからコネクタの名前を右クリックします。図 5-4 では、例として `basic-connector` コネクタを使用して、クリックする位置を示します。

図 5-4　Administration Console ナビゲーション ツリーのデプロイメント部分

注意 : [呼び出し側に許可するロールは　: Everyone] ポリシー文は、選択した EIS リソースのタイプに関連付けられているデフォルト セキュリティ ポリシーから継承されています。5-20 ページの「手順 2 : ポリシー条件を作成する」では、このデフォルト セキュリティ ポリシーをオーバライドします。詳細については、5-3 ページの「デフォルト セキュリティ ポリシー」および 5-1 ページの「セキュリティ ポリシーの粒度と継承」を参照してください。

EJB リソース

注意 : ここで説明する手順は、メッセージ駆動型 Bean (MDB) にも当てはまります。

1. WebLogic Server Administration Console の左ペインで、[デプロイメント] を展開します。

[デプロイメント] ノードを展開すると、デプロイ可能な WebLogic リソースのタイプが表示されます。

2. 作成するセキュリティ ポリシーの対象となる EJB リソースのレベルで右クリックします。

   1 つのセキュリティ ポリシーですべての EJB JAR を保護するには、[EJB] を右クリックします。特定の EJB JAR、JAR 内の EJB、または JAR 内の EJB のメソッドを保護するには、[EJB] を展開してから EJB JAR の名前を右クリックします。図 5-5 では、例として `basic-ejbapp` JAR を使用して、クリックする位置を示します。

図 5-5 Administration Console ナビゲーション ツリーのデプロイメント部分

3. すべての EJB JAR または特定の EJB JAR を対象とするセキュリティ ポリシーを作成する場合は、[ポリシーを定義] を選択して、ポリシー エディタ ページ (5-10 ページの図 5-3「ポリシー エディタ ページ」参照) を表示します。

   EJB JAR 内の特定の EJB、または JAR 内のいずれかの EJB のメソッドを対象とするセキュリティ ポリシーを作成する場合は、次の手順に従います。

   a. [個別の Bean のポリシーとロールを定義] を選択して、EJB のリストを表示します。

   b. 保護の対象が EJB 全体でも EJB 内の特定のメソッドでも、保護する特定の EJB に対応する [ポリシーを定義] リンクをクリックして、ポリシー エディタ ページ (5-10 ページの図 5-3「ポリシー エディタ ページ」参照) を表示します。

      注意 : [呼び出し側に許可するロールは : `Everyone`] ポリシー文は、選択した EJB リソースのタイプに関連付けられているデフォルト セキュリティ ポリシーから継承されています。5-20 ページの「手順 2 : ポリシー条件を作成する」では、このデフォルト セキュリティ ポリシーをオーバライドします。詳細については、5-3 ページの

「デフォルト セキュリティ ポリシー」および 5-1 ページの「セキュリティ ポリシーの粒度と継承」を参照してください。

4. EJB JAR 内の特定の EJB を保護する場合は、保護する EJB メソッドを指定するか、[ALL] を選択してすべてのメソッドを保護します。

JDBC リソース

1. WebLogic Server Administration Console の左ペインで、[サービス | JDBC] を展開します。

   [JDBC] を展開すると、さまざまな JDBC コンポーネント (接続プール、マルチプール、およびデータ ソース) に対応するノードが表示されます。

2. セキュリティ ポリシーの対象とする JDBC リソースのレベルで右クリックし、[ポリシーを定義] を選択して、ポリシー エディタ ページ (5-10 ページの図 5-3「ポリシー エディタ ページ」参照) を表示します。

   1 つのセキュリティ ポリシーですべての接続プールを保護するには、[接続プール] を右クリックします。特定の接続プールを保護するには、[接続プール] を展開してから接続プールの名前を右クリックします。個々のマルチプールを保護するには、[マルチプール] を展開してからマルチプールの名前を右クリックします。

   注意： 1 つのセキュリティ ポリシーですべてのマルチプールを保護することはできません。

   セキュリティ ポリシーでマルチプール内の接続プールへのアクセスを制御する場合、アクセスのチェックは、JDBC リソース階層の 2 つのレベルで実行されます (マルチプールのレベルで 1 回、個々の接続プールのレベルで 1 回)。こうした二重のチェックをすべてのタイプの WebLogic リソースで実行することで、セキュリティ レベルの高い方のセキュリティ ポリシーがアクセスを制御することになります。

   図 5-6 では、例として接続プールとマルチプールを使用して、クリックする位置を示します。

図 5-6 Administration Console ナビゲーション ツリーのサービス部分

注意： [呼び出し側に許可するロールは ： Everyone] ポリシー文は、選択した JDBC リソースのタイプに関連付けられているデフォルト セキュリティ ポリシーから継承されています。5-20 ページの「手順 2 : ポリシー条件を作成する」では、このデフォルト セキュリティ ポリシーをオーバライドします。詳細については、5-3 ページの「デフォルト セキュリティ ポリシー」および 5-1 ページの「セキュリティ ポリシーの粒度と継承」を参照してください。

3. 特定の接続プールを保護する場合は、[Methods] ドロップダウン メニューを使用して保護するメソッドを指定するか、[ALL] を選択してすべてのメソッドを保護します。

JMS リソース

1. WebLogic Server Administration Console の左ペインで、[サービス | JMS] を展開します。

   [JMS] を展開すると、さまざまな JMS コンポーネント (接続ファクトリ、テンプレート、送り先キーなど) に対応するノードが表示されます。

2. セキュリティ ポリシーの対象とする JMS リソースのレベルで右クリックし、[ポリシーを定義] を選択して、ポリシー エディタ ページ (5-10 ページの図 5-3「ポリシー エディタ ページ」参照) を表示します。

   すべての JMS コンポーネントを対象とするセキュリティ ポリシーを作成するには、[JMS] を右クリックします。JMS サーバ上の特定の送り先 (JMS キューまたは JMS トピック) を対象とするセキュリティ ポリシーを作成するには、[サーバ | JMS サーバ | 送り先] ノードを展開してから送り先の名前を右クリックします。図 5-7 では、例として `examplesJMSServer` を使用して、クリックする位置を示します。

図 5-7　Administration Console ナビゲーション ツリーのサービス部分

注意： [呼び出し側に許可するロールは ： Everyone] ポリシー文は、選択した JMS リソースのタイプに関連付けられているデフォルト セキュリティ ポリシーから継承されています。5-20 ページの「手順 2 : ポリシー条件を作成する」では、このデフォルト セキュリティ ポリシーをオーバライドします。詳細については、5-3 ページの「デフォルト セキュリティ ポリシー」および 5-1 ページの「セキュリティ ポリシーの粒度と継承」を参照してください。

3. JMS サーバ上の特定の送り先を保護する場合は、[Methods] ドロップダウン メニューを使用して保護するメソッドを指定するか、[ALL] を選択してすべてのメソッドを保護します。

JNDI リソース

1. WebLogic Server Administration Console の左ペインで、[サーバ] を展開します。

   [サーバ] ノードを展開すると、現在の WebLogic Server ドメインで利用可能なサーバが表示されます。

2. 作成するセキュリティ ポリシーの対象となる JNDI リソースを含むサーバの名前 (たとえば `myserver`) を右クリックします。

3. メニューから [JNDI ツリーを見る] オプションを選択します。

   新しい Administration Console ウィンドウに、このサーバの JNDI ツリーが表示されます。

4. Administration Console ウィンドウで、セキュリティ ポリシーの対象とする JNDI ツリーのレベルで右クリックし、[ポリシーを定義] を選択して、ポリシー エディタ ページ (5-10 ページの図 5-3「ポリシー エディタ ページ」参照) を表示します。

   オブジェクトのグループを保護するには、オブジェクト タイプを表すノードを右クリックします。特定のオブジェクトを保護するには、そのオブジェクトを表すノードを展開してからオブジェクトの名前を右クリックします。図 5-8 では、例として `examplesServer` JNDI ツリーを使用して、クリックする位置を示します。

