My webMethods Server 管理者ガイド

バージョン 10.11

2021 年 10 月

WEBMETHODS

このマニュアルは、My webMethods Server バージョン 10.11 およびそれ以降のリリースに適用されます。

このマニュアルに含まれる仕様は、変更されることがあります。変更内容については、それ以降のリリースノート、または次のエディションで報告されます。

Copyright © 2004-2021 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

Software AG および、すべての Software AG Software AG, Software AG USA Inc.、またはその子会社やライセンサーの商標または登録商標です。このマニュアルに記載されたその他の企業名および製品名は、それぞれの所有者の商標です。

Software AG またはその子会社、もしくはその両方の商標および特許についての詳細は、次の場所を参照してください。
http://softwareag.com/licenses/

このソフトウェアを使用する際は、Software AG のライセンス契約条件を順守する必要があります。これらの契約条件は、http://softwareag.com/licenses/ またはライセンス製品のルートインストールディレクトリから取得できる製品ドキュメントの一部として参照できます。

本ソフトウェアはサードパーティ製品の一部を含む場合があります。サードパーティの権利、ライセンス条項、追加の権利や制約については、「License Texts, Copyright Notices and Disclaimers of Third Party Products」を参照してください。特定のサードパーティのライセンス条項や制約については、「License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products」の「Legal Notices」の E 項を参照してください。これらのドキュメントは、http://softwareag.com/licenses/ またはライセンス製品のルートインストールディレクトリから取得できる製品ドキュメントの一部として参照できます。

ドキュメントID: MWS-AG-1011-20211126_JA

目次

このマニュアルについて **9**
My webMethods Server アセットのエクスポート 10
表記規則 10
オンライン情報とサポート 10
データ保護 11
webMethods Broker の廃止 11

I 起動および設定 **13**
1 My webMethods Server の概要 15
My webMethods Server の概要 16
Windows での My webMethods Server の起動および停止 17
UNIX での My webMethods Server の自動起動 17
My webMethods Server へのログイン 17
My webMethods Server からのログアウト 18
管理者アカウントへのアクセスの有効化 18
2 My webMethods Server の設定の変更 21
インストール後の設定変更 22
MySQL データベースで My webMethods Server を使用する 22
データベースを指定しないでインストールした場合 23
複数の My webMethods Server インスタンスのガイドライン 24
スタンドアロンサーバの HTTP リスナーポートの変更 24
HTTP および HTTPS リスナーポートの変更 25
JMX リスナーポートの変更 25
JCR リポジトリ RMI リスナーポートの変更 26
My webMethods Server で使用する Java の場所の指定 27
My webMethods Server でのリダイレクトの管理 28
3 My webMethods Server および HTTPS 31
My webMethods Server を HTTPS クライアントとして使用 32
HTTPS を使用した webMethods アプリケーションとの通信 36
My webMethods 管理者としての認証証明書の管理 37
4 My webMethods Server での Web サーバの使用 43
Web サーバとの統合 44
My webMethods Server および Apache 44
My webMethods Server および IIS 44
5 Docker での My webMethods Server の使用 45
My webMethods Server と Docker コンテナについて 46
Docker コンテナでの My webMethods Server の実行 46
My webMethods Server コンテナ使用時の考慮事項 46
My webMethods Server インストールの Docker イメージの作成およびカスタマイズ 47
My webMethods Server コンテナの実行およびカスタマイズ 50
最適化済み My webMethods Server コンテナの起動 62
My webMethods Server コンテナと webMethods Microservices Runtime の併用 63
My webMethods Server コンテナの監視 64
My webMethods Server コンテナのクラスタリング 69

6 コマンドラインからの My webMethods Server の実行....71
My webMethods Server の基本のコマンドライン構文....72
My webMethods Server コマンドの実行....73
コマンドラインヘルプへのアクセス....73
My webMethods Serverインスタンスの管理コマンド....73
My webMethods Serverインスタンスの操作コマンド....79
My webMethods Serverインスタンスの設定コマンド....82
My webMethods Server サービスの管理コマンド....84
My webMethods Server OSGi プロファイルコマンド....86
複数のサーバインスタンスでの My webMethods Server コマンドの開始、停止および実行....89
My webMethods Server のコマンドの JVM プロパティの設定....90
mws コマンドのログファイル....91
7 設定ファイルの変更....93
Java サービスラッパー....94
My webMethods Server の JVM 設定....97
データベースに保存された設定ファイルの変更....102
Solaris、HP-UX または Linux で My webMethods Server を 32 ビットで実行するための設定....104
HTTP リスナーの単一 IP アドレス使用の設定....105
起動時に診断を実行するかどうかの設定....108

II My webMethods Administrator の機能....111
8 My webMethods Server の設定の管理....113
ディレクトリサービスの管理....114
外部ディレクトリサービスの管理....116
外部データソースの管理....133
電子メール設定の管理....141
カレンダーの管理....143
9 ユーザ、グループ、および役割の検索....145
既存のユーザ、グループ、または役割の検索....146
拡張検索....147
保存済み検索の使用....148
.csv ファイルへの検索結果のエクスポート....151
10 ユーザとグループの管理....153
ユーザとグループの管理について....154
ユーザの管理....155
ユーザデータの管理....166
グループの管理....167
11 権限の管理....175
My webMethods での権限の管理....176
アクセス特権および機能特権の管理....179
個々のリソースの権限の管理....180
セキュリティ領域の使用....182
12 役割および My webMethods へのアクセスの管理....189
My webMethods Server における役割について....190
ユーザへの My webMethods へのアクセス権および My webMethods Users 役割の付与....191
役割の作成....192

役割の情報の編集....199
役割の削除....200
役割に関連付ける動的属性の定義....200
13 My webMethods Server のクラスタリング....205
My webMethods Server クラスタの仕組み....206
My webMethods Server クラスタの計画....210
My webMethods Server クラスタ構築の考慮事項....212
クラスタ設定の変更....213
クラスタのリソースロック設定の変更....223
クラスタの消去スレッドの数の設定....224
クラスタの監視および制御....224
クラスタからのコンポーネントの削除....228
cluster.xml ファイルの使用....228
イメージからのクラスタノードの作成....231
クラスタノードでのアプリケーションのパーティション化....233

III システム管理者の機能....245
14 属性プロバイダ....247
属性プロバイダとは....248
属性プロバイダの使用....249
プリンシパル属性プロバイダの表示の管理....254
15 セキュリティの管理....257
My webMethods Server セキュリティについて....258
認証の管理....270
ケルベロス認証の設定....274
NTLM 認証の設定....277
NTLMv2 認証の設定....278
外部設定クレデンシャルの設定....280
My webMethods Server シングルサインオンの設定....282
OAuth 2.0 認証の設定....289
メモリからのセッションパスワードの消去....293
メモリでのセッションパスワードの保持....293
ユーザ名およびパスワードのオートコンプリートのオン/オフの切り替え....294
失敗したログイン試行回数の制御....294
ログイン IP 範囲の制御....295
グローバル環境変数のパスワードの暗号化....297
コンテキストなりすましの許可....298
パスワードの複雑度ポリシーの使用....299
応答ヘッダールールの使用....302
16 分析、レポート、およびトラブルシューティング....309
分析、レポート、およびトラブルシューティングについて....310
サーバのログの制御....310
ログメッセージの表示....322
セキュリティ監査ログの管理....323
リアルタイムユーザアクティビティの監視....323
サーバイベントに関するデータの収集....324
データベース変更に関するデータの収集....326
My webMethods Server 診断ツール....327
17 My webMethods Server の設定....329

My webMethods Server の設定について....330
エイリアスの管理....330
My webMethods Server コンポーネントの展開....334
My webMethods Server キャッシュの設定....337
システム情報の表示....343
18 My webMethods Server コンテンツの管理....345
My webMethods Server コンテンツの移行....346
コンテンツ記憶領域の管理....346
管理者としてポートレットをパブリッシュ....349
検索インデックスの再構築....351
カスタム JAR ファイルの追加....351
19 ユーザインタフェースの管理....353
ロケール管理....354
サーバルールとは....354
ロケールルールの作成....356
ログインページルールの作成....358
スタートページルールの作成....360
レンダリングルールの作成....362
ルールの変更....365
ルールのコピー....366
ルールの評価順序の管理....367
ルールの削除....368
スキンルールの管理....368
シェルルールの管理....371
20 共通ディレクトリサービス API の使用....375
共通ディレクトリサービス API によるユーザ情報の管理....376
共通ディレクトリサービス API について....377
前提条件....377
CDS のバージョンの相互運用性について....378
CDS コード例....379
21 My webMethods Server からの Mobile 通知の送信....381
モバイルデバイスへのプッシュ通知の送信....382
My webMethods Server でのプッシュ通知の設定....383

IV サーバページの開発....385
22 My webMethods Server でのページの管理....387
ページの開発について....388
カスタムフォルダおよびカスタムページについて....388
My webMethods ナビゲーションのカスタマイズについて....407
My webMethods のルックアンドフィールのカスタマイズについて....412
My webMethods への簡単なフロントエンドページの作成....416
シングルサインオン用のリンクの作成....417
23 My webMethods Server でのワークスペースの管理....419
ワークスペースについて....420
ワークスペースの管理タスク....420
ワークスペース開発用のエキスパートユーザ機能....424
[ワークスペース管理] ページから実行できるワークスペースアクション....430
My webMethods ツールナビゲーションについて....438
24 スキンのカスタマイズ....441

スキンとは....442
新規スキンの作成および変更....444
[スキン管理] ページの使用....447
スキンパッケージの構成....457
25 My webMethods Server でのシェルの使用....469
シェルとは....470
新規シェルの作成....470
シェルの変更....471
HTML <head> 要素への追加タグの挿入....472
シェルセクションでのエイリアスの使用....472
シェルの削除....473
空のシェルセクションの作成....473

V Command Central を使用した My webMethods Server の管理....475
26 My webMethods Server 管理者ガイド....477
My webMethods Server インスタンスの管理....478
My webMethods Server ポートの設定....481
ディレクトリサービスの設定....482
My webMethods Server 電子メールの設定....489
My webMethods Server 環境変数の設定....490
My webMethods Server インスタンスの KPI の監視....491
信用できる認証を使用した My webMethods Server への接続....492
27 コマンドラインを使用する My webMethods Server の管理....493
My webMethods Server がサポートするコマンド....494
My webMethods Server-ENGINE がサポートする設定タイプ....496
My webMethods Server-ENGINE のライフサイクルアクション....498
My webMethods Server インスタンスの管理....499
My webMethods Server-ENGINE のランタイムの監視状態....504
My webMethods Server ランタイムの監視状態....505
28 My webMethods Server の認証....507
My webMethods Server の認証モードの変更....508
Command Central で My webMethods Server の管理者ユーザパスワードを変更するには....508
UNIX シェルスクリプトを使用した管理製品の接続クレデンシャルの変更....508
送信認証設定の確認....509
Command Central を介した管理インタフェースへのアクセス....509

このマニュアルについて

- My webMethods Server アセットのエクスポート 10
- 表記規則 10
- オンライン情報とサポート 10
- データ保護 11
- webMethods Broker の廃止 11

このマニュアルでは、システム管理者の立場から、My webMethods Server の設定および管理の方法について説明します。このマニュアルでは、My webMethods Administrator (システム管理者とは異なります) の観点から、My webMethods Server 上で実行されるアプリケーションのユーザ、グループ、および役割の管理方法について説明します。また、My webMethods Server で表示するページの開発および管理方法についても説明します。

My webMethods Server アセットのエクスポート

Deployer で使用するための My webMethods Server アセットの抽出の詳細については、『*webMethods Deployer User's Guide*』を参照してください。

表記規則

規則	説明
太字	画面上の要素を表します。
縮小フォント	*folder.subfolder.service* 形式のサービス名や場所、API、Java クラス、メソッド、プロパティを表します。
斜体	次の情報を表します。 独自の状況または環境に固有の値を指定する必要がある変数。 本文で最初に出現する新しい用語。 他のドキュメントソースへの参照。
モノスペースフォント	次の情報を表します。 入力する必要があるテキストまたはシステムで表示されるメッセージ。 プログラムコード。
{ }	選択肢のセットを表します。ここから 1 つ選択する必要があります。中カッコの内側にある情報のみを入力します。{ } 記号は入力しません。
|	構文行で相互排他的な 2 つの選択肢を区切ります。いずれかの選択肢を入力します。| 記号は入力しません。
[ ]	1 つ以上のオプションを表します。大カッコの内側にある情報のみを入力します。[ ] 記号は入力しません。
...	同じ種類の情報を複数回入力できることを示します。情報だけを入力してください。実際のコードに繰り返し記号 (...) を入力しないでください。

オンライン情報とサポート

Software AG マニュアルの Web サイト

マニュアルは、Software AG マニュアルの Web サイト (https://documentation.softwareag.com) で入手できます。

Software AG の Empower 製品サポート Web サイト

Empower のアカウントをまだお持ちでない場合は、氏名、会社名、および会社の電子メールアドレスを empower@softwareag.com まで送信し、アカウントをリクエストしてください。

アカウントを取得したら、Empower (https://empower.softwareag.com/) の [eService] セクションからオンラインでサポートインシデントを開くことができます。

製品情報は、Software AG の Empower 製品サポート Web サイト (https://empower.softwareag.com) で入手できます。

機能および拡張機能に関するリクエストの送信、製品の可用性に関する情報の取得、製品のダウンロードを実行するには、Products に移動します。

修正に関する情報を取得し、早期警告、技術論文、Knowledge Base の記事を読むには、Knowledge Center に移動します。

ご不明な点がありましたら、https://empower.softwareag.com/public_directory.aspx の [Global Support Contact Directory] でお住まいの国のローカルまたはフリーダイヤル番号を検索し、お電話でお問い合わせください。

Software AG Tech Community

マニュアルおよびその他の技術情報は、Software AG Tech Community Web サイト (https://techcommunity.softwareag.com) で入手できます。以下の操作を実行できます。

- Tech Community クレデンシャルを持っている場合は、製品マニュアルにアクセスできます。TECHcommunity クレデンシャルがない場合は、登録し、関心事の領域として [マニュアル] を指定する必要があります。
- 記事、コードサンプル、デモ、チュートリアルにアクセスする
- Software AG の専門家によって承認されたオンライン掲示板フォーラムを使用して、質問したり、ベストプラクティスを話し合ったり、他の顧客が Software AG のテクノロジをどのように使用しているかを学んだりすることが可能です。
- オープンスタンダードや Web テクノロジを取り扱う外部 Web サイトにリンクできます。

データ保護

Software AG 製品は、EU 一般データ保護規則 (GDPR) に従って個人データの処理に関する機能を提供します。該当する場合、それぞれの管理マニュアルに適切な手順が記載されています。

webMethods Broker の廃止

webMethods Broker は、webMethods 10.2 での使用が廃止されました。webMethods 10.2 で開発を開始する場合は、webMethods Broker ではなく Software AG Universal Messaging を使用してください。webMethods 10.2 へアップグレードする場合は、Universal Messaging への移行をご検討ください。webMethods Broker を継続して使用する場合、フルサポートが行われるのは、

発表されている webMethods Broker のサポート終了日までに限ります。詳細については、https://empower.softwareag.com/brokerendoflife/ を参照してください。

I 起動および設定

1 My webMethods Server の概要 15

2 My webMethods Server の設定の変更 21

3 My webMethods Server および HTTPS 31

4 My webMethods Server での Web サーバの使用 43

5 Docker での My webMethods Server の使用 45

6 コマンドラインからの My webMethods Server の実行 71

7 設定ファイルの変更 93

1 My webMethods Server の概要

- My webMethods Server の概要 16
- Windows での My webMethods Server の起動および停止 17
- UNIX での My webMethods Server の自動起動 17
- My webMethods Server へのログイン 17
- My webMethods Server からのログアウト 18
- 管理者アカウントへのアクセスの有効化 18

My webMethods Server の概要

My webMethods Server は、webMethods のアプリケーションで使用可能な機能のランタイムコンテナです。これらの機能を実行するユーザインタフェースは、My webMethods と呼ばれています。My webMethods は、ユーザが webMethods アプリケーションで機能を簡単に実行できるような環境を提供します。また、管理者はこれらの機能へのアクセスを管理できます。さらに、My webMethods Server は、追加のユーザインタフェースページを開発するための機能、および追加された機能を管理するための広範囲の管理ツールを提供します。

My webMethods Server は、実行する機能に基づいて、2 つのタイプの管理者を認識します。

- システム管理者 - My webMethods Server のシステム管理者。このユーザは、分析、設定、コンテンツ、およびユーザ管理を含む、My webMethods Server の管理を行うことができます。ユーザ ID は「`SysAdmin`」で、パスワードはイントール中に指定します。インストール中に選択したオプションによって、My webMethods Server は、`SysAdmin` ユーザの初回ログイン時に新規パスワードの設定をうながすプロンプトを表示することがあります。このユーザは My webMethods ユーザインタフェースを使用しません。
- My webMethods Administrator - My webMethods のデフォルト管理者。このユーザはユーザ管理機能を実行し、外部ディレクトリサービスを管理できます。ユーザ ID は「`Administrator`」です。デフォルトでは、このユーザはインストール後に無効になります。`SysAdmin` ユーザがログインして `Administrator` アカウントを有効にする必要があります。

ユーザ管理など、両方のタイプの管理者が実行できる機能もあります。My webMethods ユーザインタフェースとシステムユーザインタフェースで手順が異なる場合は、両方の手順を説明します。

このマニュアルは以下のように複数の部分に分かれています。次の表に、マニュアルの主要部分と、各部分に記載されているアクティビティを示します。

アクティビティ	参照先
管理者が行う初期作業、My webMethods Server の設定変更、外部 Web サーバの使用、コマンドラインからのサーバの実行。	起動および設定
My webMethods Administrator としてのユーザ、グループ、および役割の管理。	*My webMethods Administrator の機能*
システム管理者としての My webMethods Server の拡張機能の管理。	*システム管理者の機能*
ユーザインタフェースページの開発および管理。	サーバページの開発
Command Central を使用した My webMethods Server の管理。	*Command Central を使用した My webMethods Server の管理*

また、My webMethods Server についての詳細は、次のマニュアルも参照できます。

- My webMethods Server のインストールおよび初期設定 - 『*Installing Software AG Products*』。

- My webMethods アプリケーションで個々のユーザが実行可能な基本的アクティビティ - 『*Working with My webMethods*』。

Windows での My webMethods Server の起動および停止

My webMethods Server を起動するには、My webMethods Server の Windows サービスを起動します。

サーバをシャットダウンするには、My webMethods Server の Windows サービスを停止します。

サービスの名前は Software AGMy webMethods Server 10.2 です。サービスに複数のインスタンスがある場合は、2 番目のインスタンスには (2)、3 番目のインスタンスには (3) などのインデックス値が付きます。

UNIX での My webMethods Server の自動起動

UNIX システムに My webMethods Server をインストールし、システムの起動時に My webMethods Server を自動的に起動させるには、次のように My webMethods Server サービス登録スクリプトを実行します。

```
Software AG_directory/MWS/bin su -c ./mws.sh
-s serverInstance installservice serverUserAccount
```

My webMethods Server の起動の詳細については、71 ページの「コマンドラインからの My webMethods Server の実行」を参照してください。

My webMethods Server へのログイン

My webMethods Server には、Web ブラウザを使用してアクセスする管理者インタフェースがあります。`SysAdmin` ユーザだけが、インストールと起動直後に My webMethods Server にログインできます。他のすべての管理者アカウントは無効になっており、`SysAdmin` ユーザがそれらのアカウントを有効にして My webMethods へのアクセスを与える必要があります。

My webMethods Server に `SysAdmin` としてログインするには

1. Web ブラウザで URL を入力し、My webMethods Server の [ログイン] ページにアクセスします。

   ```
   http://host:port
   ```

   ここで、

   - *host* は、My webMethods Server がインストールされているマシンのホスト名。
   - *port* は、My webMethods Server が受信する要求を待機するポート。My webMethods Server のデフォルトポートは 8585 です。

たとえば、ホスト名が rubicon.company.com で、デフォルトポート (8585) を使用する場合、次の URL を入力します。

```
http://rubicon.company.com:8585
```

2. [**ユーザ名**] フィールドに「SysAdmin」と入力し、[**パスワード**] フィールドにインストール時に設定したパスワードを入力します。

   インストール中に選択したオプションによって、My webMethods Server が SysAdmin ユーザの初回ログイン時に、パスワードの変更をうながすプロンプトを表示することがあります。プロンプトが表示された場合、My webMethods Server のログイン画面で新規パスワードを指定して確認します。

3. [**ログイン**] をクリックします。

   ログインすると、My webMethods Server の Administration Dashboard が表示されます。

My webMethods Server からのログアウト

My webMethods Server からログアウトするには、以下の手順に従います。

My webMethods Server からログアウトするには

- すべての My webMethods ページの上部にある [**ログアウト**] リンクをクリックします。

管理者アカウントへのアクセスの有効化

デフォルトでは、インストール完了時に管理者権限を持つユーザアカウントは、SysAdmin ユーザを除き無効になっています。SysAdmin ユーザがこれらのアカウントを明示的に有効化し、新規パスワードを指定する必要があります。SysAdmin ユーザが無効になっていたユーザアカウントを有効にすると、管理者ユーザは My webMethods の [**マイプロファイル**] ページの [**ユーザ情報**] パネルでパスワードを変更することができます。

管理者アカウントへのアクセスを有効にし、パスワードを変更するには

1. システム管理者として、[**Administration Dashboard**] > [ユーザ管理] > [ユーザの管理] に移動します。

2. 編集するユーザを検索します。詳細については、146 ページの「既存のユーザ、グループ、または役割の検索」を参照してください。

3. 検索結果で、編集するユーザの行にある任意のリンクをクリックするか、 をクリックします。

4. [**ログインを無効にする**] チェックボックスを非選択にします。

5. [**ユーザ情報**] パネルで、[**パスワード**] フィールドに新規のパスワードを入力します。
6. 確認のため、[**パスワードの再入力**] フィールドに、新規のパスワードを再入力します。
7. [**適用**] をクリックします。

2 My webMethods Server の設定の変更

- インストール後の設定変更 22
- MySQL データベースで My webMethods Server を使用する 22
- データベースを指定しないでインストールした場合 23
- 複数の My webMethods Server インスタンスのガイドライン 24
- スタンドアロンサーバの HTTP リスナーポートの変更 24
- HTTP および HTTPS リスナーポートの変更 25
- JMX リスナーポートの変更 25
- JCR リポジトリ RMI リスナーポートの変更 26
- My webMethods Server で使用する Java の場所の指定 27
- My webMethods Server でのリダイレクトの管理 28

インストール後の設定変更

My webMethods Server のインストール時には、デフォルト設定が適用されます。サーバが使用するデータベースのタイプと場所、およびサーバが使用する HTTP ポートは指定できますが、それ以外の設定はできません。インストールが完了し、実行中の My webMethods Server のインスタンスがある場合は、設定を変更できます。

MySQL データベースで My webMethods Server を使用する

MySQL とともに My webMethods Server を使用するには、最初に My webMethods Server を起動する前に、MySQL データベースのモードを設定する必要があります。My webMethods Server で必要な MySQL モードは、`ANSI_QUOTES` および `PIPES_AS_CONCAT` です。

MySQL データベースで My webMethods Server を使用する場合は、.war ファイルの展開に失敗する場合があります。そのような問題が発生した場合、MySQL 設定ファイルで次のパラメータの値を増加し、MySQL データベースサーバを再起動します。

```
innodb_buffer_pool_size = 5G
innodb_log_file_size = 80M
max_allowed_packet = 24M
```

MySQL の設定の詳細については、MySQL のマニュアルを参照してください。

MySQL Community Edition を使用するための My webMethods Server の設定

MySQL Community Edition は、ネイティブドライバを使用する必要があります。My webMethods Server を MySQL Community Edition とともに使用する前に、ドライバをダウンロードして My webMethods Server インストールにバンドルとして追加する必要があります。

MySQL Community Edition 用に My webMethods Server を設定するには

1. 次の場所から MySQL のネイティブドライバ (`mysql-connector-java-`*`version`*`-bin.jar`) をダウンロードします。

   https://dev.mysql.com/downloads/connector/j/5.1.html

2. .jar ファイルを *`Software AG_directory`* `\MWS\lib` ディレクトリにコピーします。

3. *`Software AG_directory`* `\MWS\lib` ディレクトリに *`driver-name`*`.bnd` テキストファイルを作成します。*`driver-name`* は .jar ファイルの名前です。

   たとえば、`mysql-connector-java-`*`version`*`.bnd` のようになります。

4. 次の例の***斜体部分***で示すように、.bnd テキストファイルで OSGi バンドル変換の指示を指定します。

```
# attach as fragment to the caf.server bundle
Fragment-Host: com.webmethods.caf.server
Bundle-SymbolicName: mysql-connector-java
Bundle-Version: 5.1.41
Include-Resource: mysql-connector-java-5.1.41.jar
-exportcontents: *
Bundle-ClassPath: mysql-connector-java-5.1.41.jar
Import-Package: *;resolution:=optional
```

   Bundle-Version には、.jar ファイルのバージョン番号または任意の一意の数字を指定できます。

5. コマンドラインプロンプトで、サーバの bin ディレクトリに移動します。

```
Software AG_directory\MWS\bin
```

6. 次のコマンドを使用して、サーバインスタンスを更新します。

```
mws.bat -s serverName update
```

7. My webMethods Server を起動します。

データベースを指定しないでインストールした場合

My webMethods Server は、データベースを指定しないでインストールすることが可能です。このような状況で外部データベースを使用する場合、サーバが起動する前に、My webMethods Server のためにデータベース接続を設定する必要があります。外部データベース接続を設定または変更するには、mws.db.xml ファイルを使用します。このファイルは、以下の場所にあります。

Software AG_directory \MWS\server*serverName*\config

以下の値を指定する必要があります。

次の表では、データベース接続を設定するために mws.db.xml ファイルで指定する必要がある値を示します。

要素	値
<URL>	データベースサーバの URL
<USER>	データベースユーザ名
<PASSWORD>	データベースユーザのパスワード

ms.db.xml ファイルを保存して閉じた後、My webMethods Server を起動します。

複数の My webMethods Server インスタンスのガイドライン

同じマシン上で My webMethods Server の複数のインスタンスを実行できますが、各サーバインスタンスは、独自の外部リソースを必要とします。同じマシン上で複数のサーバインスタンスを実行することは、クラスタリングとは異なります。My webMethods Server クラスタリングの詳細については、205 ページの「My webMethods Server のクラスタリング」を参照してください。同じマシン上で複数のサーバインスタンスを実行する場合は、以下のガイドラインが適用されます。

- My webMethods Server の各インスタンスには独自のデータベースが必要です。同じデータベースサーバを使用するすべての My webMethods Server インスタンスを通じて、以下の項目を一意に設定する必要があります。
  - データベースユーザ名
  - データベース名またはテーブル領域名
- 同じマシンで同時に稼動している My webMethods Server インスタンスについては、以下のホスト/ポート番号の組み合わせ (使用されている場合) がすべてのサーバを通じて一意である必要があります。
  - HTTP ポート。デフォルトは 8585 です。
  - HTTPS ポート (使用する場合)。
  - JMX (Java Management Extensions) ポート。デフォルトは 5002 です。

スタンドアロンサーバの HTTP リスナーポートの変更

注:
webMethods Monitor を使用している場合に、My webMethods Server のデフォルト受信待機ポート 8585 を変更すると、My webMethods の [管理] > [ビジネスプロセス] ページにアクセス拒否エラーが表示されます。この問題を解決するには、Integration Server Administrator の [設定] > [リソース] ページの [**MWS SAML リゾルバ URL**] フィールドにデフォルト以外のポート番号を指定します。

スタンドアロンサーバインスタンスの HTTP リスナーポートのみを変更するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [My webMethods] > [クラスタ設定] > [基本設定]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [クラスタ管理] > [基本設定]
2. [**HTTP ポート**] フィールドで、新しいポート番号を入力し、[**サブミット**] をクリックします。

3. My webMethods Server を再起動します。

HTTP および HTTPS リスナーポートの変更

注:
webMethods Monitor を使用している場合に、My webMethods Server のデフォルト受信待機ポート 8585 を変更すると、My webMethods の [管理] > [ビジネスプロセス] ページにアクセス拒否エラーが表示されます。この問題を解決するには、Integration Server Administrator の [設定] > [リソース] ページの [**MWS SAML リゾルバ URL**] フィールドにデフォルト以外のポート番号を指定します。

My webMethods Server インスタンスの HTTP および HTTPS リスナーポートを変更するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [My webMethods] > [クラスタ設定] > [拡張またはクラスタ設定]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [クラスタ管理] > [拡張またはクラスタ設定]
2. [**HTTP ポート**] フィールドに、HTTP リスナーが使用するポート番号を入力します。

   このフィールドには必ず有効なポート番号を入力する必要があります。
3. [**HTTPS ポート**] フィールドに、HTTPS リスナーが使用するポート番号を入力します。

   このフィールドに値 0 (ゼロ) を入力すると、リスナーが無効になります。

   HTTPS リスナーで使用するデフォルトのデジタル認証の詳細については、34 ページの「セキュア接続に使用される証明書」を参照してください。
4. [**サブミット**] をクリックします。
5. My webMethods Server を再起動します。

JMX リスナーポートの変更

デフォルトの JMX リスナーポートは 5002 です。My webMethods Server の複数のインスタンスが同じコンピュータで実行している場合、または別のアプリケーションで既にそのポートを使用している場合は、別のポート番号を割り当てる必要があります。JMX リスナーポートは com.softwareag.jmx.connector.pid-5002.properties ファイルで変更します。

JMX リスナーポートを変更するには

1. テキストエディタで、サーバインスタンスの com.softwareag.jmx.connector.pid-5002.properties ファイルを開きます。ファイルは *Software AG_directory* \profiles\MWS_*serverName*\configuration\com.softwareag.platform.config.propsloader ディレクトリにあります。

2. 新しい JMX リスナーポート番号で port=*number* 行を編集して、ファイルを保存します。

3. My webMethods Server を再起動します。

JCR リポジトリ RMI リスナーポートの変更

リモート Java コンテンツリポジトリ (JCR) を使用する場合、JCR への接続は、クライアント My webMethods Server と通信するために RMI を使用して実装されます。My webMethods Server では、JCR RMI 接続に次のポートを使用します。

- **rmiPort**RMI レジストリのポート。デフォルトのポート番号は 10999 です。
- **rmiServerObjectPort**My webMethods Server によって提供されるリモートオブジェクトと通信するために、リモートクライアントによって使用されるポート。デフォルトのポート番号は 10998 です。値が 0 の場合は、ランダムなポート番号が生成されます。

重要:
サーバがファイアウォールで保護されるときに My webMethods Server へのリモート JCR 接続を許可する場合は、rmiPort と rmiServerObjectPort の両方を開くようにファイアウォールを設定する必要があります。そのようなシナリオでは、My webMethods Server が再起動するたびにポート番号が変更されないようにするため、rmiServerObjectPort に 0 以外の値を使用する必要があります。

rmiPort または rmiServerObjectPort の番号をデフォルトから変更するには

1. システム管理者として次のように選択します。[**管理用フォルダ**] > [Administration Dashboard] > [設定] > [CAF アプリケーションランタイム設定]

2. [**キーワード**] フィールドに「wm_mws_config」と入力して wm_mws_config 展開済みアプリケーションを検索します。

   My webMethods Server の検索結果で Administration アプリケーションが返されます。

3. [**Administration**] > [Web アプリケーション] > [環境エントリ] をクリックします。

4. [**rmiPort**] および [**rmiServerObjectPort**] フィールドで、ポート番号の新しい値を入力して [**適用**] をクリックします。

My webMethods Server で使用する Java の場所の指定

My webMethods Server は Java の場所を指している必要があります。デフォルトでは、My webMethods Server は *Software AG_directory* /jvm ディレクトリにインストールされた JDK の場所を指しています。

My webMethods Server で使用するために、デフォルト以外の JDK または JRE を指定できます。指定する場合でも、デフォルトの JDK は Software AG Installer で使用されるため、デフォルトの JDK を削除しないでください。

重要:
デフォルト以外の JRE または JDK を指定する場合は、自分自身でインストールした JRE および JDK と同様に、適切なベンダーからのメンテナンス更新を定期的に適用してください。

注:
Software AG による製品のテストは、Software AG Installer でインストールされた JDK を使用してのみ行っています。製品をリダイレクトして別の JDK または JRE を使用しているときに問題が発生した場合は、Software AG より、Software AG Installer でインストールされた JDK で問題が再現することを要求される場合があります。

デフォルト以外の JDK または JRE を指定するには、複数の場所で変更を行う必要があります。

/MWS/bin/ にある Setenv ファイル

次の場所にある setenv.bat または .sh ファイルで、指定された Java の場所を変更する必要があります。

Software AG_directory /MWS/bin/setenv.[bat | sh]

テキストエディタで setenv.bat または setenv.sh ファイルを開きます。JAVA_HOME パラメータがデフォルト以外の JDK または JRE のインストールディレクトリを指すように編集し、変更を保存してファイルを閉じます。次に例を示します。

```
set JAVA_HOME= C:\myjava17
```

/profiles にある Setenv ファイル

次の場所にある setenv.bat または .sh ファイルで、指定された Java の場所を変更する必要があります。

Software AG_directory /profiles/MWS_*serverName*/bin/setenv.[bat | sh]

テキストエディタで setenv.bat または setenv.sh ファイルを開きます。JAVA_EXEC パラメータがデフォルト以外の JDK または JRE のインストールディレクトリを指すように編集し、変更を保存してファイルを閉じます。次に例を示します。

```
set JAVA_EXEC= "C:\myjava17\bin\java"
```

wrapper.conf ファイル

次の場所にある wrapper.conf ファイルで、指定された Java の場所を変更する必要があります。

Software AG_directory /profiles/MWS_*serverName*/configuration/wrapper.conf

wrapper.conf ファイルをテキストエディタで開きます。wrapper.java.command パラメータがデフォルト以外の JDK または JRE のインストールディレクトリを指すように編集し、変更を保存してファイルを閉じます。次に例を示します。

```
wrapper.java.command=C:\myjava17\bin\java
```

custom_wrapper.conf ファイル

次の場所にある custom_wrapper.conf ファイルで、指定された Java の場所を変更する必要があります。

Software AG_directory /profiles/MWS_*serverName*/configuration/custom_wrapper.conf

custom_wrapper.conf ファイルをテキストエディタで開きます。set.JAVA_HOME パラメータがデフォルト以外の JDK または JRE のインストールディレクトリを指すように編集し、変更を保存してファイルを閉じます。次に例を示します。

```
set.JAVA_HOME=C:\\ myjava17
```

My webMethods Server でのリダイレクトの管理

My webMethods Server では、ページでボタンをクリックするか、その他のアクションを実行すると、外部サイトにリダイレクトされる場合があります。リダイレクトは便利な機能ですが、悪意のある外部サイトにユーザが誘導されることがあります。リンクコントロールまたはブックマークワークスペースツールを使用して作成した直接リンクは、この問題の影響を受けません。

ホワイトリストは、信用のあるエンティティのリストであり、この場合は信用のあるサーバです。[リダイレクトホワイトリスト管理] ページを使用して、My webMethods Server が安全に要求をリダイレクトできるサーバを含むホワイトリストを作成できます。

デフォルトでは、悪用から保護するために、My webMethods Server で外部サイトへのリダイレクトは許可されません。サーバでは、操作に必要なクラスタフロントエンド、`localhost` およびループバックアドレス (127.0.0.1) へのリダイレクトは許可されます。

外部サーバへのリダイレクトを許可する必要がある場合は、[リダイレクトホワイトリスト管理] ページを使用してそれらのサーバを追加できます。サーバは、ホスト名または IP アドレスで識別されます。ホワイトリストはサーバデータベースに格納され、My webMethods Server クラスタ内のすべてのサーバで使用できます。

ホワイトリストへのサーバの追加

外部アドレスをホワイトリストに追加することで、そのアドレスへのリダイレクトを許可できます。デフォルトでは、クラスタフロントエンド、`localhost` およびループバックアドレス (127.0.0.1) がリストに含まれています。

外部アドレスをホワイトリストに追加するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [リダイレクトホワイトリスト管理]

2. [**サーバの追加**] をクリックし、信用のあるサーバのホスト名または IP アドレスを入力して、[**適用**] をクリックし、[**保存**] をクリックします。

My webMethods Server で、新しいサーバへの URL リダイレクトが許可されるようになります。

ホワイトリストからのサーバの削除

ホワイトリストに外部アドレスが含まれている場合は、そのアドレスを削除できます。デフォルトでは、クラスタフロントエンド、`localhost` およびループバックアドレス (127.0.0.1) がリストに含まれており、これらは削除できません。

外部アドレスをホワイトリストから削除するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [リダイレクトホワイトリスト管理]

2. 削除するサーバのチェックボックスをオンにして、[**サーバの削除**] をクリックし、[**保存**] をクリックします。

サーバはホワイトリストから削除されます。

3 My webMethods Server および HTTPS

- My webMethods Server を HTTPS クライアントとして使用 32
- HTTPS を使用した webMethods アプリケーションとの通信 36
- My webMethods 管理者としての認証証明書の管理 37

My webMethods Server を HTTPS クライアントとして使用

HTTPS を使用して My webMethods Server をサーバに接続する場合は、My webMethods Server マシンで実行されている JVM で使用する、信用のある CA ストアファイル (cacerts ファイル) を設定する必要があります。信用のある CA ストアファイルには、My webMethods Server が HTTPS 要求を送信する先のサーバの CA 証明書が含まれている必要があります。

たとえば、webMethods Monitor または WmTaskClient パッケージを使用する場合は、監視ページに代わって My webMethods Server が発行する要求の送信先の Integration Server を指定する必要があります。HTTPS を使用して要求を発行する場合は、My webMethods Server マシンで実行されている JVM の信用のある CA ストアファイルを設定する (Integration Server の証明書に署名した認証局の CA 証明書を含める) 必要があります。

JVM の信用のある CA ストアファイルに CA 証明書をインポートするには、JVM の keytool コマンドを使用します。デフォルトでは、信用のある CA ストアファイルは以下の場所にあります。

Software AG_directory \jvm*operating_system*\jre\lib\security\cacerts

たとえば、Windows を使用している場合、信用のある CA ストアファイルの場所は *Software AG_directory* \jvm\jvm\jre\lib\security\cacerts になります。

CA 証明書のインポート

My webMethods Server JVM の信用のある CA ストアファイルに CA 証明書をインポートするには

1. 信用のある CA ストアファイルに追加する必要のある CA 証明書を見つけて、My webMethods Server が実行されているマシン上でこの証明書が使用可能であることを確認します。

2. コマンドラインプロンプトで、次のコマンドを入力して jvm\lib\security ディレクトリに移動します。

   ```
   cd Software AG_directory\jvm\operating_system\jre\lib\security
   ```

3. 以下のコマンドを入力して、信用のある CA ストアファイルに CA 証明書をインポートします。

   ```
   ..\..\..\bin\keytool -import -v -keystore
   cacerts -file <cacert.der> -alias <aliasName>
   ```

   ここで、

   - -file <*cacert.der*> は、インポートする CA 証明書を含むファイルのパスとファイル名を表します。

- -alias <*aliasName*>は、キーストアファイル内のエントリを識別するためのエイリアスを証明書に割り当てます。自分にとって意味のある値を選択してください。

たとえば、cacerts ファイルと同じディレクトリに保存されている serverCAcert.der という名前の CA 証明書をインポートし、キーストアファイルでこの新規エントリを SERVERCA として識別する場合は、以下のコマンドを使用します。

```
..\..\..\bin\keytool -import -v -keystore
cacerts -file serverCAcert.der -alias SERVERCA
```

4. keytool コマンドを入力すると、cacerts ファイルのパスワードを求めるプロンプトが表示されます。パスワードを入力します。デフォルトのパスワードは `changeit` です。

5. パスワードを入力すると、keytool コマンドから、CA 証明書のインポート操作を行うことを確認するプロンプトが表示されます。Yes を表す「y」を入力します。

6. 信用のある CA ストアファイルに CA 証明書が正しくインポートされたことを確認するため、以下のコマンドを入力します。

```
..\..\..\bin\keytool -list -keystore cacerts
```

keytool コマンドから、cacerts ファイルのパスワードを求めるプロンプトが表示されます。パスワードを入力します。

例

SSL を使用して、WmTaskClient パッケージと、同じコンピュータ上の My webMethods Server との通信を行うとします。この例では、デフォルトの My webMethods Server トラストストアを使用します。

1. まだ設定していない場合は、HTTPS ポートを使用するように My webMethods Server を設定します。たとえば、HTTPS ポートを 8586 に設定します。HTTPS ポートの設定方法の詳細については、「36 ページの「HTTPS を使用した webMethods アプリケーションとの通信」」を参照してください。

2. Integration Server で、前の手順で設定した HTTPS ポート (8586) を使用して通信するように、WmTaskClient パッケージを設定します。

3. CA 証明書を格納する一時ディレクトリを作成します (C:\temp など)。

4. コマンドラインプロンプトで、JVM keytool コマンドのディレクトリに移動します。

```
cd Software AG_directory\jvm\operating_system\jre\lib\security
```

5. 次のコマンドを入力して、デフォルトの My webMethods Server トラストストアから CA 証明書を抽出します。

```
keytool -export -alias "softwareag demo" -file c:\temp\sagca.crt
-keystore Software AG_directory\MWS\server\default\config\
security\sagdemoca.jks
```

6. プロンプトで、トラストストアのパスワードを入力します。デフォルトの My webMethods Server トラストストアの場合、パスワードは `manage` です。

7. JVM の信用のある CA ストアに CA 証明書をインポートするには、次のコマンドを入力します。

```
keytool -import -trustcacerts -file c:\temp\sagca.crt
-alias "softwareag demo" -keystore  Software AG_directory\jvm\
jvm160_32\jre\lib\security\cacerts
```

注:
64 ビットオペレーティングシステムで実行している場合は、`jvm160_32` を `jvm160_64` に変更します。

8. プロンプトで、JVM の信頼のある CA ストアファイルのパスワードを入力します。デフォルトのパスワードは `changeit` です。

9. CA 証明書のインポートを確定するには、Yes を表す「`y`」を入力します。

10. 新しい CA 証明書を使用するように、Integration Server を再起動します。

セキュア接続に使用される証明書

My webMethods Server には、HTTPS リスナーの設定とテストに使用できる 2 つのデフォルトキーストアが含まれています。

- キーストア: クライアントとサーバ間の暗号化接続の設定に使用されるキーペアが含まれます。デフォルトのキーストアファイルは localhost.p12 です。このファイルには、My webMethods Server によって使用される Jetty SSL サーバのデモ用証明書および秘密鍵が含まれています。
- トラストストア: CA (Certification Authority: 認証局) からの信頼のあるデジタル認証が含まれます。デフォルトのトラストストアファイル sagdemoca.jks には、My webMethods Server の 1 つのインスタンスが別のインスタンスまたは他の webMethods 製品からの SSL (Secure Sockets Layer) 接続を信頼できるようにする Software AG CA 証明書が含まれます。

My webMethods Server は、この 2 つのキーストアを各サーバインスタンスの次の場所に格納します。

Software AG_directory \MWS\server*serverName*\config\security\

実稼動環境では、Verisign などの商用認証局からの証明書を使用したり、内部認証局を使用したりできます。

キーストアの置換

wrapper.conf ファイルによって、SSL を使用した通信に使用される javax.net.ssl システムプロパティに値が設定されます。custom_wrapper.conf ファイルを編集して、My webMethods Server のインスタンス用のキーストアまたはトラストストアを置換できます。

キーストアまたはトラストストアを置換するには

1. テキストエディタで、サーバインスタンスの custom_wrapper.conf ファイルを開きます。ファイルは次の場所にあります。

Software AG_directory \profiles\MWS_*serverName*\configuration\

2. custom_wrapper.conf ファイルで、必要に応じて SSL プロパティの値を追加または変更して、ファイルを保存します。

```
# SSL Properties
set.JAVA_KEYSTORE=Software AG_directory\MWS\server\default\config
    \security\localhost.p12
set.JAVA_KEYSTORETYPE=pkcs12
set.JAVA_KEYSTORE_PASSWORD=encrypted_password
set.JAVA_TRUSTSTORE=Software AG_directory\MWS\server\default\config
    \security\sagdemoca.jks
set.JAVA_TRUSTSTORETYPE=jks
set.JAVA_TRUSTSTORE_PASSWORD=encrypted_password
```

3. My webMethods Server を再起動します。

キーストアの暗号化パスワードを指定するには、35 ページの「暗号化パスワードの生成」を参照してください。

暗号化パスワードの生成

キーストアを置換する場合、暗号化パスワードを指定する必要があります。パスワードを生成して、custom_wrapper.conf ファイルの適切な場所にコピーして貼り付けるには、以下の手順に従います。

この手順では、*Software AG directory*\jvm ディレクトリにインストールされている JDK、またはそれ以降のバージョンの JDK を使用する必要があります。

暗号化パスワードを生成するには

1. *Software AG_directory* \jvm\jvm\bin\java ディレクトリで、コマンドプロンプトウィンドウを開きます。

2. 次のコマンドを実行します。

   - Windows の場合

```
java -cp “Software AG directory\common\lib\wm-caf-common.jar;
Software AG directory\common\lib\wm-scg-security.jar;
Software AG directory\common\lib\wm-scg-core.jar;
Software AG directory\common\lib\ext\slf4j-api.jar;
Software AG directory\common\lib\ext\log4j\*”
com.webmethods.caf.common.CipherUtil password_to_encrypt
```

   - UNIX の場合

```
java -cp “Software AG directory/common/lib/wm-caf-common.jar:
Software AG directory/common/lib/wm-scg-security.jar:
Software AG directory/common/lib/wm-scg-core.jar:
Software AG directory/common/lib/ext/slf4j-api.jar:
Software AG directory/common/lib/ext/log4j/*”
com.webmethods.caf.common.CipherUtil password_to_encrypt
```

指定したパスワードの暗号化バージョンがコマンドから返されます。

3. 暗号化パスワードをコピーして、custom_wrapper.conf ファイル内の適切な場所に貼り付けます。

HTTPS を使用した webMethods アプリケーションとの通信

デフォルトでは、My webMethods Server は HTTP プロトコルを使用して他の webMethods アプリケーションと通信します。以下の方法で、HTTPS プロトコルを使用するように変更できます。

My webMethods Server の通信方法を HTTPS プロトコルのみに限定するには

1. My webMethods Server で、以下のいずれかを行います。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [My webMethods] > [クラスタ設定] > [拡張またはクラスタ設定]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [クラスタ管理] > [拡張またはクラスタ設定]
2. [**MWS フロントエンド URL**] フィールドで、以下のように URL を変更します。
   a. `http` を `https` に変更します。
   b. ポート番号を変更します。

      たとえば、以下の URL の場合、

      ```
      http://my_host:8585
      ```

      次のように変更します。

      ```
      https://my_host:7238
      ```
3. [**HTTP ポート**] フィールドで、値を 0 に変更します。

   注:
   My webMethods Server のクラスタを実行している場合は、クラスタ内のすべてのノードに対して HTTP ポート値を 0 に変更する必要があります。
4. [**HTTPS ポート**] フィールドで、値をポート番号に変更し、[**サブミット**] をクリックします。
5. My webMethods Server を再起動します。

6. この My webMethods Server と通信するすべての webMethods アプリケーションの管理者に、新しい HTTPS ポート番号を通知します。

My webMethods 管理者としての認証証明書の管理

My webMethods Administrator は、Integration Server またはその他の webMethods アプリケーションに接続するユーザの認証証明書を管理できます。認証証明書は、ユーザと My webMethods Server の間の接続を制御するものではありません。認証の割り当てを受けるには、ユーザがシステムディレクトリサービス、または My webMethods Server に接続している外部ディレクトリサービスのメンバーであることが必要です。ディレクトリサービスの使用の詳細については、114 ページの「ディレクトリサービスの管理」を参照してください。

ユーザの認証の割り当てに関しては、以下のルールが適用されます。

- 1 人のユーザに複数の認証を割り当てることができます。
- 認証の 1 つのインスタンスに割り当てられるユーザは 1 人だけですが、それぞれタイプが異なる認証の複数のインスタンスを追加して、各インスタンスに異なるユーザを割り当てることができます。

認証に関しては、以下のアクションを実行できます。

- 認証の追加
- 認証の検索
- 認証の詳細の表示
- 認証へのユーザの割り当て
- 認証のユーザの変更
- 認証の削除

認証証明書の追加

証明書を追加するには、以下の手順に従います。

認証証明書を追加するには

1. My webMethods で、[**ナビゲーション**] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [認証] に移動します。

2. [**新規認証の追加**] をクリックします。

3. [**参照**] をクリックし、追加する証明書ファイルの場所まで移動して、[**開く**] をクリックします。

4. [**認証タイプ**] リストで、Integration Server またはその他の webMethods アプリケーションに接続するクライアントが使用する認証証明書のタイプを選択します。

次の表に、My webMethods Server がサポートする認証のタイプとその目的を示します。

認証タイプ	目的
[SSL (デフォルト)]	メッセージの送信者を認証します。クレデンシャルはプロトコルヘッダーに含まれます。
[検証]	Integration Server の受信メッセージのデジタル署名を検証します。
[暗号化]	Integration Server の送信メッセージを暗号化します。
[検証および暗号化]	受信メッセージのデジタル署名の検証と送信メッセージの暗号化の両方を行います。ユーザのメッセージの送信と受信に同じ証明書が使用される場合に使用されます。
[メッセージ認証]	メッセージの送信者を認証します。クレデンシャルはメッセージヘッダーに含まれます。

5. [**アップロード**] をクリックします。

   [認証] パネルに証明書が表示されます。

認証証明書の検索

さまざまな条件に基づいて認証証明書を検索できます。

認証証明書を検索するには

1. My webMethods で、[**ナビゲーション**] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [検索] > [高度な検索] に移動します。

2. 証明書を検索するときに適用する条件を指定します。

次の表に、使用可能な検索条件と、高度な検索の設定方法を示します。

フィールド	説明
[認証情報]	
[タイプ]	証明書に割り当てられた証明書タイプを選択します。有効なタイプの詳細については、37 ページの「認証証明書の追加」を参照してください。デフォルトは [**いずれか**] です。
[発行元名]	証明書の発行元の共通名を入力します。空白のままにした場合、フィールドは使用されません。

フィールド	説明
[シリアル番号]	証明書に割り当てられたシリアル番号を入力します。空白のままにした場合、フィールドは使用されません。
[サブジェクト名]	サブジェクトの共通名を入力します。空白のままにした場合、フィールドは使用されません。
[有効開始日]	
[範囲]	提示された選択肢から日付の範囲を選択します。デフォルトは **[すべて]** です。
[開始日]	M/D/YYYY 形式を使用して開始日を入力します。形式が正しくない場合は、枠が赤に変わります。または、をクリックします。
[終了日]	M/D/YYYY 形式を使用して終了日を入力します。形式が正しくない場合は、枠が赤に変わります。または、をクリックします。
[有効最終日]	
[範囲]	提示された選択肢から日付の範囲を選択します。デフォルトは **[すべて]** です。
[開始日]	M/D/YYYY 形式を使用して開始日を入力します。形式が正しくない場合は、枠が赤に変わります。または、をクリックします。
[終了日]	M/D/YYYY 形式を使用して終了日を入力します。形式が正しくない場合は、枠が赤に変わります。または、をクリックします。

3. 検索条件を選択したら、**[検索]** をクリックします。

認証証明書の詳細情報の表示

認証証明書に関連する詳細情報を表示できます。

認証証明書の詳細情報を表示するには

1. My webMethods で、**[ナビゲーション]** > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [認証] に移動します。

2. **[認証]** パネルに証明書が表示されていない場合は、**[検索]** パネルを使用して証明書を検索します。詳細については、39 ページの「認証証明書の詳細情報の表示」を参照してください。

3. 証明書を見つけて、をクリックします。

   次の表に、各認証証明書で使用可能な詳細情報を示します。

証明書の詳細情報	説明
[タイプ]	証明書の追加時に割り当てられた認証タイプ。
[サブジェクト CN]	認証対象のホストの共通名。
[発行元 CN]	発行元の共通名。
[シリアル番号]	証明書に割り当てられたシリアル番号。
[有効開始日]	証明書が有効になる開始日。
[有効最終日]	証明書が有効である最終日。
[ユーザ]	ユーザの名前。

4. 証明書のリストに戻るには、[**閉じる**] をクリックします。

認証証明書へのユーザの割り当て

認証証明書の 1 つのインスタンスに割り当てられるユーザは 1 人だけです。ここでは、この手順を説明します。同じ証明書を複数のユーザに割り当てるには、ユーザごとに、別個の証明書インスタンスを追加します。証明書の追加方法の詳細については、37 ページの「認証証明書の追加」を参照してください。

ユーザを認証証明書に割り当てるには

1. My webMethods で、[**ナビゲーション**] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [認証] に移動します。

2. [**認証**] パネルに証明書が表示されていない場合は、[**検索**] パネルを使用して証明書を検索します。詳細については、38 ページの「認証証明書の検索」を参照してください。

3. 証明書を見つけて、 をクリックします。

4. [認証の編集] 領域で [**設定**] をクリックします。

5. [キーワード] フィールドにユーザ ID を入力し、[**検索**] をクリックし、[**選択済み**] ボックスにユーザを移動して、[**適用**] をクリックします。

6. [**閉じる**] をクリックします。

認証証明書のユーザの変更

既存の認証証明書のユーザを別のユーザに変更できます。

認証証明書のユーザを変更するには

1. My webMethods で、[**ナビゲーション**] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [認証] に移動します。

2. [**認証**] パネルに証明書が表示されていない場合は、[**検索**] パネルを使用して証明書を検索します。詳細については、38 ページの「認証証明書の検索」を参照してください。

3. 証明書を見つけて、 をクリックします。

4. [認証の編集] 領域で [**設定**] をクリックします。

5. [キーワード] フィールドに新規ユーザのユーザ ID を入力して、[**検索**] をクリックし、[**選択済み**] ボックスにユーザを移動して、[**適用**] をクリックします。

   このアクションにより、以前のユーザが新規ユーザで置き換えられます。

6. [**閉じる**] をクリックします。

認証証明書からのユーザの削除

既存の認証証明書に割り当てられているユーザを削除できます。

認証証明書からユーザを削除するには

1. My webMethods で、[**ナビゲーション**] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [認証] に移動します。

2. [**認証**] パネルに証明書が表示されていない場合は、[**検索**] パネルを使用して証明書を検索します。詳細については、38 ページの「認証証明書の検索」を参照してください。

3. 証明書を見つけて、 をクリックします。

4. [認証の編集] 領域で [**削除**] をクリックします。

5. [**閉じる**] をクリックします。

認証証明書の削除

認証証明書に関連する詳細情報を表示できます。

認証証明書の詳細情報を表示するには

1. My webMethods で、[**ナビゲーション**] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [認証] に移動します。

2. [**認証**] パネルに証明書が表示されていない場合は、[**検索**] パネルを使用して証明書を検索します。詳細については、38 ページの「認証証明書の検索」を参照してください。

3. 検索結果で、削除する証明書の横のチェックボックスをオンにして、[**削除**] をクリックします。

4 My webMethods Server での Web サーバの使用

- Web サーバとの統合 44
- My webMethods Server および Apache 44
- My webMethods Server および IIS 44

Web サーバとの統合

My webMethods Server は、Microsoft Internet Information Server や Apache HTTP Server などの主要な Web サーバに統合できます。分散展開シナリオで My webMethods Server をサードパーティ Web サーバと統合するための基本メカニズムでは、Web サーバにインストールおよび設定された小さなプラグインを使用する必要があります。このプラグインは、Web サーバから My webMethods Server に HTTP 要求を転送します。

My webMethods Server には、HTTP と HTTPS の両方をサポートする組み込み Web サーバである Jetty と統合されたサーブレットエンジンが用意されています。このため、独立した Web サーバ層は必ずしも必要ではありません。

外部 Web サーバ (または Web サーバのクラスタ) と共に My webMethods Server を設定する理由はいくつかあります。最も顕著な理由は、企業の IT ポリシーと手順に準拠するためです。My webMethods Server は柔軟な展開モデルをサポートしており、外部 Web サーバ (または Web サーバのクラスタ) では、個別に負荷分散できる HTTP 要求をすべて処理することが可能です。

外部 Web サーバを統合して HTTP 要求を処理するには、外部 Web サーバマシンで Web サーバプラグインを設定する必要があります。Web サーバプラグインは、多くの実稼動品質の Web サーバ製品で広範に使用されている Jakarta Web サーバのコードを利用します。

My webMethods Server および Apache

UNIX プラットフォームでは、Apache Software Foundation の Apache HTTP サーバとともに My webMethods Server を使用することができます。Apache HTTP サーバを使用する場合は、次の外部参照が役立つことがあります。

- Apache mod_proxy_http モジュールを使用する方法の詳細については、次を参照してください。http://httpd.apache.org/docs/2.2/mod/mod_proxy_http.html
- Apache mod_proxy モジュールを使用する方法の詳細については、次を参照してください。http://httpd.apache.org/docs/2.2/mod/mod_proxy.html
- Apache mod_proxy_balancer モジュールを使用する方法の詳細については、次を参照してください。http://httpd.apache.org/docs/2.2/mod/mod_proxy_balancer.html

My webMethods Server および IIS

Software AG では、My webMethods Server のインストールに Internet Information Services (IIS) コンポーネントを含めておらず、My webMethods Server でも Web サーバとしての IIS の使用を明示的にはサポートしていません。ただし、IIS を使用する場合は、次の外部参照が役立つことがあります。

- IIS を逆プロキシとして使用する方法の一般情報およびディスカッション: http://www.iis.net/
- Apache Tomcat で IIS を使用する方法: http://tomcat.apache.org/connectors-doc/webserver_howto/iis.html

5 Docker での My webMethods Server の使用

- My webMethods Server と Docker コンテナについて 46
- Docker コンテナでの My webMethods Server の実行 46
- My webMethods Server コンテナ使用時の考慮事項 46
- My webMethods Server インストールの Docker イメージの作成およびカスタマイズ .. 47
- My webMethods Server コンテナの実行およびカスタマイズ 50
- 最適化済み My webMethods Server コンテナの起動 62
- My webMethods Server コンテナと webMethods Microservices Runtime の併用 63
- My webMethods Server コンテナの監視 64
- My webMethods Server コンテナのクラスタリング 69

My webMethods Server と Docker コンテナについて

Docker は、アプリケーションを構築、展開および管理するための一連のツールおよびサービスです。Docker は、コンテナと呼ばれるアプリケーションパッケージを介して仮想化を提供します。コンテナによって、アプリケーションと、ライブラリ、設定ファイルおよびその他の依存関係を含む、そのランタイム環境全体が一緒にパッケージ化されます。コンテナは、事前定義済みイメージから作成されたインスタンスとして、Docker 環境で実行されます。また、My webMethods Server には、Docker で使用する My webMethods Server イメージを作成およびカスタマイズするための一連のツールと、個々のコンテナの設定を変更するためのさまざまなオプションが用意されています。

Docker コンテナでの My webMethods Server の実行

My webMethods Server インストールをコンテナ化する大まかな手順は、次のとおりです。

- 『Installing SAG products』の説明に従って、My webMethods Server と必要な階層化された製品をインストールします。
- My webMethods Server インスタンスを作成します。
- My webMethods Server ツールを実行して、インストールの Dockerfile を作成します。
- 生成された Dockerfile を使用して、インストールの Docker イメージを構築します。
- イメージからコンテナを実行します。

Docker コマンドラインを使用して My webMethods Server イメージから新しいコンテナを作成する場合、My webMethods Server で専用の起動スクリプトを使用して、コンテナ内でサーバインスタンスが起動されます。環境変数を使用して特定の設定を起動スクリプトに渡すことで、イメージからのインスタンスによって使用されるデータベースとは異なるデータベースを使用して My webMethods Server を起動したり、サーバログを異なる保存場所にリダイレクトしたりするなど、Docker イメージからの事前定義済み設定をオーバライドすることができます。My webMethods Server の環境変数の詳細については、次を参照してください: 50 ページの「My webMethods Server の環境変数について」

コンテナ内で起動する場合、My webMethods Server によって、データベース接続チェック、データベースの整合性チェック、ボリュームの一貫性チェックなどの一連の追加の検証チェックが実行されます。いずれかの検証チェックが失敗した場合、コンテナが終了します。サーバログを別の場所にリダイレクトしない限り、コンソール、または My webMethods Server 用に Docker によって作成されるデフォルトのボリュームで、ログを使用できます。デフォルトのボリュームの詳細については、次を参照してください: 54 ページの「My webMethods Server コンテナでのアセットとデータの管理」

My webMethods Server コンテナ使用時の考慮事項

Docker 19.03.0 以降での My webMethods Server は、Linux システムでサポートされます。サポートされる Linux システムの詳細については、Docker のマニュアルを参照してください。

Docker イメージを構築する前に、正常に設定されている My webMethods Server インストールからイメージとコンテナを作成して、すべての追加アプリケーションと階層化された製品をインストールすることをお勧めします。階層化された製品の Docker のサポートの詳細については、それぞれのマニュアルを参照してください。

単一インスタンスのインストールのみをコンテナ化することができます。コンテナのクラスタリングは、Docker 固有のアプローチと、Kubernetes などの追加のコンテナオーケストレーションテクノロジを使用して処理されます。

My webMethods Server のマニュアルでは、Docker と Docker の用語に慣れていることを前提としています。ネイティブの Docker の機能の詳細については、Docker のマニュアルを参照してください。

My webMethods Server のマニュアルでは、Docker イメージを作成するオンプレミスのインストールで、デフォルトのパスと場所を使用することを前提としています。

My webMethods Server インストールの Docker イメージの作成およびカスタマイズ

My webMethods Server には、インストールからの Dockerfile と Docker イメージの生成を容易にするための追加のツールが用意されています。`mws_container.sh` スクリプトは、My webMethods Server インストールの *`Software AG_directory`* `/MWS/tools/docker/bin` ディレクトリにあり、My webMethods Server の Dockerfile とイメージの作成、およびイメージリポジトリの操作のための一連のコマンドを含みます。

My webMethods Server インスタンスの Docker イメージを構築するための前提条件

Dockerfile を作成して My webMethods Server インスタンスの Docker イメージを構築するには、事前に次の操作を実行する必要があります。

- Docker をインストールして Docker デーモンを起動します。
- 同じマシンで、My webMethods Server とすべての必要な階層化された製品をインストールして、新しいサーバインスタンスを作成します。オプションで、インスタンスを初期化します。
- すべての必要な製品修正とカスタムアプリケーションをインストールします。

サーバインスタンスは、Software AG Installer または My webMethods Server コマンドラインユーティリティを使用して作成できます。

My webMethods Server インストールの Dockerfile の作成

`mws_container.sh` スクリプトには、My webMethods Server の単一インスタンスのインストールの Dockerfile を生成するための追加のコマンドラインツールが用意されています。

My webMethods Server の Dockerfile を生成するには

1. My webMethods Server インストールの *Software AG_directory* /MWS/tools/docker/bin ディレクトリに移動します。

2. mws_docker.sh createDockerfile を実行し、次の 1 つ以上のパラメータを指定して、生成された Dockerfile をカスタマイズします。

パラメータ	説明
-Dfile.path	オプション。生成された Dockerfile のファイルパス。完全パスまたはファイル名のみを指定します。デフォルト値は、ファイル名のみを指定した場合は *Software AG_directory* /MWS/tools/docker/scripts/*file.path*、指定しない場合は *Software AG_directory* /MWS/tools/docker/scripts/Dockerfile_MWS です。
-Dimage.name	オプション。My webMethods Server イメージを構築するための基盤となる基本イメージの名前。デフォルトは centos:7 です。
-Dcreate.user.group	オプション。ブーリアン型。イメージのユーザとグループを作成するかどうか。デフォルトは true です。
-Dbase.user.group	userid:groupid として指定します。オプション。-Dcreate.user.group が false に設定されているときに使用します。生成された Dockerfile の chown ステートメントに指定するユーザとグループを指定し、イメージのローカルのインストールからファイル/ディレクトリをコピーするときに使用します。指定したユーザとグループが存在するようにする必要があります。つまり、基本イメージまたは追加のスクリプトによってユーザ/グループが作成される必要があります。
-Dinstance.name	オプション。イメージを作成するための My webMethods Server インスタンスの名前。デフォルトは default です。
-Dinclude.jdk	オプション。ブーリアン型。イメージのホストインストールから JDK を含めるかどうか。デフォルトは「true」です。
-Dimage.java.home	オプション。Java のホームディレクトリへの完全パス。-Dinclude.jdk が false に設定されているときにのみ適用されます。
-Dport.list	オプション。Docker コンテナで公開するポートのカンマ区切りリスト。デフォルトは 8585 です。
-Dcreate.volumes	オプション。ブーリアン型。ボリュームを設定するかどうか。デフォルトは「true」です。
-Dcreate.dockerignore	オプション。ブーリアン型。.dockerignore ファイルを生成するかどうか。デフォルトは「true」です。

My webMethods Server インストールからの Docker イメージの構築

mws_container.sh スクリプトには、docker build コマンドを使用して、提供された Dockerfile から My webMethods Server の単一インスタンスのインストールのイメージを作成するための追加のコマンドラインツールが用意されています。

My webMethods Server の Docker イメージを構築するには

1. My webMethods Server インストールの *Software AG_directory* /MWS/tools/docker/bin ディレクトリに移動します。

2. mws_docker.sh build を実行し、次の 1 つ以上のパラメータを指定して、イメージの生成をカスタマイズします。

パラメータ	説明
-Duse.dockerignore	オプション。ブーリアン型。.dockerignore ファイルを使用するかどうかを指定します。デフォルトは「true」です。
-Dfile.path	オプション。構築に使用する Dockerfile のファイルパス。完全パスまたはファイル名のみを指定します。デフォルト値は、ファイル名のみを指定した場合は *Software AG_directory* /MWS/tools/docker/scripts/*file.path*、指定しない場合は *Software AG_directory* /MWS/tools/docker/scripts/Dockerfile_MWS です。
-Dimage.name	オプション。生成される My webMethods Server イメージの名前。デフォルトは mws-image です。
-Dserver	オプション。Docker レジストリの URL。指定しない場合、ローカルの Docker レジストリが使用されます。
-Duser	オプション。Docker レジストリに対して認証するユーザのユーザ名。-Dserver が指定されており、レジストリで認証が必要になる場合は必須です。
-Dpassword	オプション。Docker レジストリにログオンするユーザのパスワード。-Dserver が指定されており、レジストリで認証が必要になる場合は必須です。プレーンテキストのパスワード、またはパスワードを含むテキストファイルの完全パス (-Dpassword=file://*full_file_path* など) を指定します。

その他の Docker イメージのコマンド

mws_container.sh スクリプトには、My webMethods Server の Docker イメージを操作するための次の追加のコマンドが用意されています。

saveImage

ローカルの Docker レジストリから zip ファイルにイメージを保存します。このコマンドは、次のパラメータを受け入れます。

- `-Dimage.name`: 必須。保存するイメージの名前。
- `-Dfile.path`: オプション。イメージのファイルパス。デフォルト値は、ファイル名のみを指定した場合は *`Software AG_directory`* `/MWS/tools/docker/images/file.path`、指定しない場合は *`Software AG_directory`* `/MWS/tools/docker/images/`*`image.name`*`.zip` です。

loadImage

アーカイブからローカルの Docker レジストリにイメージをロードします。このコマンドは、次のパラメータを受け入れます。

- `-Dfile.path`: 必須。ロードするイメージのファイルパス。絶対パスまたはファイル名のみを指定します。ファイル名のみを指定した場合、イメージは *`Software AG_directory`* `/MWS/tools/docker/images/file.path` からロードされます。

pushImage

My webMethods Server イメージをリモートの Docker レジストリにプッシュします。このコマンドは、次のパラメータを受け入れます。

- `-Dimage.name`: 必須。プッシュする My webMethods Server イメージの名前。
- `-Dimage.tag`: オプション。Docker レジストリにプッシュする際にイメージに使用するタグ名。デフォルトは *image.name* です。
- `-Dserver`: 必須。Docker レジストリの URL。
- `-Drepository.name`: 必須。イメージのプッシュ先のレジストリの名前。
- `-Duser`: 必須。Docker レジストリに対して認証するユーザのユーザ名。
- `-Dpassword`: 必須。Docker レジストリにログオンするユーザのパスワード。プレーンテキストのパスワード、またはパスワードを含むテキストファイルの完全パス (`-Dpassword=file://`*`full_file_path`* など) を指定します。

My webMethods Server コンテナの実行およびカスタマイズ

My webMethods Server の環境変数について

My webMethods Server イメージから新しいコンテナを起動すると、環境変数を使用してイメージのデフォルト設定をオーバライドできます。My webMethods Server コンテナの環境変数は、

mws new コマンドの対応するパラメータと一致します。つまり、コンテナ固有の設定オプションを提供します。

My webMethods Server 環境変数は、docker run コマンドの -e/--env パラメータを使用するか、ファイルを使用して指定します。詳細については、Docker のマニュアルを参照してください。

My webMethods Server コンテナでサポートされる環境変数

My webMethods Server は、既存の My webMethods Server イメージのデフォルト設定をオーバライドするために、次の環境変数を公開します。

変数	説明
DB_TYPE	サーバインスタンスが使用するデータベースのタイプ。有効な値は次のとおりです。 ■ ms - Microsoft SQL Server ■ oracle: Oracle ■ db2: DB2 ■ mysqlee - MySQL Enterprise Edition ■ mysqlce - MySQL Community Edition ■ postgresql - PostgreSQL
DB_URL	データベースのタイプおよびドライバに基づく、データベース接続URL。Docker コマンドラインから変数を指定するときは、データベース URL を二重引用符で囲みます。
DB_USERNAME	My webMethods Server データベースへの接続時に使用するユーザ名。
DB_PASSWORD	My webMethods Server データベースユーザのパスワード。
NODE_NAME	サーバインスタンスをホストするクラスタノードのカスタム名。
APPS_DIR	My webMethods Server インスタンスの MWS/deploy ディレクトリにコピーされて、コンテナの起動時にインストールされるカスタムのアセットおよびアプリケーションを含む、コンテナファイルシステムのディレクトリ。指定しない場合、My webMethods Server では、コンテナのデフォルトの SAGHOME/MWS/volumes/apps ディレクトリが使用されます。アセットおよびアプリケーションは、バインドマウントされたホストディレクトリまたは外部ボリュームを介して提供され、コンテナの apps ディレクトリの場所にマウントされます。有効なアセット形式は war、ｃｄｐ、pdp、jar です。My webMethods Server の Docker コンテナへのアプリケーションの展開の詳細については、次を参照してください: 56 ページの「My webMethods Server コンテナのカスタムアプリケーション」

変数	説明
CONFIGS_DIR	コンテナの起動時に My webMethods Server によってロードされるその他の設定ファイルを含む、コンテナファイルシステムのディレクトリ。指定しない場合、My webMethods Server のコンテナ起動スクリプトでは、デフォルトの SAGHOME/MWS/volumes/configs が使用されて、コンテナファイルシステムの SAGHOME/MWS/volumes/configs ディレクトリにマウントされたボリュームの configs ディレクトリが確認されます (このようなボリュームが使用可能な場合)。 マウントされたボリュームの configs ディレクトリには、次の 1 つ以上のサブディレクトリを含めることができます。 ■ assets_cfg - xmlImport ファイルと、My webMethods Server スキンを指定するためのものです。 ■ cluster_config - cluster.xml ファイルなどの My webMethods Server クラスタのカスタム設定ファイルを保存します。 ■ instance_cfg - mws.db.xml や server.properties ファイルなどの My webMethods Server インスタンスのカスタム設定ファイルを保存します。 ■ jvm_cfg - カスタムの JVM 設定ファイルと証明書を指定するためのものです。 ■ profile_cfg - custom_wrapper.conf ファイルなどの My webMethods Server の OSGi プロファイルのカスタム設定ファイルを保存します。 外部ボリュームを介した My webMethods Server の Docker コンテナの設定の変更の詳細については、次を参照してください: 59 ページの「コンテナの設定の変更」
DATA_DIR	My webMethods Server が、展開されたアプリケーションに関する検索インデックスや情報などのランタイムデータを保存し、Task Engine イベントキューからのイベントを保持する、コンテナファイルシステムのディレクトリ。指定しない場合、My webMethods Server では、コンテナファイルシステムのデフォルトの SAGHOME/MWS/volumes/data ディレクトリが使用されます。
LIBS_DIR	My webMethods Server インスタンスの MWS/lib ディレクトリにコピーされてコンテナの起動時に My webMethods Server によってロードされるサードパーティのライブラリまたはその他のカスタム Jar ファイルを保持する、コンテナファイルシステムのディレクトリ。指定しない場合、My webMethods Server のコンテナ起動スクリプトでは、コンテナファイルシステムのデフォルトのディレクトリである SAGHOME/MWS/volumes/libs が使用されます。ライブラリおよび Jar は、バインドマウントされたホストディレクトリまたは外部ボリュームを介して提供されます。My webMethods Server の Docker コンテナでのサードパーティのライブラリおよびカスタム Jar の使用の詳細については、

変数	説明
	58 ページの「My webMethods Server コンテナでの外部ライブラリの使用」を参照してください。
LOGS_DIR	My webMethods Server が、My webMethods Server インスタンスによって生成されたすべてのログファイルを保持する、コンテナファイルシステムのディレクトリ。デフォルトでは、このディレクトリには次のサブディレクトリが含まれます。 ■ instance_logs - オンプレミスの My webMethods Server インストールのデフォルトの SAGDir/MWS/server/instanceName/logs ディレクトリに対応します。 ■ profile_logs - オンプレミスの My webMethods Server インストールの SAGDir/profiles/MWS_default/logs ディレクトリに対応します。 ■ cli_logs - オンプレミスのインストールの SAGDir/MWS/bin ディレクトリに保存されるすべてのログが対象です。外部ライブラリを追加するときなどの操作中に My webMethods Server によって実行される OSGi プロファイル関連の操作のログは、cli_logs/archive に保存されます。 指定しない場合、My webMethods Server では、コンテナファイルシステムのデフォルトのディレクトリである SAGHOME/MWS/volumes/logs が使用されます。

例

- 次の例では、myMWSImage イメージに基づく新しい My webMethods Server コンテナを起動し、My webMethods Server インスタンスに対応するクラスタノードのクラスタノード名を指定します。

```
docker run --env NODE_NAME=myClusterNode myMWSImage
```

- 次の例では、myMWSImage イメージに基づく新しい My webMethods Server コンテナを起動し、イメージのデフォルトのデータベース接続パラメータをオーバライドします。

```
docker run --env DB_TYPE=mysqlee --env
DB_URL="jdbc:mysql://host:port;databaseName=wmdb;relaxAutoCommit=true"
--env DB_USERNAME=wmuser DB_PASSWORD=wmpass myMWSImage
```

  起動時に、My webMethods Server インスタンスは、myMWSImage イメージのデフォルトのデータベース設定に関係なく、クレデンシャル wmuser/wmpass を使用して、指定された *host* および *port* で、wmdb という名前の MySQL Enterprise Edition データベースに接続します。

- 次の例では、mws イメージに基づく新しい My webMethods Server コンテナを起動し、イメージのデフォルトのデータベース接続パラメータをオーバライドして、デフォルトの My webMethods Server 接続ポートを、公開されたコンテナポートにマップします。

```
docker run -e DB_URL="jdbc:wm:sqlserver://mwssql2016:1433;databaseName=mwsDB"
```

```
-e DB_USERNAME=wmuser -e DB_PASSWORD=wmpass -e DB_TYPE=ms -p 8585:8585 mws
```

起動時に、My webMethods Server インスタンスは、mws イメージのデフォルトのデータベース設定に関係なく、クレデンシャル wmuser/wmpass を使用して、ホスト mwssql2016 およびポート 1433 で、mwsDB という名前の MS SQL Server データベースに接続します。My webMethods Server の接続ポート 8585 は、コンテナホストマシンの同じ番号のポートにマップされます。

