Red Hat JBoss Enterprise Application Platform 6.4

ログインモジュールのリファレンス

ログインモジュールのリファレンス

Last Updated: 2022-06-16

Red Hat JBoss Enterprise Application Platform 6.4 ログインモジュールのリファレンス

ログインモジュールのリファレンス

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律上の通知

概要

本書の目的は、Red Hat JBoss Enterprise Application Platform 6 で利用可能なログインモジュールへの参照を提供することです。Red Hat JBoss Enterprise Application Platform 6 でのログインモジュールの動作の背景情報は、Red Hat JBoss Enterprise Application Platform6 のセキュリティーアーキテクチャーのドキュメントを参照してください。

目次

はじめに

このドキュメントの目的は、JBoss EAP 6 のセキュリティードメインで使用できるさまざまなログインモジュールへの完全なリファレンスを提供することです。

第1章 ログインモジュールの概要

ログインモジュールの基本とセキュリティードメイン内でのモジュールの使用については、Red Hat JBoss Enterprise Application Platform 6 セキュリティーアーキテクチャーガイドの セキュリティードメインのセクション で説明されています。

1.1. このドキュメントの組織について

本書では、ログインモジュールを以下の機能エリアにまとめています。

ログインモジュール機能組織

- 外部 ID ストアのないログインモジュール
  - Identity ログインモジュール: 固定またはハードコーディングされたユーザー名が必要な場合に使用されます。
  - UsersRoles ログインモジュール: ローカルの Java プロパティーファイルからユーザー名とロールをロードします。
  - PropertiesUsers ログインモジュール: ローカルの Java プロパティーファイルからユーザー名のみを読み込みます。
  - SimpleUsers ログインモジュール: ログインモジュール設定で直接ユーザー名とパスワードを定義します。
  - SecureIdentity ログインモジュール: レガシーでは、静的プリンシパルおよび暗号化されたパスワードをモジュール設定で直接定義できます。
  - ConfiguredIdentity ログインモジュール: 認証されたユーザーに静的プリンシパルを割り当てます。
  - Simple ログインモジュール: テスト用のクイックセキュリティー設定のモジュール。
  - Disabled ログインモジュール: 認証を常に失敗するモジュール。
  - Anon ログインモジュール : 認証されていないユーザーのアイデンティティーを指定するモジュール。
  - Runas ログインモジュール: 認証フェーズで静的ロールを追加するためのヘルパーモジュールです。
  - RoleMapping ログインモジュール: 認証済みユーザーのロールを追加またはロールに置き換えるためのヘルパーモジュールです。
  - RealmDirect ログインモジュール: セキュリティーレルムに認証を委任します。
  - RealmUsersRoles ログインモジュール: RealmDirect に置き換わるレガシーモジュールです。
- 外部 ID ストアのあるログインモジュール
  - Database ログインモジュール: データベースを使用してユーザーおよびロールマッピングを保存します。

    - DatabaseUsers ログインモジュール: 互換性を確保するためにデータベースにエイリアスを設定します。
    - LDAP ログインモジュール: LDAP サーバーを使用してユーザーおよびロールマッピングを保存します。
    - LdapExtended ログインモジュール
    - AdvancedLdap ログインモジュール: LDAP サーバーを使用して認証する際に追加機能を提供します。
    - AdvancedAdLdap ログインモジュール: Microsoft Active Directory で使用される追加機能を提供します。
    - LdapUsers ログインモジュール: LdapExtended および AdvancedLdap に置き換わるレガシーモジュールです。
    - Kerberos ログインモジュール: Kerberos 認証で使用されます。
    - SPNEGO ログインモジュール: Kerberos 認証で使用されます。
- Certificate-Based ベースのログインモジュール
    - Certificate ログインモジュール: X509 証明書を基にユーザーを認証します。
    - CertificateRoles ログインモジュール: ロールマッピングを使用した証明書モジュールの拡張
    - DatabaseCertificate ログインモジュール: データベースに保存されているロールマッピングで、証明書モジュールを拡張します。
- EJB およびリモーティングのログインモジュール
    - Remoting ログインモジュール: リモート EJB 呼び出しのセキュア化に使用されます。
    - Client ログインモジュール: ローカルの JVM 内の、クライアントアイデンティティーを確立するための EJB 呼び出しで使用されます。
- カスタムログインモジュール

本ガイドでは、承認モジュール、パスワードスタッキング、パスワードハッシュなどの関連トピックのリファレンス情報も提供します。

1.2. 拡張階層

本ガイドに記載されているログインモジュールの大半は、実際には他のログインモジュールの設定オプションと機能を拡張しています。ログインモジュールが機能の拡張に使用する構造は、階層を形成します。

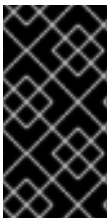
ログインモジュール拡張階層

- AbstractServerLoginModule
    - AbstractPasswordCredentialLoginModule
        - SecureIdentity ログインモジュール

        - ConfiguredIdentity ログインモジュール
    - Certificate ログインモジュール
        - CertificateRoles ログインモジュール
        - DatabaseCertificate ログインモジュール
    - CommonLoginModule
        - AdvancedLdap ログインモジュール
            - AdvancedAdLdap ログインモジュール
        - SPNEGO ログインモジュール
    - Identity ログインモジュール
    - RoleMapping ログインモジュール
    - Remoting ログインモジュール
    - UsernamePasswordLoginModule
        - Database ログインモジュール
        - LdapExtended ログインモジュール
        - Ldap ログインモジュール
        - LdapUsers ログインモジュール
        - Simple ログインモジュール
        - Anon ログインモジュール
        - RealmDirect ログインモジュール
        - UsersRoles ログインモジュール
            - RealmUsersRoles ログインモジュール
            - PropertiesUsers ログインモジュール
                - SimpleUsers ログインモジュール
- Client ログインモジュール
- DatabaseUsers ログインモジュール
- Disabled ログインモジュール
- Kerberos ログインモジュール
- RunAs ログインモジュール

階層のログインモジュールのほとんどは、JBoss EAP 6 でインスタンス化および使用される具体的な Java クラスですが、インスタンス化や使用を直接行うことができない抽象クラスがいくつかあります。これらの抽象クラスの目的は、共通の機能を提供し、他のログインモジュールが拡張するための

ベースクラスとして純粋に機能することにあります。



重要

デフォルトでは、ログインモジュールは、拡張されたログインモジュールからすべての動作とオプションを継承しますが、その動作は親ログインモジュールから上書きすることもできます。これにより、特定のオプションが親からログインモジュールによって継承され、未使用の状態になります。

第2章 抽象ログインモジュール

抽象ログインモジュールは、一般的な機能と設定オプションを提供するために他のログインモジュールによって拡張された抽象 Java クラスです。抽象ログインモジュールは直接使用することはできませんが、設定オプションを拡張するログインモジュールでも利用できます。

2.1. ABSTRACTSERVERLOGINMODULE

短縮名: AbstractServerLoginModule

フルネーム: org.jboss.security.auth.spi.AbstractServerLoginModule

AbstractServer ログインモジュールは、多くのログインモジュールのベースクラスおよびいくつかの抽象ログインモジュールとして機能します。JAAS サーバー側ログインモジュールに必要な一般的な機能を実装し、アイデンティティーとロールを保存する PicketBox 標準 Subject 使用パターンを実装します。

オプション	Type	デフォルト	説明
principalClass	完全修飾クラス名	org.jboss.security.SimplePrincipal	プリンシパル名の String 引数を取るコンストラクターが含まれる Principal 実装クラス。
module	String	none	カスタムコールバック/バリデーターの読み込みに使用できる jboss-module への参照。
unauthenticatedIdentity	String	none	これにより、認証情報を含まない要求に割り当てる必要があるプリンシパル名が定義されます。これを使用すると、保護されていないサーブレットは特定ロールを必要としない EJB でメソッドを呼び出すことができます。このようなプリンシパルには関連したロールがなく、セキュアでない EJB や、チェックされていないパーミッション制約と関連する EJB メソッドのみにアクセスできます。詳細は、Unauthenticated Identity セクションを参照してください。
password-stacking	useFirstPass または false	false	詳細は、パスワードスタッキングのセクションを参照してください。

2.1.1. 認証されていない ID

すべての要求が認証形式で受信される訳ではありません。**unauthenticatedIdentity** は、認証情報を持たないリクエストに特定の ID (例: **guest**) を割り当てるログインモジュール設定オプションです。これを使用すると、保護されていないサーブレットは特定ロールを必要としない EJB でメソッドを呼び出すことができます。このようなプリンシパルには関連したロールがなく、セキュアでない EJB や、チェックされていないパーミッション制約と関連する EJB メソッドのみにアクセスできます。たとえば、この設定オプションは UsersRoles および Remoting ログインモジュールで使用できます。

2.1.2. パスワードスタッキング

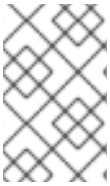
スタックでは複数のログインモジュールをチェーンでき、各ログインモジュールは認証中にクレデンシャルの検証とロールの割り当ての両方を提供します。これは多くのユースケースで機能しますが、クレデンシャルの検証とロールの割り当てが複数のユーザー管理ストアに分散されることがあります。