図 5-8 examplesServer JNDI ツリーを示す新しい Administration Console ウィンドウ

注意 : [呼び出し側に許可するロールは : `Everyone`] ポリシー文は、選択した JNDI リソースのタイプに関連付けられているデフォルト セキュリティ ポリシーから継承されています。5-20 ページの「手順 2 : ポリシー条件を作成する」では、このデフォルト セキュリティ ポリシーをオーバライドします。詳細については、5-3 ページの「デフォルト セキュリティ ポリシー」および 5-1 ページの「セキュリティ ポリシーの粒度と継承」を参照してください。

5. [Methods] ドロップダウン メニューを使用して保護する JNDI メソッドを指定するか、[ALL] を選択してすべてのメソッドを保護します。

サーバ リソース

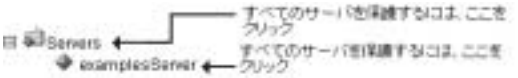
1. WebLogic Server Administration Console の左ペインで、[サーバ] を展開します。

   [サーバ] ノードを展開すると、保護可能なサーバ リソースのタイプが表示されます。

2. セキュリティ ポリシーの対象とするサーバ リソースのレベルで右クリックし、[ポリシーを定義] を選択して、ポリシー エディタ ページ (5-10 ページの図 5-3「ポリシー エディタ ページ」参照) を表示します。

すべてのサーバを対象とするセキュリティ ポリシーを作成するには、[サーバ] を右クリックします。特定のサーバを対象とするセキュリティ ポリシーを作成するには、[サーバ] を展開してからサーバの名前を右クリックします。図 5-9 では、例として examplesServer を使用して、クリックする位置を示します。

図 5-9　Administration Console ナビゲーション ツリーのサーバ部分



注意 : [呼び出し側に許可するロールは : Admin] または [呼び出し側に許可するロールは : Operator] ポリシー文は、選択したサーバ リソースのタイプに関連付けられているデフォルト セキュリティ ポリシーから継承されています。5-20 ページの「手順 2 : ポリシー条件を作成する」では、このデフォルト セキュリティ ポリシーをオーバライドします。詳細については、5-3 ページの「デフォルト セキュリティ ポリシー」および 5-3 ページの「デフォルト セキュリティ ポリシー」を参照してください。

3. [Methods] ドロップダウン メニューを使用して保護するメソッドを指定するか、[ALL] を選択してすべてのメソッドを保護します。

URL リソース

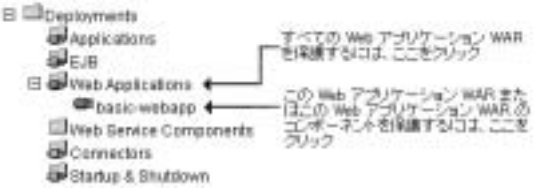
1. WebLogic Server Administration Console の左ペインで、[デプロイメント] を展開します。

   [デプロイメント] ノードを展開すると、デプロイ可能な WebLogic リソースのタイプが表示されます。

2. 作成するセキュリティ ポリシーの対象となる Web アプリケーション リソースのレベルで右クリックします。

   1 つのセキュリティ ポリシーですべての Web アプリケーション (WAR) を保護するには、[Web アプリケーション] を右クリックします。特定の WAR、または WAR のコンポーネント (たとえば特定のサーブレットや JSP) を保護するには、[Web アプリケーション] を展開してから Web アプリケーション (WAR) の名前を右クリックします。図 5-10 では、例として basic-webapp WAR を使用して、クリックする位置を示します。

図 5-10 Administration Console ナビゲーション ツリーのデプロイメント部分



3. すべての Web アプリケーション (WAR) を対象とするセキュリティ ポリシーを作成する場合は、[ポリシーを定義] を選択して、ポリシー エディタ ページ (5-10 ページの図 5-3「ポリシー エディタ ページ」参照) を表示します。

   特定の WAR または WAR 内のコンポーネントを対象とするセキュリティ ポリシーを作成する場合は、次の手順に従います。

   a. [ポリシーを定義] をクリックします。

   b. [一般] タブのテキスト フィールドに URL パターンを入力します。

      注意 : URL パターンは、Web アプリケーションに含まれる特定のサーブレットのパスです。または、/* を使用して Web アプリケーション内のすべてのサーブレットを保護することができます。

   c. [ポリシーを定義] ボタンをクリックして、ポリシー エディタ ページ (5-10 ページの図 5-3「ポリシー エディタ ページ」参照) を表示します。

      注意 : [呼び出し側に許可するロールは : Everyone] ポリシー文は、選択した URL リソースのタイプに関連付けられているデフォルト セキュリティ ポリシーから継承されています。5-20 ページの「手順 2 : ポリシー条件を作成する」では、このデフォルト セキュリティ ポリシーをオーバライドします。詳細については、5-3 ページの「デフォルト セキュリティ ポリシー」および 5-1 ページの「セキュリティ ポリシーの粒度と継承」を参照してください。

4. 特定の WAR または特定の WAR のコンポーネントを保護する場合は、保護するメソッドを指定するか、[ALL] を選択してすべてのメソッドを保護します。

手順 2：ポリシー条件を作成する

1. [ポリシー条件] リスト ボックスで、いずれかの条件をクリックします。さまざまなポリシー条件の詳細については、5-5 ページの「セキュリティ ポリシーの構成要素：ポリシー条件、式、およびポリシー文」を参照してください。

   注意： [呼び出し側に許可するロールは] 条件を使用して式を作成することをお勧めします。セキュリティ ロールに基づいて式を作成すると、複数のユーザまたはグループを考慮した 1 つのセキュリティ ポリシーを作成できます。管理の方法としてはこちらの方が効率的です。

2. [追加] をクリックしてカスタマイズ ウィンドウを表示します。

3. [アクセス可能な時間帯は] 条件を選択した場合は、[時間制約] ウィンドウを使用して開始時刻と終了時刻を選択し、[OK] をクリックします。ウィンドウが閉じて、[ポリシー文] リスト ボックスに式が表示されます。

   注意： JMS サブシステムはセキュリティ チェックを 1 回しか実行せず、[アクセス可能な時間帯は] 条件ではその後のセキュリティ チェックが必要になるので、JMS リソースを保護する場合はこの条件を使用しないでください。

   他の条件のいずれかを選択した場合は、次の手順に従います。

   a. [ユーザ]、[グループ]、または [ロール] ウィンドウを使用してユーザ、グループ、またはセキュリティ ロールの名前を入力し、[追加] ボタンをクリックします。リスト ボックスに式が表示されます。

      注意： 複数のユーザ、グループ、またはセキュリティ ロールを追加するには、この手順を複数回繰り返します。

   b. 必要に応じて、リスト ボックスの右側にあるボタンを使用して式を変更します。

      [上へ移動] および [下へ移動] をクリックすると、強調表示されたユーザ名またはグループ名の順序が変更されます。[変更] をクリックすると、式の間にある強調表示された and 文と or 文が切り替わります。[削除] をクリックすると、強調表示されたユーザ名またはグループ名が削除されます。

   c. [OK] をクリックして、ポリシー文に式を追加します。ウィンドウが閉じて、[ポリシー文] リスト ボックスに式が表示されます。

4. 必要な場合は、手順 1 から 3 を繰り返して、別のポリシー条件に基づいて式を追加します。
5. 必要に応じて、[ポリシー文] リスト ボックスの右側にあるボタンを使用して式を変更します。
   - [上へ移動] および [下へ移動] をクリックすると、強調表示された式の順序が変更されます。
   - [変更] をクリックすると、式の間にある強調表示された and 文と or 文が切り替わります。
   - [編集] をクリックすると、強調表示された式のカスタマイズ ウィンドウが再び開き、式を変更できます。
   - [削除] をクリックすると、選択した式が削除されます。
6. [ポリシー文] リスト ボックスのすべての式が正しい場合は、ページを下の方にスクロールして [適用] をクリックします。

   注意： [リセット] をクリックして、ポリシー エディタ ページを最初にロードしたときの状態に戻す (つまり、変更をすべて元に戻す) こともできます。

セキュリティ ポリシーの変更

WebLogic リソースを対象とするセキュリティ ポリシーを変更するには、次の手順に従います。

1. 5-8 ページの「手順 1 : WebLogic リソースを選択する」で説明されているように、該当する WebLogic リソースのポリシー エディタ ページに移動します。

   注意： [継承されたポリシー文] リスト ボックスをよく見て、どのセキュリティ ポリシーがオーバライドされるかを確認してください。
2. 5-20 ページの「手順 2 : ポリシー条件を作成する」を参考にして、変更を加えます。
3. [適用] をクリックして変更を保存します。