- 次の例では、myMWSImage に基づく My webMethods Server コンテナのログファイル保存場所を、コンテナファイルシステムの myLogs ディレクトリにリダイレクトし、ホストのマウントされたディレクトリ MWSLogs ですべてのログを保持します。

```
docker run -v /home/myDirs/MWSLogs:/volumes/myVolume/myLogs --env
LOGS_DIR=/volumes/myVolume/myLogs myMWSImage
```

My webMethods Server コンテナでのアセットとデータの管理

デフォルトでは、My webMethods Server コンテナは、My webMethods Server イメージの Dockerfile で宣言されたコンテナの SAGHOME/MWS/volumes ディレクトリにデータを保持します。名前付きボリュームをマップしない限り、Docker デーモンによって、MWS/volumes ディレクトリが、ホストファイルシステムのデフォルトの場所である var/lib/docker/volumes の匿名ボリューム (つまり、ランダムな名前のボリューム) にマウントされます。アーカイブ、ナビゲーション、および異なるコンテナ間のデータ共有を容易にするために、名前付きボリュームを使用できます。

My webMethods Server コンテナは空でないボリュームに接続できますが、My webMethods Server コンテナを別のコンテナのデータが既に含まれているボリュームに接続すると、予想のできない結果が生じることがあります。コンテナの起動時に、My webMethods Server によってボリュームの内容が検証され、検証が失敗した場合はコンテナが停止されます。次の表に、My webMethods Server の Dockerfile が宣言するボリュームと、コンテナ間のボリュームの再利用に適用される制限を示します。

ボリューム	環境変数	説明
MWS/volumes/apps	$APPS_DIR	起動時に My webMethods Server インスタンスに展開するカスタムアプリケーションを保存します。読み取りモード。異なるコンテナ間で再利用できます。
MWS/volumes/configs	$CONFIGS_DIR	起動時に My webMethods Server インスタンスに指定する設定ファイルを保存します。読み取りモード。異なるコンテナ間で再利用できます。
MWS/volumes/data	$DATA_DIR	My webMethods Server の検索インデックスや Task Engine ライトウェイトイベントキューからのイベントなどの、コンテナランタイムのデータを保存します。読み取りおよび書き込みモード。このボリュームは、異なるコンテナ間で再利用できません。

ボリューム	環境変数	説明
MWS/volumes/libs	$LIBS_DIR	起動時に MWS/lib ディレクトリに挿入する外部ライブラリとカスタム Jar を保存します。読み取りモード。異なるコンテナ間で再利用できます。
MWS/volumes/logs	$LOGS_DIR	コンテナランタイムのログを保存します。読み取りおよび書き込みモード。このボリュームは、異なるコンテナ間で再利用できません。
MWS/volumes/patches	$PATCHES_DIR	コンテナの起動時に My webMethods Server インスタンスにインストールするテストパッチを保存します。読み取りモード。異なるコンテナ間で再利用できます。

My webMethods Server コンテナのデフォルトのアセットとデータの場所の変更の詳細については、次を参照してください: 51 ページの「My webMethods Server コンテナでサポートされる環境変数」

ボリュームとボリュームの検証について

My webMethods Server では、My webMethods Server の Docker イメージから新しいコンテナを起動したときに、データベースの接続と整合性のチェックに加えて、ボリュームの検証も実行されます。ボリュームの検証チェックでは、次のことが確認されます。

- バインドマウントされたホストディレクトリへの指定されたパスが存在する
- 指定された名前付きボリュームとホストディレクトリに、適切な読み取り/書き込み権限がある
- コンテナで共有できるボリュームとディレクトリのみにデータが事前移入されている

いずれかのボリュームの検証チェックが失敗した場合、コンテナが終了します。コンソールとデフォルトの (名前が付けられていない) Docker ボリュームで、ログを使用できます。ボリュームと、コンテナ間のデータ共有の詳細については、54 ページの「My webMethods Server コンテナでのアセットとデータの管理」を参照してください。

My webMethods Server コンポーネントとカスタムアプリケーションのインストール

オンプレミスの My webMethods Server のインストールと同様に、[インストール管理] ポートレットを使用して、Docker コンテナで実行されているインスタンスに My webMethods Server コンポーネントをインストールできます。My webMethods Server により、コンテナが作成される Docker イメージに加えてインストールするすべてのサーバコンポーネントが、■ アイコンでマークされます。

[インストール管理] ポートレットを使用して、イメージに加えてインストールされたサーバコンポーネントを削除することもできます。ただし、カスタムアプリケーションを参照して展開することはできません。カスタムアプリケーションを My webMethods Server コンテナに展開する

には、56 ページの「My webMethods Server コンテナのカスタムアプリケーション」を参照してください。

[インストール管理] ポートレットの一般情報については、336 ページの「ポートレットまたはその他の展開可能なサーバコンポーネントのインストール」を参照してください。

My webMethods Server コンテナのカスタムアプリケーション

My webMethods Server インストールの Docker イメージの生成前にすべての必要なカスタムアプリケーションを含めるという一般的な推奨事項があるにもかかわらず、特定の状況、特に実稼動環境では、コンテナを起動するイメージに加えて、カスタムアプリケーションをコンテナにインストールすることが必要になる場合があります。My webMethods Server コンテナは、コンテナファイルシステムの事前定義済みディレクトリである `/MWS/volumes/apps` を含みます。ここに、My webMethods Server インスタンスの `/deploy` ディレクトリにコピーするアセットを配置できます。次のいずれかの方法で、コンテナにカスタムアプリケーションをインストールできます。

- `docker cp` コマンドの使用 - ホストマシンから実行中のコンテナの `/MWS/volumes/apps` ディレクトリにアプリケーションパッケージをコピーします。この方法では、アプリケーションがホット展開され、コンテナを再起動する必要がありません。ただし、アプリケーションに外部ライブラリへの依存関係が含まれる場合、この方法を使用することはできません。`docker cp` コマンドを使用してホット展開するアプリケーションは、削除することもアンインストールすることもできません。
- ホストファイルシステムからバインドマウントされたディレクトリの使用 - `docker run` コマンドの `-v` オプションを使用して、新しいコンテナの起動時に展開する 1 つ以上のアプリケーションが含まれるホストから、ディレクトリをバインドマウントします。コンテナの起動時に、My webMethods Server によってすべてのファイルがホストディレクトリからコンテナの `/MWS/volumes/apps` にコピーされ、それ以降はインスタンスの `/deploy` ディレクトリにコピーされます。アプリケーションを削除するには、マウントされたディレクトリからそのアプリケーションを削除して、コンテナを再起動します。
- 名前付きボリュームの使用 - `docker run` コマンドの `-v` オプションを使用して、新しいコンテナの起動時に展開する 1 つ以上のアプリケーションが含まれる名前付きボリュームをマウントします。コンテナの起動時に、My webMethods Server によってすべてのファイルがボリュームからコンテナの `/MWS/volumes/apps` にコピーされ、それ以降はインスタンスの `/deploy` ディレクトリにコピーされます。アプリケーションを削除するには、ボリュームからそのアプリケーションを削除して、コンテナを再起動します。

My webMethods Server では、コンテナとそのマウントされたボリューム (デフォルトでは匿名) の `MWS/volumes/data` ディレクトリの `.properties` ファイルに、すべての展開されたアプリケーションのリストが保持されます。

My webMethods Server の環境変数を使用して、`/apps` と `/data` の両方のディレクトリのコンテナの場所をオーバライドできます。詳細については、51 ページの「My webMethods Server コンテナでサポートされる環境変数」を参照してください。

例

- 次の例では、名前付きボリューム tasks を作成し、それを使用して、webM107 イメージから作成されたコンテナにタスクアプリケーションを展開します。

```
docker volume create tasks
```

```
docker run -e
DB_URL="jdbc:wm:sqlserver://dbhost:1439;databaseName=MWSDOCKER;user=wmuser;password=manage"
-e DB_TYPE=ms -v tasks:/opt/softwareag/MWS/volumes/apps -p 8585:8585 webM107
```

注:
docker run コマンドを発行する前に、カスタムのタスクアプリケーションがホストからボリュームにコピーされます。

- 次の例では、ホストディレクトリ /home/myvm/taskapps をバインドマウントして、webM107 イメージから作成されたコンテナにタスクアプリケーションを展開します。

```
docker run -e
DB_URL="jdbc:wm:sqlserver://dbhost:1439;databaseName=MWSDOCKER;user=wmuser;password=manage"
-e DB_TYPE=ms -v
/home/myvm/taskapps:/opt/softwareag/MWS/volumes/apps -p 8585:8585
webM107
```

注:
カスタムのタスクアプリケーションがバインドマウントされたディレクトリに存在する必要があります。

- 次の例では、ホストディレクトリ /home/myvm/taskapps をバインドマウントして、webM107 イメージから作成されたコンテナにタスクアプリケーションを展開し、デフォルトの /apps ディレクトリではなく、コンテナの /tasks ディレクトリにすべてのアプリケーションをコピーします。

```
docker run -e
DB_URL="jdbc:wm:sqlserver://dbhost:1439;databaseName=MWSDOCKER;user=wmuser;password=manage"
-e DB_TYPE=ms -v /home/myvm/taskapps:/opt/softwareag/MWS/tasks -e
APPS_DIR=/opt/softwareag/MWS/tasks -p 8585:8585
webM107
```

注:
カスタムのタスクアプリケーションがバインドマウントされたディレクトリに存在する必要があります。コンテナの起動時に、APPS_DIR 環境変数によって参照される MWS/tasks ディレクトリが作成されます。

xmlImport ファイルを使用したアセットの展開

My webMethods Server コンテナは、コンテナファイルシステム上の専用の場所である /MWS/volumes/configs/assets_cfg を使用できます。この場所に、外部ボリュームをマウントし、コンテナの起動時にユーザインタフェースエレメント、フォルダ、ユーザなどの My webMethods Server アセットを作成および変更する xmlImport ファイルを指定することができます。

/MWS/volumes/configs/assets_cfg を使用して xmlImport ファイルを展開するには、外部ボリュームを MWS/volumes/configs ディレクトリにマウントして、そのボリュームに assets_cfg という名前のディレクトリが含まれる必要があります。外部ボリュームをマウントしない場合、またはボリュームに assets_cfg ディレクトリが含まれない場合は、Docker デーモンによってコンテナファイルシステムに /MWS/volumes/configs/assets_cfg サブディレクトリが作成されることはありません。

保管可能な記憶領域へのコンテナログのリダイレクト

My webMethods Server イメージから新しいコンテナを作成するときに、My webMethods Server の起動スクリプトによって、wrapper.log が、コンテナの標準出力ストリーム (STDOUT) にマップされます。ランタイムで、My webMethods Server インスタンスによって、コンテナファイルシステムのデフォルトのコンテナログディレクトリである MWS/volumes/logs ディレクトリにすべてのログが保存されます。デフォルトでは、このディレクトリは、Docker デーモンによってランダムな名前を使用して作成された匿名ボリュームにマップされます。コンテナの停止時または開始時に、前回の実行のログファイルが匿名ボリュームから MWS/volumes/logs ディレクトリに戻ります。ホストファイルシステムのディレクトリをマップしたり、名前付きボリュームを作成することで、保管可能な場所にファイルを保持して、監査目的でログデータを保持することもできます。

さらに、LOGS_DIR 環境変数を使用して、コンテナファイルシステムのログの場所を変更することができます。詳細については、51 ページの「My webMethods Server コンテナでサポートされる環境変数」を参照してください。

My webMethods Server コンテナでの外部ライブラリの使用

新しいコンテナの起動時に、サードパーティのライブラリ、カスタム Jar ファイルおよびフラグメントバンドルを My webMethods Server に追加できます。デフォルトでは、My webMethods Server コンテナには、外部ボリュームをマウントできるディレクトリ SAGHOME/MWS/volumes/libs が用意されています。新しい My webMethods Server コンテナの作成時に、My webMethods Server の起動スクリプトによって、マウントされたボリュームにあるすべてのサードパーティのライブラリとカスタム Jar ファイルが、My webMethods Server による初期化の際にロードされる MWS/lib ディレクトリにコピーされます。

コンテナの起動時に、LIBS_DIR 環境変数を使用して、コンテナ内のデフォルトの場所である SAGHOME/MWS/volumes/libs ディレクトリをオーバライドできます。変数によって参照されるディレクトリがコンテナファイルシステムに存在しない場合、Docker デーモンによってディレクトリが作成されます。

サードパーティのライブラリのマニフェストをカスタマイズするか、Jar ファイルを別のバンドルへのフラグメントとして追加するには、351 ページの「カスタム JAR ファイルの追加」の説明に従ってバインドの指示を指定し、Jar ファイルと一緒にバインドファイルをボリュームに配置します。

例

次の例では、myMWSImage から新しいコンテナを起動し、インスタンスによる初期化の際に My webMethods Server によってロードされる外部ライブラリのソースとなるコンテナファイルシ

ステムのデフォルトの SAGHOME/MWS/volumes/libs ディレクトリに外部ボリュームをマウントします。

```
docker run -v var/lib/docker/volumes/myMWSVolume/test/extlibs:SAGHOME/MWS/volumes/libs
 -p 8585:8585 myMWSImage
```

次の例では、myMWSImage から新しいコンテナを起動し、コンテナファイルシステムの opt/softwareag/MWS/myLibsDir ディレクトリに外部ボリュームをマウントし、My webMethods Server によってロードされる外部ライブラリのソースとして、デフォルトの SAGHOME/MWS/volumes/libs ディレクトリを opt/softwareag/MWS/myLibsDir ディレクトリにオーバライドします。

```
docker run -v
var/lib/docker/volumes/myMWSVolume/test/extlibs:opt/softwareag/MWS/myLibsDir --env
LIBS_DIR=opt/softwareag/MWS/myLibsDir -p 8585:8585 myMWSImage
```

両方の例で、マップされたボリュームからのすべてのライブラリとカスタム Jar が MWS/lib ディレクトリにコピーされて、起動時に My webMethods Server によってロードされます。

コンテナの設定の変更

コンテナファイルシステムの /MWS/volumes/configs ディレクトリは、My webMethods Server のアセットの展開に使用可能な assets_cfg サブディレクトリに加え、起動時に My webMethods Server をロードするために必要な様々な設定ファイルを提供する専用サブディレクトリを含むことができます。

/MWS/volumes/configs/ ディレクトリで設定ファイルを提供するには、外部ボリュームをそのディレクトリにマウントする必要があります。また、ボリュームには次のサブディレクトリを 1 つ以上含めます。

- cluster_cfg - cluster.xml、systemProperties.propertes などのカスタムの設定ファイルを提供します。クラスタデータベースを初期化する My webMethods Server は、コンテナのブートストラップ時に、My webMethods Server のクラスタ内のすべてのノードで、cluster_cfg ディレクトリのファイルから設定を適用します。
- instance_cfg - カスタムの cache.xml、mws.db.xml、server.properties、websso.properties、log4j.init.properties、およびその他のインスタンス設定ファイルを提供します。
- jvm_cfg - セキュリティ認証などの様々な JVM/JRE の追加機能をコンテナの JRE に提供します。
- profile_cfg - カスタムのプロファイル設定を提供します。たとえば、custom_wrapper.conf ファイル内の JVM プロパティや jaas.config ファイル内のセキュリティ設定などです。

Docker デーモンは、マウント済みのボリュームに存在する *_cfg フォルダに対してのみ、対応する *_cfg サブディレクトリをコンテナファイルシステムに作成します (マウント済みボリュームが存在する場合)。

My webMethods Server コンテナの環境変数による設定は、ボリュームが提供する設定ファイルに対して常に優先権を持ち、設定ファイル内の該当する値を置き換えます。

注:
cluster.xml または custom_wrapper.conf files といった My webMethods Server の設定ファイルを変更し、新規コンテナを既存のイメージから起動するときにこれらのファイルの新バージョンをボリュームから提供することは可能ですが、Software AG は、オンプレミスインストールから Docker イメージファイルを生成する前に、My webMethods Server の設定を完了することを推奨しています。

コンテナ設定の例

例: My webMethods Server コンテナのデータベース接続再試行の設定

My webMethods Server コンテナのデータベース接続再試行を設定するには、外部ボリュームから新規の mws.db.xml ファイルを指定します。

1. mws.db.xml をオンプレミスインストールまたは My webMethods Server コンテナからコピーします。
2. ファイルを変更して、データベース接続再試行に必要な設定を指定します。詳細については、221 ページの「データベース接続再試行の変更」を参照してください。コンテナの起動時に正しい設定が指定されるように、必要に応じてデータベース接続 URL やユーザパスワードなどのパラメータを変更します。
3. 次の例に従い、my_mws_config/instance_cfg ディレクトリにある mws.db.xml ファイルをホストファイルシステムに保存し、起動時にコンテナの *configs* ディレクトリにマウントします。

```
docker run
-v /home/myvm/my_mws_config:/opt/softwareag/MWS/volumes/configs
-p 8586:8585 webm107
```

My webMethods Server コンテナの設定を環境変数で上書きしないかぎり、My webMethods Server は、マウントされたホストディレクトリの mws.db.xml ファイルで定義されたすべての設定を使用します。Docker CLI を使って指定した環境変数は mws.db.xml 内の対応する値をマップされたディレクトリから上書きします。

例: My webMethods Server コンテナへのセキュリティ認証の追加

新規 My webMethods Server コンテナを起動するときに、コンテナの JRE に新規セキュリティ認証を追加することができます。セキュリティ認証を追加するための高レベルの手順は次のとおりです。

1. cacerts ファイルを変更し、新規認証を持つオンプレミスインストールまたは新規認証ファイルにアクセスできるコンテナで、keytool コマンドを使って新規認証を追加します。次に、更新された cacerts ファイルを抽出します。
2. ホストファイルシステムで、*my_mws_config*/jvm_cfg/lib/security/ のパターンに一致する専用ディレクトリを作成します。次に、更新された cacerts ファイルをそのディレクトリに保存します。

3. 次のように、コンテナファイルシステム上の configs ディレクトリにマップされた *my_mws_config* ディレクトリのコンテナを実行します。

```
docker run
-e
DB_URL="jdbc:wm:sqlserver://dbhost:1439;databaseName=MWSDOCKER;user=wmuser;password=manage"
-e DB_TYPE=ms
-v /home/myvm/my_mws_config:/opt/softwareag/MWS/volumes/configs
-p 8585:8585 webM107
```

上記のサンプルコマンドは、webM107 イメージから My webMethods Server コンテナを作成します。このコンテナは MWSDOCKER という名前の MS SQL サーバデータベースを使用し、ホストファイルシステムの /home/myvm/*my_mws_config*/jvm_config/lib/security ディレクトリで提供される追加のセキュリティ認証を持ちます。

例: My webMethods Server コンテナのシングルサインオンの設定

Docker コンテナ内で実行中であり、システム全体の CA 認証ストアである cacerts を SSO キーストアとして使用する My webMethods Server インスタンスに、サードパーティの ID プロバイダのシングルサインオンを設定する高レベルの手順の例を次に示します。以下の説明は、Docker ネットワークに詳しく、Docker 環境を設定した経験がある方を対象としています。コンテナネットワークの詳細については、Docker のマニュアルを参照してください。

1. シングルサインオンを使った My webMethods Server コンテナの実行に使用する予定のイメージから、コンテナを起動します。

   注:
   これは、My webMethods Server の設定ファイルを取得し変更するために必要な手順です。しかし、My webMethods Server コンテナを起動すると My webMethods Server データベースが初期化されるため、別のデータベースインスタンスでコンテナを起動するか、docker run コマンドを使ってコンテナのエントリポイントを変更する必要があります。コンテナのエントリポイントの詳細については、Docker のマニュアルを参照してください。

2. 実行中のコンテナの /opt/softwareag/MWS/bin ディレクトリに移動し、sh mws.sh getconfig *fileName* を実行して cluster.xml および websso.properties ファイルを取得します。

3. cluster.xml ファイルを変更して My webMethods Server の HTTPS ポートを表示し、ホストファイルシステムの *my_mws_config*/cluster_cfg/ ディレクトリに保存します。

4. cacerts ファイルを keytool コマンドで変更し、外部 IDP のセキュリティ認証を追加します。次に、更新された cacerts ファイルを抽出します。コンテナが新規認証ファイルへのアクセスを持たない場合、新規認証を持つオンプレミスインストールで事前にこの手順を実行してください。

5. ホストファイルシステムで、*my_mws_config*/jvm_cfg/lib/security/ のパターンに一致する専用ディレクトリを作成します。次に、更新した cacerts ファイルをそのディレクトリに保存します。

6. websso.properties ファイルで必要なプロパティを設定し、ホストファイルシステムの *my_mws_config*/instance_cfg/ ディレクトリに保存します。詳細については、287 ページの「websso.properties ファイルでのプロパティの設定」を参照してください。

ヒント:
クラスタ全体に設定を適用したい場合は、`websso.properties` ファイルを *my_mws_config*`/cluster_cfg/` ディレクトリに保存します。

7. 次のように、コンテナを実行して、カスタムの設定ファイルを持つすべてのディレクトリを含む *my_mws_config* 親ディレクトリを、コンテナファイルシステムの `configs` ディレクトリにマップします。

```
docker run
-e
DB_URL="jdbc:wm:sqlserver://dbhost:1439;databaseName=MWSDOCKER;user=wmuser;password=manage"
-e DB_TYPE=ms
-v /home/myvm/my_mws_config:/opt/softwareag/MWS/volumes/configs
-p 8585:8585 webM107
```

   My webMethods Server は起動時に `SPMetadata.xml` および `IDPMetadata.xml` ファイルをコンテナ内の `/opt/softwareag/MWS/server/`*instanceName*`/config/` ディレクトリに生成します。

8. 上記の手順で生成された xml ファイルを使って、My webMethods Server を外部 ID プロバイダに登録して設定を完了します。詳細については、286 ページの「サードパーティ ID プロバイダを使用するためのシングルサインオンの設定」を参照してください。

9. My webMethods Server コンテナを再起動します。

最適化済み My webMethods Server コンテナの起動

My webMethods Server は、コンテナ内での初期化時に、コンテナの起動時間を自動的に最適化するメカニズムを使用します。この最適化メカニズムは、My webMethods Server と実稼働環境データベース間のネットワーク遅延により、インスタンスの初期化の実行に必要な各 SQL ステートメントの実行時間が平均 20 ミリ秒を越えると推定される場合にトリガされます。また、このメカニズムは、Derby データベースのメモリ内インスタンスを使用し、すべての初期ブートストラップアクティビティを Derby インスタンスに対して実行します。続いて、すべての必要なデータが実稼働データベースに転送されます。実稼働データベースの種類に関わらず、起動クエリの実行時間が指定されたしきい値を超えると推定されるすべてのコンテナに対して、このメカニズムが自動的に適用されます。

次の JVM プロパティで、My webMethods Server が起動時の最適化をトリガする SQL の実行時間のしきい値や、最適化メカニズムをトリガするかどうかを設定することができます。

- `Mws.inmemory.init`:: My webMethods Server のコンテナが起動時の最適化メカニズムを使用するかどうかを制御します。デフォルトでは、このプロパティは `custom_wrapper.conf` ファイルに存在せず、My webMethods Server はデータベースクエリのサンプルセットを実行し、クエリごとの平均実行時間が 20 ミリ秒を超える場合にのみ、最適化メカニズムを使用して初期化します。最適化メカニズムを常に使用するには「`true`」に設定します。SQL の実行時間に関わらず、実稼働データベースを使用して初期化するには「`false`」に設定します。

- `Sql.delay.threshold`: My webMethods Server を起動するための初期化中に、最適化アルゴリズムを使用せずに SQL クエリを実行できる、SQL クエリの平均最大実行時間の推定値です。`Mws.inmemory.init` プロパティが「true」、「false」のいずれにも設定されていない場合に適用されます。デフォルト値は `20` ミリ秒です。

マップされたボリュームの `custom_wrapper.conf` ファイルで、必要なプロパティをカスタムの JVM パラメータとして設定します。詳細については、59 ページの「コンテナの設定の変更」を参照してください。

注:
起動時の最適化メカニズムをオフにしないかぎり、すべての My webMethods Server コンテナは、最適化を適用した起動時にローカルの Derby データベースインスタンスを必要とします。Derby のスクリプトは `Software AG_directory/MWS/server/derby-scripts.zip` にある My webMethods Server インストールで取得できます。My webMethods Server インストールからカスタムイメージを作成する場合は、使用中の `.dockerignore` ファイルなどにより、フォルダがスキップされないように注意してください。

My webMethods Server コンテナと webMethods Microservices Runtime の併用

webMethods Microservices Runtime は、マイクロサービスを実行するためのライトウェイトコンテナです。Microservices Runtime は、Integration Server の機能のサブセットを提供します。

My webMethods Server コンテナと、Web サービスプロバイダとしての Microservices Runtime を併用する場合は、特定の制限が適用されます。たとえば、Microservices Runtime では SAML 認証がサポートされないため、My webMethods Server で実行されているカスタムアプリケーションおよび階層化された製品のすべての Web サービスコネクタを基本認証に切り替える必要があります。

CAF Web サービスクライアントの基本認証は、My webMethods Server がコンテナで実行されている場合はデフォルトで有効になります。[**CAF アプリケーションランタイム設定**] ページで、Web サービスクライアントが Microservices Runtime または Integration Server に対する認証に使用するユーザ名とパスワードを設定できます。詳細については、63 ページの「CAF アプリケーションの Web サービス認証の設定」を参照してください。

Microservices Runtime の詳細については、『*Developing Microservices with webMethods Microservices Runtime*』を参照してください。Docker コンテナでの webMethods Microservices Runtime の実行の詳細については、『*webMethods Integration Server Administrator's Guide*』を参照してください。

CAF アプリケーションの Web サービス認証の設定

My webMethods Server がコンテナで実行されている場合、カスタムアプリケーションおよび階層化された製品のすべての Web サービスクライアントで、デフォルトで基本認証が有効化されてオンに切り替わり、個々のアプリケーションの設定がオーバライドされます。My webMethods Server コンテナと Integration Server コンテナを併用するなどの場合は、この設定を無効にすることができます。また、基本認証のデフォルトのユーザ名とパスワードを変更することもできます。

CAF アプリケーションの Web サービス認証設定を変更するには

1. sysadmin として、**[フォルダ] > [管理用フォルダ] > [Administration Dashboard] > [設定] > [CAF アプリケーションランタイム設定]** に移動します。

2. [**グローバルデフォルトの設定**] をクリックして、[**環境エントリ**] をクリックします。

3. 次の 1 つ以上の環境エントリを変更します。

環境エントリ	説明
`wsclient.global.basicauth-enabled`	すべての Web サービスクライアントに対してグローバルに基本認証を有効にするかどうか。デフォルトは `true` (有効) です。
`wsclient.global.basicauth-username`	Microservices Runtime または Integration Server に対する認証時に指定するユーザ名。デフォルトは `Administrator` です。 **注:** このユーザは、デフォルトでオフになっています。
`wsclient.global.basicauth-password`	Microservices Runtime または Integration Server に対する認証時に指定するパスワード。デフォルトは `manage` です。 **注:** デフォルトの管理者アカウントを使用する場合、このパスワードを管理者ユーザを有効にする時に指定したパスワードと置き換える必要があります。

4. [**適用**] をクリックします。

My webMethods Server コンテナの監視

デフォルトでは、My webMethods Server の生成された Dockerfile には、データベース接続、ディスクスペースとメモリの状態、ディレクトリサービスの接続などの様々なヘルスインジケータをテストして、各コンテナの状態を確認する `HEALTHCHECK` 命令が含まれています。

My webMethods Server のコンテナヘルスチェックは、コンテナを起動してから 5 分後に実行され、前回のチェックが完了してから 15 秒後に個別のチェックをトリガし、個別のチェックが 30 秒以内に完了しない場合は失敗するように設定されます。これらの設定は、インストールの生成された Dockerfile で変更できます。`HEALTHCHECK` 命令の詳細については、Docker のマニュアルを参照してください。

My webMethods Server コンテナには、次の事前定義済みヘルスインジケータが含まれています。

- `MemoryHealthIndicator` - システムの使用可能な合計メモリが、事前定義された値を超えている場合に合格します。事前定義された値は `20` (パーセント) です。デフォルトでは有効になっています。
- `JettyThreadHealthIndicator` - 利用可能な Jetty スレッドの最大数に対する、現在の Jetty スレッド数の比率が、事前定義された値を超えている場合に合格します。事前定義された値は `20` (パーセント) です。デフォルトでは有効になっています。
- `DiskspaceHealthIndicator` - 使用可能な合計ディスクスペースが、事前定義された値を超えている場合に合格します。事前定義された値は `20` (パーセント) です。デフォルトでは有効になっています。
- `DatabaseConnectivityHealthIndicator` - コンテナがデータベースに接続して、単純なクエリーを実行できる場合に合格します。デフォルトでは有効になっています。
- `AXSRFTGenerationHealthIndicator` - My webMethods Server が正常に AXSRFT トークンを生成する場合に合格します。デフォルトでは有効になっています。
- `CustomDataSourceHealthIndicator` - コンテナが、My webMethods Server に対して定義された外部データソースに接続できる場合に合格します。デフォルトでは有効になっています。
- `DirServiceConnectivityHealthIndicator` - コンテナが、My webMethods Server に対して定義された外部データソースに接続できる場合に合格します。デフォルトでは有効になっています。
- `ProcessEngineConnectivityHealthIndicator` - コンテナが、Process Engine をホストする Integration Server に接続できる場合に合格します。デフォルトでは、Task Engine を含むコンテナで有効になっています。
- `RulesEngineConnectivityHealthIndicator` - コンテナが、Business Rules をホストする Integration Server に接続できる場合に合格します。デフォルトでは、Task Engine を含むコンテナで有効になっています。
- `UmConnectionHealthIndicator` - コンテナが、Universal Messaging に接続できる場合に合格します。*useDbJms=false* の場合、デフォルトで有効です。

My webMethods Server コンテナは *http://container_host:port/health* で、すべての設定済みヘルスインジケータで測定された現在の値を表示する、専用のエンドポイントを提供します。*health* のエンドポイントにアクセスするには、管理者役割のメンバーであるアカウントを使用した基本認証が必要です。

healthCheck.json 設定ファイルで、ヘルスインジケータのデフォルトのしきい値を設定し、各チェックのオン／オフを変更できます。ヘルスチェックの設定ファイルの詳細については、66 ページの「コンテナヘルスチェックのサンプル」を参照してください。

66 ページの「カスタムのヘルスインジケータの実装」に説明されているように、コンテナのデフォルトのヘルスインジケータに加えて、カスタムのインジケータを実装して My webMethods Server に登録できます。

カスタムのヘルスインジケータの実装

My webMethods Server コンテナのカスタムのヘルスインジケータを作成して登録する大まかな手順は、次のとおりです。

- CAF アプリケーションプロジェクトで、`IHealthIndicator` を実装するクラスを作成します。
- 必要なヘルスチェックを実行するカスタムコードをクラスに追加します。
- 同じアプリケーションプロジェクトで、インジケータプロパティを設定するカスタムコードを使って、IHealthIndicatorConfig クラスを実装するカスタムクラスを追加します。
- 66 ページの「コンテナヘルスチェックのサンプル」の説明に従って、`healthCheck.json` ファイルに新しいカスタムインジケータを追加します。
- healthCheck.json ファイルを `instance_cfg` ディレクトリに、またコンパイル済み CAF アプリケーションを別の専用ディレクトリに保存します。どちらのディレクトリも任意のボリュームまたはホストファイルシステムに存在します。
- My webMethods Server コンテナの起動時に、`instance_cfg` をコンテナファイルシステム上の `MWS/volumes/config` ディレクトリに、また CAF アプリケーションを含むディレクトリを `MWS/volumes/apps` にマウントします。

コンテナヘルスチェックのサンプル

例: healthCheck.json

次の例を使用して `healthCheck.json` ファイルを作成し、デフォルトのインジケータと一緒にカスタムのヘルスインジケータを含めます。`healthCheck.json` ファイルで指定する設定によって、すべてのヘルスチェックインジケータの設定がオーバライドされます。`healthCheck.json` ファイルに加えた変更は、コンテナを起動または再起動するときに適用されます。

```
{
 "enabled": "true",
 "indicators": {
  "MemoryHealthIndicator": {
   "enabled": "true",
   "properties": {
    "memoryAvailable": 20
   }
  },
  "JettyThreadHealthIndicator": {
   "enabled": "true",
   "properties": {
    "currentToMaxThreadRatioThreshold": 20
   }
  },
  "DiskspaceHealthIndicator": {
   "enabled": "true",
   "properties": {
    "diskSpaceAvailable": 20
   }
  },
```

```
  "DatabaseConnectivityHealthIndicator": {
   "enabled": "true"
  },
  "AXSRFTGenerationHealthIndicator": {
   "enabled": "true"
  },
  "CustomDataSourceHealthIndicator": {
   "enabled": "true"
  },
  "DirServiceConnectivityHealthIndicator": {
   "enabled": "true"
  },
  "ProcessEngineConnectivityHealthIndicator": {
   "enabled": "true"
  },
  "RulesEngineConnectivityHealthIndicator": {
   "enabled": "true"
  },
  "UmConnectionHealthIndicator": {
   "enabled": "true"
  }
  "MyFavIndicatorConfig" : {
   "enabled": "true",
   "properties": {
    "myIndicatorProperty": "propValue"
   }
  }
 }
```

MyFavIndicatorConfig はカスタムヘルスチェッククラスの名前、 *myIndicatorProperty* はコードで定義された カスタム設定プロパティです。定義する各インジケータプロパティについて、*healthCheck.json* ファイルのコードにエントリを 追加します。

例: MyCustomIndicatorConfig.java

My webMethods Server コンテナにカスタムヘルスインジケータの設定プロパティを定義するサンプルクラスです。このクラスは *healthCheck.json* ファイルから提供された設定をマップし、設定を My webMethods Server ランタイムに転送するために必要です。

```
package caf.war.TestHealthIndicators.healthtest;
import java.util.Map;
import java.util.Optional;
import com.webmethods.portal.system.init.InitializationException;
import com.webmethods.rtl.container.health.IHealthIndicatorConfig;
public class MyFavIndicatorConfig implements IHealthIndicatorConfig {
    private boolean enabled;
    private Map<String, Object> properties;
    @Override
    public boolean isEnabled() {
        return this.enabled;
    }
    public void setEnabled(boolean enabled) {
        this.enabled = enabled;
    }
    @Override
    public Map<String, Object> getProperties() {
        return this.properties;
    }
```

```
    // Implement for all custom properties, listed in healthCheck.json
    private String myIndicatorValue;
    // Implement for all custom properties, listed in healthCheck.json
    public String getMyIndicatorValue() {
        return this.myIndicatorValue;
    }
    public void setProperties(Map<String, Object> properties) throws
InitializationException {
        // Modify according to your specific data types
        this.myIndicatorValue = getConfigProperty("myIndicatorValue", String.class);
    }
    protected <T extends Object> T getConfigProperty(String key, Class<T> returnType)
 throws InitializationException {
        Object result = Optional.ofNullable(this.properties.get(key));
        return returnType.cast(result);
    }
}
```

例: MyCustomIndicator.java

My webMethods Server のコンテナにカスタムのヘルスインジケータを作成するサンプルクラスです。

```
package caf.war.TestHealthIndicators.healthtest;
import org.springframework.boot.actuate.health.Health;
import org.springframework.boot.actuate.health.Status;
import com.fasterxml.jackson.databind.JsonNode;
import com.fasterxml.jackson.databind.ObjectMapper;
import com.webmethods.rtl.container.health.IHealthIndicator;
public class MyFavIndicator implements IHealthIndicator<MyFavIndicatorConfig> {
    private MyFavIndicatorConfig config;
    private Class<MyFavIndicatorConfig> configClass;
    @Override
    public void configure(JsonNode configNode) throws Exception {
        if (configNode != null) {
           this.config = new ObjectMapper().treeToValue(configNode, this.configClass);
        }
    }
    @Override
    public MyFavIndicatorConfig getConfiguration() {
        return this.config;
    }
    // Define custom indicator values and measurement logic
    @Override
    public Health health() {
        // Determine healthStatus using the options in the HealthStatus enum in
        // IHealthIndicator, or define custom health statuses.
        // Include additional health check code here.
        // ...
        return Health.status(new Status(HealthStatus.STATUS_OK.getKey())).build();
    }
}
```

Prometheus による My webMethods Server コンテナの監視

Prometheus は、Docker インスタンスを監視し、実行中のコンテナのメトリクスを収集および分析するように設定できる、システム監視のための一連のツールです。My webMethods Server

コンテナには、Prometheus と互換性のある形式で My webMethods Server のパフォーマンスメトリクスを公開する専用のエンドポイントが `http://container_host:port/metrics` に用意されています。サポートされるメトリックタイプはゲージです。メトリクスとメトリックタイプの詳細については、Prometheus のマニュアルを参照してください。

Prometheus サーバから My webMethods Server のメトリクスのエンドポイントにアクセスするには、Admin Role のメンバーであるアカウントを使用した基本認証が必要です。

My webMethods Server のパフォーマンスメトリクスによって、さまざまなサーバキャッシュ、データベース接続、さまざまな要求およびアクションのロードと実行の時間、JVM メモリ、ユーザセッションなどの情報が提供されます。`http://container_host:port/metrics` で、Prometheus と互換性のある形式の、すべてのメトリクスのヘルプ文字列を確認できます。

My webMethods Server コンテナのクラスタリング

同じデータベースに対してコンテナを実行して同じ JMS プロバイダを使用する方法で、手動で My webMethods Server コンテナからクラスタを作成できますが、クラスタ化環境を作成する推奨される方法は、Docker Compose または Kubernetes を使用することです。

My webMethods Server コンテナからクラスタを作成して使用する場合、特定の制限が適用されます。

- 選択した方法に関係なく、オンプレミスの通常のサーバ/VM にインストールされた My webMethods Server インスタンスと、Docker コンテナで実行されるインスタンスの両方を含むクラスタである、ハイブリッドクラスタを作成することはできません。コンテナ内またはオンプレミスのインストールから実行される新しいクラスタノードを追加するときに、My webMethods Server によって、ノードタイプとクラスタタイプが一致しているかどうかが確認されて、検証が失敗した場合はノードがシャットダウンされます。
- 同じクラスタの My webMethods Server のバージョンが異なるコンテナを実行することはできません。
- [クラスタ管理] ページからは、クラスタまたは個々のクラスタメンバーを再起動することも、シャットダウンすることもできません。[**クラスタ管理**] ページからノードを保守モードにすると、コンテナの再起動時に、イメージまたは `/MWS/volumes/configs/instance_cfg` ディレクトリ (使用可能な場合) からの保守設定がオーバライドされます。
- クラスタリング用に Kubernetes を使用する場合は、`StatefulSet` オブジェクトと `Ingress` オブジェクトを使用して My webMethods Server を定義する必要があります。

My webMethods Server コンテナクラスタの設定

My webMethods Server コンテナには、外部ボリュームをマウントし、`cluster.xml` ファイルにコンテナクラスタの初期設定を指定できる専用の場所である `/MWS/volumes/configs/cluster_cfg` が用意されています。Docker Compose テンプレートまたは Kubernetes オブジェクトデスクリプタで、外部ボリュームを定義できます。クラスタが起動されると、My webMethods Server によって、ボリュームが `/MWS/volumes/configs/cluster_cfg` にマウントされているかどうかが確認されて、すべてのクラスタノードがアクセスできるように、`cluster.xml` ファイルがボリュームからデータベースにコピーされます。クラスタを再起動したら、ボリュームを介して新しい設

定を指定するまで、クラスタノードは `cluster.xml` を使用し続けます。Docker コンテナで構成される My webMethods Server クラスタを実行しているときは、ローカルの `cluster.xml` ファイルを使用することはできません。

[クラスタの設定] > [拡張またはクラスタ設定] > [JMS プロバイダ URL] > [高度な設定] を使用して、クラスタノードで使用される Universal Messaging サーバへの SSL 通信を有効にするなど、初期設定をさらに変更することができます。My webMethods Server Web ユーザインタフェースで行った変更は、コンテナファイルシステムの `MWS/volumes/data` ディレクトリ、またはそこにマウントされたボリュームに保存され、コンテナの再起動後も保持されます。これらの設定は、ノードごとに適用され、`/MWS/volumes/configs` ディレクトリにマウントされたボリュームを介して指定された設定をオーバライドします (このようなボリュームが使用可能な場合)。

My webMethods クラスタのサンプル

Software AG は、Docker Compose と Kubernetes で使用するための、一連のサンプルの My webMethods Server クラスタテンプレートファイルを用意しています。このサンプルは、GitHub リポジトリ https://github.com/SoftwareAG/mws-docker-samples で取得できます。

6 コマンドラインからの My webMethods Server の実行

- My webMethods Server の基本のコマンドライン構文 72
- My webMethods Server コマンドの実行 73
- コマンドラインヘルプへのアクセス 73
- My webMethods Serverインスタンスの管理コマンド 73
- My webMethods Serverインスタンスの操作コマンド 79
- My webMethods Serverインスタンスの設定コマンド 82
- My webMethods Server サービスの管理コマンド 84
- My webMethods Server OSGi プロファイルコマンド 86
- 複数のサーバインスタンスでの My webMethods Server コマンドの開始、停止および実行 89
- My webMethods Server のコマンドの JVM プロパティの設定 90
- mws コマンドのログファイル 91

My webMethods Server の基本のコマンドライン構文

次の表では、My webMethods Server の基本のコマンドライン構文について説明します。

Windows の場合: `mws.bat -option -option ...command parameter|-Dparameter=value`

UNIX の場合: `mws.sh -option -option ...command parameter|-Dparameter=value`

-parameter はネイティブコマンドパラメータ、*-Dparameter* は Java システムパラメータを示します。

コマンドには、スペースで区切って複数のオプションと複数のパラメータを指定できます。

Java システムパラメータを受け入れないコマンドの場合は、`custom_wrapper.conf` ファイルで追加の Java プロパティとしてパラメータを指定します。

各コマンドでサポートされているパラメータの詳細については、そのコマンドの「パラメータ」セクションを参照してください。

一部の My webMethods Server コマンドは、Docker コンテナで実行されているインスタンスには適用されません。コマンドの説明の「使用上の注意」セクションに、ランタイムの制限がリストされています。

My webMethods Server コマンドのオプション

My webMethods Server のコマンドラインインタフェースでは以下のオプションがサポートされていますが、My webMethods Server の各コマンドは、そのコマンドの「オプション」セクションに一覧されているオプションのみを受け入れます。次の表に使用可能なコマンドラインのオプションを示します。

オプション	説明
`-p`	プラットフォームのインストールディレクトリへの完全パス。デフォルトは *Software AG_directory* です。
`-d` | `debug`	サーバをデバッグモードで起動します。Java デバッグリスナーがポート `10033` で開き、DEBUG ステートメントがコンソールウィンドウに表示されます。
`-n`	クラスタ化環境で、サーバインスタンスに割り当てられたノード名。サーバがスタンドアロンで稼動している場合は不要です。
`-s`	サーバインスタンスの名前。デフォルトのサーバを制御する場合は不要です。デフォルト値は `default` です。
`-w`	コマンドを実行するまで待機する期間 (秒)。

My webMethods Server コマンドの実行

My webMethods Server コマンドラインユーティリティを使用して、基本的な管理および設定操作を実行できます。たとえば、コマンドラインからサーバを起動すると、デバッグモードを使用して、サーバのアクティビティを記録したり表示したりできるようになります。

My webMethods Server コマンドを実行するには

1. コマンドラインプロンプトで、My webMethods Server`bin` ディレクトリに移動します。

```
Software AG_directory\MWS\bin
```

2. コマンドを実行するには、必要なオプションやパラメータとともに My webMethods Server コマンドを入力し、**Enter** キーを押します。

例

```
mws new -Dserver.name=testInstance -Dhttp.port=8586 -Ddb.type=sqlserver
-Ddb.url=jdbc:wm:sqlserver://dbserver:1433;databaseName=WMDB
-Ddb.username=sa -Ddb.password=manage
```

このコマンドの例では、ポート `8586` でアクセス可能な `testInstance` という名前の新しい My webMethods Server インスタンスが作成されます。新しいサーバインスタンスは、マシン `dbserver` のポート `1433` にインストールされ、`WMDB` という名前の SQL Server データベースを使用します。データベースに接続するユーザ名は `sa`、パスワードは `manage` です。

コマンドラインヘルプへのアクセス

My webMethods Server のコマンドラインユーティリティは、`help` コマンドを使用して、コマンドプロンプトでアクセス可能なコマンドラインヘルプを提供します。`help` コマンドはオプションやパラメータを使用しません。

コマンドラインヘルプコンテンツへアクセスするには

1. コマンドプロンプトで、次のコマンドを入力して My webMethods Server の bin ディレクトリに移動します。

```
cd Software AG_directory\MWS\bin
```

2. 以下のコマンドを入力します。

```
mws help
```

My webMethods Serverインスタンスの管理コマンド

以下のコマンドを使用して、My webMethods Server インスタンスの管理操作を行います。

new

新規のサーバインスタンスを作成します。次の形式で、コマンドパラメータとしてインスタンスの詳細を指定します: `-D`*`parameter=value`*

オプション

`new` コマンドはオプションを使用しません。

パラメータ

次の表に、new コマンドが受け入れるパラメータを示します。

パラメータ	説明
`server.name`	オプション。サーバインスタンスの名前。マシン上のサーバインスタンス間で一意の名前を指定します。値を指定しない場合、インスタンス名は `default` になります。
`http.port`	オプション。サーバインスタンスが受信待機するポート番号。マシン上のサーバインスタンス間で一意のポート番号を指定します。デフォルトのポート番号は `8585` です。
`install.service`	オプション。My webMethods Server の新規インスタンスをアプリケーションとしてインストールするか、サービスとしてインストールするかを指定します。有効な値は次のとおりです。 ■ `true` - サービスとしてインストールします。 ■ `false` - デフォルトの値です。アプリケーションとしてインストールします。
`node.name`	オプション。サーバインスタンスをホストするクラスタノードの名前。名前を指定しない場合、デフォルト値は `master` になります。
`db.type`	必須。サーバインスタンスが使用するデータベースのタイプ。有効な値は次のとおりです。 ■ `ms` - Microsoft SQL Server ■ `oracle`: Oracle ■ `db2`: DB2 ■ `mysqlee` - MySQL Enterprise Edition ■ `mysqlce` - MySQL Community Edition ■ `postgresql` - PostgreSQL

パラメータ	説明
db.url	必須。データベースのタイプおよびドライバに基づく、データベース接続 URL。 UNIX システムでは、データベース接続 URL のすべてのセミコロン文字をバックスラッシュでエスケープするか、データベース URL パラメータの全体を次のように二重引用符で囲む必要があります。 `"-Ddb.url=jdbc:wm:database://host:port;databaseName=name"`
db.driver	オプション。My webMethods Server データベースへの接続に使用する JDBC ドライバのクラス名。Software AG が提供するものとは異なる JDBC ドライバを使用する場合は、このフィールドは必須です。
db.username	すべての外部データベースに必須。My webMethods Server データベースに割り当てられたユーザ名。
db.password	すべての外部データベースに必須。My webMethods Server データベースユーザのパスワード。
jndiProviderUrl	必須。JMS プロバイダとして使用する Universal Messaging サーバの URL。JNDI プロバイダ URL を指定しない場合、あるいは Universal Messaging サーバが使用できない場合は、My webMethods Server インスタンスが保守モードで起動します。
https.port	オプション。HTTPS リスナポート。値 0 を指定すると、リスナーが無効になります。
debug.port	オプション。Java デバッグポート。デフォルトのポート番号は 10033 です。
jmx.port	オプション。新規サーバインスタンスの JMX ポート。デフォルトのポート番号は 5002 です。
http.proxy.host	オプション。プロキシホスト名。
http.proxy.port	オプション。プロキシポート番号。
http.proxy.user	オプション。プロキシユーザ名。
http.proxy.password	オプション。プロキシのパスワード。
components.include	オプション。新規サーバインスタンスに含める *Software AG_directory*\MWS\components ディレクトリ内のコンポーネント。
components.exclude	オプション。新規サーバインスタンスから除外する *Software AG_directory*\MWS\components ディレクトリ内のコンポーネント。

パラメータ	説明
`components.overwrite`	オプション。*Software AG_directory*\MWS\components ディレクトリ内のコンポーネントファイルより古い *Software AG_directory*\MWS\server\serverName\deploy ディレクトリ内のコンポーネントファイルを上書きするかどうかを指定します。有効な値は次のとおりです。 ■ `true` - \deploy ディレクトリ内のコンポーネントファイルを上書きします。 ■ `false` - デフォルトの値です。\deploy ディレクトリ内のコンポーネントファイルを上書きしません。

使用上の注意

- 新規サーバインスタンスを作成する前に、『*Installing Software AG Products*』の説明に従い、Database Component Configurator を使用して、インスタンス用に独自のデータベースまたはテーブルスペースを作成してください。My webMethods Server クラスタにノードを作成する場合は必要ありません。
- 新規サーバインスタンスの作成後に、`init` コマンドを使用してインスタンスを初期化してください。新規 My webMethods Server インスタンスの最初の初期化は、完了するまで数分かかります。最初に初期化が行われた後、サーバは自動的にシャットダウンします。
- My webMethods Server がコンテナで実行されている場合はサポートされません。

例

- サーバ `db_server` にインストールされた外部 MS SQL Server データベース `my_wm_msql` を使用し、ポート `8090` で稼働する `test` という名前の新規インスタンスを作成する場合、次のように指定します。

```
> mws new -Dserver.name=test -Dhttp.port=8090
-Ddb.type=ms -Ddb.url=jdbc:wm:sqlserver://db_server:1433;
DatabaseName=my_wm_msql;SelectMethod=direct;MaxPooledStatements=100
-Ddb.username=mws_user -Ddb.password=password
[Configuration output displayed in console window....]
> mws -s test run
```

  My webMethods Server は、ユーザ名 `mws_user`、パスワード `password` を使用してデータベースに接続します。

init

My webMethods Server を起動し、My webMethods Server データベースを初期化してから、サーバを停止します。My webMethods Server インスタンスの作成または更新後にこのコマンドを使用します。

オプション

`-p`

`-s`

`-d`

`-n`

パラメータ

次の形式で Java システムプロパティをパラメータとして受け入れます。

```
-Dany.java.system.property=value
```

使用上の注意

- My webMethods Server がコンテナで実行されている場合はサポートされません。

update

クラスパスを更新し、新規または更新済みコンポーネントを展開します。

オプション

`-p`

`-s`

`-w`

パラメータ

次の表に、update コマンドが受け入れるパラメータを示します。

パラメータ	説明
`node.name`	オプション。My webMethods Server インスタンスをホストするクラスタノードの新しい名前を指定します。
`db.type`	オプション。サーバインスタンスが使用するデータベースのタイプ。有効な値は次のとおりです。 ■ `ms` - Microsoft SQL Server ■ `oracle`: Oracle ■ `db2`: DB2

パラメータ	説明
	■ mysqlee - MySQL Enterprise Edition ■ mysqlce - MySQL Community Edition
db.url	オプション。データベース接続 URL。UNIX システムでは、データベース URL パラメータを次のように二重引用符で囲む必要があります。 "-Ddb.url=jdbc:wm:*database*://*host:port*;databaseName=*name*"
db.driver	オプション。データベースへの接続時に使用する JDBC ドライバのクラス名。Software AG が提供するものとは異なる JDBC ドライバを使用する場合に必須です。
db.user	すべての外部データベースに必須。データベースユーザのユーザ名。
db.password	すべての外部データベースに必須。データベースユーザのパスワード。
components.overwrite	オプション。Software AG_directory\MWS\components ディレクトリ内のコンポーネントファイルより古い Software AG_directory\MWS\server\serverName \deploy ディレクトリ内のコンポーネントファイルを上書きするかどうかを指定します。有効な値は次のとおりです。 ■ true - サーバの deploy ディレクトリ内のコンポーネントファイルを上書きします。 ■ false - deploy ディレクトリ内のコンポーネントファイルを上書きしません。これがデフォルト値です。
dont.update.classpath	オプション。生成された My webMethods Server クラスパスを更新するかどうかを指定します。有効な値は次のとおりです。 ■ true - サーバクラスパスを更新しません。 ■ false - サーバクラスパスを更新します。これがデフォルト値です。

使用上の注意

- My webMethods Server がコンテナで実行されている場合はサポートされません。

delete

既存の My webMethods Server インスタンスを削除します。

オプション

-p

-s

-w

パラメータ

delete コマンドはパラメータを使用しません。

使用上の注意

- delete コマンドは、インスタンスを削除する前に確認のプロンプトを表示しません。このコマンドを使用する場合は十分に注意してください。
- My webMethods Server がコンテナで実行されている場合はサポートされません。

My webMethods Serverインスタンスの操作コマンド

以下のコマンドを使用して、My webMethods Server インスタンス管理操作を行います。

run

同じコンソールウィンドウでサーバを起動します。

オプション

-p

-s

-n

-d

パラメータ

stop コマンドは、次の形式で Java システムプロパティをパラメータとして受け入れます。

```
-Dany.java.system.property=value
```

使用上の注意

- My webMethods Server がコンテナで実行されている場合はサポートされません。

start

新規のコンソールウィンドウでサーバを起動します。UNIX ベースのオペレーティングシステムの場合、プロセスがバックグラウンド実行モードで開始されます。

オプション

-p

-s

-n

-d

-w

パラメータ

start コマンドは、次の形式で Java システムプロパティをパラメータとして受け入れます。

```
-Dany.java.system.property=value
```

使用上の注意

- My webMethods Server がコンテナで実行されている場合はサポートされません。

restart

稼動中のサーバを停止してから、再び起動します。

オプション

-p

-s

-n

-d

-w

パラメータ

restart コマンドは、次の形式で Java システムプロパティをパラメータとして受け入れます。

```
-Dany.java.system.property=value
```

使用上の注意

- My webMethods Server がコンテナで実行されている場合はサポートされません。

stop

実行中のサーバを停止します。

オプション

```
-p

-s

-n

-w
```

パラメータ

`stop` コマンドは、次の形式で Java システムプロパティをパラメータとして受け入れます。

```
-Dany.java.system.property=value
```

使用上の注意

- My webMethods Server がコンテナで実行されている場合はサポートされません。

ping

実行中のサーバインスタンスに関する情報 (たとえば、サーバポート、役割、フロントエンド URL、初期化時間など) を返します。サーバが停止すると、`Portal ping failed.` と返します。`ping` コマンドを実行するには認証が必要です。

オプション

```
-s

-n

-w
```

パラメータ

次の表に、ping コマンドが受け入れるパラメータを示します。

パラメータ	説明
mws_username	必須。コマンドの実行を認証する My webMethods Server 管理ユーザのユーザ名です。
mws_password	必須。コマンドの実行を認証する My webMethods Server 管理ユーザのパスワードです。

例

ユーザ名 Administrator、パスワード manage を使用する、test という名前のサーバインスタンスに ping を実行するには、次のように指定します。

```
mws -s test ping Administrator manage
```

updatesinfo

My webMethods Server にインストールされた修正、My webMethods Server Common Library、インストール済みの webMethods アプリケーションのユーザインタフェースに関する情報を表示します。

オプション

-s

-n

パラメータ

updatesinfo コマンドはパラメータを使用しません。

My webMethods Serverインスタンスの設定コマンド

以下のコマンドを使用して、My webMethods Server インスタンス設定操作を行います。

getconfig

My webMethods Server データベースから設定ファイルをダウンロードします。以下のように、設定ファイルの名前をコマンドのパラメータとして指定します。

```
mws getconfig fileName
```

My webMethods Server データベースに格納されている設定ファイルのリストおよび詳細については、102 ページの「データベースに保存された設定ファイルの変更」を参照してください。

putconfig

My webMethods Server データベースに設定ファイルをアップロードします。以下のように、設定ファイルの名前をコマンドのパラメータとして指定します。

```
mws getconfig fileName
```

My webMethods Server データベースに格納されている設定ファイルのリストおよび詳細については、102 ページの「データベースに保存された設定ファイルの変更」を参照してください。

export

マイグレーション用に My webMethods Server ランタイムデータをエクスポートします。

オプション

```
-s
```

パラメータ

次の表に、export コマンドが受け入れるパラメータを示します。

パラメータ	説明
descriptor_filefull_path	必須。マイグレーションデータのエクスポートに使用する記述子ファイルへの完全パス。
target_folder	必須。ランタイムデータのエクスポートに使用するフォルダへの完全パス。
mws_username	必須。コマンドの実行を認証する My webMethods Server 管理ユーザのユーザ名です。
mws_password	必須。コマンドの実行を認証する My webMethods Server 管理ユーザのパスワードです。

import

既存の My webMethods Server インストールからランタイムデータをインポートします。

オプション

```
-s
```

パラメータ

次の表に、import コマンドが受け入れるパラメータを示します。

パラメータ	説明
info_properties_filefull_path	必須。インポートする .properties ファイルへの完全パス。
mws_username	必須。コマンドの実行を認証する My webMethods Server 管理ユーザのユーザ名です。
mws_password	必須。コマンドの実行を認証する My webMethods Server 管理ユーザのパスワードです。

My webMethods Server サービスの管理コマンド

Windows サービスや UNIX デーモンとして My webMethods Server をインストールまたは使用する場合に、サーバインスタンスを管理または操作するには、以下のコマンドを使用します。

installservice

installservice コマンドは、Windows のサービスまたは UNIX システムのデーモンとしてMy webMethods Server を登録します。

オプション

-p

-s

-n

-d

パラメータ

Windows システムでは、installservice コマンドは次の形式で Java システムプロパティをパラメータとして受け入れます。

```
-Dany.java.system.property=value
```

UNIX システムでは、サービスをインストールするユーザのユーザ名を指定する必要があります。

使用上の注意

- My webMethods Server がコンテナで実行されている場合はサポートされません。

uninstallservice

`uninstallservice` コマンドは、Windows サービスまたは UNIX デーモンとして My webMethods Server を登録解除します。

オプション

`-p`

`-s`

パラメータ

コマンドは追加のパラメータを受け入れません。

使用上の注意

- My webMethods Server がコンテナで実行されている場合はサポートされません。

startservice

Windows にのみ適用されます。`startservice` は、Windows サービスとしてインストールされた My webMethods Server インスタンスを起動します。

オプション

`-p`

`-s`

パラメータ

`startservice` コマンドはパラメータを使用しません。

使用上の注意

- My webMethods Server がコンテナで実行されている場合はサポートされません。

stopservice

Windows にのみ適用されます。Windows サービスとしてインストールされた My webMethods Server インスタンスを停止します。

オプション

`-p`

```
-s
```

パラメータ

`stopservice` コマンドはパラメータを使用しません。

使用上の注意

- My webMethods Server がコンテナで実行されている場合はサポートされません。

restartservice

Windows にのみ適用されます。Windows サービスとしてインストールされた My webMethods Server インスタンスを停止してから起動します。

オプション

```
-p

-s
```

パラメータ

`restartservice` コマンドはパラメータを使用しません。

使用上の注意

- My webMethods Server がコンテナで実行されている場合はサポートされません。

My webMethods Server OSGi プロファイルコマンド

以下のコマンドを使用して、My webMethods Server インスタンスの OSGi プロファイルの管理を行います。

create-osgi-profile

My webMethods Server バンドルを使用して、指定されたサーバインスタンスに対して、OSGi プロファイルを作成します。プロファイル名は、サーバインスタンスの名前の前に `MWS_` が付加されたものになります。

オプション

```
-p

-s
```

パラメータ

`create-osgi-profile` コマンドはパラメータを使用しません。

使用上の注意

- 破損したプロファイルを削除または廃止した場合は、`create-osgi-profile` コマンドを使用して、サーバインスタンス用に新しい OSGi プロファイルを作成します。
- My webMethods Server がコンテナで実行されている場合はサポートされません。

update-osgi-profile

指定されたサーバインスタンスの OSGi プロファイルを更新します。サーバ名、サービス名、JAVA_HOME 変数をリセットして、サーバインスタンスに関連付けられた OSGi バンドルを更新します。

オプション

`-p`

`-s`

パラメータ

`update-osgi-profile` コマンドはパラメータを使用しません。

使用上の注意

- サーバインスタンスの設定を変更する場合、または新しい webMethods アプリケーションをインストールして既存のサーバインスタンス上で実行する場合は、`update-osgi-profile` コマンドを使用します。
- OSGi プロファイルを更新する前に、`custom_wrapper.conf` 設定ファイルのバックアップコピーを作成します。
- My webMethods Server がコンテナで実行されている場合はサポートされません。

deprecate-osgi-profile

指定されたサーバインスタンスの OSGi プロファイルの名前を変更して、日時のタイムスタンプを追加し、プロファイルをアーカイブを行います。これは、OSGi プロファイルを更新し、バックアップコピーを保存するのと同じ操作になります。

オプション

`-p`

```
-s
```

パラメータ

`deprecate-osgi-profile` コマンドはパラメータを使用しません。

使用上の注意

- OSGi プロファイルを廃止する場合、プロファイルの名前を変更して、サーバインスタンスとの関連付けが解除されるようにします。名前を変更すると、`create-osgi-profile` コマンドを使用して、新しいプロファイルを作成できます。
- My webMethods Server がコンテナで実行されている場合はサポートされません。

delete-osgi-profile

指定されたサーバインスタンスの OSGi プロファイルを削除します。

オプション

```
-p
```

```
-s
```

パラメータ

`delete-osgi-profile` コマンドはパラメータを使用しません。

使用上の注意

- インスタンスの新しいプロファイルを作成する前に、`delete-osgi-profile` コマンドを使用して破損した OSGi プロファイルを削除します。
- OSGi プロファイルを削除する前に、`custom_wrapper.conf` 設定ファイルのバックアップコピーを作成します。
- My webMethods Server がコンテナで実行されている場合はサポートされません。

例

```
mws -s testInstance delete-osgi-profile
...
[Returns confirmation]
mws -s testInstance create-osgi-profile
...
[Returns confirmation]
```

このコマンドラインの例では、*testInstance* という名前の My webMethods Server インスタンスの OSGi プロファイルを削除してから再作成しています。

複数のサーバインスタンスでの My webMethods Server コマンドの開始、停止および実行

すべてのサーバインスタンスで一度に My webMethods Server コマンドを開始、停止または実行する場合、マシンにインストールされているすべてのサーバインスタンスに関連付けられているコマンドがあります。

インスタンスがコンテナで実行されている場合は、複数のサーバインスタンスに対して My webMethods Server コマンドを実行することはできません。デフォルトでは、これらのスクリプトは正式な My webMethods Server イメージと Docker スクリプトから除外され、カスタムイメージに明示的に含まれる場合でも反映されません。

すべてのサーバインスタンスでコマンドを開始、停止または実行するには

1. コマンドラインプロンプトで、次のコマンドを入力してコマンドのホームディレクトリに移動します。

```
cd Software AG_directory\MWS\bin
```

2. 以下のコマンドのいずれかを入力します。

次の表に、複数サーバのインスタンス上で My webMethods Server コマンドを実行するためのコマンド構文を示します。

目的	オペレーティングシステム	コマンド

すべてのサーバインスタンスでコマンドを実行する。次のコマンドはサポートされていません。
`run`
`new`
`ant`
	Windows	`mwsall.bat`

	UNIX	`mwsall.sh`
すべてのサーバインスタンスを開始する。サーバインスタンスは、アルファベット順に開始されます。Windows オペレーティングシステムの場合、登録済みのサービスがある場合にはサーバインスタンスをサービスとして開始し、登録済みのサービスがない場合にはインスタンスをプロセスとして開始します。UNIX ベースのオペレーティングシステムの場合、インスタンスをプロセスとしてのみ開始します。	Windows	`startall.bat`
	UNIX	`startall.sh`

目的	オペレーティングシステム	コマンド
サーバインスタンスを停止する。コマンドは、すべてのサーバインスタンスに `mws.{bat|sh} stop` を呼び出します。	Windows	`stopall.bat`
	UNIX	`stopall.sh`

My webMethods Server のコマンドの JVM プロパティの設定

My webMethods Server のコマンドラインユーティリティは、メインサーバのランタイム環境とは異なる別の JVM で実行されます。My webMethods Server のコマンドを実行すると、ユーティリティは custom_wrapper.conf ファイルで My webMethods Server に指定した JVM の設定に関わらず、デフォルト設定を使用します。My webMethods Server のコマンドユーティリティの JVM 設定をカスタマイズしたい場合、サーバインスタンスの `Software AG_directory\MWS\server\serverName\config\` ディレクトリにある `server.properties` ファイルで必要な設定を行います。

例: server.properties 設定ファイル

```
# JVM configuration
jvm.arg=-Xms32m
jvm.arg=-Xmx1024m
jvm.arg=-server

# SSL options
java.option=-Djavax.net.ssl.keyStore="${server.home}/config/security/localhost.p12"
java.option=-Djavax.net.ssl.keyStorePassword={DES}vrFIelCdkow=
java.option=-Djavax.net.ssl.keyStoreType=pkcs12

java.option=-Djavax.net.ssl.trustStore="${server.home}/config/security/sagdemoca.jks"
java.option=-Djavax.net.ssl.trustStorePassword={DES}vrFIelCdkow=
java.option=-Djavax.net.ssl.trustStoreType=jks
# Proxy configuration
java.option=-DproxySet=false
java.option=-Dhttp.proxyHost=
java.option=-Dhttp.proxyPort=
java.option=-Dhttp.proxyUser=
java.option=-Dhttp.proxyPassword=
java.option=-Dhttp.nonProxyHosts=
```

JVM 設定のオプションは以下のように指定します。

```
jvm.arg=jvm.configuration.argument
```

Java システムプロパティは以下のように指定します。

```
java.option=-Dany.java.system.property=value
```

mws コマンドのログファイル

一部の mws コマンドについては、My webMethods Server によって、コマンド出力のログがコンソールに記録されることに加えて、コマンド実行からの情報を含むログファイルが作成されます。ログファイル名は、次の形式で使用されます。

command-name_server-name.log

ここで、

- *command-name* is the name of the mws command.
- *server-name* は、コマンドの影響を受けるサーバインスタンスの名前です。

たとえば、次のコマンドを発行した場合:

```
mws -s alpha create-osgi-profile
```

生成されるログファイルは、create-osgi-profile_alpha.log という名前になります。

mws コマンドのログファイルは次の場所に書き込まれます。

Software AG_directory \MWS\bin

mws コマンドが正常に終了しなかった場合、次の終了コードを検索すると、該当ログファイルをすばやく確認できます。

```
ExitCode: 13
```

このコードの後に続く、不明な要件 (不明なバンドルまたは機能 ID など) を確認する必要があります。失敗の明確な理由が確認できない場合、Software AG Global Support が解析できるように、ログファイルのコピーを用意してください。

My webMethods Server がコンテナ内で実行されている場合、コンテナの起動時にテストパッチまたはサードパーティのライブラリを適用するなどの特定のアクションのみによって、インスタンスの bin ディレクトリにログファイルが書き込まれます。ログファイルのボリュームを指定していないときに、これらのログファイルを保持する場合は、コンテナをシャットダウンする前にバックアップコピーを作成します。

7 設定ファイルの変更

- Java サービスラッパー 94
- My webMethods Server の JVM 設定 97
- データベースに保存された設定ファイルの変更 102
- Solaris、HP-UX または Linux で My webMethods Server を 32 ビットで実行するための設定 104
- HTTP リスナーの単一 IP アドレス使用の設定 105
- 起動時に診断を実行するかどうかの設定 108

Java サービスラッパー

My webMethods Server は Software AG Common Platform で実行され、これは Java 仮想マシン (JVM) で実行されます。Java Service Wrapper は、Tanuki Software, Ltd. によって開発されたアプリケーションです。これは、My webMethods Server を実行する JVM を起動するユーティリティプログラムです。

JVM の起動に加えて、Java Service Wrapperには JVM の監視機能、コンソール出力のログ機能、およびスレッドダンプの生成機能が備わっています。Java Service Wrapper の詳細については、『*Software AG Infrastructure Administrator's Guide*』を参照してください。

Java サービスラッパー設定ファイル

My webMethods Server の場合、Java Service Wrapper の設定ファイルは、次のディレクトリにあります。

Software AG_directory \profiles\MWS_*serverName*\configuration

My webMethods Server の起動時に、ラッパー設定ファイルのプロパティにより、JVM の設定と Java Service Wrapperのログ機能および監視機能の動作が決定されます。

次の表では、各ファイルに含まれるプロパティについて説明します。

ファイル名	説明
wrapper.conf	My webMethods Server によってインストールされたプロパティ設定を含みます。 Software AG に依頼された場合を除いて、このファイルの内容を変更しないでください。
custom_wrapper.conf	wrapper.conf 内のインストール済み設定を変更するプロパティを含みます。 Java Service Wrapperのプロパティ設定を変更する必要がある場合、このファイル内で変更を行います。
wrapper.conf.template	My webMethods Server が更新またはアップグレードされるときに wrapper.conf ファイルに適用される設定が含まれます。このファイルは変更しないでください。 このファイルは以下の場所にあります。 *Software AG_directory* \MWS\server*serverName*\config\

JVM 設定

Java Service Wrapperが JVM を起動するとき、特に、Java ヒープのサイズおよびクラスパス内のディレクトリを指定する設定を提供します。

JVM 設定プロパティ

Java Service Wrapper設定ファイル内の `wrapper.java` プロパティは、My webMethods Server を実行する JVM の設定を決定します。

My webMethods Server がインストールする JVM プロパティ設定は、多くの環境に適しています。ただし、インストール済みの設定が環境の要件を満たしていない場合、これらのプロパティを変更できます。手順とのその他の情報については、97 ページの「My webMethods Server の JVM 設定」を参照してください。

ラッパーログ

310 ページの「サーバのログの制御」で説明しているように、My webMethods Server には独自のログ記録メカニズムがあります。また、Java Service Wrapperは、コンソール出力をログファイルに記録します。ログには、ラッパー自体からコンソールに送信された出力と、My webMethods Server を実行する JVM から送信された出力が含まれます。ラッパーログは、特に Windows サービスとして My webMethods Server を実行するときに役立ちます。これは、通常、コンソール出力がこのモードでは使用できないからです。

My webMethods Server の Java Service Wrapperログは、次のファイルに記録されます。

Software AG_directory \profiles\MWS_*serverName*\logs\wrapper.log

ログを表示するには、テキストエディタでログファイルを開きます。

ログプロパティ

ラッパー設定ファイル内の `wrapper.console` および `wrapper.log` プロパティにより、ラッパーログの内容、形式、および動作が決定されます。

My webMethods Server がインストールするログ設定は、多くの環境に適しています。ただし、インストール済みの設定が環境の要件を満たしていない場合、以下のプロパティを変更できます。手順とのその他の情報については、*Software AG Infrastructure Administrator's Guide*を参照してください。

次の表に使用可能なログプロパティを示します。

プロパティ	値
`wrapper.logfile.maxsize`	ログファイルの最大サイズです。
`wrapper.logfile.maxfiles`	古いログの保管数です。

障害の監視

Java Service Wrapperは、特定の状態に関して JVM を監視し、これらの状態を検出した時点で、JVM を再起動したり、その他のアクションを実行したりできます。

次の表では、My webMethods Server で使用したり、ユーザが設定したりできる障害監視機能について説明しています。

機能	有効であるか	ユーザ設定の可否
JVM タイムアウト	はい	[Yes]97 ページの「My webMethods Server の設定」で wrapper.ping プロパティを参照してください。
デッドロック検出	いいえ	[Yes]99 ページの「ラッパー JVM チェックの設定」で wrapper.check.deadlock プロパティを参照してください。
コンソールフィルタリング	はい	[Yes]96 ページの「コンソールフィルタリングプロパティ」を参照してください。

障害監視のオプションについては、『*Software AG Infrastructure Administrator's Guide*』を参照してください。

コンソールフィルタリングプロパティ

ラッパー設定ファイル内の wrapper.filter プロパティは、ラッパーが特定のメッセージに関してコンソールを監視するかどうかを決定します。My webMethods Server は、特定のメッセージに関してコンソールを監視する定義済みフィルタをインストールします。インストール済みフィルタについては、99 ページの「JVM メモリ不足チェックの設定」を参照してください。

以下のプロパティを使用して、Java Service Wrapperにその他のフィルタを追加できます。手順とのその他の情報については、*Software AG Infrastructure Administrator's Guide*を参照してください。

次の表に、コンソールフィルタリングを設定するカスタム JMS プロパティを示します。

プロパティ	値
wrapper.filter.trigger.*n*	コンソール出力で検出したいテキストストリング。
wrapper.filter.action.*n*	Java Service Wrapperがテキストストリングを検出した場合に実行されるアクション。
wrapper.filter.allow_wildcards.*n*	wrapper.filter.trigger.*n*に示されるワイルドカード文字を Java Service Wrapper で処理するかどうかを指定するフラグ (TRUE または FALSE)。
wrapper.filter.message.*n*	Java Service Wrapperがテキストストリングを検出した場合に表示されるメッセージ。

スレッドダンプの生成

Windows サービスとして My webMethods Server を実行している場合、Java Service Wrapper はスレッドダンプを生成するためのユーティリティを提供します。スレッドダンプは、スレッドのブロックまたはデッドロックを引き起こす可能性のある、スレッドの競合の問題を見つけるのに役立ちます。

Java ラッパーサービスを使用してスレッドダンプを生成する方法については、『*Software AG Infrastructure Administrator's Guide*』を参照してください。

My webMethods Server の JVM 設定

custom_wrapper.conf ファイルで My webMethods Server インスタンスの JVM 設定を変更できます。

JVM 設定を変更するには

1. テキストエディタで、サーバインスタンスの custom_wrapper.conf ファイルを開きます。ファイルは次の場所にあります。

   Software AG_directory \profiles\MWS_*serverName*\configuration\

2. 必要に応じてステートメントを追加または変更し、ファイルを保存します。

3. My webMethods Server を再起動します。

My webMethods Server の設定

custom_wrapper.conf ファイルを使用して、My webMethods Server インスタンスの ping タイムアウトを設定できます。次の表に、custom_wrapper.conf ファイルに追加するカスタム JVM プロパティを示します。

パラメータ	説明
`wrapper.ping.timeout=`*`value`*	整数。JVM からの ping 応答間の待機時間(秒)。デフォルト値は `120` です。「`0`」に設定した場合は、ラッパーはタイムアウトしませんが、サーバが一時停止しているかどうかを示す実際のインジケータも存在しなくなります。

初期メモリ制限および最大メモリ制限の設定

起動時に JVM によって割り当てられるメモリの初期量および最大量を設定できます。次の表に、Java サービスラッパーのデフォルトで設定されている初期メモリ制限および最大メモリ制限を示します。

パラメータ	値 (MB 単位)
wrapper.java.initmemory=	128
wrapper.java.maxmemory=	1024

JVM 自体に設定されているデフォルト値を使用するには、custom_wrapper.conf ファイルのプロパティにゼロ (0) を指定します。次に例を示します。

```
wrapper.java.maxmemory=0
```

注:
wrapper.java.initmemory メモリプロパティの値を指定する場合は、wrapper.java.maxmemory プロパティの値よりも小さくなるようにしてください。

UNIX で最大メモリ制限を増やすには

UNIX または Linux システムでは、My webMethods Server による初期化が失敗した場合、使用可能な最大メモリ量または最大オープンファイル制限を増やす必要がある場合があります。これは、複数の言語パックをインストールしている場合や、複数の webMethods コンポーネントがサーバにインストールされている場合に必要になる可能性があります。

custom_wrapper.conf ファイルで、最大メモリ制限を MB 単位で指定します。たとえば、使用可能な最大メモリを倍にするには、次の値を変更します。

```
wrapper.java.maxmemory=1024
```

次のように変更します。

```
wrapper.java.maxmemory=2048
```

各 UNIX プラットフォームには、オープンファイル制限を設定する独自の方法があります。次に、Linux で最大オープンファイル制限パラメータを増やす方法の例を示します。

```
ulimit -c unlimited ulimit -n 8192
```

Linux またはその他の UNIX プラットフォームでこれらの変更を行う場合は、管理者に問い合わせてください。

この custom_wrapper.conf の編集の詳細については、97 ページの「My webMethods Server の JVM 設定」を参照してください。

ラッパー JVM チェックの設定

Java Service Wrapperでは、JVM 内のデッドロックスレッドまたはメモリ不足状態の内部検出を実行できます。デフォルトでは、デッドロック検出は My webMethods Server サービス内で無効になっているため、この機能を使用する場合は、`custom_wrapper.conf` ファイルで手動で有効にする必要があります。