ユーザーは中央の LDAP サーバーで管理されますが、アプリケーション固有のロールはアプリケーションのリレーショナルデータベースに格納される場合を考えてみましょう。password-stacking モジュールオプションはこの関係をキャプチャーします。

パスワードスタッキングを使用するには、各ログインモジュールは、**<module-option>** セクションにある **password-stacking** 属性を **useFirstPass** に設定する必要があります。パスワードスタッキングに設定した以前のモジュールがユーザーを認証した場合、他のすべてのスタッキングモジュールがユーザーによって認証されたこととなり、承認の手順でロールの提供のみを行います。

password-stacking オプションを **useFirstPass** に設定すると、このモジュールは最初にプロパティー名 **javax.security.auth.login.name** で共有されたユーザー名を検索
し、**javax.security.auth.login.password** で共有されたパスワードを検索します。

これらのプロパティーが見つかった場合、プリンシパル名とパスワードとして使用されます。見つからなかった場合、プリンシパル名とパスワードはこのログインモジュールによって設定され、プリンシパル名は **javax.security.auth.login.password**、パスワードは **javax.security.auth.login.password** 以下に格納されます。



注記

パスワードスタッキングを使用する場合は、すべてのモジュールが必要になるように設定します。これにより、すべてのモジュールが考慮され、承認プロセスにロールを公開することができるようになります。

2.2. USERNAMEPASSWORDLOGINMODULE

短縮名: UsernamePasswordLoginModule

フルネーム: org.jboss.security.auth.spi.UsernamePasswordLoginModule

親: AbstractServerLoginModule

UsernamePassword ログインモジュールは、ログインプロセスで **identity == String username**, **credentials == String password** view を制限する抽象ログインモジュールです。以下のフィールドに加えて、AbstractServerLoginModule からすべてのフィールドを継承します。

オプション	Type	デフォルト	説明

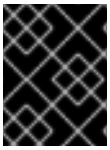
オプション	Type	デフォルト	説明
ignorePasswordCase	boolean	false	パスワードの比較で大文字と小文字を無視するかどうかを示すフラグ。
digestCallback	完全修飾クラス名	none	入力パスワードをハッシュするために salts などの事前/ ポストダイジェストコンテンツが含まれる org.jboss.crypto.digest.DigestCallback 実装のクラス名。hashAlgorithm が指定され、hashUserPassword が true に設定されている場合のみ使用されます。
storeDigestCallback	完全修飾クラス名	none	入力パスワードをハッシュするために salts などのストア/予測ダイジェストコンテンツが含まれる org.jboss.crypto.digest.DigestCallback 実装のクラス名。hashStorePassword が true で hashAlgorithm が指定されている場合にのみ使用されます。
throwValidateError	boolean	false	検証エラーをクライアントに公開すべきかどうかを示すフラグ。
inputValidator	完全修飾クラス名	none	クライアントが提供するユーザー名およびパスワードを検証するために使用される **org.jboss.security.auth.spi.InputValidator** 実装のインスタンス。

注記

パスワードハッシュに関する UsernamePassword ログインモジュールオプションについては、次のセクションで説明します。

2.2.1. パスワードのハッシュ化

ログインモジュールのほとんどは、クライアントが提供するパスワードをユーザー管理システムに保存されたパスワードと比較する必要があります。通常、これらのモジュールはプレーンテキストのパスワードを使用しますが、プレーンテキストのパスワードがサーバー側に保存されないようにするため、ハッシュ化されたパスワードをサポートするよう設定できます。JBoss EAP 6 は、ユーザーパスワードおよびストアパスワードがハッシュ化された場合だけでなく、ハッシュアルゴリズム、エンコーディング、および文字セットを設定する機能をサポートします。



重要

Red Hat JBoss Enterprise Application Platform Common Criteria 認定リリースは、パスワードハッシュに SHA-256 のみをサポートします。

以下は、UsernamePassword ログインモジュールが親となるログインモジュールの一部として設定できるパスワードハッシュオプションです。

オプション	Type	デフォルト	説明
hashAlgorithm	パスワードハッシュアルゴリズムを表す文字列。	none	パスワードをハッシュするために使用される java.security.MessageDigest アルゴリズムの名前。デフォルトがないため、ハッシュを有効にするには、このオプションを指定する必要があります。一般的な値は SHA-256、SHA-1、および MD5 です。HashAlgorithm が指定され、**hashUserPassword** が **true** に設定されている場合、CallbackHandler から取得したクリアテキストパスワードは、UsernamePasswordLoginModule.validatePassword に inputPassword 引数として渡される前にハッシュされます。
hashEncoding	String	base64	hashAlgorithm も設定されている場合はハッシュ化されたパスワードの文字列形式。base64、hex、または rfc2617 のいずれかのエンコーディングタイプを指定できます。

オプション	Type	デフォルト	説明
hashCharset	String	コンテナーのランタイム環境に設定されるデフォルトのエンコーディング	パスワード文字列をバイト配列に変換する際に使用する charset/エンコーディングの名前。
hashUserPassword	boolean	true	ユーザーが入力したパスワードをハッシュ化するかどうかを示すフラグ。ハッシュ化されたユーザーパスワードは、ログインモジュール内の値と比較されます. これは、パスワードのハッシュです。
hashStorePassword	boolean	false	返されたストアパスワードをハッシュ化するかどうかを示すフラグ。これは、ユーザーパスワードのハッシュと、比較対象のサーバーからの要求固有のトークンを送信するダイジェスト認証に使用されます。ハッシュアルゴリズム (ダイジェストの場合は rfc2617) を利用してサーバー側のハッシュを計算し、クライアントから送信されたハッシュ値と一致させる必要があります。
passwordIsA1Hash	boolean		digestCallback または storeDigestCallback として設定される場合に**org.jboss.security.auth.callback.RFC2617Digest** が使用するフラグ。True の場合、着信パスワードはハッシュ化されているため、ハッシュ化されません。

2.3. ABSTRACTPASSWORDCREDENTIALLOGINMODULE

短縮名: AbstractPasswordCredentialLoginModule

フルネーム : org.picketbox.datasource.security.AbstractPasswordCredentialLoginModule

親: AbstractServerLoginModule

AbstractPasswordCredentialLoginModule は、PasswordCredentials を処理するベースログインモジュールです。

2.4. COMMONLOGINMODULE

短縮名: CommonLoginModule

フルネーム: org.jboss.security.negotiation.common.CommonLoginModule

親: AbstractServerLoginModule

CommonLoginModule は、JBoss Negotiation 内の一部のログインモジュールのベースログインモジュールとして機能する抽象ログインモジュールです。

第3章 外部 ID ストアのないログインモジュール

3.1. IDENTITY ログインモジュール

短縮名: Identity

フルネーム: org.jboss.security.auth.spi.IdentityLoginModule

親: AbstractServerLoginModule

Identity ログインモジュールは、ハードコードされたユーザー名をモジュールに対して認証されたサブジェクトに関連付ける簡単なログインモジュールです。このモジュールは、プリンシパルのオプションによって指定された名前を使用して SimplePrincipal インスタンスを作成します。このログインモジュールは、固定のアイデンティティーをサービスに提供する必要がある場合に便利です。また、指定のプリンシパルに関連するセキュリティーや関連するロールをテストするために、開発環境でも使用できます。

表3.1 IdentityLoginModule オプション

オプション	Type	デフォルト	説明
principal	String	guest	プリンシパルに使用する名前。
roles	文字列のカンマ区切りリスト	none	サブジェクトに割り当てられるロールのカンマ区切りの一覧。

3.2. USERSROLES ログインモジュール

短縮名: UsersRoles

フルネーム: org.jboss.security.auth.spi.UsersRolesLoginModule

親: UsernamePasswordLoginModule

UsersRoles ログインモジュールは、Java プロパティーファイルからロードされる複数のユーザーおよびユーザーロールをサポートする簡単なログインモジュールです。このログインモジュールの主な目的は、アプリケーションとともにデプロイされたプロパティーファイルを使用して複数のユーザーおよびロールのセキュリティー設定を簡単にテストすることです。

表3.2 UsersRoles ログインモジュールオプション

オプション	Type	デフォルト	説明
usersProperties	ファイルまたはリソースへのパス。	users.properties	ユーザー/パスワード間のマッピングが含まれるファイルまたはリソースです。ファイルの形式は **username**=password です。

オプション	Type	デフォルト	説明
rolesProperties	ファイルまたはリソースへのパス。	roles.properties	ユーザー/ ロール間のマッピングが含まれるファイルまたはリソースです。ファイルの形式は **username**=role1,role2,role3 です。
defaultUsersProperties	String	**defaultUsers.properties**	UserProperties プロパティーに渡されるデフォルトのプロパティーとして使用される **username-to-password** マッピングが含まれる properties リソースの名前。
defaultRolesProperties	String	**defaultRoles.properties**	**UserProperties** プロパティーに渡されるデフォルトのプロパティーとして使用される username-to-roles マッピングが含まれる properties リソースの名前。
roleGroupSeperator	String	.	ユーザー名とロールのグループ名を分離するのに使用する文字 (例 **jduke.CallerPrincipal=...** の .)。