セキュリティ ポリシーの削除

WebLogic リソースを対象とするセキュリティ ポリシーを削除するには、次の手順に従います。

1. 5-8 ページの「手順 1 : WebLogic リソースを選択する」で説明されているように、該当する WebLogic リソースのポリシー エディタ ページに移動します。
2. [削除] をクリックして、セキュリティ ポリシー全体を削除します。
3. [適用] をクリックして変更を保存します。

6 例：Administration Console を使用した URL (Web) リソースの保護

この例では、すべてのデプロイ済み Web アプリケーションへのアクセスを、デフォルト グローバル セキュリティ ロールが付与されているユーザに制限します。次に、basicauth Web アプリケーションへのアクセスを別のユーザに制限します。最後に、スコープ ロールを使用して、Web アプリケーション内の特定の JSP (welcome.jsp) に対するセキュリティをさらに強化します。

注意： この例に進む前に、2-7 ページの「URL リソースおよび EJB リソースを保護する方法」、2-10 ページの「URL リソースおよび EJB リソースを保護するための前提条件」、および 4-3 ページの「セキュリティ ロールのタイプ：グローバル ロールとスコープ ロール」に目を通しておいてください。

WebLogic Server Administration Console を使用して URL (Web) リソースを保護するには、次の手順に従います。

- 6-2 ページの「手順 1：サーバと前提設定を指定する」
- 6-3 ページの「手順 2：ユーザを作成する」
- 6-4 ページの「手順 3：ユーザをグループに追加する」
- 6-4 ページの「手順 4：グループにグローバル ロールを付与する」
- 6-5 ページの「手順 5：グローバル ロールを使用してすべての URL (Web) リソースのセキュリティ ポリシーを作成する」
- 6-6 ページの「手順 6：Web アプリケーションへのアクセスを試行する」
- 6-7 ページの「手順 7: basicauth Web アプリケーションへのアクセスを制限する」
- 6-9 ページの「手順 8：スコープ ロールを作成する」

- 6-10 ページの「手順 9 : グループにスコープ ロールを付与する」
- 6-10 ページの「手順 10 : スコープ ロールを使用してウエルカム JSP へのアクセスを制限する」

手順 1 : サーバと前提設定を指定する

1. 2-11 ページの「fullyDelegateAuthorization フラグの変更方法」の指示に従って、`fullyDelegateAuthorization` フラグを `true` に設定します。

   注意 : この設定の意味 : すべての URL (Web) および EJB リソースに対して WebLogic Security サービスによるセキュリティ チェックを実行するように WebLogic Server に指示します。詳細については、2-10 ページの「fullyDelegateAuthorization フラグについて」を参照してください。

2. Windows の [スタート] メニューから、[プログラム | BEA WebLogic Platform 7.0 | WebLogic Server 7.0 | Server Tour and Examples | Launch Examples Server] を選択して `examplesServer` というサーバを起動します。

   `examplesServer` が起動するとコンソールに `fullyDelegateAuthorization` フラグが表示され、ブラウザに [BEA WebLogic Server Out-of-the-Box Examples Index Page] が表示されます。

3. [BEA WebLogic Server Out-of-the-Box Examples Index Page] の上部にある [Administration Console] リンクをクリックします。

4. [サインイン] ボタンをクリックして `examplesServer` の Administration Console にサインインします。

5. Administration Console の左側のナビゲーション ツリーを使用して、[セキュリティ | レルム] を展開します。

6. `myrealm` セキュリティ レルムをクリックします。

7. [一般] タブで、[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスをクリックします (つまり、ボックスにチェック マークを入れます)。

   注意 : この設定の意味 : Administration Console を使用して、Web アプリケーションおよび EJB リソースのセキュリティを設定するように

WebLogic Server に指示します。詳細については、2-15 ページの「[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスについて」を参照してください。

8. [適用] をクリックして変更を保存します。

手順 2：ユーザを作成する

1. Administration Console の左側のナビゲーション ツリーを使用して、[セキュリティ | レルム] を展開します。
2. `myrealm` セキュリティ レルムを展開します。
3. [ユーザ] をクリックします。

   [ユーザを選択] には、WebLogic 認証プロバイダのデータベースで現在定義されているすべてのユーザが表示されます。
4. [新しいユーザのコンフィグレーション] リンクをクリックして、[ユーザの作成] ページを表示します。
5. [一般] タブの [名前] フィールドに `Tom` と入力します。
6. 必要な場合は、[記述] フィールドにユーザの説明を入力します。
7. [パスワード] および [パスワードの確認] フィールドに `webexample` と入力します。
8. [適用] をクリックして変更を保存します。
9. 手順 4 から 8 を繰り返して `Neil` というユーザを作成します。
10. ナビゲーション ツリーを使用して [ユーザ] をクリックし、ユーザ `Tom` および `Neil` が追加されていることを確認します。

    [ユーザを選択] ページを表示すると、Tom と Neil が WebLogic 認証プロバイダのデータベースに追加されていることがわかります。

手順 3：ユーザをグループに追加する

注意： 新しいグループを作成する代わりに、ここでは WebLogic Server のデフォルト グループの 1 つを使用します。

1. [ユーザを選択] ページで、ユーザ名 `Tom` のリンクをクリックします。
2. [グループ] タブをクリックします。
3. [指定できるグループ] リスト ボックスで、`Administrators` グループをクリックして強調表示します。
4. 強調表示された右矢印をクリックして、[指定できるグループ] リスト ボックスから [現在のグループ] リスト ボックスに `Administrators` グループを移動します。
5. [適用] をクリックして変更を保存します。

注意： `Neil` は `Administrators` グループに追加しないでください。

手順 4：グループにグローバル ロールを付与する

注意： `Administrators` というデフォルト グループには `Admin` というデフォルト グローバル ロールが自動的に付与されるため、グローバル ロールを作成したり、そのグローバル ロールを `Administrators` グループに付与したりする必要はありません。

ただし、グループにグローバル ロールが付与されていることを確認する場合は、次の手順に従います。

1. ナビゲーション ツリーを使用して [ロール] をクリックします。

   [ロールの選択] ページには、WebLogic ロール マッピング プロバイダのデータベースで現在定義されているすべてのグローバル ロールが表示されます。

2. グローバル ロール名 Admin のリンクをクリックします。
3. [条件] タブをクリックします。

   [ロール文] リスト ボックスに次のように表示されます。

   呼び出し側をメンバとするグループは

   ```
   Administrators
   ```

手順 5 : グローバル ロールを使用してすべての URL (Web) リソースのセキュリティ ポリシーを作成する

1. ナビゲーション ツリーを使用して、[デプロイメント] を展開してから、[Web アプリケーション] を右クリックします。
2. メニューから [ポリシーを定義] オプションを選択して、ポリシー エディタ ページを表示します。

   注意 : このオプションの意味 : すべてのデプロイ済み Web アプリケーションとそのコンポーネントを対象とするセキュリティ ポリシーを作成します。
3. [ポリシー条件] リスト ボックスで、[呼び出し側に許可するロールは] を強調表示します。
4. [追加] をクリックして [ロール] ウィンドウを表示します。
5. [ロール名の入力] フィールドに Admin と入力します。
6. [追加] をクリックしてから [OK] をクリックします。

   [ロール] ウィンドウが閉じます。[ポリシー文] リスト ボックスに次のように表示されます。

   呼び出し側をメンバとするグループは

   ```
   Everyone
   ```

   and 呼び出し側に許可するロールは

```
Admin
```

注意： 表示される [呼び出し側をメンバとするグループは] ポリシー条件は、URL リソースのデフォルト セキュリティ ポリシーの一部です。詳細については、5-3 ページの「デフォルト セキュリティ ポリシー」を参照してください。

7. [呼び出し側をメンバとするグループは] ポリシー条件を強調表示し、[削除] をクリックします。

   [ポリシー文] リスト ボックスに次のように表示されます。

   呼び出し側に許可するロールは

   ```
   Admin
   ```

8. [適用] をクリックして変更を保存します。

手順 6：Web アプリケーションへのアクセスを試行する

注意： この節で説明する手順はすべて、Windows 環境での作業を想定しています。

1. 「Basic Authentication Sample Web Application」を入手します (dev2dev Web サイトの「Code Samples: Weblogic Server 」からダウンロード可)。
2. `basicauth.zip` ファイルを一時ディレクトリ (`C:\basicauth` など) に展開します。
3. `basicauth` Web アプリケーションをデプロイして、`examplesServer` に割り当てます。