次の表では、JVM チェックを設定する追加パラメータについて説明します。

パラメータ	説明
`wrapper.check. deadlock=`*`true`*`|`*`false`*	ブーリアン型。JVM デッドロックチェックを有効または無効にします。デフォルト値は `FALSE` で、デッドロックチェックは無効になっています。
`wrapper.check.deadlock. interval=`*`interval`*	整数。デッドロックの検出間隔 (秒単位)。デフォルト値は `60` です。
`wrapper.check.deadlock. action=`*`action`*	文字列。デッドロックが発生したときに実行するアクション。デフォルト値は `RESTART` です。
`wrapper.check.deadlock. output=`*`output`*	文字列。ラッパーの出力。有効な値は次のとおりです。 ■ `FULL` ■ `SIMPLE` ■ `NONE` デフォルト値は `FULL` です。

JVM メモリ不足チェックの設定

JVM のコンソール出力をフィルタ処理することで、メモリ不足状態を制御および解決するプロパティを設定できます。これらのプロパティでは以下の構文が使用されます。

```
wrapper.filter.element.<n>=parameter
```

*n*の数は、1 から始めて、連続するエレメントごとに 1 ずつ増やします。追加パラメータのチェーンは、1 から始めて連続している必要があります。ただし、`wrapper.ignore_sequence_gaps` プロパティを有効にした場合は、シーケンスが破壊される可能性があります。

次の表に、使用可能なパラメータ、推奨値および対応する説明を示します。

パラメータ	推奨値	説明
wrapper.filter. trigger.*n*=	java.lang.OutOfMemory Error	JVMのコンソール出力での、メモリ不足状態のクエリーを有効にします。
wrapper.filter. action.*n*=	RESTART	設定済みフィルタに一致 (メモリ不足状態が発生) した場合のアクションを制御します。
wrapper.filter. message.*n*=	The JVM has run out of memory.	フィルタに一致 (メモリ不足状態が発生) した場合に表示されるメッセージを制御します。

オンプレミスインストールの起動の最適化設定

コンテナ内から起動する場合 My webMethods Server は最適化メカニズムを使って起動にかかる時間を短縮します。この最適化メカニズムは、Derby データベースのメモリ内インスタンスを起動して My webMethods Server をブートストラップし、ネットワーク遅延のオーバーヘッドを避けながら、データを実稼働データベースに転送します。デフォルトでは、オンプレミスの My webMethods Server インストールは最適化起動メカニズムを使用しませんが、次の JVM プロパティを custom_wrapper.conf ファイルに加えて、このメカニズムを有効にすることができます。

パラメータ	説明
mws.inmemory.init	ブーリアン型。起動時にデータベースクエリの最適化メカニズムを使用するかどうか。デフォルト値は「false」です。起動時の最適化を有効にするには、「true」に設定します。

追加の JVM パラメータ

一部のパラメータは、My webMethods Server ではなく JVM 自体に関連しています。My webMethods Server の custom_wrapper.conf ファイルにカスタムの JVM パラメータを設定するには、次の構文を使用します。

```
wrapper.java.additional.n=parameter
```

n の数は、1 から始めて、追加のエレメントごとに 1 ずつ増やします。追加パラメータのチェーンは、1 から始めて連続している必要があります。次に例を示します。

```
# Java Additional Parameters
wrapper.java.additional.1=-Dosgi.install.area="%OSGI_INSTALL_AREA%"
wrapper.java.additional.2=-Declipse.ignoreApp=true
wrapper.java.additional.3=-Dosgi.noShutdown=true
```

次の表に、JVM 設定のカスタマイズで使用可能な追加パラメータの一部を示します。

パラメータ	説明
`wrapper.java.additional.n=-DproxySet=true`	グラフィカルユーザインタフェースウィンドウを使用せずにプログラムを開始します。
`wrapper.java.additional.n=-Dhttp.proxyHost=host_name`	プロキシサーバのホスト名。
`wrapper.java.additional.n=-Dhttp.proxyPort=port_number`	プロキシサーバのポート番号。
`wrapper.java.additional.n=-Dhttp.proxyUser=user_name`	認証に使用されるユーザ名。
`wrapper.java.additional.n=-Dhttp.proxyPassword=password`	認証に使用されるパスワード。
`wrapper.java.additional.n=-Dhttp.nonProxyHosts=bypass_hosts`	プロキシをバイパスして直接到達する必要があるホストのリスト。エントリは、垂直バー (|) によって区切られます。

Axis HTTP クライアントの設定

My webMethods Server のポートレットから Integration Server の Web サービスに対して複数の同時 SOAP 要求を行うと、応答時間が長くなるか、要求がタイムアウトになる場合があります。このような場合は、custom_wrapper.conf ファイルに追加パラメータを含めることにより、My webMethods Server の組み込み Axis クライアントの接続設定を変更できます。

次の表に、My webMethods Server の Axis クライアントを設定するのに使用するカスタム JVM パラメータを示します。

パラメータ	説明
`axisclient.DefaultMaxConnections PerHost`	ホスト上のサービスへの同時接続の最大数。 デフォルト値は `10` です。
`axisclient.MaxTotalConnections`	単一のクライアントインスタンスのアクティブな接続の最大数。 デフォルト値は `100` です。

custom_wrapper.conf ファイルの変更の詳細については、94 ページの「Java サービスラッパー」を参照してください。

データベースに保存された設定ファイルの変更

デフォルトでは、ほとんどのサーバ設定は、ローカルファイルシステムではなく My webMethods Server データベースに保存されます。My webMethods Server 設定ファイルを編集するには、`mws getconfig` コマンドを使用して、データベースからファイルをダウンロードする必要があります。設定ファイルを変更する場合、My webMethods Server インスタンスを実行していても、停止していても問題はありませんが、データベースは実行している必要があります。

変更できる設定ファイルの詳細については、103 ページの「データベースに保存された設定ファイル」を参照してください。

My webMethods Server の設定ファイルを編集するには

1. コマンドラインプロンプトで、次のコマンドを入力してサーバの bin ディレクトリに移動します。

   ```
   cd Software AG_directory\MWS\bin
   ```

2. My webMethods Server データベースから設定ファイルを抽出するには、次のコマンドを入力します。

   ```
   mws getconfig fileName
   ```

   fileName は編集する設定ファイルの名前です。

   次に例を示します。

   ```
   mws getconfig cluster.xml
   ```

   ```
   mws getconfig logging.properties
   ```

3. ダウンロードして設定ファイルをテキストエディタで開き、必要に応じて変更します。

   ファイルは次の場所にあります。

   Software AG_directory \MWS\server*serverName*\config

4. 変更したファイルを My webMethods Server データベースに展開するには、次のコマンドを入力します。

   ```
   mws putconfig fileName
   ```

5. *serverName*\config ディレクトリからファイルを削除します。

   ファイルを削除しないと、サーバインスタンスはローカルバージョンの設定ファイルを引き続き使用します。

6. 次のコマンドを使用して、ノードを再起動します。

   ```
   mws -s serverName restart
   ```

設定ファイルに加えた変更は、再起動するまで適用されません。

データベースに保存された設定ファイル

次の表に、`mws getconfig` コマンドを使用してダウンロードできるファイルとその目的を示します。

設定ファイル	目的
cache.xml	キャッシュサイズを増加します。
cluster.xml	クラスタ内の各ノードの設定データを指定および保存します。
defaultPortletAppWeb.xml	Web アプリケーションのプロパティを変更します。
defaultPartitionPortlets.properties	デフォルトクラスタパーティションのフェーズプロバイダによって影響を受けるポートレットをリストします。複数のクラスタパーティションがある場合、それぞれに *partitionName*PortletsProperties.xml というファイル名を持つ複数のポートレットプロパティファイルを持つことができます。
email.properties	電子メールサーバを設定します。
iconMap.properties	ui\images ディレクトリにあるアイコンに MIME タイプをマッピングします。
jetty.xml	Jetty サーバを設定します。
log4j.init.properties	内部デバッグをオンまたはオフにします (Apache Log4J ログパッケージ)。
log4j.override.properties	Apache Log4J ログパッケージのプロパティを変更します。
logging.properties	ログフォルダとパターンをカスタマイズします。
mimeTypes.properties	MIME タイプをカスタマイズします。
perfUtil.properties	パフォーマンスサービスを有効および無効にします。
phaseProvider.xml	サーバのポーリング間隔を変更します。複数のクラスタパーティションがある場合、それぞれに *partitionName*PhaseProvider.xml というファイル名を持つ複数のフェーズプロバイダファイルを持つことができます。
storageConfig.xml	コンテンツサービスを設定します。
systemPaths.properties	ログまたは temp フォルダの場所を設定します。

設定ファイル	目的
systemProperties.properties	パーシストされたシステムプロパティを保存します。
defaultPortletAppWeb.xml	Web アプリケーションのプロパティを変更します。
email.properties	電子メールサーバを設定します。
websso.properties	Security Assertion Markup Language (SAML) を使用して、サードパーティ ID プロバイダ (IDP) のシングルサインオンを設定します。

注:
ファイル名およびファイルの内容は大文字と小文字が区別されます。

Solaris、HP-UX または Linux で My webMethods Server を 32 ビットで実行するための設定

Solaris、HP-UX または Linux システムでは、コンピュータのアーキテクチャが 32 ビットと 64 ビットのどちらであるかを指定する必要があります。デフォルトでは、My webMethods Server は 64 ビットモードで実行するように設定されています。サーバを 32 ビットモードに切り替えることができます。

Solaris または HP-UX で My webMethods Server が 32 ビットモードで実行されるように設定するには

1. 次のコマンドを使用して My webMethods Server を停止します。

   ```
   mws -option stop
   ```

   オプションの詳細については、72 ページの「My webMethods Server の基本のコマンドライン構文」を参照してください。

2. 次のファイルをテキストエディタで開きます。

   Software AG_directory /MWS/bin/setenv.sh

3. 次のテキストを探します。

   ```
   rem to use 64 bit JVM uncomment next line
   set JAVA_D64=-d64
   ```

   2 行目をコメントにします。

   ```
   rem to use 64 bit JVM uncomment next line
   rem set JAVA_D64=-d64
   ```

4. ファイルを保存して閉じます。

5. 次のコマンドを使用して OSGi プロファイルを更新します。

```
mws -option update-osgi-profile
```

6. My webMethods Server を再起動します。

HTTP リスナーの単一 IP アドレス使用の設定

My webMethods Server では、そのリスナーポートは使用可能なすべての IP アドレスでバインドされます。複数のネットワークインタフェースを持つサーバハードウェアに My webMethods Server をインストールした場合、HTTP リスナーポートを単一 IP アドレスにバインドできます。

Jetty リスナーの IP アドレス使用の設定

Jetty では、My webMethods Server 上に複数の受信待機ポートが作成されます。

- HTTP および HTTPS 用のポート
- HTTP および HTTPS 用のランダムなソケットアクセプタポート

複数のネットワークインタフェースを持つサーバハードウェアに My webMethods Server をインストールした場合、HTTP リスナーポートを単一 IP アドレスにバインドできます。次の手順を使用して、特定の IP アドレスを、Jetty リスナーによって使用されるポートに設定します。

Jetty 受信待機ポートを設定するには、`custom_wrapper.conf` 設定ファイルで IP アドレスを指定する必要があります。

Jetty リスナーによって特定の IP アドレスが使用されるように設定するには

1. テキストエディタで、サーバインスタンスの `custom_wrapper.conf` 設定ファイルを開きます。

2. 次の JVM プロパティを追加します。

   ```
   wrapper.java.additional.nnn=-Djetty.host=IP_address
   ```

   ここで、*nnn* は JVM プロパティの連続する数であり、*IP_address* は HTTP リスナーポートをバインドする IP アドレスです。

3. ファイルを保存して、My webMethods Server を再起動します。

JCR リポジトリ RMI サーバの IP アドレスの設定

Java コンテンツリポジトリ (JCR) を使用する場合、JCR への接続は、My webMethods Server と通信するために RMI を使用して実装されます。デフォルトの JCR リスナーポートは 10999 です。

複数のネットワークインタフェースを持つサーバハードウェアに My webMethods Server をインストールした場合、HTTP リスナーポートを単一 IP アドレスにバインドできます。

custom_wrapper.conf ファイルで、次のパラメータの値を指定します。

```
wrapper.java.additional.204=-Dmws.jcr.rmi.bind.address=IP_address|host_name
```

ここで、*IP_address* または *host_name* は JCR リポジトリ RMI サーバリスナーの IP アドレスまたはホスト名です。

このファイルの編集の詳細については、97 ページの「My webMethods Server の JVM 設定」を参照してください。

GLUE Web サービスレジストリでの IP アドレス使用の設定

My webMethods Server のインスタンスに Central Configuration がインストールされている場合、GLUE Web サービスレジストリは 16000～16025 の範囲のポートを使用するように作成されました。特定の IP アドレスを使用するには、`GlueServiceRegistryProperties.xml` ファイルで IP アドレスを設定します。

複数のネットワークインタフェースを持つサーバハードウェアに My webMethods Server をインストールした場合、HTTP リスナーポートを単一 IP アドレスにバインドできます。次の手順を使用して、特定の IP アドレスを、GLUE Web サービスレジストリによって使用されるポートに設定します。

特定の IP アドレスを設定するには

1. My webMethods Server のインストールディレクトリに進み、`MWS/server/default/config/engine` に移動して、`GlueServiceRegistryProperties.xml` ファイルをエディタで開きます。

2. 次のプロパティを探します。

   ```
   <entry key="service.address">0.0.0.0</entry>
   ```

3. 値 0.0.0.0 を IP アドレスで置き換えます。

マルチホームマシンでの My webMethods Server の設定

マルチホームマシン (複数のネットワークインタフェースと IP アドレスを持つマシン) で My webMethods Server を設定する場合は、以下の手順に従って My webMethods Server リスナーをシングル IP アドレスにバインドします。

マルチホームマシンでの My webMethods Server を設定するには

1. My webMethods Server をシャットダウンします。

2. 次の説明に従って、HTTP リスナー と HTTPS リスナーを IP アドレスにバインドします。サーバインスタンスごとにこの手順を実行します。

a. *Software AG_directory* /profiles/*MWS_instanceName*/configuration ディレクトリに移動し、`custom_wrapper.conf` ファイルをテキストエディタで開きます。

b. 次のカスタム JVM プロパティを追加します。

```
wrapper.java.additional.nnn=-Djetty.host=IP_address
```

ここで、*nnn* は JVM プロパティの連続する数であり、*IP_address* は HTTP リスナーポートをバインドする IP アドレスです。

c. `custom_wrapper.conf` ファイルを保存します。

3. 各クラスタノードについて、 *Software AG_directory* /MWS/server/*serverName*/config/engine ディレクトリに移動し、GlueServiceRegistryProperties.xml ファイルを開き、次のエレメントで IP アドレスを指定します。

```
<entry key="service.address">IP_address</entry>
```

4. 各クラスタノードについて、以下のように SOAP モニタ および JCR サーバを IP アドレスにバインドします。

a. *Software AG_directory* /profiles/MWS_*serverName*/configuration ディレクトリに移動し、wrapper.conf ファイルと custom_wrapper.conf ファイルを開きます。

b. 次のパラメータの値を指定します。

```
wrapper.java.additional.203=-Dsoap.monitor.bind.address=
 IP_address|host_name
```

ここで、*IP_address* または *host_name* は SOAP モニタポートレットリスナーの IP アドレスまたはホスト名です。

```
wrapper.java.additional.204=-Dmws.jcr.rmi.bind.address=
 IP_address|host_name
```

ここで、*IP_address* または *host_name* は JCR リポジトリ RMI サーバリスナーの IP アドレスまたはホスト名です。

5. JMX サーバは wrapper.conf ディレクティブを使用して IP アドレスをバインドしないため、代わりに JMX コネクタプロパティファイルでバインドを指定する必要があります。各クラスタノードについて、以下の手順に従います。

a. *Software AG_directory* /profiles/MWS/*serverName*/configuration/com.softwareag.platform.config.propsloader ディレクトリに移動し、com.softwareag.jmx.connector.pid-500*number*.properties ファイルをバックアップします。

b. ファイルを開き、次の行を追加します。

```
host=IP_address
```

6. My webMethods Server を再起動します。

起動時に診断を実行するかどうかの設定

デフォルトでは、起動時に My webMethods Server によって以下の診断アクションが実行されます。

- My webMethods Server に適用されている修正の判別
- 潜在的なデータベースエラーを検索する dbintegritycheck ツールの実行

phaseProvider.xml 設定ファイルを更新することによって、これらの診断アクションのいずれかまたは両方が My webMethods Server によって実行されないようにすることができます。phaseProvider.xml 設定ファイルは、データベースまたは *My webMethods Server_directory* \server*serverName*\config にあります。

My webMethods Server 起動時におけるインストール済み修正の表示の防止

My webMethods Server では、サーバ起動時にインストール済み修正が判別され、インストール済み修正のリストが _full_.log ファイルに記録されます。

修正のリストを判別するのに数分かかる場合があるため、サーバ起動時のこのアクションを無効にする場合があります。起動時にインストール済み修正が My webMethods Server によって表示されないようにするには、phaseProvider.xml 設定ファイルの `startupDiagnostics` セクションを次に示すように編集します。

```
<Phase class="com.webmethods.portal.system.init.impl.DefaultPhase"
               enabled="true" name="startupDiagnostics">
      <PhaseInfo class="com.webmethods.portal.system.impl.UpdatesInfo"
               enabled="false" name="updatesInfo"/>
   .
   .
   .
</Phase>
```

My webMethods Server 起動時における dbintegritycheck の実行の防止

My webMethods Server の起動時には、dbintegritycheck ツールが自動的に実行され、結果が _full_.log ファイルに記録されます。dbintegritycheck ツールでは、My webMethods Server に展開されている Dynamic Business Object (DBO) を確認し、DBO が誤ってインストールまたはアップグレードされた場合に発生した可能性がある潜在的なエラーを検索します。

このアクションは有効にしておくことをお勧めしますが、起動時に dbintegritycheck が My webMethods Server によって実行されないようにすることができます。そのためには、phaseProvider.xml 設定ファイルの `startupDiagnostics` セクションを次に示すように編集します。

```
<Phase class ="com.webmethods.portal.system.init.impl.DefaultPhase"
      enabled="true" name="startupDiagnostics">
   <PhaseInfo
      class="com.webmethods.portal.system.impl.DbIntegrityCheckLauncher"
      enabled="false" name="dbIntegrityCheck"/>     .
   .
   .
</Phase>
```

II My webMethods Administrator の機能

8 My webMethods Server の設定の管理 113

9 ユーザ、グループ、および役割の検索 145

10 ユーザとグループの管理 153

11 権限の管理 175

12 役割および My webMethods へのアクセスの管理 189

13 My webMethods Server のクラスタリング 205

8 My webMethods Server の設定の管理

- ディレクトリサービスの管理 114
- 外部ディレクトリサービスの管理 116
- 外部データソースの管理 133
- 電子メール設定の管理 141
- カレンダーの管理 143

ディレクトリサービスの管理

ディレクトリは、それぞれが名前や電子メールアドレスなどの属性のセットを持つエントリ (この場合は個々のユーザ) のコレクションを含むという点で、データベースに似ています。ディレクトリサービスには、ディレクトリのエントリに関する情報を配信するメカニズムが用意されています。

My webMethods Server には、内部ディレクトリサービスが含まれています。ただし、LDAP (Lightweight Directory Access Protocol) などの外部ディレクトリサービスを使用する場合は、外部ディレクトリサービスのユーザおよびグループ情報にもアクセスできるように My webMethods Server を設定することができます。

内部ディレクトリサービス

My webMethods Server インストールのデフォルトで提供される内部ディレクトリサービスは、システムディレクトリサービスと呼ばれています。My webMethods Server では、ユーザ、グループ、および役割に関してユーザがシステムディレクトリサービスに定義した情報は、My webMethods Server データベースに保存されます。必要なユーザおよびグループの数がそれほど多くない場合は、システムディレクトリサービスを使用してください。

次の表に、システムディレクトリサービス内でデフォルトで定義されているユーザおよびグループを示します。

ユーザ名	説明
My webMethods Administrator	My webMethods のデフォルト管理者。このユーザはユーザ管理機能を実行し、外部ディレクトリサービスを管理できます。インストール時のユーザ ID は「`Administrator`」で、このユーザアカウントのログインは無効になっています。 **重要:** `SysAdmin` ユーザがログインし、このアカウントで My webMethods Server にアクセスするためのパスワードを変更する必要があります。
System Admin	My webMethods Server のシステム管理者。このユーザは、分析、設定、コンテンツ、およびユーザ管理を含む、My webMethods Server の管理を行うことができます。インストール時のユーザ ID は「`SysAdmin`」で、パスワードはイントール中に指定します。この管理者は My webMethods ユーザインタフェースを使用しません。 **重要:** インストール中に選択したオプションによって、My webMethods Server は、このユーザの初回ログイン時に新規パスワードの設定をうながすプロンプトを表示することがあります。
Deleted Items	管理特権を持つユーザがシステムから削除された場合に、このユーザが実行した作業を保存するために My webMethods Server が内部で使用する

ユーザ名	説明
	ユーザアカウント。インストール時のユーザアカウントは「DeletedItems」、パスワードは「「manage」」です。 **重要:** このユーザのパスワードは変更してください。
Designer	My webMethods Server のページ設計者。このユーザは、システム管理者と同じ My webMethods Server 特権を所有します。インストール時のユーザ ID は「`Designer`」で、このユーザアカウントのログインは無効になっています。このアカウントで実行可能な機能の詳細については、サーバページ*の開発*を参照してください。 **重要:** `SysAdmin` ユーザがログインし、このアカウントで My webMethods Server にアクセスするためのパスワードを変更する必要があります。
Guest	匿名のユーザ。このユーザは、ログインページなどの、Anonymous アクセスを許可するページを読み取ることができます。管理者によって権限が明示的に付与されていない場合、コンテンツの読み取り、変更または削除はできません。インストール時のこのユーザのユーザ ID は「`Guest`」です。
webMethods System	My webMethods Server により Web サービスの起動に内部的に使用されるユーザアカウント。My webMethods Server では、1 つのサーバから別のサーバに対する Web サービス認証にこのアカウントが使用されます。インストール時のユーザ ID は「`WEBM_SYSUSER`」で、このユーザアカウントのログインは無効になっています。 **重要:** このユーザアカウントは削除しないでください。`SysAdmin` ユーザがログインし、このアカウントで My webMethods Server にアクセスするためのパスワードを変更する必要があります。このアカウントのパスワードを変更した場合は、このアカウントを使用して webMethods と通信する My webMethods Server アプリケーションの管理者に変更を知らせる必要があります。
webMethods Cluster	My webMethods Server によりクラスタ内のサーバ間の認証に内部的に使用されるユーザアカウント。インストール時のユーザアカウントは「`WEBM_CLUSTERUSER`」で、パスワードは「「manage」」です。 **重要:** このユーザアカウントは削除しないでください。クラスタの 1 つのノードでこのユーザのパスワードを変更し、パスワードを有効にするためにすべてのノードを再起動します。

セキュリティ保護のため、SysAdmin ユーザは管理権限を持つシステムユーザのデフォルトパスワードを必ず変更してください。これらのユーザのパスワードを変更する方法の詳細については、156 ページの「ユーザ情報の編集」を参照してください。

外部ディレクトリサービス

システムディレクトリサービスに加えて、My webMethods Server では複数の外部ディレクトリサービスをサポートできます。このため、非常に大規模で多様なユーザグループを管理できます。1 つ以上のディレクトリサービスを使用している場合、My webMethods Server をこれらのサービスを接続できます。さらに、データベースをディレクトリサービスとして使用できます。または、カスタムサービスを作成して、My webMethods Server 内でディレクトリプロバイダを外部のディレクトリサービスに接続できます。そのようなサービスの詳細については、*webMethods CAF and My webMethods Server Java API Reference*を参照してください。

注:
ログイン時に、長時間かかるユーザ検索またはグループ検索が役割キャッシュまたはグループキャッシュ計算に含まれていると、My webMethods Server および接続先の LDAP サーバでパフォーマンスが低下します。詳細については、342 ページの「役割キャッシュライフサイクルまたはグループキャッシュライフサイクル計算の設定」を参照してください。

内部システムディレクトリサービスの設定

内部システムディレクトリサービスについては、My webMethods Server データベースの設定以外の作業は必要ありません。データベースの設定は、『*Installing Software AG Products*』で説明しているように、My webMethods Server のインストール時に行う必要があります。

外部ディレクトリサービスの管理

外部 LDAP、ADSI、または ADAM ディレクトリサービスの設定

外部の LDAP、ADSI または ADAM ディレクトリサービスを使用するように My webMethods Server を設定するには、以下の手順に従います。

外部の LDAP、ADSI、または ADAM ディレクトリサービスを設定するには

1. 次のページに移動します。
   - My webMethods Administrator の場合: [ナビゲーション] > [アプリケーション] > [管理] > [My webMethods] > [ディレクトリサービス] > [ディレクトリサービスの一覧表示]
   - sysadmin の場合: [管理] > [ユーザ管理] > [ディレクトリサービス管理] > [ディレクトリサービスの一覧表示]
2. [新規ディレクトリサービスの作成] タブで、次のオプションからディレクトリサービスのタイプを選択します。
   - **LDAP** - Lightweight Directory Access Protocol。

- **ADSI** - Active Directory Service Interfaces。
- **ADAM** - Active Directory Application Mode。

3. [次へ] をクリックします。
4. 新規ディレクトリサービスを設定するフォームを入力します。

   使用可能な設定プロパティの詳細については、117 ページの「LDAP、ADSI、ADAM ディレクトリサービスのプロパティ」を参照してください。
5. ページの下部で、[**終了**] をクリックします。

ヒント:
設定をテストするには、外部ディレクトリサービス内に定義されているユーザまたはグループを検索するクエリーを実行します。クエリーの実行方法の詳細については、146 ページの「既存のユーザ、グループ、または役割の検索」を参照してください。

LDAP、ADSI、ADAM ディレクトリサービスのプロパティ

LDAP、ADSI、または ADAM タイプの外部ディレクトリサービスを設定する場合、以下のプロパティを設定できます。

[全般] セクション

次の表に、[全般] セクションで設定するディレクトリサービスのプロパティを示します。

プロパティ	説明
[名前]	必須。外部ディレクトリサービスを識別する名前。My webMethods Server は、この名前を使用してユーザインタフェース内で外部ディレクトリサービスを表示します。
[説明]	外部ディレクトリサービスに関する説明的なコメント。
[キーワード]	外部ディレクトリサービスの検索に使用する 1 つ以上のキーワード。

[キャッシュ] セクション

次の表に、[キャッシュ] セクションで設定するディレクトリサービスのプロパティを示します。

プロパティ	説明
[キャッシュ容量]	必須。キャッシュに入れるデータベースクエリーの数。デフォルトは `1000` です。 キャッシュされたクエリーの数が指定された容量に達した場合、My webMethods Server はキャッシュエントリを最も古いエントリから順に削除します。

プロパティ	説明
[キャッシュタイムアウト]	必須。キャッシュ容量を超えない限り、クエリーがキャッシュに残る時間。デフォルトは 1 時間です。 指定されたキャッシュ容量に達していなくても、キャッシュタイムアウトの期限が切れると、My webMethods Server はキャッシュエントリを削除します。

My webMethods Server はすべてのキャッシュをメモリに保存し、再起動されたときにすべてのキャッシュエントリを消去します。

[接続情報] セクション

次の表に、[接続情報] セクションで設定するディレクトリサービスのプロパティを示します。

プロパティ	説明
[サービスの有効化]	ディレクトリサービスを有効および無効にします。デフォルトは [はい (このサービスを有効にする)] です。
[接続エラーのしきい値]	必須。このサービスを無効にするまでに発生する最大接続エラー数です。デフォルトは `10` です。
[自動再接続]	接続エラーのしきい値に達した後にサービスが無効になった場合、またはネットワーク障害や定期メンテナンスのためにサーバへの接続が失われた場合に、ディレクトリサーバへの再接続を試行します。デフォルトでは有効になっています。
[自動再接続間隔]	次の再接続試行まで待機する期間 (秒)。デフォルトは `6` です。
[プロバイダ URL]	必須。次の構文を使用した外部ディレクトリサーバの URL。 `ldap://host_name:port_number`
[基本 DN]	必須。ディレクトリサーバにクエリーを行う際に使用するルート識別名。たとえば、「`ou=mywebMethods,o=webmethods.com`」と入力します。
[ユーザ DN]	ユーザの検索やロードの際に使用する追加のユーザ DN。
[グループ DN]	ユーザの検索やロードの際に使用する追加のグループ DN。
[セキュリティプリンシパル]	必須。外部ディレクトリサーバへのログインに必要な識別名。
[セキュリティクレデンシャル]	必須。外部ディレクトリサーバへのログインに必要なパスワード。
[フェイルオーバー URL]	[プロバイダ URL] フィールドで指定したプライマリ LDAP サーバの障害の際に My webMethods Server がフェイルオーバーで使用する、別の LDAP サーバの URL。複数の場合はスペースで区切ります。

プロパティ	説明
[検索タイムアウト]	必須。LDAP 検索クエリーが期限切れになるまでの最大時間(秒)デフォルトは `0` で、クエリーが期限切れになりません。 custom_wrapper.conf ファイルで接続タイムアウトを設定しない限り、My webMethods Server は LDAP サーバへの接続タイムアウトの定義に [検索タイムアウト] の値を使用します。LDAP サーバの接続タイムアウトの設定の詳細については、122 ページの「LDAP ディレクトリサービスの接続タイムアウトの設定」を参照してください。
[デフォルトワイルドカード検索を有効にする]	必須。ディレクトリ検索でワイルドカード文字の使用を有効または無効にします。デフォルトは [はい (デフォルトワイルドカード検索を有効化)] です。 ワイルドカード検索を無効にすると、大規模なサーバの場合にパフォーマンスが向上します。ワイルドカードを使用する場合、サーバでは内部インデックスが検索に使用されません。
[グループ全域ディレクトリサービスを有効にする]	必須。My webMethods Server で設定されたすべての外部ディレクトリサービスに対してグループメンバーシップのクエリーを行うかどうかを示します。このオプションを有効にすると、すべてのディレクトリサービスに対してグループメンバーシップのクエリーが行われるため、ログインパフォーマンスが著しく低下します。デフォルトは [いいえ、グループ全域ディレクトリサービス] です。 詳細については、123 ページの「ディレクトリサービス間のグループメンバーシップ」を参照してください。
[GroupQuickSearch を有効にする]	Active Directory で必須。再帰検索の代わりに 1 つのクエリーを使用している Active Directory ユーザのグループメンバーシップを特定するかどうかを示します。このオプションを有効にすると、検索では 1 つのクエリーを使用するため、ログインパフォーマンスが向上します。ユーザは、Active Directory セキュリティグループまたは標準グループのいずれかに属している必要があります。デフォルトは [無効] です。
[Active Directory ドメイン URL]	Active Directory にのみ適用されます。複数の Active Directory サブドメイン URL をスペースで区切って指定します。

[高度なオブジェクトフィルタ] セクション

次の表に、[高度なオブジェクトフィルタ] セクションで設定するディレクトリサービスのプロパティを示します。

プロパティ	説明
[ユーザオブジェクトフィルタ]	ユーザの検索時にすべてのクエリーに対して My webMethods Server が適用する LDAP フィルタ。My webMethods Server で公開されるオブジェクトのタイプを制限する技術 LDAP クエリーを使用します。 **注:** My webMethods Server ディレクトリのデバッグログを調べて、クエリーが正しく動作していることを確認することをお勧めします。
[グループオブジェクトフィルタ]	グループの検索時にすべてのクエリーに対して My webMethods Server が適用する LDAP フィルタ。My webMethods Server で公開されるオブジェクトのタイプを制限する技術 LDAP クエリーを使用します。 **注:** My webMethods Server ディレクトリのデバッグログを調べて、クエリーが正しく動作していることを確認してください。
[ネストされたグループを使用する]	ネストされた LDAP グループ内での検索を有効または無効にします。デフォルト値は [いいえ (ネストされたグループを使用しない)] です。
[仮想リスト表示制御 (VLV) を使用する]	LDAP クエリーのオブジェクトのサブセットを取得するために、仮想リスト表示制御 (VLV) を使用するかどうかを指定します。デフォルト値は [いいえ (VLV 制御を使用しない)] です。LDAP サーバ制御の自動設定が無効になっている場合にのみ適用されます。
[サーバ側ページング制御を使用する]	LDAP クエリーの結果をページングするために、サーバ側ページング制御を使用するかどうかを指定します。デフォルト値は [いいえ (ページング制御を使用しない)] です。LDAP サーバ制御の自動設定が無効になっている場合にのみ適用されます。
[サーバ側並べ替え制御を使用する]	LDAP クエリーの結果を特定の順番に並べ替えるために、サーバ側並べ替え制御を使用するかどうかを指定します。デフォルト値は [いいえ (並べ替え制御を使用しない)] です。LDAP サーバ制御の自動設定が無効になっている場合にのみ適用されます。
[サーバ側制御を自動設定]	My webMethods Server による LDAP サーバ制御の自動設定を有効または無効にします。デフォルト値は [はい (制御を自動設定)] です。

[ユーザ属性] セクション

次の表に、[ユーザ属性] セクションで設定するディレクトリサービスのプロパティを示します。

プロパティ	説明
[ユーザオブジェクトクラス]	必須。外部ディレクトリサービスのユーザオブジェクトクラス属性。デフォルトは「person」です。
[ユーザ ID]	必須。外部ディレクトリサービスのユーザ ID 属性。デフォルトは「uid」です。
[名]	必須。外部ディレクトリサービスの名属性。デフォルトは「sn」です。
[姓]	必須。外部ディレクトリサービスの姓属性。デフォルトは「givenName」です。
[氏名]	必須。外部ディレクトリサービスの氏名属性。デフォルトは「cn」です。
[電子メール]	必須。外部ディレクトリサービスの電子メールアドレス属性。デフォルトは「mail」です。
[パスワード]	必須。外部ディレクトリサービスのパスワード属性。
[ユーザが無効]	ユーザが無効化されていると識別する外部ディレクトリサービス内の属性の名前。デフォルトは「true」です。
[「ユーザが無効」値の正規表現]	外部ディレクトリサービスの「ユーザが無効」属性を検証するために使用される正規表現。
[UUID]	ユーザの汎用的に一意な識別子として使用される属性の名前。最大 128 文字の文字列 (たとえば「cn」または「email」) を指定できます。 **注:** 既存のディレクトリサービスの **[UUID]** の値を変更する場合、UserDirectory_UpdateUUID ユーティリティを実行して、ディレクトリサービスのユーザの UUID 値を更新する必要があります。 詳細については、124 ページの「ユーザへの Universally Unique Identifier (UUID) の設定」を参照してください。

[グループ属性] セクション

次の表に、[グループ属性] セクションで設定するディレクトリサービスのプロパティを示します。

プロパティ	説明
[グループオブジェクトクラス]	必須。外部ディレクトリサービスのグループオブジェクトクラス属性。デフォルトは「groupofuniquenames」です。

プロパティ	説明
[グループ ID]	必須。外部ディレクトリサービスのグループ ID 属性。デフォルトは「cn」です。
[グループ名]	必須。外部ディレクトリサービスのグループ名属性。デフォルトは「cn」です。
[グループのメンバー]	必須。外部ディレクトリサービスのグループのメンバー属性。デフォルトは「uniquemember」です。
[グループ電子メール]	必須。外部ディレクトリサービスのグループ電子メール属性。デフォルトは「mail」です。

[接続プール] セクション

次の表に、[接続プール] セクションで設定するディレクトリサービスのプロパティを示します。

プロパティ	説明
[接続数の最小値]	外部ディレクトリサーバに対して常に開いておく最小接続数。デフォルトは 1 です。
[接続数の最大値]	外部ディレクトリサーバに対して常に開いておく最大接続数。デフォルトは 20 です。
[最大接続時間]	外部ディレクトリサーバへの接続を再利用するまで開いておく最大時間。LDAP 接続が検索プロセス中に開いておくように、この値はサーバによって LDAP 検索ごとにリセットされます。デフォルトは 10 分です。
[クリーンアップの間隔]	期限切れの LDAP 接続をクリーンアップする時間間隔。デフォルトは 1 分です。

注:
一部の LDAP 実装では、ページング Cookie は特定の LDAP 接続にバインドされます。最大接続数プロパティの値が同時 LDAP 検索を処理するのに十分に大きく、最大接続時間プロパティの値が指定した時間範囲内で検索を終了させるのに十分に長いことを確認します。

LDAP ディレクトリサービスの接続タイムアウトの設定

デフォルトでは、My webMethods Serverは [検索タイムアウト] プロパティの値を使用して、LDAP サーバへの接続のタイムアウトを定義します。ただし、[検索タイムアウト] プロパティの主な目的は、LDAP クエリーのタイムアウトを指定することです。

場合によって、[検索タイムアウト] の値とは異なる接続タイムアウト値を指定する必要があることがあります。これを行うには、My webMethods Server の custom_wrapper.conf ファイルで -Dcom.webmethods.portal.portlet.wm_xt_ldapdirsvc.service.connection.timeout パラメータを設定します。LDAP ディレクトリサービスの接続タイムアウトパラメータのデフォルト値は 10 秒です。接続タイムアウトパラメータを設定せず、[検索タイムアウト] の値が 0 に設定されてい

る場合、My webMethods Server はデフォルトの接続タイムアウト値である 10 秒を使用します。

検索タイムアウトおよびLDAPディレクトリサービスの設定の詳細については、「116 ページの「外部 LDAP、ADSI、または ADAM ディレクトリサービスの設定」」を参照してください。

LDAP ディレクトリサービスの接続タイムアウトを設定するには、以下の手順に従います。

1. *Software AG_directory* \profiles\MWS_*serverName*\configuration に移動して、テキストエディタで custom_wrapper.conf ファイルを開きます。

2. 次のパラメータを追加します。

```
wrapper.java.additional.number=
-Dcom.webmethods.portal.portlet.wm_xt_ldapdirsvc.service.connection.timeout=
time_in_seconds
```

   *number*は一意のシーケンス番号で、ファイルに既存しているパラメータの数で決まります。

3. ファイルを保存して、My webMethods Server を再起動します。

ディレクトリサービス間のグループメンバーシップ

My webMethods Server 上に複数の LDAP、ADSI または ADAM ディレクトリサービスを設定している場合、サーバでは、設定したすべてのディレクトリサービスに対して、グループメンバーシップをクエリーすることができます。この機能は、ディレクトリツリーのある分岐内にユーザがある必要があり、その分岐がグループから離れているために、ユーザとグループの両方を網羅するルートで 1 つのみのディレクトリサービスをマウントすることが非効率である場合に役立ちます。代わりに、2つのディレクトリサービスを設定できます。1つのサービスがユーザ分岐のルートを指し、もう 1 つのサービスがグループ分岐のルートを指します。たとえば、以下のようなディレクトリ構造があるとします。

o=MyCompany, ou=Americas, ou=US, ou=Groups
o=MyCompany, ou=Americas, ou=US, ou=Users
o=MyCompany, ou=Americas, ou=Mexico, ou=Groups
o=MyCompany, ou=Americas, ou=Mexico, ou=Users
and so forth....

My webMethods Server は、o=MyCompany を指す 1 つのディレクトリ サービスでは適切に実行されません。代わりに、管理者は ou=Americas およびその他の 地域 OU を指す複数のディレクトリサービスを作成できます。ただし、 グループに複数の地域のメンバーがいる場合があるとします。これは、国際的な 大組織では一般的です。その場合、グループのメンバーシップを複数の ディレクトリサービスにまたがって指定できます。

設定したすべてのディレクトリサービス間でグループメンバーシップをクエリーできるようにするには、各ディレクトリサービスの [**グループ全域ディレクトリサービスを有効にする**] を [**はい、グループ全域ディレクトリサービスを有効にする**] に設定します。

注:

この機能を有効にすると、ログインパフォーマンスが著しく低下する可能性があります。

Universally Unique Identifier (UUID) の解決の有効化

My webMethods Server で Universally Unique Identifier (UUID) の解決を有効にすることができます。UUID の解決を有効にする前に、UserDirectory_UpdateUUIDPortlet.war ファイルをインストールします。UserDirectory_UpdateUUIDPortlet.war ファイルのインストールの詳細については、124 ページの「ユーザへの Universally Unique Identifier (UUID) の設定」を参照してください。

My webMethods Server で UUID の解決を有効にするには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [My webMethods] > [ディレクトリサービス]
   - システム管理者として次のように選択します。[**フォルダ**] > [管理用フォルダ] > [Administration Dashboard] > [ユーザ管理] > [ディレクトリサービス管理]
2. [**UUID の解決を有効にする**] タブをクリックします。
3. [**UUID の解決を有効にする**] チェックボックスを選択し、[**適用**] をクリックします。

ユーザへの Universally Unique Identifier (UUID) の設定

デフォルトでは、My webMethods Server で設定された外部ディレクトリサービスは、ユーザを識別するために DN を使用します。この場合、ユーザが 1 つの組織単位から別の組織単位に転送されると、My webMethods Server はユーザの DN が変更されたためにそのユーザを認識することができません。

たとえば、commonName が James であるユーザ (cn=James) が ABC 組織 (o = ABC) のマーケティングユニット (ou=marketing) で働いているとして、My webMethods Server がLDAP を使用するように設定されている場合、James の DN は dn: cn=James,ou=marketing,o=ABC になります。ジェームズがファイナンスユニットに転送されると、James の DN は dn: cn=James,ou=finance,o=ABC になります。DN の変更後、My webMethods Server は James を認識できません。James がメンバーである役割も LDAP で James を検索できず、「unable to get role members- cn=James,ou=marketing,o=ABC」という警告をスローします。

UUID を使用してユーザを識別するように My webMethods Server を設定できます。My webMethods Server システム管理者は、システムのメンテナンスに適した時間に UUID を設定する必要があります。

重要:

ユーザベースが大規模な場合は、UserDirectory_UpdateUUID ユーティリティは、外部ユーザディレクトリを My webMethods Server と再設定するために、長時間かけることがあります。無効なユーザレコードを削除するには、UserDirectory_UpdateUUID ユーティリティを実行する前に、クリーンアップユーティリティを実行します。

UUID を使用してユーザを識別するように My webMethods Server を設定するには

1. UUID の解決を有効にします。UUID の解決の有効化の詳細については、124 ページの「Universally Unique Identifier (UUID) の解決の有効化」を参照してください。

2. 外部ディレクトリサービスを設定して、**UUID** プロパティを指定します。外部ディレクトリサービスの設定の詳細については、116 ページの「外部 LDAP、ADSI、または ADAM ディレクトリサービスの設定」を参照してください。

   重要:
   既存のディレクトリサービスの UUID プロパティの値を変更する場合、UserDirectory_UpdateUUID ユーティリティを実行して、My webMethods Server 内でディレクトリサービスのユーザの UUID 値を更新する必要があります。

3. UserDirectory_UpdateUUIDPortlet.war ファイルが *Software AG_directory*\MWS\components\extras\userdirectory フォルダにあることを確認してください。

4. [**フォルダ**] > [管理用フォルダ] > [Administration Dashboard] > [設定] > [インストール管理] > [Extras] > [userdirectory] > [UserDirectory_UpdateUUIDPortlet.war] に移動します。

5. [**インストールコンポーネント**] をクリックします。

6. [**フォルダ**] > [システム] > [ポートレット] > [UserDirectory_UpdateUUIDPortlet] に移動して、ブラウザで UserDirectory_UpdateUUID ユーティリティポートレットを開きます。

7. [**Update UUID**] をクリックして、設定済みのすべてのディレクトリサービスユーザについて、My webMethods Server で UUID 値を更新します。

   次のような場合、UUID フィールド値は NULL になります。

   - 無効なユーザ
   - システムユーザ
   - DB ユーザ

   重要:
   UserDirectory_UpdateUUID ユーティリティは一回のみ実行することをお勧めします。

無効なユーザの削除

My webMethods Server は、ディレクトリサービスから削除されたり、ディレクトリサービス内で移動したユーザは認識できません。そのようなユーザは無効なユーザと見なされます。

システム管理者は、Cleanup_InvalidUsers ユーティリティを使用して、My webMethods Server から無効なユーザを削除できます。Cleanup_InvalidUsers ユーティリティでは、無効なユーザを削除するほかに、My webMethods Server の役割から無効なユーザのメンバーシップを削除できます。

UserDirectory_UpdateUUID ユーティリティを実行する前に、Cleanup_InvalidUsers ユーティリティを実行する必要があります。UserDirectory_UpdateUUID ユーティリティの詳細については、124 ページの「ユーザへの Universally Unique Identifier (UUID) の設定」を参照してください。

重要:
ユーザベースが大規模な場合は、Cleanup_InvalidUsers ユーティリティの実行に長時間かかかる場合があります。システムメンテナンスに適した時間に Cleanup_InvalidUsers ユーティリティを実行する必要があります。

無効なユーザを削除するには

1. My webMethods Server データベースのバックアップを取ります。

2. [**フォルダ**] > [管理用フォルダ] > [Administration Dashboard] > [設定] > [インストール管理] > [Extras] > [UserDirectory] > [UserDirectory_UpdateUUIDPortlet.war] に移動します。

3. [**インストールコンポーネント**] をクリックします。

4. [**フォルダ**] > [システム] > [ポートレット] > [UserDirectory_UpdateUUIDPortlet] に移動して、[**Cleanup_InvalidUsers**] をクリックします。

5. [**Directory Service Name**] フィールドに、My webMethods Server で設定されたディレクトリサービスの名前を入力します。ワイルドカード文字 (*) を 1 つ入力して、my My webMethods Server で設定されたすべてのディレクトリサービスを指定できます。

6. [**Only log invalid user entries**] を選択し、[**Cleanup**] をクリックします。

   すべての無効なユーザが *Software AG_directory* \MWS\server\default\logs_full_.log ファイルにリストされます。ファイルを見直し、リストされたユーザがアクティブな My webMethods Server ユーザではないことを確認します。

7. [**Log and remove invalid user entries**] を選択し、[**Cleanup**] をクリックします。

8. My webMethods Server を再起動します。

外部データベースディレクトリサービスの設定

外部データベースディレクトリサービスを使用するように My webMethods Server を設定するには、以下の手順に従います。

注:
データベースディレクトリサービスを使用するには、最初に外部データソースとしてのデータベースに接続する必要があります。詳細については、133 ページの「外部データソースの管理」を参照してください。

外部データベースディレクトリサービスを設定するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [My webMethods] > [ディレクトリサービス] > [新規ディレクトリサービスの作成]
   - システム管理者として次のように選択します。[**管理**] > [ユーザ管理] > [ディレクトリサービス管理] > [新規ディレクトリサービスの作成]
2. [**ディレクトリタイプ**] フィールドで [**データベース**] をクリックします。
3. [**進む**] をクリックします。
4. [データベースディレクトリサービスの作成] ページで次の値を指定します。

次の表に、外部データベースディレクトリサービスを設定するのに必要なプロパティを示します。

セクション名	プロパティ	説明
[全般]	[名前]	必須。外部データベースディレクトリサービスを識別する名前。My webMethods Server は、ユーザインタフェース内で外部データベースディレクトリサービスを特定する必要がある場合にこの名前を使用します。
	[説明]	オプション。外部データベースディレクトリサービスに関する説明的なコメント。
[属性]	[ユーザ ID]	必須。ユーザ ID 値を指定するクエリーフィールドの名前。
	[ユーザ DN]	必須。グループの識別名の値を指定するクエリーフィールドの名前。
	[ユーザの名]	必須。ユーザの名を指定するクエリーフィールドの名前。
	[ユーザの姓]	必須。ユーザの姓を指定するクエリーフィールドの名前。
	[ユーザの氏名]	オプション。ユーザの氏名を指定するクエリーフィールドの名前。属性を指定しないと、氏名は、[**ユーザの名**] 属性と [**ユーザの姓**] 属性から導出されます。
	[ユーザ電子メール}	オプション。ユーザの電子メールアドレスを指定するクエリーフィールドの名前。

セクション名	プロパティ	説明
	[ユーザが無効]	オプション。ユーザが無効化されていると識別する外部ディレクトリサービス内の属性の名前。
	[「ユーザが無効」値の正規表現]	オプション。外部ディレクトリサービスの「ユーザが無効」属性を検証するために使用される正規表現。
	[グループ ID]	必須。グループ ID 値を指定するクエリーフィールドの名前。
	[グループ DN]	必須。グループの識別名の値を指定するクエリーフィールドの名前。
	[グループ名]	必須。グループ名を指定するクエリーフィールドの名前。
	[グループ電子メール]	オプション。グループの電子メールアドレスを指定するクエリーフィールドの名前。
[設定]	**[認証ハンドラ]**	必須。データベースの認証を処理するページ。デフォルトでは、My webMethods Server にクリアテキスト認証ハンドラが用意されています。
[データベース]	**[データソース]**	必須。データストアとして使用するデータベース。データベースをリストに表示するには、まず [データソース管理] ページを使用して外部データベースに接続する必要があります。
	[ID に基づくユーザ検索のクエリー]	必須。ユーザ ID に基づいてユーザレコードを返す SQL クエリー。 このクエリーは、この表の [**属性**] の下で説明している、ユーザの全属性を返します。
	[認証のクエリー]	必須。認証の継続的なユーザクレデンシャルを返す SQL クエリー。
	[さまざまなクエリー]	オプション。必要に応じて追加の SQL クエリーを定義できます。クエリーのタイプごとにサンプル言語が提供されています。
[キャッシュ]	**[キャッシュ有効]**	必須。データベースに対する負荷を軽減するよう、My webMethods Server でキャッシュ済みのデータを可能な限り使用するかどうかを指定します。次の項目から選択します。 ■ [**はい**]: キャッシュを有効にする場合。 ■ [**いいえ**]: キャッシュを無効にする場合。

セクション名	プロパティ	説明
	[キャッシュ容量]	必須。キャッシュに入れるデータベースクエリーの数。デフォルトは 1000 です。
	[キャッシュタイムアウト]	必須。キャッシュ容量を超えない限り、クエリーがキャッシュに残る時間。デフォルトは 1 です。

5. ページの下部で、[**終了**] をクリックします。

ヒント:
外部ディレクトリサービスが正しく設定されたことを確認するには、外部ディレクトリサービス内に定義されているユーザまたはグループを検索するクエリーを実行します。クエリーの実行方法の詳細については、「146 ページの「既存のユーザ、グループ、または役割の検索」」を参照してください。

外部で定義されたユーザに My webMethods からの操作を許可する

My webMethods Server は外部ディレクトリサービスの設定を通じて、必要な情報を入手して外部ディレクトリサービスに接続し、情報を取得することができます。外部ディレクトリサービスを設定すると、外部ディレクトリサービスで定義されたユーザは My webMethods にログインできるようになりますが、それ以外の操作を実行する権限は与えられません。My webMethods からの操作の実行をユーザに許可するには、以下の追加ステップを実行する必要があります。

- My webMethods へのアクセスを許可するユーザを識別するため、1 つ以上の LDAP クエリー役割またはデータベース役割を作成します。外部ディレクトリサービスの役割の作成方法の詳細については、129 ページの「My webMethods へのアクセスを許可する外部定義ユーザを定義する役割の作成」を参照してください。
- 作成した役割を、My webMethods Users 役割に追加します。詳細については、130 ページの「My webMethods Users 役割への役割の追加」を参照してください。

My webMethods へのアクセスを許可する外部定義ユーザを定義する役割の作成

設定した外部ディレクトリサービスごとに、My webMethods Server へのアクセスを許可するユーザまたはユーザグループを識別する 1 つ以上の役割を作成します。

- LDAP、ADSI、または ADAM ディレクトリサービスについては、LDAP クエリー役割を作成します。
- データベースディレクトリサービスについては、データベース役割を作成します。

My webMethods へのアクセスを許可するユーザを識別する役割を作成するには

1. 以下のいずれかの手順に従って、正しいページに移動します。

- My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [役割] > [役割の追加] を選択します。
- システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [役割の管理] > [役割の追加]

2. [**役割名**] フィールドで、「SystemX-My webMethods Users」など、新しい役割に割り当てる名前を指定します。

3. 作成する役割のタイプを指定するため、[**選択された項目**] ボックスに移動します。
   - [**LDAP クエリー役割プロバイダ**]: LDAP、ADSI、または ADAM ディレクトリサービス。
   - [**データベース役割プロバイダ**]: データベースディレクトリサービス。

4. [**役割の作成**] をクリックします。

5. 役割のメンバーシップを指定します。メンバーシップの定義方法の詳細については、次を参照してください。
   - LDAP、ADSI、または ADAM ディレクトリサービスについては、194 ページの「LDAP クエリー役割の追加」を参照してください。
   - データベースディレクトリサービスについては、198 ページの「データベース役割の追加」を参照してください。

6. [**保存**] をクリックします。

My webMethods Users 役割への役割の追加

作成した役割によって識別されるユーザに My webMethods へのアクセスを付与するには、この役割を My webMethods Server Users 役割に追加します。

My webMethods へのアクセスをユーザに付与するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [役割]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [役割の管理]

2. [**キーワード**] テキストボックスに、「`My webMethods Users`」と入力し、[**検索**] をクリックします。

3. 検索結果パネルで、My webMethods Users 役割について [**編集**] をクリックします。

4. [**メンバー**] タブをクリックします。

5. [**メンバーの編集**] をクリックします。

6. 追加するプリンシパルを検索し、My webMethods Users 役割のメンバーとする役割ごとに、役割名を [**選択済み**] パネルに移動します。

7. My webMethods Users 役割に割り当てるユーザを識別するすべての役割を選択したら、[**適用**] をクリックします。

8. [**保存**] をクリックして、My webMethods Users 役割のメンバーシップの変更を適用します。

ディレクトリサービスの設定の更新

外部ディレクトリサービスまたはデータベースディレクトリサービスを設定した後で、一部のプロパティに指定した値の更新が必要になることがあります。ディレクトリサービスに関連付けられたプロパティの値を更新するには、以下の手順に従います。

ディレクトリサービスの設定を更新するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [My webMethods] > [ディレクトリサービス] > [ディレクトリサービスの一覧表示]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [ディレクトリサービス管理] > [ディレクトリサービスの一覧表示]
2. [**ディレクトリサービスの一覧表示**] パネルで、以下のいずれかを行います。
   - 変更するディレクトリサービス設定の名前をクリックします。
   - My webMethods でディレクトリサービス設定の または をクリックし、[プロパティ] を選択します。
3. ディレクトリサービスのプロパティを変更します。
4. 変更を行ったら、[**適用**] をクリックします。

ディレクトリサービスの検索順序の更新

ユーザのアクションによっては、My webMethods Server が複数のディレクトリサービスに対してクエリーを行うことになります。My webMethods Server が使用可能なディレクトリサービスを検索する順序を制御するには、以下の手順に従います。

My webMethods Server による外部ディレクトリサービスの検索順序を更新するには

1. 次のページに移動します。

- My webMethods 管理者として、次のように選択します。[ナビゲーション] > [アプリケーション] > [管理] > [My webMethods] > [ディレクトリサービス] > [ディレクトリの検索順序の変更]
- システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [ディレクトリサービス管理] > [ディレクトリの検索順序の変更]

  My webMethods Server は、[**選択**] リストに指定された順序で外部ディレクトリサービスを検索します。

2. リストの順序を変更するには、必要に応じてディレクトリサービスを上下に移動します。

3. 検索順序を設定したら、[適用] をクリックします。

4. 変更した内容を有効にするために、My webMethods Server を再起動します。

注:
検索順序を設定しても、My webMethods Server によってユーザインタフェース内でディレクトリサービスがリスト表示される順序に影響はありません。

ユーザアカウントの無効化

外部ディレクトリサービス内の特定の属性の値に基づいて、My webMethods Server へのユーザのログインを阻止することができます。

注:
Integration Server や Optimize など、認証に共通ディレクトリサービスを使用する webMethods 製品は、この機能の影響を受けます。My webMethods Server で無効になっているユーザは、他の製品でも同様に無効になります。

外部ディレクトリサービスのユーザアカウントを無効にするには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [My webMethods] > [ディレクトリサービス] > [ディレクトリサービスの一覧表示]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [ディレクトリサービス管理] > [ディレクトリサービスの一覧表示]
2. [**ディレクトリサービスの一覧表示**] パネルで、以下のいずれかを行います。
   - ディレクトリサービスの名前をクリックします。
   - My webMethods でディレクトリサービスの または をクリックし、[プロパティ] を選択します。
3. ユーザアカウントの無効化に必要なプロパティを探します。

- LDAP、ADSI または ADAM ディレクトリサービスの場合は、[**ユーザ属性**] セクションを探します。
- データベースディレクトリサービスの場合は、[**属性**] セクションを探します。

4. [**ユーザが無効**] フィールドに、「ユーザが無効」状態を決定する外部ディレクトリサービス内の属性の名前を入力します。

   正確な値は、外部ディレクトリサービスおよび無効にするユーザのクラスによって決まります。

5. [**「ユーザが無効」値の正規表現**] フィールドに、[**ユーザが無効**] プロパティの値と照合する正規表現を入力します。値が一致した場合、ユーザは無効になります。

6. 変更を行ったら、[**適用**] をクリックします。

ディレクトリサービス設定の削除

My webMethods Server から外部ディレクトリサービスに定義されたユーザおよびグループへのアクセスが不要になった場合は、以下の手順に従って、外部ディレクトリサービスの設定情報を削除できます。

外部ディレクトリサービスの設定を削除するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [My webMethods] > [ディレクトリサービス] > [ディレクトリサービスの一覧表示]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [ディレクトリサービス管理] > [ディレクトリサービスの一覧表示]

2. My webMethods でディレクトリサービス設定の または をクリックし、[削除] を選択します。

外部データソースの管理

[データソース管理] ページで、データベースなどの外部データソースに接続して、My webMethods Server で使用可能にできます。たとえば、データベースディレクトリサービスを作成して使用する前には、データソースを設定する必要があります。

[データソース管理] ページで接続がサポートされているデータベースは、Microsoft SQL Server、Oracle、DB2 Universal、Sybase Adaptive Server、Informix です。また、ODBC やカスタムの接続も設定できます。

注:
DB2 Universal、Sybase Adaptive Server、または Informix データベースに接続するデータソースを設定する前に、それぞれのデータベースアプリケーションに対応するデータベースドラ

イバが必要です。My webMethods Server のディストリビューションには、DB2 Universal、Sybase Adaptive Server、または Informix データベースのデータベースドライバは含まれていません。

Microsoft SQL Server データソースの追加

Microsoft SQL Server データソースを My webMethods Server に接続するには、以下の手順に従います。

Microsoft SQL Server データベースの新規データソースを追加するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [My webMethods] > [データソース] > [データソースの追加]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [データソース管理] > [データソースの追加]
2. [**データソースの表示**] パネルで My webMethods Server によって使用される一意のデータソース名を [**データソース名**] に入力します。
3. データベースディレクトリサービスに使用するデータソースの識別に使用する表示名を [**表示名**] に入力します。
4. [**サーバタイプ**] リストから [**MS SQL Server**] を選択し、[**次へ**] をクリックします。
5. SQL Server ホスト名を入力します。
6. SQL Server で使用されるポート番号を入力します。デフォルトポートは 1433です。
7. データベース名を入力します。
8. 接続先のデータベースに対して最低でも読み取りアクセス権のある有効な SQL Server ユーザ名とパスワードを入力します。
9. [**サブミット**] をクリックします。

Oracle データソースの追加

Oracle データソースを My webMethods Server に接続するには、以下の手順に従います。

Oracle データベースの新規データソースを追加するには

1. 以下のいずれかの手順に従って、正しいページに移動します。

- My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [My webMethods] > [データソース] > [データソースの追加]
- システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [データソース管理] > [データソースの追加]

2. [**データソースの表示**] パネルで My webMethods Server によって使用される一意のデータソース名を [**データソース名**] に入力します。
3. データベースディレクトリサービスに使用するデータソースの識別に使用する表示名を [**表示名**] に入力します。
4. [**サーバタイプ**] リストから [**Oracle**] を選択し、[**次へ**] をクリックします。
5. Oracle ホスト名を入力します。
6. Oracle ホストが稼動しているポート番号を入力します。デフォルトポートは 1521です。
7. データベースのインスタンス名 (SID) を入力します。
8. 接続先のデータベースに対して最低でも読み取りアクセス権のある有効な Oracle データベースユーザ名とパスワードを入力します。
9. [**サブミット**] をクリックします。

DB2 Universal データソースの追加

DB2 Universal データソースを My webMethods Server に接続するには、以下の手順に従います。

DB2 Universal データベースの新規データソースを追加するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [My webMethods] > [データソース] > [データソースの追加]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [データソース管理] > [データソースの追加]
2. [**データソースの表示**] パネルで My webMethods Server によって使用される一意のデータソース名を [**データソース名**] に入力します。
3. データベースディレクトリサービスに使用するデータソースの識別に使用する表示名を [**表示名**] に入力します。
4. [**サーバタイプ**] リストから [**DB2 Universal**] を選択し、[**次へ**] をクリックします。

5. DB2 ホスト名を入力します。

6. DB2 が稼動しているポート番号を入力します。

7. データベースのインスタンス名を入力します。

8. 接続先のデータベースに対して最低でも読み取りアクセス権のある有効な DB2 ユーザ名とパスワードを入力します。

9. [**サブミット**] をクリックします。

Sybase Adaptive Server データソースの追加

Sybase Adaptive Server データソースを My webMethods Server に接続するには、以下の手順に従います。

Sybase Adaptive Server データベースの新規データソースを追加するには

1. 以下のいずれかの手順に従って、正しいページに移動します。

   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [My webMethods] > [データソース] > [データソースの追加]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [データソース管理] > [データソースの追加]

2. [**データソースの表示**] パネルで My webMethods Server によって使用される一意のデータソース名を [**データソース名**] に入力します。

3. データベースディレクトリサービスに使用するデータソースの識別に使用する表示名を [**表示名**] に入力します。

4. [**サーバタイプ**] リストから [**Sybase Adaptive Server**] を選択し、[**次へ**] をクリックします。

5. Sybase Server ホスト名を入力します。

6. Sybase Server が稼動しているポート番号を入力します。

7. データベースのインスタンス名を入力します。

8. 接続先のデータベースに対して最低でも読み取りアクセス権のある有効な Sybase Server ユーザ名とパスワードを入力します。

9. [**サブミット**] をクリックします。

Informix データソースの追加

Informix データソースを My webMethods Server に接続するには、以下の手順に従います。

Informix データベースの新規データソースを追加するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [My webMethods] > [データソース] > [データソースの追加]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [データソース管理] > [データソースの追加]
2. [**データソースの表示**] パネルで My webMethods Server によって使用される一意のデータソース名を [**データソース名**] に入力します。
3. データベースディレクトリサービスに使用するデータソースの識別に使用する表示名を [**表示名**] に入力します。
4. [**サーバタイプ**] リストから [**Informix**] を選択し、[**次へ**] をクリックします。
5. Informix ホスト名を入力します。
6. Informix ホストが稼動しているポート番号を入力します。
7. データベース名を入力します。
8. Informix サーバ名を入力します。
9. 接続先のデータベースに対して最低でも読み取りアクセス権のある有効な Informix ユーザ名とパスワードを入力します。
10. [**サブミット**] をクリックします。

汎用 ODBC データソースの追加

注:
My webMethods Server では、オペレーティングシステムレベル (Windows Server など) で手動で設定した ODBC 接続を使用できます。My webMethods Server は、標準の Java JDBC-ODBC ブリッジドライバを使用して、基礎となるオペレーティングシステム上の ODBC データソースに接続します。オペレーティングシステムレベルでの ODBC データソースの設定方法の詳細については、Microsoft ベンダードキュメントを参照してください。

汎用 ODBC データソースを My webMethods Server に接続するには、以下の手順に従います。

汎用 ODBC データベースの新規データソースを追加するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [My webMethods] > [データソース] > [データソースの追加]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [データソース管理] > [データソースの追加]
2. [**データソースの表示**] パネルで My webMethods Server によって使用される一意のデータソース名を [**データソース名**] に入力します。
3. データベースディレクトリサービスに使用するデータソースの識別に使用する表示名を [**表示名**] に入力します。
4. [**サーバタイプ**] リストから [**汎用 ODBC**] を選択し、[**次へ**] をクリックします。
5. オペレーティングシステムレベルで設定した ODBC データソースと一致する ODBC データソース名を入力します。
6. 接続先のデータベースに対して最低でも読み取りアクセス権のある有効なユーザ名とパスワードを入力します。
7. [**サブミット**] をクリックします。

カスタムデータソースの追加

注:
このオプションは高度なデータソース設定であり、有効な JDBC ドライバクラス名、接続 URL、ユーザ名およびパスワードを指定する必要があります。サーバから接続するデータベースアプリケーションの適切なデータベースドライバを探す場所についての具体的な指示は、ベンダーのドキュメントを調べてください。

カスタムデータソースを My webMethods Server に接続するには、ドライバの JAR ファイルを com.webmethods.caf.server バンドルのフラグメントとして宣言してから、My webMethods Server ユーザインタフェースを使用してデータソースを追加する必要があります。

カスタムデータベースの新規データソースを追加するには

1. My webMethods Server が実行中であれば、停止します。
2. ドライバの `*.jar` ファイルを *Software AG_directory* \MWS\lib ディレクトリにコピーします。

3. *Software AG_directory* \MWS\lib ディレクトリで、*driver-name*.bnd (*driver-name* はドライバの .jar ファイルの名前) という名前のテキストファイルを作成します。

   たとえば、Oracle シンドライバ ojdbc6.jar を使用する場合、ファイル名は ojdbc6.bnd です。

4. このファイルで、OSGi バンドル変換の指示を指定します。

   次の例で、斜体部分を適宜置換します。Bundle-Version には、JAR ファイルのバージョン番号を使用するのが一般的ですが、任意の一意の値であれば有効です。実際、この例に示す値を使用できます。

   ```
   # attach as fragment to the caf.server bundle
   Fragment-Host: com.webmethods.caf.server
   Bundle-SymbolicName: mws.jar.ojdbc6
   Bundle-Version: 0.9.0.v${tstamp}
   Include-Resource: ojdbc6.jar
   -exportcontents: *
   Bundle-ClassPath: ojdbc6.jar
   Import-Package: *;resolution:=optional
   ```

5. コマンドラインプロンプトで、サーバの bin ディレクトリに移動します。

   ```
   Software AG_directory\MWS\bin
   ```

6. 次のコマンドを入力します。

   ```
   mws.bat -s serverName update-osgi-profile
   ```

7. My webMethods Server インスタンスを再起動します。

8. 以下のいずれかの手順に従って、正しいページに移動します。

   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [My webMethods] > [データソース] > [データソースの追加]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [データソース管理] > [データソースの追加]

9. [**データソースの表示**] パネルで My webMethods Server によって使用される一意のデータソース名を [**データソース名**] に入力します。

10. データベースディレクトリサービスに使用するデータソースの識別に使用する表示名を [**表示名**] に入力します。

11. [**サーバタイプ**] リストから [**カスタム JDBC**] を選択し、[**次へ**] をクリックします。

12. データソース接続に使用するカスタムドライバの JDBC 接続クラスを入力します。

13. 有効な接続 URL を入力します。

14. 接続先のデータベースに対して最低でも読み取りアクセス権のある有効なユーザ名とパスワードを入力します。

15. [**サブミット**] をクリックします。

データソースの変更

注:
デフォルトのデータソースである My webMethods Server データベースは変更できません。

My webMethods Server へのデータソース接続を変更するには、以下の手順に従います。

既存のデータソースを変更するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [My webMethods] > [データソース] > [データソースの表示]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [データソース管理] > [データソースの表示]
2. 変更するデータソースの (My webMethods の をクリックし、[変更] を選択します。
3. [データソースのプロパティ] ウィザードの最初のページで、次を指定します。

次の表に、既存のデータソースを変更する際に設定するプロパティを示します。

フィールド	説明
[データソース名]	[**データソースの表示**] パネルで My webMethods Server によって使用されるデータソース名として、別の一意の名前を入力します。
[表示名]	データベースディレクトリサービスに使用するデータソースの識別に使用する表示名として、別の名前を入力します。
[サーバタイプ]	リストから別のデータベースタイプを選択します。

4. [**進む**] をクリックします。
5. [データソースプロパティ] ウィザードの 2 ページ目で、データソースの場所、名前、およびパスワードを識別する 1 つ以上のフィールドを変更します。
6. [**サブミット**] をクリックします。

データソースの削除

注:
デフォルトのデータソースである My webMethods Server データベースは削除できません。

My webMethods Server へのデータソース接続を削除するには、以下の手順に従います。

データソースを削除するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**管理**] > [My webMethods] > [データソース] > [データソースの表示]
   - システム管理者として次のように選択します。[**管理**] > [設定] > [データソース管理] > [データソースの表示]
2. My webMethods で変更するデータソースの または をクリックし、[削除] を選択します。

電子メール設定の管理

[電子メール管理] ページでは、電子メールの処理時にサーバで使用されるメールサーバ設定を設定します。

サーバ通知を送信するように電子メールサーバを設定するには

1. 以下のいずれかに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [My webMethods] > [電子メールサーバ]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [電子メール管理]
2. [電子メールのプロパティ] ページで次を指定します。

次の表に、My webMethods Server の電子メール通知設定に必要なプロパティを示します。

プロパティ	説明
[転送プロトコル]	使用する電子メールプロトコルを指定します。デフォルトの唯一有効な値は、「`smtp`」です。
[SMTP ホスト]	SMTP サーバを指定します。サーバのホスト名を指定します。たとえば、「`smtp.server.com`」と入力します。複数のホストを指定する場合は、1 行に 1 つのアドレスを入力します。

プロパティ	説明
[SMTP ポート]	ポート番号を指定します。SMTP サーバのポート番号を指定します。たとえば、「25」と入力します。
[SMTP ユーザ名]	オプション。My webMethods Server が認証のために提示するユーザ名を指定します。SMTP サーバで認証が必要な場合に、認証チャレンジを満たすために提示するユーザ名を指定します。
[SMTP パスワード]	オプション。[**SMTP ユーザ名**] に関連付けられているパスワードを指定します。SMTP サーバで認証が必要な場合に、適切なパスワードを指定します。
[SMTP TLS 有効]	オプション。暗号化 SMTP 接続 (TLS による) を使用するかどうかを示します。SMTP サーバで認証が必要な場合に、以下のいずれかに設定します。 ■ [**はい**]: TLS は有効です。 ■ [**いいえ**]: (デフォルト) TLS は有効ではありません。
[SMTP タイムアウト]	サーバからの応答を待機する最大時間を定義します。ミリ秒単位で指定します。デフォルト値は 60000 です。
[SMTP 接続タイムアウト]	特定の SMTP セッションの最大時間を定義します。ミリ秒単位で指定します。デフォルト値は 60000 です。
[SMTP Debug 有効]	オプション。電子メールアクティビティのデバッグを有効にするかどうかを示します。My webMethods Server は、デバッグ情報を My webMethods Server ログに書き込みます。以下のいずれかに設定します。 ■ [**はい**]: デバッグは有効です。 ■ [**いいえ**]: (デフォルト) デバッグは有効ではありません。
[送信者名]	デフォルトの「送信者」名を定義します。My webMethods Server が SMTP サーバを使用して送信する電子メールメッセージの [送信者] フィールドで使用されるデフォルトの名前を指定します。 **注:** このフィールド内のテキストは、RFC822 インターネットテキストメッセージ規格の要件の対象です。たとえば、「(重要)」のようなカッコ内のテキストはコメントとして扱われ、メッセージの作成時には削除されます。また、「[ステータス]」のような大カッコ内のテキストはオプションエレメントとして扱われ、同様に削除されます。
[送信者の電子メールアドレス]	デフォルトの「送信者」電子メールアドレスを定義します。My webMethods Server が SMTP サーバを使用して送信する電子メールメッセージの [送信者] フィールドで使用されるデフォルトの電子メールアドレスを指定します。

プロパティ	説明
[スキン]	オプション。My webMethods Server 電子メール通知をレンダリングするときに使用するスキンを指定します。My webMethods Server スキンを指定します。
[管理電子メールアドレス]	My webMethods Server の管理者の電子メールアドレスを定義します。これは、サーバに代わって送信される管理電子メールメッセージの「送信者」アドレスとして使用されます。

3. [**設定の保存**] をクリックします。

カレンダーの管理

Software AG Designer および My webMethods Server は、タスクの定義、割り当て、および動作を円滑に行うためのビジネスカレンダーとユーザカレンダーの使用をサポートします。ビジネスとユーザの両方のカレンダーは、My webMethods Server でセットアップと設定を行います。それぞれのタイプのカレンダーは、個別に設定されているため、ビジネスカレンダーのみ、ユーザカレンダーのみ、両方と分けて定義することができます。

ユーザカレンダーとビジネスカレンダーは、タスクの割り当てとスケジュールで、Task Engine により最も高い頻度で使用されます。ユーザカレンダーとビジネスカレンダーの作成と管理の詳細については、『*webMethods Task Engine User's Guide*』を参照してください。

9 ユーザ、グループ、および役割の検索

- 既存のユーザ、グループ、または役割の検索 146
- 拡張検索 147
- 保存済み検索の使用 148
- .csv ファイルへの検索結果のエクスポート 151

既存のユーザ、グループ、または役割の検索

以下のものを検索できます。

- 内部システムディレクトリサービス内に定義されたユーザおよびグループ
- 外部ディレクトリサービス内に定義されたユーザおよびグループ
- 役割

ユーザ、グループ、または役割を検索するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。**[ナビゲーション] > [アプリケーション] > [管理] > [ユーザ管理] > [*User_type*] > [検索]**
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [*User_type* の管理] > [検索]

   User_type は、[**ユーザ**]、[**グループ**]、または [**役割**] です。
2. ユーザおよびグループの場合は、[**名前**] リストから、ユーザまたはグループが定義されているディレクトリサービスを選択します。

   [**任意のディレクトリ**] を選択した場合は、サーバと接続しているすべてのディレクトリサービスが検索されます。
3. [検索] パネルの [キーワード] フィールドで、次のような検索条件を指定します。
   - ユーザを検索するには、名、姓、電子メールアドレス、またはユーザ ID を入力します。
   - グループを検索するには、グループ名、グループ ID、またはグループの電子メールアドレスを入力します。
   - 役割を検索するには、役割名または役割 ID を入力します。

   検索では大文字と小文字が区別されません。テキストボックスを空白のままにすると、選択されたディレクトリサービス内のすべてのエントリに関する情報が My webMethods Server から返されます。
4. [**実行**] をクリックします。

   My webMethods Server に、結果が表形式で表示されます。

検索結果のエクスポートの詳細については、151 ページの「.csv ファイルへの検索結果のエクスポート」を参照してください。

拡張検索

システム管理者は、ユーザとグループの情報および拡張属性に基づいて、ユーザまたはグループの拡張検索を実行できます。役割の拡張検索は実行できません。

ユーザまたはグループの拡張検索を実行するには

1. 以下のいずれかの手順に従って、正しいページに移動します。

   - My webMethods で次のように選択します。**[ナビゲーション] > [アプリケーション] > [管理] > [ユーザ管理] > [*User_type*] > [検索]**
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [*User_type* の管理] > [検索]

   User_type は、[**ユーザ**]、[**グループ**]、または [**役割**] です。

2. ユーザまたはグループの場合は、[**ディレクトリサービス**] リストから、ユーザまたはグループが定義されているディレクトリサービスを選択します。

   [**任意のディレクトリ**] を選択した場合は、サーバと接続しているすべてのディレクトリサービスが検索されます。

3. ユーザの場合は、[**コア属性**] パネルで、[ユーザ情報] の一部である下記のフィールドのいずれかに入力します。

次の表に、ユーザのコア属性を示します。

フィールド	説明
[**名前**]	保存済み検索で使用する名を入力します。
[**姓**]	保存済み検索で使用する姓を入力します。
[**ユーザ ID**]	保存済み検索で使用するユーザ ID を入力します。
[**電子メールアドレス**]	保存済み検索で使用する電子メールアドレスを入力します。

4. グループの場合は、[**コア属性**] パネルで、[グループ情報] の一部である下記のフィールドのいずれかに入力します。

次の表に、グループのコア属性を示します。

フィールド	説明
[**グループ名**]	保存済み検索で使用するグループ名を入力します。
[**グループ ID**]	保存済み検索で使用するグループ ID を入力します。