3.3. PROPERTIESUSERS ログインモジュール

短縮名: PropertiesUsers

フルネーム : org.jboss.security.auth.spi.PropertiesUsersLoginModule

親: UsersRoles Login Module

プロパティーファイルを使用して認証用のユーザー名とパスワードを保存する PropertiesUsers ログインモジュール。承認 (ロールマッピング) は提供されません。このモジュールは、テストにのみ適しています。

3.4. SIMPLEUSERS ログインモジュール

短縮名: SimpleUsers

フルネーム: org.jboss.security.auth.spi.SimpleUsersLoginModule

親 : PropertiesUsers ログインモジュール

module-option を使用してユーザー名とパスワードを保存する SimpleUsers ログインモジュール。**module-option** の **name** 属性と **value** 属性は、ユーザー名とパスワードを指定します。これはテスト用にのみ含まれており、実稼働環境には適していません。

3.5. SECUREIDENTITY ログインモジュール

短縮名: SecureIdentity

フルネーム: org.picketbox.datasource.security.SecureIdentityLoginModule

親: AbstractPasswordCredentialLoginModule

SecurityIdentity ログインモジュールは、レガシー目的で提供されるモジュールです。これにより、ユーザーはパスワードを暗号化し、静的プリンシパルで暗号化されたパスワードを使用できます。アプリケーションが SecureIdentity を使用する場合は、パスワード vault メカニズムの使用を検討してください。

表3.3 SecureIdentity ログインモジュールオプション

オプション	Type	デフォルト	説明
username	String	none	認証のユーザー名
password	暗号化された文字列	""	認証に使用するパスワード。パスワードを暗号化するには、コマンドラインでモジュールを直接使用します (例: **java org.picketbox.datasource.security.SecureIdentityLoginModule password_to_encrypt**)。このコマンドの結果をモジュールオプションの値フィールドに貼り付けます。デフォルト値は空の String です。
managedConnectionFactoryName	JCA リソース	none	データソースの JCA 接続ファクトリーの名前。

3.6. CONFIGUREDIDENTITY ログインモジュール

短縮名: ConfiguredIdentity

フルネーム: org.picketbox.datasource.security.ConfiguredIdentityLoginModule

親: AbstractPasswordCredentialLoginModule

ConfiguredIdentity ログインモジュールは、モジュールオプションに指定されたプリンシパルとモジュールに対して認証されたサブジェクトを関連付けます。使用される Principal クラスのタイプは **org.jboss.security.SimplePrincipal** です。

表3.4 ConfiguredIdentity ログインモジュールオプション

オプション	Type	デフォルト	説明
username	String	none	認証のユーザー名
password	暗号化された文字列	""	認証に使用するパスワード。vault メカニズムを介して暗号化できます。デフォルト値は空の String です。
principal	プリンシパルの名前	none	モジュールに対して認証されたサブジェクトに関連付けられるプリンシパル。

3.7. SIMPLE ログインモジュール

短縮名: Simple

フルネーム: org.jboss.security.auth.spi.SimpleServerLoginModule

親: UsernamePasswordLoginModule

Simple ログインモジュールは、テスト目的でセキュリティーをすばやくセットアップするためのモジュールです。以下の単純なアルゴリズムを実装します。

- パスワードが null の場合、ユーザーを認証して **guest** のアイデンティティーと **guest** のロールを割り当てます。
- それ以外の場合は、パスワードがユーザーと同じ場合は、**username** と **user** および **guest** ロールの両方に同一のアイデンティティーを割り当てます。
- そうしないと、認証に失敗します。

Simple ログインモジュールにはオプションがありません。

3.8. DISABLED ログインモジュール

短縮名: Disabled

フルネーム : org.jboss.security.auth.spi.DisabledLoginModule

常に認証が失敗するログインモジュール。JAAS が **other** セキュリティードメインを使用するようにフォールバックしない場合など、無効にする必要のあるセキュリティードメインに使用されます。

表3.5 無効化されたログインモジュールオプション

オプション	Type	デフォルト	説明
jboss.security.security_domain	String		エラーメッセージに表示されるセキュリティードメインの名前。

3.9. ANON ログインモジュール

短縮名: Anon

フルネーム: org.jboss.security.auth.spi.AnonLoginModule

親: UsernamePasswordLoginModule

unauthenticatedIdentity プロパティーを介して認証されていないユーザーのアイデンティティーの指定を可能にする簡単なログインモジュール。このログインモジュールには、UsernamePasswordLoginModule の継承オプション以外のオプションはありません。

3.10. RUNAS ログインモジュール

短縮名: RunAs

フルネーム: org.jboss.security.auth.spi.RunAsLoginModule

RunAS ログインモジュールは、認証のログインフェーズの間に **run as** ロールをスタックにプッシュし、ログインフェーズ後にコミットまたは中断フェーズで **run as** ロールをスタックからポップするヘルパーモジュールです。このログインモジュールの目的は、セキュアな EJB にアクセスするログインモジュールなど、セキュアなリソースにアクセスして認証を実行する必要のあるその他のログインモジュールにロールを提供することです。RunAs ログインモジュールは、**run as** ロールの構築が必要なログインモジュールよりも先に設定する必要があります。

表3.6 RunAs ログインモジュールオプション

オプション	Type	デフォルト	説明
roleName	ロール名	nobody	ログインフェーズで、**run as** として使われるロールの名前 。
principalName	プリンシパル名	nobody	ログインフェーズで、**run as** プリンシパルとして使用するプリンシパルの名前。指定しないと、nobody のデフォルト値が使用されます。
principalClass	完全修飾クラス名	org.jboss.security.SimplePrincipal	プリンシパル名の String 引数を取るコンストラクターが含まれる Principal 実装クラス。

3.11. ROLEMAPPING ログインモジュール

短縮名: RoleMapping

フルネーム: org.jboss.security.auth.spi.RoleMappingLoginModule

親: AbstractServerLoginModule

RoleMapping ログインモジュールは、1 つ以上の宣言的ロールへの認証プロセスの最終結果となるロールのマッピングをサポートするログインモジュールです。たとえば、ユーザー **John** のロールが **ldapAdmin** と **testAdmin** で、**web.xml** または **ejb-jar.xml** ファイルで定義されたアクセスの宣言的ロールは **admin** であると認証プロセスによって判断された場合、このログインモジュールは管理者ロールを **John** にマップします。RoleMapping ログインモジュールは、以前マップされたロールのマッピングを変更するため、ログインモジュール設定でオプションのモジュールとして定義する必要があります。

表3.7 RoleMapping ログインモジュールオプション

オプション	Type	デフォルト	説明
rolesProperties	プロパティーファイルまたはリソースの完全修飾ファイルパスまたは完全修飾ファイル名。	none	ロールを置き換えるロールにマップするプロパティーファイルまたはリソースの完全修飾ファイルパスおよび名前。形式は **original_role=role1,role2,role3** になります。
replaceRole	true または false	false	現在のロールに追加するか、現在のロールをマップされたロールに置き換えるか。True に設定された場合を置き換えます。

3.12. REALMDIRECT ログインモジュール

短縮名: RealmDirect

フルネーム: org.jboss.as.security.RealmDirectLoginModule

親: UsernamePasswordLoginModule

Realm Direct ログインモジュールは、認証および承認の決定に既存のセキュリティーレルムを使用できるようにします。このモジュールを設定すると、認証の決定に参照されるレルムを使用してアイデンティティー情報を検索し、承認の決定のためにそのセキュリティーレルムに委譲します。たとえば、JBoss EAP 6 に同梱される事前設定された **other** セキュリティードメインには RealmDirect ログインモジュールがあります。このモジュールに参照されるレルムがない場合、デフォルトで **ApplicationRealm** セキュリティーレルムが使用されます。

表3.8 RealmDirect ログインモジュールオプション

オプション	Type	デフォルト	説明
realm	String	ApplicationRealm	必要なレルムの名前。

3.13. REALMUSERSROLES ログインモジュール

短縮名: RealmUsersRoles

フルネーム: org.jboss.as.security.RealmUsersRolesLoginModule

親: UsersRoles Login Module

所定のレルムからユーザーを認証できるログインモジュール。リモーティング呼び出しに使用されます。**RealmUsersRoles** の代わりに RealmDirect を使用することが推奨されます。

表3.9 RealmUsersRoles ログインモジュールオプション

オプション	Type	デフォルト	説明
realm	String	ApplicationRealm	必要なレルムの名前。
hashAlgorithm	String	REALM	継承された UsernamePassword ログインモジュールからのオプションに対して、UsernamePassword LoginModule によって設定される静的な値。
hashStorePassword	String	false	継承された UsernamePassword ログインモジュールからのオプションに対して、UsernamePassword LoginModule によって設定される静的な値。