   注意： Web アプリケーションをデプロイする手順については、『WebLogic Server アプリケーションの開発』の「Administration Console を使用した J2EE アプリケーションのデプロイ」を参照してください。

4. Web ブラウザを開いて `http://localhost:7001/basicauth` と入力します。

   ユーザ名とパスワードを要求されます。

5. ユーザ名フィールドに Neil、パスワード フィールドに webexample と入力して、[OK] をクリックします。

   ユーザ名とパスワードを再び要求されます。

6. ユーザ名フィールドに Tom、パスワード フィールドに webexample と入力して、[OK] をクリックします。

   ブラウザに図 6-1 のようなページが表示されます。

図 6-1 ブラウザベースの認証サンプルの Web ページ

グローバル セキュリティ ロール Admin (ユーザ Tom は付与されているが、ユーザ Neil は付与されていない) に基づいたセキュリティ ポリシーによって、(basicauth Web アプリケーションを含む) すべての URL (Web) リソースを保護したため、このような結果になります。

注意 : この手順の後に誤って Web ブラウザを閉じてしまった場合は、http://localhost:7001/console と入力し、[サインイン] ボタンをクリックすると Administration Console に戻ります。WebLogic Server を実行しているコンソール ウィンドウを誤って閉じてしまい、6-2 ページの「手順 1 : サーバと前提設定を指定する」の手順 1 から 3 を実行しようとする場合、まず Tom/webexample を使用してログインする必要があります。この手順ですべての Web アプリケーションを保護すると、examplesWebapp も保護されるからです。

手順 7: basicauth Web アプリケーションへのアクセスを制限する

1. Administration Console の左側のナビゲーション ツリーを使用して、[Web アプリケーション] を展開してから、basicauth を右クリックします。

2. メニューから [ポリシーを定義] オプションを選択します。

   注意： このオプションの意味 : 特定の Web アプリケーションまたは Web アプリケーション内の特定のコンポーネントに対してセキュリティ ポリシーを作成できます。

3. [一般] タブの [URL パターン] フィールドに `/*` を入力します。

   注意： `/*` という URL パターンを使用すると、`basicauth` Web アプリケーション内のすべてのコンポーネント (JSP とサーブレットを含む) が保護されます。

4. [ポリシーを定義] ボタンをクリックして続行します。

5. [ポリシー条件] リスト ボックスで、[呼び出し側のユーザ名は] を強調表示します。

   注意： [Methods] ドロップダウン メニューに表示される値は変更しないでください (`ALL` と表示されています)。

6. [追加] をクリックして [ユーザ] ウィンドウを表示します。

7. [ユーザ名の入力] フィールドに `Neil` と入力します。

8. [追加] をクリックしてから [OK] をクリックします。

   [ユーザ] ウィンドウが閉じます。[ポリシー文] リスト ボックスに次のように表示されます。

   呼び出し側のユーザ名は

   ```
   Neil
   ```

   注意： `basicauth` Web アプリケーションのセキュリティ ポリシーを定義すると、「手順 5 : グローバル ロールを使用してすべての URL (Web) リソースのセキュリティ ポリシーを作成する」ですべての URL (Web) リソースに対して定義したセキュリティ ポリシーがオーバライドされることに注意してください。

   呼び出し側に許可するロールは

   ```
   Admin
   ```

   上記は [継承されたポリシー文] リスト ボックスに表示されます。

9. [適用] をクリックして変更を保存します。

10. 6-6 ページの「手順 6 : Web アプリケーションへのアクセスを試行する」の 4 から 6 を繰り返します。

    basicauth Web アプリケーションの動作は反対になります。つまり、ユーザ Tom として basicauth Web アプリケーションにアクセスしようとすると、ユーザ名とパスワードを再び要求されます。ユーザ Neil としてアクセスすると、ブラウザには図 6-1 で示したページが表示されますが、「Welcome Neil」と表示されます。

    特定のユーザ (このケースではユーザ Neil) に基づくセキュリティ ポリシーによって basicauth Web アプリケーション内のすべてのコンポーネントを保護したため、このような結果になります。

手順 8 : スコープ ロールを作成する

1. Administration Console の左側のナビゲーション ツリーを使用して、basicauth を右クリックします。
2. メニューから [ロールを定義] オプションを選択します。

   注意 : このオプションの意味 : 特定の Web アプリケーションを対象とするセキュリティ ロールを作成できます。それ以降、そのスコープ ロールはこの Web アプリケーションのセキュリティ ポリシーでのみ使用されます。

3. [一般] タブの [URL パターン] フィールドに /* を入力します。

   注意 : /* という URL パターンを使用すると、セキュリティ ロールの対象は basicauth Web アプリケーション内のすべてのコンポーネント (JSP とサーブレットを含む) になります。

4. [ロールを定義] ボタンをクリックして続行します。
5. [新しい Role のコンフィグレーション] リンクをクリックして、[ロールを作成] ページを表示します。
6. [一般] タブの [名前] フィールドに AppAdmin と入力します。
7. [適用] をクリックして変更を保存します。

手順 9：グループにスコープ ロールを付与する

1. [条件] タブをクリックします。
2. [ロール条件] リスト ボックスで、[呼び出し側をメンバとするグループは] を強調表示します。
3. [追加] をクリックして [グループ] ウィンドウを表示します。
4. [グループ名の入力] フィールドに `Administrators` と入力します。
5. [追加] をクリックしてから [OK] をクリックします。

   [グループ] ウィンドウが閉じます。[ロール文] リスト ボックスに次のように表示されます。

   呼び出し側をメンバとするグループは

   ```
   Administrators
   ```
6. [適用] をクリックして変更を保存します。

手順 10：スコープ ロールを使用してウエルカム JSP へのアクセスを制限する

1. Administration Console の左側のナビゲーション ツリーを使用して、`basicauth` を右クリックします。
2. メニューから [ポリシーを定義] オプションを選択します。

   注意： このオプションの意味：特定の Web アプリケーションまたは Web アプリケーション内の特定のコンポーネントに対してセキュリティ ポリシーを作成できます。
3. [一般] タブの [URL パターン] フィールドに `/welcome.jsp` と入力します。
4. [ポリシーを定義] ボタンをクリックして続行します。

5. [ポリシー条件] リスト ボックスで、[呼び出し側に許可するロールは] を強調表示します。

   注意： [Methods] ドロップダウン メニューに表示される値は変更しないでください (ALL と表示されています)。

6. [追加] をクリックして [ロール] ウィンドウを表示します。

7. [ロール名の入力] フィールドに AppAdmin と入力します。

8. [追加] をクリックしてから [OK] をクリックします。

   [ロール] ウィンドウが閉じます。[ポリシー文] リスト ボックスに次のように表示されます。

   呼び出し側に許可するロールは

   ```
   AppAdmin
   ```

   注意： welcome.jsp に対するこのセキュリティ ポリシーを定義すると、「手順 7: basicauth Web アプリケーションへのアクセスを制限する」で basicauth Web アプリケーションに対して定義したセキュリティ ポリシーがオーバライドされることに注意してください。具体的には、以下の継承されたポリシー文がオーバライドされます。

   呼び出し側のユーザ名は

   ```
   Neil
   ```

   上記の文は、[継承されたポリシー文] リスト ボックスに表示されます。

9. [適用] をクリックして変更を保存します。

10. 6-6 ページの「手順 6 : Web アプリケーションへのアクセスを試行する」の 4 から 6 を繰り返します。

    basicauth Web アプリケーションの動作は反対になります。つまり、ユーザ Neil として basicauth Web アプリケーションの welcome.jsp にアクセスしようとすると、ユーザ名とパスワードを再び要求されます。ユーザ Tom としてアクセスすると、ブラウザには図 6-1 で示したページが表示されます。

    スコープ セキュリティ ロール AppAdmin (ユーザ Tom は付与されているが、ユーザ Neil は付与されていない) に基づいたセキュリティ ポリシーによって、welcome.jsp ページを保護したため、このような結果になります。