フィールド	説明
[電子メールアドレス]	保存済み検索で使用するグループの電子メールアドレスを入力します。

5. [**拡張属性**] パネルで、以下の方法を使用して、拡張検索を選択します。

   a. [**属性プロバイダ**] リストから属性プロバイダを選択します。

      属性プロバイダの詳細については、247 ページの「属性プロバイダ」を参照してください。

   b. [**属性値**] フィールドで、保存済み検索に使用する値を入力または変更します。

6. [**実行**] をクリックします。

   My webMethods Server に、結果が表形式で表示されます。

検索結果のエクスポートの詳細については、151 ページの「.csv ファイルへの検索結果のエクスポート」を参照してください。

保存済み検索の使用

定期的に実行する検索については、検索条件を保存して再利用できます。

保存済み検索の作成

ユーザ、グループ、または役割の保存済み検索を作成するには、以下の手順に従います。

保存済み検索を作成するには

1. 以下のいずれかの手順に従って、正しいページに移動します。

   - My webMethods で次のように選択します。**[ナビゲーション] > [アプリケーション] > [管理] > [ユーザ管理] > [*User_type*] > [検索]**
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [*User_type* の管理] > [検索]

   User_type は、[**ユーザ**]、[**グループ**]、または [**役割**] です。

2. ユーザおよびグループの場合は、[**ディレクトリサービス**] リストで、検索するユーザを含むディレクトリサービスを選択します。

   [**任意のディレクトリ**] を選択した場合は、サーバと接続しているすべてのディレクトリサービスが検索されます。

3. [検索] パネルの検索フィールドで、以下のいずれかを指定します。

- すべてのユーザ ID または一部のユーザ ID - ユーザの検索
- すべてのグループ ID または一部のグループ ID - グループの検索
- すべての役割名または一部の役割名 - 役割の検索

4. [保存] をクリックします。

5. [検索の保存] ダイアログボックスの [**検索名**] フィールドに、検索条件を特定できる名前を入力し、[**OK**] をクリックします。

保存済み検索の使用

保存済み検索を使用して、条件に一致するユーザ、グループ、または役割を検索できます。

保存済み検索を実行するには

1. 以下のいずれかの手順に従って、正しいページに移動します。

   - My webMethods で次のように選択します。**[ナビゲーション] > [アプリケーション] > [管理] > [ユーザ管理] > [*User_type*] > [保存済み]**
   - システム管理者として次のように選択します。**[Administration Dashboard] > [ユーザ管理] > [*User_type* の管理] > [保存済み]**

     User_type は、[**ユーザ**]、[**グループ**]、または [**役割**] です。

2. [**保存された検索**] リストで保存済み検索の名前を選択し、[**実行**] をクリックします。

保存済み検索の変更

既存の保存済み検索を変更できます。

保存済み検索を変更するには

1. 以下のいずれかの手順に従って、正しいページに移動します。

   - My webMethods で次のように選択します。**[ナビゲーション] > [アプリケーション] > [管理] > [ユーザ管理] > [*User_type*] > [保存済み]**
   - システム管理者として次のように選択します。**[Administration Dashboard] > [ユーザ管理] > [*User_type* の管理] > [保存済み]**

     User_type は、[**ユーザ**]、[**グループ**]、または [**役割**] です。

2. [**保存された検索**] リストで、変更する保存済み検索の名前を選択して [**詳細**] をクリックします。

3. 次のいずれかの手順に従います。

   - [**保存済み**] パネルの検索フィールドで、検索条件を変更します。
   - ユーザおよびグループの場合は、[**ディレクトリサービス**] リストで、検索を行う対象のディレクトリサービスを変更します。
   - ユーザの場合は、[コア属性] パネルで次のいずれかのフィールドを追加、変更、または削除します。
   - **名**
   - **姓**
   - **ユーザ ID**
   - **電子メールアドレス**

   157 ページの「ユーザ情報」で説明されているように、コアユーザ属性は [ユーザ情報] パネルのユーザ属性フィールドと一致します。

   - グループの場合は、[コア属性] パネルで次のいずれかのフィールドを追加、変更、または削除します。
   - **グループ名**
   - **グループ ID**
   - **電子メールアドレス**

   169 ページの「グループ情報」で説明されているように、コアユーザ属性は [グループ情報] パネルのグループ属性フィールドと一致します。

   - [**拡張属性**] パネルで、以下の方法を使用して、拡張検索を選択します。
     1. [**属性プロバイダ**] リストから属性プロバイダを選択します。

        属性プロバイダの詳細については、247 ページの「属性プロバイダ」を参照してください。

     2. [**属性値**] フィールドで、保存済み検索に使用する値を入力または変更します。

4. [**保存**] をクリックして、保存済み検索を更新します。

保存済み検索の削除

保存済み検索が不要になった場合には、削除できます。

保存済み検索を削除するには

1. 以下のいずれかの手順に従って、正しいページに移動します。

- My webMethods で次のように選択します。[**ナビゲーション**] > [**アプリケーション**] > [**管理**] > [**ユーザ管理**] > [***User_type***] > [**保存済み**]
- システム管理者として次のように選択します。[**Administration Dashboard**] > [**ユーザ管理**] > [***User_type* の管理**] > [**保存済み**]

  User_type は、[**ユーザ**]、[**グループ**]、または [**役割**] です。

2. [**保存された検索**] リストで、削除する保存済み検索の名前を選択して [**削除**] をクリックします。

.csv ファイルへの検索結果のエクスポート

検索結果パネルに [**表のエクスポート**] 機能が含まれている場合、検索結果をカンマ区切りテキストファイル (.csv ファイル) にエクスポートできます。

検索結果を .csv ファイルにエクスポートすると、.csv ファイルを Microsoft Excel、Microsoft Access、または .csv ファイル形式をサポートしているその他のアプリケーションにインポートできます。

検索結果をエクスポートするには

1. 検索結果パネルで [**表のエクスポート**] をクリックします。
2. [**文字エンコーディング**] リストから、使用する文字エンコーディングを選択します。
3. [**エクスポート**] をクリックします。
4. ブラウザのファイルダウンロード機能を使用して、.csv ファイルを保存する場所を参照します。

10 ユーザとグループの管理

- ユーザとグループの管理について 154
- ユーザの管理 155
- ユーザデータの管理 166
- グループの管理 167

ユーザとグループの管理について

My webMethods Administrator またはシステム管理者 (または、190 ページの「My webMethods Server における役割について」で説明する管理特権を持つユーザ) は、ユーザおよびグループを管理できます。

ユーザ

My webMethods にアクセスするには、システムがユーザの定義情報にアクセスできる必要があります。ユーザの定義は以下の方法で行うことができます。

- **内部システムディレクトリサービスにユーザを追加する**システムへのログインの際に入力するユーザ ID およびユーザパスワードなど、ユーザに関するすべての情報を指定します。内部システムディレクトリ内のユーザ管理方法の詳細については、155 ページの「ユーザの管理」を参照してください。
- **外部ディレクトリサービス内の定義済みユーザにアクセスする**1 つ以上の外部ディレクトリサービス内にユーザが定義されている場合は、外部ディレクトリサービスを使用するように My webMethods Server を設定できます。結果として、これらのユーザは My webMethods にアクセスして使用できるようになります。詳細については、116 ページの「外部ディレクトリサービスの管理」を参照してください。

注:
My webMethods Server では、内部システムディレクトリサービスに定義されたユーザと外部ディレクトリサービスに定義されたユーザの両方を組み合わせて使用できます。

グループ

ユーザの集合を論理的に編成してグループを作成すれば、個々のユーザを識別する代わりに、グループ名を使用してユーザのグループを識別できます。たとえば、ユーザのグループを役割に割り当てる場合、個々のユーザを指定する代わりに、これらのユーザを含むグループを役割に割り当てることができます。

グループを定義するには、以下を実行します。

- **内部システムディレクトリサービスにグループを追加する**グループに関する情報を指定し、メンバーシップを定義します。個々のユーザまたは他のグループをメンバーとして割り当てることができます。内部システムディレクトリサービス内に定義されたグループに割り当てるユーザおよびグループは、内部システムディレクトリサービスに定義されている必要があります。つまり、外部ディレクトリサービス内に定義されたユーザまたはグループを内部的に定義されたグループに割り当てることはできません。内部システムディレクトリ内のグループ管理方法の詳細については、167 ページの「グループの管理」を参照してください。
- **外部ディレクトリサービス内の定義済みグループにアクセスする**1 つ以上の外部ディレクトリサービス内に定義されたグループを使用する場合は、外部ディレクトリサービスを使用するように My webMethods Server を設定します。詳細については、116 ページの「外部ディレクトリサービスの管理」を参照してください。

注:
My webMethods Server では、内部システムディレクトリサービスに定義されたグループと外部ディレクトリサービスに定義されたグループの両方を組み合わせて使用できます。

ユーザの管理

ユーザの追加

ユーザが My webMethods にアクセスして使用できるようにするには、ユーザを内部システムディレクトリサービスに追加するか、ユーザが定義されている外部ディレクトリサービスを使用するように My webMethods Server を設定する必要があります。外部ディレクトリサービスの設定の詳細については、116ページの「外部ディレクトリサービスの管理」を参照してください。以下の手順は、内部システムディレクトリサービスにユーザを追加する方法を示しています。

注:
My webMethods Administrator が新規システムユーザを追加した場合、そのユーザは自動的に My webMethods Users 役割に追加されます。システム管理者が新規ユーザを追加した場合、そのユーザは自動的には My webMethods Users 役割に追加されません。

ユーザを追加するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [ユーザ] > [ユーザの追加]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [ユーザの管理] > [ユーザの追加]
2. 内部システムディレクトリサービスに追加するユーザに関して、以下のフィールドに入力します。

次の表に、システムディレクトリサービスに新しいユーザを追加するときに設定する必須フィールドとオプションフィールドを示します。

フィールド	説明
[ユーザ ID]	追加するユーザに割り当てるユーザ ID。My webMethods Server は、ユーザ ID を使用してユーザの識別名 (DN) を作成します。 ユーザ ID は 1～226 文字の ASCII 英数字で指定します。空白を含めることはできません。ユーザ ID では大文字と小文字が区別されません。 **注:** My webMethods Server では、ユーザが指定した大文字/小文字を使用して内部システムディレクトリサービスにユーザ ID が

フィールド	説明
	追加されます。一般的に My webMethods Server では、ユーザ ID の大文字と小文字は区別されませんが、大文字と小文字を区別するアクション (たとえば HTTP 認証) の場合は、ユーザが指定した大文字/小文字が My webMethods Server によって使用されます。
[パスワード}	新規ユーザのパスワード。
[パスワードの確認]	[**パスワード**] フィールドに指定したものと同じパスワード。
[名]	追加するユーザの名。My webMethods Server では、ユーザインタフェースの各ページでユーザの名前を表示する際に、ユーザの名および姓が使用されます。
[姓]	追加するユーザの姓。
[電子メールアドレス]	追加するユーザの電子メールアドレス (オプション)。My webMethods Server からユーザに電子メールメッセージで通知を送信する必要がある場合に、電子メールアドレスが使用されます。

3. [**作成**] をクリックします。

ユーザ情報の編集

内部システムディレクトリサービスに定義されたユーザの情報を編集できます。外部ディレクトリサービスに定義されたユーザについては、My webMethods Server 固有の情報のみ編集可能です。My webMethods Server が外部ディレクトリから取得する設定値を変更するには、外部ディレクトリサービスのディレクトリを更新する必要があります。My webMethods Server が維持するユーザ関連のフィールドのリストについては、157 ページの「ユーザ情報」を参照してください。

ユーザを編集するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [ユーザ]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [ユーザの管理]
2. 編集するユーザを検索します。詳細については、146 ページの「既存のユーザ、グループ、または役割の検索」を参照してください。

3. 検索結果で、編集するユーザの行にある任意のリンクをクリックするか、 をクリックします。

4. ユーザの情報を変更し、[**保存**] をクリックします。

   フィールドが更新可能であるか表示専用であるかを含めて、各パネルの全フィールドの説明については、「157 ページの「ユーザ情報」」を参照してください。

   重要:
   My webMethods Server は、さまざまなパネルに情報をグループ化して表示します。1 つのパネルで情報を変更したら、別のパネルを選択する前に、必ず [**保存**] をクリックしてください。

ユーザ情報

My webMethods Administrator および sysadmin ユーザは、どちらもユーザの情報を編集できます。My webMethods の [ユーザの編集] ページのユーザ属性は、sysadmin インタフェースのものとは異なります。

次の表は、My webMethods Server が維持するユーザ関連の情報を示しています。

注:
すべてのロケール属性のデフォルト設定は、NULL 値を表しています。

ユーザ情報

[ユーザ情報] セクションには、内部システムディレクトリサービスにユーザを追加する際に指定するすべての属性が含まれています。外部ディレクトリサービス内のユーザの [ユーザ情報] 属性を編集することはできません。このセクションはユーザプロファイルの一部です。一部のフィールド (パスワードおよび電子メールアドレス) はユーザによる更新が可能です。次の表に、[ユーザ情報] セクションに含まれる属性を示します。

フィールド	説明
[ユーザ ID]	新しいユーザを追加する場合に必須。My webMethods へのログインでユーザが入力する ID。ユーザがこのフィールドを更新することはできません。
[パスワード]	新しいユーザを追加する場合に必須。My webMethods へのログインでユーザが入力する必要のあるパスワード。ユーザインタフェースでは My webMethods Server がこのフィールドをマスクします。ユーザのパスワードを変更するには、[パスワード] および [パスワードの確認] フィールドを更新します。
[パスワードの確認]	新しいユーザを追加する場合に必須。[**パスワード**] フィールドと同じ。
[名前]	新しいユーザを追加する場合に必須。ユーザの名前には、次の特殊文字は使用できません: >、'、"、および &。

フィールド	説明
[姓]	新しいユーザを追加する場合に必須。ユーザの姓には、次の特殊文字は使用できません: <、>、'、"、および &。
[電子メールアドレス]	My webMethods Server がユーザに通知を送信するのに使用する電子メールアドレス。次の特殊文字は使用できません: <、>、'、"、および &。
[識別名 (DN)]	ユーザの識別名。指定されたユーザ情報をもとに My webMethods Server がこの値を生成します。ユーザがこのフィールドを更新することはできません。
[ログインを無効にする]	内部システムディレクトリ内のユーザにのみ適用。ユーザに My webMethods へのログインを許可するかどうかを示します。

[ユーザ属性]

[ユーザプロファイル] タブには、ユーザが属するディレクトリサービスとは無関係に、My webMethods Server が維持する属性が含まれます。次の表に、[ユーザプロファイル] タブに含まれる属性を示します。

フィールド	説明
[名前]	[ユーザ情報] セクションに定義されたユーザの名。[ユーザプロファイル] タブでこのフィールドを編集することはできません。名前を変更するには、[ユーザ情報] セクションで値を編集します。
[姓]	[ユーザ情報] セクションに定義されたユーザの姓。[ユーザプロファイル] タブでこのフィールドを編集することはできません。姓を変更するには、[ユーザ情報] セクションで値を編集します。
[ミドルネーム]	ユーザのミドルネーム。
[タイトル]	Mr.、Mrs.、Ms. などのユーザの敬称。使用したい敬称がリストにない場合は、[その他] を選択します。
[名前サフィックス]	ユーザ名の後に表示するサフィックス。Jr.、Sr.、PhD、III など (該当する場合)。使用したいサフィックスがリストにない場合は、[その他] を選択します。
[優先する言語/ロケール]	ユーザの言語およびロケール。
[住所 1][住所 2]	ユーザの住所の番地。
[カスタム住所表記]	ユーザの住所に郵便番号以上の情報が必要な場合に指定する追加情報 (特別な説明など)。
[市町村]	ユーザが所在する市町村。
[都道府県]	ユーザが所在する都道府県。

フィールド	説明
[郵便番号]	ユーザの郵便番号。たとえば、ユーザの所在地が米国の場合は ZIP コードです。
[国/地域 ID]	ユーザが所在する国。
[電話 1: 市外局番]	ユーザの市外局番。
[電話 1: 番号]	ユーザの電話番号。
[電話 1: 内線]	ユーザの内線番号 (該当する場合)。
[電話 1: 国番号]	ユーザの電話番号に関連付けられた国番号。
[デフォルトの日付形式]	日付を表示するときに My webMethods で使用する形式。日付形式の詳細については、SimpleDateFormat の Javadoc を参照してください。
[デフォルトの時間形式]	時間を表示するときに My webMethods で使用する形式。時間形式の詳細については、SimpleDateFormat の Javadoc を参照してください。
[デフォルトのタイムゾーン]	時間を表示するときに My webMethods で使用するタイムゾーン。
[デフォルトの数値形式]	数値を表示するときに My webMethods で使用する形式。数値形式の詳細については、DecimalDateFormat の Javadoc を参照してください。
[デフォルトの通貨形式]	通貨を表示するときに My webMethods で使用する形式。通貨形式の詳細については、DecimalDateFormat の Javadoc を参照してください。
[Android サブスクリプション]	ユーザプロファイル用にモバイル通知の受信をサブスクライブする Android デバイスを表示します。
[iOS サブスクリプション]	ユーザプロファイル用にモバイル通知の受信をサブスクライブする iOS デバイスを表示します。
[作成日時]	ユーザプロファイルが作成された日時を表示します。ユーザがこのフィールドを更新することはできません。
[最終更新日時]	[ユーザの編集] ページでユーザプロファイルが最後に変更された日時を表示します。ユーザがこのフィールドを更新することはできません。

[ユーザ設定] タブでは、My webMethods で情報を表示するときに使用される設定の一覧を示します。ユーザ設定は、それぞれのプロファイルから更新できます。詳細については、「*Working with My webMethods*」を参照してください。次の表に、[ユーザ設定] タブに含まれる属性を示します。

フィールド	説明
[開始ページ]	プロファイル用の開始ページを示します。変更するには、以下のオプションのいずれか 1 つを選択します。 ■ [アプリケーションページ] - アプリケーションページを開始ページとして開きます。 ■ [ワークスペース] - ワークスペースを開始ページとして開きます。 ■ [消去] - デフォルト設定に復元します。
[1 ページのアイテム数]	このユーザに対してアイテムを表に表示する際に、1 ページに含めるアイテムの数。
[ログイン時に開いてるタブを閉じる]	My webMethods Server へのログイン時に、前回のセッションで開いていたすべてのタブを閉じるかどうかを示します。
[ログイン時に最後にアクティブだったタブを開く]	My webMethods Server へのログイン時に、前回のセッションで最後にアクティブだったタブを開くかどうかを示します。
[ワークスペースの削除時に確認を表示]	ユーザがワークスペースを削除するときに、My webMethods Server が確認ダイアログボックスを表示するかどうかを示します。
[ウィンドウの削除時に確認を表示]	ユーザがワークスペースからウィンドウを削除するときに、My webMethods Server が確認ダイアログボックスを表示するかどうかを示します。
[すべてのタブを閉じるときに確認を表示]	ユーザが開いているすべてのタブを閉じるときに、My webMethods Server が確認ダイアログボックスを表示するかどうかを示します。
[別のワークスペースに移動するときに自動的に保存]	ユーザが別のワークスペースに移動するときに、My webMethods Server が現在のワークスペースを自動的に保存するかどうかを示します。

[役割] タブには、ユーザに割り当てられた役割、および各役割に関連する動的属性が表示されます。役割および動的属性の詳細については、192 ページの「役割の作成」および 200 ページの「役割に関連付ける動的属性の定義」 を参照してください。

注:
ユーザに動的役割を割り当てた場合、役割のリストが常に正確であるとは限りません。My webMethods Server は、ユーザが My webMethods にログインしたときに、属性を評価し、ユーザが属する役割を判別します。ユーザセッション中にユーザ属性が変更され、役割のメンバーとしての適格性が失われた場合でも、ユーザは次回のログイン時まで My webMethods Server によってその役割のメンバーと見なされます。

次の表に、[役割] タブに含まれる属性を示します。

フィールド	説明
[役割の優先順位]	ユーザに割り当てられた役割。優先順位に従って一覧表示されます。順序を変更するには、役割を上下に移動します。
[役割メンバー属性]	選択した役割に関連付けられた動的属性。動的属性は、役割に関してさらに詳細な情報を提供します。属性ごとに、下記の情報が表示されます。 ■ [属性] - 役割に属性を追加する際に指定する表示名。 ■ [データタイプ] - 属性のデータタイプ。 ■ [役割値] - 属性のデフォルト値。ユーザ固有の値が優先されない限り、すべてのユーザにこの値が割り当てられます。 ■ [ユーザ値] - この属性のユーザ固有の値。 たとえば、「お客様サービス」という役割に [場所] 属性を追加して、「お客様サービス」役割に割り当てられたユーザがどこに配置されているかを示すことができます。役割およびユーザ属性の変更の詳細については、200 ページの「役割に関連付ける動的属性の定義」および 201 ページの「動的属性のユーザ固有の値の指定」を参照してください。

[カレンダー] タブには、ビジネスまたは個人で使用するためのユーザカレンダーに関する情報が表示されます。ユーザカレンダーとビジネスカレンダーの作成と管理の詳細については、『*webMethods Task Engine User's Guide*』を参照してください。次の表に、[カレンダー] タブに含まれる属性を示します。

フィールド	説明
[ユーザビジネスカレンダー]	ビジネスカレンダーとして使用されるカレンダー。
[ユーザパーソナルカレンダー]	パーソナルカレンダーとして使用されるカレンダー。

[LDAP 属性] タブ - ユーザが外部ディレクトリサービスに定義されている場合は、このタブに、外部ディレクトリサービスからの固有の属性のセットが表示されます。LDAP 属性は sysadmin が設定する必要があります。詳細については、251 ページの「外部ディレクトリサービスの LDAP 属性の公開」を参照してください。

[データベース属性] タブ - ユーザが外部データベースディレクトリサービスに定義されている場合は、このタブに、外部データベースディレクトリサービスからの固有の属性のセットが表示されます。データベース属性は sysadmin が設定する必要があります。詳細については、252 ページの「外部ディレクトリサービスのデータベース属性の公開」を参照してください。

[グループ] タブには、ユーザが属するグループが表示されます。次の表に、[グループ] タブに含まれる属性を示します。

フィールド	
[場所]	My webMethods Server がユーザのグループメンバーシップについての情報を取得したディレクトリを示します。
[グループメンバーシップ]	選択可能なグループメンバーシップのオプション。矢印ボタンを使用してオプションを [選択された項目] フィールドに移動し、[適用] ボタンをクリックして新しいグループに追加します。
[選択された項目]	ユーザに割り当てられたグループ (ある場合)矢印ボタンを使用してオプションを [グループメンバーシップ] フィールドに移動し、[適用] ボタンをクリックしてユーザのグループメンバーシップから削除します。

ユーザデータのエクスポート

My webMethods ユーザインタフェースまたは API コールを使用して、プロファイルデータをエクスポートできます。

My webMethods Server は .json ファイル形式でユーザのパーソナルデータをエクスポートします。

My webMethods のユーザデータをエクスポートするには

1. [ナビゲーション] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [ユーザ] に移動します。
2. [ユーザ] リストで、プロファイルデータをエクスポートするユーザのユーザ名をクリックします。
3. [ユーザ属性] セクションの [ユーザプロファイル] タブで、[ユーザデータのエクスポート] をクリックします。

My webMethods Server REST API を使用するユーザデータのエクスポート

My webMethods Server REST API を使用しているユーザのデータをエクスポートするには、以下のエンドポイントに対して GET 要求を行います。

```
http://server_url:8585/api/users/personalData/userID
```

使用上の注意

この API には Administrator 特権が必要です。

例

ホスト `mws.company.com` のポート `8585` で稼働している My webMethods Server から、ID `myUser` のプロファイルデータをエクスポートする場合、次の URL を使用します。

```
http://mws.company.com:8585/api/users/personalData/myUser
```

グループへのユーザの割り当て

内部システムディレクトリサービス内で定義したユーザを、同様に内部システムディレクトリサービス内で定義したグループに割り当てることができます。グループの作成の詳細については、167 ページの「グループの追加」を参照してください。

注:
外部ディレクトリサービス内に定義したユーザを、内部システムディレクトリ内で定義したグループに割り当てることはできません。また、内部システムディレクトリサービス内で定義したユーザを外部的に定義されたグループに割り当てることもできません。ただし、内部ユーザと外部ユーザの両方を 1 つの役割に割り当てることは可能です。189 ページの「役割および My webMethods へのアクセスの管理」を参照してください。

システムディレクトリサービス内のユーザをシステムディレクトリサービス内のグループに割り当てるには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [ユーザ]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [ユーザの管理]
2. グループに割り当てるユーザを検索します。詳細については、146 ページの「既存のユーザ、グループ、または役割の検索」を参照してください。[**名前**] リストから必ず「**system**」を選択してください。
3. 検索結果で、編集するユーザの行にある任意のリンクをクリックするか、 をクリックします。
4. [**グループ**] をクリックします。
5. ユーザをメンバーとするグループを [**選択された項目**] ボックスに移動します。
6. すべてのグループを [**選択された項目**] ボックスに移動したら、[**保存**] (システム管理の場合は [**適用**]) をクリックします。

ユーザのログインの無効化

内部システムディレクトリサービスに定義されているユーザに、My webMethods Server へのログイン機能を使用させないようにできます。外部ディレクトリサービスに定義されているユーザのログインを無効にするには、132 ページの「ユーザアカウントの無効化」を参照してください。

注:
Integration Server や Optimize など、認証に共通ディレクトリサービスを使用する webMethods 製品は、この機能の影響を受けます。My webMethods Server で無効になっているユーザは、他の製品でも同様に無効になります。

ユーザのログインを無効にするには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [ユーザ]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [ユーザの管理]
2. 編集するユーザを検索します。詳細については、146 ページの「既存のユーザ、グループ、または役割の検索」を参照してください。
3. 検索結果で、無効にするユーザの行にある任意のリンクをクリックするか、 をクリックします。
4. [**ログインが無効**] オプションを選択し、[**保存**] をクリックします。

ユーザの削除

内部システムディレクトリサービス内の定義済みユーザを削除できます。

内部システムディレクトリサービスからユーザを削除するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [ユーザ]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [ユーザの管理]
2. 削除するユーザを検索します。詳細については、146 ページの「既存のユーザ、グループ、または役割の検索」を参照してください。[**名前**] リストから必ず「**system**」を選択してください。

3. 検索結果で、削除するユーザのユーザ ID の横のチェックボックスをオンにして、[**削除**] をクリックします。

My webMethods Server REST API を使用するユーザの削除

My webMethods ユーザが My webMethods Server REST API を使用している場合、このユーザを削除するには以下のエンドポイントに対して DELETE 要求を行います。

```
http://server_url:8585/api/users/userID
```

使用上の注意

- この API には Administrator 特権が必要です。
- この API で削除された LDAP ユーザおよびデータベースユーザは、外部ディレクトリから削除されるまで My webMethods にログインできます。

例

ホスト `mws.company.com` のポート `8585` で稼働している My webMethods Server から ID `myUser` のユーザプロファイルを削除する場合、次の URL を使用します。

```
http://mws.company.com:8585/api/users/myUser
```

ユーザホームフォルダの検索 (システム管理者のみ)

システム管理者はユーザのパーソナルフォルダを検索および参照できます。権限の変更によりユーザがアイテムを使用できなくなった場合や、ユーザがサーバのアクティブユーザでなくなったときにコンテンツを削除する場合は、この機能を使用します。

ユーザのホームフォルダを検索するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [ユーザホームフォルダの検索]
2. [**参照**] をクリックします。
3. ユーザを選択するには、[**選択された項目**] ボックスにユーザを移動し、[**選択**] をクリックします。
4. 選択したユーザのホームフォルダを開くには、[**適用**] をクリックします。

ユーザデータの管理

GDPR (General Data Protection Regulation: EU一般データ保護規則) などのデータ保護の法律や規則では、ユーザプロファイルを削除した後でも、特定のユーザデータの取り扱いが必要となる場合があります。

ユーザデータの完全性を確保するために、My webMethods Server は My webMethods 内でユーザプロファイルが最初に登録されたときのタイムスタンプと、現在のログインセッションの前に最後に変更が行われたときのタイムスタンプを保存しています。両方のタイムスタンプは、ユーザおよび管理者がプロファイルのユーザ情報の一部として確認できます。My webMethods ユーザは、プロファイルのタイムスタンプを表示して、すべてのプロファイルの変更を確認できます。

管理者がユーザインタフェースまたはAPIを使用してプロファイルを削除した場合、またはユーザが自分のプロファイルを削除した場合、My webMethods Server クラスタはユーザに関するすべての情報をデータベースおよびキャッシュから消去します。ただし、ユーザアカウントを削除しても、ログファイル内のユーザを識別するデータは削除されないため、管理者は My webMethods Server ログデータをクリーンアップするときの参照としてタイムスタンプを使用できます。

また、管理者は My webMethods Server を設定して、ユーザのプロファイルがユーザまたは管理者のいずれかによって変更された場合に、ユーザに電子メール通知を送信したり、通知メールに含まれる詳細のレベルを決めたりすることができます。

GDPR およびデータ保護設定の詳細については、以下を参照してください。

- 166 ページの「GDPR 設定の指定」
- 157 ページの「ユーザ情報」
- 162 ページの「ユーザデータのエクスポート」
- 164 ページの「ユーザの削除」
- 310 ページの「サーバのログの制御」

GDPR 設定の指定

My webMethods の [GDPR 設定] ページで、GDPR 準拠のユーザデータ保護オプションを有効または無効にします。

1. Administrator として、[アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [GDPR 設定] に移動します。

2. GDPR オプションを有効にするには、[GDPR の有効化] チェックボックスをオンにします。

3. オプションで、以下の手順を実行して電子メール通知を設定します。

   a. [通知タイプの選択] ドロップダウンリストから、電子メール通知を設定するユーザプロファイルイベントを選択します。

使用可能な通知タイプは、[更新通知] と [削除通知] です。

b. [通知の有効化] ドロップダウンリストから、イベントの電子メール通知を有効または無効にします。

c. [通知件名] フィールドでカスタマイズした件名を入力するか、デフォルトのテキストのまま残します。

d. [通知内容] フィールドで、必要に応じてメッセージ本文を変更します。

二重カッコで囲まれた文字列は、My webMethods Server が使用する変数であり、メッセージ内の My webMethods ユーザのユーザ名を含めるために使用します。`{{user}}` 変数はメッセージを受信するユーザ、`{{actor}}` 変数はプロファイルに変更を加えるユーザ、`{{changes}}` 変数はプロファイルの変更についての詳細を示します (設定されている場合)。次の例のように、テキスト内の変数を移動することができます。

デフォルトのメッセージ:

```
Dear {{user}},
Your personal data has been modified.
{{changes}}
The account was updated by user logged-in as {{actor}}.
```

変更後のメッセージ:

```
Hello {{user}},
You are receiving this message because
{{actor}} has modified your profile data.
Modifications:
{{changes}}
```

4. オプションで、変更したプロファイル属性のどの情報を含めるかを指定します。詳細オプションは、通知メッセージ内の `{{changes}}` 変数を置き換えます。オプションは次のとおりです:

- [詳細なし] - デフォルト。変更した属性についての詳細情報を含めません。
- [変更した属性名を含む] - 変更した属性の名前のみを通知メッセージに含めます。
- [変更した属性と値を含む] - 変更した属性の名前と値を通知メッセージに含めます。

5. [保存] をクリックします。

通知設定およびメッセージテキストを通知タイプごとに別々に設定します。

グループの管理

グループの追加

内部システムディレクトリサービス内にユーザのグループを定義できます。そのためには、最初にグループを作成します。グループを作成したら、グループにメンバーを追加できます。

グループを作成するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [グループ] > [グループの追加]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [グループの管理] > [グループの追加]
2. 内部システムディレクトリサービスに追加するグループに関して、以下のフィールドに入力します。

次の表に、システムディレクトリサービスに新しいグループを追加するときに設定する必須フィールドとオプションフィールドを示します。

フィールド	説明
[グループ ID]	グループの ID。My webMethods Server は、グループの DN (Distinguished Name : 識別名) でこの ID を使用します。 グループ ID は 1～255 文字の ASCII 英数字で指定します。空白を含めることはできません。グループ ID では大文字と小文字が区別されません。
[グループ名]	追加するグループに割り当てる名前。グループ名は 1～255 文字で指定します。
[電子メールアドレス]	追加するグループの電子メールアドレス (オプション)。

3. [**作成**] をクリックします。

グループ情報の編集

内部システムディレクトリサービスに定義されたグループの情報を編集できます。グループが外部ディレクトリサービスに定義されている場合は、My webMethods Server が外部ディレクトリから取得する設定値を変更するには、外部ディレクトリサービスのディレクトリを更新する必要があります。My webMethods Server が維持するグループ関連のフィールドのリストについては、169 ページの「グループ情報」を参照してください。

グループを編集するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [グループ]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [グループの管理]

2. 編集するグループを検索します。詳細については、146 ページの「既存のユーザ、グループ、または役割の検索」を参照してください。

3. 検索結果で、編集するグループの行にある任意のリンクをクリックするか、 をクリックします。

4. グループの情報を変更し、[**保存**] (システム管理の場合は [**適用**]) をクリックします。

   フィールドが更新可能であるか表示専用であるかを含めて、各パネルの全フィールドの説明については、「169 ページの「グループ情報」」を参照してください。

   重要:
   My webMethods Server は、さまざまなパネルに情報をグループ化して表示します。1 つのパネルで情報を変更したら、別のパネルを選択する前に、必ず [**保存**] (システム管理の場合は [**適用**]) をクリックしてください。

グループ情報

次の表は、My webMethods Server が維持するグループ関連の情報を示しています。この表のパネル欄は、[グループの編集] ページのどのパネルにフィールドが配置されているかを示します。

パネル	説明
[**グループ情報**]	グループの追加時にユーザが指定する属性。

フィールド	説明
[**グループ ID**]	グループに割り当てられたグループ ID。グループ ID はグループの追加時に定義されます。変更はできません。
[**グループ名**]	グループのグループ名。
[**電子メールアドレス**]	グループの電子メールアドレス。
[**識別名 (DN)**]	グループの識別名。ユーザがこのフィールドを更新することはできません。このフィールドは、グループに関して定義された情報を使用して、My webMethods Server によって形成されます。

パネル	説明
[**グループ**]	現在のグループが属するグループ。

フィールド	説明
[**場所**]	My webMethods Server で、左側パネルの [**グループメンバーシップ**] フィールドに表示されたアイテムがどこから入手されたかを示します。
[**グループメンバーシップ**]	左側のパネルには、選択可能なアイテムが表示されます。

パネル	説明	
	[選択された項目]	右側のパネル、[**選択された項目**] には、グループが割り当てられているグループが表示されます (ある場合)。
	[検索]	[**検索**] フィールドを使用して、左側パネルの [**グループメンバーシップ**] からアイテムを迅速に検索できます。左側のパネルに、1 ページに表示しきれないほど多くのアイテムが含まれている場合は、この方法が便利です。
[グループメンバー]	このグループに割り当てられたメンバー。メンバーはユーザまたは他のグループです。	
	フィールド	**説明**
	[場所]	My webMethods Server で、左側パネルの [**グループのメンバー**] フィールドに表示されたアイテムがどこから入手されたかを示します。
	[グループメンバー]	左側のパネルには、選択可能なアイテムが表示されます。
	[選択された項目]	右側のパネル、[**選択したアイテム**] には、グループのメンバーであるユーザおよびグループが表示されます。
	[検索]	[**検索**] フィールドを使用して、左側パネルの [**グループのメンバー**] からアイテムを迅速に検索できます。左側のパネルに、1 ページに表示しきれないほど多くのアイテムが含まれている場合は、この方法が便利です。
[LDAP 属性]	グループが外部ディレクトリサービスに定義されている場合は、このパネルに、外部ディレクトリサービスからの属性のセットが表示されます。このパネルのフィールドは、外部ディレクトリサービスに基づいています。LDAP 属性はシステム管理者が設定する必要があります。251 ページの「外部ディレクトリサービスの LDAP 属性の公開」を参照してください。	
[データベース属性]	ユーザが外部データベースディレクトリサービスに定義されている場合は、このパネルに、外部データベースディレクトリサービスからの属性のセットが表示されます。このパネルのフィールドは、外部データベースディレクトリサービスに基づいています。データベース属性はシステム管理者が設定する必要があります。252 ページの「外部ディレクトリサービスのデータベース属性の公開」を参照してください。	

グループのメンバーの管理

グループのメンバーは、ユーザまたは他のグループです。内部システムディレクトリサービスに定義されたグループに、同じディレクトリサービス内に定義されているメンバーを追加できます。

注:

外部ディレクトリサービス内に定義されたユーザまたはグループを操作するには、外部ディレクトリサービスから提供されているメカニズムを使用します。

内部システムサービスディレクトリに定義されたグループのメンバーを管理するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [グループ]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [グループの管理]
2. 編集するグループを検索します。詳細については、146 ページの「既存のユーザ、グループ、または役割の検索」を参照してください。
3. 検索結果で、編集するグループの行にある任意のリンクをクリックするか、 をクリックします。
4. [**グループのメンバー**] をクリックします。
5. グループのメンバーを管理するには、以下のいずれかを行います。
   - ユーザ (システムディレクトリサービス内) をグループに追加するには、[**選択された項目**] ボックスにユーザを移動します。
   - グループからユーザを削除するには、[**選択された項目**] ボックスからユーザを移動します。
   - グループ (システムディレクトリサービス内) をグループに追加するには、[**選択された項目**] ボックスにグループを移動します。
   - グループからグループを削除するには、[**選択された項目**] ボックスからグループを移動します。
6. グループに追加するすべてのメンバーが [**選択したアイテム**] リストに表示されたら、[**保存**] (システム管理の場合は [**適用**]) をクリックします。

グループを別のグループのメンバーにする

両方のグループが内部システムディレクトリサービスに定義されていれば、グループを別のグループのメンバーにすることができます。あるグループを別のグループのメンバーにすると、最初のグループのすべてのメンバーが、2 番目のグループのメンバーになります。

注:
外部ディレクトリサービス内に定義したグループを、内部システムディレクトリ内で定義したグループに割り当てることはできません。また、内部システムディレクトリサービス内で定義したグループを外部的に定義されたグループに割り当てることもできません。ただし、

内部グループと外部グループの両方を 1 つの役割に割り当てることは可能です。189 ページの「役割および My webMethods へのアクセスの管理」を参照してください。

グループを別のグループのメンバーにするには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [グループ]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [グループの管理]
2. 編集するグループを検索します。詳細については、116 ページの「外部ディレクトリサービスの管理」を参照してください。
3. 検索結果で、編集するグループの行にある任意のリンクをクリックするか、 をクリックします。
4. [**グループ**] をクリックします。
5. グループの別グループにおけるメンバーシップを管理するには、以下のいずれかを実行します。
   - 現在のグループを別のグループ (システムディレクトリサービス内) のメンバーにするには、親グループを [**選択された項目**] ボックスに移動します。
   - 現在のグループを別のグループのメンバーから除外するには、[**選択された項目**] ボックスから親グループを移動します。
6. 現在のグループをメンバーとして割り当てるすべてのグループが [**選択された項目**] リストに表示されたら、[**保存**] (システム管理の場合は [**適用**]) をクリックします。

グループの削除

内部システムディレクトリサービス内の定義済みグループを削除できます。

注:
グループを削除すると、グループの定義は削除されますが、削除したグループの個々のメンバー (ユーザまたは他のグループ) は削除されません。

内部システムディレクトリサービスからグループを削除するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [グループ]

- システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [グループの管理]

2. 削除するグループを検索します。詳細については、146 ページの「既存のユーザ、グループ、または役割の検索」を参照してください。[**名前**] リストから必ず「**system**」を選択してください。

3. 検索結果で、削除するグループの名前の横のチェックボックスをオンにして、[**削除**] をクリックします。

11 権限の管理

- My webMethods での権限の管理 176
- アクセス特権および機能特権の管理 179
- 個々のリソースの権限の管理 180
- セキュリティ領域の使用 182

My webMethods での権限の管理

My webMethods 管理者は、次のリソースの権限を管理します。

- webMethods アプリケーション
- タスク
- ワークスペース

システム管理者は、次のリソースの権限を管理します。

- webMethods アプリケーション
- タスク
- ワークスペース
- コンテンツオブジェクト
- ポートレットタイプ
- セキュリティ領域

どちらの管理者も、ユーザ、グループ、および役割の権限を必要な組み合わせで管理できます。

注:
[権限の管理] ページから組み込み My webMethods Server 役割の権限を変更しないでください。My webMethods Server に修正をインストールすると、変更が取り消される場合があります。

権限の追加

権限割り当ての基本的なワークフローでは、権限を割り当てるリソースを検索します。

権限を割り当てるには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [システム全体] > [権限の管理]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [権限の管理]
2. [検索] パネルの [**リソースタイプ**] リストから、管理するリソースタイプを選択します。

次の表に、権限を適用できるリソースを示します。

リソースタイプ	説明
[webMethods アプリケーション]	この My webMethods インスタンスにインストールされている webMethods アプリケーション。
タスク	webMethods Task Engine に関連付けられているタスク。
[ワークスペース]	この My webMethods インスタンスに作成済みのワークスペース。
[コンテンツオブジェクト]	(システム管理者のみ) この My webMethods Server インスタンスの任意のリソース (ファイル、フォルダ、ページなど)。
[ポートレットタイプ]	(システム管理者のみ) この My webMethods Server インスタンスにインストールされているポートレットタイプ。
[セキュリティ領域]	(システム管理者のみ) この My webMethods Server インスタンスに作成されているセキュリティ領域。詳細については、182 ページの「セキュリティ領域の使用」を参照してください。

3. 必要であれば、フィルタを適用して検索を絞り込みます。

   リソースタイプを選択すると、定義済みフィルタ条件のセットと共に [**フィルタ**] リストが表示されます。インストール済みのアプリケーションがすべて含まれる [webMethods アプリケーション] リソースタイプには、フィルタ条件がありません。

   a. [**フィルタ**] リストから、フィルタ条件を選択します。

      たとえば、[ワークスペース名] などがあります。

   b. 検索する値を入力します。

   c. フィルタ条件をさらに追加する必要がある場合は、管理者のタイプに応じて または をクリックします。

4. [**検索**] をクリックします。

   検索結果が [**検索結果**] リストに表示されます。

5. リソースを [選択済み] リストに移動します。

   検索を複数回実行すると、複数のリソースをリストに追加できます。また、検索を保存すれば、同じリソースを簡単に再検索できるようになります。詳細については、148 ページの「保存済み検索の使用」を参照してください。

6. [権限の編集] パネルで、[**ユーザ/グループ/役割の追加**] をクリックし、プリンシパルを検索して [**選択済み**] リストに移動します。

   検索を複数回実行すると、複数のプリンシパルをリストに追加できます。詳細については、146 ページの「既存のユーザ、グループ、または役割の検索」を参照してください。

7. すべてのプリンシパルを選択したら、[**追加**] をクリックします。

   選択したリソースタイプに付与できる権限または拒否される権限を含む [権限] パネルが、ツリー形式で表示されます。

8. 設定の [**許可**] または [**拒否**] オプションをクリックし、[**OK**] をクリックしてから [**適用**] をクリックします。

   どちらのオプションも選択されなかった場合、設定の権限は別のソースから決定されます。

権限の変更

サーバリソースに対して設定済みの権限を変更できます。ユーザ、グループ、または役割 (プリンシパル) の権限を変更したり、これらを完全に削除したりできます。

権限を変更するには

1. 以下のいずれかの手順に従って、正しいページに移動します。

   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [システム全体] > [権限の管理]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [権限の管理]

2. 変更するリソースに保存済み検索がある場合は、以下の手順に従います。

   a. [保存済み] タブをクリックし、[**保存された検索**] リストから保存済み検索を選択して [検索] をクリックします。

   b. 検索結果を [**選択済み**] リストに移動して、[**次へ**] をクリックします。

3. 保存済み検索がない場合にリソースを検索するには、以下の手順に従います。

   a. [検索] パネルの [**リソースタイプ**] リストから、管理するリソースを選択します。

   b. 必要であれば、1 つ以上のフィルタを適用して検索を絞り込みます。

   c. [**検索**] をクリックします。

   d. リソースを [選択済み] リストに移動します。

   e. [**進む**] をクリックします。

4. プリンシパルの権限を変更するには、以下の手順に従います。

a. [権限の編集] パネルの [**権限**] 列で、ユーザ、グループ、または役割のリンクをクリックします。

b. 必要に応じて各種設定の権限を変更して、[**OK**] をクリックします。

5. ユーザ、グループ、または役割の権限を削除するには、その行のチェックボックスをオンにして、[**削除**] をクリックします。

6. [**適用**] をクリックします。

アクセス特権および機能特権の管理

My webMethods での権限は、アクセス特権および機能特権の下にグループ化されています。

アクセス特権は、ナビゲーションパネルのアプリケーションと機能を表示し、My webMethods の関連付けられたページにアクセスするためのユーザ、グループ、または役割の権限を管理します。ユーザにアクセス特権を付与すると、そのユーザに一部の機能特権も付与され、ユーザはアプリケーションや機能を表示したり変更したりすることができるようになります。

機能特権は、アプリケーションまたは機能の中で、ワークスペースの作成や変更などを行うためのユーザ、グループ、または役割を決定します。機能特権を付与するには、アクセス特権を付与する必要があります。

アクセス特権および機能特権のリストは、インストールされている webMethods アプリケーションによって決まります。My webMethods 管理者およびシステム管理者は、ユーザ、グループ、または役割に、これらの特権を付与したり拒否したりできます。

ヒント:
複数のユーザ、グループおよび役割に同じ特権のセットを設定する必要がある場合は、すべてのユーザ、グループ、および役割を 1 つの役割に集約することを検討してください。

アクセス特権および機能特権を管理するには

1. 以下のいずれかの手順に従って、正しいページに移動します。

   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [システム全体] > [権限の管理]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [権限の管理]

2. [検索] パネルの [リソースタイプ] リストから、[webMethods アプリケーション] を選択します。

   デフォルトでは、既に My webMethods アプリケーションが [選択済み] リストに含まれています。

3. [**進む**] をクリックします。

4. [権限の管理] パネルで [追加] をクリックします。

5. ユーザ、グループ、または役割を検索して [選択済み] リストに移し、[追加] をクリックします。

   注:
   このアクションでは、ユーザ、グループ、役割ごとにのみ追加できます。3 つのプリンシパルを同時に追加することはできません。

   [権限] パネルが表示され、アクセス特権および機能特権に許可できる権限または拒否される権限を含むツリーリストが表示されます。

6. ツリーリストで、必要に応じてオプションを選択し、アクセス特権および機能特権を割り当てます。

   - 付与する特権ごとに [**許可**] をクリックします。
   - 拒否する特権ごとに [**拒否**] をクリックします。
   - 特権を明示的に付与も拒否もしない場合は、両方のオプションをクリアします。権限は別のソースから決定されます。

7. [**OK**] をクリックして、[**適用**] をクリックします。

個々のリソースの権限の管理

システム管理者は、My webMethods Server 内の個々のリソースを管理できます。サーバリソースまたはサーバリソースの階層に対する任意のプリンシパル (ユーザ、グループ、または役割) のアクセス権を制御できます。あるサーバリソースへのアクセスを拒否すると、プリンシパルは、サーバナビゲーション (同世代リソースや親リソースのコンテンツリスト内など) においてもそのリソースを表示できなくなります。

注:
このトピックでは、webMethods アプリケーションの一部ではない、ユーザが開発したページおよびその他のサーバリソースに対する権限の管理について説明します。

サーバリソースの所有者の表示および変更

サーバリソースの所有者を表示および変更するには

1. ページなどのサーバリソースで、 または をクリックして、[権限] を選択します。

2. [所有者] タブをクリックします。

   [所有者] パネルに、サーバリソースの所有者が表示されます。

3. リソース所有者を変更するには、以下の手順に従います。

a. [キーワード] フィールドにユーザ ID を入力し、[**検索**] をクリックし、[**選択済み**] ボックスにユーザを移動して、[**適用**] をクリックします。

b. [**適用**] をクリックします。

サーバリソース権限へのプリンシパルの追加

プリンシパルとは、ユーザ、グループ、または役割です。

サーバリソースに対する権限にプリンシパルを追加するには

1. ページなどのサーバリソースで、または をクリックして、[権限] を選択します。
2. [権限の編集] パネルで、[**ユーザ/グループ/役割の追加**] をクリックし、プリンシパルを検索して [**選択済み**] リストに移動します。

   検索を複数回実行すると、複数のプリンシパルをリストに追加できます。詳細については、146 ページの「既存のユーザ、グループ、または役割の検索」を参照してください。
3. すべてのプリンシパルを選択したら、[**追加**] をクリックします。

   選択したリソースに付与できる権限または拒否される権限を含む [権限] パネルが、ツリー形式で表示されます。
4. ツリーリストで、必要に応じてオプションを選択します。
   - 明示的に権限を付与するには [**許可**] をクリックします。
   - 明示的に権限を拒否するには [**拒否**] をクリックします。
   - 権限を明示的に付与も拒否もしない場合は、両方のオプションをクリアします。権限は別のソースから決定されます。
5. [**OK**] をクリックして、[**適用**] をクリックします。

サーバリソースに対する権限の変更

サーバリソースに対する権限を変更するには

1. ページなどのサーバリソースで、または をクリックして、[権限] を選択します。
2. プリンシパルの権限を変更するには、以下の手順に従います。

   a. [権限の編集] パネルの [**権限**] 列で、プリンシパルのリンクをクリックします。

b. 必要に応じて各種設定の権限を変更して、[**OK**] をクリックします。

- 明示的に権限を付与するには [**許可**] をクリックします。
- 明示的に権限を拒否するには [**拒否**] をクリックします。
- 権限を明示的に付与も拒否もしない場合は、両方のオプションをクリアします。権限は別のソースから決定されます。

3. [**適用**] をクリックします。

サーバリソース権限からのプリンシパルの削除

プリンシパルとは、ユーザ、グループ、または役割です。

サーバリソースに対する権限からプリンシパルを削除するには

1. ページなどのサーバリソースで、 または をクリックして、[権限] を選択します。
2. [権限の編集] パネルで、プリンシパルのチェックボックスをオンにして [**削除**] をクリックします。
3. [**適用**] をクリックします。

セキュリティ領域の使用

システム管理者が180 ページの「個々のリソースの権限の管理」の説明に従って権限を管理する場合は、一度に 1 つのリソースを管理します。この方法は、小規模なサーバでは十分ですが、ページとユーザの数が増えると管理が煩雑になる可能性があります。セキュリティ領域により、システム管理者はユーザ、グループ、または役割に基づいてリソースに対する権限を管理できるため、大規模なサーバの管理が簡単になります。セキュリティ領域をリソースに適用すると、個別に設定した権限は、明示的に設定を選択しない限り適用されません。

わかりやすくするために、セキュリティ領域をコンテナというフォルダに編成することができます。次の表に、My webMethods Server がデフォルトで提供するセキュリティ領域のコンテナを示します。

コンテナ	セキュリティ領域
フォーラム領域	フォーラムに対する権限を管理するセキュリティ領域
My webMethods セキュリティ領域	My webMethods リソースに対する権限を管理するセキュリティ領域
ポータルリソース	My webMethods Server リソースに対する権限を管理するセキュリティ領域

[セキュリティ領域管理] ページでは、コンテナの作成、名前の変更、および削除を行えます。

My webMethods Server リソースに対する権限を管理するデフォルトのセキュリティ領域がいくつかあり、そのすべてが [ポータルリソース] コンテナにあります。システム管理者には、サーバリソースの読み取り、変更、または削除を行う権限があります。次の表に示すように、他のユーザにも権限があります。

セキュリティ領域	管理される権限
管理コマンド	管理リソース。サーバ管理者にのみ、リソースの読み取り、変更、または削除を許可します。
ディレクトリ管理コマンド	ユーザ、グループおよび役割を管理するリソース。ユーザにリソースの表示または実行を許可します。Anonymous ユーザはアクセスを拒否されます。
ディレクトリサービスコマンド	ディレクトリサービスを管理するリソース。ユーザにページの表示または実行を許可します。Anonymous ユーザはアクセスを拒否されます。
ポータル開発者コマンド	ページとコンテンツの開発および保守用のリソース。[Developers] グループのメンバーは、リソースの読み取り、変更、または削除の権限を付与されます。ユーザにリソースの表示または実行を許可します。Anonymous ユーザはアクセスを拒否されます。
パブリックコマンド	ログインなど、サーバとのやりとり用のリソース。Anonymous ユーザを含むすべてのユーザにリソースの読み取りまたは実行を許可しますが、変更または削除は許可しません。
制限付きコマンド	ログイン後のサーバとのやりとり用のリソース。ログインしたユーザにリソースの読み取りまたは実行を許可しますが、変更または削除は許可しません。Anonymous ユーザはアクセスを拒否されます。
ユーザプロファイル管理コマンド	サーバのルックアンドフィールを制御するリソース。ユーザにリソースの表示または実行を許可します。Anonymous ユーザはアクセスを拒否されます。

[セキュリティ領域管理] ページで、コンテンツに使用できるセキュリティ領域の作成および管理を行います。

コンテナの作成

デフォルトのコンテナと同じレベルでコンテナを作成することも、コンテナ内にコンテナを作成することもできます。

新規コンテナを作成するには

1. システム管理者として次のように選択します。[**管理**] > [設定] > [セキュリティ領域管理]

2. [セキュリティ領域管理] ページで、以下のいずれかを実行します。
   - 現在のレベルで新しいコンテナを作成するには、[新規コンテナの作成] タブをクリックします。
   - ネストされたコンテナを作成するには、既存のコンテナの名前をクリックしてから [新規コンテナの作成] タブをクリックします。
3. [新規コンテナの作成] タブの [名前] フィールドに、コンテナの表示名を入力します。
4. [説明] フィールドに、説明を入力します (オプション)。
5. [コンテナの作成] をクリックします。

コンテナの削除

重要:
コンテナを削除すると、その中にあるセキュリティ領域やその他のコンテナも削除されます。

コンテナを削除するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [セキュリティ領域管理]
2. 削除するコンテナの をクリックし、[コンテナの削除] を選択します。

コンテナの名前の変更

コンテナの名前を変更するには、以下の手順に従います。

コンテナの名前を変更するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [セキュリティ領域管理]
2. 名前を変更するコンテナの をクリックし、[コンテナの変更] を選択します。
3. [**名前**] フィールドに、新しい名前を入力します。
4. [**説明**] フィールドに、新しい説明を入力します (オプション)。
5. [**更新**] をクリックします。

セキュリティ領域の作成

新規セキュリティ領域を作成するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [セキュリティ領域管理]

2. セキュリティ領域を作成するコンテナの名前をクリックします。

   注:
   コンテナ内にセキュリティ領域を作成すると、それを別のコンテナには移動できません。

3. [**新規セキュリティ領域の作成**] をクリックします。

4. [**名前**] フィールドに、セキュリティ領域の表示名を入力します。

5. [**説明**] フィールドに、説明を入力します (オプション)。

6. 拡張タイプを追加するには、拡張タイプを [**選択された項目**] ボックスに移動して、[**選択**] をクリックします (オプション)。

7. 外部のポリシープロバイダを追加するには、ポリシープロバイダを [**選択された項目**] ボックスに移動します (オプション)。

8. [**セキュリティ領域の作成**] をクリックします。

9. 新しいセキュリティ領域の をクリックし、[権限の設定] を選択します。

10. [権限の編集] パネルで、[**ユーザ/グループ/役割の追加**] をクリックし、プリンシパルを検索して [**選択済み**] リストに移動します。

    検索を複数回実行すると、複数のプリンシパルをリストに追加できます。詳細については、146 ページの「既存のユーザ、グループ、または役割の検索」を参照してください。

11. すべてのプリンシパルを選択したら、[**追加**] をクリックします。

    そのセキュリティ領域に付与できる権限または拒否される権限を含む [権限] パネルが、ツリー形式で表示されます。

12. ツリーリストで、必要に応じてオプションを選択します。

    - 明示的に権限を付与するには [**許可**] をクリックします。
    - 明示的に権限を拒否するには [**拒否**] をクリックします。
    - 権限を明示的に付与も拒否もしない場合は、両方のオプションをクリアします。権限は別のソースから決定されます。

13. [**OK**] をクリックして、[**適用**] をクリックします。

セキュリティ領域の削除

セキュリティ領域を削除するには、以下の手順に従います。

セキュリティ領域を削除するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [セキュリティ領域管理]
2. セキュリティ領域が存在するコンテナの名前をクリックします。
3. 削除するセキュリティ領域の をクリックし、[セキュリティ領域の削除] を選択します。

セキュリティ領域の名前の変更

セキュリティ領域の名前を変更するには、以下の手順に従います。

セキュリティ領域の名前を変更するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [セキュリティ領域管理]
2. セキュリティ領域が存在するコンテナの名前をクリックします。
3. 名前を変更するセキュリティ領域の をクリックし、[セキュリティ領域の変更] を選択します。
4. [**名前**] フィールドに、新しい名前を入力します。
5. [**説明**] フィールドに、新しい説明を入力します (オプション)。
6. [**更新**] をクリックします。

セキュリティ領域へのリソースの追加

セキュリティ領域にリソースを追加するには、以下の手順に従います。

セキュリティ領域にリソースを追加するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [セキュリティ領域管理]

2. セキュリティ領域が存在するコンテナの名前をクリックします。

3. 管理するセキュリティ領域の をクリックし、[オブジェクトの管理] をクリックします。

4. [セキュリティ領域の管理] パネルで、[ポータルリソースの追加] をクリックします。

5. 左のパネルで、セキュリティ領域に追加するサーバリソースを参照します。

6. セキュリティ領域でサーバリソースを管理できるようにするには、リソースを [選択された項目] ボックスに移動して、[アイテムの追加] をクリックします。

セキュリティ領域からのリソースの削除

セキュリティ領域からリソースを削除するには、以下の手順に従います。

セキュリティ領域からリソースを削除するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [セキュリティ領域管理]

2. セキュリティ領域が存在するコンテナの名前をクリックします。

3. 管理するセキュリティ領域の をクリックし、[オブジェクトの管理] をクリックします。

4. 削除するリソースの をクリックして、[削除] を選択します。

12 役割および My webMethods へのアクセスの管理

- My webMethods Server における役割について 190
- ユーザへの My webMethods へのアクセス権および My webMethods Users 役割の付与 . 191
- 役割の作成 192
- 役割の情報の編集 199
- 役割の削除 200
- 役割に関連付ける動的属性の定義 200

My webMethods Server における役割について

役割とは、ユーザ、グループまたはその他の役割のコレクションです。My webMethods Server のインストールと共にデフォルトの役割のセットがインストールされます。この初期のセットに、ユーザ、グループ、またはその他の役割を追加できます。

次の表に、My webMethods Server で使用可能なデフォルトの役割と、それらの役割でアクセス可能なリソースを示します。

デフォルトの役割	説明
Admin Role	My webMethods Server のすべてのリソースへのアクセスを提供します。デフォルトでは、SysAdmin および Designer ユーザがこの役割のメンバーになります。
My webMethods Administrators	この役割のデフォルトメンバーである My webMethods Administrator が必要とする、ユーザ管理およびその他の機能へのアクセスを提供します。
My webMethods Users	My webMethods アプリケーションのすべてのユーザに対して、My webMethods ユーザインタフェースへのアクセスを提供します。My webMethods Server の管理者はこの役割のデフォルトメンバーですが、その他すべてのユーザは追加する必要があります。詳細については、191 ページの「ユーザへの My webMethods へのアクセス権および My webMethods Users 役割の付与」を参照してください。

異なるディレクトリサービスからのメンバーを 1 つの役割に割り当てることができます。つまり、内部ディレクトリサービスに定義されたユーザ、グループ、役割と同様に、外部ディレクトリサービスに定義されたユーザとグループをメンバーに含めることができます。個々のメンバーを特定して割り当てるグループの場合と同様に、役割のメンバーシップは静的なものにすることができます。ただし、役割のメンバーシップを動的なものにすることもできます。役割のメンバーシップを再帰的なものとして、役割同士を相互のメンバーとすることができます。

次の表は、役割のメンバーシップのさまざまな定義方法を示すと共に、メンバーシップが静的か動的か、およびそれぞれのタイプの役割のメンバーシップの定義方法に関する詳細情報の参照先を示しています。

メンバーシップの定義方法	静的または動的	参照先
役割のメンバーとするユーザ、グループ、およびその他の役割を指定する	静的	193 ページの「静的役割の追加」
外部ディレクトリサービスをクエリーする LDAP クエリーを指定して、役割に割り当てるユーザまたはグループを判別する	動的	194 ページの「LDAP クエリー役割の追加」

メンバーシップの定義方法	静的または動的	参照先
My webMethods Server が実行するルールの条件を指定して、役割に割り当てるユーザおよびグループを判別する	動的	195 ページの「ルールベース役割の追加」
データベースディレクトリサービスをクエリーするデータベースクエリーを指定して、役割に割り当てるユーザまたはグループを判別する	動的	198 ページの「データベース役割の追加」

役割に動的属性を関連付けて、役割に関する詳細情報を指定できます。たとえば、「お客様サービス」という役割がある場合、管理者は [場所] 属性を追加して、「お客様サービス」役割に割り当てられたユーザがどこに配置されているかを示すことができます。役割に属性を追加する際に、属性の値を割り当てることができます。これが、その属性のデフォルト値になります。前述の例で、メインサービスセンターがオハイオ州にあるとします。この場合は、「お客様サービス」役割に [場所] 属性を追加するときに、「Ohio」という値を割り当てます。また、動的属性にユーザ固有の値を割り当てることもできます。もう一度前述の例を使用して、「お客様サービス」役割のメンバーであるユーザが、オハイオ州ではなくコロラド州に配属されているとします。「お客様サービス」役割の [場所] 属性について、「Colorado」というユーザ固有の値を割り当てることができます。

動的属性の詳細については、以下を参照してください。

- 201 ページの「動的属性のユーザ固有の値の指定」
- 202 ページの「役割に割り当てた動的属性の削除」

ユーザへの My webMethods へのアクセス権および My webMethods Users 役割の付与

My webMethods Server には、My webMethods Users 役割が事前定義されています。My webMethods Users 役割は、My webMethods へのアクセスを制御します。つまり、ユーザインタフェースにログインするためには、ユーザが My webMethods Users 役割のメンバーである必要があります。My webMethods Users 役割は静的な役割です。つまり、My webMethods Users 役割のメンバーシップは、リスト (ユーザ、グループ、およびその他の役割を含む) で指定されています。

内部システムディレクトリサービスに定義されたユーザ

内部システムディレクトリサービスにユーザを追加すると、My webMethods Server は新規ユーザを My webMethods Users 役割に自動的に割り当てます。これ以上のユーザアクションは必要ありません。

外部定義ユーザ

ユーザおよびグループに関する情報を外部ディレクトリサービスから入手できるように My webMethods Server を設定できます。外部ディレクトリサービスの設定後、追加のアクションとして、My webMethods からのアクセスが必要な外部定義ユーザを識別する必要があります。これらのユーザを My webMethods Users 役割で指定しておく必要があります。My webMethods Server での外部ディレクトリサービスの使用方法の詳細については、116 ページの「外部 LDAP、ADSI、または ADAM ディレクトリサービスの設定」および 129 ページの「外部で定義されたユーザに My webMethods からの操作を許可する」を参照してください。

役割の作成

My webMethods Server 管理者は、さまざまなタイプのユーザ役割を作成できます。My webMethods Server では、タイプに関係なく、すべての役割が同じ方法で使用されますが、役割のメンバーシップはタイプごとに異なります。

次の表では、作成可能な役割のタイプと、役割のタイプごとに役割のメンバーシップを識別する方法について説明します。

役割	メンバーシップの識別方法
静的役割	作成する役割のメンバーとするユーザ、グループ、または役割のコレクションを指定します。役割を手動で編集してメンバーシップを変更しない限り、役割のメンバーシップが変わることはありません。詳細については、193 ページの「静的役割の追加」を参照してください。 **注:** これは、グループの場合と同じですが、役割のメンバーシップは複数のディレクトリサービスをまたがって指定できる点が異なります。
LDAP クエリー役割	LDAP クエリーを指定します。クエリーと一致したユーザ、グループ、および役割が役割のメンバーとなります。役割のメンバーシップは、ランタイムのクエリーの結果に基づき、動的です。詳細については、194 ページの「LDAP クエリー役割の追加」を参照してください。
ルールベース役割	メンバーシップを判別するために My webMethods Server が実行するルールを指定します。ルールと一致したユーザ、グループ、および役割が役割のメンバーとなります。役割のメンバーシップは、ランタイムのルールの実行結果に基づき、動的です。詳細については、195 ページの「ルールベース役割の追加」を参照してください。
データベース役割	データベースディレクトリサービスに対するクエリーを指定します。クエリーと一致したユーザ、グループ、および役割が役割のメンバーとなります。役割のメンバーシップは、ランタイムのクエリーの結果に基づき、動的です。詳細については、198 ページの「データベース役割の追加」を参照してください。

静的役割の追加

静的役割は、ユーザ、グループ、および他の役割の単純なコレクションです。

静的役割を作成するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[ナビゲーション] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [役割] > [役割の追加]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [役割の管理] > [役割の追加]
2. [**役割名**] フィールドに、新しい役割に割り当てる名前を入力します。

   有効な役割名には、文字、数字、アンダースコア、または空白文字のみが含まれます。
3. [**静的役割プロバイダ**] を選択するには、その役割プロバイダを [**選択された項目**] ボックスに移動します。
4. [**役割の作成**] をクリックします。

静的役割のメンバーの編集

静的役割のメンバーを編集するには、以下の手順に従います。

静的役割のメンバーを編集するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[ナビゲーション] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [役割]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [役割の管理]
2. メンバーを追加する静的役割を検索します。詳細については、146 ページの「既存のユーザ、グループ、または役割の検索」を参照してください。
3. 検索結果で、役割名をクリックするか、 をクリックします。
4. [メンバー] パネルで、[**メンバーの編集**] をクリックします。
5. メンバーを追加するには、以下の手順に従います。

a. [**検索対象**] で、[**ユーザ**]、[**グループ**]、または [**役割**] オプションを選択します。

b. [**キーワード**] フィールドで、検索するユーザ、グループ、または役割を表すキーワードを入力し、[**検索**] をクリックします。

c. 1 つ以上のユーザ、グループ、または役割を [**選択済み**] ボックスに移動します。

6. 静的役割からメンバーを削除するには、[**選択済み**] ボックスからメンバーを移動します。

7. [**適用**] をクリックします。

LDAP クエリー役割の追加

LDAP クエリー役割は、外部ディレクトリサービスへの LDAP クエリーに基づきます。クエリーの要件を満たすすべてのユーザまたはグループは、役割のメンバーです。

LDAP クエリー役割を作成するには

1. 以下のいずれかの手順に従って、正しいページに移動します。

   - My webMethods で次のように選択します。[ナビゲーション] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [役割] > [役割の追加]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [役割の管理] > [役割の追加]

2. [**役割名**] フィールドに、新しい役割に割り当てる名前を入力します。

   有効な役割名には、文字、数字、アンダースコア、または空白文字のみが含まれます。

3. [**LDAP クエリー役割プロバイダ**] を選択するには、その役割プロバイダを [**選択された項目**] ボックスに移動します。

4. [**役割の作成**] をクリックします。

5. [**LDAP クエリー**] フィールドに、有効な LDAP クエリーを入力します。

6. [**LDAP クエリー**] フィールドのクエリーに簡略化された LDAP クエリーの構文が含まれている場合は、[**単純クエリー**] オプションを選択します。

   複雑な LDAP クエリーを作成する場合以外、クエリーの構文は扱いにくい可能性があります。[**単純クエリー**] オプションを使用すると、構文が自動的に入力されます。たとえば、ユーザ ID が abrown である管理者によって管理されるすべてのユーザを検索する場合、単純なクエリーの構文は `manager=abrown` です。

7. [**LDAP ディレクトリサービス**] で、[**参照**] をクリックします。

8. LDAP ディレクトリサービスを [**選択された項目**] ボックスに移動し、[**選択**] をクリックします。

9. [**プリンシパルタイプ**] リストで、クエリーで [**ユーザ**] と [**グループ**] のどちらを検索するかを選択します。

10. LDAP クエリーを更新するには、[**保存**] (システム管理の場合は [**適用**]) をクリックします。

ルールベース役割の追加

ルールベース役割は、サーバルールに基づきます。ルールと一致するすべてのユーザ、グループ、または役割は、役割のメンバーです。

ルールベース役割を作成するには

1. 以下のいずれかの手順に従って、正しいページに移動します。

   - My webMethods で次のように選択します。[ナビゲーション] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [役割] > [役割の追加]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [役割の管理] > [役割の追加]

2. [**役割名**] フィールドに、新しい役割に割り当てる名前を入力します。

   有効な役割名には、文字、数字、アンダースコア、または空白文字のみが含まれます。

3. [**データベース役割プロバイダ**] を選択するには、その役割プロバイダを [**選択された項目**] ボックスに移動します。

4. [**役割の作成**] をクリックします。

5. [**一致条件**] 見出しの下で、[**下のすべての条件に一致**] または [**下の任意の条件に一致**] をルールベース役割の条件として選択します。

6. 以下のガイドラインを使用して、ルールベース役割の適切な一致条件を入力します。

   [**ユーザ DN の値**]: 現在のユーザのディレクトリ DN (Distinguished Name : 識別名) の任意の部分と一致する正規表現。フィールドに、一致させる DN の部分を入力します。

   たとえば、`ou=Engineering.*ou=US` は、次の DN を持つユーザと一致します。

   ```
   uid=joe,ou=Development,ou=Engineering,ou=Midwest,ou=US,o=webMethods
   ```

   [**ドメイン名表現**]: My webMethods Server に登録されている現在のユーザのディレクトリサービス名の任意の部分と一致する正規表現。フィールドに、一致させるディレクトリサービス名を入力します。

たとえば、us (引用符なし) は US Corporate ディレクトリサービスのユーザと一致します。これは、パートナなど、異なるユーザディレクトリに存在する可能性のあるユーザのルックアンドフィールを管理するための効果的な方法です。

[**グループ DN 表現と役割 DN 表現**]: 現在のユーザがメンバーとなっている任意のグループまたは役割の任意の部分と一致する正規表現。フィールドに、一致させる DN の部分を入力します。

たとえば、ou=Engineering は、次の DN を持つグループに属するユーザと一致します。

```
cn=portal,ou=Engineering,ou=Midwest,ou=US,o=webMethods.
```

[**ユーザ属性**]: ユーザ属性と、ユーザのレコードから取得した値の 1 つ以上の組み合わせです。ユーザ属性が複数ある場合、[**一致条件**] に設定した値により、属性をどのように一致させるかが決まります。

- [下のすべての条件に一致] - 各正規表現は、現在のユーザの対応する属性値の一部と一致する必要があります。
- [下の任意の条件に一致] - リスト内の任意の正規表現は、現在のユーザの対応する属性値の一部と一致する必要があります。

たとえば、すべての条件と一致するようにルールが設定されており、ユーザ属性の組み合わせが次のように設定されていると仮定します。

名前	値
office	Bellevue
telephonenumber	(425) 564-0000

また、現在のユーザの属性値は次のとおりです。

名前	値 (現在のユーザ)
office	Bellevue
telephonenumber	(206) 123-4567

office 属性の値は一致しますが、telephonenumber 属性の値は一致しないため、現在のユーザはルールと一致しません。しかし、任意の条件と一致するようにルールが設定されていれば、上記の例でルールは現在のユーザと一致します。

属性と値の組み合わせを作成するには、[**追加**] をクリックします。プロンプトで、属性名を入力し、[**OK**] をクリックします。プロンプトで、一致させる値を入力し、[**OK**] をクリックします。

[**要求ヘッダー**]: HTTP ヘッダー属性と値の 1 つ以上の組み合わせ。ブラウザエージェント文字列や、ユーザに受け入れられる MIME タイプの種類など、HTTP ヘッダー内に表示されるものであればいずれも一致させることができます。ルールは正規表現でも、単純なテキスト文字列でもかまいません。属性と値の組み合わせが複数ある場合は、[**一致条件**] に設定した値により、属性をどのように一致させるかが決まります。

- [下のすべての条件に一致] - 各正規表現は、要求ヘッダーの対応する属性値の一部と一致する必要があります。
- [**下の任意の条件に一致**] - リスト内の任意の正規表現は、要求ヘッダーの対応する属性値の一部と一致する必要があります。

たとえば、すべての条件と一致するようにルールが設定されており、要求ヘッダーの組み合わせが次のように設定されていると仮定します。

名前	値
`Accept-Charset`	utf-8
`Accept-Language`	ja

また、現在のユーザの要求ヘッダー値は次のとおりです。

名前	値 (現在のユーザ)
`Accept-Charset`	ISO-8859-1,utf-8;q=0.7
`Accept-Language`	en-us,en;q=0.5

Accept-Charset ヘッダーの値は一致しますが、Accept-Language ヘッダーの値は一致しないため、現在のユーザはルールと一致しません。しかし、任意の条件と一致するようにルールが設定されていれば、ルールは現在のユーザと一致します。

属性と値の組み合わせを作成するには、[**追加**] をクリックします。プロンプトで、属性名を入力し、[**OK**] をクリックします。プロンプトで、一致させる値を入力し、[**OK**] をクリックします。

[**親リソース**]: 現在のリソースまたは現在のリソースの親と一致するリソース。リソースを選択するには、[**参照**] をクリックしてリソースセレクタを開き、ルールを一致させる対象のリソースを選択します。エイリアスが参照しているリソースを一致させる場合は、[**エイリアスを使用**] をクリックして My webMethods Server 上の既存のエイリアスを選択します。

[**リソースタイプ**]: 現在のリソースタイプに一致するリソースタイプ。リソースタイプを選択するには、[**参照**] をクリックしてリソースセレクタを開き、ルールを一致させる対象のリソースタイプを [拡張タイプ] フォルダから選択します。エイリアスが参照しているリソースタイプを一致させる場合は、[**エイリアスを使用**] をクリックして My webMethods Server 上の既存のエイリアスを選択します。

[**リソースプロパティ**]: リソースのプロパティと値の 1 つ以上の組み合わせ。リソースに関連付けられているプロパティの内部名がわかっている場合は、その名前と一致させることができます。プロパティと値の組み合わせが複数ある場合は、[**一致条件**] に設定した値により、プロパティをどのように一致させるかが決まります。

- [下のすべての条件に一致] - 各正規表現は、要求ヘッダーの対応する属性値の一部と一致する必要があります。
- [**下の任意の条件に一致**] - リスト内の任意の正規表現は、要求ヘッダーの対応する属性値の一部と一致する必要があります。

たとえば、PDF ファイルを一致させる場合、プロパティと属性の組み合わせは `mimeType=pdf` です。

プロパティと値の組み合わせを作成するには、[**追加**] をクリックします。プロンプトで、属性名を入力し、[**OK**] をクリックします。プロンプトで、一致させる値を入力し、[**OK**] をクリックします。

7. [**適用**] をクリックします。

データベース役割の追加

データベース役割は、データベースディレクトリサービスへのクエリーに基づきます。ルールと一致するすべてのユーザ、グループ、または役割は、役割のメンバーです。

注:
データベース役割を作成するには、最初に外部データソースとしてのデータベースに接続する必要があります。133 ページの「外部データソースの管理」を参照してください。

データベース役割を作成するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[ナビゲーション] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [役割] > [役割の追加]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [役割の管理] > [役割の追加]
2. [**役割名**] フィールドに、新しい役割に割り当てる名前を入力します。

   有効な役割名には、文字、数字、アンダースコア、または空白文字のみが含まれます。
3. [**データベース役割プロバイダ**] を選択するには、その役割プロバイダを [**選択された項目**] ボックスに移動します。
4. [**役割の作成**] をクリックします。
5. [**データソース**] リストから、データストアとして使用するデータベースを選択します。
6. 役割にユーザを含めることができる場合は、[**ユーザのクエリー**] フィールドに、その役割のメンバーにするデータベース内の特定のユーザのレコードを返す SQL クエリーを入力します。