第4章 外部 ID ストアのあるログインモジュール

4.1. DATABASE ログインモジュール

短縮名: データベース

フルネーム: org.jboss.security.auth.spi.DatabaseServerLoginModule

親: UsernamePasswordLoginModule

Database ログインモジュールは、認証とロールマッピングをサポートする JDBC (Java Database Connectivity) ベースのログインモジュールです。このログインモジュールは、ユーザー名、パスワード、およびロール情報がリレーショナルデータベースに格納される場合に使用されます。このログインモジュールは、想定される形式のプリンシパルおよびロールが含まれる論理テーブルへの参照を提供して動作します。

表4.1 Database ログインモジュールオプション

オプション	Type	デフォルト	説明
dsJndiName	JNDI リソース	java:/DefaultDS	認証情報を格納している JNDI リソースの名前。
principalsQuery	準備済み SQL ステートメント	select **Password** from **Principals** where **PrincipalID**=?	プリンシパルに関する情報を取得するための準備済み SQL クエリー。
rolesQuery	準備済み SQL ステートメント	none	ロールに関する情報を取得するための準備済み SQL クエリー。これは、「'select **Role**, **RoleGroup** from **Roles** where **PrincipalID**=?'」のクエリーと同等です。ここでは、**Role** はロール名で、**RoleGroup** 列値は常に大文字の **R** または **CallerPrincipal** を持つ **Roles** のいずれかにしてください。
suspendResume	boolean	true	データベースの操作中に既存の JTA トランザクションを一時停止するかどうか。
transactionManagerJndiName	JNDI リソース	java:/TransactionManager	ログインモジュールによって使用されるトランザクションマネージャーの JNDI 名。

4.2. DATABASEUSERS ログインモジュール

短縮名: DatabaseUsers

フルネーム: org.jboss.security.DatabaseUsers

互換性の理由から Database ログインモジュールのエイリアス。

4.3. LDAP ログインモジュール

短縮名: Ldap

フルネーム: org.jboss.security.auth.spi.LdapLoginModule

親: UsernamePasswordLoginModule

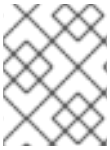
Ldap ログインモジュールは、LDAP サーバーに対して認証を行うログインモジュール実装です。security サブシステムは接続情報 java.naming.security.principal を使用して、LDAP サーバーに接続します。この bindDN は、JNDI 初期コンテキストを使用した場合にユーザーおよびロールの baseCtxDN および rolesCtxDN ツリーを検索する権限があります。ユーザーが認証を試みると、LDAP ログインモジュールは LDAP サーバーへ接続し、ユーザーのクレデンシャルを LDAP サーバーに渡します。認証に成功すると、JBoss EAP 内のそのユーザーに InitialLDAPContext が作成され、ユーザーのロールが入力されます。

表4.2 LDAP ログインモジュールオプション

オプション	Type	デフォルト	説明
principalDNPrefix	文字列		ユーザー DN を形成するためにユーザー名に追加される接頭辞。ユーザーにユーザー名を要求し、principalDNPrefix および principalDNSuffix を使用して完全修飾 DN をビルドできます。
principalDNSuffix	文字列		ユーザー DN を形成するためにユーザー名に追加される接尾辞。ユーザーにユーザー名を要求し、principalDNPrefix および principalDNSuffix を使用して完全修飾 DN をビルドできます。
rolesCtxDN	完全修飾 DN	none	ユーザーロールを検索するコンテキストの完全修飾 DN。

オプション	Type	デフォルト	説明
userRolesCtxDNAttributeName	attribute	none	ユーザーロールを検索するコンテキストの DN を含むユーザーオブジェクトの属性です。これは、ユーザーのロールを検索するコンテキストがユーザーごとに一意である可能性がある点で rolesCtxDN とは異なります。
roleAttributeID	attribute	roles	ユーザーロールを含む属性の名前。
roleAttributeIsDN	true または false	false	**roleAttributeID** にロールオブジェクトの完全修飾 DN が含まれるかどうか。false の場合は、コンテキスト名の roleNameAttributeId 属性の値からこのロール名が取得されます。Microsoft Active Directory などの特定のディレクトリースキーマでは、この属性を true に設定する必要があります。
roleNameAttributeID	attribute	name	ロール名を含む roleCtxDN コンテキスト内の属性の名前。roleAttributeIsDN プロパティーを true に設定すると、このプロパティーはロールオブジェクトの名前属性の検索に使用されます。
uidAttributeID	attribute	uid	ユーザー ID に対応する UserRolesAttributeDN の属性名。これは、ユーザーロールの特定に使用されます。

オプション	Type	デフォルト	説明
matchOnUserDN	true または false	false	ユーザーの完全識別名またはユーザー名のみで、ユーザーロールの検索と一致するかどうか。True の場合、完全なユーザー DN が一致値として使用されます。False の場合、ユーザー名のみが uidAttributeName 属性に対する一致値として使用されます。
allowEmptyPasswords	true または false	false	空のパスワードを許可するかどうか。ほとんどの LDAP サーバーは、空のパスワードを匿名ログイン試行として処理します。空のパスワードを拒否するには、これを false に設定します。
searchTimeLimit	integer	10000 (10 秒)	ユーザーまたはロールの検索のタイムアウト (ミリ秒単位)。
searchScope	OBJECT_SCOPE, ONELEVEL_SCOPE, SUBTREE_SCOPE のいずれか	SUBTREE_SCOPE	使用する検索範囲。
jaasSecurityDomain	String	none	**java.naming.security.credentials** を復号化するために使用される JaasSecurityDomain の JMX ObjectName。暗号化されたパスワードの形式は、このオプションで渡されたオブジェクトで呼び出される **encrypt64(byte[])** メソッドによって返されます。



注記

LDAP サーバーへの接続と初期コンテキストの作成に関連する追加の LDAP コンテキストプロパティーは、ここで 詳しく説明します。

注記

このログインモジュールは親 UsernamePasswordLoginModule から **ignorePasswordCase** オプションを継承しますが、特定のログインモジュールでは使用されません。

4.4. LDAPEXTENDED ログインモジュール

短縮名: LdapExtended

フルネーム: org.jboss.security.auth.spi.LdapExtLoginModule

親: UsernamePasswordLoginModule

LdapExtended ログインモジュールは、ユーザーと認証で関連ロールを検索します。ロールは再帰的にクエリーを行い、DN に従って階層的なロール構造を移動します。LoginModule オプションには、JNDI プロバイダーがサポートする指定の LDAP によってオプションがサポートされるかどうかが含まれます。

認証は 2 つの手順で行われます。

1. LDAP サーバーへの最初のバインドは、bindDN オプションおよび bindCredential オプションを使用して行われます。**bindDN** は LDAP ユーザーであり、ユーザーとロールの **baseCtxDN** および **rolesCtxDN** ツリーの両方を検索する機能があります。認証するユーザー DN は、**baseFilter** 属性で指定されたフィルターを使用してクエリーされます。

2. 生成されるユーザー DN は、ユーザー DN をプリンシパル名として使用し、コールバックハンドラーが取得したパスワードをプリンシパルの認証情報として使用して LDAP サーバーにバインドすることで認証されます。

表4.3 LdapExtended ログインモジュール

オプション	Type	デフォルト	説明
baseCtxDN	完全修飾 DN	none	ユーザーの検索を開始するため、トップレベルのコンテキストの固定 DN です。
bindCredential	文字列 (オプションで暗号化)	none	DN の認証情報を保存するために使用されます。
bindDN	完全修飾 DN	none	ユーザーおよびロールクエリーの LDAP サーバーに対してバインドするために使用される DN です。この DN には、**baseCtxDN** および **rolesCtxDN** 値の読み取りおよび検索パーミッションが必要です。

オプション	Type	デフォルト	説明
baseFilter	LDAP フィルター文字列。	none	認証するユーザーのコンテキストを見つけるために使用される検索フィルター。**{0}** 式を使用しているフィルターに、入力**ユーザー名**、またはログインモジュールコールバックから取得した **userDN** が置換されます。検索フィルターの一般的な例は **(uid={0})** です。
jaasSecurityDomain	String	none	パスワードの復号に使用する JaasSecurityDomain の JMX ObjectName。
rolesCtxDN	完全修飾 DN	none	ユーザーロールを検索するためのコンテキストの固定 DN です。これは、実際のロールがである DN ではなく、ユーザーロールを含むオブジェクトがある DN です。たとえば、Microsoft Active Directory サーバーでは、ユーザーアカウントが DN になります。
roleFilter	LDAP フィルター文字列。	none	認証されたユーザーに関連付けられたロールを見つけるために使用される検索フィルター。**{0}** 式を使用しているフィルターに、入力**ユーザー名**、またはログインモジュールコールバックから取得した **userDN** が置換されます。認証済み **userDN** は **{1}** が使用されたフィルターに置き換えられます。入力ユーザー名の検索フィルター例は **(member={0})** です。認証済み userDN を照合する他の例は **(member={1})** です。
roleAttributeID	attribute	role	ユーザーロールを含む属性の名前。