7　例：エンタープライズ JavaBean (EJB) リソースの保護

この例では、`ejb20_basic_statelessSession` JAR 内のすべて EJB へのアクセスを、作成したグローバルセキュリティ ロールが付与されているユーザに制限します。次に、この EJB JAR に含まれる `statelessSession` EJB へのアクセスを別のユーザに制限します。最後に、特定の EJB メソッド (`create()` メソッドと `buy()` メソッド) に対するセキュリティをさらに強化します。

注意： この例に進む前に、2-7 ページの「URL リソースおよび EJB リソースを保護する方法」、2-10 ページの「URL リソースおよび EJB リソースを保護するための前提条件」、および 4-3 ページの「セキュリティ ロールのタイプ : グローバル ロールとスコープ ロール」に目を通しておいてください。

WebLogic Server Administration Console を使用してエンタープライズ JavaBean (EJB) を保護するには、次の手順に従います。

- 7-2 ページの「手順 1 : サーバと前提設定を指定する」
- 7-3 ページの「手順 2 : グループを作成する」
- 7-3 ページの「手順 3 : ユーザを作成する」
- 7-4 ページの「手順 4 : ユーザをグループに追加する」
- 7-4 ページの「手順 5 : グローバル ロールを作成する」
- 7-5 ページの「手順 6 : グループにグローバル ロールを付与する」
- 7-6 ページの「手順 7 : グローバル ロールを使用して statelessSession EJB JAR のセキュリティ ポリシーを作成する」
- 7-7 ページの「手順 8 : クライアント アプリケーションから EJB へのアクセスを試行する」
- 7-9 ページの「手順 9 : statelessSession EJB へのアクセスを制限する」

- 7-11 ページの「手順 10 : create() および buy() EJB メソッドへのアクセスを制限する」

手順 1 : サーバと前提設定を指定する

1. 2-11 ページの「fullyDelegateAuthorization フラグの変更方法」の指示に従って、`fullyDelegateAuthorization` フラグを `true` に設定します。

   注意 : この設定の意味 : すべての URL (Web) および EJB リソースに対して WebLogic Security サービスによるセキュリティ チェックを実行するよう WebLogic Server に指示します。詳細については、2-10 ページの「fullyDelegateAuthorization フラグについて」を参照してください。

2. Windows の [スタート] メニューから、[プログラム | BEA WebLogic Platform 7.0 | WebLogic Server 7.0 | Server Tour and Examples | Launch Examples Server] を選択して `examplesServer` というサーバを起動します。

   examplesServer が起動するとコンソールに `fullyDelegateAuthorization` フラグが表示され、ブラウザに [BEA WebLogic Server Out-of-the-Box Examples Index Page] が表示されます。

3. [BEA WebLogic Server Out-of-the-Box Examples Index Page] の上部にある [Administration Console] リンクをクリックします。

4. [サインイン] ボタンをクリックして `examplesServer` の Administration Console にサインインします。

5. Administration Console の左側のナビゲーション ツリーを使用して、[セキュリティ | レルム] を展開します。

6. `myrealm` セキュリティ レルムをクリックします。

7. [一般] タブで、[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスをクリックします (つまり、ボックスにチェック マークを入れます)。

   注意 : この設定の意味 : Administration Console を使用して、Web アプリケーションおよび EJB リソースのセキュリティを設定するように WebLogic Server に指示します。詳細については、2-15 ページの「[デ

プロイメント記述子内のセキュリティ データを無視] チェック ボックスについて」を参照してください。

8. [適用] をクリックして変更を保存します。

手順 2：グループを作成する

1. Administration Console の左側のナビゲーション ツリーを使用して、[セキュリティ | レルム] を展開します。
2. `myrealm` セキュリティ レルムを展開します。
3. [グループ] をクリックします。

   [グループを選択] ページには、WebLogic 認証プロバイダのデータベースで現在定義されているすべてのグループが表示されます。
4. [新しいグループのコンフィグレーション] リンクをクリックして、[グループの作成] ページを表示します。
5. [一般] タブの [名前] フィールドに `Testers` と入力します。
6. 必要な場合は、[記述] フィールドにグループの説明を入力します。
7. [適用] をクリックして変更を保存します。

手順 3：ユーザを作成する

1. ナビゲーション ツリーを使用して [ユーザ] をクリックします。

   [ユーザを選択] ページには、WebLogic 認証プロバイダのデータベースで現在定義されているすべてのユーザが表示されます。
2. [新しいユーザのコンフィグレーション] リンクをクリックして、[ユーザの作成] ページを表示します。
3. [一般] タブの [名前] フィールドに `Stephanie` と入力します。

4. 必要な場合は、[記述] フィールドにユーザの説明を入力します。
5. [パスワード] および [パスワードの確認] フィールドに `ejbexample` と入力します。
6. [適用] をクリックして変更を保存します。
7. 手順 2 から 6 を繰り返して `Jen` というユーザを作成します。
8. ナビゲーション ツリーを使用して [ユーザ] をクリックし、ユーザ `Stephanie` および `Jen` が追加されていることを確認します。

   [ユーザを選択] ページを表示すると、Stephanie と Jen が WebLogic 認証プロバイダのデータベースに追加されていることがわかります。

手順 4：ユーザをグループに追加する

1. [ユーザを選択] ページで、ユーザ名 `Stephanie` のリンクをクリックします。
2. [グループ] タブをクリックします。
3. [指定できるグループ] リスト ボックスで、`Testers` グループを強調表示します。
4. 強調表示された右矢印をクリックして、[指定できるグループ] リスト ボックスから [現在のグループ] リスト ボックスに `Testers` グループを移動します。
5. [適用] をクリックして変更を保存します。

注意： `Jen` は `Testers` グループに追加しないでください。

手順 5：グローバル ロールを作成する

1. ナビゲーション ツリーを使用して [ロール] をクリックします。

[ロールの選択] ページには、WebLogic ロール マッピング プロバイダのデータベースで現在定義されているすべてのグローバル ロールが表示されます。

2. [新しい Role のコンフィグレーション] リンクをクリックして、[ロールを作成] ページを表示します。
3. [一般] タブの [名前] フィールドに `QA` と入力します。
4. [適用] をクリックして変更を保存します。

手順 6：グループにグローバル ロールを付与する

1. [条件] タブをクリックします。
2. [ロール条件] リスト ボックスで、[呼び出し側をメンバとするグループは] を強調表示します。
3. [追加] をクリックして [グループ] ウィンドウを表示します。
4. [グループ名の入力] フィールドに `Testers` と入力します。
5. [追加] をクリックしてから [OK] をクリックします。

   [グループ] ウィンドウが閉じます。[ロール文] リスト ボックスに次のように表示されます。

   呼び出し側をメンバとするグループは

   ```
   Testers
   ```

6. [適用] をクリックして変更を保存します。

手順 7：グローバル ロールを使用して statelessSession EJB JAR のセキュリティ ポリシーを作成する

1. ナビゲーション ツリーを使用して、[デプロイメント | EJB] を展開します。
2. `ejb20_basic_statelessSession.jar` を右クリックします。
3. メニューから [ポリシーを定義] オプションを選択します。

   注意： このオプションの意味 : EJB JAR レベルでセキュリティ ポリシーを作成します。JAR 内のすべての EJB とその EJB 内のすべてのメソッドが含まれます。
4. [ポリシー条件] リスト ボックスで、[呼び出し側に許可するロールは] を強調表示します。
5. [追加] をクリックして [ロール] ウィンドウを表示します。
6. [ロール名の入力] フィールドに `QA` と入力します。
7. [追加] をクリックしてから [OK] をクリックします。

   [ロール] ウィンドウが閉じます。[ポリシー文] リスト ボックスに次のように表示されます。

   呼び出し側に許可するロールは

   ```
   QA
   ```

   注意： `ejb20_basic_statelessSession.jar` のセキュリティ ポリシーを定義すると、EJB リソース タイプ対して既に定義されているセキュリティ ポリシーがオーバライドされることに注意してください。具体的には、以下の継承されたポリシー文がオーバライドされます。