   クエリーに対するパラメータは次のとおりです。
   - {uid}: プリンシパルの一意の ID
   - {dn}: プリンシパルの識別名

有効なクエリーの例は、次のようになります。

```
select * from user-roles where roleID='Admin' and userid='{uid}'
```

7. 役割にグループを含めることができる場合は、[**グループのクエリー**] フィールドに、その役割のメンバーにするデータベース内の特定のグループのレコードを返す SQL クエリーを入力します。

8. [**保存**] をクリックします。

役割の情報の編集

役割に関する情報を更新するには、以下の手順に従います。

役割を編集するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[ナビゲーション] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [役割]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [役割の管理]
2. 編集する役割を検索します。詳細については、146 ページの「既存のユーザ、グループ、または役割の検索」を参照してください。
3. 検索結果で、役割名をクリックするか、 をクリックします。
4. 各パネルで、必要に応じて役割情報を編集し、[保存] (システム管理の場合は [適用]) をクリックします。

   重要:
   My webMethods Server は、さまざまなパネルに情報をグループ化して表示します。1 つのパネルで情報を変更したら、別のパネルを選択する前に、必ず [**保存**] (システム管理の場合は [**適用**]) をクリックしてください。

   次の表は、役割に関連して表示されるパネルを示しています。

パネル	参照先
[役割情報]	193 ページの「静的役割の追加」 194 ページの「LDAP クエリー役割の追加」 198 ページの「データベース役割の追加」
動的属性	200 ページの「役割に関連付ける動的属性の定義」

役割の削除

役割が不要になったら、役割を削除できます。

注:
役割を削除しても、役割のメンバー (ユーザ、グループ、およびその他の役割) が削除されることはありません。

役割を削除するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[ナビゲーション] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [役割]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [役割の管理]
2. 削除する役割を検索します。詳細については、146 ページの「既存のユーザ、グループ、または役割の検索」を参照してください。
3. 検索結果で、削除する役割の横のチェックボックスをオンにして、[**削除**] をクリックします。

役割に関連付ける動的属性の定義

役割に動的属性を関連付けることができます。webMethodsの一部のアプリケーションでは動的属性が使用されます。

役割の属性を定義する際に、属性の値を割り当てることができます。役割に割り当てられた値は、デフォルト値と見なされます。属性にユーザ固有の値が定義されていない限り、すべてのユーザでこのデフォルト値が使用されます。

役割の動的属性を定義するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[ナビゲーション] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [役割]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [役割の管理]
2. 役割を検索します。詳細については、146 ページの「既存のユーザ、グループ、または役割の検索」を参照してください。

3. 検索結果で、役割名をクリックするか、 をクリックします。

4. [**動的属性**] をクリックします。

5. [**属性の追加**] をクリックします。

6. [**属性の追加**] ページで、次を指定します。

次の表に、動的役割属性を追加する際に設定するプロパティを示します。

フィールド	説明
[属性名]	動的属性の内部名。この名前に使用できる文字には制限がありません。
[表示名]	この属性の情報をユーザインタフェースに表示する際に My webMethods Server で使用する名前。
[データタイプ]	属性のデータタイプ。リストからデータタイプを選択します。
[値]	属性に割り当てる値。これが属性のデフォルト値です。属性にユーザ固有の値が定義されていない限り、この役割に割り当てられたすべてのユーザでこの値が使用されます。詳細については、201 ページの「動的属性のユーザ固有の値の指定」を参照してください。

7. [**保存**] をクリックします。

動的属性のユーザ固有の値の指定

ユーザ固有の値が割り当てられていない限り、動的属性に割り当てられたデフォルト値がすべてのユーザのデフォルト値となります。動的属性にユーザ固有の値を割り当てるには、以下の手順に従います。

動的属性にユーザ固有の値を割り当てるには

1. 以下のいずれかの手順に従って、正しいページに移動します。

   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [ユーザ]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [ユーザの管理]

2. ユーザ固有の値を動的属性に割り当てるユーザを検索します。詳細については、146 ページの「既存のユーザ、グループ、または役割の検索」を参照してください。

3. 検索結果で、ユーザの行にある任意のリンクをクリックするか、 をクリックします。

4. [**役割**] をクリックします。

5. [**役割メンバー属性**] パネルで、ユーザ固有の値を割り当てる動的属性を見つけて、[**ユーザ値**] 列に値を入力します。

   属性は役割別にグループ化され、役割に割り当てられた属性のリストの先頭に役割名が表示されます。

6. [**保存**] (システム管理の場合は [**適用**]) をクリックします。

役割に割り当てられた動的属性の順序の変更

役割の動的属性が表示される順序を変更できます。順序の変更は、ユーザの [**役割メンバー属性**] パネルでの属性の表示にのみ影響します。My webMethods Server または webMethods アプリケーションでの属性の使用方法に影響はありません。

役割に割り当てた動的属性の順序を変更するには

1. 以下のいずれかの手順に従って、正しいページに移動します。

   - My webMethods で次のように選択します。[ナビゲーション] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [役割]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [役割の管理]

2. 役割を検索します。詳細については、146 ページの「既存のユーザ、グループ、または役割の検索」を参照してください。

3. 検索結果で、役割名をクリックするか、 をクリックします。

4. [**動的属性**] をクリックし、[**属性の順序を変更**] をクリックします。

5. 動的属性の順序を変更するには、必要に応じて動的属性を上下に移動します。

6. 動的属性の順序を変更したら、[**保存**] (システム管理の場合は [**適用**]) をクリックします。

役割に割り当てた動的属性の削除

役割に割り当てた 1 つ以上の動的属性が不要になった場合は、動的属性を削除できます。

役割に割り当てた 1 つ以上の動的属性を削除するには

1. 以下のいずれかの手順に従って、正しいページに移動します。

- My webMethods で次のように選択します。[ナビゲーション] > [アプリケーション] > [管理] > [システム全体] > [ユーザ管理] > [役割]
- システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [役割の管理]

2. 役割を検索します。詳細については、146 ページの「既存のユーザ、グループ、または役割の検索」を参照してください。

3. 検索結果で、役割名をクリックするか、 をクリックします。

4. [**動的属性**] をクリックします。

5. 削除する動的属性の横のチェックボックスをオンにして、[**選択した属性の削除**] をクリックします。

6. [**保存**] をクリックします。

13 My webMethods Server のクラスタリング

- My webMethods Server クラスタの仕組み 206
- My webMethods Server クラスタの計画 210
- My webMethods Server クラスタ構築の考慮事項 212
- クラスタ設定の変更 213
- クラスタのリソースロック設定の変更 223
- クラスタの消去スレッドの数の設定 224
- クラスタの監視および制御 224
- クラスタからのコンポーネントの削除 228
- cluster.xml ファイルの使用 228
- イメージからのクラスタノードの作成 231
- クラスタノードでのアプリケーションのパーティション化 233

My webMethods Server クラスタの仕組み

My webMethods Server クラスタとは、My webMethods Server データベースを共有する複数のサーバインスタンスを同時に実行する、アクティブ/アクティブ環境です。My webMethods Server クラスタでは、複数のサーバに負荷を分散することで、高度なスケーラビリティが実現されます。このモデルは、スタンバイサーバを使用するアクティブ/パッシブ環境とは異なります。My webMethods Server クラスタは、共有リソースを使用することによって高可用性を実現します。これにより、ノードの 1 つが機能しなくなっても、クラスタとしては機能し続けることができます。

クラスタを作成するには、クラスタ内の各マシンに My webMethods Server のインスタンスをインストールする必要があります。クラスタのすべてのノードは、システムコンテンツサービスが保存する共有設定情報およびデータを含む、同じ My webMethods Server データベースを共有します。

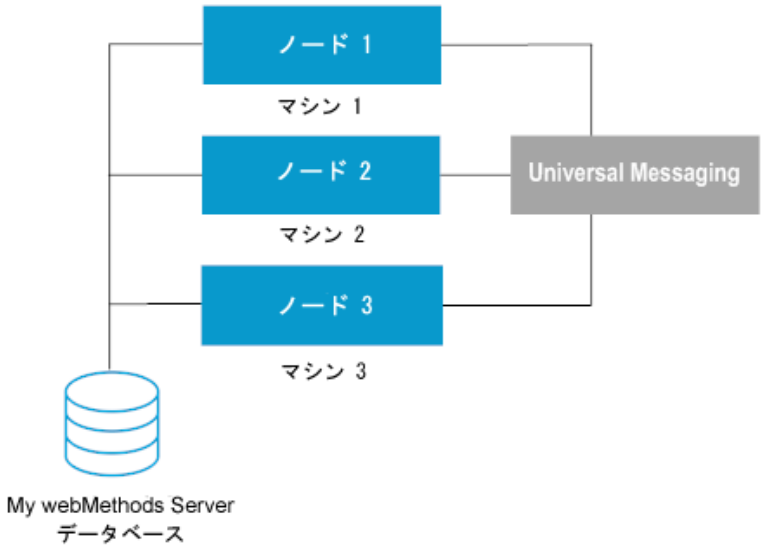
システムコンテンツサービスは、My webMethods Server で実行している webMethods ユーザおよびアプリケーションによってパブリッシュされたファイルやオブジェクトを保存および取得します。システムコンテンツサービスは、My webMethods Server とともにインストールされます。追加の設定を行う必要はありません。パブリッシュされたコンテンツは、クラスタ内のすべてのノードで使用可能になります。

クラスタ内の各ノードには、同じ webMethods アプリケーション (Task Engine など)、My webMethods ユーザインタフェースおよび言語パックのセットをインストールする必要があります。あるノードにアプリケーションがインストールされていない場合、そのノードでは、アプリケーションのコンポーネントが機能しません。

クラスタ内のノードは、以下のチャネルを使用して、実行時に情報を交換します。

- My webMethods Server データベース - クラスタのブックキープはこの方法で維持されます。
- HTTP - クラスタノードが別のノードにコマンドの実行を委任する場合は、ターゲットサーバの HTTP ポートを使用します。
- Universal Messaging サーバ - My webMethods Server ノードは、Universal Messaging サーバを使用してノード間のキャッシュを同期する JMS イベントのやり取りを行います。

クラスタアーキテクチャ



フロントエンド URL

実稼動環境では、webMethods クラスタと通信するすべてのユーザおよびすべての My webMethods Server アプリケーションが、クラスタ用に設定されたフロントエンド URL を使用する必要があります (213 ページの「フロントエンド URL の設定」を参照)。さらに、実稼動クラスタで、207 ページの「My webMethods Server クラスタでの高可用性」に示すロードバランサまたは外部 Web サーバが使用されている必要があります。

実際の運用では、ユーザはロードバランサを通じてフロントエンド URL にログインします。それを受けて、ロードバランサは各ノードにコールを分配します。1 つのクラスタに設定できるフロントエンド URL は 1 つだけです。

注:
ロードバランサはスティッキーセッションを使用するように設定する必要があります。一度確立されたユーザセッションは、セッションが終了するまで同じサーバマシンにルーティングされます。

My webMethods Server ページを生成する HTTP 要求が発行された場合、My webMethods Server では、要求のルーティング方法 (ロードバランサ経由での直接のルーティング、または Web サーバ経由での間接的ルーティング) に関係なく、その要求の実際のホスト名とポート番号が使用されます。このような場合には、フロントエンド URL は必要ありません。

電子メール通知内から My webMethods Server URL を生成する場合や、Task Engine 内からタスクへのリンクを作成する場合などのいくつかのユースケースにおいては、それらの処理時に HTTP 要求を使用することができないため、フロントエンド URL 設定が必要になります。

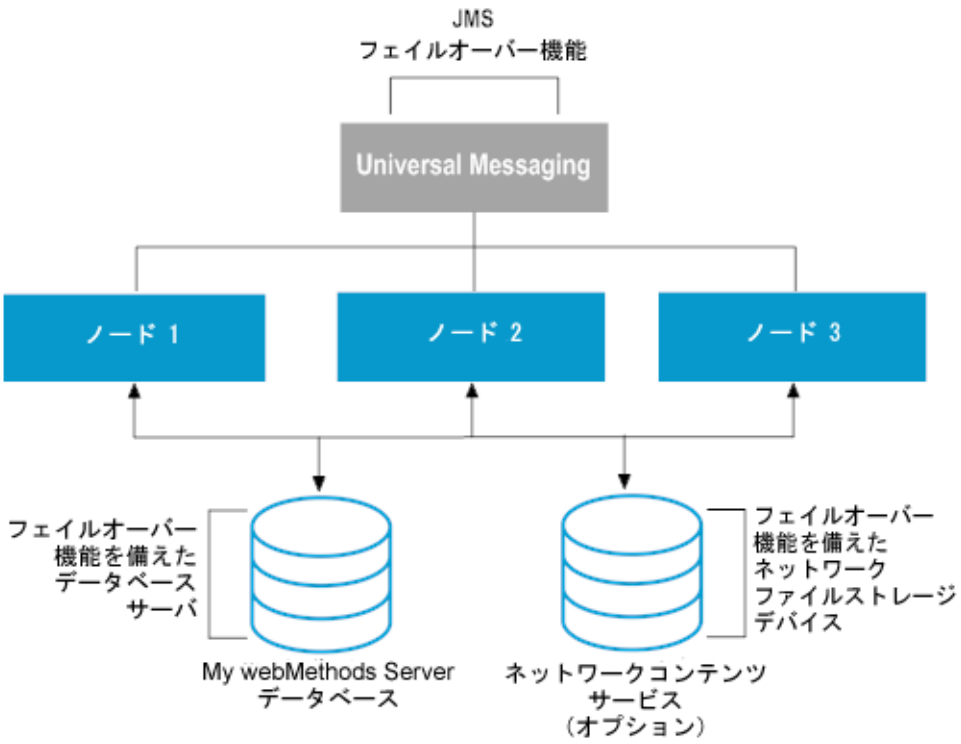
My webMethods Server クラスタでの高可用性

My webMethods Server クラスタ単独では高可用性が提供されません。クラスタ内のノードが保存するデータの可用性も確保する必要があります。高可用性を提供するには、My webMethods Server データベースによって使用されるデータベースサーバにフェイルオーバー機能が必要です。

クラスタでデフォルトのシステムコンテンツサービスを使用している場合、システムコンテンツサービスは My webMethods Server データベースを使用しているため、データベースサーバのフェイルオーバー機能によって保護されます。ただし、ネットワークコンテンツサービスを使用している場合にクラスタの高可用性をサポートするには、ネットワークファイルストレージデバイスにフェイルオーバー機能が備えられている必要があります。

このマニュアルでは、サードパーティのフェイルオーバーデバイスの設定については説明していません。

高可用な My webMethods Server クラスタの要件



My webMethods Server クラスタでのコンポーネント展開

My webMethods Server クラスタでは、ポートレットや展開可能パッケージなどのコンポーネントを展開する場合に、非対称モードが使用されます。

非対称コンポーネント展開モードでは、ノードに展開されたコンポーネントは、クラスタ内の他のノードに自動的には展開されません。非対称展開モードでは、展開する必要がある各ノードにコンポーネントを手動でローカル展開する必要があります。このモードでは、クラスタの異なるノードで異なるバージョンのコンポーネントを同時に実行することができます。コンポーネントの登録は My webMethods Server データベースに一元化されているため、登録は、コンポーネントが最初に展開されるノードによって 1 回のみ実行されます。この依存性が原因で、コンポーネントの登録または My webMethods Server 分類に影響する可能性がある変更をコンポーネントに対して行うと、異なるノードにある他のバージョンのコンポーネントが動作しなくなる場合があります。コンポーネントに対して以下のような変更を行うと、My webMethods Server データベーススキーマが変更される可能性があります。

- ポートレット設定の追加または削除
- DBO (Dynamic Business Object) テーブルのフィールドの追加、削除または変更
- タスクタイプビジネスデータのフィールドの追加、削除または変更
- My webMethods Server 分類の追加、削除または更新

非対称コンポーネント展開モードでは、変更したコンポーネントを 1 つ以上のノードに展開してテストした後で、クラスタ全体にそのコンポーネントを展開できます。

クラスタの役割

クラスタ内のノードに割り当てることができる、いくつかのクラスタの役割があります。以下に例を示します。

- [**自動展開**] クラスタ役割では、([自動展開] 役割を持つノードの) *Software AG_directory* \MWS\server*serverName*\deploy ディレクトリにコピーされたポートレットの自動展開が可能です。クラスタには複数の自動展開ノードを設定できます。この役割はデフォルトで有効ですが、自動展開機能を使用する場合にのみ必要です。実稼動環境では、サーバが許可なく変更される可能性を減らすため、多くの場合、この役割を無効にすることが求められます。この役割はクラスタの高可用性に影響を与えません。
- [**通知**] クラスタ役割は、電子メール通知およびユーザに送信される通知の書式設定と送信を管理します。クラスタには複数の通知ノードを設定できます。[通知] 役割はデフォルトで有効ですが、実稼働環境で通知を使用する場合にのみ必要です。たとえば、Task Engine の実行中に [通知] 役割が有効になっていない場合、ユーザはタスク通知を受信しません。高可用性の場合、この役割に複数のクラスタノードを設定することをお勧めします。すべての通知ノードが停止した場合、通知要求メッセージはキューに蓄積され続けます。1 つ以上の通知ノードがオンラインに戻り、処理を開始するまで、通知の生成および配信は行われません。
- [**検索**] 役割は埋め込み検索エンジンに公開されているすべてのコンテンツのインデックス作成、検索インデックスの保守、および検索の実行を管理します。[検索] 役割は、デフォルトで有効です。検索機能が必要なすべてのクラスタノードで有効にしてください。[検索] 役割はサーバ操作にとって重要です。この役割がないと動作しない主要な機能があります。
  - 権限の管理
  - [ワークスペース管理] と [ワークスペースへの追加]

  検索のパフォーマンスと信頼性を向上させるため、各クラスタノードには検索インデックスのローカルコピーが保持されます。1 つのクラスタノードで検索インデックスが破損した場合、そのノードで検索インデックスを再構築する間、ロードバランサからそのノードを削除できます。その間、他のノードでは要求の処理を継続できます。
- [**タスクエンジン**] クラスタ役割は、タスクのキューイング、タスクルールの処理、タスクデータの検索と取得など、Task Engine のすべてのアクティビティを管理します。クラスタは複数の Task Engine 役割を持つことができます。この役割はデフォルトで有効ですが、クラスタで実際に Task Engine を実行している場合にのみ必要です。高可用性の場合、特定のノードがロードバランサ設定に含まれておらず、エンドユーザまたは Process Engine の要求を処理しない場合を除き、すべてのノードでこの役割を有効にすることをお勧めします。

クラスタの各種役割の割り当てのガイドライン

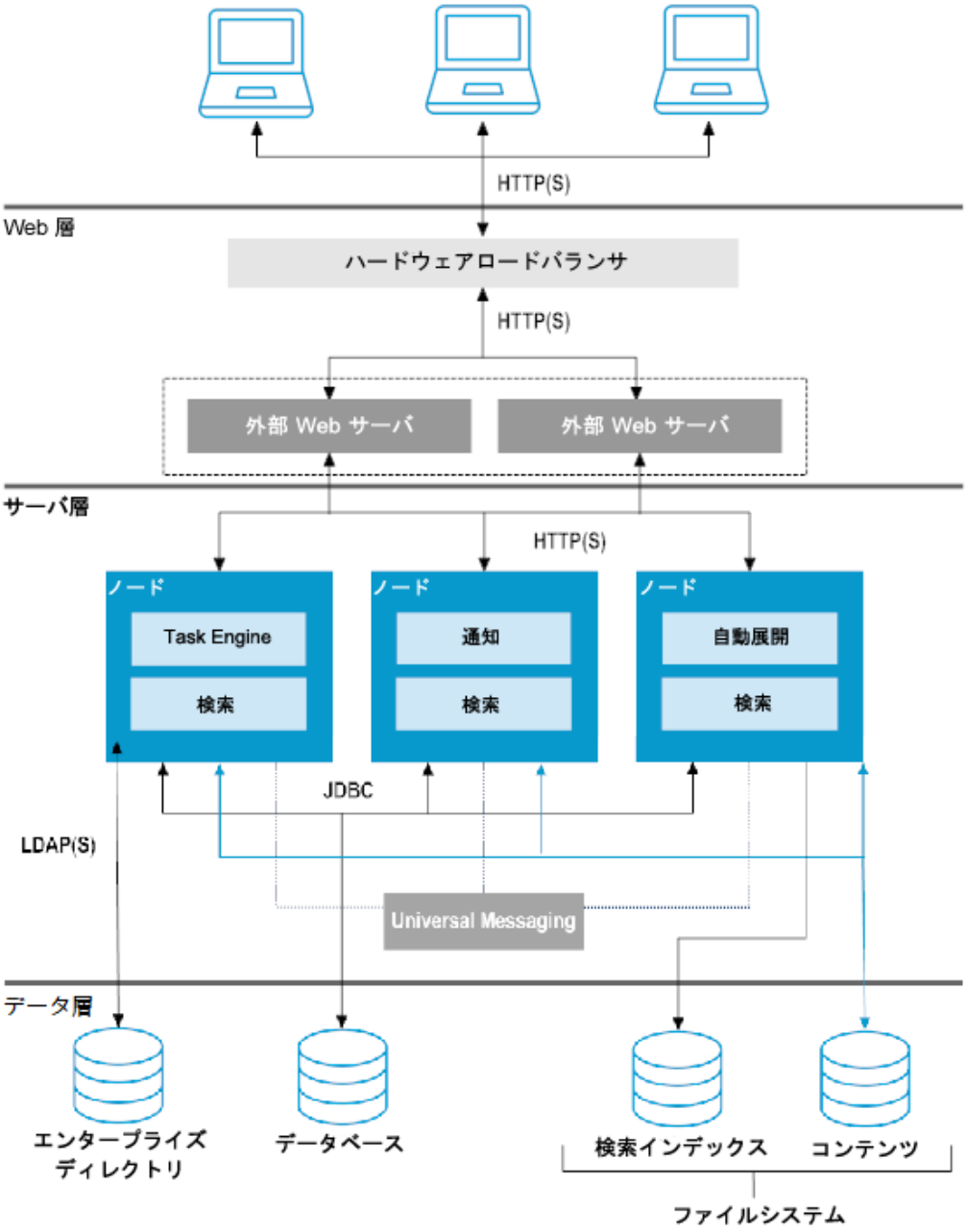
クラスタ内の各ノードには、以下のガイドラインに従ってクラスタ役割を割り当てることができます。クラスタノードに割り当てられた役割を変更した場合、割り当て内容を有効にするには、ノードを再起動する必要があります。

- [自動展開]、[Task Engine]、および [通知] 役割は、必要に応じて任意の数のノードに割り当てることができます。
- 1 つ以上の役割を実行するノードを、さらに分割してクラスタリングできます。スケーラビリティ上の要件から、追加のクラスタ役割を設定して、トラフィックの増加に対応しなければならない場合に、このような処理が必要になることがあります。
- 各ノードの設定データは、cluster.xml ファイルに格納されます。クラスタのプロパティの適切なデータ構造と値がわかっている場合は、cluster.xml ファイルでクラスタのデータを直接編集できます。詳細については、228 ページの「cluster.xml ファイルの使用」を参照してください。
- クラスタ設定に対する変更を有効にするには、各ノードを再起動する必要があります。

My webMethods Server クラスタの計画

以下の図は、My webMethods Server で設定できる多くの分散展開のタイプの 1 つを示しています。

My webMethods Server クラスタネットワークのダイアグラム



上記の図に示したクラスタでは、それぞれ独自の役割を持つ 3 つのサーバマシンが稼動しているものと想定しています。Web 層で定義されているマシンには、ハードウェアベースのロードバランサと 2 台の Web サーバマシンが含まれます。以降の節では、管理者がクラスタを構築するために実行するプロセスを示しています。

負荷分散された外部 Web サーバクラスタ設定に Apache を設定して展開する手順については、43 ページの「My webMethods Server での Web サーバの使用」で説明しています。

My webMethods Server クラスタ構築の考慮事項

My webMethods Server クラスタを構築するときは、以下の点を考慮します。

- クラスタ内のすべてのサーバは、*同じ* My webMethods Server データベースを使用する必要があります。クラスタ内の各ノードをインストールするときは、同じ My webMethods Server データベース JDBC URL を使用することを確認してください。
- クラスタ内の各ノードには、*t同じ* webMethods アプリケーション (Task Engine など)、My webMethods ユーザインタフェースおよび言語パックのセットをインストールする必要があります。
- My webMethods Server クラスタを構築するには、『*Installing Software AG Products*』の指示に従って、クラスタ内の各ノードをインストールします。以下の情報を入力する必要があります。
  - 各サーバインスタンスの HTTP ポート番号。デフォルト値は 8585 です。2 つのサーバノードを同じマシン上にインストールする場合は、ノードごとに異なる HTTP ポートを使用する必要があります。
  - My webMethods Server データベースの JDBC URL。URL は、データベースサーバのフェイルオーバーをサポートするために必要な追加のプロパティを含んでいる必要があります。
  - JMS プロバイダとして使用する Universal Messaging サーバまたは Universal Messaging クラスタの URL。My webMethods Server ノードは、Universal Messaging URL が指定されていない場合、または Universal Messaging サーバが使用できない場合、保守モードになります。インストール中に Universal Messaging URL を指定しなかった場合は、Administrator としてログインし、My webMethods の [クラスタ管理] ページで URL を設定する必要があります。
- クラスタ内のノードは、ニーズに応じて同時に起動することも、1 つずつ起動することもできます。各ノードを最初に起動するときに、ブートストラップアクティビティが実行されて、My webMethods Server データベースから共有設定ファイルがロードされます。ノードは自動的にクラスタに追加されます。217 ページの「クラスタへのノードの追加」の手順に従って、クラスタにノードを手動で追加することもできます。

  重要:
  クラスタノードを最初に起動している間にクラスタ設定を変更しないでください。ノードでは、最初の起動時に自動的に自己設定が実行されるため、起動中に設定を変更するとノードが正しく設定されない危険性があります。

  ノードの起動が正常に完了したら、ブラウザウィンドウを開き、システム管理者として My webMethods Server にログインして、ノードが正常に動作していることを確認できます。
- 実稼動クラスタを作成している場合は、以下の手順を実行します。
  - ネットワークコンテンツサービスが必要であることが判明している場合は、347 ページの「新規コンテンツサービスの設定」の手順に従って設定します。

- 既にロードバランサまたは外部 Web サーバが設定されている場合は、213 ページの「フロントエンド URL の設定」の手順に従って、それらを指すようにフロントエンド URL を変更します。

クラスタ設定の変更

以下の機能を実行して、スタンドアロンサーバ、クラスタまたはクラスタ内のノードの設定を変更できます。

フロントエンド URL の設定

My webMethods Server クラスタと通信するすべてのユーザおよびすべての webMethods アプリケーションが、クラスタ用に設定されたフロントエンド URL を使用する必要があります。この値は、デフォルトで、クラスタの 1 つ目のノードの URL となります。この URL は、ロードバランサまたは外部 Web サーバを指すことができます。1 つのクラスタに設定できるフロントエンド URL は 1 つだけです。

クラスタのフロントエンド URL を変更するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [My webMethods] > [クラスタ設定] > [拡張またはクラスタ設定]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [クラスタ管理] > [拡張またはクラスタ設定]
2. [**MWS フロントエンド URL**] フィールドに、完全修飾 URL を入力します。
   - `http://host_name:port_number`

     または
   - `https://host_name:port_number` (セキュアな接続のみ使用してクラスタにアクセスする場合)

     port_number は [**MWS フロントエンドのセキュアポート**] フィールドと同じ値であることが必要です。
3. セキュアな My webMethods Server コンテンツにアクセスするには、[**MWS フロントエンドのセキュアポート**] フィールドに HTTPS ポート番号を入力します。

   デフォルトの値は `-1` です。これは、クラスタ用に設定されたセキュアなフロントエンド URL がないことを意味します。
4. [**サブミット**] をクリックします。
5. My webMethods Server クラスタを再起動します。

Universal Messaging サーバへの接続の設定

デフォルトでは、My webMethods Server は Software AG Universal Messaging サーバを Java Message Service (JMS) プロバイダとして使用します。ただし、Universal Messaging サーバへの接続はデフォルトでは設定されていません。My webMethods Server クラスタ内のノードを初めて起動すると、ノードは保守モードで起動します。保守モードを終了するには、クラスタノードにログインし、Universal Messaging サーバへの接続を設定する必要があります。クラスタ内の他のノードがネットワーク経由でアクセス可能な場合は、My webMethods Server は自動的に設定を反映します。My webMethods Server がノードの設定を自動的に更新しない場合は、ノードにログインし、[クラスタ管理] ページからクラスタ設定を再ロードします。

注:
クラスタ内のノードは、Universal Messaging サーバとの接続が切断されると、保守モードになります。サーバが使用可能になると、ノードは保守モードを終了します。

Universal Messaging サーバへの接続を設定するには

1. 次のページに移動します。
   - My webMethods の管理者として、次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [My webMethods] > [クラスタの設定]
   - sysadmin として、次のように選択します。[**Administration Dashboard**] > [設定] > [クラスタ管理]
2. [**拡張またはクラスタ設定**] タブで、[**JNDI プロバイダ URL**] フィールドに Universal Messaging サーバの場所を指定します。

   ローカルインストールの Universal Messaging サーバのデフォルトの URL は次のとおりです。

   ```
   nsp://localhost:9000
   ```
3. [**サブミット**] をクリックします。

Universal Messaging サーバへのセキュア接続の設定

Universal Messaging において [**クラスタ設定**] ページの [**拡張またはクラスタ設定**] タブで、My webMethods クラスタのノードごとに別々に、My webMethods Server へのセキュア接続を設定します。

Universal Messaging サーバへの SSL 接続を設定するには

1. My webMethods 管理者としてクラスタノードにログオンします。
2. [**アプリケーション**] > [管理] > [My webMethods] > [クラスタ設定] > [拡張またはクラスタ設定] に移動します。

3. [**拡張またはクラスタ設定**] タブで、[**拡張設定**] をクリックします。

4. [**SSL 設定**] ダイアログで、次のフィールドの値を指定します。

次の表に、SSL を使用して Universal Messaging に接続するために設定するフィールドを示します。

フィールド	説明
[キーストアのパス]	クライアント証明書用キーストアの場所への完全修飾パス。
[キーストアのパスワード]	指定されたキーストアのパスワード。
[キーストアの証明書名]	クライアント証明書検証が Universal Messaging で有効な場合に必要です。認証に使用する指定キーストアの証明書のエイリアスです。クライアント証明書検証の詳細については、Universal Messaging のマニュアルを参照してください。
[トラストストアのパス]	CA 証明書用トラストストアの場所への完全修飾パス。
[トラストストアのパスワード]	指定されたトラストストアのパスワード。
[SSL プロトコル]	オプション。セキュア接続に使用するプロトコルのタイプ。デフォルト値は TLS です。
[暗号スイート]	オプション。暗号化に使用する暗号スイートの名前のカンマ区切りリスト。使用可能な暗号スイートの詳細については、Universal Messaging のマニュアルを参照してください。

5. [**サブミット**] をクリックします。

My webMethods Server では、ノードのプロファイル設定に現在のクラスタノードの SSL 設定を保存します。Universal Messaging の custom_wrapper.conf ファイルに My webMethods Server SSL を設定した場合、My webMethods の SSL 設定を変更すると、custom_wrapper.conf ファイルの設定が上書きされます。

また、Universal Messaging の custom_wrapper.conf ファイルを使用して、My webMethods Server への SSL 接続を設定することもできます。custom_wrapper.conf ファイルの SSL 設定を変更する方法、および共通ディレクトリサービス (CDS) 用の Universal Messaging への SSL 接続を設定する方法については、215 ページの「CDS 用の Universal Messaging サーバへのセキュア接続の設定 」を参照してください。

CDS 用の Universal Messaging サーバへのセキュア接続の設定

My webMethods Server クラスタノードと Universal Messaging サーバ間の SSL 接続を設定するには、My webMethods Server の custom_wrapper.conf ファイルに追加の JVM プロパティを指定します。Universal Messaging に Integration Server および共通ディレクトリサービス (CDS) を含むクラスタインストール用の SSL 接続を設定する場合、CDS を使用する Integration Server の custom_wrapper.conf ファイルにもこれらのプロパティを追加する必要があります。

custom_wrapper.conf ファイルの詳細については、「97 ページの「My webMethods Server の JVM 設定」」を参照してください。Integration Server の custom_wrapper.conf ファイルの編集の詳細については、*webMethods Integration Server Administrator's Guide*を参照してください。

次の表に、CDS の Universal Messaging サーバへのセキュア接続を設定するために追加する JVM プロパティを示します。

プロパティ	説明
`nirvana.ssl.keystore.path`	クライアント証明書用キーストアの場所への完全修飾パス。
`nirvana.ssl.keystore.pass`	指定されたキーストアのパスワード。
`nirvana.ssl.keystore.cert`	クライアント証明書検証が Universal Messaging で有効な場合に必要です。認証に使用するために指定されたキーストアの証明書のエイリアスです。
`nirvana.ssl.truststore.path`	CA 証明書用トラストストアの場所への完全修飾パス。
`nirvana.ssl.truststore.pass`	指定されたトラストストアのパスワード。
`nirvana.ssl.protocol`	セキュア接続に使用するプロトコルのタイプ。セキュア接続用のプロトコルを指定しない場合、My webMethods Server および CDS では TLS を使用します。

Universal Messaging Server への接続クレデンシャルの設定

デフォルトでは、My webMethods Server および CDS は mwsJMSAdmin ユーザを使用して Universal Messaging サーバに接続します。特定の Universal Messaging のセキュリティ設定では、メッセージング領域に対して mwsJMSAdmin ユーザの権限を明示的に付与する必要があります。

My webMethods Server の custom_wrapper.conf ファイル、または CDS をホストする Integration Server インスタンスの custom_wrapper.conf ファイルに次のシステムプロパティを設定することにより、特定の My webMethods Server ユーザを Universal Messaging に接続するように設定できます。

次の表に、Universal Messaging へ接続するためのクレデンシャルを設定するシステムプロパティを示します。

パラメータ	説明
`lax.nl.env.PRINCIPAL`	Universal Messaging サーバに接続するユーザの名前。
`lax.nl.env.PASSWORD`	Universal Messaging サーバで認証するユーザのパスワード。

Universal Messaging サーバに対して適切なアクセス権を持つユーザが設定されている必要があります。Universal Messaging セキュリティ設定の詳細については、Universal Messaging のマニュアルを参照してください。

My webMethods Server の custom_wrapper.conf ファイルの使用の詳細については、94 ページの「Java サービスラッパー」を参照してください。

Integration Server の custom_wrapper.conf ファイルの使用の詳細については、『*webMethods Integration Server Administrator's Guide*』を参照してください。

My webMethods Server クラスタ ID

複数の My webMethods Server クラスタ、または個々のノードは、同じ Universal Messaging Server を使用することができます。ノードがクラスタに属していることを識別するために、My webMethods Server は [**MWS クラスタ ID**] フィールドでクラスタに一意の識別子を生成します。My webMethods Server によって [**MWS クラスタ ID**] が JMS トピックおよびキューに付加され、メッセージが区別されます。これは、Universal Messaging を JMS プロバイダとして使用する場合のクラスタを対象としています。[**MWS クラスタ ID**] フィールドは編集できません。

クラスタへのノードの追加

ノードは、最初に起動したときに、クラスタに自動的に追加されます。新しいノードは、クラスタ内の他のノードと同じ My webMethods Server データベースを使用する必要があります。また、クラスタ内の各ノードで同じ webMethods アプリケーション (Task Engine など)、My webMethods ユーザインタフェースおよび言語パックのセットを持っている必要があります。

クラスタにノードを追加するには

1. 『*Installing Software AG Products*』の指示に従って、クラスタノードをインストールします。以下の情報を入力する必要があります。

   - このサーバインスタンスの HTTP ポート番号。デフォルト値は 8585 です。
   - クラスタで使用する My webMethods Server データベースの JDBC URL。この URL は、データベースサーバのフェイルオーバーをサポートするために必要な追加のプロパティを含んでいる必要があります。

2. My webMethods Server インスタンスを起動します。

   追加のノードを最初に起動するときに、ブートストラップアクティビティが実行されて、My webMethods Server データベースから共有設定ファイルがロードされます。ノードは自動的にクラスタに追加されます。

3. 新しいノードが初期化された後、226 ページの「クラスタ内のすべてのノードの再起動または停止」の説明に従って、[**クラスタ管理**] ページでクラスタ設定を再ロードするか、クラスタを再起動します。

クラスタ内のノードの変更

クラスタ内のノードまたは My webMethods Server のスタンドアロンインスタンスのいくつかのプロパティを変更できます。ノードの一般的な変更理由には、HTTPS ポートの指定によるセキュリティの追加、外部 Web サーバのサポートの追加、クラスタ役割の変更などがあります。クラスタノードまたはスタンドアロンサーバのプロパティを手動で変更することもできます。228 ページの「cluster.xml ファイルの使用」を参照してください。

スタンドアロンサーバまたはクラスタ内のノードを変更するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [My webMethods] > [クラスタ設定] > [拡張またはクラスタ設定]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [クラスタ管理] > [拡張またはクラスタ設定]
2. スタンドアロンサーバまたはクラスタ内のノードに対して、以下のいずれかを行います。
   - [**ホスト**] フィールドで、ノードが実行されるマシンのホスト名 (該当する場合はドメインを含む) または IP アドレスを変更します。

     指定したホスト名または IP アドレスは、ノードが実行される正しいマシンに解決される必要があります。この値は、クラスタ内のすべてのノードに対して有効である必要があります。サイトのホスト名とネットワーク設定の詳細については、ネットワーク管理者に確認してください。
   - [**HTTP ポート**] フィールドで、HTTP リスナーが使用するポート番号を変更します。

     このフィールドには必ず有効なポート番号を入力する必要があります。
   - [**HTTPS ポート**] フィールドに、HTTPS リスナーが使用するポート番号を入力します。

     このフィールドに値 0 (ゼロ) を入力すると、リスナーが無効になります。

     注:
     My webMethods Server には、HTTPS リスナーの設定とテストに使用できるサンプル HTTP 証明書が含まれています。このサンプルは *Software AG_directory* \MWS\server*serverName*\config\localhost.p12 ファイルにあります (34 ページの「セキュア接続に使用される証明書」を参照)。実稼動環境では、Verisign などの信頼性のある認証局から実際の証明書を取得してください。
   - ノードの 1 つ以上の役割を選択するには、役割のチェックボックスをオンにします。
3. [**サブミット**] をクリックします。
4. 変更した内容を有効にするには、「226 ページの「クラスタ内のすべてのノードの再起動または停止」」または「225 ページの「クラスタ内の個々のノードの再起動または停止」」の手順に従って、クラスタまたはスタンドアロンサーバを再起動します。

ノードへの [検索] 役割の割り当て

[検索] 役割は、デフォルトで有効です。必要に応じて、クラスタのノードに [検索] 役割を割り当てることができます。

ノードに [検索] 役割を割り当てるには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [My webMethods] > [クラスタ設定] > [拡張またはクラスタ設定]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [クラスタ管理] > [拡張またはクラスタ設定]
2. 役割を割り当てるノードの [**検索**] オプションを選択し、[検索] 役割を割り当てないノードのオプションをクリアします。
3. 変更した内容を有効にするには、「226 ページの「クラスタ内のすべてのノードの再起動または停止」」または「225 ページの「クラスタ内の個々のノードの再起動または停止」」の手順に従って、クラスタまたはスタンドアロンサーバを再起動します。

ノードの名前変更

クラスタ内のノードには、クラスタにノードを追加したときに自動的にノード名が割り当てられます。My webMethods Server は、My webMethods 内の [**拡張またはクラスタ設定**] ページの [**名前**] 列と、[**クラスタの状態および制御**] の [**ノード名**] 列にクラスタノードの名前を表示します。

次のように、`mws update` コマンドを使用してクラスタノードの名前を変更できます。

クラスタ内のノードの名前を変更するには

1. ノードをホストするマシンのコマンドラインプロンプトで、My webMethods Server の `bin` ディレクトリに移動します。

   Software AG_directory \MWS\bin

2. 次のコマンドを実行して、ノード名を変更します。

   ```
   mws -s instanceName update -Dnode.name=newNodeName
   ```

   ここで、*instanceName* は、ノードに対応するサーバインスタンスの名前で、*newNodeName* は、クラスタノードの新しい名前です。インスタンス名はそのままです。

クラスタノード名は次のファイルで更新されます。

- cluster.xml

- *Software AG_directory* \MWS\server*serverName*\config\cluster.node.properties
- *Software AG_directory* \MWS\server*serverName*\config\cluster.node.properties.bak
- *Software AG_directory* \profiles\MWS_*serverName*\configuration\custom_wrapper.conf

クラスタからのノードの削除

My webMethods Server クラスタのノードが使用されなくなった場合は、以下の手順に従ってクラスタからノードを削除する必要があります。

クラスタからノードを削除するには

1. 225 ページの「クラスタ内の個々のノードの再起動または停止」の手順に従って、クラスタから削除するノードを別のノードから停止します。

2. 以下のコマンドを入力して、My webMethods Server データベースから cluster.xml ファイルを取得します。

```
mws getconfig cluster.xml
```

3. cluster.xml ファイルをテキストエディタで開きます。ファイルは以下の場所にあります。

   Software AG_directory \MWS\server*serverName*\config

4. cluster.xml ファイルで、削除するノードを検索し、このノードの <Component> 要素全体を削除して、ファイルを保存します。次に例を示します。

```
<Component class="com.webmethods.portal.system.cluster.impl.Server"
    enabled="true" name="nodeName">
.
.
.
</Component>
```

5. 以下のコマンドを入力して、変更した cluster.xml ファイルを My webMethods Server データベースに展開します。

```
mws putconfig cluster.xml
```

6. *serverName*\config ディレクトリから cluster.xml ファイルを削除します。

7. 226 ページの「クラスタ内のすべてのノードの再起動または停止」の説明に従って、[**クラスタ管理**] ページでクラスタ設定を再ロードするか、クラスタを再起動します。

ノードのアンインストール

クラスタ内のノードである My webMethods Server インスタンスをアンインストールすると、ノードはクラスタ設定から自動的に削除されます。アンインストール時に My webMethods Server

データベースが使用できない場合は、220ページの「クラスタからのノードの削除」の手順に従ってノードを手動で削除する必要があります。

データベース接続再試行の変更

サーバと My webMethods Server データベースとの接続が切断された場合、サーバは接続の再確立を試みます。たとえば、高可用性環境でデータベースのフェイルオーバーが実行される場合などに再試行が行われます。サーバにおける接続の再試行回数、および再試行の間隔を変更できます。クラスタ内に複数のサーバがある場合は、各マシンで個別にこれらの値を変更する必要があります。また、同じマシンに複数のサーバインスタンスがある場合は、各サーバインスタンスの値を変更する必要があります。

各サーバインスタンスの My webMethods Server データベース接続情報は、以下の場所にある mws.db.xml ファイルに保持されます。

Software AG_directory \MWS\server*serverName*\config

次の表に、データベース接続再試行で使用可能な設定を示します。

値	説明
再試行回数	接続の切断、デッドロック、その他の通常は想定されない SQL エラーが発生した場合に My webMethods Server で SQL ステートメントの再実行を試みる回数。 ■ 0: サーバは接続の再確立を試みません。 ■ 1 以上の整数: サーバは、指定された回数だけ接続を再試行します。 通常、再試行回数は 3 回以下に設定します。4 回以上の再試行が必要になる場合は、接続やデータベースサーバに問題があります。
再試行間隔	サーバが再試行を行う間隔 (ミリ秒単位)。再試行間隔の値は、フェイルオーバーが実行されるのに十分な時間とする必要があり、データベースサーバの設定に依存します。

単一の My webMethods Server データベース接続の再試行動作を変更するには、以下の手順に従います。

サーバインスタンスのデータベース再試行動作を変更するには

1. mws.db.xml ファイルをテキストエディタで開きます。ファイルは以下の場所にあります。

   Software AG_directory \MWS\server*serverName*\config

2. クラスタ内の各ノードの再試行回数を変更するには、mws.db.xml の <PARAMS></PARAMS> タグ内に `<RETRYCOUNT>` 要素を追加します。

```
<RETRYCOUNT>2</RETRYCOUNT>
```

3. クラスタ内の各ノードを対象として、ミリ秒単位の再試行間隔を変更するには、mws.db.xml の <PARAMS></PARAMS> タグ内に <RETRYDELAY> 要素を追加します。

```
<RETRYDELAY>300</RETRYDELAY>
```

結合された要素は、以下のようになっています。

```
<PARAMS>
   .
   .
   .
   <!-- retry count on error -->
   <RETRYCOUNT>2</RETRYCOUNT>
   <!-- delay in ms between retries  -->
   <RETRYDELAY>300</RETRYDELAY>
</PARAMS>
```

4. ファイルを保存して、My webMethods Server インスタンスを再起動します。

5. クラスタ内の各ノードでこの手順を繰り返します。

JMS 通信にデータベースを使用するための My webMethods Server の設定

My webMethods Server と Integration Server の共通ディレクトリサービスコンポーネント間の通信のため、または少数の JMS イベントを交換する My webMethods Server クラスタのためにデータベースを JMS プロバイダとして使用できます。詳細については、223 ページの「JMS プロバイダとしてデータベースを使用するときの考慮事項」を参照してください。

My webMethods Server を設定してデータベースを JMS プロバイダとして使用するには

1. getconfig コマンドを使用して、My webMethods Server データベースから cluster.xm ファイルを取り出します。

2. <Cluster> 要素で、次のように useDbJms="true" 属性を挿入します。

```
<Cluster useDbJms="true" clusterId="706790370
```

3. ファイルを保存して putconfig コマンドを実行し、ファイルをデータベースに展開します。

4. *serverName*\config ディレクトリから cluster.xml ファイルのローカルコピーを削除します。

cluster.xml ファイルを削除しないと、このノードでは引き続きローカルバージョンのファイルが使用されます。cluster.xml ファイルの使用の詳細については、228 ページの「cluster.xml ファイルの使用」を参照してください。

JMS プロバイダとしてデータベースを使用するときの考慮事項

My webMethods Server クラスタと CDS では、ノード間の同期に JMS プロバイダを使用する必要があります。My webMethods Server ノード間、および CDS を使用する場合に My webMethods Server と Integration Server との間を同期するデフォルトの JMS プロバイダは Universal Messaging です。

222 ページの「JMS 通信にデータベースを使用するための My webMethods Server の設定」 で説明されているように、クラスタを同期する JMS プロバイダとしてデータベースを使用するように My webMethods Server を設定できます。JMS プロバイダとしてデータベースを使用するときは、次の点を考慮します。

- My webMethods Server はクラスタのメッセージングに実稼働データベースを使用します。JMS に個別のデータソースを設定することはできません。
- クラスタノードが多数のイベントを交換する場合、実稼働データベースの負荷が増加します。これにより、クラスタの全体的なパフォーマンスに影響を与える場合があります。
- My webMethods Server は JMS のフェイルオーバー機能を提供しておらず、JMS プロバイダのフェイルオーバー機能に依存しています。

クラスタのリソースロック設定の変更

My webMethods Server クラスタで追加コンポーネントをインストールする場合、クラスタノードは更新されるリソースのデータベースのロックを取得します。コンポーネントのインストールは、特定のリソースを長時間ロックしたままにすることがあります。クラスタノード間でリソースを同時にロックしようとしてタイムアウトが発生しないようにするために、My webMethods Server の custom_wrapper.conf ファイルに次の追加の JVM パラメータを使用して、ロックの有効性と待機時間を設定できます。

```
com.webmethods.task.lock.max.time

com.webmethods.task.lock.max.wait.time
```

My webMethods Server クラスタのリソースロック設定を変更するには

1. My webMethods Server をシャットダウンします。
2. `custom_wrapper.conf` ファイルをテキストエディタで開きます。

   ファイルは次の場所にあります。

   Software AG_directory \profiles\MWS_*serverName*\configuration
3. `custom_wrapper.conf` ファイルに、次の追加パラメータを追加することにより、リソースロックとロック待機時間の制限を指定します。

次の表に、クラスタのリソースロックを設定する追加の JVM パラメータを示します。

パラメータ	説明
`-Dcom.webmethods.task.lock.max.wait.time=time_in_milliseconds`	ノードがリソースのロックを取得するまで待機できる最大時間 (ミリ秒)。デフォルト値は 300000 (5 分) です。
`-Dcom.webmethods.task.lock.max.time=time_in_milliseconds`	ノードがリソースをロックできる最大時間 (ミリ秒)。デフォルト値は 300000 (5 分) です。

4. `custom_wrapper.conf` ファイルを保存して、My webMethods Server を再起動します。

`custom_wrapper.conf` ファイルに追加の JVM パラメータを設定する方法の詳細については、97 ページの「My webMethods Server の JVM 設定」を参照してください。

クラスタの消去スレッドの数の設定

ユーザがアイテムを消去すると、My webMethods Server はアイテムに削除のマークを付けます。事前設定されたスケジュールに従って、またはシステム管理者によるコンテンツ消去の手動トリガーの結果として、My webMethods Server がすべての保留アイテムを消去するまで、アイテムはシステムに残ります。

消去操作が My webMethods Server クラスタ内で実行される場合、自動展開の役割があるすべてのクラスタノードが保留アイテムの消去を試行します。多数のアイテムに削除のマークが付けられた場合、消去操作は長時間実行されることがあります。システムから削除されたアイテムを削除するスレッドの数を設定し、`purge.schedule.maxthreads` の追加の JVM プロパティを使用して消去作業を速くすることができます。プロパティのデフォルト値は `10` です。

消去実行スレッドの数を設定するには

1. custom_wrapper.conf ファイルをテキストエディタで開きます。

2. custom_wrapper.conf ファイルに、次の JVM プロパティを追加します。

   ```
   purge.schedule.maxthreads=number_of_threads
   ```

3. ファイルを保存して、My webMethods Server を再起動します。

   custom_wrapper.conf ファイルの使用の詳細については、97 ページの「My webMethods Server の JVM 設定」を参照してください。

クラスタの監視および制御

クラスタまたはスタンドアロンサーバ内のノードの状態情報を表示したり、個々のノードを再起動または停止したり、クラスタ全体を再起動または停止したり、個々のノードを保守モードにしたりできます。

クラスタの状態情報の確認

クラスタ内のノードまたはスタンドアロンサーバの状態情報を手動で更新できます。

クラスタ内のノードの状態情報を表示するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [My webMethods] > [クラスタ設定] > [クラスタの状態および制御]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [クラスタ管理] > [クラスタの状態および制御]
2. [**クラスタの状態および制御**] ページで状態情報を更新するには、[**リフレッシュ**] をクリックします。

次の表に、[クラスタの状態および制御] ページで表示されるフィールドを示します。

フィールド	説明
	ノードが実行されているかどうかを示すインジケータ。個別の状態を更新するには、ノードのアイコンをクリックします。すべてのノードの状態を更新するには、 をクリックします。状態アイコンの意味は以下のとおりです。 ノードは実行中です ノードは停止しています
[ノード名]	ノードに割り当てられているノード名。ノード名をクリックすると、ブラウザは現在のウィンドウを使用してノードに直接接続します。この名前は、219 ページの「ノードの名前変更」の手順に従って手動で変更できます。
[役割]	現在ノードに割り当てられているクラスタの役割。
[起動時間]	ノードが最後に起動されてからの時間 (日、時間、分)。
[アクティブなユーザ]	ノードのアクティブユーザ数。この数には、ゲストセッションおよび破棄されたセッションの数が含まれることがあるため、ログインしている正確なユーザ数を示さない場合があります。
アクション	ノードに対して実行できるアクションです。詳細については、225 ページの「クラスタ内の個々のノードの再起動または停止」を参照してください。

クラスタ内の個々のノードの再起動または停止

クラスタ内の個々のノードを再起動または停止できます。

クラスタ内の個々のノードを再起動または停止するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [My webMethods] > [クラスタ設定] > [クラスタの状態および制御]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [クラスタ管理] > [クラスタの状態および制御]
2. アクションを実行するノードで、以下のいずれかを実行します。
   - [**再起動**] をクリックします。

     ノードが停止して、● アイコンが表示されます。ノードが再起動されたら、アイコンが ■ に戻ります。
   - [**シャットダウン**] をクリックします。

     ノードが停止します。ノードを停止した後は、このページから再起動することができません。ホストマシンで My webMethods Server インスタンスを起動する必要があります。

クラスタ内のすべてのノードの再起動または停止

単一のコマンドを使用して、クラスタ内のすべてのノードを再起動または停止できます。

クラスタ内のすべてのノードを再起動または停止するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [My webMethods] > [クラスタ設定] > [クラスタの状態および制御]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [クラスタ管理] > [クラスタの状態および制御]
2. 以下のいずれかを行います。
   - [**クラスタの再起動**] をクリックします。

     テーブルの最初のノードから 1 つずつ順番にノードが停止および再起動されます。最初のノードがこのコマンドを発行するノードである場合は、次のノードから再起動が開始されます。コマンドを発行するノードは、他のすべてのノードが再起動された後に再起動されます。ノードが再起動された後、ページは自動的にはリフレッシュされません。1 分ほど経過した後に、手動でブラウザをリフレッシュする必要があります。クラスタが完全に再起動されたら、ログイン画面が表示されます。
   - [**クラスタのシャットダウン**] をクリックします。

すべてのノードが停止します。ノードを再起動するには、各ホストマシンで My webMethods Server インスタンスを起動する必要があります。

個々のノードの保守モード化

クラスタの個々のノードを保守モードにして、問題の診断や、ファイルシステムのクリーンアップや、サーバ設定の変更などの操作を実行できます。

My webMethods Server が保守モードの場合、サーバの動作は、ユーザのタイプによって異なります。

- **My webMethods およびシステム管理者**: My webMethods 管理者およびシステム管理者は、サーバが保守モードの間、My webMethods Server にログインし、通常と同様の通信ができます。
- **My webMethods ユーザ**: My webMethods ユーザは、サーバが保守モードの間、ログインできません。既にログインしている My webMethods ユーザは、別のサーバページを読み込もうとすると、保守モードページにリダイレクトされます。

個々のノードを保守モードにするには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [My webMethods] > [クラスタ設定] > [クラスタの状態および制御]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [クラスタ管理] > [クラスタの状態および制御]
2. 保守モードにするノードについて、[アクション] 列の [**保守**] をクリックします。
3. [保守] ダイアログボックスで、次のフィールドの値を指定します。
   a. [**保守モード中**] フィールドで、[**はい**] をクリックします。
   b. [**経過時間**] フィールドで、以下の値のいずれかを選択します。
      - **永続** - My webMethods Server が再起動されたときノードは保守モードのままになる
      - **一時的** - My webMethods Server が再起動されたときノードの保守モードが解除される
   c. (オプション) [**原因**] フィールドで、ノードを保守モードにする理由を指定します。理由を指定しないと、My webMethods Server は保守ページでデフォルトの保守メッセージを表示します。
4. [**適用**] をクリックします。

[クラスタ設定] ページの [**クラスタの状態および制御**] タブで、My webMethods Server はノードの [保守中] 列に [**はい**] を表示します。

個々のノードの保守モードからの解除

個々のノードを保守モードから解除するには

1. 以下のいずれかの手順に従って、正しいページに移動します。
   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [My webMethods] > [クラスタ設定] > [クラスタの状態および制御]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [クラスタ管理] > [クラスタの状態および制御]
2. 保守モードを解除するノードについて、[アクション] 列の [**保守**] をクリックします。
3. [**保守モード中**] フィールドで、[**いいえ**] をクリックします。
4. [**適用**] をクリックします。

[クラスタ設定] ページの [**クラスタの状態および制御**] タブで、My webMethods Server はノードの [保守中] に [**いいえ**] を表示します。

クラスタからのコンポーネントの削除

クラスタ内のすべてのノードは同じ My webMethods Server データベースを使用するため、展開されたすべてのコンポーネントの登録情報はすべてのノードで共有されます。あるノードでコンポーネントを削除すると、データベースから登録が削除され、そのコンポーネントはどのノードでも使用できなくなります。

クラスタからコンポーネントを削除するには

- 各ノードの展開フォルダからコンポーネントファイルを削除します (各ノードで [自動展開] 役割が有効な場合)。

  または

- Deployer を使用して、クラスタ内の各ノードでコンポーネントをロールバックします。

cluster.xml ファイルの使用

cluster.xml ファイルには、スタンドアロンサーバまたはクラスタ内のすべてのサーバの設定情報が含まれています。このファイルは、My webMethods Server データベース内にあります。クラスタにノードを追加すると、My webMethods Server によってこのファイルに新規サーバが追加されます。また、[クラスタ管理] ページで変更を行うと、このファイルが変更されます。

「219ページの「ノードの名前変更」」および「220ページの「クラスタからのノードの削除」」の手順に従って、手動で cluster.xml ファイルを変更することもできます。

以下に、ファイルの一部を示します。この部分では、クラスタ内のノードの基本的な設定が示されています。

```
<Component class="com.webmethods.portal.system.cluster.impl.Server"
   enabled="true"
   name="server-one-node59581">
   <Properties host="server-one" name="http port="8585"/>
   <Properties host="server-one" name="https" port="0"/>
   <Role name="autodeploy"/>
   <Role name="taskengine"/>
   <Role name="notification"/>
</Component>
```

次の表に、ファイルの一部に含まれているプロパティと、それらの値の例を示します。

プロパティ	値の例
ノード名	server-one-node59581
ホスト名	server-one
HTTP ポート	8585
HTTPS ポート	0: HTTPS リスナが無効であることを示しています。
役割	このサーバでサポートされているクラスタ役割が [自動展開]、[Task Engine] および [通知] であることを示しています。

次の表に、cluster.xml ファイルを変更して実行するタスクを示します。

アクション	説明
ノードの追加	上記の例に示したような完全な <Component> 要素をファイルに追加します。この場合、このノードが既に存在しており、クラスタ内の他のノードと同じ My webMethods Server データベースを使用している必要があります。cluster.xml ファイル内のノード名は、そのノードのホストマシン上の custom_wrapper.conf ファイル内のノード名と同じであるか、またはノード起動時にコマンドライン (mws -n *nodeName* start) で指定する必要があります。
ノードの名前変更	ノードのノード名を変更します。
ノード属性の変更	ノード名、HTTP ポート、HTTPS ポート、クラスタ役割など、ノードの任意の属性を編集します。
ノードの削除	ノードを定義する <Component> 要素全体を削除します。
クラスタ設定の変更	クラスタプロパティの表の説明に従い、クラスタプロパティを編集します。

次の表に、cluster.xml ファイルを使用して設定できるクラスタのプロパティを示します。

プロパティ	説明
`frontEndUrl`	My webMethods Server クラスタの URL。完全修飾 URL を指定します (例: `http://host_name:port_number`、`https://host_name:port_number`)。デフォルト値は、クラスタの 1 つ目のノードの URL となります。
`frontEndSecurePort`	My webMethods Server クラスタがセキュアな通信で使用する HTTPS ポート。デフォルトの値は `-1` です。これは、設定されたセキュアなフロントエンド URL がないことを意味します。

cluster.xml ファイルの編集

cluster.xml ファイルを編集するには、My webMethods Server インスタンスが実行されている必要はありませんが、My webMethods Server データベースは実行されている必要があります。

cluster.xml ファイルを編集するには

1. クラスタノードをホストする任意のマシンのコマンドラインプロンプトで、サーバインスタンスの bin ディレクトリに移動します。

   Software AG_directory \MWS\bin

2. 以下のコマンドを入力して、My webMethods Server データベースから cluster.xml ファイルを取得します。

   ```
   mws getconfig cluster.xml
   ```

3. cluster.xml ファイルをテキストエディタで開きます。

   ファイルは次の場所にあります。

   Software AG_directory \MWS\server*serverName*\config

4. cluster.xml ファイルのバックアップコピーを作成します。

5. 必要に応じて cluster.xml ファイルの内容を変更します。

6. コマンドラインプロンプトで以下のコマンドを入力して、変更した cluster.xml ファイルを My webMethods Server データベースに展開します。

   ```
   mws putconfig cluster.xml
   ```

7. *serverName*\config ディレクトリから cluster.xml ファイルを削除します。

cluster.xml ファイルを削除しないと、このノードでは引き続きローカルバージョンのファイルが使用されます。

8. 226 ページの「クラスタ内のすべてのノードの再起動または停止」の手順に従って、クラスタを再起動します。

cluster.xml ファイルへの変更を元に戻すには

[クラスタ管理] ページを使用して cluster.xml ファイルを変更した後、その変更によってエラーが発生した場合は、cluster.xml.bak ファイルを使用して前の設定に戻すことができます。My webMethods Server インスタンスが実行されている必要はありませんが、My webMethods Server データベースは実行されている必要があります。

cluster.xml ファイルへの変更を元に戻すには

1. クラスタノードをホストする任意のマシンのコマンドラインプロンプトで、サーバインスタンスの bin ディレクトリに移動します。

   Software AG_directory \MWS\bin

2. 以下のコマンドを入力して、My webMethods Server データベースから cluster.xml.bak ファイルを取得します。

```
mws getconfig cluster.xml.bak
```

3. ファイルの名前を cluster.xml.bak から cluster.xml に変更します。

4. コマンドラインプロンプトで以下のコマンドを入力して、変更した cluster.xml ファイルを My webMethods Server データベースに展開します。

```
mws putconfig cluster.xml
```

5. *serverName*\config ディレクトリから cluster.xml ファイルを削除します。

   cluster.xml ファイルを削除しないと、このノードでは引き続きローカルバージョンのファイルが使用されます。

6. 226 ページの「クラスタ内のすべてのノードの再起動または停止」の手順に従って、クラスタを再起動します。

イメージからのクラスタノードの作成

クラスタを作成する方法の 1 つに、インストール済みの My webMethods Server インスタンスおよび他の製品が存在するマシンのイメージを保存して、そのイメージを新しいマシンに移動する方法があります。この方法を使用すると、インストール時間を短縮でき、新しいマシンに同一の設定を展開できます。単なる My webMethods Server のインストールではなく、完全に初期化が完了している My webMethods Server インスタンスのイメージを作成することをお勧めし

ます。My webMethods Server インスタンスは、最初に起動された後、管理者がインスタンスに正常にログインしてから停止すると完全に初期化された状態になります。

クラスタで新しいノードを使用する前に、いくつかの項目を変更する必要があります。新しいマシンは My webMethods Server データベースにアクセスできる必要がありますが、以下の変更を行うまでは My webMethods Server を起動しないでください。

イメージから作成されたノードを追加するには

1. 新しいイメージが配置されたマシンのコマンドラインプロンプトで、サーバインスタンスの bin ディレクトリに移動します。

   Software AG_directory \MWS\bin

2. 以下のコマンドを入力して、My webMethods Server データベースから cluster.xml ファイルを取得します。

   ```
   mws getconfig cluster.xml
   ```

3. cluster.xml ファイルをテキストエディタで開きます。

   ファイルは次の場所にあります。

   Software AG_directory \MWS\server*serverName*\config

4. 元のマシンのサーバインスタンスの <Component> 要素全体をコピーして、cluster.xml ファイルに貼り付けます。

5. 新しい <Component> 要素で、以下の変更を行います。

   a. `name` の値を、新しいノードを説明するノード名に変更します。

   b. `host` プロパティを、新しいマシンのホスト名に変更します。

      次に例を示します。

      ```
      name="server-one-node59581"><Properties
      ```

      ```
      host="server-one" name="http"
      ```

      これを、以下のように変更します。

      ```
      name="node2"><Properties
      ```

      ```
      host="server-two" name="http"
      ```

   c. 必要に応じて、新しいノードが担当するクラスタ役割を反映するように <Role> の値を変更します。

6. コマンドラインプロンプトで以下のコマンドを入力して、変更した cluster.xml ファイルを My webMethods Server データベースに展開します。

```
mws putconfig cluster.xml
```

7. *serverName*\config ディレクトリから cluster.xml ファイルを削除します。

   cluster.xml ファイルを削除しないと、このノードでは引き続きローカルバージョンのファイルが使用されます。

8. テキストエディタで、サーバインスタンスの custom_wrapper.conf ファイルを開きます。ファイルは次の場所にあります。

   Software AG_directory \profiles\MWS_*serverName*\configuration\

9. custom_wrapper.conf ファイルで、NODE_NAME ステートメントの値を変更して、ファイルを保存します。

   ```
   set.NODE_NAME=nodeName
   ```

10. My webMethods Server インスタンスを起動します。

11. 起動が正常に完了したら、ブラウザウィンドウを開き、SysAdmin として My webMethods Server にログインして、サーバインスタンスが正常に動作していることを確認します。

12. クラスタ内の各ノードを再起動します。

    ノードは、手動で再起動することも、226 ページの「クラスタ内のすべてのノードの再起動または停止」の手順に従って My webMethods Server 内から再起動することもできます。

クラスタノードでのアプリケーションのパーティション化

My webMethods Server クラスタにおいて、パーティション化とは、クラスタのノード間にアプリケーションを分割することをいいます。クラスタパーティションを設定することによって、どのアプリケーションをノードで実行し、どのアプリケーションを実行しないかを制御できます。パーティション化は、非対称モード (208 ページの「My webMethods Server クラスタでのコンポーネント展開」) とは異なるものであり、非対称モードではコンポーネントを手動でノードに展開します。パーティション化はコンポーネントの実際の実行を制御します。

パーティションが必要な理由

クラスタ内にパーティションを作成する実用的な理由は複数あります。

- クライアントの分割

  ロードバランサルールを使用することで、1 組のユーザをあるノードに、別の組のユーザを別のノードにルーティングすることができます。パーティション内のユーザは、そのユーザに提供するよう選択されたアプリケーションのセットにのみアクセスできます。

- アプリケーション管理を上回る優れた制御

特定のアプリケーションが 1 つのパーティション内でのみ実行されるため、残りのクラスタが通常操作を続行している間に簡単にアプリケーションを更新できます。

- キャッシュの安定性とメモリ使用量

  ある特定のノードで高トラフィックのためにパフォーマンスが低下した場合でも、他のノードのユーザが影響を受けることはありません。たとえば現在の Task Engine アーキテクチャでは、タスクは通常、メモリ内キャッシュに配置されます。キャッシュ内に存在するタスクインスタンスが多すぎる場合、使用可能なメモリを超過する可能性があります。

パーティションのガイドライン

クラスタ内にパーティションを作成する方法について、理解しておく必要のある項目がいくつかあります。

- デフォルトで、クラスタにはパーティションが 1 つあります。パーティションが 1 つしかない場合、明示的にノードをパーティションの一部としてリストしているかどうかに関係なく、すべてのノードがそのパーティションの一部になります。
- 個々のノードは 1 つのパーティションの一部になります。パーティションの一部として特別にリストされていないノードはすべて、設定ファイルに定義されている最後のパーティションのメンバーになります。234 ページの「パーティションの作成と変更」を参照してください。
- 各パーティションは、ポートレットが有効か無効かを管理する phaseProvider.xml ファイルに関連付けられます。パーティションが特定のフェーズプロバイダファイルに関連付けられていない場合は、デフォルトで、デフォルトフェーズプロバイダに関連付けられます。

パーティションの作成と変更

237 ページの「デフォルトのパーティション設定」で説明しているように、デフォルトでは、すべてのクラスタノードが含まれるパーティションが 1 つあります。追加のパーティションを作成または変更するには、パーティション設定ファイルを変更する必要があります。

My webMethods Server は設定ファイルを使用して、クラスタのパーティション化を管理します。

- clusterPartitions.xml: このファイルを変更して、クラスタにパーティションを追加し、各パーティションにノードを割り当てます。235 ページの「クラスタパーティションファイル」を参照してください。
- *partitionName*PhaseProvider.xml: これらのファイルを作成および変更して、ポートレットが有効か無効かを指定します。236 ページの「フェーズプロバイダファイル」を参照してください。

  パーティションごとに専用のフェーズプロバイダファイルを持つことができます。関連付けられた固有のフェーズプロバイダを持たないパーティションは、デフォルトの phaseProvider.xml ファイルを使用します。
- *partitionName*Portlets.properties: フェーズプロバイダごとに、これらのファイルの 1 つを作成および変更します。このファイルでは、特定のパーティションのフェーズプロバイダに

よって影響を受けるポートレットをリストします。237 ページの「ポートレットプロパティファイル」を参照してください。

パーティションごとに専用のポートレットプロパティファイルを持つことができます。パーティション内のすべてのポートレットが有効な場合、ポートレットプロパティファイルは必要ありません。

注:
ファイル名およびファイルの内容は大文字と小文字が区別されます。

My webMethods Server は、これらの設定ファイルを My webMethods Server データベースに保存します。これらのファイルにアクセスして変更できるようにするには、102 ページの「データベースに保存された設定ファイルの変更」を参照してください。

クラスタパーティションファイル

クラスタパーティションファイル (clusterPartitions.xml) では、クラスタのパーティションを定義し、各パーティションにノードを割り当てます。clusterPartitions.xml ファイルのパーティションのデータ構造は、次のとおりです。

```
<ClusterPartitions>
   <Partition name="partitionName"
      frontEndUrl="http://some.url.com">
      <Component name="componentName">
      </Component>
   </Partition>
</ClusterPartitions>
```

ここで、

次の表では、clusterPartitions.xml ファイルで設定可能な属性の期待値について説明します。

partitionName	パーティションに割り当てる名前。他の設定ファイルでは、この名前を使用してパーティションを識別します。
some.url.com	代替フロントエンド URL。 `frontEndUrl` 属性はオプションで、この属性を使用すると、クラスタ全体とは異なるフロントエンド URL をパーティションで使用できます。この属性を省略した場合、パーティションは cluster.xml ファイルの `frontEndUrl` 値を使用します。
componentName	パーティションの一部であるコンポーネント (クラスタノード) の名前。コンポーネント名は、cluster.xml ファイルのコンポーネント名と同じです。

clusterPartitions.xml のエントリは次のルールに従います。

- クラスタのパーティションごとに `Partition` エレメントが 1 つある。常に、1 つ以上の `Partition` エレメントがある。

- 各 Component エレメントは、パーティションのメンバーであるクラスタノードを示す。1 つのパーティションに指定できるノードは 0 個以上。
- ファイルに Partition エレメントが 1 つしかない場合、Component エレメントで示されているかどうかに関係なく、すべてのクラスタノードがそのパーティションのメンバーになる。
- Partition エレメント内に特別に示されていないクラスタノードは、すべて自動的に、ファイル内に定義されている最後のパーティションのメンバーになる。

フェーズプロバイダファイル

フェーズプロバイダファイル (*partitionName*PhaseProvider.xml) は、ポートレットがパーティションに対して有効か無効かを決定します。パーティションごとに専用のフェーズプロバイダファイルを持つことができます。関連付けられた固有のフェーズプロバイダを持たないパーティションは、デフォルトの phaseProvider.xml ファイルを使用します。このトピックに適用されるフェーズプロバイダファイル部分は、CoreServices フェーズ内にネストされる portlet フェーズに含まれます。

```
<!-- All services come on line  -->
<Phase name="CoreServices" enabled="true"
   class="com.webmethods.portal.system.init.impl.DefaultPhase">
      .
      .
   <PhaseInfo name="portlet" enabled="true"
      class="com.webmethods.portal.service.portlet.impl.PortletProvider"
      initFile="config:/adminPortlets.properties"
      initFileComponentsEnabled="false"
   />
      .
      .
/>
```

ここで、

次の表では、フェーズプロバイダファイルで設定可能な属性の期待値について説明します。

initFile	パーティションのフェーズプロバイダによって影響を受けるポートレットのリストが含まれるポートレットプロパティファイルを指定します。
initFileComponentsEnabled	パーティションポートレットプロパティファイルにリストされているポートレットが有効か無効かを指定します。 ■ false: リストされているポートレットは無効になります。 ■ true: 有効になるのはリストされているポートレットのみです。

上記の例の場合、ポートレットプロパティファイルは adminPortlets.properties で initFileComponentsEnabled 属性は false であり、これは、そのファイルに無効になるポートレットのリストが含まれることを意味します。

initFile および initFileComponentsEnabled 属性は必須ではありません。これらの属性を省略すると、すべてのポートレットがパーティション内で有効になります。

注:
パーティション用の新しいフェーズプロバイダファイルを作成するには、デフォルトの phaseProvider.xml ファイルのコピーを作成し、必要に応じて変更します。

ポートレットプロパティファイル

ポートレットプロパティファイル (*partition_name*Portlets.properties) は、特定のパーティションのフェーズプロバイダによって影響を受けるポートレットをリストします。開発またはテスト目的のために、個々のポートレットのフェーズプロバイダを上書きすることができます。

ポートレットプロパティファイルの構造は簡単です。次の例に示すように、各ポートレットを個別の行に配置して、後ろに = 記号を続けます。

```
wm_task_chart___taskchart=
wm_task_search___taskinboxsearchbar=
wm_task_search___taskinboxsearchresults=
wm_task_search___tlmsearchresults=
wm_task_search___tlmsearchquery=
```

ポートレットプロパティファイルは、すべてのポートレットが有効になっているパーティションには必要ありません。このような場合には、フェーズプロバイダファイルの initFile および initFileComponentsEnabled 属性を省略することができます。そうでない場合は、これらの属性でポートレットプロパティファイルの使用が管理されます。

フェーズプロバイダファイルの一時的な上書き

開発またはテスト目的のために、個々のファイルのフェーズプロバイダのデフォルト動作を一時的に上書きすることができます。これを行うには、次のように = 記号の前に + または - 記号を置きます。

- +=: フェーズプロバイダに関係なく、ポートレットを有効にします。
- -=: フェーズプロバイダに関係なく、ポートレットを無効にします。

たとえば、initFileComponentsEnabled 属性が false になっているとすると、ポートレットプロパティファイル内にリストされているすべてのポートレットが無効になります。上記リスト内の任意のポートレットを有効にするには、次のように = を += に変更します。

```
wm_task_search___processsearchbar+=
```

後で + 記号を削除すると、そのポートレットはリスト内の他のポートレットと同様に無効になります。

デフォルトのパーティション設定

デフォルトでは、すべてのクラスタノードが含まれる、default という名前のパーティションが 1 つあります。デフォルトパーティションの設定ファイルセットは、次のファイルで構成されています。

- clusterPartitions.xml: デフォルト状態で、このファイルはデフォルトパーティションのみを定義します。
- phaseProvider.xml: このフェーズプロバイダファイルは、デフォルトパーティションをサポートします。次の属性が設定されています。

```
initFile="config:/defaultPartitionPortlets.properties"
initFileComponentsEnabled="false"/>
```

- defaultPartitionPortlets.properties: このファイルは、デフォルトでは空です。

デフォルト設定では、デフォルトパーティション内のすべてのポートレットが有効になります。

例: クラスタパーティションの作成

設定ファイルを編集してクラスタパーティションを作成するには、以下の手順に従います。この手順は、デモを意図したものあり、現実的な例ではありません。この手順の前提として、1 つのクラスタが存在し、そこに次の 5 つのノードがあります。

default

node2

node3

node4

node5

この例では、クラスタに次の 3 つのパーティションを作成します。

default (すべてのポートレットが有効)

partition2 (指定したポートレットのグループが無効)

partition3 (指定したポートレットのグループが無効だが、一部は一時的に有効)

設定ファイルを編集してパーティションを作成するには

1. コマンドラインプロンプトで、次のコマンドを入力してサーバの bin ディレクトリに移動します。

```
cd Software AG_directory\MWS\bin
```

2. My webMethods Server データベースから clusterPartitions.xml ファイルを抽出するには、次のコマンドを入力します。

```
mws getconfig clusterPartitions.xml
```

3. My webMethods Server データベースから phaseProvider.xml ファイルを抽出するには、次のコマンドを入力します。

```
mws getconfig phaseProvider.xml
```

4. My webMethods Server データベースから defaultPartitionPortlets.properties ファイルを抽出するには、次のコマンドを入力します。

```
mws getconfig defaultPartitionPortlets.properties
```

5. ダウンロードした clusterPartitions.xml ファイルをテキストエディタで開きます。

   ダウンロードしたファイルはすべて次の場所にあります。

   Software AG_directory \MWS\server*serverName*\config

6. 235 ページの「クラスタパーティションファイル」の構文を使用して clusterPartitions.xml ファイルを編集し、partition2 および partition3 をクラスタに追加します。