オプション	Type	デフォルト	説明
roleAttributeIsDN	true または false	false	**roleAttributeID** にロールオブジェクトの完全修飾 DN が含まれるかどうか。false の場合は、コンテキスト名の **roleNameAttributeId** 属性の値からこのロール名が取得されます。Microsoft Active Directory などの特定のディレクトリースキーマでは、この属性を **true** に設定する必要があります。
defaultRole	ロール名	none	すべての認証ユーザーに含まれるロール
parseRoleNameFromDN	true または false	false	クエリーによって返された DN に **roleNameAttributeID** が含まれるかどうかを示すフラグ。**true** に設定した場合には、DN は **roleNameATtributeID** に対してチェックされます。**false** に設定すると、DN は **roleNameATtributeID** に対して確認されません。このフラグにより、LDAP クエリーのパフォーマンスを向上できます。
parseUsername	true または false	false	DN が **username** 名用に解析されるかどうかを示すフラグ。**true** に設定した場合には、DN はユーザー名に対して解析されます。**false** に設定した場合には、DN はユーザー名に対して解析されません。このオプションは、**usernameBeginString** および **usernameEndString** とともに使用されます。

オプション	Type	デフォルト	説明
usernameBeginString	String	none	DN の最初から削除される文字列を定義して、ユーザー名を表示します。このオプションは **usernameEndString** と合わせて使用し、**parseUsername** が **true** に設定されている場合にのみ考慮されます。
usernameEndString	String	none	DN の最後から削除され、**username** を表示する文字列を定義します。このオプションは **usernameBeginString** とともに使用され、**parseUsername** が **true** に設定されている場合のみ考慮されます。
roleNameAttributeID	attribute	name	ロール名を含む **roleCtxDN** コンテキスト内の属性の名前。**roleAttributeIsDN** プロパティーを true に設定すると、このプロパティーはロールオブジェクトの名前属性の検索に使用されます。
distinguishedNameAttribute	attribute	distinguishedName	ユーザーの DN を含むユーザーエントリーの属性の名前。これは、ユーザー自体の DN に特殊文字 (たとえば、正しいユーザーマッピングを防ぐバックスラッシュ) が含まれている場合に必要になることがあります。属性が存在しない場合は、エントリーの DN が使用されます。
roleRecursion	Integer	0	ロール検索の再帰レベルの数は、一致するコンテキストの下に続きます。これを 0 に設定して再帰を無効にします。

オプション	Type	デフォルト	説明
searchTimeLimit	integer	10000 (10 秒)	ユーザーまたはロールの検索のタイムアウト (ミリ秒単位)。
searchScope	OBJECT_SCOPE, ONELEVEL_SCOPE, SUBTREE_SCOPE のいずれか	SUBTREE_SCOPE	使用する検索範囲。
allowEmptyPasswords	true または false	false	空のパスワードを許可するかどうか。ほとんどの LDAP サーバーは、空のパスワードを匿名ログイン試行として処理します。空のパスワードを拒否するには、これを **false** に設定します。
referralUserAttributeIDToCheck	attribute	none	紹介を使用しない場合、このオプションは無視することができます。リファーラルを使用し、ロールオブジェクトがリファーラル内部にあると、このオプションは特定のロール (例: **member**) に対して定義されたユーザーが含まれる属性名を示します。ユーザーはこの属性名の内容に対して確認されます。このオプションが設定されていないとチェックは常に失敗するため、ロールオブジェクトはリファーラルツリーに保存できません。

注記

LDAP サーバーへの接続と初期コンテキストの作成に関連する追加の LDAP コンテキストプロパティーは、ここで 詳しく説明します。

注記

このログインモジュールは親 UsernamePasswordLoginModule から **ignorePasswordCase** オプションを継承しますが、特定のログインモジュールでは使用されません。

4.5. ADVANCEDLDAP ログインモジュール

短縮名: AdvancedLdap

フルネーム: org.jboss.security.negotiation.AdvancedLdapLoginModule

親: CommonLoginModule

AdvancedLdap ログインモジュールは、SASL や JAAS セキュリティードメインの使用などの追加機能を提供するモジュールです。ユーザーが LDAP を SPNEGO 認証で使用する場合や、LDAP サーバーを使用中に認証フェーズの一部を省略したい場合は、SPNEGO ログインモジュールとチェーンされた AdvancedLdap ログインモジュールの使用を検討してください。または AdvancedLdap ログインモジュールのみの使用を検討してください。

AdvancedLdap ログインモジュールは、以下の点で LdapExtended ログインモジュールとは異なります。

- トップレベルのロールは **roleAttributeID** のみに対してクエリーされ、**roleNameAttributeID** にはクエリーされません。
- **roleAttributeIsDN** モジュールプロパティーが **false** に設定されている場合、**recurseRoles** モジュールオプションが **true** に設定されていても、再帰ロール検索は無効になります。

表4.4 AdvancedLdap ログインモジュールオプション

オプション	Type	デフォルト	説明
bindDN	完全修飾 DN	none	ユーザーおよびロールクエリーの LDAP サーバーに対してバインドするために使用される DN です。この DN には、baseCtxDN および rolesCtxDN 値の読み取りおよび検索パーミッションが必要です。
bindCredential	文字列 (オプションで暗号化)	none	DN の認証情報を保存するために使用されます。
jaasSecurityDomain	String	none	パスワードの復号に使用する JaasSecurityDomain の JMX ObjectName。
java.naming.provider.url	String	**java.naming.security.protocol** の値が **SSL** の場合、ldap://localhost:686, otherwise ldap://localhost:389	ディレクトリーサーバーの URI。
baseCtxDN	完全修飾 DN	none	検索のベースとして使用する識別名。

オプション	Type	デフォルト	説明
baseFilter	LDAP 検索フィルターを表す文字列。	none	検索結果を絞り込むために使用するフィルター。
searchTimeLimit	integer	10000 (10 秒)	ユーザーまたはロールの検索のタイムアウト (ミリ秒単位)。
roleAttributeID	LDAP 属性を表す文字列値。	none	承認ロールの名前が含まれる LDAP 属性です。
roleAttributeIsDN	true または false	false	ロール属性が識別名 (DN) であるかどうか。
rolesCtxDN	完全修飾 DN	none	ユーザーロールを検索するコンテキストの完全修飾 DN。
roleFilter	LDAP フィルター文字列。	none	認証済みユーザーと関連付けられたロールを検索するために使用される検索フィルター。{0} 式を使用しているフィルターに、入力ユーザー名、またはログインモジュールコールバックから取得した userDN が置換されます。認証済み userDN は {1} が使用されたフィルターに置き換えられます。入力ユーザー名に一致する検索フィルター例は (**member**={0}) です。認証済み userDN に一致する他の例は (**member**={1}) です。
recurseRoles	true または false	false	ロールの **roleAttributeID** を再帰的に検索するかどうか。
roleNameAttributeID	LDAP 属性を表す文字列。	none	実際の role 属性が含まれる **roleAttributeID** 内に含まれる属性です。

オプション	Type	デフォルト	説明
referralUserAttributeIDToCheck	attribute	none	紹介を使用しない場合、このオプションは無視することができます。リファーラルを使用し、ロールオブジェクトがリファーラル内部にあると、このオプションは特定のロール (例: **member**) に対して定義されたユーザーが含まれる属性名を示します。ユーザーはこの属性名の内容に対して確認されます。このオプションが設定されていない場合、チェックは常に失敗するため、ロールオブジェクトは参照ツリーに格納できません。
searchScope	OBJECT_SCOPE, ONELEVEL_SCOPE, SUBTREE_SCOPE のいずれか	SUBTREE_SCOPE	使用する検索範囲。
allowEmptyPassword	true または false	false	空のパスワードを許可するかどうか。ほとんどの LDAP サーバーは、空のパスワードを匿名ログイン試行として処理します。空のパスワードを拒否するには、これを false に設定します。
bindAuthentication	文字列	システムプロパティー **java.naming.security.authentication** が設定されている場合、この値はその値を使用し、それ以外の場合は、デフォルトで **simple** に設定されます。	ディレクトリーサーバーへのバインドに使用する SASL 認証のタイプ。

注記

LDAP サーバーへの接続と初期コンテキストの作成に関連する追加の LDAP コンテキストプロパティーは、ここで 詳しく説明します。

4.6. ADVANCEDADLDAP ログインモジュール

短縮名: AdvancedAdLdap

フルネーム: org.jboss.security.negotiation.AdvancedADLoginModule

親 : AdvancedLdap ログインモジュール

AdvancedAdLdap ログインモジュールは、Microsoft Active Directory に関連するパラメーターを追加しますが、AdvancedLdap ログインモジュール で使用できるもの以外に設定可能なオプションはありません。

注記

LDAP サーバーへの接続と初期コンテキストの作成に関連する追加の LDAP コンテキストプロパティーは、ここで 詳しく説明します。

4.7. LDAP 接続オプション

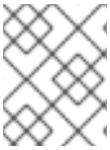
LDAP 接続情報は、JNDI 初期コンテキストの作成に使用される環境オブジェクトに渡される設定オプションとして提供されます。これらの設定オプションは、Ldap ログインモジュール、LdapExtended ログインモジュール 、AdvancedLdap ログインモジュール、および AdvancedAdLdap ログインモジュールで使用できます。