   呼び出し側をメンバとするグループは

   ```
   Everyone
   ```

   上記は [継承されたポリシー文] リスト ボックスに表示されます。

   この [呼び出し側をメンバとするグループは] ポリシー条件は、EJB リソースのデフォルト セキュリティ ポリシーの一部です。詳細につ

いては、5-3 ページの「デフォルト セキュリティ ポリシー」を参照してください。

8. [適用] をクリックして変更を保存します。

手順 8：クライアント アプリケーションから EJB へのアクセスを試行する

注意： この節で説明する手順はすべて、Windows 環境での作業を想定しています。

1. DOS シェルを開いて、`cd` *`WL_HOME`*`\samples\server\config\examples` と入力します。*`WL_HOME`* は、WebLogic Platform の最上位のインストール ディレクトリです。
2. `setExamplesEnv.cmd` と入力して、環境を設定します。
3. `cd ..\..\src\examples\security\jaas` と入力します。
4. `ant` と入力してサンプルをビルドします。
5. `sample_jaas.config` ファイルを *`WL_HOME`*`\samples\server\src\examples\security\jaas` ディレクトリから *`JAVA_HOME`*`\jre\lib\security` ディレクトリに手動でコピーします。*`JAVA_HOME`* は Java SDK がインストールされているディレクトリです。
6. `java.security` ファイル (*`JAVA_HOME`*`\jre\lib\security` に格納) を編集して、ファイルの最後に次の行を (すべて 1 行で) 追加します。

   ```
   login.config.url.1=file:${java.home}/lib/security/
   sample_jaas.config
   ```

7. `examplesServer` を再起動します。詳細については、『管理者ガイド』の「WebLogic Server の起動と停止」を参照してください。
8. *`WL_HOME`*`\samples\server\src\examples\security\jaas` ディレクトリで、`build.xml` ファイルを次のように編集します。

a. ファイルの最後までスクロールして <target name="run"> という行を見つけます (コード リスト 7-1 に太字で示されています)。

b. <arg line> 要素で、ユーザ名とパスワード (現在は weblogic weblogic) を Stephanie ejbexample に変更します (コード リスト 7-1 に太字で示されています)。

c. build.xml ファイルを保存します。

コード リスト 7-1　build.xml ファイルの該当する部分

```
<!-- Run the example -->
   <target name="run" >
       <java classname="examples.security.jaas.SampleClient"
       fork="yes" failonerror="true">
          <arg line="t3://localhost:${PORT} weblogic weblogic"/>
          <classpath>
             <pathelement path="${CLASSPATH};${CLIENT_CLASSES}/
             ejb20_basic_statelessSession_client.jar;
             ${CLIENT_CLASSES}/utils_common.jar"/>
          </classpath>
       </java>
    </target>
```

9. 同じディレクトリ (*WL_HOME*\samples\server\src\examples\security\jaas) で、ant run と入力します。

   以下のような出力が表示されます。

```
Buildfile: build.xml

run:
   [java] username: Stephanie
   [java] password: **********
   [java] URL: t3://localhost:7001
   [java] Creating a trader
   [java] Buying 100 shares of BEAS.
   [java] Buying 200 shares of MSFT.
   [java] Buying 300 shares of AMZN.
   [java] Buying 400 shares of HWP.
   [java] Selling 100 shares of BEAS.
   [java] Selling 200 shares of MSFT.
   [java] Selling 300 shares of AMZN.
   [java] Selling 400 shares of HWP.
   [java] Removing the trader
```

```
BUILD SUCCESSFUL
Total time: 5 seconds
```

セキュリティ ポリシーで保護した ejb20_basic_statelessSession.jar に格納されている EJB をクライアント アプリケーションが呼び出したため、このような結果になります。

10. build.xml ファイルのユーザ名とパスワードとして Jen ejbexample を使用し、手順 8 と 9 を繰り返します。

    以下で始まる出力が表示されます。

```
run:
   [java] username: Jen
   [java] password: **********
   [java] URL: t3://localhost:7001
   [java] Creating a trader
   [java] java.rmi.AccessException: Security violation: User
Jen has insufficient permission to access method; nested
exception is:
   [java] java.lang.SecurityException: Security violation:
User Jen has insufficient permission to access method
```

セキュリティ ポリシーで保護した ejb20_basic_statelessSession.jar に格納されている EJB をクライアント アプリケーションが呼び出したため、このような結果になります。

手順 9 : statelessSession EJB へのアクセスを制限する

1. Administration Console の左側のナビゲーション ツリーを使用して、ejb20_basic_statelessSession.jar を右クリックします。
2. メニューから [個別の Bean のポリシーとロールを定義] オプションを選択します。

   JAR ファイル内のすべての EJB (この場合は statelessSession EJB のみ) を示すテーブルが表示されます。

注意 : このオプションの意味 : EJB レベル (セキュリティ ポリシーは EJB 内のすべてのメソッドに適用される)、または EJB 内の特定のメソッド レベルでセキュリティ ポリシーを作成できます。

3. `statelessSession` EJB の [ポリシーを定義] リンクをクリックします。
4. [ポリシー条件] リスト ボックスで、[呼び出し側のユーザ名は] を強調表示します。

   注意 : [Methods] ドロップダウン メニューに表示される値は変更しないでください (`ALL` と表示されています)。
5. [追加] をクリックして [ユーザ] ウィンドウを表示します。
6. [ユーザ名の入力] フィールドに `Jen` と入力します。
7. [追加] をクリックしてから [OK] をクリックします。

   [ユーザ] ウィンドウが閉じます。[ポリシー文] リスト ボックスに次のように表示されます。

   呼び出し側のユーザ名は

   ```
   Jen
   ```

   注意 : `statelessSession` EJB に対するこのセキュリティ ポリシーを定義すると、「手順 7 : グローバル ロールを使用して statelessSession EJB JAR のセキュリティ ポリシーを作成する」で EJB JAR に対して定義したセキュリティ ポリシーがオーバライドされることに注意してください。具体的には、以下の継承されたポリシー文がオーバライドされます。

   呼び出し側に許可するロールは

   ```
   QA
   ```

   上記は [継承されたポリシー文] リスト ボックスに表示されます。
8. [適用] をクリックして変更を保存します。
9. 7-7 ページの「手順 8 : クライアント アプリケーションから EJB へのアクセスを試行する」の 8 から 10 を繰り返します。

   クライアント アプリケーションからの出力は、前述の場合と反対になります。つまり、`Stephanie` は `statelessSession` EJB へのアクセスを拒否され、`Jen` はアクセスを許可されます。

セキュリティ ポリシーで保護した EJB をクライアント アプリケーションが呼び出したため、このような結果になります。

手順 10 : create() および buy() EJB メソッドへのアクセスを制限する

1. Administration Console の左側のナビゲーション ツリーを使用して、`ejb20_basic_statelessSession.jar` を右クリックします。
2. メニューから [個別の Bean のポリシーとロールを定義] オプションを選択します。

   JAR ファイル内のすべての EJB (この場合は `statelessSession` EJB のみ) を示すテーブルが表示されます。

   注意 : このオプションの意味 : EJB レベル (セキュリティ ポリシーは EJB 内のすべてのメソッドに適用される)、または EJB 内の特定のメソッド レベルでセキュリティ ポリシーを作成できます。
3. `statelessSession` EJB の [ポリシーを定義] リンクをクリックします。
4. [Methods] ドロップダウン メニューを使用して、`create() - HOME` メソッドを選択します。
5. [ポリシー条件] リスト ボックスで、[呼び出し側をメンバとするグループは] を強調表示します。
6. [追加] をクリックして [グループ] ウィンドウを表示します。
7. [グループ名の入力] フィールドに `Testers` と入力します。
8. [追加] をクリックしてから [OK] をクリックします。

   [グループ] ウィンドウが閉じます。[ポリシー文] リスト ボックスに次のように表示されます。

   呼び出し側をメンバとするグループは

   ```
   Testers
   ```

注意 : create() メソッドに対するこのセキュリティ ポリシーを定義すると、「手順 9 : statelessSession EJB へのアクセスを制限する」で statelessSession EJB に対して定義したセキュリティ ポリシーがオーバライドされることに注意してください。具体的には、以下の継承されたポリシー文がオーバライドされます。