   変更後の clusterPartitions.xml ファイルは次のようになります。

```
<?xml version="1.0" encoding="UTF-8"?>
<ClusterPartitions>
    <Partition name="default" frontEndUrl="http://my.company.server:8585">
        <Component name="default">
        </Component>
    </Partition>
    <Partition name="partition2">
        <Component name="node2">
        </Component>
        <Component name="node3">
        </Component>
    </Partition>
    <Partition name="partition3">
    </Partition>
</ClusterPartitions>
```

   このファイルを基に、次のように設定されます。

   - デフォルトパーティションにデフォルトノードが含まれる
   - partition2 に node2 および node3 が含まれる
   - partition3 に node4 および node5 が含まれる (ファイル内の最後のパーティションに、他のパーティションに特別に割り当てられていないすべてのノードが含まれる)

7. clusterPartitions.xml ファイルを保存して閉じます。

   この結果、237 ページの「デフォルトのパーティション設定」の説明に従ってデフォルトパーティションが作成され、すべてのポートレットが有効になります。

8. phaseProvider.xml のコピーを作成し、partition2PhaseProvider.xml という名前を付けます。

9. partition2PhaseProvider.xml ファイルをテキストエディタで開きます。

10. `CoreServices` フェーズ内にネストされている `portlet` フェーズで、ファイルが次のようになるように、`defaultPartitionPortlets.properties` を `partition2Portlets.properties` に変更します。

```
<PhaseInfo name="portlet" enabled="true"
    class="com.webmethods.portal.service.portlet.impl.PortletProvider"
    initFile="config:/partition2Portlets.properties"
    initFileComponentsEnabled="false"/>
```

これで、フェーズプロバイダが partition2Portlets.properties を指すようになります。initFileComponentsEnabled が false に設定されているため、このファイルにリストされているポートレットは無効になります。

11. partition2Portlets.properties ファイルを保存して閉じます。

12. defaultPartitionPortlets.properties のコピーを作成し、partition2Portlets.properties という名前を付けます。

13. partition2Portlets.properties ファイルをテキストエディタで開いて、partition2 で無効にするポートレットのリストを追加します。

    次に例を示します。

```
# These portlets will not be initialized.
wm_task_chart___taskchart=
wm_task_search___taskinboxsearchbar=
wm_task_search___taskinboxsearchresults=
wm_task_search___tlmsearchresults=
wm_task_search___tlmsearchquery=
```

    ユーザインタフェースでは、 これらのポートレットは次のように使用されます。

    - taskchart は [タスクチャート] を表示する
    - tlmsearchquery は [タスク一覧の管理] を表示する
    - 残りのポートレットは [受信トレイ] を表示する

14. partition2Portlets.properties ファイルを保存して閉じます。

    この結果、特定のポートレットが partition2 (node2 および node3) で無効になります。

    注:
    partition3 の手順では、フェーズプロバイダファイルを一時的に上書きする方法を示します。結果的に、partition2 の手順とほとんどが重複するため、partition3 の手順は簡単に説明します。

15. phaseProvider.xml のコピーを作成し、partition3PhaseProvider.xml という名前を付けます。

16. partition3PhaseProvider.xml で、`defaultPartitionPortlets.properties` を `partitionPortlets.properties` に変更します。

17. partition2Portlets.properties のコピーを作成し、partition3Portlets.properties という名前を付けます。

    この結果、partition2 と同じポートレットを無効にするように partition3 が設定されます。

18. partition3Portlets.properties で、最後の 2 つのポートレットの末尾にある = の前に + を置きます。

    次に例を示します。

    ```
    # These portlets will not be initialized.
    wm_task_chart___taskchart=
    wm_task_search___taskinboxsearchbar=
    wm_task_search___taskinboxsearchresults=
    wm_task_search___tlmsearchresults+=
    wm_task_search___tlmsearchquery+=
    ```

    注:
    この例の場合、ポートレットの選択は任意です。

    この結果、partition3 (node4 および node5) では、partition2 と同じポートレットが無効になりますが、ポートレットのサブセットは有効になります。

19. 新しい変更済みのファイルを My webMethods Server データベースに展開するには、ファイルごとに次のコマンドを入力します。

    ```
    mws putconfig fileName
    ```

20. *serverName*\config ディレクトリからこれらのファイルをすべて削除します。

    このファイルを削除しないと、クラスタは引き続きローカルバージョンの設定ファイルを使用します。

21. 次のコマンドを使用して、クラスタ内の各ノードを再起動します。

    ```
    mws -  serverName restart
    ```

    設定ファイルに加えた変更は、再起動するまで適用されません。

コンポーネントが無効になっているかどうかの確認

238 ページの「例: クラスタパーティションの作成」で説明したシナリオで、ポートレットが無効になっているかどうかを確認するには、次の一般的な操作を実行します。

1. clusterPartitions.xml ファイルで、各パーティションに別々の `frontEndUrl` 属性を割り当てます。

2. デフォルトの partition2 のフロントエンド URL を参照し、My webMethods Administrator としてログインします。

3. ナビゲーションパネルで、[タスク一覧の管理] をクリックします。

   この選択は次のポートレットに対応しています。

   ```
   wm_task_search___tlmsearchresults=
   wm_task_search___tlmsearchquery=
   ```

[タスク一覧の管理] がエラーなしで表示されます。

4. partition2 のフロントエンド URL を参照し、My webMethods Administrator としてログインします。

5. ナビゲーションパネルで、[タスク一覧の管理] をクリックします。

ポートレットは無効になっているため、エラーが表示されます。

My webMethods Server ユーザインタフェースでのパーティションの表示

パーティションを作成または変更するには、234 ページの「パーティションの作成と変更」で説明した設定ファイルを使用する必要があります。ただし、既存パーティションに関する情報は、My webMethods Server ユーザインタフェースで表示できます。

パーティションに関する情報を表示するには

1. 以下のいずれかの手順に従って、正しいページに移動します。

   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [My webMethods] > [クラスタ設定] > [クラスタパーティション]
   - システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [クラスタ管理] > [クラスタパーティション]

   次の表に、[クラスタパーティション] ページで表示される情報を示します。

列名	目的
[名前]	クラスタパーティションファイルに定義されているパーティション名。パーティション名をクリックすると、そのパーティションの情報を含むページが表示されます。
[フロントエンド URL]	パーティションで使用されるフロントエンド URL。
[ノード]	パーティション内のノードの数。
[パーティションルール]	フェーズプロバイダファイルによってパーティションに適用されるルールであり、コンポーネントが [**有効**] であるか [**無効**] であるかを決定します。
[管理されたコンポーネント]	ポートレットプロパティファイルによって管理されている、パーティション内のコンポーネント (Web アプリケーションおよびポートレット) の数。管理されたコンポーネントの状態は、フェーズプロバイダファイルで設定されているパーティションルールとは異なる可能性があります。個々のコンポーネントの状態については、243 ページの「パーティションページの [コンポーネント] タブ」を参照してください。

列名	目的
[詳細]	: パーティションに関する具体的な情報が含まれるページへのリンク。

2. 特定のパーティションに関する情報を検索するには、[**名前**] 列または をクリックしてください。

パーティションページの [ノード] タブ

パーティションページの [**ノード**] タブには、そのパーティションに含まれるノードの情報が含まれています。このタブの情報は、クラスタパーティションファイルに定義されているノードを反映します。

次の表に、My webMethods Server が [ノード] タブに表示する情報を示します。

列	目的
[名前]	クラスタパーティションファイルに定義されている、ノードに割り当てられた名前。
[ホスト]	ノードが配置されているホストマシンの名前。
[HTTP ポート]	ノードで使用される HTTP ポート。
[HTTPS ポート]	ノードの HTTPS リスナーで使用されるポート。値 0 は、HTTPS リスナーが無効になっていることを示します。

パーティションページの [コンポーネント] タブ

パーティションページの [**コンポーネント**] タブには、そのパーティションに含まれる Web アプリケーションおよびポートレットの情報が含まれています。このタブの情報は、パーティションのフェーズプロバイダファイルおよびパーティションプロパティファイルで作成した設定を反映します。

コンポーネントのリスト方法に合わせて、2 つのツリー表示があります。

- カテゴリ - カテゴリ別にコンポーネントをリストします。
- ポートレット - カテゴリに関係なく、コンポーネントを順にリストします。

2 つのビューの間を切り替えるには、[**カテゴリツリー表示**] オプションまたは [**ポートレットツリー表示**] オプションを選択します。

次の表は、My webMethods Server で表示される使用可能なコンポーネントの詳細について説明しています。

列	目的
[コンポーネント]	コンポーネントのツリー表示。ノードを展開すると、Web アプリケーションに含まれるポートレットが表示されます。
[カテゴリ]	(ポートレットツリー表示のみ) コンポーネントが属するカテゴリの名前。「管理」や「コミュニケーション」などです。
[名前]	My webMethods Server にインストールされているコンポーネントの名前。この名前は、[**コンポーネント**] 列で使用されている説明的な名前とは異なります。
[状態]	コンポーネントの状態。状態アイコンの意味は以下のとおりです。
	コンポーネントはこのパーティションで有効になっています。
	コンポーネントはこのパーティションで無効になっています。

III システム管理者の機能

14 属性プロバイダ 247

15 セキュリティの管理 257

16 分析、レポート、およびトラブルシューティング 309

17 My webMethods Server の設定 329

18 My webMethods Server コンテンツの管理 345

19 ユーザインタフェースの管理 353

20 共通ディレクトリサービス API の使用 375

21 My webMethods Server からの Mobile 通知の送信 381

14 属性プロバイダ

- 属性プロバイダとは 248
- 属性プロバイダの使用 249
- プリンシパル属性プロバイダの表示の管理 254

属性プロバイダとは

My webMethods Server ユーザに属する属性の提供および使用については、さまざまな方法があります。各ユーザの基本となる属性には、My webMethods Server 属性 (ホームページ、スキン、およびページあたりの表示) があります。また、以下のプリンシパル属性プロバイダがあります。

次の表に、使用可能な属性プロバイダとその機能を示します。

属性プロバイダ	説明
コア属性	ユーザ ID や電子メールアドレスなどのコア属性のセット。ユーザがシステムディレクトリサービスに属する場合、一部のフィールドは編集可能です。ユーザが外部ディレクトリサービスに属する場合は、すべてのフィールドが読み取り専用です。249 ページの「コア属性の属性プロバイダ」を参照してください。
[ユーザプロファイル]	ユーザがどのディレクトリサービスのメンバーであるかにかかわりなく保守できるユーザ属性の豊富なセット。属性は、My webMethods Server データベースに格納されます。一度各ユーザに設定すると、ユーザプロファイル属性は My webMethods Server 内でのグローバルなワイヤリングに使用できます。250 ページの「ユーザプロファイル属性プロバイダ」を参照してください。
[LDAP]	外部ディレクトリサービスの属性のセット。特定のディレクトリサービスから公開する属性を指定できます。250 ページの「LDAP 属性プロバイダ」を参照してください。
[データベース]	外部データベースディレクトリサービスの属性のセット。特定のディレクトリサービスから公開する属性を指定できます。252 ページの「データベース属性プロバイダ」を参照してください。
通知	ユーザが My webMethods Server からの通知を受信できる電子メールなどのアドレスのセット。253 ページの「通知属性プロバイダ」を参照してください。
動的	ユーザがメンバーとなっている役割に応じて値が変化する属性のセット。253 ページの「動的属性プロバイダ」を参照してください。

プリンシパル属性プロバイダは、公開する属性をポートレットのワイヤリングとして使用可能にできるため便利です。たとえば、ユーザがページを表示する際に、あるポートレットが郵便番号を使用して特定の情報を表示するとします。郵便番号がプリンシパル属性プロバイダからワイヤリングによって提供された場合、郵便番号属性がディレクトリサービス内で変更されると、ポートレットは変更された属性値を使用します。

プリンシパル属性プロバイダは、デフォルトでは無効です。有効にするには、254 ページの「プリンシパル属性プロバイダの表示の管理」で説明するように、[プリンシパルプロファイル管理] ページを使用します。

属性プロバイダの使用

ユーザ、グループまたは役割の [プロファイル] ページには、各種属性セットと、グループまたは役割のメンバーシップが表示されます。ユーザまたはグループが属するディレクトリサービスによって、一部の属性の編集や、グローバルワイヤリングでの使用目的で公開可能です。

[プロファイル] ページの例を表示するには、システム管理者として、標準リンクで [**マイプロファイル**] をクリックします。表示されるページは、SysAdmin の [プロファイル] ページです。

任意のユーザ、グループまたは役割の [プロファイル] ページを表示するには、以下の検索手順に従います。

ユーザ、グループまたは役割の [プロファイル] ページを検索するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [*User_type* の管理] > [検索]

   User_type は、[**ユーザ**]、[**グループ**]、または [**役割**] です。

2. 編集するユーザを検索します。詳細については、146 ページの「既存のユーザ、グループ、または役割の検索」を参照してください。

3. 検索結果で、ユーザ、グループ、または役割の行にある任意のリンクをクリックするか、 をクリックします。

   ユーザ、グループ、または役割の [プロファイル] ページが表示されます。

コア属性の属性プロバイダ

コア属性の属性プロバイダには、ユーザ ID や電子メールアドレスなどの属性セットが含まれます。ディレクトリサービスメンバーシップによっては、一部のフィールドが編集可能です。

[ユーザ情報] パネル

個々のユーザの場合、コア属性の属性プロバイダの内容は、[プロファイル] ページの [**ユーザ情報**] パネルに表示されます。システムディレクトリサービス内のユーザの場合、電子メールアドレスなどの一部のフィールドは編集可能です。このパネルの情報を編集するには、156 ページの「ユーザ情報の編集」を参照してください。

外部ディレクトリサービスのメンバーの場合、この情報を編集できませんが、405 ページの「プリンシパル属性からポートレットプロパティへのワイヤリング」で説明するグローバルワイヤリング機能では編集可能です。

[グループ情報] パネル

グループの場合、コア属性の属性プロバイダの内容は、[プロファイル] ページの [**グループ情報**] パネルに表示されます。システムディレクトリサービス内のグループの場合、電子メールアドレ

スなどの一部のフィールドは編集可能です。このパネルの情報を編集するには、168 ページの「グループ情報の編集」を参照してください。

外部ディレクトリサービスのグループの場合、この情報を編集できませんが、405 ページの「プリンシパル属性からポートレットプロパティへのワイヤリング」で説明するグローバルワイヤリング機能では編集可能です。

[役割情報] パネル

役割の場合、コア属性の属性プロバイダの内容は、[プロファイル] ページの [**役割情報**] パネルに表示されます。フィールドは編集できませんが、情報は 405 ページの「プリンシパル属性からポートレットプロパティへのワイヤリング」で説明するグローバルワイヤリング機能で使用可能です。

ユーザ設定属性プロバイダ

ユーザ設定は、My webMethods Server のすべてのユーザの基本属性です。My webMethods Server の管理者、システム管理者、または個々のユーザ (権限を付与されている場合) は、これらの基本属性を編集できます。ユーザ設定を編集するには、156 ページの「ユーザ情報の編集」を参照してください。

ユーザプロファイル属性プロバイダ

ユーザプロファイル属性プロバイダには、ユーザがどのディレクトリサービスのメンバーであるかに関係なく保守できるユーザ属性の豊富なセットがあります。属性は、My webMethods Server データベースに格納されます。各ユーザに対して一度設定すると、ユーザプロファイル属性は 405 ページの「プリンシパル属性からポートレットプロパティへのワイヤリング」で説明するグローバルワイヤリング機能に使用できます。

My webMethods Server の管理者、システム管理者、または個々のユーザ (権限を付与されている場合) は、ユーザプロファイル属性を編集できます。ユーザごとに個別にユーザプロファイル属性を入力するか、ユーザによる入力を許可する必要があります。必要に応じてすべての属性またはその一部を使用できます。ユーザプロファイル属性を編集するには、156 ページの「ユーザ情報の編集」を参照してください。

LDAP 属性プロバイダ

LDAP 属性プロバイダには、ユーザまたはグループが属する外部ディレクトリサービスの、指定した属性セットが表示されます。[**LDAP 属性**] パネルに表示される属性は編集できませんが、405 ページの「プリンシパル属性からポートレットプロパティへのワイヤリング」で説明するグローバルワイヤリング機能で使用可能です。

注:
LDAP 属性プロバイダがデフォルトで有効化されていない場合は、254 ページの「プリンシパル属性プロバイダの表示の管理」を参照してください。

LDAP属性プロバイダは、システムディレクトリサービスのユーザまたはグループには適用されません。類似する属性が、250 ページの「ユーザプロファイル属性プロバイダ」で説明したユーザプロファイル属性プロバイダに含まれています。

LDAP 属性プロバイダの表示

[**LDAP 属性**] パネルの [プロファイル] ページの内容は変更できませんが、必要であれば表示できます。ユーザまたはグループの [プロファイル] ページを検索するには、146 ページの「既存のユーザ、グループ、または役割の検索」を参照してください。

外部ディレクトリサービスの LDAP 属性の公開

LDAP 属性プロバイダには、外部ディレクトリサービスから公開されているユーザ属性が表示されます。選択した属性を公開し、グローバルワイヤリング機能で使用できます。

外部ディレクトリサービスの LDAP 属性を公開するには

1. システム管理者として次のように選択します。[**フォルダ**] > [システム] > [サービス]* > [ディレクトリ] > [プリンシパル属性プロバイダ] > [*Provider_type*]

   [*Provider_type*] は、[**ユーザプリンシパル属性プロバイダ**] または [**グループプリンシパル属性プロバイダ**] です。

   * [**次へ**] を何回かクリックしないと、表に表示される [**サービス**] を確認できないことがあります。

2. [LDAP 属性] で、 をクリックして、[プロパティ] を選択します。

   [LDAP 属性] ページの [プロパティ] が表示されます。

3. [**LDAP 属性名**] で、[**追加**] をクリックします。

4. 外部ディレクトリサービスで使用されている属性名を正確に入力し、[**OK**] をクリックします。

   たとえば、特定のディレクトリサービスでの電子メールの属性名は、「`mail`」などになります。

5. [**LDAP 属性タイトル**] で、[**追加**] をクリックします。

6. My webMethods Server 内で使用する属性の表示名を入力し、[**OK**] をクリックします。

   たとえば、`mail` 属性の場合、「`Email`」という表示名を入力できます。

7. 複数の LDAP 属性がある場合は、[**LDAP 属性名**] リストと [**LDAP 属性タイトル**] リストで順序が同じになるように注意してください。

   属性とタイトルがリストに表示される順序によって、[プロファイル] ページの [**LDAP 属性**] パネルに表示される順序が決まります。

8. [**適用**] をクリックします。

データベース属性プロバイダ

データベース属性プロバイダには、ユーザまたはグループが属する外部データベースディレクトリサービスの、指定した属性セットが表示されます。[**データベース属性**] パネルに表示される属性は編集できませんが、405 ページの「プリンシパル属性からポートレットプロパティへのワイヤリング」で説明するグローバルワイヤリング機能で使用可能です。

注:
データベース属性プロバイダがデフォルトで有効化されていない場合は、254 ページの「プリンシパル属性プロバイダの表示の管理」を参照してください。

LDAP属性プロバイダは、システムディレクトリサービスのユーザまたはグループには適用されません。類似する属性が、250 ページの「ユーザプロファイル属性プロバイダ」で説明したユーザプロファイル属性プロバイダに含まれています。

データベース属性プロバイダの表示

[**データベース属性**] パネルの [プロファイル] ページの内容は変更できませんが、必要であれば表示できます。ユーザまたはグループの [プロファイル] ページを検索するには、146 ページの「既存のユーザ、グループ、または役割の検索」を参照してください。

外部ディレクトリサービスのデータベース属性の公開

データベース属性プロバイダには、外部データベースディレクトリサービスから公開されているユーザまたはグループ属性が表示されます。選択した属性を公開し、グローバルワイヤリング機能で使用できます。

外部データベースディレクトリサービスのデータベース属性を公開するには

1. システム管理者として次のように選択します。[**フォルダ**] > [システム] > [サービス]* > [ディレクトリ] > [プリンシパル属性プロバイダ] > [*Provider_type*]

   [*Provider_type*] は、[**ユーザプリンシパル属性プロバイダ**] または [**グループプリンシパル属性プロバイダ**] です。

   * [**次へ**] を何回かクリックしないと、表に表示される [**サービス**] を確認できないことがあります。

2. [データベース属性] で、 をクリックして、[プロパティ] をクリックします。

   [データベース属性] ページの [プロパティ] が表示されます。

3. [**属性名**] で、[**追加**] をクリックします。

4. 外部データベースディレクトリサービスで使用されている属性名を正確に入力し、[**OK**] をクリックします。

重要:
ここで使用する属性は、データベースディレクトリサービスの [ID に基づくユーザ検索のクエリー] 属性によって返されます。126 ページの「外部データベースディレクトリサービスの設定」を参照してください。

たとえば、特定のディレクトリサービスでの郵便番号の属性名は、「`zipcode`」などになります。

5. [**属性タイトル**] で、[**追加**] をクリックします。

6. My webMethods Server 内で使用する属性の表示名を入力し、[**OK**] をクリックします。

   たとえば、`zipcode` 属性の場合、「`Zip Code`」という表示名を入力できます。

7. 複数のデータベース属性がある場合は、[**属性名**] リストと [**属性タイトル**] リストで順序が同じになるように注意してください。

   属性とタイトルがリストに表示される順序によって、[プロファイル] ページの [**データベース属性**] パネルに表示される順序が決まります。

8. [**適用**] をクリックします。

通知属性プロバイダ

通知属性プロバイダを使用すると、ユーザが通知を受信するためのさまざまなアドレスを指定できます。My webMethods Server の管理者、システム管理者、または個々のユーザ (権限を付与されている場合) は、これらの基本属性を編集できます。ユーザ設定を編集するには、156 ページの「ユーザ情報の編集」を参照してください。

注:
通知属性プロバイダは、デフォルトでは無効です。254 ページの「プリンシパル属性プロバイダの表示の管理」を参照してください。

ユーザ設定は、My webMethods Server のすべてのユーザの基本属性です。My webMethods Server の管理者、システム管理者、または個々のユーザ (権限を付与されている場合) は、これらを編集できます。ユーザの通知を編集するには、156 ページの「ユーザ情報の編集」を参照してください。

動的属性プロバイダ

動的属性プロバイダを使用すると、役割の属性を指定できます。役割のメンバーであるユーザ、グループ、または役割は、その役割のすべての動的属性を所持します。ユーザが複数の役割のメンバーで、複数の役割に同じキーを持つ属性がある場合、優先する役割を指定できます。また、役割が提供する属性値に優先する属性値をユーザに割り当てることができます。動的属性は、156 ページの「ユーザ情報の編集」で説明するグローバルワイヤリング機能に使用できます。

注:

動的属性プロバイダは、役割に対してのみ有効です。ユーザおよびグループには、メンバーになっている役割に基づく動的属性があります。

動的属性に関して実行可能な機能の詳細については、200 ページの「役割に関連付ける動的属性の定義」を参照してください。

プリンシパル属性プロバイダの表示の管理

プロファイルページにデフォルトですべてのプリンシパル属性プロバイダが表示されていた場合、ページはプロバイダでいっぱいになり、読みにくくなる可能性があります。[プリンシパルプロファイル管理] ページで、[プロファイル] ページに表示するプリンシパル属性プロバイダと、それらの表示順序を選択できます。ユーザ、グループ、および役割のプリンシパル属性プロバイダを設定できます。

次の表に、使用可能なプリンシパル属性プロバイダのリストと、各リストが適用されるプロファイルページを示します。

リスト	[プロファイル] ページ
[ユーザ属性プロバイダ]	ユーザの [プロファイル] ページ。デフォルトでは、コア属性プロバイダが [**ユーザ情報**] パネルとして表示されます。その他のデフォルト属性プロバイダは、グループ、役割、ユーザ設定です。個々のユーザに適用可能な任意のプリンシパル属性プロバイダを追加できます。
[グループ属性プロバイダ]	グループの [プロファイル] ページ。デフォルトでは、コア属性プロバイダが [**グループ情報**] パネルとして表示されます。デフォルト属性プロバイダは、グループおよびグループメンバーです。グループに適用可能な任意のプリンシパル属性プロバイダを追加できます。
[役割属性プロバイダ]	役割の [プロファイル] ページ。デフォルトでは、コア属性プロバイダが [**役割情報**] パネルとして表示されます。その他のデフォルト属性プロバイダは、動的属性です。

[プリンシパルプロファイル管理] ページでは、以下のアクションを実行できます。

- プロファイルページにプリンシパル属性プロバイダを追加する
- プロファイルページでプリンシパル属性プロバイダの位置を並べ替える
- プロファイルページからプリンシパル属性プロバイダを削除する

プリンシパル属性プロバイダの追加

[プロファイル] ページにプリンシパル属性プロバイダを追加するには、以下の手順に従います。

[プロファイル] ページにプリンシパル属性プロバイダを追加するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [プリンシパルプロファイル管理]

2. [**ユーザ属性プロバイダ**]、[**グループ属性プロバイダ**]、または [**役割属性プロバイダ**] 領域で、[**追加**] をクリックします。

3. 追加するプリンシパル属性プロバイダを指定するには、プリンシパル属性プロバイダを [**選択された項目**] ボックスに移動し、[**選択**] をクリックします。

4. ページの下部で、[**適用**] をクリックします。

プリンシパル属性プロバイダの表示順の変更

[プロファイル] ページでプリンシパル属性プロバイダが表示される順序を変更するには、以下の手順に従います。

プリンシパル属性プロバイダの表示順を変更するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [プリンシパルプロファイル管理]

2. [**ユーザ属性プロバイダ**]、[**グループ属性プロバイダ**]、または [**役割属性プロバイダ**] 領域で、必要に応じて上下に移動して、プリンシパル属性プロバイダの順序を変更します。

   リスト内の最初の属性プロバイダが、[プロファイル] ページでは左端に表示され、その後に 2 番目、3 番目の属性プロバイダが続きます。

3. ページの下部で、[**適用**] をクリックします。

プリンシパル属性プロバイダの削除

[プロファイル] ページからプリンシパル属性プロバイダを削除するには、以下の手順に従います。

[プロファイル] ページからプリンシパル属性プロバイダを削除するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザ管理] > [プリンシパルプロファイル管理]

2. [**ユーザ属性プロバイダ**]、[**グループ属性プロバイダ**]、または [**役割属性プロバイダ**] 領域で、リストからプリンシパル属性プロバイダを選択し、[**削除**] をクリックします。

3. ページの下部で、[**適用**] をクリックします。

15 セキュリティの管理

- My webMethods Server セキュリティについて 258
- 認証の管理 270
- ケルベロス認証の設定 274
- NTLM 認証の設定 277
- NTLMv2 認証の設定 278
- 外部設定クレデンシャルの設定 280
- My webMethods Server シングルサインオンの設定 282
- OAuth 2.0 認証の設定 289
- メモリからのセッションパスワードの消去 293
- メモリでのセッションパスワードの保持 293
- ユーザ名およびパスワードのオートコンプリートのオン/オフの切り替え 294
- 失敗したログイン試行回数の制御 294
- ログイン IP 範囲の制御 295
- グローバル環境変数のパスワードの暗号化 297
- コンテキストなりすましの許可 298
- パスワードの複雑度ポリシーの使用 299
- 応答ヘッダールールの使用 302

My webMethods Server セキュリティについて

My webMethods Server には、サーバ全体のセキュリティインフラストラクチャに役立つ機能が多数備わっています。セキュリティについて考える場合、常に認証 (Auth) と承認 (AZ) を分けて論じる必要があります。ほとんどの場合、認証と承認は関連していますが、この 2 つの概念は別個のものであり、2つが連係することで全体的なセキュリティソリューションに大きな効果をもたらします。

認証とは、コンピュータトランザクションの当事者が偽者ではないということの保証です。認証には通常、パスワード、証明書、PIN、または識別情報の確認に使用できるその他の情報を使用します。認証の目的は、単に当事者が当事者の申告どおり本人であることを確認することです。

承認とは、適切なアクセスコントロール情報を評価することにより、当事者が、特定のリソースに対して指定のタイプのアクセス権を許可されるかどうかを確認するプロセスです。通常、承認は認証のコンテキストにおいて行われます。当事者は、認証されると、さまざまなタイプのアクティビティを実行する権限を与えられます。

My webMethods Server には、認証と承認の両方のインフラストラクチャが組み込まれています。また、My webMethods Server は、既存のセキュリティインフラストラクチャを再利用して、認証と承認の両方に活用するために拡張できるよう設計されています。

My webMethods Server では、サーバ全体、またはフォルダ、ページ、ポートレット、リンク、ドキュメント、ファイル、カスタムオブジェクトを含む個々のサーバリソースに対して、認証と承認の両方を適用できます。

サーバの認証

My webMethods Server では、ユーザが本人であることを証明するさまざまな方法をサポートしています。これらのさまざまな方法は、認証スキームと呼ばれます。これらのスキームは、ユーザクレデンシャルを収集し、その信頼性を検証するための方法が異なるだけです。

フォーム認証

フォーム認証は、My webMethods Server のデフォルトの認証スキームです。この認証スキームでは、ユーザに対してフォームを提示し、そのフォームを POST する方法により、サーバに渡す必要なクレデンシャルを収集します (データはサーバの標準入力に渡されます)。認証に使用されるフォームまたはページは、簡単にカスタマイズできます。各種の条件に基づいてさまざまなフォームやページを提示することも容易です。たとえば、モバイル機器からサーバにアクセスするユーザと、ブラウザからサーバにアクセスするユーザに対して、異なるログイン方法を提供することができます。

Anonymous 認証

Anonymous 認証スキームは、ユーザにクレデンシャルを要求しない場合に使用されます。My webMethods Server は Anonymous 要求を受け付け、特別な Guest ユーザとしてサーバとのセッションを設定しますが、ユーザがクレデンシャルを要求されることはありません。Anonymous 認証スキームは、パブリックに公開される可能性があり機密情報が含まれていないサーバの非保護領域に使用されます。要求をセッションと Guest のユーザ ID に関連付けることで、管理者

は、Guest ユーザの権限を制御して、Anonymous アクセスの動作を拡張できます。Guest は、システムディレクトリサービスの一部としてインストールされるデフォルトのユーザの 1 つです。また、Anonymous ユーザのセッションアクティビティは、レポート要件として追跡することができます。

Basic 認証

Basic 認証は、伝送メカニズムとして HTTP を使用するプログラム向けの、最も互換性の高い本来の認証スキームの 1 つです。しかし、ユーザ名とパスワードを暗号化せずにサーバに送信するため、安全性が最も低いスキームでもあります。クレデンシャルは通常、HTTP ヘッダーパラメータとして渡されます。Basic 認証の場合、ユーザにはネイティブウィンドウシステムでレンダリングされるポップアップウィンドウが表示されます。たとえば、Windows で Basic 認証を使用すると、Windows のダイアログボックスが開き、ユーザにクレデンシャルの入力を求めてから、要求を受け入れます。

ケルベロス認証

ケルベロスは、保護されていないネットワーク上でクライアントと My webMethods Server が安全に通信できるようにする、ネットワーク認証プロトコルです。ケルベロス認証スキームは、クライアントと My webMethods Server の相互認証を提供し、対称暗号化と信頼できるサードパーティの KDC (key distribution center: キー配布センター) を使用して同一性を検証します。

Windows 認証と統合されたケルベロス認証を使用するように My webMethods Server を設定して、シングルサインオンを利用できます。この認証スキームを使用すると、既に Windows によって認証された有効な My webMethods Server ユーザは、追加のクレデンシャルを指定することなく My webMethods Server にアクセスできます。

ケルベロスと Windows 認証を使用してシングルサインオンを設定する場合は、274 ページの「ケルベロス認証の設定」を参照してください。

NTLM 認証

NTLM (Windows NT LAN Manager) は、各種 Microsoft ネットワークプロトコルの実装で使用され、NTLMSSP (NTLM Security Support Provider) でサポートされている認証プロトコルです。本来はセキュリティで保護された DCE/RPC の認証およびネゴシエーションに使用される NTLM ですが、統合されたシングルサインオンメカニズムとして Microsoft のシステム全体でも使用されます。Windows システム上では、NTLM がセットアップされ、My webMethods Server 用の認証スキームとして設定されている場合、既に Windows ドメインにログインしているユーザは、各サーバリソースについての再認証が不要です。NTLM または NTLM バージョン 2 (NTLMv2) 認証を選択できます。

NTLM 認証を使用するには、277 ページの「NTLM 認証の設定」で説明しているように、ドメインのプライマリドメインコントローラを明示的に指定する必要があります。

NTLMv2 を使用するには、IOPLEX Software から Jespa Java ソフトウェアライブラリを購入する必要があります。NTLMv2 の設定の詳細については、278 ページの「NTLMv2 認証の設定」を参照してください。

注:

webMethods NTLM 認証または NTLMv2 のいずれかを使用できますが、両方を同時に使用することはできません。

HTTP ヘッダー認証

サードパーティのセキュリティおよびアクセスコントロール製品 (Computer Associates、Oblix など) の外部 HTTP 認証クレデンシャルを受け入れるように、My webMethods Server を設定できます。これらのクレデンシャルは、プラットフォームおよび Web サーバによっては大文字と小文字が区別され、通常は sm_user または SM_USER などのヘッダーになります。

My webMethods Server 内で HTTP ヘッダー認証を設定し、セットアップすると、サーバではサードパーティの認証エンジンからのクレデンシャルが使用されます。通常、これらのサードパーティは、セキュリティエージェントを使用して、サーバに到達する前に要求を捕捉します。この要求におけるイベントの基本的な流れは次のとおりです。

1. ユーザがサーバリソースへの移動を試みます。

2. サーバに接続する前に、サードパーティのセキュリティエージェントで適切なクレデンシャルが確認されない場合、エージェントはクレデンシャルを収集するメカニズムにユーザをリダイレクトします。

3. ユーザは、クレデンシャルを提供すると、元のサーバリソースにリダイレクトされます。

4. サーバでは該当する HTTP ヘッダーを読み取り、ユーザを適切にマッピングします。

   サーバとサードパーティのセキュリティエージェントとのこのようなインタラクションを設定するには、以下の処理を行う必要があります。

5. My webMethods Server のインストール後、サーバを保護するようにサードパーティ製品を設定します。これには通常、サーバ URL を保護するポリシーを作成する必要があります。

6. サーバとサードパーティのセキュリティ製品が、同じディレクトリストアを参照するように設定されていることを確認します。ディレクトリサービスの詳細については、「133 ページの「外部データソースの管理」」を参照してください。

7. サーバが正しい HTTP ヘッダーを検索するように設定します。詳細については、280 ページの「外部設定クレデンシャルの設定」を参照してください。

   注:
   Computer Associates 社の SiteMinder の場合、SiteMinder でのログアウト URI の指定も必要です。SiteMinder Administrator アプレットで、logoutURI 属性を「/?method=logout」に変更します。

   重要:
   [HTTP ヘッダー認証管理] ページは、サードパーティのセキュリティプロバイダを使用している場合にのみ有効にします。ページを有効にすると、サーバはすべてのユーザが認証済みであるかのように動作します。

拡張済みおよび拡張可能な認証スキーム

My webMethods Server には、開発者が独自のカスタム認証スキームを提供する方法が用意されています。カスタム認証スキームを開発するには、ポートレットを作成し、正しいインタフェースを実装して、そのポートレットをサーバに登録します。一度作成すると、新規の認証スキームは、My webMethods Server の一部として提供されるその他の認証スキームと同様に、セキュリティインフラストラクチャに加わります。

My webMethods Server には、展開全体に適用されるデフォルトの認証スキームの概念があります。新たに設定されたサーバでは、デフォルトの認証スキームとしてフォーム認証を使用します。サーバは、ユーザにユーザ名およびパスワードの入力を求めるフォームを使用して、保護されたリソースに対する最初の要求の正当性を調べます。

サーバに対するデフォルトの認証スキームは、登録済みの認証スキームのいずれかにいつでも変更できます。詳細については、271 ページの「デフォルトの認証スキームの指定」を参照してください。

サーバリソースには認証スキームを設定でき、このスキームは、展開全体に対する設定より優先されます。完全に匿名アクセス可能な一連のページとポートレットがあり、それ以外のページとポートレットにはユーザクレデンシャルの提示が必要であるとします。これは、Anonymous 認証スキームを、認証を必要としないリソースと関連付けることで可能になります。サーバリソースの認証スキームの管理については、271 ページの「サーバリソースへの認証スキームの割り当て」を参照してください。

ログインおよびトップページの動作の拡張

ログインプロセスとイベントのフローを理解するには、システム管理者が、展開をどのように拡張して、カスタムログインページを動作させるかの例を分析することが役立ちます。以下の手順では、ソリューション作成のために、Anonymous アクセス権、フォームベースの認証、およびログインページの概念を利用しています。手順の一部では、開発者の知識が必要です。

1. ログインポートレットを持つページを設計します。ページの作成後は、ページの認証スキームを Anonymous に設定し、誰でもページにアクセスできて、ログインポートレットが提示されるようにします。

   オプションで、ページの他の部分に対するアクセス権を設定できるため、ログインページは、ユーザの識別情報によって外観が異なります。ログインページのパーソナライズに関するより広範な要件に対応するために、ルールに基づいてカスタムログインページを設定することも簡単です。

2. ページの認証スキームを Anonymous に設定した後、Guest ユーザがログインポートレットを表示できることを確認します。Guest ユーザの詳細については、114 ページの「ディレクトリサービスの管理」を参照してください。

   タイトルバーの削除、明示的な指示の追加、あるいはその他のビジネス要件の実装により、ページのルックアンドフィールを変更することもできます。

3. ログイン後のユーザのリダイレクト先を制御できます。ログインポートレットの [プロパティ] ページで、[ログインターゲット] プロパティをユーザがリダイレクトされるページに変更します。[ログインターゲット] は静的に指定するか、または、エイリアスを指定することもで

きます。「/user.current.start.page」のようなエイリアスを使用する場合、ユーザログインに関する情報に基づいて、異なるスタートページを制御するスタートページルールを設定することもできます。

要求が認証されなかった場合、適切なログインページに移動するようにリダイレクトすることも可能です。そのためには、ページに割り当てられた認証スキームの [URI のリダイレクト] プロパティを変更します。認証されていないユーザがページを要求した場合、そのユーザは指定したページにリダイレクトされます。ログインターゲットと同様に、リダイレクト URI は、静的でもエイリアスでもかまいません。

Integration Server で使用可能なカスタム認証者の作成

My webMethods Server のデータベースディレクトリサービスで使用されるカスタム認証者ポートレットを実装する場合、および共通ディレクトリサービスを使用するように Integration Server が設定されている場合は、追加の手順を実行して、ユーザが Integration Server にログインするための認証を有効にする必要があります。

カスタム認証者を Integration Server で使用可能にするには

1. すべての webMethods コンポーネントを停止します。

2. 任意の .zip エクストラクタを使用して、カスタム認証者の .pdp ファイルを開きます。

3. .pdp から /WEB-INF/lib/*custom_authenticator*.jar を解凍します (名前はカスタムコンポーネントの命名規則に応じて異なります)。

4. すべての Integration Server インスタンスでカスタム認証者を使用可能にするには、抽出された .jar ファイルを次の場所にコピーします。

   Integration Server_directory /lib/jars/

5. 特定の Integration Server インスタンスでカスタム認証者を使用可能にするには、抽出された .jar ファイルを次の場所にコピーします。

   Integration Server_directory /instances/*serverName*/lib/jars/

6. 共通ディレクトリサービスの設定ファイルを変更するには、以下の手順に従います。

   a. 次の .jar ファイルを探します。

      Software AG_directory \common\lib\wm-mws-library.jar

   b. wm-mws-library.jar ファイルから次の 2 つのファイルを解凍します。

      ```
      initPortlets.properties initXTypes.properties
      ```

   c. この2つのファイルをテキストエディタで開き、次の構文を使用してカスタム認証者ポートレットの名前をファイルの最後の行として追加します。

custom_authenticator=

これらのファイルでは例として別のポートレット名を使用します。

d. 各ファイルを保存して、両方のファイルを wm-mws-library.jar ファイルに再パッケージ化します。

7. すべての webMethods コンポーネントを再起動します。

SAML (Security Assertion Markup Language)

My webMethods Server では、セキュリティ情報交換の XML ベースのフレームワークである SAML を介したシングルサインオンをサポートしています。SAML を使用すると、ターゲットコンピュータ上のアプリケーションは、ソースコンピュータのアサーションに基づいてアクセス権を付与します。

My webMethods Server は呼び出しプログラム、すなわちセキュリティプロバイダにすることも、ターゲット Web アプリケーションのユーザサインオンを認証するように設定することもできます。詳細については、282 ページの「My webMethods Server シングルサインオンの設定」を参照してください。

My webMethods Server は、サードパーティ ID プロバイダ (IDP) によって開始されるシングルサインオンをサポートします。My webMethods Server (サービスプロバイダ) とサードパーティ IDP の間で認証および認証データを交換するための SAML 2.0 を使用するように My webMethods Server を設定できます。この場合、My webMethods Server が SAML コンシューマになり、サードパーティ ID プロバイダ (IDP) が SAML 認証局になります。サードパーティ IDP によって開始されるシングルサインオンの設定の詳細については、「286 ページの「サードパーティ ID プロバイダを使用するためのシングルサインオンの設定」」を参照してください。

OAuth 2.0 認証

My webMethods Server で OAuth 2.0 認証フローを設定し、Google や、Twitter、Salesforce などの外部の ID プロバイダからのクレデンシャルで My webMethods にログインできます。この認証フローでは、My webMethods Server は OAuth 2.0 プロトコルを OpenID Connect ID レイヤと共に、認証および承認の方法として使用します。

OAuth 2.0 認証フローを使用してログインすると、My webMethods Server はユーザを、ユーザのクレデンシャルを指定する外部の ID プロバイダの承認エンドポイント (たとえば Google 認証サーバ) にリダイレクトします。認証後、ID プロバイダは、ユーザを My webMethods Server エンドポイントにリダイレクトします。My webMethods Server はプロバイダから ID とアクセストークンを取得し、ID トークンを検証し、UserInfo クレームのリストを要求します。プロバイダによって返されるクレームに基づいて、My webMethods Server は内部ユーザアカウントを登録し、アカウントをログインセッションに関連付けます。

My webMethods Server で OAuth 2.0 と OpenID Connect 認証フローを設定するには、289 ページの「OAuth 2.0 認証の設定」を参照してください。

サーバ認証

ユーザ要求がサーバによって認証されると、通常は、要求を作成したユーザがそのリソースに対して操作を行うための必要な特権を持っていることを確認するなんらかの承認チェックを行う必要があります。My webMethods Server では、承認チェックを行う最も一般的な方法は、ACL (Access Control List : アクセスコントロールリスト) を評価することです。ACL は、ページ、ポートレットなど、あらゆる種類のサーバリソースと関連付けることができます。

動詞やメカニズム (ビジネスロジックのグループ化) のようなより高度な概念もサーバリソースであり、したがってこれらも ACL 評価モデルに加わります。この機能を使用すると、開発者はサーバの機能をプログラムでロックダウンできます。

My webMethods Server の承認エンジンを理解するため、ACL の構成を見ていきます。ACL は、ACE (Access Control Entries : アクセスコントロールエントリ) のリストです。ACE は、プリンシパルと呼ばれるエレメントと権限セットと呼ばれる要素を含む単純な構造です。

プリンシパルとは、ユーザ、グループまたは役割です。次の表にプリンシパルの例を示します。

プリンシパル	例
[ユーザ]	Myles Perkins
[グループ]	Perkins 家のメンバー
[役割]	Myles 氏を指定する役割定義 (例:「役職」属性の値が「プロダクトマネージャ」に設定されているユーザ)

権限セットとは、サーバリソースに対して実行できるアクションのグループです。権限セットの例は、「読み取り機能を許可する」です。権限セット自体は、機能と設定の 2 つの個別の部分に分けられます。さまざまなタイプのサーバリソースには、それぞれに関連付けられた各種機能があります。たとえば、ページには「ページにポートレットを追加する」などの機能があるのに対して、フォルダには「サブフォルダを作成する」や「フォルダ内のアイテムを読み取る」などの機能があります。

権限セットのもう一方の部分である設定には、「委任」、「拒否」、「許可」、または「なし」の 4 つの値を設定できます。権限セットを構成する各機能は、1 つの設定値を持ちます。権限セットは、機能と設定の多数の組み合わせで構成されます。次に権限セットの例を示します。

```
DENY + create sub folder
```

```
GRANT + read
```

この特定の権限セットは、機能と設定の２つの組み合わせで構成されています。フォルダリソースに関連付けられた場合、この権限セットは、プリンシパルに対して、サブフォルダを作成する権限の付与は拒否しますが、実際にフォルダを読み取る権限は付与します。

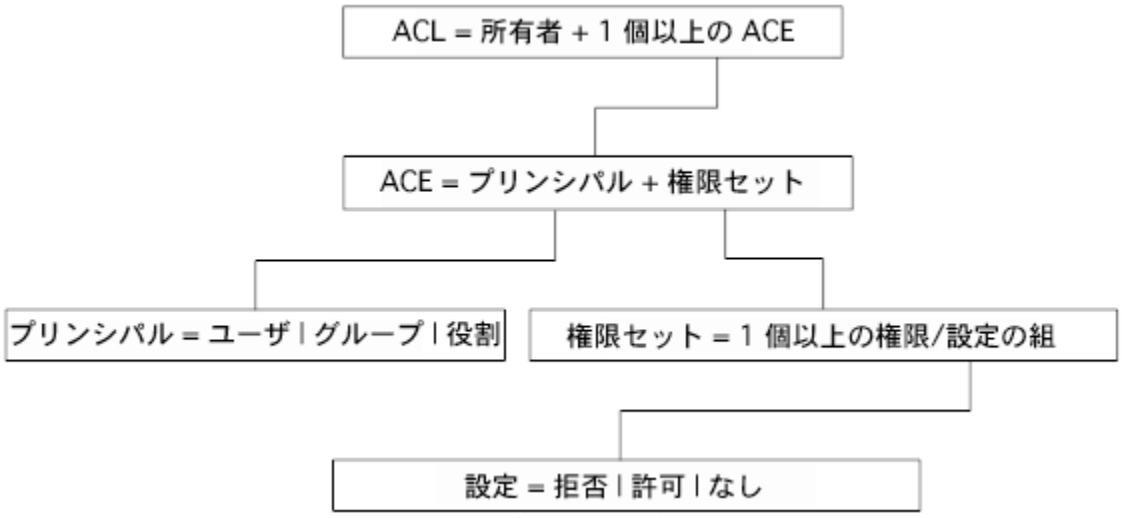
次の表では、各設定の効果について説明しています。

設定値	効果
DENY	機能を実行するためのアクセスを拒否します。

設定値	効果
GRANT	機能を実行するためのアクセスを明示的に許可します。
DELEGATE	明示的にアクセスを許可し、この機能を別のプリンシパルに割り当てる権限を与えます。
NONE	明示的な設定は行いません。このサーバリソースに対する承認は、別のソースから決定されます。

次の図は、この節で説明されている関係を示しています。

ACL の構造



たとえば、「エンジニアリング」ページに対する Brian 氏と営業グループの読み取りアクセスを拒否するには、次の設定を行います。

- サーバリソースは「エンジニアリング」ページです。
- 「エンジニアリング」ページには関連付けられた ACL があり、それには次に示すような ACE が含まれています。

プリンシパル	権限セット
Brian	読み取り拒否
営業グループ	読み取り拒否

サーバリソースに対する権限の制御

サーバリソースに対するアクセス権の変更が承認されている場合、My webMethods Server の [権限] ポートレットを使用して、サーバリソースにアクセスコントロールを割り当てます。たとえば、264 ページの「サーバ認証」で説明した例の「エンジニアリング」フォルダの所有者である場合、[権限] ポートレットのウィザードにより、1 つ以上のプリンシパルと権限セットを選択

し、それらをそのフォルダに関連付けることができます。権限の制御の詳細については、175 ページの「権限の管理」を参照してください。

新規に作成されたオブジェクトごとに、明示的にアクセスを許可および拒否する必要はありません。少し注意して分類を行うと、潜在的な保守の負担を最小限に抑えることができます。My webMethods Server では、サーバオブジェクトの作成時に、オブジェクトに対するアクセス権の静的伝播という方法を使用します。つまり、サーバリソースは、作成時にその親リソースからアクセス権を受け取ります。その後、親のアクセス権に変更が加えられても、これらのアクセス権は子オブジェクトでは動的に更新されません。ただし、[権限] ポートレットを使用すると、親オブジェクトが子オブジェクトに対してアクセス権を明示的に適用するようにできます。

アクセス権に関する静的伝播と親子のインタラクションを説明するために、「エンジニアリング」フォルダの例に戻ります。この例では、「エンジニアリング」フォルダに「BRIAN 拒否」の権限があります。「エンジニアリング」フォルダの所有者として、ユーザは「機密プロジェクト」というサブフォルダを作成します。静的伝播のために、新規の「機密プロジェクト」フォルダには作成時に「BRIAN 拒否」権限が付与されます。所有者が、「エンジニアリング」フォルダに戻って、Brian にアクセス権を許可するように権限を変更しても、Brian には「機密プロジェクト」サブフォルダに対するアクセス権は付与されません。

承認の決定

これまで、承認の決定を行うという概念について説明してきました。次に、実行時に実際にアクセス権がどのように決定されるかを見ていきます。サーバリソースが要求されると、サーバはそのリソースに関連付けられた ACL を、現在の要求が生成されたコンテキストに対して評価します。ユーザがページに対するアクセス権を要求する場合、ユーザの要求を受け付けるかどうかを決定するために、そのページの ACL が評価されます。

承認の決定には、発生する可能性のある競合の大部分を処理する簡単なルールがいくつかあります。

- 拒否は常に許可に優先します (セキュリティの処理では用心しすぎということはありません)。
- ユーザは常にグループや役割に優先します。

これらのルールと、競合解決のためにルールがどのように適用されるかを説明するために、「エンジニアリング」フォルダの例に戻ります。次の例では、「エンジニアリング」フォルダに関連付けられた ACL に 3 つの ACE エントリがあります。

```
BRIAN + DENY READ
```

```
MARKETING GROUP + DENY READ
```

```
BRIAN + GRANT READ
```

Brian は、営業グループのメンバーであっても (たとえメンバーでなかったとしても)、「エンジニアリング」フォルダへのアクセスを拒否されます。ユーザベースの ACE は、グループベースの ACE より優先されるため、営業グループの ACE には影響がありません。続いて、BRIAN のアクセスに対する拒否と許可の競合は、拒否が常に優先されるため、アクセスを拒否することにより解決されます。

リスト、ページ、子オブジェクト、および検索

前述のように、プリンシパルはユーザ、グループ、または役割のいずれかです。プリンシパルの情報は、ディレクトリサービスから取得されます。My webMethods Server には、114 ページの「ディレクトリサービスの管理」 で説明しているように、システムディレクトリサービスが組み込まれているほか、外部ディレクトリサービスに接続する機能があります。これら外部ディレクトリサービスの例としては、Active Directory、LDAP サーバ、ADAM、および RDBMS があります。My webMethods Server 承認決定のためのグループおよび役割情報は、ユーザがサーバにログインしたときに確認されます。ユーザのグループメンバーシップがアクティブなセッション中に変更された場合、ユーザがログアウトし再びログインするまで、変更はサーバに反映されません。ユーザ、グループ、役割、およびディレクトリサービスの詳細については、「153 ページの「ユーザとグループの管理」」を参照してください。

セキュリティ領域

My webMethods Server には、セキュリティモデルを強化するためのセキュリティ領域という機能が用意されています。セキュリティ領域とは、同じ ACL を共有するサーバリソースの集合です。セキュリティ領域を使用すると、多数のサーバリソースに対する権限を容易に管理できます。システム管理者は、リソースをセキュリティ領域に直接追加することで、アクセスコントロールのためのプリンシパル情報をその領域に追加できます。

サーバリソースが多数あり、アクセスレベルが少ししかない場合、セキュリティ領域は非常に役に立ちます。たとえば、多数のポートレット、ページ、および分類領域を持つ大きな顧客向けサーバがあるとします。しかし、このサーバには、管理する必要のあるアクセス権のレベルが Gold、Silver、および Bronze の 3 つしかないとします。各レベルは、適切なページ、ポートレット、およびその中の分類エレメントを持つセキュリティ領域で表されており、システム管理者は、新規顧客を該当するセキュリティ領域に追加し、顧客に正しいレベルのアクセス権を付与するだけです。同様に、顧客をあるレベルから別のレベルに変更することも、1 ステップで行える簡単な操作です。

適切な展開で使用される場合、セキュリティ領域は、管理の負担を最小限に抑えるだけでなく、セキュリティモデルをサポートするために必要な基礎レコードの数を大幅に削減することで、さらに大きな効果をもたらします。たとえば、サーバに 500,000 のサーバリソースがあり、全員同じアクセス権を持つ 50 人のユーザの権限を管理しているとします。

- ACL による権限の管理には、My webMethods Server データベース内に 2,500 万レコードが必要です。
- セキュリティ領域による権限の管理では、1 つのセキュリティ領域と 50 人のメンバーを持つ 1 つの役割を使用するため、My webMethods Server データベース内に合計 3 レコードが必要です。

サーバリソースをセキュリティ領域に追加した場合、セキュリティ領域のアクセスコントロールは、個々の ACL およびそのリソースの認証スキームよりも優先されることに注意してください。

セキュリティ領域の管理の詳細については、182 ページの「セキュリティ領域の使用」を参照してください。

サーバ動詞およびアクセスコントロール

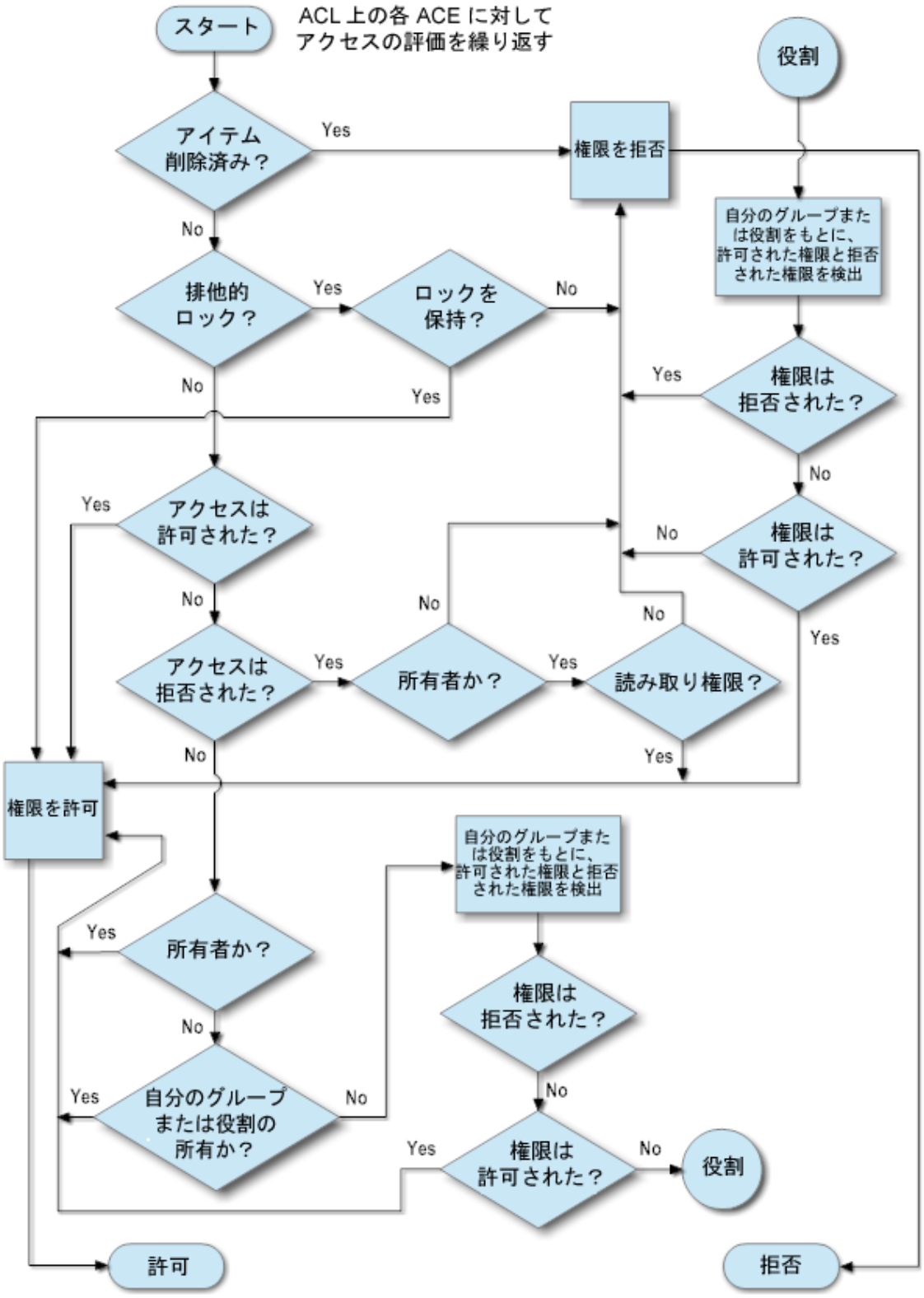
サーバ動詞とは、パブリッシュ、削除、更新、サブスクライブ、権限の設定などの操作を指すもので、My webMethods Server API を介して使用できます。前述したように、サーバ動詞は、サーバのセキュリティモデルにも加えることのできるサーバリソースです。このように、サーバの機能に対する細分化されたアクセス権を、Administrative Dashboard からだけでなく、プログラムによって制御できます。サーバ動詞には通常 2 レベルのセキュリティチェックがあり、次の順に実行されます。

1. ユーザは、サーバ動詞そのものに対するアクセス権を持っているか。

2. ユーザは、サーバ動詞が実行される対象のリソースに対する権限を持っているか。

システム管理者は、[セキュリティ領域管理] ページを使用して、サーバ動詞へのアクセス権を制御できます。My webMethods Server には、デフォルトのセキュリティ領域が付属しており、管理者が各種サーバ機能に対するアクセス権を管理する際に役立ちます。デフォルトのセキュリティ領域の詳細については、182 ページの「セキュリティ領域の使用」を参照してください。

My webMethods Server での承認の決定


スタート
ACL 上の各 ACE に対して
アクセスの評価を繰り返す
役割
アイテム
削除済み？
Yes
権限を拒否
No
自分のグループまた
は役割をもとに、
許可された権限と拒否
された権限を検出
排他的
ロック？
Yes
ロックを
保持？
No
No
Yes
Yes
権限は
拒否された？
No
Yes
アクセスは
許可された？
No
権限は
許可された？
No
No
No
Yes
アクセスは
拒否された？
Yes
所有者か？
Yes
読み取り権限？
No
Yes
権限を許可
自分のグループまた
は役割をもとに、
許可された権限と拒否
された権限を検出
Yes
所有者か？
No
権限は
拒否された？
Yes
自分のグループ
または役割の
所有か？
No
No
Yes
権限は
許可された？
No
役割
許可
拒否

認証の管理

認証スキームとは、ユーザクレデンシャルを収集し、その真偽を検証する方法です。My webMethods Server 内で、デフォルトの認証スキームを指定することにより、サーバの認証を全体的に管理できます。初期設定では、フォーム認証スキームがすべてのサーバリソースに対するデフォルトです。さらに、各サーバリソースには認証スキームを設定でき、このスキームは、展開全体に対する設定より優先されます。

注:
セキュリティ領域は、常に認証スキームより優先されます。

次の表に、My webMethods Server が使用する認証スキームを示します。

スキーム	説明
Anonymous	サーバリソースに対する無制限のアクセス権を許可します。パブリックに公開される可能性があり機密情報が含まれていないサーバの非保護領域に使用されます。ユーザはクレデンシャルの提示を要求されないため、Anonymous 認証スキームは、ログインページに適しています。
フォーム	認証されていないユーザにフォームを提示し、必要なクレデンシャルを収集して、サーバに渡します。フォーム認証スキームは、認証されていない要求をデフォルトのログインページにリダイレクトするため、すべてのサーバリソースに対するデフォルトです。
基本	通常、クレデンシャルを HTTP ヘッダーのパラメータとして渡します。Basic 認証の場合、ユーザにはネイティブウィンドウシステムでレンダリングされるポップアップウィンドウが表示されます。
ケルベロス	Windows でのユーザのシングルサインオンを有効にします。Windows によって認証済みの My webMethods Server ユーザは、My webMethods Server にアクセスするためにログインし直す必要がありません。詳細については、274 ページの「ケルベロス認証の設定」を参照してください。
HTTP ヘッダー	サードパーティのセキュリティおよびアクセスコントロール製品 (Computer Associates、Oblix など) の外部 HTTP 認証クレデンシャルを受け入れます。この認証スキームを有効にすると、サーバでは他のすべての認証スキームが無視されます。詳細については、280 ページの「外部設定クレデンシャルの設定」を参照してください。
NTLM	さまざまな Microsoft ネットワークプロトコルの実装の認証に使用されます。Windows の展開では、NTLM 認証スキームがサーバに対するデフォルトである場合、ユーザは Windows ドメインにログインしていれば、サーバリソースに対する再認証を行う必要がありません。NTLM 認証の詳細については、以下を参照してください。 ■ 277 ページの「NTLM 認証の設定」 ■ 278 ページの「NTLMv2 認証の設定」

スキーム	説明
SAML	SAML を介したシングルサインオン (SSO) をサポートします。SAML を使用すると、ターゲットコンピュータ上のアプリケーションは、ソースコンピュータのアサーションに基づいてアクセス権を付与します。シングルサインオンの詳細については、282 ページの「My webMethods Server シングルサインオンの設定」を参照してください。
OAuth 2.0	サードパーティの ID プロバイダからのクレデンシャルを使用して、My webMethods にログインし、階層化されたアプリケーションや他のサーバリソースにアクセスにできます。詳細については、289 ページの「OAuth 2.0 認証の設定」を参照してください。

デフォルトの認証スキームの指定

My webMethods Server が初期化されると、フォーム認証スキームがデフォルトになります。フォーム認証スキームは、認証されていない要求をデフォルトのログインページにリダイレクトします。My webMethods Server に対するデフォルトの認証スキームは、登録されている認証スキームのいずれかに変更できます。

注:
サードパーティのセキュリティプロバイダからのクレデンシャルを受け入れるために HTTP ヘッダー認証スキームを使用する場合は、次の手順を使用しないでください。代わりに、280 ページの「外部設定クレデンシャルの設定」で説明する [HTTP ヘッダー認証管理] ページを使用します。

My webMethods Server のデフォルトの認証スキームを変更するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [エイリアス管理]
2. [**キーワード**] タブで、「`auth.scheme.default`」と入力して [**検索**] をクリックします。
3. 「auth.scheme.default」の をクリックします。
4. [エイリアスの更新] パネルで、[**参照**] をクリックして [**フォルダ**] > [システム] > [認証スキーム] を参照します。
5. 適切な認証スキームを [**選択された項目**] に移動し、[**選択**] をクリックします。
6. [**更新**] をクリックします。

サーバリソースへの認証スキームの割り当て

各サーバリソースには認証スキームを設定でき、このスキームは、My webMethods Server のデフォルトの認証スキームより優先されます。具体的な手順は、リソースがセキュリティ領域のメ

ンバーであるかどうかによって決まります。有効な認証スキームについては、270ページの「認証の管理」を参照してください。

個々のサーバリソースの認証スキームを割り当てるには

1. システム管理者として [**ホーム**] > [フォルダ] を選択し、個々のサーバリソースに移動します。
2. をクリックし、[権限] を選択します。
3. [**セキュリティ領域**] タブをクリックします。
4. [**セキュリティ領域名**] フィールドに「**セキュリティ領域が割り当てられていません。**」と表示される場合は、以下の手順に従います。
   a. [**認証**] タブをクリックします。
   b. [**認証スキーム**] リストで、サーバリソースに適用する認証スキームを選択して、[**適用**] をクリックします。
5. [**セキュリティ領域名**] フィールドにセキュリティ領域名が表示される場合は、以下の手順に従います。
   a. セキュリティ領域名をコピーして、ページバナーの入力フィールドにコピーし、[**検索**] をクリックします。

      または、[**フォルダ**] > [管理用フォルダ] > [Administration Dashboard] > [設定] > [セキュリティ領域管理] に移動して、セキュリティ領域を探すこともできます。
   b. をクリックし、[権限] を選択します。
   c. [**認証**] タブをクリックします。
   d. [**認証スキーム**] リストで、セキュリティ領域に適用する認証スキームを選択して、[**適用**] をクリックします。

ログイン後のユーザのリダイレクト

デフォルトでは、サーバにログインしたユーザは、同じページにリダイレクトされます。ログインポートレットの [ログインターゲット] プロパティを変更すると、ログインに成功したユーザを指定したページにリダイレクトできます。

ログイン後にユーザを別のページにリダイレクトするには

1. ログインポートレットのタイトルバーの右端で、をクリックし、[プロパティ] を選択します。

2. [**ログインターゲット**] プロパティに対して、以下のいずれかを行います。

   - [参照] をクリックし、ターゲットのページを [選択された項目] ボックスに移動して、[選択] をクリックします。
   - [エイリアスを使用] をクリックし、[エイリアス名] フィールドでユーザをリダイレクトする先のページのエイリアスを入力します。[**テスト**] をクリックして、そのエイリアスが有効で、エイリアスターゲットが正しいものであるかどうかを確認します。エイリアスが正しければ、[**選択**] をクリックします。

   注:
   「`/user.current.start.page`」と入力した場合、ユーザはスタートページルールで定義されているスタートページにリダイレクトされます。

3. [**適用**] をクリックします。

認証されない要求のリダイレクト

フォーム認証スキームを使用している場合、保護されているサーバリソースに対する要求が認証されないと、ユーザはデフォルトのログインページにリダイレクトされます。代わりに、カスタムログインページや非保護のコンテンツを提供しているページなどの指定したターゲットにユーザをリダイレクトすることができます。

認証されない要求を別のページにリダイレクトするには

1. システム管理者として次のように選択します。[**管理**] > [システム] > [認証スキーム]
2. デフォルトの認証スキームの をクリックし、[プロパティ] を選択します。

   デフォルトでは、フォーム認証スキームにより、認証されない要求はデフォルトのログインページにリダイレクトされます。[プロパティ] ページを見れば、認証スキームがデフォルトであるかどうかを確認できます。デフォルトの場合は、[**エイリアス**] リストに auth.scheme.default エイリアスが含まれています。

3. 認証スキームの [プロパティ] ページの [**リダイレクトの実行**] リストで、[**はい (リダイレクトする)**] が選択されていることを確認します。
4. [**URI のリダイレクト**] で、以下のいずれかを行います。

   - [参照] をクリックし、ターゲットのページを [選択された項目] ボックスに移動して、[選択] をクリックします。
   - [エイリアスを使用] をクリックし、[エイリアス名] フィールドでユーザをリダイレクトする先のエイリアス名を入力します。[**テスト**] をクリックして、そのエイリアスが有効で、エイリアスターゲットが正しいものであるかどうかを確認します。エイリアスが正しければ、[**選択**] をクリックします。

5. [**適用**] をクリックします。

ケルベロス認証の設定

次の大まかな手順を実行して、Windows 統合認証とケルベロスによる My webMethods Server のシングルサインオンを有効にします。

1. ケルベロス認証の KDC として、Active Directory と、Active Directory をホストする Windows Server を設定します。
2. Active Directory のディレクトリサービスと、My webMethods Server のケルベロス認証を設定します。
3. ケルベロスによるシングルサインオンの Web クライアントを設定します。

ケルベロストークンの処理中に例外が発生したなどの原因でケルベロスによる SSO を使用した要求の認証が失敗した場合、My webMethods Server でデフォルト形式の認証スキームが使用されて、認証されなかった要求がログインページにリダイレクトされます。デフォルトの認証スキーム、または認証されなかった要求のターゲットページを変更すると、My webMethods Server で新しいデフォルトのスキームまたはページが使用されます。

環境に複数の Web クライアントが含まれる場合、ケルベロス認証を使用するように設定できるのは、特定のクライアントのみです。未設定の Web クライアントは、ケルベロス認証スキームが My webMethods Server へのログインで使用されたときに空白ページを表示します。このような場合、Web クライアントをリフレッシュして、デフォルトの認証スキームを使用して、ログインページを使用してログインすることができます。

ケルベロス認証用の Windows Server および Active Directory の設定

以下の手順を実行して、ケルベロス認証用の KDC として Active Directory を設定します。他の指定がない限り、Active Directory をホストする Windows Server マシンで手順を実行します。

ケルベロス認証用のキー配布センターとして Active Directory を設定するには

1. Active Directory で Kerberos プリンシパルのユーザアカウントを設定します。暗号化は選択しないでください。デフォルトの暗号は RC4 です。
2. ktpass コマンドラインユーティリティと次のコマンドを使用して、サービスプリンシパル名 (SPN) と keytab ファイルを作成します。

```
ktpass -out <Keytab_File_Name>.keytab -princ
HTTP/<FQDN_of_Active_Directory_Server>@<Domain_Name> -mapUser
 <FQDN_of_Active_Directory_Server>@<FQDN_of_MWS_Server_Machine> -mapop
set<MWS_Server_User_Password> -crypto all -ptype KRB5_NT_PRINCIPAL -kvno 0
```

例:

```
ktpass -out MWS_Kerberos_User.keytab -princ HTTP/VMHOSTNAME.SPARTA.RNDLAB.
LOC@SPARTA.RNDLAB.LOC
-mapUser Bob@SPARTA.RNDLAB.LOC
-mapOp set -pass pass12345 -crypto all
```

```
-pType KRB5_NT_PRINCIPAL -kvno 0
```

MWS_Kerberos_User は keytab ファイルの名前、Bob はユーザ、SPARTA.RNDLAB.LOC は AD サーバの完全修飾ドメイン名です。

3. keytab ファイルを、My webMethods Server をホストしているマシンの任意のディレクトリにコピーします。

4. 以下の java コマンドを <JAVA_INSTALL> /jre/bin から実行して、keytab ファイルが正しく作成されていることを確認します。

```
kinit -J-Dsun.security.krb5.debug=true -k
  -t <Keytab_file_absolute_path> HTTP/<FQDN_of_Active_Directory_Server>
```

ケルベロス認証用の My webMethods Server の設定

以下の手順を使用して、My webMethods Server をケルベロス認証用に設定します。

ケルベロス認証のための My webMethods Server の設定

1. My webMethods Server に Administrator としてログインし、ケルベロス認証サーバ用に設定した Active Directory の LDAP を設定するか、My webMethods Server でクライアントユーザアカウントを作成します。Windows サーバマシンのユーザアカウントも LDAP または My webMethods Server ユーザアカウントに含める必要があります。

   LDAP 設定の詳細については、116 ページの「外部 LDAP、ADSI、または ADAM ディレクトリサービスの設定」を参照してください。

2. *Software AG_directory* \profiles\MWS_default\configuration\jaas.conf ファイルを編集して以下のコードをファイルの末尾に追加し、ファイルを保存します。

```
spnego-server{
com.sun.security.auth.module.Krb5LoginModule required
doNotPrompt=true
principal="HTTP/<FQDN_of_Active_Directory_Server>"
useKeyTab=true
keyTab="<Keytab_file_absolute_path>"
storeKey=true
isInitiator=false
debug=false;
};
```

3. *Software AG_directory* \profiles\MWS_default\configuration\custom_wrapper.conf ファイルを編集して以下のプロパティを追加し、ファイルを保存します。

```
wrapper.java.additional.602=-Dsun.security.krb5.debug=false
wrapper.java.additional.603=-Djavax.security.auth.
   useSubjectCredsOnly=false
```

4. My webMethods Server を再起動します。

5. My webMethods Server にシステム管理者としてログインします。

6. [**設定**] > [ケルベロス認証管理] の順に移動し、領域 (KDC で管理されるすべてのマシンを指定する) と KDC サーバに適切な値を指定します。

7. [**設定**] > [エイリアス管理] の順に移動し、My webMethods Server のデフォルト認証スキーマをケルベロスに変更します。手順については、『271 ページの「デフォルトの認証スキームの指定」』を参照してください。

8. My webMethods Server を再起動します。

ケルベロス認証を使用するための Web クライアントの設定

ケルベロス認証をサポートするための Internet Explorer の設定

Internet Explorer を設定するには、以下の手順に従います。

1. Internet Explorer で、[**ツール**] > [インターネットオプション] の順に選択します。

2. [**セキュリティ**] をクリックして、以下を実行します。

   a. [**ローカルイントラネット**] ゾーンを選択します。

   b. [**サイト**] をクリックします。

   c. [**イントラネットのネットワークを自動的に検出する**] チェックボックスをオフにします。

   d. 以下のすべてのオプションを選択します。

      - **ほかのゾーンに指定されていないローカル(イントラネット)のサイトをすべて含める**
      - **プロキシサーバを使用しないサイトをすべて含める**
      - **すべてのネットワークパス (UNC) を含める**

   e. [**詳細設定**] をクリックして、[**Web サイト**] リストに Windows Server のアドレスが示されていることを確認します。

      Windows Server のアドレスについては、「274 ページの「ケルベロス認証用の Windows Server および Active Directory の設定」」を参照してください。

   f. [**閉じる**] および [**OK**] をクリックして、[セキュリティ] 画面に戻ります。

   g. セキュリティレベルを設定するには、[**レベルのカスタマイズ**] をクリックし、[セキュリティ設定] の [**ユーザ認証**] までスクロールして、[**イントラネットゾーンでのみ自動的にログオンする**] を選択します (未選択の場合)。

3. プロキシサーバが設定されている場合は、以下を実行してサーバドメインを含めます。

   a. [**接続**] > [LAN の設定] > [プロキシサーバ] で、[**LAN にプロキシサーバを使用する**] を選択します。

   b. [**詳細設定**] > [例外] をクリックして、Active Directory のドメイン名を追加します。

ケルベロス認証をサポートするための Mozilla Firefox の設定

Mozilla Firefox を設定するには、以下の手順に従います。

1. アドレスバーに「`about:config`」と入力して Enter キーを押します。
2. network.negotiate-auth.trusted-uris までスクロールダウンして、ダブルクリックします。
3. 設定済みの Windows Server のドメイン名を信用される URL リストに追加するために、既存の値を追加または変更します。URL のカンマ区切りリストを使用してください。

ケルベロス認証をサポートするための Google Chrome の設定

Google Chrome を設定するには、以下の手順に従います。

1. コマンドラインで `Chrome.exe --auth-server-whitelist="*.Active_Directory_Domain_Name"` コマンドを使用して、Google Chrome を起動します。

NTLM 認証の設定

NTLM 認証を使用すると、Windows ドメインにログインしているユーザは、サーバにログインするために再認証を行う必要はありません。プライマリドメインコントローラは、ドメイン内のすべてのアカウントの処理を担当する Microsoft Windows サーバです。NTLM 認証を使用するには、NTLM 認証スキームにおけるプライマリドメインコントローラを明示的に指定する必要があります。

NTLM 認証用のプライマリドメインコントローラを指定するには

1. NTLMv2 認証を設定した場合は、無効にする必要があります。詳細については、279 ページの「NTLMv2 認証の無効化」を参照してください。
2. システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [NTLM 認証管理]
3. [プロパティ] ページの [**ドメインコントローラ名**] フィールドに、プライマリドメインコントローラのホスト名を入力し、[**サブミット**] をクリックします。

NTLM 認証の無効化

NTLM が有効な場合は、NTLMv2 を設定する前に無効にする必要があります。

NTLM 認証を無効にするには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [NTLM 認証管理]

2. [**ドメインコントローラ名**] フィールドで、空白を含むすべての文字を削除し、[**サブミット**] をクリックします。

NTLMv2 認証の設定

NTLM バージョン 2 (NTLMv2) を使用するには、IOPLEX Software から最新バージョンの Jespa Java ソフトウェアライブラリを購入してインストールする必要があります。