使用される標準 LDAP JNDI プロパティーには以下が含まれます。

オプション	Type	デフォルト	説明
java.naming.factory.initial	クラス名	com.sun.jndi.ldap.LdapCtxFactory	InitialContextFactory 実装クラス名。
java.naming.provider.url	ldap:// URL	Java.naming.security.protocol の値が SSL、ldap://localhost:636 の場合、それ以外は ldap://localhost:389	LDAP サーバーの URL。
java.naming.security.authentication	SASL メカニズムの none、simple、または name	デフォルトは **simple** です。プロパティーが明示的に定義されていない場合、この動作はサービスプロバイダーによって決定されます。	LDAP サーバーにバインドするために使用するセキュリティーレベル。
java.naming.security.protocol	トランスポートプロトコル	指定されていない場合、プロバイダーによって決定されます。	SSL などのセキュアなアクセスに使用するトランスポートプロトコル。
java.naming.security.principal	String	none	サービスへの呼び出し元を認証するためのプリンシパルの名前。これは、以下で説明されているその他のプロパティーから構築されます。

オプション	Type	デフォルト	説明
java.naming.security.credentials	認証情報のタイプ	none	認証スキームによって使用される認証情報のタイプ。ハッシュ化されたパスワード、クリアテキストのパスワード、キー、証明書などが例となります。このプロパティーが指定されていない場合、この動作はサービスプロバイダーによって決定されます。

ユーザー認証は、ログインモジュール設定オプションに基づいて LDAP サーバーに接続することで実行されます。LDAP サーバーへの接続は、LDAP JNDI プロパティーで構成される環境で **InitialLdapContext** を作成して行います。実際に使用される初期のコンテキスト実装は、設定された初期コンテキストファクトリーメソッドによって異なります。初期コンテキストファクトリーは **java.naming.factory.initial** プロパティーを使用して定義され、その設定を提供された環境プロパティーから取得します (例: **java.naming.provider.url**)。これにより、任意のプロパティーおよび関連するログインモジュールオプションがカスタムの初期コンテキストファクトリーに使用できます。



注記

javax.naming.Context interface javadoc で利用可能な、初期コンテキストを作成するための追加のデフォルトオプションおよび共通オプション。

4.8. LDAPUSERS ログインモジュール

短縮名: LdapUsers

フルネーム: org.jboss.security.auth.spi.LdapUsersLoginModule

親: UsernamePasswordLoginModule

LdapUsers モジュールは、LdapExtended モジュールおよび AdvancedLdap モジュールに置き換えられました。

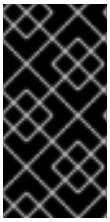
4.9. KERBEROS ログインモジュール

短縮名:: Kerberos

フルネーム: org.jboss.security.negotiation.KerberosLoginModule

Kerberos ログインモジュールは、GSSAPI を使用して Kerberos ログイン認証を実行します。このログインモジュールは、Oracle JDK の JDK 提供モジュール **com.sun.security.auth.module.Krb5LoginModule** と IBM JDK の **com.ibm.security.auth.module.Krb5LoginModule** をラップし、クレデンシャル委任の追加ロジックを提供し、**GSSCredential** を設定された Subject に追加します。

このモジュールは、認証およびロールマッピングを処理する別のモジュールとペアにする必要があります。



重要

以下の表は、**org.jboss.security.negotiation.KerberosLoginModule** で利用可能なオプションを示していますが、JDK によって提供されるモジュールからのオプションも設定できます。各 JDK モジュールオプションの詳細は、Oracle および IBM Java のドキュメントを参照してください。

表4.5 Kerberos ログインモジュールオプション

オプション	Type	デフォルト	説明
delegationCredential	IGNORE、REQUIRE または USE	IGNORE	このログインモジュールが委任を処理する方法を定義します。**IGNORE** は、委任クレデンシャルを使用せず、通常の Kerberos 認証を実行するように指定します。**USE** は、Subject を設定できる場合 **GSSCredential** の使用を指定し、利用できない場合は標準の Kerberos 認証にフォールバックします。**REQUIRE** は、**GSSCredential** を使用せず、認証が失敗した場合に認証に失敗するように指定します。
addGSSCredential	boolean	false	**GSSCredential** を、設定された Subject のプライベート認証情報に追加できるようにします。
wrapGSSCredential	boolean	false	無効化を防ぐために、Subject に追加される **GSSCredential** をラップすべきかどうかを指定します。**GSSCredential** が Subject に追加されていない場合は効果がありません。
credentialLifetime	Integer	**GSSCredential.DEFAULT_LIFETIME**	**GSSCredential**の有効期間 (秒単位)。負の値の場合は、これを **GSSCredential.indefinite_LIFETIME** に設定します。

4.10. SPNEGO ログインモジュール

短縮名: SPNEGO

フルネーム: org.jboss.security.negotiation.spnego.SPNEGOLoginModule

親: CommonLoginModule

SPNEGO ログインモジュールは、KDC で呼び出し元のアイデンティティーとクレデンシャルを確立するログインモジュールの実装です。モジュールは、SPNEGO、Simple、および Protected GSSAPI Negotiation メカニズムを実装し、JBoss Negotiation プロジェクトの一部です。この認証を AdvancedLdap ログインモジュールとチェーンされた設定で使用すると、LDAP サーバーと連携できます。Web アプリケーションは、このログインモジュールを使用するために、アプリケーション内で **org.jboss.security.negotiation.NegotiationAuthenticator** バルブも有効にする必要があります。

表4.6 SPNEGO ログインモジュールオプション

オプション	Type	デフォルト	説明
serverSecurityDomain	文字列	null	Kerberos ログインモジュールを介してサーバーサービスの ID を取得するために使用されるドメインを定義します。このプロパティーを設定する必要があります。
removeRealmFromPrincipal	boolean	false	さらなる処理を行う前に Kerberos レルムをプリンシパルから削除する必要があることを指定します。
usernamePasswordDomain	String	null	Kerberos が失敗した場合にフェイルオーバーログインとして使用する必要のある設定内の別のセキュリティードメインを指定します。

第5章 CERTIFICATE-BASED ベースのログインモジュール

5.1. CERTIFICATE ログインモジュール

短縮名: Certificate

フルネーム: org.jboss.security.auth.spi.BaseCertLoginModule

親: AbstractServerLoginModule

Certificate ログインモジュールは、X509 証明書を基にユーザーを認証します。このログインモジュールの典型的なユースケースが、web 層の **CLIENT-CERT** 認証です。証明書ログインモジュールは認証のみを実行するため、セキュアな web または EJB コンポーネントへのアクセスを完全に定義するには、承認ロールを取得できる他のログインモジュールと組み合わせる必要があります。このログインモジュールの 2 つのサブクラスである **CertRolesLoginModule** および **DatabaseCertLoginModule** は動作を拡張し、プロパティーファイルまたはデータベースから承認ロールを取得します。

表5.1 Certificate ログインモジュールオプション

オプション	Type	デフォルト	説明
securityDomain	String	other	信頼できる証明書を保持するトラストストアの JSSE 設定を持つセキュリティードメインの名前。
verifier	class	none	ログイン証明書の検証に使用する **org.jboss.security.auth.certs.X509CertificateVerifier** のクラス名。

5.2. CERTIFICATEROLES ログインモジュール

短縮名: CertificateRoles

フルネーム: org.jboss.security.auth.spi.CertRolesLoginModule

親: Certificate ログインモジュール

CertificateRoles ログインモジュールは、以下のオプションを使用してプロパティーファイルからロールマッピング機能を追加します。

表5.2 CertificateRoles ログインモジュールオプション

オプション	Type	デフォルト	説明

オプション	Type	デフォルト	説明
rolesProperties	String	roles.properties	各ユーザーに割り当てるロールを含むリソースまたはファイルの名前です。ロールプロパティーファイルは、username=role1,role2 の形式で指定する必要があります.ここで、ユーザー名は証明書の DN となり、等号 (=) および空白文字をエスケープします。以下の例では、**CN\=unit-tests-client,\ OU\=Red\ Hat\ Inc.,\ O\=Red\ Hat\ Inc.,\ ST\=North\ Carolina,\ C\=US** の形式を使用しています。
defaultRolesProperties	String	defaultRoles.properties	rolesProperties ファイルが見つからない場合は、フォールバックするリソースまたはファイルの名前です。
roleGroupSeparator	1 文字	. (シングルピリオド)	rolesProperties ファイルのロールグループの区切り文字として使用する文字。

5.3. DATABASECERTIFICATE ログインモジュール

短縮名: DatabaseCertificate

フルネーム: org.jboss.security.auth.spi.DatabaseCertLoginModule

親: Certificate ログインモジュール

DatabaseCertificate ログインモジュールは、以下の追加オプションを使用して、データベーステーブルからマッピング機能を追加します。

表5.3 DatabaseCertificate ログインモジュールオプション

オプション	Type	デフォルト	説明
dsJndiName	JNDI リソース	java:/DefaultDS	認証情報を格納している JNDI リソースの名前。