呼び出し側のユーザ名は

```
Jen
```

これは、[Methods] ドロップダウン メニューから ALL を選択すると [ポリシー文] リスト ボックスに表示されます。

9. [適用] をクリックして変更を保存します。
10. 同じ [ポリシー文] を使用して手順 4 から 9 を繰り返し、buy(java.lang.String, int) - REMOTE メソッドを保護します。
11. 7-7 ページの「手順 8 : クライアント アプリケーションから EJB へのアクセスを試行する」の 8 から 10 を繰り返します。

ユーザ Stephanie の場合と Jen の場合では、クライアント アプリケーションの出力が異なるメソッドで失敗します。sell() メソッドはクライアント アプリケーションで create() および buy() メソッドの後になるので、ユーザ Stephanie はこのメソッドでアクセスを拒否されます (サンプル出力については「コード リスト 7-2」を参照)。ユーザ Jen は create() メソッドでアクセスを拒否されます (サンプル出力については「コード リスト 7-3」を参照)。

コード リスト 7-2　ユーザ Stephanie の出力 : sell() メソッドでアクセス拒否

```
Buildfile: build.xml

run:
   [java] username: Stephanie
   [java] password: **********
   [java] URL: t3://localhost:7001
   [java] Creating a trader
   [java] Buying 100 shares of BEAS.
   [java] Buying 200 shares of MSFT.
   [java] Buying 300 shares of AMZN.
   [java] Buying 400 shares of HWP.
   [java] Selling 100 shares of BEAS.

   [java] java.rmi.AccessException: Security Violation: User:
'Stephanie' has insufficient permission to access EJB: type=<ejb>,
application=_appsdir_ejb20_basic_statelessSession_ear,
```

```
module=ejb20_basic_statelessSession.jar, ejb=statelessSession,
method=sell, methodInterface=Remote,
signature={java.lang.String,int}.
```

コード リスト 7-3　ユーザ Jen の出力 : create() メソッドでアクセス拒否

```
Buildfile: build.xml

run:
   [java] username: Jen
   [java] password: **********
   [java] URL: t3://localhost:7001
   [java] Creating a trader

   [java] java.rmi.AccessException: Security violation: User Jen
has insufficient permission to access method; nested exception is:
      [java] java.lang.SecurityException: Security violation: User
Jen has insufficient permission to access method
```

セキュリティ ポリシーで保護した EJB メソッドをクライアント アプリケーションが呼び出したため、このような結果になります。

8 例：basicauth Web アプリケーションのセキュリティ コンフィグレーションのコピーと再初期化

この例では、basicauth Web アプリケーションのセキュリティ コンフィグレーションをコンフィグレーション済みの認可プロバイダおよびロール マッピング プロバイダのデータベースにコピーして、以後 Administration Console からセキュリティ ロールとセキュリティ ポリシーを変更できるようにします。Administration Console を使用してセキュリティ ポリシーを変更したら、元のデプロイメント記述子を使用して basicauth Web アプリケーションのセキュリティ コンフィグレーションを再初期化します。したがって、この例は次の手順で構成されます。

- 8-2 ページの「手順 1 : basicauth Web アプリケーションのセキュリティ コンフィグレーションをコピーする」
- 8-7 ページの「手順 2 : Administration Console を使用したセキュリティ ポリシーの変更」
- 8-8 ページの「手順 3 : basicauth Web アプリケーションのセキュリティ コンフィグレーションを再初期化する」

注意： この例に進む前に、2-7 ページの「URL リソースおよび EJB リソースを保護する方法」、2-10 ページの「URL リソースおよび EJB リソースを保護するための前提条件」、および 2-18 ページの「組み合わせた方法による URL および EJB リソースの保護」に目を通しておいてください。

手順 1 : basicauth Web アプリケーションのセキュリティ コンフィグレーションをコピーする

basicauth Web アプリケーションのセキュリティ コンフィグレーションをコピーするには、次の手順に従います。

- 8-2 ページの「手順 1 : basicauth Web アプリケーションを入手する」
- 8-3 ページの「手順 2 : 事前設定を変更して Web アプリケーションをデプロイする」
- 8-4 ページの「手順 3 : コピーしたセキュリティ ポリシーを検証する (省略可能)」
- 8-6 ページの「手順 4 : コピーしたセキュリティ ロールを検証する (省略可能)」
- 8-7 ページの「手順 5 : [デプロイメント記述子内のセキュリティ データを無視] の設定を元に戻す」

手順 1 : basicauth Web アプリケーションを入手する

注意 : この節で説明する手順はすべて、Windows 環境での作業を想定しています。

1. 「Basic Authentication Sample Web Application」を入手します (dev2dev Web サイトの「Code Samples: Weblogic Server 」からダウンロード可)。
2. basicauth.zip ファイルを一時ディレクトリ (C:\basicauth など) に展開します。

手順 2 : 事前設定を変更して Web アプリケーションをデプロイする

注意 : この節で説明する手順はすべて、Windows 環境での作業を想定しています。

1. 2-11 ページの「fullyDelegateAuthorization フラグの変更方法」の指示に従って、`fullyDelegateAuthorization` フラグを `true` に設定します。

   注意 : この設定の意味 : すべての URL (Web) および EJB リソースに対して WebLogic Security サービスによるセキュリティ チェックを実行するよう WebLogic Server に指示します。詳細については、2-10 ページの「fullyDelegateAuthorization フラグについて」を参照してください。

   `fullyDelegateAuthorization` フラグが既に `true` に設定されている場合は、そのまま手順 2 に進みます。

2. Windows の [スタート] メニューから、[プログラム | BEA WebLogic Platform 7.0 | WebLogic Server 7.0 | Server Tour and Examples | Launch Examples Server] を選択して `examplesServer` というサーバを起動します。

   `examplesServer` が起動するとコンソールに `fullyDelegateAuthorization` フラグが表示され、ブラウザに [BEA WebLogic Server Out-of-the-Box Examples Index Page] が表示されます。

3. Administration Console の左側のナビゲーション ツリーを使用して、[セキュリティ | レルム] を展開します。

4. `myrealm` セキュリティ レルムを展開します。

5. [一般] タブで、[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスのチェックをはずします (このチェック ボックスのデフォルト設定なので、既にチェックがはずされている場合があります)。

   注意 : この設定の意味 : リソースをデプロイするたびに、URL (Web) および EJB リソースのセキュリティをデプロイメント記述子からコンフィグレーション済みの認可プロバイダとロール マッピング プロバイダのデータベースにコピーするよう WebLogic Server に指示します。詳細については、2-15 ページの「[デプロイメント記述子内のセキュリ

ティ データを無視] チェック ボックスについて」を参照してください。

6. [適用] をクリックして変更を保存します。
7. 手順 1 で fullyDelegateAuthorization フラグを true に設定する必要があった場合 (つまり、目的の値が設定されていなかった場合)、サーバを再起動します。詳細については、『管理者ガイド』の「WebLogic Server の起動と停止」を参照してください。

   手順 1 で fullyDelegateAuthorization フラグの値を変更していない場合は、サーバを再起動しないで手順 8 に進みます。
8. basicauth Web アプリケーションをデプロイして、examplesServer に割り当てます。

   注意： Web アプリケーションをデプロイする手順については、『WebLogic Server アプリケーションの開発』の「Administration Console を使用した J2EE アプリケーションのデプロイ」を参照してください。

手順 3：コピーしたセキュリティ ポリシーを検証する (省略可能)

1. basicauth Web アプリケーションの web.xml デプロイメント記述子を開き、<url-pattern> および <http-method> 要素の内容と、<auth-constraint> 要素の <role-name> 下位要素の内容を記録しておきます。コード リスト 8-1 では、web.xml デプロイメント記述子ファイルの該当する部分を太字で示しています。

コード リスト 8-1　basicauth Web アプリケーションの web.xml デプロイメント記述子

```
<!DOCTYPE web-app (View Source for full doctype...)>
<web-app>
   <welcome-file-list>
      <welcome-file>welcome.jsp</welcome-file>
   </welcome-file-list>
   <security-constraint>
      <web-resource-collection>
         <web-resource-name>Success</web-resource-name>
```

```
            <url-pattern>/welcome.jsp</url-pattern>
            <http-method>GET</http-method>
            <http-method>POST</http-method>
         </web-resource-collection>
         <auth-constraint>
           <role-name>developers</role-name>
         </auth-constraint>
      </security-constraint>
      <login-config>
         <auth-method>BASIC</auth-method>
         <realm-name>default</realm-name>
      </login-config>
      <security-role>
         <role-name>developers</role-name>
      </security-role>
</web-app>
```