NTLMv2 認証を設定するには

1. NTLM認証を設定した場合は、無効にする必要があります。詳細については、278 ページの「NTLM 認証の無効化」を参照してください。

2. Jespa をダウンロードしたときに提供されたマニュアルの説明に従って、Jespa Java ソフトウェアライブラリをインストールおよび設定します。

   Jespa ライブラリは次の場所にあります。

   Software AG_directory /MWS/lib/

3. My webMethods Server を停止します。

4. wm_ntlmv2authadmin.pdp から jcifs-1.3.17.jar を抽出します。このファイルは以下の場所にあります。

   Software AG_directory /MWS/components/admin/configuration/ wm_ntlmv2authadmin.pdp

5. この jcifs-1.3.17.jar の他に、Jespa Java ソフトウェアライブラリで jespa-1.1.11.jar を取得して、次の場所にコピーします。

   Software AG_directory /MWS/lib

6. コマンドラインで、`mws update` を実行します。

7. My webMethods Server を再起動します。

8. システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [NTLMv2 認証管理]

9. [**NTLMv2 が有効**] フィールドで、[**はい (NTLMv2 は有効)**] が設定されていることを確認します。

10. Jespa のマニュアルで提供される情報に従って、残りのフィールドのプロパティを設定します。

11. 設定をテストする場合は、[**はい (保存前に設定をテストする)**] を選択します。

12. [**サブミット**] をクリックします。

13. 新しいライブラリが設定されていることを確認するには、以下の手順に従います。

    a. コマンドラインプロンプトで、次のコマンドを入力してサーバの bin ディレクトリに移動します。

    ```
    cd Software AG_directory\MWS\bin
    ```

    b. 「`mws stop`」と入力します。

    c. 「`mws update`」と入力します。

    d. 「`mws start`」と入力します。

NTLMv2 認証の無効化

NTLMv2 が有効な場合は、NTLM を設定する前に無効にする必要があります。

NTLMv2 認証を無効にするには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [NTLMv2 認証管理]

2. [**NTLMv2 が有効**] フィールドで、[**いいえ (NTLMv2 は無効)**] が設定されていることを確認し、[**サブミット**] をクリックします。

3. Jespa および Java CIFS ライブラリを次の場所から削除します。

   Software AG_directory /MWS/lib/ext

4. ライブラリの参照を解除し、サーバを再設定するには、以下の手順に従います。

   a. コマンドラインプロンプトで、次のコマンドを入力してサーバの bin ディレクトリに移動します。

```
cd /Software AG_directory\MWS\bin
```

b. 「mws stop」と入力します。

c. 「mws update」と入力します。

d. 「mws start」と入力します。

外部設定クレデンシャルの設定

[HTTP ヘッダー認証管理] ページでは、システム管理者が、SiteMinder (Computer Associates) や Oblix などサードパーティのセキュリティおよびアクセスコントロール製品から外部 HTTP 認証クレデンシャルを受け入れるように My webMethods Server を設定できます。これらのクレデンシャルでは大文字と小文字が区別されます。プラットフォームと Web サーバによって異なりますが、通常は sm_user か SM_USER のいずれかになります。

認証の有効化

重要:
[HTTP ヘッダー認証管理] ページは、サードパーティのセキュリティプロバイダを使用している場合にのみ有効にします。ページを有効にすると、サーバはすべてのユーザが認証済みであるかのように動作します。

サードパーティのセキュリティおよびアクセスコントロール製品の認証を受け入れるには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [HTTP ヘッダー認証管理]
2. [**ユーザヘッダー名**] で、「sm_user」または「SM_user」と入力します。
3. [**HTTP ヘッダー認証管理を有効にする**] チェックボックスをオンにします。
4. 必要に応じて、[**ログアウト URL**] フィールドに、ユーザがサーバをログアウトした後にリダイレクトされる先の URL を入力します。
5. [**サブミット**] をクリックします。
6. ベンダーの製品ドキュメントの指示に従って、サードパーティのセキュリティおよびアクセスコントロールソフトウェアを設定します。

注:
外部セキュリティおよびアクセスコントロール製品を適切に設定するには、My webMethods Server とサードパーティ製品の両方が同じディレクトリサーバインスタンスを指している必要があります。

HTTP ヘッダー認証の問題をログで確認

HTTP ヘッダー認証が正常に機能しない場合、ログファイルを確認して、問題の診断に役立てることができます。HTTP ヘッダー認証のログメッセージは、portalLogin カテゴリに割り当てられます。HTTP ヘッダー認証のログメッセージを表示するには、ログしきい値を変更する必要があります。コンソール、_full.log ファイルおよび portalLogin.log ファイルに書き込むためのデフォルトのしきい値は、INFO ログレベルに設定されていますが、HTTP ヘッダー認証のログメッセージでは、それより低い DEBUG ログレベルが使用されます。

サーバのログファイルは、 *Software AG_directory* \MWS\server*serverName*\logs ディレクトリにあります。ログ収集の制御の詳細については、322 ページの「ログメッセージの表示」を参照してください。ログメッセージの検索の詳細については、322 ページの「ログメッセージの表示」を参照してください。

ログインのログしきい値の設定

ログメッセージを出力に書き込むには、カテゴリと出力の設定をどちらも DEBUG に設定する必要があります。ログしきい値の設定の詳細については、310 ページの「サーバのログの制御」を参照してください。

HTTP ヘッダー認証のカテゴリおよび出力のしきい値を設定するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [分析] > [ログ設定] > [ログのしきい値]

2. [**カテゴリしきい値**] リストで、以下のログカテゴリの一部またはすべてに対して [DEBUG] ログレベルを選択します。

次の表に使用可能なログカテゴリを示します。

ログカテゴリ	説明
root	コンソールおよび _full.log ファイルの出力を制御します。
portalLogin	portalLogin.log ファイルの出力を制御します。

3. [**出力しきい値**] リストで、以下のログ出力タイプの一部またはすべてに対して [DEBUG] ログレベルを選択します。

次の表に使用可能なログ出力を示します。

ログ出力	説明
Console	コンソールに送信されるログメッセージの出力を制御します。
_full.log	_full.log ファイルに送信されるログメッセージの出力を制御します。

ログ出力	説明
portalLogin	portalLogin.log ファイルに送信されるログメッセージの出力を制御します。

4. [適用] をクリックします。

HTTP ヘッダー認証ログでの問題の確認

HTTP ヘッダー認証を有効にすると、サーバはすべてのユーザが認証済みであるかのように動作します。これを念頭に置くと、ログメッセージは、以下の節で説明するような、3つの結果を示すと考えられます。

ログインが成功した場合

HTTP ヘッダー認証を使用したログイン成功のメッセージは、次の例のようになります。

```
Date_and Time (portalLogin : DEBUG) - HttpHeaderHandler Auth Handler

   looking for: user_name

Date_and Time (portalLogin : DEBUG) - Found userID: user_name
```

user_name は、HTTP ヘッダー認証でログインしたユーザの名前です。

HTTP ヘッダー認証が無効な場合

HTTP ヘッダー認証を有効にしていない場合、ログメッセージは次の例のようになります。

```
Date_and Time (portalLogin : DEBUG) - HttpHeaderHandler Auth Handler is

   not enabled
```

HTTP ヘッダー認証を有効にするには、280 ページの「認証の有効化」を 参照してください。

サードパーティのサイトが問題の原因である場合

サードパーティサイトが正しく設定されていない場合、HTTP ヘッダー認証は失敗します。その結果生成されるログメッセージは、次の例のようになります。

```
Date_and Time (portalLogin : DEBUG) - HttpHeaderHandler Auth Handler

   looking for:

Date_and Time (portalLogin : DEBUG) - No value found!
```

My webMethods Server シングルサインオンの設定

シングルサインオン (SSO) により、ユーザがあるアプリケーションにログインしても、別途ログインすることなく他のアプリケーションを使用できます。My webMethods Server では、セキュ

リティ情報交換の XML ベースのフレームワークである SAML を介したシングルサインオンをサポートしています。SAML を使用すると、ターゲットコンピュータ上のエンティティは、ユーザがソースコンピュータにログインしたというソースコンピュータからのアサーションを基にアクセス権を付与します。アサーションへの署名に使用する証明書を My webMethods Server のターゲットインスタンス上のトラストストアに追加する必要があります。詳細については、32 ページの「CA 証明書のインポート」を参照してください。

My webMethods Server では、以下のようにシングルサインオン機能が提供されています。

- ソースサーバと 1 つ以上のターゲットサーバ間
- あるサーバと、シングルサインオン機能を持つ他の webMethods アプリケーション間
- サーバと、SAML をサポートするサードパーティアプリケーション間
- サーバとサードパーティの ID プロバイダ (IDP) 間
- (廃止) サーバと、ユーザサインオンを認証するアーティファクトレシーバ、およびターゲット Web アプリケーションの間

このモデルを使用すると、1 つのサーバがソースになり、ユーザの集中ログインが可能となります。ソースサーバ上のページのリンクは、任意の数の SAML 対応エンティティを指します。また、ターゲットサーバでは、ターゲットサーバのトラストストアにソースサーバの証明書がある限り、任意の数のサーバからアサーションを受け入れることもできます。

シングルサインオンを利用するには、ユーザがソースサーバとターゲットエンティティの両方で認識されている必要があります。ほとんどの場合、同じディレクトリサービスを使用することでユーザを共通に認識できます。

シングルサインオンのターゲットとしてのサーバの設定

サーバは、一度に 1 つのシングルサインオンソースに対してのみターゲットとなります。

シングルサインオンのターゲットとしてサーバを設定するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [SAML 認証管理]

2. [プロパティ] ページで次を指定します。

次の表に、シングルサインオンのターゲットとしてサーバを設定するのに必要なプロパティを示します。

プロパティ	説明
[アーティファクトパラメータ名]	(廃止) 別の webMethods サーバとの SAML 接続である場合は、デフォルト値の [SAMLart] を変更しないでください。サードパーティソースへの SAML 接続である場合は、サードパーティアプリケーションで使用されているアーティファクトパラメータ名を入力します。

プロパティ	説明
[アサーションパラメータ名]	サーバが SAML アサーション値を検索する HTTP 要求パラメータ名。デフォルト値は SAMLResponse です。
[セキュリティプロバイダ URI]	(廃止 - [アーティファクトパラメータ名] と共に使用) SAML セキュリティプロバイダ (ソース) の URI を入力します。これが別の webMethods サーバとの接続である場合は、次の構文を使用します。`server:port/services/SAML` *server* はソースサーバが稼動しているホストで、*port* はサーバのポート番号です。デフォルトのポート番号は 8585 です。

3. [**サブミット**] をクリックします。

ソースサーバでの SAML リンクの設定

任意のページで、サーバなどの SAML ターゲットエンティティへのリンクを追加できます。ターゲットがソースサーバからの SAML アサーションを受け入れる場合は、既知のユーザがリンクをクリックしたときに、ログインクレデンシャルは要求されません。ターゲットエンティティがソースサーバからの SAML アサーションを受け入れない場合、またはユーザがターゲットエンティティで認識されていない場合は、ログインクレデンシャルが要求されることがあります。

(廃止 - SAML アーティファクトレシーバでのみ有効) SAML 仕様では、アーティファクトレシーバと呼ばれる媒介が、ターゲットの Web アプリケーションに代わって認証を実行できます。このような場合、SAML ソースは 2 つの URL を必要とします。一方はアーティファクトレシーバ用で、もう一方はターゲット Web アプリケーション用です。編集権限のある任意のページに 1 つ以上の SAML リンクを配置できます。

編集権限のある任意のページに 1 つ以上の SAML リンクを配置できます。

ソースページに SAML リンクを作成するには

1. ページの右上で をクリックし、[ページを編集] を選択します。
2. [**使用可能なポートレット**] パネルの [**ルート**] リストで、[**リンク**] をクリックします。
3. [**使用可能なポートレット**] パネルの [**リンク**] リストで、[**wm_xt_ssolink**] ポートレットをドラッグし、ページ上のリンクを追加する位置にドロップします。

   カーソルを有効なページ位置に移動するたびに、カーソル位置の下に赤いボックスが表示されます。このボックスは、マウスボタンを離した場合にポートレットが配置される場所を示します。
4. ページ制御領域の左側で、[**保存**] をクリックします。
5. シングルサインオンポートレットのタイトルバーの右端で、 をクリックし、[プロパティ] を選択します。

6. [プロパティ] ページで次を指定します。

次の表に、SAML リンクを設定するときに変更するプロパティを示します。

フィールド	説明
[名前]	`wm_xt_ssolink` をリンクの内容に合ったテキストで置換します。
[SAML タイプ]	使用する SAML 仕様のバージョンを選択します。 ■ SAML2 POST ■ SAML1 POST ■ SAML Artifact — (廃止)
[SAML 認証 URL]	ターゲットコンピュータ上のリソースの URL を入力します。サーバ上の任意のページをターゲットにできます。 (廃止) SAML アーティファクトレシーバを介して Web アプリケーションに接続している場合は、このフィールドをアーティファクトレシーバ URL 用に使用します。
[POST または GET の使用]	(廃止 - [SAML タイプ] フィールドが [SAML Artifact] に設定されている場合にのみ有効) ターゲットコンピュータにデータを渡すために使用するメソッドを決定します。 POST — HTTP 要求の本文でデータを渡します。デフォルトである POST は、シングルサインオンデータに対して推奨される方法です。 [GET を使用] — URL に付加した疑問符の後のストリングとしてデータを渡します。
[アサーションパラメータ名]	サーバが SAML アサーション値を検索する HTTP 要求パラメータ名。デフォルト値は SAMLResponse です。
[アーティファクトパラメータ名]	(廃止 - [SAML タイプ] フィールドが [SAML Artifact] に設定されている場合にのみ有効) 別のサーバまたは他の webMethods 製品との SAML 接続である場合は、デフォルト値の [SAMLart] を変更しないでください。サードパーティソースへの SAML 接続である場合は、サードパーティアプリケーションで使用されているアーティファクトパラメータ名を入力します。
[アプリケーションターゲット URL]	(廃止) SAML アーティファクトレシーバの URL を **[SAML 認証 URL]** フィールドに入力した場合は、Web アプリケーションの URL を入力します。それ以外の場合は、このフィールドを空白のままにします。

7. **[適用]** をクリックします。

サードパーティ ID プロバイダを使用するためのシングルサインオンの設定

My webMethods Server を SAML コンシューマとして使用し、サードパーティ ID プロバイダ (IDP) を SAML 認証局として使用することにより、IDP によって開始されるシングルサインオン (SSO) を有効にできます。

サードパーティ IDP を使用して SSO を設定するには、以下の手順に従います。

1. 安全な HTTPS ポートを使用するように My webMethods Server が設定されていることを確認します。

2. websso.properties ファイルでプロパティを設定します。ファイル内でのプロパティ設定の詳細については、『287 ページの「websso.properties ファイルでのプロパティの設定」』を参照してください。

3. IDP の証明書を My webMethods Server トラストストアにインポートして、信用のある IDP ソースから受信した有効なメッセージを My webMethods Server が信頼するようにします。JVM の keytool コマンドを使用します。My webMethods Server トラストストアへの認証のインポートの詳細については、32 ページの「CA 証明書のインポート」を参照してください。

4. My webMethods Server を起動します。

   起動時に、My webMethods Server は *Software AG_directory* \MWS\server*serverName*\config ディレクトリに `SPMetadata.xml` および `IDPMetadata.xml` という 2 つのメタデータファイルを作成します。

5. サービス プロバイダ (My webMethods Server) を外部の ID プロバイダに登録するには、*Software AG_directory* \MWS\server*serverName*\config\SPMetadata.xml ファイルを My webMethods Server から IDP サーバにコピーします。

   ID プロバイダは、`SPMetdata.xml` からの My webMethods Server インスタンスのエンドポイントロケーションを使用して、サービスプロバイダとして My webMethods Server を表示します。

6. Software AG は、暗号化されたアサーションをデフォルトでサポートしています。*Software AG_directory* \jvm\operating_system\jre\lib\security フォルダにある以下のデフォルトの JCE ポリシーファイルを最新の JCE ファイルに置き換えます。

   - local_policy.jar
   - US_export_policy.jar

7. My webMethods Server を再起動します。

8. IDP によって開始されるシングルサインオンを確認します。

Mircrosoft ADFS (Active Directory Federation Services) をサードパーティ IDP として使用している場合は、「287 ページの「IDP (Mircrosoft ADFS) によって開始されるシングルサインオンの確認」」を参照してください。

IDP (Mircrosoft ADFS) によって開始されるシングルサインオンの確認

1. IDP の SSO URL をブラウザに入力し、有効な IDP クレデンシャルを使用してログインします。

2. Mircrosoft ADFS に登録されている My webMethods Server インスタンスを選択します。

   - IDP はユーザを認証応答と共に My webMethods Server にリダイレクトし、SAML POST バインドを使用して SAML 応答トークンを POST パラメータとして My webMethods Server に送信します。
   - My webMethods Server は SAML 応答内の詳細な署名情報に基づいて、SAML 応答を検証します。アサーションの署名は、メタデータファイル内の ID プロバイダの公開キーを使用して検証されます。
   - My webMethods Server は SAML 応答を処理し、トークンに存在する詳細なユーザ情報を検証しますユーザ ID は My webMethods Server (SP) と IDP の間で合意済みの ID です。ユーザ ID が My webMethods Server に登録されている場合、My webMethods Server がユーザアクセスを認証します。

3. ログインクレデンシャルを再入力せずに My webMethods Server にアクセスできるか確認します。

websso.properties ファイルでのプロパティの設定

websso.properties ファイルで、サードパーティ ID プロバイダ (IDP) を使用するシングルサインオンに必要なプロパティを設定します。

websso.properties ファイルでプロパティを設定するには

1. コマンドラインプロンプトで、次のコマンドを入力してサーバの bin ディレクトリにアクセスします。

```
cd Software AG_directory\MWS\bin
```

2. 次のコマンドを入力して、My webMethods Server データベースから websso.properties ファイルを抽出します。

```
mws.{sh|bat} -s serverName getconfig websso.properties
```

3. websso.properties ファイルをテキストエディタで開きます。ファイルは次の場所にあります。

   Software AG_directory \MWS\server*serverName*\config\

4. websso.properties ファイルでプロパティを設定します。この各プロパティの詳細については、288 ページの「サードパーティ IDP を使用するシングルサインオンのプロパティ設定」を参照してください。

5. 次のコマンドを入力して、My webMethods Server データベースで websso.properties ファイルを保存します。

```
mws.{sh|bat} -s serverName putconfig websso.properties
```

6. 次の場所から websso.properties ファイルを削除します。

   Software AG_directory \MWS\server*serverName*\config\

   このファイルを削除しないと、サーバは引き続き websso.properties ファイルのこの場所のローカルバージョンを使用します。

7. My webMethods Server を再起動します。

サードパーティ IDP を使用するシングルサインオンのプロパティ設定

Software AG_directory \MWS\server*serverName*\config\websso.properties ファイルで、サードパーティ IDP を使用するシングルサインオンに必要な値を指定します。

次の表に、My webMethods Server のサードパーティ ID プロバイダを使用するシングルサインオンの設定に必要なプロパティを示します。

プロパティ	説明
SSO_KEYSTORE	SAML2.0 を使用する SSO 通信に使用されるキーストア。 My webMethods Server は、キーストアを *Software AG_directory* \MWS\server*serverName*\config\security ディレクトリに保存します。SSO_KEYSTORE の値は、config ディレクトリへの絶対パスまたは相対パスです。 デフォルトのキーストア localhost.p12 は、config\security ディレクトリにあります。
SSO_KEYSTORE_PASSWORD	キーストアのパスワード。 キーストアのパスワードは、プレーンテキストまたは暗号化したテキストにできます。パスワードの暗号化の詳細については、「35 ページの「暗号化パスワードの生成」」を参照してください。
SSO_KEYSTORE_TYPE	キーストアタイプ。 JKS または PKCS12 にできます。

プロパティ	説明
SSO_SIGN_ALIAS	SAML 応答の署名に使用するエイリアス名。
SSO_SIGN_ALIAS_PASSWORD	署名エイリアスのパスワード。
SSO_ENCRYPT_ALIAS	SAML 応答の暗号化に使用するエイリアス名。
SSO_ENCRYPT_ALIAS_PASSWORD	エイリアスの暗号化のパスワード。
SSO_DEFAULT_ALIAS	署名エイリアス (SSO_SIGN_ALIAS) と暗号化エイリアス (SSO_ENCRYPT_ALIAS) が同じ場合のデフォルトのエイリアス名。SSO_DEFAULT_ALIAS の値を指定すると、デフォルトエイリアスのパスワードはキーストアのパスワードと同じであると見なされます。 署名エイリアスと暗号化エイリアスを指定する場合、デフォルトエイリアスの指定は不要です。SSO_DEFAULT_ALIAS の値を指定すると、以下のプロパティの値セットは無視されます。 ■ SSO_SIGN_ALIAS ■ SSO_SIGN_ALIAS_PASSWORD ■ SSO_ENCRYPT_ALIAS ■ SSO_ENCRYPT_ALIAS_PASSWORD
SSO_IDP_METADATA_URL	ID プロバイダのメタデータファイルの URL。

websso.properties ファイルのプロパティ設定の例

```
SSO_KEYSTORE=config/security/localhost.p12
SSO_KEYSTORE_PASSWORD={AES}Y5IgMqjfvkgbg7p5VUZztw\=\=
SSO_KEYSTORE_TYPE=PKCS12
SSO_SIGN_ALIAS=localhost
SSO_SIGN_ALIAS_PASSWORD={AES}Y5IgMqjfvkgbg7p5VUZztw\=\=
SSO_ENCRYPT_ALIAS=localhost
SSO_ENCRYPT_ALIAS_PASSWORD={AES}Y5IgMqjfvkgbg7p5VUZztw\=\=
SSO_DEFAULT_ALIAS=localhost
SSO_IDP_METADATA_URL=
"http://example.org/metadata.xml"
```

OAuth 2.0 認証の設定

OAuth 2.0 の認証スキームでは、Google や、Twitter、Salesforce、または OAuth 2.0 と Open ID Connect プロトコルをサポートし、検出サービスを公開しているその他の ID プロバイダからのアカウントを使用して、My webMethods にログインできます。ユーザ役割と役割属性に基づいて、ユーザがアクセスできるサーバリソースや階層化された製品を制御できます。ユーザ役割は、ログイン時に動的にユーザのメンバーシップを確認するために ID プロバイダによって発行

されるクレームを受け入れるように設定します。My webMethods Server で、異なる ID プロバイダに対して、複数の OAuth 2.0 サービスを設定できます。

外部アカウントのユーザおよび役割

デフォルトでは、My webMethods Server は外部アカウントごとに、OAuth 2.0 の認証スキームを使用してログインする新しいシステムユーザを作成します。これらのシステムユーザは、OAuthSinkRole というデフォルトの役割に割り当てられます。カスタムの役割を作成し、OAuth 2.0 で認証されたユーザに割り当てるように設定できます。My webMethods Server 役割の詳細については、189 ページの「役割および My webMethods へのアクセスの管理」を参照してください。

必要に応じて、内部ユーザを外部ユーザアカウントに登録するカスタムサービスを作成することもできます。詳細については、*webMethods CAF and My webMethods Server Java API Reference* を参照してください。

OAuth 2.0 および OpenID Connect での認証フローの設定

ユーザがサードパーティ ID プロバイダからのクレデンシャルで My webMethods にログインできるようにするには、次の手順で行います。

1. My webMethods Server を認証サーバ/ID プロバイダに登録します。詳細については、290 ページの「My webMethods Server の ID プロバイダへの登録」を参照してください。
2. My webMethods Server で OAuth 2.0 サービスを設定します。詳細については、291 ページの「 OAuth 2.0 サービスの設定」を参照してください。
3. My webMethods ログインページをカスタマイズするか、ID プロバイダにユーザをリダイレクトするカスタムログインページを作成します。詳細については、292 ページの「OAuth 2.0 認証用の My webMethods ログインページのカスタマイズ」を参照してください。

My webMethods Server の ID プロバイダへの登録

My webMethods Server をクライアントアプリケーションとして登録する方法については、認証サーバ/ID プロバイダの公式ドキュメントを参照してください。後で、My webMethods Server で OAuth 2.0 サービスを設定するために、ID プロバイダへのユーザ登録から次の情報が必要です。

- ユーザの OpenID Connect プロバイダの検出サービスの URL
- My webMethods Server の OAuth 2.0 クライアント識別子
- My webMethods Server の OAuth 2.0 クライアントシークレット
- OpenID Connect プロバイダで事前登録されたリダイレクト URL
- My webMethods Server でユーザを登録できるアクセスクレーム

OAuth 2.0 サービスの設定

Administrator ユーザおよび sysadmin ユーザは、OAuth 2.0 サービスの追加、削除、変更ができます。

新規 OAuth 2.0 サービスを設定するには

1. [**OAuth2 管理**] ページに移動して、[**OAuth 設定の追加**] タブをクリックします。
   - sysadmin として、[**フォルダ**] > [My webMethods アプリケーション] > [ファブリックタスク] > [管理] > [My webMethods] > [OAuth2 管理] に移動します。
   - 管理者として、次のように選択します。[**アプリケーション**] > [管理] > [My webMethods] > [OAuth2 管理]
2. [**OAuth 設定の追加**] タブで、次を指定します。

次の表に、OAuth サービスを設定するのに必要なプロパティを示します。

フィールド	説明
[名前]	必須。OAuth 2.0 サービスの名前。
[サービスを有効化]	必須。ドロップダウンリストからオプションを選択して、OAuth 2.0 サービスを有効または無効にします。デフォルトでは、新しく作成されたサービスは有効になります。
[検出ドキュメント URL]	必須。OpenID Connect プロバイダの検出サービスの URL (プロバイダへのユーザ登録から)
[OAuth 2.0 クライアント ID]	必須。認証サーバで有効な OAuth 2.0 クライアント識別子 (プロバイダへのユーザ登録から)
[OAuth 2.0 クライアントシークレット]	必須。OAuh 2.0 承認に使用されるクライアントシークレット (プロバイダへのユーザ登録から)
[OpenID Connect スコープ]	必須。OpenID Connect によって定義されている要求される承認のスコープ 。デフォルトは「`openid,profile,email`」です。使用可能なスコープの詳細については、ID プロバイダのマニュアルを参照してください。
[リダイレクト URI]	必須。ID プロバイダの登録時に指定した My webMethods Server の URL。
[OpenID Connect プロバイダ名]	必須。OpenID Connect のプロバイダの名前。My webMethods Server は事前設定済みの [***provider_name* でログイン**] ボタンにこの名前を表示します。詳細については、292 ページの「OAuth 2.0 認証用の My

フィールド	説明
	webMethods ログインページのカスタマイズ」を参照してください。
[アクセスクレーム]	オプション。My webMethods Server でユーザを登録できるアクセスクレーム。使用可能なクレームの詳細については、ID プロバイダのマニュアルを参照してください。
[ユーザサービス名]	オプション。OAuth 2.0 フローを使用して認証された外部アカウントのために、My webMethods の内部ユーザを作成するカスタムサービスの名前。OAuth 2.0 認証フローにおけるユーザおよび役割の詳細については、289 ページの「OAuth 2.0 認証の設定」を参照してください。
[役割名]	オプション。カスタムの OAuth 属性に入れ込む役割の名前。OAuth 2.0 フローを使用して認証する My webMethods ユーザは、この役割に割り当てられます。デフォルトは「**OAuthSinkRole**」です。
[役割メンバー属性]	オプション。役割のメンバーシップ属性に追加するクレームのリスト。クレームのコンマ区切りのリストを指定します。使用可能なクレームの詳細については、ID プロバイダのマニュアルを参照してください。

3. [**サブミット**] をクリックします。

設定を送信すると、My webMethods Server の [**認証 URL**] フィールドに認証 URL が生成されます。この URL は、ログインのオプションとして OAuth 2.0 設定を追加する場合に必要です。OAuth 2.0 認証を My webMethods ログインページに追加する方法については、292 ページの「OAuth 2.0 認証用の My webMethods ログインページのカスタマイズ」を参照してください。

OAuth 2.0 認証用の My webMethods ログインページのカスタマイズ

My webMethods のデフォルトのログインオプションに、外部プロバイダの名前が付いたログインボタンを追加できます。ボタンのプロバイダ名は、[**OAuth2 設定の追加**] タブで指定した [**OpenID Connect プロバイダ名**] と同じ です。

OAuth 2.0 ログインオプションを追加するには

1. sysadmin として、[**フォルダ**] > [システム] > [ポートレット] > [管理] > [OAuth2 プロバイダ設定] に移動します。

2. [**ツール**] > [ページを編集] をクリックして、ログインポートレットを [**OAuth2 プロバイダ設定**] ページにドラッグします。

3. 必要に応じて、ポートレット設定を変更し、変更を保存します。

デフォルトの My webMethods ログインページとしてログインポートレットを追加した OAuth 設定を表示するように、ログインページルールを変更します。ログインページルールの詳細については、358 ページの「ログインページルールの作成」を参照してください。

メモリからのセッションパスワードの消去

デフォルトでは、ユーザがログインすると、ユーザがログアウトするかセッションがタイムアウトするまで、パスワードが保持されます。ただし、ログインの完了直後にメモリからパスワードを消去するように My webMethods Server を設定することもできます。この設定を行うと、My webMethods Server に提示されたすべてのパスワードが消去されます。個別にパスワードを消去することはできません。

メモリからセッションパスワードを消去するには

1. システム管理者として次のように選択します。[**管理**] > [フォルダ] > [システム] > [Manager] > [sessionManager] > [default] > [validate]

2. [**設定 XML**] で、[**編集**] をクリックします。

3. [テキスト編集領域] で、次のテキストを変更します。

   ```
   clearPassword= false
   ```

   次のように変更します。

   ```
   clearPassword= true
   ```

4. [**更新**] をクリックして、ファイルを保存し、エディタを閉じます。

5. [**適用**] をクリックします。

   この設定は明示的に変更しない限り、My webMethods Server を停止して再起動しても残ります。

メモリでのセッションパスワードの保持

メモリにセッションパスワードを保持するには

1. システム管理者として次のように選択します。[**管理**] > [フォルダ] > [システム] > [Manager] > [sessionManager] > [default] > [validate]

2. [**設定 XML**] で、[**編集**] をクリックします。

3. [テキスト編集領域] で、次のテキストを変更します。

```
clearPassword= true
```

次のように変更します。

```
clearPassword= false
```

4. [**更新**] をクリックして、ファイルを保存し、エディタを閉じます。

5. [**適用**] をクリックします。

ユーザ名およびパスワードのオートコンプリートのオン/オフの切り替え

ユーザが My webMethods Server にログオンするときに、ブラウザにユーザ名およびパスワードを記憶させることができます。オートコンプリート機能をオンまたはオフにするように My webMethods Server login ポートレットを設定できます。デフォルトでは、オートコンプリートはオンです。

ユーザ名およびパスワードのオートコンプリートをオンまたはオフにするには

1. システム管理者として、次のように選択します。[**フォルダ**] > [My webMethods アプリケーション] > [webMethods アプリケーションデータ] > [My webMethods ログインページ]。

2. ブラウザのアドレスバーで My webMethods Server URL の末尾に「?layout=details」を付加して、login ポートレットを表示します。

3. login ポートレットの をクリックして [プロパティ] を選択します。

4. 次のいずれかの手順に従います。

   - ユーザ名およびパスワードのオートコンプリートをオフにするには、[**オートコンプリートを許可**] チェックボックスをオフにします。
   - ユーザ名およびパスワードのオートコンプリートをオンにするには、[**オートコンプリートを許可**] チェックボックスをオンにします。チェックボックスは、デフォルトでオンになっています。

5. [**適用**] をクリックします。

失敗したログイン試行回数の制御

My webMethods Server は、ユーザアカウントを一時的にロックするまでに許容される、失敗したログイン試行回数を制御します。[失敗したログインの管理] では、My webMethods ユーザによる失敗したログインの最大試行回数や、My webMethods Server がその後のログイン試行を一時停止する期間をシステム管理者によって設定できます。

[失敗したログインの管理] ページで行う変更は、サーバへのすべてのログインに適用されます。

失敗したログイン試行回数を制御するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [失敗したログインの管理]

2. [**ログイン失敗時の動作の設定**] セクションで、次を指定します。

次の表に、失敗したログインの試行回数を制御するために設定するプロパティを示します。

プロパティ	説明
[ウィンドウの経過時間]	失敗したログイン試行をチェックする期間 (ミリ秒)。デフォルトは *30000* です。
[ウィンドウ内での最大試行回数]	ウィンドウ内で許容されるログイン試行の最大回数。デフォルトは *10* です。
[ロックアウト期間]	ユーザをサーバからロックアウトする期間 (ミリ秒)。デフォルトは *30000* です。

3. [適用] をクリックします。

ログイン IP 範囲の制御

[Lockdown] ポートレットを使用して、My webMethods Server へのログインを許可するユーザの IP 範囲を制御できます。最大 4 種類の IP 範囲を許可できます。このポートレットでは、ログインするユーザのクレデンシャルや認証メカニズムは考慮されません。ログインできない場合、ユーザは選択されたエラーページにリダイレクトされます。このポートレットは、ゲストアクセスのみを許可するサイトや、セキュリティで保護された場所からのアクセスのみを許可するサイトの場合に役立ちます。

重要:
IP 範囲の設定にエラーがあった場合は、ログインして問題を訂正することができないことがあります。問題を訂正するには、My webMethods Server を実行しているマシンに物理的にアクセスする必要があります。

[Lockdown] ポートレットの展開

[Lockdown] ポートレットは、My webMethods Server の初期化時に自動的に展開されるわけではありません。このポートレットは My webMethods Server インストールに含まれていますが、使用する前に展開する必要があります。

[Lockdown] ポートレットを展開するには

1. webMethods インストールディレクトリの以下の場所で、[Lockdown] ポートレットを探します。

\ *Software AG_directory* \MWS\components\extras\security\wm_lockdown.pdp

2. サーバインスタンスの展開ディレクトリに wm_lockdown.pdp ファイルをコピーします。

   \ *Software AG_directory* \MWS\server*serverName*\deploy

   My webMethods Server はこのポートレットの存在を検出して、自動的にポートレットを展開します。

[Lockdown] ポートレットの設定

[Lockdown] ポートレットを My webMethods Server に展開した後、ポートレットの設定を変更できます。

[Lockdown] ポートレットを設定するには

1. システム管理者として次のように選択します。[**管理**] > [フォルダ] > [システム] > [ポートレット] > [管理] > [Portal Lockdown Administration]

2. [エラーページ] プロパティで次のいずれかのボタンをクリックし、ログインに失敗したときにユーザがリダイレクトされるページを選択します。

   - [参照] - [ポータルリソースの選択] ダイアログウィンドウで、ターゲットページを [選択された項目] ボックスに移動し、[選択] をクリックします。
   - [**エイリアスを使用**] - [**エイリアス名**] フィールドにターゲットページのエイリアスを入力します。[**テスト**] をクリックして、そのエイリアスが有効で、エイリアスターゲットが正しいものであるかどうかを確認します。エイリアスが正しければ、[**選択**] をクリックします。

3. [**E-mail Address for Login Notification**] フィールドに、許可対象外の IP アドレスからログインが試みられた場合に My webMethods Server から通知を送信する先の電子メールアドレスを入力します (オプション)。

4. My webMethods Server へのログインを許可するユーザの IP 範囲を最大 4 種類指定します。

   各範囲について、[**Start IP Range**] および [**End IP Range**] フィールドに値を入力します。

5. [**Is Active**] フィールドで、[**True**] オプションを選択します。

6. [**サブミット**] をクリックします。

   ロックダウンが直ちに有効になります。

My webMethods Server での Lockdown ポートレットの無効化

システム管理者として My webMethods Server にログインできる場合は、以下の手順に従って、[Lockdown] ポートレットを無効化できます。

My webMethods Server で [Lockdown] ポートレットを無効化するには

1. システム管理者として次のように選択します。[**管理**] > [フォルダ] > [システム] > [ポートレット] > [管理] > [Portal Lockdown Administration]
2. [**Is Active**] フィールドで、[**False**] オプションを選択します。
3. [**サブミット**] をクリックします。

[Lockdown] ポートレットの手動での無効化

システム管理者として My webMethods Server にログインできない場合は、My webMethods Server がインストールされているマシンに物理的にアクセスして、portal.properties ファイルを変更する必要があります。このファイルの内容はすべてシステムプロパティとして設定され、My webMethods Server の起動後、ポートレットの初期化前に読み込まれます。

[Lockdown] ポートレットを手動で無効化するには

1. My webMethods Server がインストールされているマシン上で、以下のファイルを探します。

   \ *Software AG_directory* \MWS\server*serverName*\config\portal.properties

2. portal.properties ファイルをエディタで開き、ファイルの末尾に次の行を追加します。

   ```
   lockdown.disable=false
   ```

   たとえば、説明コメントを含めて、次のように入力します。

   ```
   #======================
   ```

   ```
   # Lockdown Portlet
   ```

   ```
   lockdown.disable=false
   ```

3. My webMethods Server を再起動します。

   注:
   IP アドレス範囲の問題を訂正した後、portal.properties ファイルからこの行を削除しない限り、[Lockdown] ポートレットを再度アクティブにすることはできません。

グローバル環境変数のパスワードの暗号化

グローバルデフォルト環境変数のカスタムエントリのパスワードフィールドを暗号化できます。

グローバルデフォルト環境変数のパスワードフィールドを暗号化するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [CAF アプリケーションランタイム設定]

2. 結果タイトルバーで、 をクリックします。

3. [**パスワード環境エントリ名**] フィールドで、既存のエントリの後にカンマで区切ってカスタム環境エントリの名前を入力し、[**保存**] をクリックします。

   たとえば、「`wsclient-password,jcr/systemPassword,My_Password`」です。

4. [**グローバルデフォルトの設定**] をクリックします。

5. ツリー表示で、[**環境エントリ**] をクリックし、次に [**新規エントリの追加**] をクリックします。

6. [**名前**] フィールドに、カスタム環境エントリの名前 (「`My_Password`」など) を入力し、[**追加**] をクリックします。

7. 新しく作成された環境エントリにパスワード値を入力し、[**適用**] をクリックします。

環境エントリを正しく設定した場合は、フィールドにパスワードを入力するときにパスワードはマスクされます。また、パスワード値はデータベースに格納される前に暗号化されます。

パスワード値のコンシューマが元の値を取得する必要がある場合は (外部サーバにログインする場合など)、CipherUtil.decrypt(value) Java API を使用して元の値を取得できます。CipherUtil クラスの詳細については、『*webMethods CAF and My webMethods Server Java API Reference*』を参照してください。

コンテキストなりすましの許可

ユーザが別のユーザになりすますのを許可する場合があります。たとえば、デフォルトの Java コンテンツリポジトリ (JCR) クライアントによって、なりすましが使用されます。JCR セッションは、システムユーザ (デフォルトでは SysAdmin) として既知のパスワードを使用して作成され、その後、JCR セッションはなりすましを使用して実際の現在のユーザに切り替えます。このなりすましによって、現在のユーザが JCR リポジトリにログインできます。

SysAdmin ユーザを使用してコンテキストなりすましを実装することはお勧めしません。このことを実行する別の方法は、ユーザになりすまし機能特権セットを持つユーザを作成することです。

ユーザに対してユーザになりすまし機能特権を設定するには、「179 ページの「アクセス特権および機能特権の管理」」を参照してください。

[権限] パネルで、[ユーザになりすまし] は [**機能特権**] > [MWS] > [ユーザになりすまし] にあります。

パスワードの複雑度ポリシーの使用

パスワードの複雑度ポリシーによって、ブルートフォース攻撃に対してユーザパスワードの抵抗力を高める要件が実施されます。パスワードの複雑度クラスを作成し、システムディレクトリサービスで使用するために My webMethods Server に追加できます。この機能は外部ディレクトリサービスには使用できません。

Composite Application Framework (CAF) アプリケーションにパスワードの複雑度ポリシーの Java クラスを含めます。My webMethods Server にアプリケーションを展開すると、サーバによって Java クラスが登録されます。これにより、サーバへのそれ以降の要求で、パスワードの複雑度ポリシーコンポーネントが使用できます。パスワードの複雑度ポリシーの実装に、My webMethods Server の再起動は必要ありません。

Java クラスを作成する前に、Designer で CAF アプリケーションを作成する必要があります。CAF アプリケーションの作成、使用、および展開の詳細については、『*webMethods CAF and OpenUI Development Help*』を参照してください。

カスタムのパスワードの複雑度を Java クラスとして実装するには

1. Designer の既存の CAF アプリケーションで、ISystemPasswordComplexityPolicy インタフェースを実装する Java クラスを作成します。コンポーネントは @Component 注釈による OSGi サービスとして登録されます。

   次の例では、カスタムのパスワードの複雑度ポリシーの実装方法を示します。

```
package caf.war.cafapp1.dir;
import java.util.Map;
import java.util.regex.Pattern;

import org.osgi.service.component.annotations.Activate;
import org.osgi.service.component.annotations.Component;
import org.osgi.service.component.annotations.Deactivate;
import org.osgi.service.component.annotations.Reference;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import com.webmethods.portal.resources.Ui;
import com.webmethods.portal.service.dir.IDirPrincipal;
import com.webmethods.portal.service.dir.ISystemPasswordComplexityPolicy;
import com.webmethods.portal.service.global.IGlobalProvider;
import com.webmethods.portal.service.global.IMessageInfo;
import com.webmethods.portal.system.init.InitializationException;

/**
 * Custom implementation of password complexity policy
 */
@Component (
   service = {
       ISystemPasswordComplexityPolicy.class
   },
   property = {
       "name:String=CustomSystemPasswordComplexityPolicy1"
   },
   immediate = true
```

```
)
public class CustomSystemPasswordComplexityPolicy1 implements
        ISystemPasswordComplexityPolicy {
    private Logger logger = LoggerFactory.getLogger(getClass());

    //references to other services
    private IGlobalProvider gp = null;

    /**
     * Reference injection of the other OSGi service
     */
    @Reference
    protected void bindGlobalProvider(IGlobalProvider globalProvider) {
        if (logger.isTraceEnabled()) {
            logger.trace("Bound IGlobalProvider");
        }

        this.gp = globalProvider;
    }

    /**
     * Activation of OSGi declaritive service
     * @param config the configuration properties of the OSGi service
     * @throws InitializationException if activation fails
     */
    @Activate
    protected void activate(Map<String,Object> config) throws
InitializationException {
        if (logger.isInfoEnabled()) {
            logger.info("Activating service");
        }
    }

    /**
     * Deactivation of OSGi declaritive service
     */
    @Deactivate
    protected void deactivate() {
        if (logger.isInfoEnabled()) {
            logger.info("Deactivating service");
        }
    }

    /**
     * Check the candidate password to make sure the value satisfies the
complexity
     * requirements.If the password is not complex enough, this method
should throw
     * an {@link InvalidPasswordException} with the reason.If no exception
is thrown
     * the password is valid.
     *
     * @param candidatePassword the password to check
     * @throws InvalidPasswordException if the password is not valid
     */
    @Override
    public void checkPasswordForNewUser(String candidatePassword)
            throws InvalidPasswordException {
        if (candidatePassword == null || candidatePassword.length() < 6) {
            IMessageInfo messageInfo = gp.getMessageInfo(Ui.class,
```

```
                    "password.too.short", null);
            String msg = messageInfo.getLocalizedMessage(false);
            throw new InvalidPasswordException(msg);
        }

        boolean hasUpper =
!candidatePassword.toLowerCase().equals(candidatePassword);
        boolean hasLower =
!candidatePassword.toUpperCase().equals(candidatePassword);
        if (!(hasUpper && hasLower)) {
            IMessageInfo messageInfo = gp.getMessageInfo(Ui.class,
                    "password.mix.case", null);
            String msg = messageInfo.getLocalizedMessage(false);
            throw new InvalidPasswordException(msg);
        }

        boolean hasNumberOrSpecialChar = Pattern.matches(".*[\\W\\d]+.*$",
candidatePassword);
        if (!hasNumberOrSpecialChar) {
            IMessageInfo messageInfo = gp.getMessageInfo(Ui.class,
                    "password.special.char", null);
            String msg = messageInfo.getLocalizedMessage(false);
            throw new InvalidPasswordException(msg);
        }
    }

    /**
     * Check the candidate password to make sure the value satisfies the
complexity
     * requirements.If the password is not complex enough, this method
should throw
     * an {@link InvalidPasswordException} with the reason.If no exception
is thrown
     * the password is valid.
     *
     * @param user the user whose password is being checked.
     * @param candidatePassword the password to check
     * @throws InvalidPasswordException if the password is not valid
     */
    @Override
    public void checkPasswordForExistingUser(IDirPrincipal user,
            String candidatePassword) throws InvalidPasswordException {
        checkPasswordForNewUser(candidatePassword);
    }

    /**
     * Return how long a password is valid (in milliseconds) before it
expires.The user
     * will not be able to login after this time duration has expired
and the user must be
     * reset by an administrator or a custom reset password page.
     *
     * @param user the user to get the expiration value for.
     * @return return -1 for no password expiration, or the duration (in ms)
     */
    @Override
    public long getPasswordExpirationDuration(IDirPrincipal user) {
        try {
            if ("sysadmin".equals(user.getName())) {
                return -1; //don't expire the sysadmin password.
```

```
            }
        } catch (Exception e) {
            logger.warn(e.getMessage(), e);
        }
        return 1 * 60 * 60 * 1000; //1 hour
    }

    /**
     * Returns a description of the expected pattern a password must have.
This text
     * is displayed on the create user and update user pages of the UI.
     *
     * @return password descriptive text.
     */
    @Override
    public String getPasswordPatternText() {
        IMessageInfo messageInfo = gp.getMessageInfo(Ui.class,
                "password.complexity.description", null);
        return messageInfo.getLocalizedMessage(false);
    }
  }
```

2. アプリケーションを My webMethods Server に展開します。

3. My webMethods Server ディレクトリサービスページを開くには、http://*hostname*:*port*/webm.apps.config.directory.service.admin に移動します。*hostname* は My webMethods Server のホスト名です。*port* は My webMethods Server のポート番号です。

4. [ディレクトリサービスの一覧表示] タブで、[system] ディレクトリサービスをクリックします。

5. [パスワードの複雑度クラス] フィールドに、CAF アプリケーションで作成した Java クラスの名前を選択し、[適用] をクリックします。

6. Java クラスをデバッグし、テストします。

7. 必要に応じて Java クラスをさらに変更し、クラスが適切に動作するまで手順 2 と 6 を繰り返します。

応答ヘッダールールの使用

My webMethods Server では、要求メッセージの受信と解釈の後で送信される HTTP 応答メッセージに対して、ルールを作成して管理することができます。

応答ヘッダールールの表示

応答ヘッダールールを表示するには

1. システム管理者として、[**管理用フォルダ**] > [Administration Dashboard] > [ユーザインタフェース] > [応答ヘッダールールの管理] を選択します。

2. 選択されていない場合は、[**ルールの表示**] タブをクリックします。

使用可能なルールが [**ルール名**] リストに表示されます。ルールが定義されていない場合、リストは空になります。各ルールに対して、次の情報が表示されます。

- ルール名。これはクリック可能なリンクになっており、リンクをクリックすると [**ルールの変更**] が開き、ルールの条件を表示したり、ルールを変更したりできます。
- ルールの作成者が入力したルールに関する説明。
- ルールが現在有効になっているかどうか。

応答ヘッダールールの作成

応答ヘッダールールを作成するには

1. システム管理者として、[**管理用フォルダ**] > [Administration Dashboard] > [ユーザインタフェース] > [応答ヘッダールールの管理] を選択します。

2. [**新規ルールの作成**] タブをクリックします。

3. 以下の手順に従います。

   - [**名前**] フィールドに、ルールの名前を入力します。
   - ルールの説明を [**説明**] フィールドに入力します。
   - 作成時にルールを有効にするには、[**有効**] チェックボックスをオンにします。
   - 作成時にルールを無効にするには、[**有効**] チェックボックスをオフにします。

4. [**条件**] フィールドで、使用可能な条件要素のボタンを使用して、ルールの条件式を定義します。次の条件要素を使用できます。

   - **[現在のユーザ]**
   - **[グループ/役割メンバーシップ]**
   - **[ユーザの属性]**
   - **[要求]**
   - **[親リソース]**
   - **[現在のリソースタイプ]**
   - **[リソースプロパティ]**
   - **[演算子の追加]**

使用するボタンをクリックして、次に開かれたダイアログボックスを使用して、条件式に追加する値を選択します。次に例を示します。

```
portalResource isDescendant ("webm.apps.data.page.login") ||
portalResource isDescendant ("portlet.login")
```

また式を直接入力したり、 作成後に式を変更したりすることもできます。

5. [**結果**] フィールドの横にある [**追加**] をクリックして、応答ヘッダーのキーフィールド名と値を指定します。必要に応じて、追加のキー/値のペアを追加できます。キー/値のペアを削除するには、リストでペアを選択して [**削除**] をクリックします。

6. [**ルールの作成**] をクリックします。

応答ヘッダールールの変更

応答ヘッダールールを変更するには

1. システム管理者として、[**管理用フォルダ**] > [Administration Dashboard] > [ユーザインタフェース] > [応答ヘッダールールの管理] を選択します。

2. 以下のいずれかを行います。

   - [**ルール名**] フィールドでルール名をクリックします。
   - 使用するルールの をクリックして、[ルールの変更] を選択します。

3. [**ルールの変更**] タブで、必要に応じてルールを変更します。

   - [**条件**] フィールドで、使用可能な条件要素のボタンを使用して、ルールに条件式を追加するか、式を直接入力して変更します。

4. [**結果**] フィールドで応答ヘッダーのキーフィールド名と値を選択し、以下のいずれかの手順に従います。

   - [**追加**] をクリックして、追加のキー/値のペアを指定します。
   - [**変更**] をクリックして、既存のキー/値のペアを変更します。
   - [**削除**] をクリックして、キー/値のペアを削除します。

5. [**ルールの更新**] をクリックします。

応答ヘッダールールのコピー

[応答ヘッダールールの管理] ページで、ルールをコピーして新しいルールを作成できます。ルールをコピーして、新しいルールの名前と説明を入力すると、ルールの条件と結果の情報が元のルールから新しいルールにコピーされます。

注:
ルールをコピーするとき、元のルールの [**有効**] チェックボックスの設定も新しいルールにコピーされます。元のルールの [**有効**] チェックボックスがオンになっている場合、新たにコピーしたルールは作成直後に有効になります。

応答ヘッダールールをコピーするには

1. システム管理者として、[**管理用フォルダ**] > [Administration Dashboard] > [ユーザインタフェース] > [応答ヘッダールールの管理] を選択します。
2. 使用するルールの をクリックして、[ルールのコピー] を選択します。
3. [**ルールのコピー**] タブで、新規ルールの名前と説明を入力します。
4. [**ルールのコピー**] をクリックします。

応答ヘッダールールの削除

応答ヘッダールールを削除するには

1. システム管理者として、[**管理用フォルダ**] > [Administration Dashboard] > [ユーザインタフェース] > [応答ヘッダールールの管理] を選択します。
2. 使用するルールの をクリックして、[ルールの削除] を選択します。
3. 確認ダイアログボックスで [**OK**] をクリックしてルールを削除します。

応答ヘッダールールの評価順序の変更

応答ヘッダールールの評価順序を変更するには

1. システム管理者として、[**管理用フォルダ**] > [Administration Dashboard] > [ユーザインタフェース] > [応答ヘッダールールの管理] を選択します。
2. [**ルール評価順序の変更**] タブをクリックします。
3. [**評価順序**] リストでルール名を選択して、右側にある矢印ボタンを使用して、評価順序の中で選択したルールを上下に移動します。
4. [**ルールの更新**] をクリックします。

デフォルトの Internet Explorer 互換性設定の変更

My webMethods Server のデフォルトでは、Internet Explorer の互換性モードは IE8 に設定されています。My webMethods Server の Internet Explorer 互換性モードの詳細については、「306 ページの「デフォルトの応答ヘッダールールについて」」を参照してください。

Internet Explorer ドキュメント互換性設定を変更するには

1. システム管理者として、[**管理用フォルダ**] > [Administration Dashboard] > [ユーザインタフェース] > [応答ヘッダールールの管理] を選択します。
2. [**IE - parameter for compatibility mode**] ルールを見つけて、以下のいずれかの手順に従います。
   - [**ルール名**] フィールドでルール名をクリックします。
   - 使用するルールの をクリックして、[ルールの変更] を選択します。
3. [**結果**] フィールドの横にある [**更新**] をクリックし、[**値**] フィールドを編集して、[**適用**] をクリックします。

   デフォルト値は `IE-8` です。
4. [**ルールの更新**] をクリックします。

デフォルトの応答ヘッダールールについて

次の表に、一般的な My webMethods Server インストールで使用可能な応答ヘッダールールを示します。

ルール名	説明
[ログインページの同一起点フレーム処理以外を拒否]	デフォルトでは有効になっています。このルールは、X-Frame-Options HTTP 応答ヘッダーを実装することで、ログインページでのクロスサイトスクリプティング攻撃やクリックジャッキング攻撃に対して、防御機能を提供します。コンテンツが他のサイトに埋め込まれないようにするために、このヘッダーは、<frame> または <iframe> でのページのレンダリングをブラウザに許可するかどうかを指定します。キー/値のペアは次のようになります。 `X-Frame-Options SAMEORIGIN` ページは、ページ自体と同じ作成元のフレームに限り、表示することができます。
[ログインページですべてのフレーム処理を拒否]	デフォルトでは無効になっています。これは、クロスサイトスクリプティング攻撃やクリックジャッキング攻撃に対応するさらに

ルール名	説明
	厳格なログインページルールです。キー/値のペアは次のようになります。 `X-Frame-Options DENY` この場合、サイトでの動作に関係なく、ページはフレームに表示できません。
[IE-互換モード用のパラメータ]	デフォルトでは有効になっています。この設定では、HTML ページのレンダリングに、Internet Explorer の標準的なドキュメントタイプを設定します。デフォルト値は IE8 です。

X-Frame-Options ヘッダー応答の基本的なサポートは、以下に示すバージョン以降のブラウザで使用できます。

- Chrome 4.1.249.1042
- Firefox 3.6.9
- Gecko 1.9.2.9
- Internet Explorer 8.0
- Opera 10.5
- Safari 4.0

コンテンツセキュリティポリシーの設定

デフォルトのコンテンツセキュリティポリシーヘッダーを変更するカスタムヘッダー応答ルールを特定のサーバリソースに対して指定するか、追加の JVM パラメータを使用して My webMethods Server のポリシーをグローバルに設定できます。次のカスタム JVM パラメータを使用して、デフォルトのコンテンツセキュリティポリシーを有効または無効にするか、信用できるコンテンツのソースを設定します。

- `com.webmethods.content.security.disabled` - デフォルト値は `false` です。コンテンツセキュリティポリシーを無効にするには、`true` に設定します。
- `com.webmethods.content.security.hosts` - このパラメータを使用して、追加で許可されるホストを指定します。複数の場合は間隔で区切ります。

My webMethods Server の custom_wrapper.conf ファイルに JVM ファイルを追加する方法の詳細については、97 ページの「My webMethods Server の JVM 設定」を参照してください。

応答ヘッダールールの使用の詳細については、302 ページの「応答ヘッダールールの使用」を参照してください。

16 分析、レポート、およびトラブルシューティング

- 分析、レポート、およびトラブルシューティングについて 310
- サーバのログの制御 310
- ログメッセージの表示 322
- セキュリティ監査ログの管理 323
- リアルタイムユーザアクティビティの監視 323
- サーバイベントに関するデータの収集 324
- データベース変更に関するデータの収集 326
- My webMethods Server 診断ツール 327

分析、レポート、およびトラブルシューティングについて

My webMethods Server には、管理者がサーバ展開の分析、レポート、管理、および保守を行うための多数のツールが用意されています。

サーバのログの制御

My webMethods Server は、実行時に、Web アプリケーションおよびポートレットアプリケーションの、さまざまなカテゴリのログ情報を収集します。ログ情報は、ロガーおよび出力の 2 段階で収集されます。

- **ロガー** この段階では、カテゴリ、Web アプリケーションおよびポートレットアプリケーションごとに、収集するメッセージのレベルを定義できます。
- **出力** この段階では、My webMethods Server がコンソールおよびログファイルに書き込むメッセージのレベルを決定します。

各段階は個別に制御できます。ロガーしきい値は、出力しきい値より優先されます。ログメッセージがロガーしきい値に達していないので、あるカテゴリまたはアプリケーションのメッセージが収集されず、My webMethods Server がログメッセージを破棄した場合、そのメッセージは出力 (つまり、コンソールまたはログファイル) に書き込まれません。

[ログ設定] ページを使用すると、サーバのログを制御できます。詳細については、310 ページの「ログのしきい値について」と312 ページの「ロガーしきい値と出力しきい値の設定」を参照してください。

My webMethods Server がログメッセージを出力ログファイルに書き込むと、ファイルのサイズが大きくなります。My webMethods Server は、定期的に新しいファイルセットにロールオーバするため、特定の日時の検索や、必要に応じた古いログファイルの破棄が簡単になります。出力ログファイル、My webMethods Server がファイルをロールオーバする頻度、およびロールオーバの制御方法の詳細については、314 ページの「ログファイルのロールオーバ期間について」および317 ページの「ログファイルのロールオーバ期間の変更」を参照してください。

ログのしきい値について

My webMethods Server が収集するログメッセージを制御するには、ログのしきい値を定義します。各ログメッセージには、あるログレベルが割り当てられます。しきい値は、My webMethods Server が収集するメッセージのログレベルを示しています。My webMethods Server は、指定したレベル以上のレベルを持つメッセージをログに記録します。そのため、しきい値を設定すると、ログファイルが増大することを制限できます。収集する情報を増やす場合はログレベルを低く設定し、収集する情報を減らす場合はログレベルを高く設定します。

次の表では、低レベルから高レベルの順で、レベルについて説明しています。

ログレベル	説明
TRACE	すべてのレベルのメッセージをログに記録する場合は、しきい値をこのレベルに設定します。このレベルでは最も詳細な情報が得られますが、ログファイルのサイズは急速に大きくなります。
DEBUG	サーバは、サーバイベント内の複数のポイントで、DEBUG メッセージを発行します。DEBUG レベル以上のメッセージ (INFO、WARN など) をすべて収集する場合は、しきい値をこのレベルに設定します。このレベルは問題のデバッグに役立ちますが、ログファイルのサイズは急速に大きくなります。
INFO	サーバは、サーバイベントが発生したことを示す INFO メッセージを発行します。INFO レベル以上のメッセージをすべて収集する場合は、しきい値をこのレベルに設定します。
WARN	サーバは、重大ではないエラーを警告する WARN メッセージを発行します。WARN、ERROR、および FATAL のメッセージを収集する場合は、しきい値をこのレベルに設定します。
ERROR	サーバは、致命的でないエラーが発生した場合に ERROR メッセージを発行します。ERROR および FATAL のメッセージを収集する場合は、しきい値をこのレベルに設定します。
FATAL	サーバは、致命的なエラーが発生した場合に ERROR メッセージを発行します。FATAL メッセージのみを収集する場合は、しきい値をこのレベルに設定します。

しきい値を定義するには、上記の表に示したレベルのいずれかを割り当てます。

- **カテゴリ、Web アプリケーションまたはポートレットアプリケーションのロガーしきい値にログレベルを割り当てる** My webMethods Server がそのカテゴリまたはアプリケーションで収集するメッセージを制御します。My webMethods Server がメッセージを収集できるカテゴリおよびアプリケーションは、[ログ設定] ページで表示できます。
- **出力しきい値にログレベルを割り当てる** My webMethods Server がコンソールまたは以下のログファイルのいずれかに書き込むメッセージを制御します。
  - **_full_.log** すべてのカテゴリ、Web アプリケーションおよびポートレットアプリケーションの、すべてのレベルのメッセージを記録できます。
  - **_problems_.log** すべてのカテゴリ、Web アプリケーションおよびポートレットアプリケーションのメッセージを記録できます。ただし、レベルは WARN メッセージ以上のメッセージ (WARN、ERROR または FATAL メッセージ) に制限されます。
  - **_errors_.xm_** すべてのカテゴリ、Web アプリケーションおよびポートレットアプリケーションのメッセージを含む XML フラグメントです。デフォルトでは、FATAL メッセージのみが記録されます。

    注:
    ルート XML 要素を持つエラーログに含まれる XML フラグメントを折り返して、整形式 XML を生成できます。

ロガーしきい値は、出力しきい値より優先されます。サーバは、ログメッセージを収集すると、最初は指定されたロガーにメッセージを送信します。メッセージがしきい値に達していない場合、メッセージは破棄されます。ただし、メッセージのログレベルがロガーしきい値に達しているか超えている場合、サーバはメッセージをログ出力に送信します。メッセージのログレベルが出力しきい値に達していない場合、メッセージは破棄されます。最後に、メッセージが出力しきい値に達しているか超えている場合、サーバはメッセージを出力に書き込みます。つまり、カテゴリまたはアプリケーションに設定されたしきい値 (ロガーしきい値) に達していないメッセージは破棄され、出力に書き込まれません。

ロガーしきい値と出力しきい値の設定

[ログ設定] ページで、ロガーしきい値と出力しきい値の両方を設定できます。

注:
[ログ設定] ページでしきい値を更新すると、その設定は再度更新されるまで永続します。サーバがシャットダウンされるまで継続するデバッグ設定を My webMethods Server で一時的に使用する場合は、-d起動オプションを使用してサーバを起動します。詳細については、312 ページの「デバッグしきい値の一時的な設定」を参照してください。

ロガーしきい値または出力しきい値を設定するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [分析] > [ログ設定]

2. 変更する各カテゴリ、Web アプリケーションおよびポートレットアプリケーションについて、[**ロガーしきい値**] 列で、受け入れるメッセージの最低レベルのログレベルを選択します。

3. 変更する各出力しきい値について、[**出力しきい値**] リストで、受け入れるメッセージの最低レベルのログレベルを選択します。

   ヒント:
   すべてのロガーしきい値または出力しきい値を同じログレベルに設定するには、[ロガーしきい値] ラベルまたは [出力しきい値] ラベルの右にある をクリックし、使用するログレベルを選択します。

4. [**適用**] をクリックします。

デバッグしきい値の一時的な設定

デフォルトのサーバの設定では、Java システムプロパティ (logj4.default.log.level) を使用して、いくつかのカテゴリでログのしきい値が WARN に設定されています。-d 起動オプションを指定してサーバを起動すると、logj4.default.log.level 変数が DEBUG に設定されます。そのため、カテゴリのメッセージは、サーバがシャットダウンされるまで DEBUG しきい値で収集されます。サーバをデバッグモードで実行するには、この -d起動オプションを使用します。この方法によって、多くのカテゴリで一時的にログレベルが高くなり、トラブルシューティングを簡単に実行できるようになります。

ログ設定の次のセクションは、-d 起動オプションによって影響を受けるカテゴリを示します。

```
# default level controled by -d=DEBUG otherwise =WARN
log4j.category.Framework=${log4j.default.log.level}
log4j.category.frameworkInit=${log4j.default.log.level}
log4j.category.dataAccess=${log4j.default.log.level}
log4j.category.jsp=${log4j.default.log.level}
log4j.category.jsf=${log4j.default.log.level}
log4j.category.directory=${log4j.default.log.level}
log4j.category.portlet=${log4j.default.log.level}
log4j.category.classManager=${log4j.default.log.level}
log4j.category.taglibs=${log4j.default.log.level}
log4j.category.mail=${log4j.default.log.level}
log4j.category.search=${log4j.default.log.level}
log4j.category.messaging=${log4j.default.log.level}
log4j.category.notifications=${log4j.default.log.level}
log4j.category.schedule=${log4j.default.log.level}
log4j.category.version=${log4j.default.log.level}
log4j.category.task=${log4j.default.log.level}
log4j.category.webservice=${log4j.default.log.level}
log4j.category.wsclient=${log4j.default.log.level}
```

-d 起動オプションを使用してサーバを起動 する方法の詳細については、72 ページの「My webMethods Server の基本のコマンドライン構文」を参照してください。

ファイルへのしきい値設定のエクスポート

ロガーしきい値および出力しきい値の設定は、log4.override.properties ファイルにエクスポートできます。log4.override.properties ファイルにエクスポートした設定は、後でこのファイルから My webMethods Server データベースにインポートできます。

しきい値設定をエクスポートするのは、バックアップコピーを保存する場合や、同じクラスタ内にない別の My webMethods Server インスタンスで同じ設定を使用する場合です。

しきい値設定をエクスポートするには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [分析] > [ログ設定]
2. [**エクスポート**] をクリックします。
3. 保存前にテキストエディタでファイルを開いて表示する場合は、以下の手順に従います。
   a. [**プログラムから開く**] オプションを選択します。
   b. ファイルの表示に使用するテキストエディタを選択します。
   c. [**OK**] をクリックします。

      My webMethods Server によってデータベースからファイルがダウンロードされ、選択したエディタでファイルが開きます。

d. テキストエディタの保存機能を使用して、ファイルを保存します。

4. ファイルを開かずに保存する場合は、以下の手順に従います。

a. [**ファイルの保存**] を選択します。

b. [**OK**] をクリックします。

My webMethods Server によって、しきい値設定が log4.override.properties ファイルにダウンロードされ、デスクトップ上にファイルが配置されます。

ファイルからのしきい値設定のインポート

ロガーしきい値および出力しきい値の設定を log4.override.properties ファイルにエクスポート済みの場合、My webMethods Server インスタンスに設定をインポートするには、以下の手順に従います。

しきい値設定をインポートするには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [分析] > [ログ設定]

2. [**インポート**] をクリックします。

3. [**参照**] をクリックし、インポートする設定を含む log4.override.properties ファイルの場所まで移動して、[**開く**] をクリックします。

4. [**インポートモード**] フィールドで、設定をマージするか、設定を完全に置き換えるかを選択します。

- log4.override.properties ファイルの設定を、My webMethods Server データベースにある現在のしきい値設定とマージする場合は、[**既存の設定とマージする**] を選択します。
- My webMethods Server データベースにあるしきい値設定を、log4.override.properties ファイルの設定で完全に置き換える場合は、[**既存の設定と置き換える**] を選択します。

5. [**設定のインポート**] をクリックします。

ログファイルのロールオーバ期間について

ロギングプロセスは、日付やファイルサイズに基づいて、定期的に新しいファイルセットにロールオーバします。My webMethods Server が日付に基づいてファイルをロールオーバするときは、古いログファイルの名前に日付を追加してファイル名を変更するので、古いログデータを参照する必要がある場合に、特定の日付のデータを簡単に検索できます。My webMethods Server がサイズに基づいてロールオーバするときは、指定されたバックアップファイル数が維持され、古いデータは廃棄されます。

My webMethods Server のログファイルは、次のディレクトリにあります。

Software AG_directory \MWS\server*serverName*\logs

このディレクトリからサーバログを手動で削除することができます。または、ログアペンダのタイムスタンプおよび設定したロールオーバ期間に基づいて、ログを削除するカスタムロジックを開発することもできます。

次の表に、My webMethods Server が作成するログファイルと、ログのデフォルトのロールオーバ期間を示します。

ログファイル	デフォルトの内容	デフォルトのロールオーバ期間
full.log	すべてのカテゴリおよびすべての有効しきい値のログメッセージ (TRACE、DEBUG、INFO、WARN、ERROR、FATAL など)	毎日午前 0 時 ログがロールオーバされるとき、過去のログメッセージは、そのログに含まれるログ情報の日付を反映する名前のファイルにロールオーバされます。 _full_.yyyy-MM-dd.log My webMethods Server がこのログをロールオーバする頻度をカスタマイズできます。
problems.log	すべてのカテゴリの WARN、ERROR、FATAL ログメッセージ	毎日午前 0 時 ログがロールオーバされるとき、過去のログメッセージは、そのログに含まれるログ情報の日付を反映する名前のファイルにロールオーバされます。 _problems_.yyyy-MM-dd.log My webMethods Server がこのログをロールオーバする頻度をカスタマイズできます。
install.log	インストールの DEBUG レベルログメッセージ	ログサイズが 100 MB に達した場合 ログがロールオーバされると、古いログメッセージは、次の命名規則 (*N* は番号) のバックアップファイルにロールオーバされます。 install.*N*.log たとえば、初めてログがロールオーバされたとき、バックアップログの名前は install.1.log になります。デフォルトでは、My webMethods Server はバックアップを 3 つまで保持します。

ログファイル	デフォルトの内容	デフォルトのロールオーバ期間
		My webMethods Server がこのファイルをロールオーバする前のファイルの最大サイズ制限、およびサーバが保持するバックアップファイル数をカスタマイズできます。
caf.log	CAF アプリケーションログメッセージ	毎日午前 0 時 ログがロールオーバされるとき、過去のログメッセージは、そのログに含まれるログ情報の日付を反映する名前のファイルにロールオーバされます。 caf.yyyy-MM-dd.log My webMethods Server がこのログをロールオーバする頻度はカスタマイズできません。
schema.log	x- タイプオブジェクトの作成、変更、または削除の DDL ステートメント実行のログメッセージ	ログサイズが 5 MB に達した場合 ログがロールオーバされると、古いログメッセージは、次の命名規則 (*N* は番号) のバックアップファイルにロールオーバされます。 schema.*N*.log たとえば、初めてログがロールオーバされたとき、バックアップログの名前は schema.1.log になります。デフォルトでは、My webMethods Server はバックアップを 10 個まで保持します。 My webMethods Server がこのファイルをロールオーバする前のファイルの最大サイズ制限、およびサーバが保持するバックアップファイル数をカスタマイズできます。
errors.xm_	すべてのカテゴリの FATAL ログメッセージの XML フラグメント ルート XML 要素を持つエラーログに含まれる XML フラグメントを折り返して、整形式 XML を生成できます。	このログはロールオーバされません。

ロールオーバ期間をカスタマイズする方法の詳細については、317 ページの「ログファイルのロールオーバ期間の変更」を参照してください。

ログファイルのロールオーバ期間の変更

My webMethods Server は、My webMethods Server データベース内に、ログファイルをロールオーバする方法の設定を保持します。そのため、クラスタ内でサーバを実行している場合は、すべてのサーバインスタンスが同じロールオーバ設定を使用します。ロールオーバ設定を更新するには、My webMethods Server でローカルファイル (名前は log4j.init.properties) に設定をダウンロードし、ダウンロードしたファイルのロールオーバ期間を変更して、変更をデータベースに再度アップロードします。

My webMethods Server がログファイルをロールオーバする場合は、次の項目に基づいてロールオーバを行います。

- **日時 (_full_.log ファイルおよび _problems_.log ファイルで使用)**

  ロールオーバ設定は、アペンダタイプ org.apache.log4j.DailyRollingFileAppender によって制御されます。ロールオーバ期間を更新する場合は、アペンダに新しい DatePattern を指定します。次の表に、ログのロールオーバの設定で使用可能な DatePattern を示します。

日付パターン	ロールオーバのタイミング
`'.'yyyy-MM`	各月の最初。 たとえば、_full_.log の場合、DatePattern を「`'.'yyyy-MM`」に設定すると、2010 年 1 月 31 日の午前 0 時に、My webMethods Server がログデータをファイル _full_.2010-01.log にコピーします。My webMethods Server は、次の月に _full_.log ファイルがロールオーバされるまで、2 月分のメッセージをこのファイルに記録します。
`'.'yyyy-ww`	各週の最初の日。週の最初の日はロケールによって異なります。 たとえば、週の最初の日が日曜日で、_problems_.log の場合に DatePattern を「`'.'yyyy-ww`」に設定するとします。2010 年 3 月 15 日 (土曜日) の午前 0 時に、My webMethods Server はこの年の 19 週目のログデータをファイル _problems_2010-19 にコピーします。My webMethods Server は、次の週に _problems_.log ファイルがロールオーバされるまで、2010 年の 20 週目のメッセージをこのファイルに記録します。
`'.'yyyy-MM-dd`	毎日午前 0 時。 たとえば、_full_.log の場合、DatePattern を「`'.'yyyy-MM-dd`」に設定すると、2010 年 2 月 22 日の午前 0 時に、My webMethods Server がログデータをファイル _full_2010-02-22 にコピーします。My webMethods Server は、次の日に _full_.log ファイルがロールオーバされるまで、2010 年 2 月 23 日のメッセージをこのファイルに記録します。
`'.'yyyy-MM-dd-a`	毎日 2 回 (午後 12 時および午前 0 時)。

日付パターン	ロールオーバのタイミング
	たとえば、_problems_.log の場合、DatePattern を「'.'yyyy-MM-dd-a」に設定すると、2010 年 2 月 22 日の午後 12 時に、My webMethods Server がログデータをファイル _problems_2010-02-22-A.M にコピーします。My webMethods Server は、午前 0 時に _problems_.log ファイルが _problems_2010-02-22-P.M ファイルにロールオーバされるまで、2010 年 2 月 22 日の午後のメッセージをこのファイルに記録します。その後、My webMethods Server は 2010 年 2 月 23 日のメッセージを _problems_.log ファイルに記録します。
'.'yyyy-MM-dd-HH	毎日 1 時間ごと。 たとえば、_full_.log の場合、DatePattern を「'.'yyyy-MM-dd-HH」に設定すると、2010 年 2 月 22 日の午前 11 時頃に、My webMethods Server が 10 時台のログデータをファイル _full_2010-02-22-10 にコピーします。My webMethods Server は、次の時間の始めに _full_.log ファイルがロールオーバされるまで、11 時台のメッセージをこのファイルに記録します。
'.'yyyy-MM-dd-HH-mm	毎日 1 分ごと。 たとえば、_full_.log の場合、DatePattern を「'.'yyyy-MM-dd-HH-mm」に設定すると、2010 年 2 月 22 日の午前 11 時 46 分頃に、My webMethods Server がログデータをファイル _full_2010-02-22-11-45 にコピーします。My webMethods Server は、1 分後に _full_.log ファイルがロールオーバされるまで、次の 1 分のメッセージをこのファイルに記録します。

- **ファイルサイズ (install.log ファイルで使用)**

  ロールオーバ設定は、org.apache.log4j.RollingFileAppender によって制御されます。install.log ファイルで収集されたメッセージのログレベル、ロールオーバ前のログファイルの最大サイズ、および My webMethods Server が保持するバックアップログファイルの数を更新できます。

注:
次の手順に、標準で定義されているデフォルトアペンダ (org.apache.log4j.DailyRollingFileAppender および org.apache.log4j.RollingFileAppender) のパラメータを更新して、ロールオーバ期間を変更する方法を示します。ただし、必要であれば、ニーズを満たす別のアペンダを使用するように設定を更新できます。たとえば、サイズに基づいてロールオーバする場合は _problems_.log に RollingFileAppender を使用し、ログをロールオーバしない場合は FileAppender を使用するなどの変更ができます。log4j ライブラリがサポートする任意のアペンダを使用できます。詳細については、http://logging.apache.org/log4j/1.2/index.html を参照してください。

ログファイルのロールオーバ期間を変更するには

1. My webMethods Server データベースから log4j.init.properties ファイルをダウンロードします。

   a. コマンドラインプロンプトで、ディレクトリを変更して、サーバの bin ディレクトリに移動します。

      Software AG_directory \MWS\bin

   b. データベースから log4j.init.properties ファイルを抽出するには、次のコマンドを入力します。

      ```
      mws -s serverName getconfig log4j.init.properties
      ```

2. log4j.init.properties ファイルを編集します。

   a. getconfig コマンドによって log4j.init.properties ファイルが配置されている、次のディレクトリに移動します。

      Software AG_directory \MWS\server*serverName*\config

   b. テキストエディタで log4j.init.properties ファイルを開きます。

3. _full_.log ファイルのロールオーバ設定を変更するには、以下の手順に従います。

   a. ファイルの次の部分を探します。

      ```
      log4j.appender.rootFile=org.apache.log4j.DailyRollingFileAppender
      ```

      ```
      log4j.appender.rootFile.DatePattern='.'yyyy-MM-dd
      ```

      ```
      log4j.appender.rootFile.File=${log4j.logging.dir}/_full_.log
      ```

   b. 日付パターンを更新して、_full_.log ファイルがロールオーバされるタイミングを定義します。

4. _problems_.log ファイルのロールオーバ設定を変更するには、以下の手順に従います。

   a. ファイルの次の部分を探します。

      ```
      log4j.appender.rootErrorsFile=org.apache.log4j.
          DailyRollingFileAppender
      ```