オプション	Type	デフォルト	説明
rolesQuery	準備済み SQL ステートメント	select **Role**,**RoleGroup** from **Roles** where **PrincipalID**=?	ロールをマッピングするために実行される SQL の準備済みステートメント。これは、「'select **Role**, **RoleGroup** from **Roles** where **PrincipalID**=?'」のクエリーと同等です。ここでは、**Role** は ロール名で、**RoleGroup** 列値は常に大文字の **R** または **CallerPrincipal** を持つ **Roles** のいずれかにしてください。
suspendResume	true または false	true	データベースの操作中に既存の JTA トランザクションを一時停止するかどうか。
transactionManagerJndiName	JNDI リソース	java:/TransactionManager	ログインモジュールによって使用されるトランザクションマネージャーの JNDI 名。

第6章 EJB およびリモーティングのログインモジュール

6.1. REMOTING ログインモジュール

短縮名: Remoting

フルネーム: org.jboss.as.security.remoting.RemotingLoginModule

親: AbstractServerLoginModule

Remoting ログインモジュールを使用すると、リモーティングを通じたリモート EJB 呼び出しが SASL ベースの認証を実行できます。これにより、リモートユーザーは SASL 経由でアイデンティティーを確立でき、EJB 呼び出しを行うときにそのアイデンティティーが認証および承認に使用されます。

表6.1 Remoting ログインモジュールオプション

オプション	Type	デフォルト	説明
useClientCert	boolean	false	**true** の場合、ログインモジュールは接続の **SSLSession** を取得し、パスワードの代わりにピアの **X509Certificate** を置き換えます。

6.2. CLIENT ログインモジュール

短縮名: Client

フルネーム: org.jboss.security.ClientLoginModule

Client ログインモジュールは、呼び出し元のアイデンティティーおよびクレデンシャルの確立時に JBoss EAP 6 クライアントによって使用されるログインモジュールの実装です。新しい SecurityContext を作成してプリンシパルとクレデンシャルに割り当て、SecurityContext を ThreadLocal セキュリティーコンテキストに設定します。Client ログインモジュールは、クライアントが現在のスレッドの呼び出し元を確立するために唯一サポートされるログインモジュールです。セキュリティー環境が JBoss EAP EJB security サブシステムを使用するよう透過的に設定されていない EJB クライアントとして動作するサーバー環境とスタンドアロンクライアントアプリケーションは、Client ログインモジュールを使用する必要があります。

注記

JBoss EAP 6.3 以降では、EJB およびリモートクライアント内でインターセプターを設定して発信者の ID を変更することもできます。JBoss EAP に同梱されている **ejb-security-interceptors** クイックスタートには、完全な動作例が含まれます。クイックスタートをダウンロードしてインストールする方法は、Red Hat JBoss Enterprise Application Platform 6 開発ガイド の 最初のアプリケーションの実行セクション を参照してください。

警告

このログインモジュールは認証を実行しません。サーバー上の後続の認証のために、提供されたログイン情報をサーバー EJB 呼び出しレイヤーにコピーすることもほとんどありません。JBoss EAP 6 内では、JVM 内の呼び出しに対してユーザーのアイデンティティーを切り替える目的で場合のみサポートされます。リモートクライアントがアイデンティティーを確立する目的では**サポートされません**。

表6.2 Client ログインモジュールオプション

オプション	Type	デフォルト	説明
マルチスレッド	true または false	true	各スレッドに独自のプリンシパルおよび認証情報ストレージがある場合は true に設定されます。仮想マシンのすべてのスレッドが同じアイデンティティーおよび認証情報を共有することを示すには、false に設定します。
password-stacking	**useFirstPass** または false	false	**UseFirstPass** に設定して、このログインモジュールがアイデンティティーとして使用する LoginContext に保存されている情報を検索することを示します。このオプションは、このログインモジュールと他のログインモジュールをスタックする際に使用できます。
restore-login-identity	true または false	false	**login()** メソッドの開始時に表示されるアイデンティティーおよび認証情報が **logout()** メソッドの呼び出し後に復元される必要がある場合は true に設定します。

第7章 カスタムログインモジュール

JBoss EAP セキュリティーフレームワークとバンドルされるログインモジュールがセキュリティー環境の要件に対応できない場合、カスタムログインモジュール実装を作成できます。**org.jboss.security.AuthenticationManager** は、Subject プリンシパルの特定の使用パターンを必要とします。**org.jboss.security.AuthenticationManager** と動作するログインモジュールを作成するには、JAAS Subject クラスの情報ストレージ機能と、これらの機能の想定される使用方法を完全に理解する必要があります。カスタムログインモジュールは **javax.security.auth.spi.LoginModule** の実装である必要があります。カスタム認証モジュールの作成に関する詳細は、API ドキュメントを参照してください。

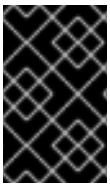
第8章 承認モジュール

以下のモジュールは、承認サービスを提供します。

code	クラス
DenyAll	org.jboss.security.authorization.modules.AllDenyAuthorizationModule
PermitAll	org.jboss.security.authorization.modules.AllPermitAuthorizationModule
Delegating	org.jboss.security.authorization.modules.DelegatingAuthorizationModule
web	org.jboss.security.authorization.modules.web.WebAuthorizationModule
JACC	org.jboss.security.authorization.modules.JACCAuthorizationModule
XACML	org.jboss.security.authorization.modules.XACMLAuthorizationModule

AbstractAuthorizationModule

これは、上書きが必要なベース承認モジュールで、他の認可モジュールへ委任する機能を提供します。このベース承認モジュールは、オーバーライドクラスに **delegateMap** プロパティーも提供します。これにより、特定コンポーネントに対して委任モジュールを宣言できます。これにより、**web**、**ejb** など、各レイヤーの承認を処理するためのより特殊なクラスが有効になります。これは、ユーザーの承認に使用される情報がアクセスされるリソース間で異なる可能性があるためです。たとえば、承認モジュールはパーミッションに基づくものであっても、**web** および **ejb** リソースの異なるパーミッションタイプを持つことができます。デフォルトでは、承認モジュールは可能なすべてのリソースおよびパーミッションタイプに対応するよう強制されますが、**delegateMap** オプションを設定すると、モジュールは異なるリソースタイプの特定のクラスに委譲できます。**delegateMap** オプションは、コンマ区切りのモジュール一覧を取ります。各モジュールのプレフィックスは、関連するコンポーネントによって指定されます.たとえば、**<module-option name="delegateMap">web=xxx.yyy.MyWebDelegate,ejb=xxx.yyy.MyEJBDelegate</module-option>** のようになります。



重要

delegateMap オプションを設定する場合、すべての委譲は **authorize (Resource)** メソッドを実装し、提供された承認モジュールと同じように **invokeDelegate (Resource)** メソッドを呼び出しる必要があります。これを行わないと、委譲は呼び出されません。

AllDenyAuthorizationModule

これは、承認要求を常に拒否する簡単な承認モジュールです。設定オプションは利用できません。

AllPermitAuthorizationModule

これは、常に承認要求を許可する簡単な承認モジュールです。設定オプションは利用できません。

DelegatingAuthorizationModule

これは、設定された委譲に決定を委譲するデフォルトの認可モジュールです。このモジュールは、**delegateMap** オプションもサポートします。

WebAuthorizationModule

デフォルトの Tomcat 承認ロジック (permit all) を持つデフォルトの Web 認証モジュールです。

JACCAuthorizationModule

このモジュールは、Web コンテナー承認要求の場合は WebJACCPolicyModuleDelegate、EJB コンテナー要求の場合は EJBJACCPolicyModuleDelegate を使用して JACC セマンティクスを有効にします。このモジュールは、**delegateMap** オプションもサポートします。

XACMLAuthorizationModule

このモジュールは、Web コンテナーおよび EJB コンテナー、WebXACMLPolicyModuleDelegate および EJBXACMLPolicyModuleDelegate の委譲を使用して XACML 承認を有効にします。登録したポリシーに基づいて PDP オブジェクトを作成し、それに対して web または EJB リクエストを評価します。このモジュールは、**delegateMap** オプションもサポートします。

第9章 セキュリティーマッピングモジュール

JBoss EAP 6 では、以下のセキュリティーマッピングモジュールが提供されています。

クラス	code	Type
org.jboss.security.mapping.providers.role.PropertiesRolesMappingProvider	PropertiesRoles	role
org.jboss.security.mapping.providers.role.SimpleRolesMappingProvider	SimpleRoles	role
org.jboss.security.mapping.providers.DeploymentRolesMappingProvider	DeploymentRoles	role
org.jboss.security.mapping.providers.role.DatabaseRolesMappingProvider	DatabaseRoles	role
org.jboss.security.mapping.providers.role.LdapRolesMappingProvider	LdapRoles	role
org.jboss.security.mapping.providers.attribute.LdapAttributeMappingProvider	LdapAttributes	attribute
org.jboss.security.mapping.providers.DeploymentRoleToRolesMappingProvider		role
org.jboss.security.mapping.providers.attribute.DefaultAttributeMappingProvider		attribute