2. Administration Console の左側のナビゲーション ツリーを使用して、[Web アプリケーション] を展開してから、basicauth を右クリックします。
3. メニューから [ポリシーを定義] オプションを選択します。
4. [一般] タブの [URL パターン] テキスト フィールドに /welcome.jsp と入力します。
5. [ポリシーを定義] ボタンをクリックして続行します。
6. ポリシー エディタ ページで、[Methods] ドロップダウン メニューを使用して、POST メソッドを選択します。

   [呼び出し側に許可するロールは] ポリシー条件が強調表示されます。[ポリシー文] リスト ボックスに次のように表示されます。

   呼び出し側に許可するロールは

   ```
   developers
   ```

7. [Methods] ドロップダウン メニューを使用して、GET メソッドを選択します。

   [呼び出し側に許可するロールは] ポリシー条件が強調表示されます。[ポリシー文] リスト ボックスに次のように表示されます。

   呼び出し側に許可するロールは

   ```
   developers
   ```

手順 4: コピーしたセキュリティ ロールを検証する (省略可能)

1. basicauth Web アプリケーションの weblogic.xml デプロイメント記述子を開き、<security-role-assignment> 要素の内容、特に <role-name> および <principal-name> 下位要素の内容を記録しておきます。コード リスト 8-2 では、weblogic.xml デプロイメント記述子ファイルの該当する部分を太字で示しています。

コード リスト 8-2　basicauth Web アプリケーションの weblogic.xml デプロイメント記述子

```
<!DOCTYPE weblogic-web-app (View Source for full doctype...)>
   <weblogic-web-app>
      <security-role-assignment>
         <role-name>developers</role-name>
         <principal-name>myGroup</principal-name>
      </security-role-assignment>
   </weblogic-web-app>
```

2. Administration Console の左側のナビゲーション ツリーを使用して、basicauth Web アプリケーションを右クリックします。
3. メニューから [ロールを定義] オプションを選択します。
4. [一般] タブの [URL パターン] テキスト フィールドに /* を入力します。
5. [ロールを定義] ボタンをクリックして続行します。

   [ロールの選択] ページに、developers というスコープ ロールが表示されます。
6. developers という名前のリンクをクリックします。
7. [条件] タブをクリックします。

   [ロール文] リスト ボックスに、デプロイメント記述子の対応する <principal-name> 要素 (ここでは myGroup というグループ) の内容に基づいたロール文が表示されます。

手順 5 : [デプロイメント記述子内のセキュリティ データを無視] の設定を元に戻す

警告 : この手順は必須です。この設定を元に戻さないと、URL (Web) リソースを再デプロイした場合に、セキュリティ コンフィグレーションの整合性が失われる可能性があります。

1. Administration Console の左側のナビゲーション ツリーを使用して、[セキュリティ | レルム] を展開します。
2. セキュリティ レルムの名前 (`myrealm` など) をクリックします。
3. [一般] タブで、[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスをクリックします (つまり、ボックスにチェック マークを入れます)。

   注意 : この設定の意味 : Administration Console を使用して、Web アプリケーションおよび EJB リソースのセキュリティを設定するように WebLogic Server に指示します。詳細については、2-15 ページの「[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスについて」を参照してください。
4. [適用] をクリックして変更を保存します。

手順 2 : Administration Console を使用したセキュリティ ポリシーの変更

1. Administration Console の左側のナビゲーション ツリーを使用して、`basicauth` を右クリックします。
2. メニューから [ポリシーを定義] オプションを選択します。
3. [一般] タブの [URL パターン] テキスト フィールドに `/welcome.jsp` と入力します。
4. [ポリシーを定義] ボタンをクリックして続行します。

5. ポリシー エディタ ページで、[Methods] ドロップダウン メニューを使用して、POST メソッドを選択します。
6. [ポリシー条件] リスト ボックスで、[アクセス可能な時間帯は] ポリシー条件を強調表示します。
7. [追加] をクリックします。
8. [時間制約] ウィンドウの [OK] をクリックして、デフォルトの開始時刻と終了時刻を選択します。

   [ポリシー文] リスト ボックスに次のように表示されます。

   呼び出し側に許可するロールは

   ```
   developers
   ```

   アクセス可能な時間帯は

   ```
   08:00:00 and 19:00:00
   ```

9. [適用] をクリックして変更を保存します。
10. [Methods] ドロップダウン メニューから POST メソッドを選択し、[ポリシー文] リスト ボックスに 2 つの式が存在することを確認します。

手順 3 : basicauth Web アプリケーションのセキュリティ コンフィグレーションを再初期化する

basicauth Web アプリケーションのセキュリティ コンフィグレーションを再初期化するには、次の手順に従います。

- 8-9 ページの「手順 1 : [デプロイメント記述子内のセキュリティ データを無視] の設定を変更する」
- 8-9 ページの「手順 2 : basicauth Web アプリケーションを再デプロイする」
- 8-10 ページの「手順 3 : セキュリティ コンフィグレーションが再初期化されたことを検証する (省略可能)」

- 8-11 ページの「手順 4 : [デプロイメント記述子内のセキュリティ データを無視] の設定を元に戻す」

手順 1 : [デプロイメント記述子内のセキュリティ データを無視] の設定を変更する

1. Administration Console の左側のナビゲーション ツリーを使用して、[セキュリティ | レルム] を展開します。
2. `myrealm` セキュリティ レルムを展開します。
3. [一般] タブで、[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスのチェックをはずします

   注意 : この設定の意味 : リソースをデプロイするたびに、URL (Web) および EJB リソースのセキュリティをデプロイメント記述子からコンフィグレーション済みの認可プロバイダとロール マッピング プロバイダのデータベースにコピーするよう WebLogic Server に指示します。詳細については、2-15 ページの「[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスについて」を参照してください。
4. [適用] をクリックして変更を保存します。

手順 2 : basicauth Web アプリケーションを再デプロイする

1. Administration Console の左側のナビゲーション ツリーを使用して、[デプロイメント | Web アプリケーション] を展開します。
2. `basicauth` Web アプリケーションをクリックします。
3. `basicauth` Web アプリケーションと同じ行にあるごみ箱アイコンをクリックします。

4. [はい] をクリックしてから [続行] リンクをクリックして、basicauth Web アプリケーションを削除します。

   削除した Web アプリケーションまたは EJB はテーブルに表示されなくなります。

5. examplesServer を対象として、basicauth Web アプリケーションを再デプロイします。

   注意 : Web アプリケーションおよび EJB をデプロイする手順については、『WebLogic Server アプリケーションの開発』の「デプロイメント ツールおよび手順」を参照してください。

手順 3 : セキュリティ コンフィグレーションが再初期化されたことを検証する (省略可能)

1. Administration Console の左側のナビゲーション ツリーを使用して、basicauth を右クリックします。
2. メニューから [ポリシーを定義] オプションを選択します。
3. [一般] タブの [URL パターン] テキスト フィールドに /welcome.jsp と入力します。
4. [ポリシーを定義] ボタンをクリックして続行します。
5. ポリシー エディタ ページで、[Methods] ドロップダウン メニューを使用して、POST メソッドを選択します。

   [ポリシー文] リスト ボックスに次のように表示されます。

   呼び出し側に許可するロールは

   ```
   developers
   ```

手順 4 : [デプロイメント記述子内のセキュリティ データを無視] の設定を元に戻す

警告 : この手順は必須です。この設定を元に戻さないと、URL (Web) リソースを再デプロイした場合に、セキュリティ コンフィグレーションの整合性が失われる可能性があります。

1. Administration Console の左側のナビゲーション ツリーを使用して、[セキュリティ | レルム] を展開します。
2. セキュリティ レルムの名前 (`myrealm` など) をクリックします。
3. [一般] タブで、[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスをクリックします (つまり、ボックスにチェック マークを入れます)。

   注意 : この設定の意味 : Administration Console を使用して、Web アプリケーションおよび EJB リソースのセキュリティを設定するように WebLogic Server に指示します。詳細については、2-15 ページの「[デプロイメント記述子内のセキュリティ データを無視] チェック ボックスについて」を参照してください。
4. [適用] をクリックして変更を保存します。

索引

W

あ

い

か

く

れ

ろ