9.1. PROPERTIESROLESMAPPINGPROVIDER

コード: PropertiesRoles

クラス: org.jboss.security.mapping.providers.role.PropertiesRolesMappingProvider

タイプ: role

以下の形式でプロパティーファイルからロールを読み取る。**username=role1,role2,...**

オプション	型	説明
rolesProperties	文字列	フォーマットされたファイル名のプロパティー。JBoss EAP 6 変数の拡張は **${jboss.variable}**の形式で使用できます。

9.2. SIMPLEROLESMAPPINGPROVIDER

コード: SimpleRoles

クラス: org.jboss.security.mapping.providers.role.SimpleRolesMappingProvider

タイプ: role

オプションマップからロールを読み取る簡単な MappingProvider。option 属性 name はロールを割り当てるプリンシパルの名前です. attribute の値は、プリンシパルに割り当てるカンマ区切りのロール名です。

例

```
<module-option name="JavaDuke" value="JBossAdmin,Admin"/>
<module-option name="joe" value="Users"/>
```

9.3. DEPLOYMENTROLESMAPPINGPROVIDER

コード: DeploymentRoles

クラス: org.jboss.security.mapping.providers.DeploymentRolesMappingProvider

タイプ: role

jboss-web.xml および **jboss-app.xml** デプロイメント記述子で実行できるロールマッピングに対してプリンシパルを考慮に入れるロールマッピングモジュール。

例

```
<jboss-web>
...
  <security-role>
     <role-name>Support</role-name>
     <principal-name>Mark</principal-name>
     <principal-name>Tom</principal-name>
  </security-role>
...
</jboss-web>
```

9.4. DATABASEROLESMAPPINGPROVIDER

コード: DatabaseRoles

クラス: org.jboss.security.mapping.providers.role.DatabaseRolesMappingProvider

タイプ: role

データベースからロールを読み取る MappingProvider。

オプション	型	説明
dsJndiName	String	ロールをユーザーにマップするために使用されるデータソースの JNDI 名。
rolesQuery	String	このオプションは、**select RoleName from Roles where User=?** と同等の準備済みセテートメントである必要があります。? は現在のプリンシパル名に置き換えます。
suspendResume	boolean	**true** の場合、ロールの検索中に現在のスレッドに関連付けられたトランザクションを一時停止および再開します。
transactionManagerJndiName	文字列	Transaction mamager の JNDI 名 (デフォルトは **java:/TransactionManager**)

9.5. LDAPROLESMAPPINGPROVIDER

コード: LdapRoles

クラス: org.jboss.security.mapping.providers.role.LdapRolesMappingProvider

タイプ: role

ロールを検索するため LDAP サーバーを使用してロールを割り当てるマッピングプロバイダー。

オプション	型	説明
bindDN	String	ユーザーおよびロールクエリーの LDAP サーバーに対してバインドするために使用される DN です。この DN には、**baseCtxDN** および **rolesCtxDN** 値の読み取りおよび検索パーミッションが必要です。
bindCredential	String	**bindDN** のパスワード。これは、vault メカニズムを介して暗号化できます。

オプション	型	説明
rolesCtxDN	String	ユーザーロールを検索するためのコンテキストの固定 DN です。これは、実際のロールがである DN ではなく、ユーザーロールを含むオブジェクトがある DN です。たとえば、Microsoft Active Directory サーバーでは、ユーザーアカウントが DN になります。
roleAttributeID	文字列	承認ロールの名前が含まれる LDAP 属性です。
roleAttributeIsDN	boolean	**roleAttributeID** にロールオブジェクトの完全修飾 DN が含まれるかどうか。**false** の場合は、コンテキスト名の **roleNameAttributeId** 属性の値からこのロール名が取得されます。Microsoft Active Directory などの特定のディレクトリースキーマでは、この属性を **true** に設定する必要があります。
roleNameAttributeID	文字列	ロール名を含む **roleCtxDN** コンテキスト内の属性の名前。**roleAttributeIsDN** プロパティーを **true** に設定すると、このプロパティーはロールオブジェクトの名前属性の検索に使用されます。
parseRoleNameFromDN	boolean	クエリーによって返された DN に **roleNameAttributeID** が含まれるかどうかを示すフラグ。**true** に設定した場合には、DN は **roleNameATtributeID** に対してチェックされます。**false** に設定すると、DN は **roleNameATtributeID** に対して確認されません。このフラグにより、LDAP クエリーのパフォーマンスを向上できます。

オプション	型	説明
roleFilter	String	認証されたユーザーに関連付けられたロールを見つけるために使用される検索フィルター。{0} 式を使用しているフィルターに、入力**ユーザー名**、またはログインモジュールコールバックから取得した **userDN** が置換されます。認証済み **userDN** は {1} が使用されたフィルターに置き換えられます。入力 **ユーザー名** に一致する検索フィルター例は **(member={0})** です。認証済み **userDN** に一致する他の例は **(member={1})** です。
roleRecursion	number	ロール検索の再帰レベルの数は、一致するコンテキストの下に続きます。これを 0 に設定して再帰を無効にします。
searchTimeLimit	number	ユーザー/ロール検索のタイムアウト (ミリ秒単位)。デフォルト値は 10000 です。
searchScope	文字列	使用する検索範囲。

9.6. LDAPATTRIBUTEMAPPINGPROVIDER

コード: LdapAttributes

クラス: org.jboss.security.mapping.providers.attribute.LdapAttributeMappingProvider

タイプ: attribute

LDAP からサブジェクトに属性をマッピングします。オプションには、LDAP JNDI プロバイダーがサポートするオプションが含まれます。

標準プロパティー名の例

```
Context.INITIAL_CONTEXT_FACTORY = "java.naming.factory.initial"
Context.SECURITY_PROTOCOL = "java.naming.security.protocol"
Context.PROVIDER_URL = "java.naming.provider.url"
Context.SECURITY_AUTHENTICATION = "java.naming.security.authentication"
```

オプション	型	説明

オプション	型	説明
bindDN	String	ユーザーおよびロールクエリーの LDAP サーバーに対してバインドするために使用される DN です。この DN には、**baseCtxDN** および **rolesCtxDN** 値の読み取りおよび検索パーミッションが必要です。
bindCredential	String	BindDN のパスワード。これは、**jaasSecurityDomain** が指定されている場合に暗号化できます。
baseCtxDN	String	ユーザーの検索を開始するためのコンテキストの固定 DN です。
baseFilter	String	認証するユーザーのコンテキストを見つけるために使用される検索フィルター。{0} 式を使用しているフィルターに、入力**ユーザー名**、またはログインモジュールコールバックから取得した **userDN** が置換されます。この置換の動作は、**DirContext.search(Name, String, Object[], SearchControls cons)** メソッドから実行されます。一般的な検索フィルターの例は **(uid={0})** です。
searchTimeLimit	number	ユーザー/ロール検索のタイムアウト (ミリ秒単位)。デフォルト値は 10000 です。
attributeList	String	ユーザーの属性のカンマ区切りリスト。例: **mail,cn,sn,employeeType,employeeNumber**。

オプション	型	説明
jaasSecurityDomain	String	**java.naming.security.credentials** の復号化に使用する **JaasSecurityDomain**。パスワードの暗号化された形式は、**JaasSecurityDomain#encrypt64(byte)** メソッドによって返されるものです。**org.jboss.security.plugins.PBEUtils** を使用して、暗号化されたフォームを生成することもできます。

9.7. DEPLOYMENTROLETOROLESMAPPINGPROVIDER

Class: org.jboss.security.mapping.providers.DeploymentRoleToRolesMappingProvider

タイプ: role

jboss-web.xml および **jboss-app.xml** デプロイメント記述子で実行できるロールマッピングに対してプリンシパルを考慮に入れるロール間のマッピングモジュール。この場合、**principal-name** は、他のロールをマップするためのロールを示します。

例

```
<jboss-web>
...
   <security-role>
     <role-name>Employee</role-name>
     <principal-name>Support</principal-name>
     <principal-name>Sales</principal-name>
   </security-role>
...
</jboss-web>
```

上記の例では、**Support** または **Sales** のロールを持つ各プリンシパルには、**Employee** のロールも割り当てられます。

注記

このマッピングプロバイダーにはコードが関連付けられていないため、設定時には完全なクラス名が **code** フィールドになければなりません。

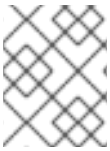
9.8. DEFAULTATTRIBUTEMAPPINGPROVIDER

クラス: org.jboss.security.mapping.providers.attribute.DefaultAttributeMappingProvider

タイプ: attribute

モジュールをチェックし、マッピングコンテキストからプリンシパル名を見つけ、**principalName** + **.email** という名前のモジュールオプションから属性電子メールアドレスを作成し、それを指定のプリンシパルへマップします。

オプション	型	説明
principalName	文字列	属性の電子メールアドレスを作成するために使用されるプリンシパル名。



注記

このマッピングプロバイダーにはコードが関連付けられていないため、設定時には完全なクラス名が **code** フィールドになければなりません。