      ```
      log4j.appender.rootErrorsFile.DatePattern='.'yyyy-MM-dd
      ```

      ```
      log4j.appender.rootErrorsFile.File=${log4j.logging.dir}/
          _problems_.log
      ```

   b. 日付パターンを更新して、_problems_.log ファイルがロールオーバされるタイミングを定義します。

5. install.log ファイルのロールオーバ設定を変更するには、以下の手順に従います。

   a. ファイルの次の部分を探します。

```
log4j.appender.installFile=org.apache.log4j.RollingFileAppender

log4j.appender.installFile.threshold=DEBUG

log4j.appender.installFile.MaxFileSize=100MB

log4j.appender.installFile.MaxBackupIndex=3

log4j.appender.installFile.File=${log4j.logging.dir}/install.log
```

   b. ファイルのサイズ制限を変更するには、`log4j.appender.installFile.MaxFileSize` 行の値を更新して、別の最大ファイルサイズを指定します。

   c. My webMethods Server で保持するバックアップ数を変更するには、`log4j.appender.installFile.MaxBackupIndex` 行の値を更新します。

   install.log のしきい値レベルを変更する `log4j.appender.installFile.threshold` 行は更新しないことをお勧めします。インストールプロセスでは INFO メッセージが発行されるため、しきい値を高くするとメッセージがログに記録されなくなります。

6. schema.log ファイルのロールオーバ設定を変更するには、以下の手順に従います。

   a. ファイルの次の部分を探します。

```
log4j.appender.DDLSchemaFileAppender=org.apache.log4j.
    RollingFileAppender

log4j.appender.DDLSchemaFileAppender.threshold=INFO

log4j.appender.DDLSchemaFileAppender.MaxFileSize=5MB

log4j.appender.DDLSchemaFileAppender.MaxBackupIndex=10

log4j.appender.DDLSchemaFileAppender.File=${log4j.logging.dir}
    /schema.log

log4j.appender.DDLSchemaFileAppender.layout=org.apache.log4j.
    PatternLayout

log4j.appender.DDLSchemaFileAppender.layout.ConversionPattern=
    ${log4j.mes
sage.pattern}
```

   b. ファイルのサイズ制限を変更するには、`log4j.appender.DDLSchemaFileAppender.MaxFileSize` 行の値を更新して、別の最大ファイルサイズを指定します。

   c. My webMethods Server で保持するバックアップ数を変更するには、`log4j.appender.DDLSchemaFileAppender.MaxBackupIndex` 行の値を更新します。

7. log4j.init.properties ファイルを保存します。
8. 変更したファイルを My webMethods Server データベースに展開します。
   a. コマンドラインプロンプトで、ディレクトリを変更して、サーバの bin ディレクトリに移動します。

      Software AG_directory \MWS\bin

   b. log4j.init.properties ファイルをデータベースに書き込むには、次のコマンドを入力します。

      ```
      mws -s serverName putconfig log4j.init.properties
      ```

   c. log4j.init.properties ファイルのローカルコピーを削除します。

      このファイルを削除しないと、サーバインスタンスは引き続きファイルのローカルバージョンを使用します。
9. クラスタを再起動します。

   変更は、クラスが再起動されるまで有効にはなりません。

デフォルトのログディレクトリの変更

デフォルトでは、サーバインスタンスのログは次の場所に格納されます。

Software AG_directory \MWS\server*serverName*\logs

systemPaths.properties ファイルを変更することによって、このディレクトリの場所を変更できます。

デフォルトのログディレクトリの変更

1. My webMethods Server データベースから systemPaths.properties ファイルをダウンロードします。
   a. コマンドラインプロンプトで、ディレクトリを変更して、サーバの bin ディレクトリに移動します。

      Software AG_directory \MWS\bin

   b. データベースから systemPaths.properties ファイルを抽出するには、次のコマンドを入力します。

      ```
      mws -s serverName getconfig systemPaths.properties
      ```
2. テキストエディタで、次の場所にある systemPaths.properties ファイルを開きます。

Software AG_directory \MWS\server*serverName*\config

3. 新しいログディレクトリの場所を指すように、次の行を変更してファイルを保存します。

```
system.path.logs=root:/logs
```

4. 変更したファイルを My webMethods Server データベースに展開します。

   a. コマンドラインプロンプトで、ディレクトリを変更して、サーバの bin ディレクトリに移動します。

      Software AG_directory \MWS\bin

   b. systemPaths.properties ファイルをデータベースに書き込むには、次のコマンドを入力します。

```
mws -s serverName putconfig systemPaths.properties
```

   c. systemPaths.properties ファイルのローカルコピーを削除します。

      このファイルを削除しないと、サーバインスタンスは引き続きファイルのローカルバージョンを使用します。

5. サーバインスタンスを再起動します。

   変更は、サーバが再起動されるまで有効にはなりません。

ログメッセージの表示

[ログビューア] ページを使用すると、_full_.log、_problems_log および _errors_.xm の各ファイルの最新メッセージを表示できます。

[ログビューア] ページを活用するには、必要に応じた適切なメッセージを確実に収集する必要があります。メッセージ収集条件を設定するには、[ログ設定] ページを使用します。詳細については、310ページの「ログのしきい値について」と312ページの「ロガーしきい値と出力しきい値の設定」を参照してください。

ログメッセージを表示するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [分析] > [ログビューア]

2. 表示する行数を選択します。

3. メッセージを表示するログを選択します。

4. [**今すぐリフレッシュ**] をクリックします。

セキュリティ監査ログの管理

デフォルトで、My webMethods Server は監査可能なすべてのイベントに関して監査ログを記録します。[監査管理] ページを使用すると、監査ログの有効化/無効化、ログに記録するイベントの選択を行うことができます。デフォルトで、監査ログは有効になっており、使用可能なすべての監査可能イベントがログに記録されます。

My webMethods Server は監査ログを audit.log ファイルに書き込みます。このファイルは、次のディレクトリにあります。

Software AG_directory \MWS\server*serverName*\logs

セキュリティ監査ログを管理するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [監査管理]

2. 監査ログを有効または無効にするには、以下の手順に従います。

   a. 監査ログを有効にするには、[**監査の有効化**] をオンにします。

      これがデフォルト設定です。

   b. 監査ログを無効にするには、[**監査の有効化**] をオフにします。

3. [監査可能] 列で、ログに記録するイベントをオンにして、ログに記録しないイベントをオフにします。

   監査ログが無効になっている場合は、オンにしたイベントもログに記録されません。

4. [**適用**] をクリックします。

変更内容は即座に反映されます。

リアルタイムユーザアクティビティの監視

[セッションの監視] ページを使用して、サーバ展開のリアルタイムユーザアクティビティを監視し、電子メールを通じてアクティブなユーザに状態メッセージを送信できます。アクティブなユーザに対して、システム管理者は 2 つの重要な機能を実行できます。

- ユーザのプロファイル情報を表示する
- このページからユーザに電子メールを直接送信する

すべてのアクティブなサーバセッションを表示するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [分析] > [セッションの監視]

2. アクティブセッションのリストで、ユーザの名前をクリックしてそのユーザのプロファイルを表示します (オプション)。

3. アクティブセッションのリストで、 をクリックします (オプション)。

   作業しているマシンに電子メールクライアントがインストールされている場合は、電子メールメッセージウィンドウが表示されます。このウィンドウで電子メールを作成し、選択したユーザに送信できます。[ユーザ情報] に有効な電子メールアドレスが指定されていない場合、[**宛先**] フィールドは空になります。

サーバイベントに関するデータの収集

[Events Collector] ページでは、サーバ上のイベントに関するデータを収集し、他のページでもそれらのデータを使用できるようにします。

[Events Collector] ページをサーバに展開すると、特定のタイプのサーバイベントに関する情報を取得してサーバデータベースに格納します。このページでは、以下のイベントを取得できます。

- ログインおよびログアウト
- ユーザがページを参照する場合などの Get イベント
- オブジェクトが作成、更新、移動または削除される場合などの操作イベント

このページは、各イベントについて、そのイベントに関連付けられたユーザ、イベントの日時、マシンのホスト名、および可能な場合は実行中の操作に関する情報を収集します。

このページで収集されたデータを利用するために、他のページはサーバデータベースに対するクエリーを実行し、結果をページに表示します。Software AGTECHcommunity web サイト (http://techcommunity.softwareag.com) では、これらのクエリーを実行するページの例を検索することができます。これらのページを使用するには、以下の操作を行う必要があります。

1. [Events Collector] ページおよびサンプルページをサーバに展開する
2. [Events Collector] ページを設定する
3. サーバイベントを表示するサンプルポートレットをページに入れる

サンプルページにはポートレットのソースコードが含まれているため、ポートレットを Software AG Designer にインポートして、どのように機能するかを見ることができます。[Events Collector] ページがデータをサーバデータベースに格納するために使用するデータベーススキーマの例は、326 ページの「[Events Collector] データベーススキーマ」を参照してください。

[Events Collector] ページの展開

[Events Collector] ページは、My webMethods Server 標準インストールの一部ですが、デフォルトでは展開されません。ページを使用するには、まずサーバにページを展開する必要があります。

サーバに [Events Collector Configuration] ページを展開するには

1. My webMethods Server ディレクトリ構造の次の場所で、[Events Collector Configuration] ページを検索します。

   Software AG_directory \MWS\components\extras\analysis\wm_eventscollector.pdp

2. wm_eventscollector.pdp ファイルをコピーし、deploy ディレクトリに貼り付けます。

   Software AG_directory \MWS\server*server_name*\deploy

   server_name はサーバの名前です。数秒後、ページは自動的にサーバ上に展開されます。

[Events Collector Configuration] ページの設定

デフォルトでは、[Events Collector Configuration] ページは、展開すると直ちにイベントに関するデータの収集を開始できるようになりますが、データの保存期間を変更したり、ページを無効にしたりすることが必要になる場合があります。

[Events Collector Configuration] ページを設定するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [分析] > [Events Collector Configuration]

   注:
   [分析] フォルダに [Events Collector Configuration] ページが見つからない場合は、ページが展開されていないと考えられます。詳細については、325 ページの「[Events Collector] ページの展開」を参照してください。

2. [**Collection Enabled**] チェックボックスをオンまたはオフにして、サーバイベントデータの収集を有効または無効にします。

   デフォルトでは、データ収集は有効です。

3. [**Keep Data for**] リストで、サーバイベントデータを保存する期間を選択します。

   選択の範囲は、[**1 日**] から [**One Year**] までです。デフォルトは、[**One Month**] です。

4. [**適用**] をクリックします。

[Events Collector] データベーススキーマ

[Events Collector] ページでは、次のデータベーススキーマが使用されます。

```
tblwEvents (main table where events data is being collected)
  idEvent    - Primary key
  idType     - Foreign key to tblwEventTypes.Stores the type
                  of an event.
  idHost     - Foreign key to tblwEventHosts.Stores the host
                  where the event occurs.
  timestamp  - Time stamp of an event, defined as the number of
                  milliseconds since epoch(java.lang.System.
                  currentDateMillis()).
  userID     - Database ID of the user who performed an operation.
  thingID_1  - object_1.For example, for Get type events this is
                  the database ID of the object being viewed.
  thingID_2  - object_2.Used in rare cases where two objects are
                  involved, for example when an object is created.
                  Then object_1 is the database ID of the container,
                  and object_2 is the database ID of an object that
                  was created.
  action     - Used for Login events: 1 - user logged in, 2 - user
                  logged out.
tblwEventHosts (stores mapping between hostID and hostname)
  idHost     - Host id.
  hostname   - Actual host name where the event occurs.
tblwEventTypes (stores mapping between eventID and eventTypeName)
  idType     - Event type ID.
 eventType   - Event type name.
Possible eventType name values:
  com.webmethods.portal.event.add.impl.CreateEvent
    - New object is created.
  com.webmethods.portal.event.impl.GetEvent
    - Object is being browsed.
  com.webmethods.portal.event.system.impl.LoginEvent
    - User logs in/out.
  com.webmethods.portal.event.modify.impl.UpdateEvent
    - Object is updated.
  com.webmethods.portal.event.remove.impl.DeleteEvent
    - Object is deleted.
```

データベース変更に関するデータの収集

My webMethods Server データベースに対する DDL ステートメント実行のログ情報を収集できます。データベース変更に関するすべてのデータは、収集されて `schema.log` ファイルに書き込まれます。データベース変更情報のログの設定は、[Administration Dashboard] > [分析] > [ログ設定] ページにあります。

ログしきい値の設定の詳細については、310 ページの「サーバのログの制御」を参照してください。

収集されたデータの表示方法の詳細については、322 ページの「ログメッセージの表示」を参照してください。

My webMethods Server 診断ツール

My webMethods Server 診断ツールを使用して、サーバ操作に関するデータを収集および分析できます。My webMethods Server 診断ツールには、診断コマンドラインツールと診断ポートレットという 2 つのタイプのツールがあります。ここでは、診断ツールの概要について説明します。詳細については、*Diagnosing My webMethods Server*を参照してください。

診断コマンドラインツールを使用して、以下の作業を実行できます。

- threaddump ツールを使用して、ローカルまたはリモートシステム上のサーバのスレッド実行に関するデータを収集します。
- envcapture ツールを使用して、トラブルシューティングの支援のためにSoftware AG Global Supportに提供する環境固有のサーバ情報を取得します。
- envdiff ツールを使用して、環境固有のサーバ情報の取得結果である XML ファイルを比較します。
- memorydump ツールを使用して、サーバのメモリ割り当て情報を取得します。
- loganalyzer ツールを使用して、errors.log ファイルで報告された問題を識別および分析します。

診断ポートレットを使用して、以下の作業を実行できます。

- [スレッドダンプ] ツールを使用して、サーバスレッドでのスレッド実行デッドロックを監視します。
- [パフォーマンス分析] ツールを使用し、埋め込みパフォーマンス監視サービスを使用して、サービスおよびカスタムポートレットまたはアプリケーションのパフォーマンスを測定および分析します。
- [パフォーマンス統計] ツールを使用して、カテゴリ別にグループ化されたさまざまなサーバアクションのパフォーマンスに関する統計および分析情報を表示します。
- [ログ分析] ツールを使用して、log4j メカニズムに従ってサーバで作成されたログファイルを読み取って分析します。
- [メモリの監視] ツールを使用して、Java 仮想マシン (JVM) のメモリ使用量を監視したり、設定したしきい値に達した場合に管理者に通知を送信します。

17 My webMethods Server の設定

- My webMethods Server の設定について 330
- エイリアスの管理 330
- My webMethods Server コンポーネントの展開 334
- My webMethods Server キャッシュの設定 337
- システム情報の表示 343

My webMethods Server の設定について

My webMethods Server には、サーバの設定に使用できる多数のツールが用意されています。これらの機能は、My webMethods Server インスタンスのインストールおよび設定後に実行します。

エイリアスの管理

[エイリアス管理] ページでは、URL エイリアスをサーバオブジェクトとして管理できます。このページでカスタム URL エイリアスを作成、表示、変更、または削除し、サーバのさまざまな部分への URL を簡単にすることができます。

たとえば、販売部門用にサーバの領域を作成する必要があり、サーバの [パブリックフォルダ] に販売チームのフォルダが既に作成されている場合、そのフォルダは次のようにわかりにくい URL で参照されることがあります。`http://server:port/meta/default/folder/0000002216`

販売チームが販売サーバの場所を簡単に覚えられるように、[エイリアス管理] ページを使用して、次のようにユーザフレンドリな URL を作成できます。`http://server:port/Sales`

[プロパティ] ページでのサーバリソースへのエイリアスの作成

サーバリソースの [プロパティ] ページで、サーバリソースのエイリアスを作成できます。

[プロパティ] ページでサーバリソースのエイリアスを作成するには

1. システム管理者として、サーバリソースが置かれているページまで移動します。
2. サーバリソースの をクリックして、[プロパティ] を選択します。
3. [プロパティ] ページの [**エイリアス**] フィールドで、[**追加**] をクリックします。
4. サーバリソースのエイリアス名を入力して、[**OK**] をクリックします。

   エイリアス名には空白を含めないでください。含めると、エイリアスは正しく機能しません。
5. [**適用**] をクリックします。

[エイリアス管理] ページでのサーバリソースへのエイリアスの作成

[エイリアス管理] ページで、サーバリソースのエイリアスを作成できます。

[エイリアス管理] ページでサーバリソースのエイリアスを作成するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [エイリアス管理]

2. [**作成**] をクリックします。

3. [エイリアス名] フィールドで、作成する新しいエイリアスの名前 (例えば「Sales」など) を入力します。

   エイリアス名には空白を含めないでください。

4. [ターゲット] で新しいエイリアスに次のいずれかのターゲットを選択します。

   - [リソース] - フォルダ、ポートレット、アイテムなどのサーバリソースをターゲットにします。リソースを指定するには、[参照] をクリックして [選択された項目] ボックスにリソースを移動し、[選択] をクリックします。
   - [パス] - URL などの外部リソースをターゲットにします。`http://www.softwareag.com` などの外部リソースにパスを指定します。

   注:
   エイリアスのターゲットにサーバリソースを使用する場合は、[ストリングを追加] フィールドを使用してパラメータを渡すか、エイリアスが参照するリソースに対してサーバコマンドを呼び出すことができます。

5. [エイリアスの追加] をクリックします。

6. 新規エイリアスのユーザフレンドリな URL を参照して、エイリアスが適切に動作することを確認します。

   デフォルトでは、作成したエイリアスはサーバのルート URL に付加されます。たとえば、「Sales」というエイリアスを作成した場合は、「`http://server:port/Sales`」という URL を入力して新規エイリアスにアクセスします。

エイリアスの検索

[エイリアス管理] ページを使用して、既存のエイリアスを検索できます。ページにより検索結果がリストに加えられ、ここからエイリアスの変更や削除、またはターゲットリソースの表示ができます。

ヒント:
エイリアス検索フィールドでは、名前内の任意のテキストの代わりに 1 つのワイルドカード文字 (*) を使用できます。

単純エイリアス検索の実行

エイリアスを検索するには、以下の手順に従います。

エイリアスの単純検索を実行するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [エイリアス管理] > [検索]

2. 検索するエイリアスの名前を入力します。

   システムエイリアスを検索条件に含めたり、検索条件から除外したりするには、332 ページの「システムエイリアスの追加または除外」を参照してください。

3. [**実行**] をクリックします。

   検索に一致したすべてのエイリアスが表に表示されます。

システムエイリアスの追加または除外

検索時にシステムエイリアスを追加または除外するには、以下の手順に従います。

検索時にシステムエイリアスを追加または除外するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [エイリアス管理] > [検索]

2. 検索するエイリアスの名前を入力します。

3. [**絞り込む**] をクリックします。

4. [**システムエイリアスを含む**] リストで、以下のいずれかを選択します。

   - [**はい**] - 検索にシステムエイリアスを含みます。
   - [いいえ] - 検索からシステムエイリアスを除外します。

5. [**実行**] をクリックします。

6. 絞り込まれた検索パネルを閉じるには、[**閉じる**] をクリックします。

拡張エイリアス検索の実行

エイリアスの拡張検索を実行するには、以下の手順に従います。

エイリアスの拡張検索を実行するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [エイリアス管理] > [高度な検索]

2. 検索するエイリアスの名前を入力します。

3. [**システムエイリアスを含む**] で、次のいずれかのオプションを選択します。

- [はい] - デフォルト。検索にシステムエイリアスを含みます。
- [**いいえ**] - 検索からシステムエイリアスを除外します。

4. [エイリアスのターゲット] で、以下のいずれかを行います。
   - [参照] をクリックし、ターゲットのリソースを [選択された項目] ボックスに移動して、[選択] をクリックします。
   - [**エイリアスを使用**] をクリックし、[**エイリアス名**] フィールドにターゲットリソースのエイリアスを入力します。[**テスト**] をクリックして、そのエイリアスが有効で、エイリアスターゲットが正しいものであるかどうかを確認します。エイリアスが正しければ、[**選択**] をクリックします。
5. [**実行**] をクリックします。

保存済みエイリアス検索の使用

いつでも使用できるようにエイリアス検索を保存できます。保存済みエイリアス検索の使用方法は、ユーザ、グループ、役割の保存済み検索を使用する場合と同じですが、実行される場所は次の場所になります。

- システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [エイリアス管理]

  ユーザ、グループ、役割の保存済み検索の詳細については、「148 ページの「保存済み検索の使用」」を参照してください。

別のサーバリソースを指すようにエイリアスを変更

別のサーバリソースを指すようにエイリアスを変更するには、以下の手順に従います。

別のサーバリソースを指すように既存のエイリアスを変更するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [エイリアス管理]
2. [**エイリアス検索**] パネルを使用して、変更するエイリアスを探します。
3. 検索結果で、変更するエイリアスの をクリックします。
4. [新規ターゲット] パネルで新しいエイリアスに次のいずれかのターゲットを選択します。
   - [リソース] - フォルダ、ポートレット、アイテムなどのサーバリソースをターゲットにします。リソースを指定するには、[参照] をクリックして [選択された項目] ボックスにリソースを移動し、[選択] をクリックします。
   - [パス] - URL などの外部リソースをターゲットにします。`http://www.softwareag.com` などの外部リソースにパスを指定します。

> **注:**
> エイリアスのターゲットにサーバリソースを使用する場合は、[ストリングを追加] フィールドを使用してパラメータを渡すか、エイリアスが参照するリソースに対してサーバコマンドを呼び出すことができます。

5. [**更新**] をクリックします。

エイリアスの削除

エイリアスを削除するには、以下の手順に従います。

既存のエイリアスを削除するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [エイリアス管理]
2. [**エイリアス検索**] パネルを使用して、変更するエイリアスを探します。
3. 検索結果で、削除するエイリアスの をクリックします。
4. 検索結果で、削除するエイリアスの横のチェックボックスをオンにして、[**削除**] をクリックします。

My webMethods Server コンポーネントの展開

ポートレットや DBO などのサーバコンポーネントをサーバにインストールする際、システム管理者は以下のオプションを使用できます。

- Administration Dashboard の [インストール管理] ページを使用します。
- サーバのファイルシステムの Deploy フォルダを使用します。このフォルダにより、サーバ管理者と開発者は、新規に開発したポートレットパッケージ (ポートレット、CAF アプリケーション、タスクアプリケーション、または展開可能パッケージ) を、サーバによって定期的にポーリングされる特定のディレクトリにコピーまたは貼り付けできます。サーバがこのフォルダに新しい展開可能コンポーネントを検出した場合、これらのコンポーネントが自動的に抽出され、サーバにインストールされます。サーバが Deploy ディレクトリをポーリングして新規コンポーネントを検出する頻度を指定するポーリング間隔を設定するオプションがあります。

ポーリング間隔の変更

組織で複数のポートレットを開発している場合、このインストール方法は、ポートレットを一度に手動でインストールする方法よりも便利です。ファイルシステムでの Deploy フォルダのデフォルトの場所は、次のとおりです。

Software AG_directory \MWS\server*serverName*\deploy

注:
サーバのファイルシステムで PhaseProvider.xml 設定ファイルを変更して、ポーリングをオンまたはオフにできます。ポーリング間隔を変更するには、以下の手順に従います。

ポーリング間隔を変更するには

1. コマンドラインプロンプトで、次のコマンドを入力してサーバの bin ディレクトリに移動します。

```
cd Software AG_directory\MWS\bin
```

2. 次のように `getconfig` コマンドを使用して、My webMethods Server データベースから `phaseProvider.xml` ファイルを取り出します。

```
mws getconfig phaseProvider.xml
```

3. 次の場所に移動します。

   Software AG_directory \MWS\server*serverName*\config

4. テキストエディタまたは同等の XML 編集機能を使用して `phaseProvider.xml` 設定ファイルを開きます。次の XML フラグメントを探します。

```
<Phase name="deploySync" enabled="true"
    class="com.webmethods.portal.system.init.impl.MasterServerPhase">
```

```
<PhaseInfo name="startTimedSyncDeploy" enabled="true"
    class="com.webmethods.portal.bizPolicy.biz.install.impl.
```

```
SyncDeployService" interval="5" />
```

```
</Phase>
```

5. ポーリングをオフにするには、enabled 属性を `true` から `false` に変更します。

6. ポーリング間隔を変更するには、interval 属性を目的の値に変更します。デフォルト設定は 5 秒です。

   注:
   この設定は、全体的なパフォーマンスには影響しません。

7. ファイルを保存します。

8. 次のように `putconfig` コマンドを使用して、変更したファイルを My webMethods Server データベースに展開します。

```
mws putconfig phaseProvider.xml
```

9. *serverName*\config ディレクトリからファイルを削除します。

ファイルを削除しないと、このサーバインスタンスがローカルバージョンの設定ファイルで引き続き使用されます。

10. サーバインスタンスを再起動します。

Deploy フォルダを使用したポートレットのインストール

Deploy フォルダを使用してポートレットをサーバにインストールするには、以下の手順に従います。

Deploy フォルダを使用してポートレットをインストールするには

- 展開するサーバコンポーネントをコピーし、展開する deploy ディレクトリに貼り付けます。

  注:
  いずれかのサーバコンポーネントの展開に失敗した場合、サーバのファイルシステムの Deploy フォルダに Failed ディレクトリが自動的に作成されます。適切にインストールされないコンポーネントは、すべて Failed ディレクトリにコピーされます。

ポートレットまたはその他の展開可能なサーバコンポーネントのインストール

[インストール管理] ページを使用して、ポートレットまたはその他の展開可能なサーバコンポーネントをインストールするには、以下の手順に従います。

ポートレットまたはその他の展開可能サーバコンポーネントをインストールするには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [インストール管理]
2. [**新規コンポーネントをインストール**] をクリックします。
3. [**ローカルの場所またはネットワーク上の場所**] または [**リモートの場所**] を選択します。
4. 以下のいずれかを行います。
   - 展開可能コンポーネントがローカルファイルシステム上にある場合は、[**参照**] をクリックし、その場所に移動します。
   - 展開可能コンポーネントがリモートの場所にある場合は、コンポーネントのフルパスを入力します。
5. [**進む**] をクリックします。
6. [新規コンポーネントの情報] を確認し、[**インストール**] をクリックします。

コンポーネントが正常にインストールされた場合は、インストールが成功したことを示す確認メッセージが表示されます。

注:
インストールが失敗したコンポーネントは自動的にアンインストールされます。ログファイルをチェックして、インストールのエラーをトラブルシューティングしてください。

ポートレットまたはその他の展開可能なサーバコンポーネントのアンインストール

コンポーネントをアンインストールする前に、そのコンポーネントを削除するとユーザのページのインスタンスすべてにどのような影響があるかを判断します。アンインストールを行うと、削除されたポートレットの各インスタンスを含むすべてのページが破棄され、ポートレットワイヤリング機能を使用してそのポートレットにワイヤリングされているすべてのポートレットが分断されます。

たとえば、別のポートレットにワイヤリングされているポートレットを削除するとき、ワイヤリング関係については警告されません。

ポートレットをアンインストールする前にポートレットの [状態] プロパティを [非表示] または [無効] に変更し、ポートレットを段階的に削除する方法があります。実行されようとしているアンインストールがユーザに通知され、ユーザがページから削除した後であれば、安全にアンインストールすることができます。

重要:
アンインストールしたポートレットのインスタンスが破棄されると、そのポートレットが使用されていた各ページでエラーが発生します。ポートレットとそのインスタンス、設定ファイル、ポートレットデータベーステーブル、およびポートレットパッケージングファイルのデータも削除される場合があります。再インストールを行っても、ポートレットのアンインストールが原因で破棄された参照は復元されません。

[インストール管理] ページを使用してコンポーネントをアンインストールするには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [設定] > [インストール管理]

2. ツリーリストでアンインストールする 1 つ以上のコンポーネントを選択し、[**選択項目のアンインストール**] をクリックして、[**次へ**] をクリックします。

3. アンインストール操作を確認するには、[**アンインストール**] をクリックします。

My webMethods Server キャッシュの設定

[Cache Configuration] ポートレットを使用して、My webMethods Server コンポーネントのデフォルトのキャッシュ設定を変更します。[Cache Configuration] ポートレットは、デフォルトではインストールされません。My webMethods Server ポートレットをインストールする方法の詳

細については、336 ページの「ポートレットまたはその他の展開可能なサーバコンポーネントのインストール」を参照してください。

cache.xml ファイルを使用してキャッシュ設定を指定する方法については、次を参照してください。340 ページの「cache.xml ファイルのキャッシュ設定の変更」

個々の OSGi コンポーネントのキャッシュ設定を指定する方法については、次を参照してください。340 ページの「cache.xml ファイルのキャッシュ設定の変更」

My webMethods Server キャッシュを設定するには

1. sysadmin として、[フォルダ] > [管理用フォルダ] > [Administration Dashboard] > [設定] > [Cache Configuration] に移動します。

2. 設定したいキャッシュタイプが最初のページに表示されない場合は、[Next] をクリックします。

3. [Cache Max Size] フィールドで、設定するキャッシュタイプの新しいキャッシュの最大サイズを入力します。

4. オプションで、[Cache Detailed Report] を確認します。

   [Cache Detailed Report] には、各キャッシュエントリについての詳細情報が表示されます。キャッシュタイプとキャッシュ属性の詳細については、338 ページの「キャッシュタイプおよびキャッシュ属性」を参照してください。

5. [Apply] をクリックします。

   1 つのインスタンスに構成されたキャッシュ設定は、My webMethods Server クラスタ全体に自動的に反映されます。

キャッシュタイプおよびキャッシュ属性

キャッシュタイプ

以下の表に、[Cache Configuration] ポートレットで使用可能なキャッシュタイプを示します。一部のキャッシュタイプは読み取り専用です。

キャッシュ名	説明
AccessCache	ユーザのアクセス権を保存するための非一時的なキャッシュ。
AliasCache	エイリアスキーを保存するための非一時的なキャッシュ。
BpmProcessDefinitionCache	BPM プロセス定義を保存するための一時的なキャッシュ。
BpmProcessInstanceCache	BPM プロセスインスタンスを保存するための一時的なキャッシュ。

キャッシュ名	説明
ContainerCache	xtype コンテナをリフレッシュするための一時的なキャッシュ。
GroupCache	プリンシパルがメンバーになっているグループを保存するための非一時的なキャッシュ。
InvalidLoginsCache	無効なログイン試行データを保存するための非一時的なキャッシュ。
JMSTransactionCache	JMS トランザクションデータを保存するための非一時的なキャッシュ。
PortletTransientData	ポートレットで使用されるオブジェクトを保存するための非一時的なキャッシュ。
PresentationCache	レンダリングされた HTML フラグメントを保存するための非一時的なキャッシュ。
RoleCache	プリンシパルがメンバーになっている役割を保存するための非一時的なキャッシュ。
TaskDataCache	タスクデータを保存するための一時的なキャッシュ。
ThingCache	My webMethods Server オブジェクトの ThingID を保存するための一時的なキャッシュ。
ThingRelationCache	ThingID とオブジェクト間の関係を保存するための一時的なキャッシュ。
AliasCache の一時的なキャッシュ	ユーザのアクセス権を保存するための一時的なキャッシュ。
PresentationCache の一時的なキャッシュ	エイリアスキーを保存するための一時的なキャッシュ。

キャッシュ属性

以下の表に、[Cache Configuration] ポートレット上のキャッシュタイプの詳細キャッシュレポートで使用できるキャッシュ属性を示します。一部の属性は、キャッシュタイプ固有のものです。

キャッシュ属性	説明
キャッシュサイズ	キャッシュに保存されているアイテムの数
最大サイズ	キャッシュに保持するアイテムの最大数。
有効なエントリ	キャッシュに保存されている有効なエントリの数。有効なエントリとは、オブジェクトに解決されるアイテムです。
無効なエントリ	キャッシュに保存されている無効なエントリの数。無効なエントリとは、オブジェクトに解決されないアイテムです。

キャッシュ属性	説明
期限切れエントリ	有効期限に達したアイテムの数。
期限付きエントリ	有効期限のあるアイテムの数。
キャッシュヒット数	キャッシュ内に見つかったアイテムの数。
キャッシュヒットなし	キャッシュ内で見つからなかったアイテムの数。
依存関係の合計数	依存関係のあるキャッシュエントリの合計数。
依存関係を持つエントリ	キャッシュの他のアイテムに依存関係を持つエントリの数。

cache.xml ファイルのキャッシュ設定の変更

My webMethods Server のキャッシュタイプの詳細設定を行うには、cache.xml ファイルを使用します。cache.xml ファイルは、My webMethods Server データベースに格納されています。cache.xml ファイルに変更を加えると、My webMethods Server クラスタ内のすべてのノードにその変更が適用されます。

ファイルのローカルコピーは、デフォルトでは *Software AG_directory* \MWS\server*serverName*\config に保存されています。My webMethods Server インスタンスを設定して、デフォルト以外の場所に保存されている cache.xml ファイルを使用するには、そのインスタンスの custom_wrapper.conf に次の JVM プロパティを追加します。

```
-Dcom.webmethods.mws.cachemanager.configfile=path
```

path は cache.xml ファイルが含まれるディレクトリへのパスを示します。JVM プロパティの設定の詳細については、97 ページの「My webMethods Server の JVM 設定」を参照してください。

cache.xml ファイルでカスタム設定を指定しない限り、それぞれのキャッシュタイプの OSGi プロパティを使用したキャッシュ設定が適用されます。OSGi プロパティは、各キャッシュタイプのコンポーネント jar の OSGI-INF/ ディレクトリに保存されています。341 ページの「カスタムプロパティファイルを使ったキャッシュ設定の変更」で説明されているように、コンポーネントごとに個別のキャッシュ設定を行うこともできます。

cache.xml ファイルのキャッシュ設定を変更するには

1. コマンドラインプロンプトで、サーバの bin ディレクトリに移動します。

   ```
   Software AG_directory\MWS\bin
   ```

2. `getconfig` コマンドを使用して、My webMethods Server データベースから cache.xml ファイルを取り出します。

   ```
   mws getconfig cache.xml
   ```

3. cache.xml ファイルをテキストエディタで開き、必要に応じてキャッシュ設定を編集します。

cache.xml ファイルのデフォルトのダウンロード場所は、 *Software AG_directory* \MWS\server*serverName*\config です。

4. ファイルを保存して閉じます。

5. 更新したファイルを My webMethods Server データベースに展開するには、次のコマンドを入力します。

```
mws putconfig cache.xml
```

6. *serverName*\config ディレクトリから cache.xml ファイルのローカルコピーを削除します。

   コピーを削除しないと、このサーバインスタンスがローカルバージョンの設定ファイルで引き続き使用されます。

7. `restart` コマンドを使用して、ノードを再起動します。

```
mws -s serverName restart
```

   サーバの再起動後に設定ファイルへの変更が適用されます。

カスタムプロパティファイルを使ったキャッシュ設定の変更

.property ファイルを作成し、My webMethods Server がサポートする各キャッシュタイプの OSGi コンポーネントのキャッシュ設定を個別に設定することができます。プロパティファイルに指定された設定は、単一のキャッシュタイプに適用され、現在のノード上でのみ適用されます。My webMethods Server の cache.xml ファイルで指定された設定は、.property ファイル内の個別のキャッシュ設定を上書きします。

プロパティファイルを使用してキャッシュ設定を変更するには

1. テキストエディタでキャッシュクラスの完全修飾名と同じ名前の .properties ファイルを作成します。

2. 必要なプロパティをキーと値のペアで追加します。各ペアは、引用符や区切り文字を使用せずに、1 行ずつ指定します。

3. 次の場所にプロパティファイルを保存します。

   SoftwareAG_directory\profiles\MWS_*instanceName*\configuration\com.softwareag.platform.config.propsloader

例

My webMethods Server のエイリアスキャッシュを設定する場合:

1. `com.webmethods.portal.service.cache.impl.AliasCache.properties` という名前のファイルを作成します。

2. ファイルに次のエントリを追加して、エイリアスキャッシュの最大サイズを1002222 エントリに設定します: `maxSize=1002222`

3. `com.webmethods.portal.service.cache.impl.AliasCache.properties` ファイルを `\profiles\MWS_default\configuration\com.softwareag.platform.config.propsloader` ディレクトリに保存します。

役割キャッシュライフサイクルまたはグループキャッシュライフサイクル計算の設定

デフォルトでは、ユーザの役割キャッシュまたはグループキャッシュは、ユーザログインごとに再計算されます。長時間かかるユーザ検索またはグループ検索が計算に含まれていると、My webMethods Server および接続先の LDAP サーバでパフォーマンスが低下します。cache.xml ファイルの設定オプションを使用して、役割キャッシュまたはグループキャッシュをユーザログイン間で継続するようにすると、パフォーマンスが大幅に向上できます。

ユーザログイン時に役割が計算されなくなるため、この最適化によって動的役割の機能が変更されます。デフォルト設定では、役割はユーザログイン時に計算され、ユーザはその時点で役割変更を取得または喪失します。新しい設定では、動的クエリー役割によって結果が変更された場合、ユーザはその変更を即時には取得しません。ただし、システム管理者が行う役割変更は認識されます。このモードでは、静的役割は新しい設定で正しく機能します。動的役割については、管理者が役割のメンバーシップの変更または詳細の追加を行うか、さまざまなキャッシュがタイムアウトするまで、適切な変更は戻されません。

最適化では、グループ定義がMy webMethods Server によって制御されないため、キャッシュエントリがタイムアウトするかシステム再起動が発生するまで、グループキャッシュは更新されません。ユーザがグループメンバーシップを頻繁に変更する場合、グループキャッシュの最適化は適切でないことがあります。ログイン時に役割キャッシュまたはグループキャッシュの再計算を実行するかどうかの選択には、パフォーマンスと機能との兼ね合いを考慮します。

My webMethods Server の cache.xml ファイルの編集の詳細については、次を参照してください。 340 ページの「cache.xml ファイルのキャッシュ設定の変更」

役割キャッシュライフサイクルまたはグループキャッシュライフサイクル計算を設定するには

1. My webMethods Server データベースから cache.xml ファイルを取得します。

2. cache.xml ファイルをテキストエディタで開き、キャッシュを探します。

   - 役割キャッシュ:

```
<cache name="RoleCache"
class="com.webmethods.portal.service.cache.impl.RoleCache"
    maxSize="1000"
    defaultCacheTimeout="3600000"
    roleCacheLifecycle="0"
    isClustered="true"
    ID="11"
    enabled="true"/>
```

- グループキャッシュ:

```
<cache name="GroupCache"
class="com.webmethods.portal.service.cache.impl.GroupCache"
    maxSize="1000"
    defaultCacheTimeout="3600000"
    groupCacheLifecycle="0"
    isClustered="true"
    ID="19"
    enabled="true"/>
```

3. `roleCacheLifecycle` または `groupCacheLifecycle` の値を必要に応じて変更します。

次の表に、グループキャッシュまたは役割キャッシュに指定可能な値を示します。

値	アクション
`0`	(デフォルト) ユーザログイン時にキャッシュを再計算します。
`1`	ユーザログイン間でキャッシュを継続させます。

4. cache.xml ファイルを保存して閉じます。

5. 変更したファイルを My webMethods Server データベースに展開し、ファイルのローカルコピーを削除します。

6. My webMethods Server ノードを再起動します。

システム情報の表示

[システム情報] ページでは、サーバの現在の状態についての豊富な情報を提供します。各パネルを開いたときに、このページにより動的に情報が収集されます。

サーバの現在の状態に関するシステム情報を表示するには

- システム管理者として次のように選択します。[**Administration Dashboard**] > [分析] > [システム情報]

[システム情報] ページのパネル

[システム情報] ページには 5 つのパネルがあります。パネルをクリックして前面に表示すると、ページでは表示するデータが動的に収集されます。

次の表に、[システム情報] ページで使用可能なパネルとその内容を示します。

パネルと見出し	説明
[要求/応答]	ユーザの Web 要求から収集された情報

パネルと見出し	説明
[要求情報]	要求されたパスを説明する標準的な cgi-bin パラメータ
[要求ヘッダー]	受信 HTTP ヘッダー
[要求パラメータ]	URL の受信 HTTP パラメータ
[要求属性]	現在の要求に格納されている属性 (オブジェクト)
[応答情報]	要求から収集されたエンコーディングやロケールなどのその他の情報
[セッションのその他の情報]	ユーザセッション情報
[セッション属性]	ユーザのセッションに格納されている属性 (オブジェクト)
[ロケール情報]	ユーザの現在のロケール
[プレゼンテーションデータ]	このユーザの要求のレンダリングに使用されるさまざまな情報
[セッション属性]	このユーザに関連付けられたポートレットコントローラセッションオブジェクト
[要求属性]	このユーザの要求に関連付けられたポートレットコントローラ要求オブジェクト
[アプリケーション属性]	サーバ全体 (全ユーザ間) で共有される情報
[システム情報]	クラスパス、パスなどの環境変数
[サーバ情報]	現在のフロントエンドサーバに関する情報
[コンテキスト情報]	サーブレットオブジェクト情報

18 My webMethods Server コンテンツの管理

- My webMethods Server コンテンツの移行 346
- コンテンツ記憶領域の管理 346
- 管理者としてポートレットをパブリッシュ 349
- 検索インデックスの再構築 351
- カスタム JAR ファイルの追加 351

My webMethods Server コンテンツの移行

[コンテンツマイグレーションウィザード] ページでは、あるサーバインスタンスから別のインスタンス (開発環境からステージング、実稼動環境など) にサーバコンテンツを移行できます。

このページを使用して、ドキュメント、フォルダ、外部リンク、内部リンク (エイリアスを使用)、ページ (レイアウトを含む)、ポートレット、DBO (Dynamic Business Objects)、権限、サブスクリプション、ポートレットワイヤリングプロパティといったタイプのサーバコンテンツを移行できます。

コンテンツ移行の考慮事項

コンテンツ移行では、ソースのサーバインスタンスからコンテンツをエクスポートし、次にターゲットのサーバインスタンスにコンテンツをインポートするという 2 段階のアクティビティを実行します。これらの処理を実行する前に、次の点を考慮する必要があります。

- **ポートレットと DBO の移行**: 開発サーバでポートレットを開発またはインストールし、そのポートレットのインスタンスを含むページを移行する場合は、これらをターゲットサーバに展開してから、開発サーバインスタンスで開発またはインストールされたページやポートレットのパブリッシュ済みインスタンスを移行する必要があります。
- **パブリッシュされたコンテンツの移行**: コンテンツ管理システムにパブリッシュされたコンテンツは、あるサーバインスタンスから別のインスタンスに移行できます。開発環境を使用して、コンテンツ管理システムにパブリッシュされるアイテムの権限を設定している場合は、権限も移行するオプションがあります。
- **リンクの移行**: あるページから別のページへのリンクなど、他のオブジェクトを参照している内部リンクを適切に移行するには、ベース URL を使用する代わりにこれらのリンクのエイリアスを作成します。

  たとえば、既存のページ、ここでは次の URL を初期 URL として持つ販売ページへのリンクをパブリッシュするとします。

  ```
  http://server/meta/default/folder/0000002132
  ```

  この場合は、この URL を指すエイリアスを作成しますが、次のようにわかりやすい URL にします。

  ```
  http://server/Sales
  ```

- **権限とサブスクリプションの移行**: ソースからターゲットのサーバインスタンスに権限とサブスクリプションを適切に移行するには、必ず両方のサーバが同じディレクトリサービスを指すようにします。

コンテンツ記憶領域の管理

システム管理者は、[コンテンツサービス] ページを使用して、サーバにパブリッシュするコンテンツに使用できる保存場所を管理します。パブリッシュされたコンテンツは、コンテンツサービスで設定された場所に物理的に格納されます。通常、コンテンツは、バックアップの目的、およ

び My webMethods Server クラスタにおけるフェイルオーバー機能や高可用性を提供する目的で別のファイルサーバに格納されます。

My webMethods Server は、次のような場合に、コンテンツサービス (システムまたはネットワーク) にファイルのバイナリコンテンツを保存します。

- サーバにファイルをパブリッシュする場合
- ファイルを更新する場合
- バージョン管理されているファイルの新規バージョンをチェックインする場合

ファイルオブジェクトが My webMethods Server に存在する間は、バイナリコンテンツがコンテンツサービスに物理的に格納されます。My webMethods Server 上の対応するファイルオブジェクトを削除すると、バイナリコンテンツに削除のマークが付けられますが、実際のファイルは、定期的にスケジュールされる消去期間まで削除されません。通常、My webMethods Server はサーバ時刻の午前 2 時に、削除されたオブジェクトを消去します。

新しいデフォルトコンテンツサービスを設定すると、すべての新規コンテンツはそのサービスに格納されますが、既存のサービスに格納されているコンテンツは引き続きそのサービスを使用してアクセスされます。

コンテンツサービス

デフォルトのシステムコンテンツサービスは、ファイルを My webMethods Server データベースに保存します。システムコンテンツサービスの設定を変更することはできません。

My webMethods Server のユーザが大きなファイル (1 MB 以上) や数多くのファイルを保存する場合、このようなファイルの保存用としてはシステムコンテンツサービスが適切ではない可能性があります。このような場合は、より大きな容量を提供できるネットワークコンテンツサービスを設定する必要があります。My webMethods Server では、ファイルシステムを使用したネットワークファイル記憶領域がサポートされています。

Windows におけるファイルシステムコンテンツサービスでは、UNC (Uniform Naming Convention) パスを使用して、ネットワークファイルストレージデバイスに接続します。UNIX では、ネットワークファイル記憶領域はローカルリソースとしてマウントされる必要があります。クラスタの高可用性をサポートするために、ネットワークファイルストレージデバイスではフェイルオーバーサポートが提供されている必要があります。

ネットワークコンテンツサービスの設定の詳細については、347 ページの「新規コンテンツサービスの設定」を参照してください。

新規コンテンツサービスの設定

この手順は、オンプレミスのインストールからの My webMethods Server インスタンスのみに適用されます。My webMethods Server が Docker コンテナで実行されている場合は、新規コンテンツサービスを設定することはできません。

サーバリポジトリの新規コンテンツサービスを設定するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [コンテンツ] > [コンテンツサービス] > [新規コンテンツサービスの作成]

2. [**サービス名**] フィールドに、新規コンテンツサービスの名前を入力し、[**次へ**] をクリックします。

   サービス名は 1～255 文字の ASCII 英数字で指定します。空白を含めることはできません。

3. コンテンツサービスの物理的な保存場所を入力します。

   Windows では、有効な UNC パスを使用します。パスは大文字と小文字が区別されません。次に例を示します。

   ```
   \\server\volume\directory\
   ```

   UNIX では、マウントされたネットワーク記憶領域の場所のパスを使用します。次に例を示します。

   ```
   /mounted_location/folder
   ```

   注:
   このパスの例では、ネットワーク管理者が適切なセキュリティ設定を提供しており、クラスタ内のすべてのサーバからネットワークファイルシステムに読み取りおよび書き込みアクセスを実行できることを想定しています。My webMethods Server が Windows サービスとして実行されている場合は、サービスを実行するユーザアカウントが、ネットワークファイルシステムへの適切なアクセス特権を持っていることを確認してください。

4. [**サブミット**] をクリックします。

5. オプションで、新規コンテンツサービスをデフォルトのコンテンツサービスにするには、をクリックし、[デフォルトに設定] を選択します。

コンテンツサービスからのコンテンツのインポート

既存のコンテンツサービスからコンテンツをインポートするには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [コンテンツ] > [コンテンツサービス] > [コンテンツサービスの表示]

2. 既存のコンテンツサービスのコンテンツ移行先となるコンテンツサービスを見つけて をクリックし、[コンテンツのインポート] を選択します。

3. [**ターゲットフォルダ**] プロパティに対して、以下のいずれかを行います。

   - [**参照**] をクリックし、ターゲットのページを [**選択された項目**] ボックスに移動して、[**選択**] をクリックします。

- [エイリアスを使用] をクリックし、[エイリアス名] フィールドでユーザをリダイレクトする先のページのエイリアスを入力します。[**テスト**] をクリックして、そのエイリアスが有効で、エイリアスターゲットが正しいものであるかどうかを確認します。エイリアスが正しければ、[**選択**] をクリックします。

4. [**インポート**] をクリックします。

コンテンツの最大サイズの設定

サーバリポジトリにパブリッシュされるコンテンツの最大ファイルサイズを設定するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [コンテンツ] > [コンテンツサービス] > [パブリッシュの制約事項の設定]

2. [**最大パブリッシュサイズ (MB)**] フィールドに、最大パブリッシュサイズを入力します (MB 単位)。

3. [**適用**] をクリックします。

コンテンツとして許可されるファイル拡張子の指定

デフォルトでは、サーバリポジトリに保存できるファイルタイプに制限はありません。ただし、保存できるファイルタイプを制限することができます。

コンテンツとして許可されるファイル拡張子を指定するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [コンテンツ] > [コンテンツサービス] > [パブリッシュの制約事項の設定]

2. [**許可されるファイル拡張子**] フィールドで、許可されるファイル拡張子をカンマで区切って指定します。

   デフォルト値の * では、あらゆるファイルタイプが使用可能になります。許可されるファイル拡張子を指定する場合は、次のようになります。

   ```
   zip,doc,xsl,ppt,pdf,gif,jpg,png
   ```

3. [**適用**] をクリックします。

管理者としてポートレットをパブリッシュ

[パブリッシュ] ページは、通常のユーザには一般に公開されない拡張パブリッシュ機能を、システム管理者に提供します。[パブリッシュ] ページでは、管理者が、ファイル、フォルダ、フォーム、リンク、および特定のポートレットインスタンスなどのさまざまなタイプのコンテンツをパ

ブリッシュできます。Dynamic Business Object やカスタムフォームなどのカスタムコンテンツタイプも、このページからパブリッシュできます。

[パブリッシュ] ページを使用してコンテンツをパブリッシュするには

1. システム管理者として次のように選択します。[**管理**] > [コンテンツ] > [パブリッシュ]
2. パブリッシュするコンテンツタイプに対応するオプションを選択します。デフォルトのオプションは、[ファイル]、[フォルダ]、[フォーム] (DBO のみ)、[リンク] および [ポートレット] です。
3. 特定のコンテンツタイプについて、ドロップダウンメニューからオプションの 1 つを選択します。

   注:
   以前に、特定のコンテンツタイプに基づくカスタムオブジェクトを作成してある場合は、これらのオブジェクトがそのコンテンツタイプのドロップダウンメニューのオプションとして表示されます。たとえば、[フォルダ] コンテンツタイプの下の [RSS フィード] オプションは DBO です。これは、フォルダオブジェクトタイプに RSS ニュースフィードをパブリッシュするためのカスタム属性とビジネスロジックを利用して [フォルダ] オブジェクトタイプを拡張するために作成されました。
4. [**進む**] をクリックします。
5. [場所] 見出しから、[**参照**] をクリックして、パブリッシュするコンテンツアイテムの親フォルダ位置を選択します。

   注:
   既存のエイリアスで参照されている場所にコンテンツアイテムをパブリッシュする場合は、[**エイリアスを使用**] をクリックします。
6. [**進む**] をクリックします。
7. パブリッシュするコンテンツアイテムの名前を入力します。
8. パブリッシュするコンテンツアイテムの説明を入力します (オプション)。
9. パブリッシュするコンテンツのタイプに応じて、特定のコンテンツタイプの [拡張] プロパティに入力します ([RSS フィード] コンテンツアイテムの場合は [RSS フィード URL] など)。
10. [**進む**] をクリックします。
11. [**Finish**] をクリックします。

検索インデックスの再構築

システム管理者は [検索管理] ページを使用して、My webMethods Server で提供される Lucene 検索エンジンの検索インデックスを再構築できます。

検索インデックスの再構築では、サーバにパブリッシュされたすべてのコンテンツのインデックスが再作成され、デフォルトの検索インデックスが再び更新されます。システム管理者は、検索インデックスがなんらかの理由で破損し、検索が機能しなくなった場合に、この操作を行う必要があります。

ヒント:
サーバに、コンテンツ管理システムにパブリッシュされたコンテンツが多数存在する場合、この操作の実行には長時間かかることがあります。この操作は、ピーク外の時間帯に実行してください。

検索インデックスを再同期するには

1. システム管理者として次のように選択します。[**管理**] > [設定] > [検索管理]

2. [**再構築の開始**] をクリックします。

カスタム JAR ファイルの追加

My webMethods Server は、Open Services Gateway initiative (OSGi) フレームワークで実行されます。サーバインスタンスの専用ディレクトリにカスタム JAR ファイルを追加して、`mws update` を実行してこれらの JAR ファイルを OSGi バンドルに変換することができます。update コマンドの実行中に、My webMethods Server によって、JAR ファイルのデフォルトの内容を含むバンドルのマニフェストが生成されます。専用の *Software AG_directory* /MWS/lib ディレクトリにある JAR ファイルに既に OSGi バンドルのマニフェストが含まれる場合、My webMethods Server で、そのファイルのマニフェストの生成がスキップされます。

また、JAR ファイルを別のバンドルへのフラグメントとして追加する場合などで、バンドルファイルに指示を指定することで、デフォルトのマニフェストの内容をカスタマイズできます。

以下の手順は、カスタム JAR ファイルを追加する方法および JAR ファイルをフラグメントバンドルとして添付する方法を示しています。バンドルの詳細については、『*OSGi Service Platform Core Specification*』を参照してください。

カスタム JAR ファイルを追加するには

1. JAR ファイルを以下の場所にコピーします。

   Software AG_directory /MWS/lib

2. オプション。カスタムのマニフェストを含めるか、JAR ファイルを別のバンドルへのフラグメントとして追加するには、以下の手順に従います。

a. *jar_file_name*.bnd という名前でファイルを作成します。*jar_file_name* は、.jar 拡張子を除いたカスタム JAR ファイルの名前です。

b. バインドファイルで、JAR ファイルのバインド方法に関するカスタムの指示を追加します。次に例を示します。

```
# attach as fragment to the caf.server bundle
Fragment-Host: com.webmethods.caf.server
```

c. JAR ファイルを配置した同じディレクトリにバインドファイルを配置します。

Software AG_directory /MWS/lib

3. サーバインスタンスに対して update コマンドを実行します。

```
Software AG_directory/MWS/bin/mws.[bat | sh] update
```

19 ユーザインタフェースの管理

- ロケール管理 354
- サーバルールとは 354
- ロケールルールの作成 356
- ログインページルールの作成 358
- スタートページルールの作成 360
- レンダリングルールの作成 362
- ルールの変更 365
- ルールのコピー 366
- ルールの評価順序の管理 367
- ルールの削除 368
- スキンルールの管理 368
- シェルルールの管理 371

ロケール管理

My webMethods Server には、いくつかの決定ポイントがあり、ユーザのロケールは次の順序で決定されます。

1. ユーザのロケール設定。[ユーザプロファイル] に指定されている場合。詳細については、157 ページの「ユーザ情報」を参照してください。
2. ロケールルール。詳細については、次のセクションを参照してください。356 ページの「ロケールルールの作成」
3. ブラウザのロケール設定。
4. サーバが稼働しているコンピュータのシステムロケール。これがデフォルトです。

他のルールで定義されていない場合は、My webMethods Serverの手順に従って、[ロケール管理] ページを使用して、356 ページの「ロケールルールの作成」 のデフォルトのロケールルールを設定できます。

注:
Linux 環境で My webMethods Server をデーモンサービスとして実行する場合は、*Software AG_directory* /profiles/MWS_*instanceName*/bin/sagmws_*version* スクリプトファイルで、LANG 変数でロケールを設定します (例: `export LANG=es_ES.UTF-8`)。

My webMethods Server のデフォルトのロケールルールを設定するには

1. システム管理者として次のように選択します。[**管理**] > [ユーザインタフェース] > [ロケール管理]
2. [**ロケール**] リストで、My webMethods Server のデフォルトルールとして使用するロケールを選択します。
3. [**適用**] をクリックします。

サーバルールとは

My webMethods Server では、ユーザがどのページを使用してサーバにログインするか、またはページがどのような外観でユーザに提示されるかなど、ユーザのさまざまな活動を制御するためのルールが使用されています。ルールを使用すれば、アプリケーション全体のデフォルトの動作を定義したり、特定のユーザ、グループ、あるいは役割の環境を動的に制御することができます。

次の表は、My webMethods でユーザが作成可能なルールのタイプを示しています。

ルールのタイプ	説明
ロケールルール	ユーザプロファイルで定義されていない場合に、ユーザセッションで使用されるロケールを指示するためのルール。ロケールルールの作成の詳細については、356 ページの「ロケールルールの作成」を参照してください。
ログインページルール	使用するログインページを決定するルール。たとえば、ユーザがファイアウォールの内側にいるか外側にいるかに応じて、それらのユーザを異なるログインページにリダイレクトできます。ログインページルールの作成の詳細については、358 ページの「ログインページルールの作成」を参照してください。
スタートページルール	使用するスタートページを決定するルール。スタートページとは、ユーザがログイン後にサーバによりリダイレクトされるページです。スタートページルールの作成の詳細については、360 ページの「スタートページルールの作成」を参照してください。
レンダリングルール	使用するレンダラを決定するルール。レンダラは、ユーザインタフェース書式設定機能で、レンダリングルールを定義することによって、特定のサーバオブジェクトに割り当てることができます。 事実上すべてのサーバオブジェクトタイプに対して、レンダリングルールを定義できます。レンダリングルールは、明示的なルール定義を通じて呼び出すことのできる一般的なオブジェクトタイプに、一貫したルックアンドフィールを提供するのに役立ちます。レンダリングルールの作成の詳細については、362 ページの「レンダリングルールの作成」を参照してください。
スキンルール	使用するスキンを決定するルール。スキンとは、ユーザインタフェースのルックアンドフィールを定義する、インストール可能な My webMethods Server コンポーネントです。スキンルールは、特定のユーザ、グループ、またはサーバリソースに対してどのスキンを表示するかを定義します。たとえば、1 つのサーバで従業員と顧客の両方を処理し、各種のユーザに対して異なるグラフィック、色、およびフォントのセットの要件がある場合に、スキンルールを定義して、特定のユーザグループに対応するスキンを割り当てることができます。スキンルールの作成の詳細については、368 ページの「スキンルールの作成」を参照してください。
シェルルール	使用するシェルを決定するルール。シェルとは、My webMethods Server ヘッダー、フッター、およびポートレットのタイトルバーを生成するインストール可能なコンポーネントです。シェルルールは、特定のユーザ、グループ、または役割に対してどのシェルエレメントを表示するかを定義します。たとえば、1 つのサーバで従業員と顧客の両方を処理する場合、シェルルールを使用して特定のユーザグループに、対応するシェルを割り当てることができます。シェルルールの作成の詳細については、371 ページの「シェルルールの作成」を参照してください。

さらに、役割 (ユーザ、グループ、およびその他の役割の集合) を作成するためのルールを使用できます。ルールタイプに使用される条件と同じタイプの条件に基づいてメンバーを定義する、

ルールベースの役割を作成できます。詳細については、195 ページの「ルールベース役割の追加」を参照してください。

ロケールルールの作成

ユーザプロファイルで定義されていない場合は、[ロケールルールの管理] ページを使用して、ユーザセッションで使用されるロケールを指示できます。

新規ロケールルールを作成するには

1. システム管理者として次のように選択します。[**管理**] > [ユーザインタフェース] > [ロケールルールの管理] > [新規ルールの作成]

2. [**名前**] フィールドに、ルールの名前を入力します。

3. [**説明**] フィールドに、新規ルールの説明を入力します (オプション)。

4. 作成中にルールを無効にしない場合は、[**有効**] を選択したままにします。

5. 個々のユーザ単位で条件を追加するには、[**現在のユーザ**] をクリックして、以下の手順に従います。

   a. [**キーワード**] フィールドで、検索するユーザを表すキーワードを入力し、[**検索**] をクリックします。

   b. 1 つ以上のユーザを [**選択済み**] ボックスに移動し、[**適用**] をクリックします。

6. グループまたは役割のメンバーシップに基づいて条件を追加するには、[**グループ/役割メンバーシップ**] をクリックして、以下の手順に従います。

   a. [**検索対象**] で、[**グループ**]、または [**役割**] オプションを選択します。

   b. [**キーワード**] フィールドで、検索するグループまたは役割を表すキーワードを入力し、[**検索**] をクリックします。

   c. 1 つ以上のグループまたは役割を [**選択済み**] ボックスに移動し、[**適用**] をクリックします。

7. ユーザ属性に基づいて条件を追加するには、[**ユーザ属性の選択**] リストの [**ユーザ属性**] をクリックし、ユーザ属性を選択して、[**適用**] をクリックします。

   このリスト内のユーザ属性の詳細については、157 ページの「ユーザ情報」を参照してください。

8. グローバルセッション属性に基づいて条件を追加するには、[**グローバルセッション**] をクリックして、以下の手順に従います。

a. [**グローバルセッションの属性名**] フィールドで、グローバルセッション属性の名前を入力します。

b. [**変数値**] フィールドで、グローバルセッション属性の値を入力します。

c. [**適用**] をクリックします。

9. 要求ヘッダーに基づいて条件を追加するには、[**要求**] をクリックして、以下の手順に従います。

a. 式を選択して、[**サブミット**] をクリックします。

b. [**条件**] フィールドで、式を完成します。

たとえば、任意の HTTP GET 要求を一致させるとします。[**条件**] フィールドにはウィザードによって `#{request.method}` が表示されます。残りの情報はユーザが入力します。

```
#{request.method} == "GET"
```

10. 現在のリソースまたは現在のリソースの親を照合する条件を追加するには、[**親リソース**] をクリックして、以下の手順に従います。

a. リソースの子を検索するには、リソースの名前をクリックします。

b. リソースを選択するには、リソースの左にあるオプションボタンをクリックします。

c. [**適用**] をクリックします。

11. 現在のリソースタイプを照合する条件を追加するには、[**現在のリソースタイプ**] をクリックし、リソースタイプを選択して、以下の手順に従います。

a. リソースの子を検索するには、リソースの名前をクリックします。

b. リソースを選択するには、リソースの左にあるオプションボタンをクリックします。

c. [**適用**] をクリックします。

12. リソースプロパティおよびその値に関する条件を追加するには、[**リソースプロパティ**] をクリックして、以下の手順に従います。

a. [**プロパティ名**] フィールドに、プロパティ名を入力します。

b. [**プロパティ値**] フィールドに、プロパティの値を入力します。

c. [**適用**] をクリックします。

たとえば、PDF 形式のファイルを一致させたい場合、プロパティ名は mimeType、プロパティ値は pdf になります。

13. [**結果**] リストで、ルールで使用するロケールを選択します。

14. [**ルールの作成**] をクリックします。

ログインページルールの作成

[ログインページルールの管理] ページでは、どのログインページを使用するかを指示するルールを定義できます。ログインページルールを定義して、特定のユーザ、グループ、または役割のデフォルトのログインページを動的に設定できます。

新規ログインページルールを作成するには

1. システム管理者として次のように選択します。[**管理**] > [ユーザインタフェース] > [ログインページルールの管理] > [新規ルールの作成]

2. [**名前**] フィールドに、ルールの名前を入力します。

3. [**説明**] フィールドに、新規ルールの説明を入力します (オプション)。

4. 作成中にルールを無効にしない場合は、[**有効**] を選択したままにします。

5. 個々のユーザ単位で条件を追加するには、[**現在のユーザ**] をクリックして、以下の手順に従います。

   a. [**キーワード**] フィールドで、検索するユーザを表すキーワードを入力し、[**検索**] をクリックします。

   b. 1 つ以上のユーザを [**選択済み**] ボックスに移動し、[**適用**] をクリックします。

6. グループまたは役割のメンバーシップに基づいて条件を追加するには、[**グループ/役割メンバーシップ**] をクリックして、以下の手順に従います。

   a. [**検索対象**] で、[**グループ**]、または [**役割**] オプションを選択します。

   b. [**キーワード**] フィールドで、検索するグループまたは役割を表すキーワードを入力し、[**検索**] をクリックします。

   c. 1 つ以上のグループまたは役割を [**選択済み**] ボックスに移動し、[**適用**] をクリックします。

7. ユーザ属性に基づいて条件を追加するには、[**ユーザ属性の選択**] リストの [**ユーザ属性**] をクリックし、ユーザ属性を選択して、[**適用**] をクリックします。

このリスト内のユーザ属性の詳細については、157 ページの「ユーザ情報」を参照してください。

8. グローバルセッション属性に基づいて条件を追加するには、[**グローバルセッション**] をクリックして、以下の手順に従います。

   a. [**グローバルセッションの属性名**] フィールドで、グローバルセッション属性の名前を入力します。

   b. [**変数値**] フィールドで、グローバルセッション属性の値を入力します。

   c. [**適用**] をクリックします。

9. 要求ヘッダーに基づいて条件を追加するには、[**要求**] をクリックして、以下の手順に従います。

   a. 式を選択して、[**サブミット**] をクリックします。

   b. [**条件**] フィールドで、式を完成します。

   たとえば、任意の HTTP GET 要求を一致させるとします。[**条件**] フィールドにはウィザードによって `#{request.method}` が表示されます。残りの情報はユーザが入力します。

   ```
   #{request.method} == "GET"
   ```

10. 現在のリソースまたは現在のリソースの親を照合する条件を追加するには、[**親リソース**] をクリックして、以下の手順に従います。

    a. リソースの子を検索するには、リソースの名前をクリックします。

    b. リソースを選択するには、リソースの左にあるオプションボタンをクリックします。

    c. [**適用**] をクリックします。

11. 現在のリソースタイプを照合する条件を追加するには、[**現在のリソースタイプ**] をクリックし、リソースタイプを選択して、以下の手順に従います。

    a. リソースの子を検索するには、リソースの名前をクリックします。

    b. リソースを選択するには、リソースの左にあるオプションボタンをクリックします。

    c. [**適用**] をクリックします。

12. リソースプロパティおよびその値に関する条件を追加するには、[**リソースプロパティ**] をクリックして、以下の手順に従います。

    a. [**プロパティ名**] フィールドに、プロパティ名を入力します。

b. [**プロパティ値**] フィールドに、プロパティの値を入力します。

c. [**適用**] をクリックします。

たとえば、PDF 形式のファイルを一致させたい場合、プロパティ名は `mimeType`、プロパティ値は `pdf` になります。

13. [**結果**] フィールドで、ログインページのエイリアスを入力するか、[**選択**] をクリックしてページを参照して、以下の手順に従います。

a. リソースの子を検索するには、リソースの名前をクリックします。

b. リソースを選択するには、リソースの左にあるオプションボタンをクリックします。

c. [**適用**] をクリックします。

14. [**ルールの作成**] をクリックします。

スタートページルールの作成

[スタートページルールの管理] ページでは、システム管理者が、どのスタートページを使用するかを指示するルールを定義できます。スタートページとは、ユーザがログイン後にサーバによりリダイレクトされるページです。スタートページルールを定義して、特定のユーザ、グループ、または役割のデフォルトのスタートページを動的に設定できます。

新規スタートページルールを作成するには

1. システム管理者として次のように選択します。[**管理**] > [ユーザインタフェース] > [スタートページルールの管理] > [新規ルールの作成]

2. [**名前**] フィールドに、ルールの名前を入力します。

3. [**説明**] フィールドに、新規ルールの説明を入力します (オプション)。

4. 作成中にルールを無効にしない場合は、[**有効**] を選択したままにします。

5. 個々のユーザ単位で条件を追加するには、[**現在のユーザ**] をクリックして、以下の手順に従います。

a. [**キーワード**] フィールドで、検索するユーザを表すキーワードを入力し、[**検索**] をクリックします。

b. 1 つ以上のユーザを [**選択済み**] ボックスに移動し、[**適用**] をクリックします。

6. グループまたは役割のメンバーシップに基づいて条件を追加するには、[**グループ/役割メンバーシップ**] をクリックして、以下の手順に従います。

   a. [**検索対象**] で、[**グループ**]、または [**役割**] オプションを選択します。

   b. [**キーワード**] フィールドで、検索するグループまたは役割を表すキーワードを入力し、[**検索**] をクリックします。

   c. 1 つ以上のグループまたは役割を [**選択済み**] ボックスに移動し、[**適用**] をクリックします。

7. ユーザ属性に基づいて条件を追加するには、[**ユーザ属性の選択**] リストの [**ユーザ属性**] をクリックし、ユーザ属性を選択して、[**適用**] をクリックします。

   このリスト内のユーザ属性の詳細については、157 ページの「ユーザ情報」を参照してください。

8. グローバルセッション属性に基づいて条件を追加するには、[**グローバルセッション**] をクリックして、以下の手順に従います。

   a. [**グローバルセッションの属性名**] フィールドで、グローバルセッション属性の名前を入力します。

   b. [**変数値**] フィールドで、グローバルセッション属性の値を入力します。

   c. [**適用**] をクリックします。

9. 要求ヘッダーに基づいて条件を追加するには、[**要求**] をクリックして、以下の手順に従います。

   a. 式を選択して、[**サブミット**] をクリックします。

   b. [**条件**] フィールドで、式を完成します。

   たとえば、任意の HTTP GET 要求を一致させるとします。[**条件**] フィールドにはウィザードによって `#{request.method}` が表示されます。残りの情報はユーザが入力します。

   ```
   #{request.method} == "GET"
   ```

10. 現在のリソースまたは現在のリソースの親を照合する条件を追加するには、[**親リソース**] をクリックして、以下の手順に従います。

    a. リソースの子を検索するには、リソースの名前をクリックします。

    b. リソースを選択するには、リソースの左にあるオプションボタンをクリックします。

    c. [**適用**] をクリックします。

11. 現在のリソースタイプを照合する条件を追加するには、[**現在のリソースタイプ**] をクリックし、リソースタイプを選択して、以下の手順に従います。

    a. リソースの子を検索するには、リソースの名前をクリックします。

    b. リソースを選択するには、リソースの左にあるオプションボタンをクリックします。

    c. [**適用**] をクリックします。

12. リソースプロパティおよびその値に関する条件を追加するには、[**リソースプロパティ**] をクリックして、以下の手順に従います。

    a. [**プロパティ名**] フィールドに、プロパティ名を入力します。

    b. [**プロパティ値**] フィールドに、プロパティの値を入力します。

    c. [**適用**] をクリックします。

    たとえば、PDF 形式のファイルを一致させたい場合、プロパティ名は `mimeType`、プロパティ値は `pdf` になります。

13. [**結果**] フィールドで、スタートページのエイリアスを入力するか、[**選択**] をクリックしてページを参照して、以下の手順に従います。

    a. リソースの子を検索するには、リソースの名前をクリックします。

    b. リソースを選択するには、リソースの左にあるオプションボタンをクリックします。

    c. [**適用**] をクリックします。

14. [**ルールの作成**] をクリックします。

レンダリングルールの作成

[レンダリングルールの管理] ページでは、システム管理者が、フォルダ、ページ、ポートレットなどの特定のサーバオブジェクトのレンダリングルールを設定できます。たとえば、すべてのフォルダでコンテンツの詳細ビューを表示する必要のある管理者は、すべてのフォルダオブジェクトに「details」レンダラを適用するレンダリングルールを作成できます。以下の手順で適用可能な各条件タイプでは、複数のインスタンスを、一度に 1 つずつ追加できます。

新規レンダリングルールを作成するには

1. システム管理者として次のように選択します。[**管理**] > [ユーザインタフェース] > [レンダリングルールの管理] > [新規ルールの作成]

2. [**名前**] フィールドに、ルールの名前を入力します。

   例: `folder-thumbnail view` (イメージファイルの場合)

3. [**説明**] フィールドに、新規ルールの説明を入力します (オプション)。

4. 作成中にルールを無効にしない場合は、[**有効**] を選択したままにします。

5. 個々のユーザ単位で条件を追加するには、[**現在のユーザ**] をクリックして、以下の手順に従います。

   a. [**キーワード**] フィールドで、検索するユーザを表すキーワードを入力し、[**検索**] をクリックします。

   b. 1 つ以上のユーザを [**選択済み**] ボックスに移動し、[**適用**] をクリックします。

6. グループまたは役割のメンバーシップに基づいて条件を追加するには、[**グループ/役割メンバーシップ**] をクリックして、以下の手順に従います。

   a. [**検索対象**] で、[**グループ**]、または [**役割**] オプションを選択します。

   b. [**キーワード**] フィールドで、検索するグループまたは役割を表すキーワードを入力し、[**検索**] をクリックします。

   c. 1 つ以上のグループまたは役割を [**選択済み**] ボックスに移動し、[**適用**] をクリックします。

7. ユーザ属性に基づいて条件を追加するには、[**ユーザ属性の選択**] リストの [**ユーザ属性**] をクリックし、ユーザ属性を選択して、[**適用**] をクリックします。

   このリスト内のユーザ属性の詳細については、157 ページの「ユーザ情報」を参照してください。

8. グローバルセッション属性に基づいて条件を追加するには、[**グローバルセッション**] をクリックして、以下の手順に従います。

   a. [**グローバルセッションの属性名**] フィールドで、グローバルセッション属性の名前を入力します。

   b. [**変数値**] フィールドで、グローバルセッション属性の値を入力します。

   c. [**適用**] をクリックします。

9. 要求ヘッダーに基づいて条件を追加するには、[**要求**] をクリックして、以下の手順に従います。

   a. 式を選択して、[**サブミット**] をクリックします。

b. [**条件**] フィールドで、式を完成します。

たとえば、任意の HTTP GET 要求を一致させるとします。[**条件**] フィールドにはウィザードによって `#{request.method}` が表示されます。残りの情報はユーザが入力します。

```
#{request.method} == "GET"
```

10. 現在のリソースまたは現在のリソースの親を照合する条件を追加するには、[**親リソース**] をクリックして、以下の手順に従います。

a. リソースの子を検索するには、リソースの名前をクリックします。

b. リソースを選択するには、リソースの左にあるオプションボタンをクリックします。

c. [**適用**] をクリックします。

11. 現在のリソースタイプを照合する条件を追加するには、[**現在のリソースタイプ**] をクリックし、リソースタイプを選択して、以下の手順に従います。

a. リソースの子を検索するには、リソースの名前をクリックします。

b. リソースを選択するには、リソースの左にあるオプションボタンをクリックします。

c. [**適用**] をクリックします。

12. リソースプロパティおよびその値に関する条件を追加するには、[**リソースプロパティ**] をクリックして、以下の手順に従います。

a. [**プロパティ名**] フィールドに、プロパティ名を入力します。

b. [**プロパティ値**] フィールドに、プロパティの値を入力します。

c. [**適用**] をクリックします。

たとえば、PDF 形式のファイルを一致させたい場合、プロパティ名は `mimeType`、プロパティ値は `pdf` になります。

13. [**結果**] リストで、ターゲットレンダラを選択します。

選択するレンダラは、次の手順で定義する評価条件に一致するすべてのサーバオブジェクトに適用されます。たとえば、[thumbnails] レンダラは、サーバにパブリッシュされるイメージのサムネイルビューを表示する際に役立ちます。

14. [**ルールの作成**] をクリックします。

ルールの変更

ルールが作成された後は、編集可能な任意の部分を変更できます。

ルールを変更するには

1. システム管理者として次のように選択します。[**管理**] > [ユーザインタフェース] > [*rule-type* ルールの管理] > [ルールの表示]

   rule-type は、変更するルールを表します。

2. 変更するルールの をクリックして、[ルールの変更] を選択します。

3. 次のいずれかの手順に従います。

   - [**名前**] フィールドに、ルールの新規の名前を入力します。
   - [**説明**] フィールドに、ルールの新規の説明を入力します (オプション)。
   - ルールを無効にするには、[**有効**] をクリアします。
   - 個々のユーザ単位で条件を変更するには、[**現在のユーザ**] をクリックして、以下の手順に従います。

     1. [**キーワード**] フィールドで、検索するユーザを表すキーワードを入力し、[**検索**] をクリックします。
     2. 1 つ以上のユーザを [**選択済み**] ボックスに移動し、[**適用**] をクリックします。

   - グループまたは役割のメンバーシップに基づいて条件を変更するには、[**グループ/役割メンバーシップ**] をクリックして、以下の手順に従います。

     1. [**検索対象**] で、[**グループ**]、または [**役割**] オプションを選択します。
     2. [**キーワード**] フィールドで、検索するグループまたは役割を表すキーワードを入力し、[**検索**] をクリックします。
     3. 1 つ以上のグループまたは役割を [**選択済み**] ボックスに移動し、[**適用**] をクリックします。

   - ユーザ属性に基づいて条件を変更するには、[**ユーザ属性の選択**] リストの [**ユーザ属性**] をクリックし、ユーザ属性を選択して、[**適用**] をクリックします。

     このリスト内のユーザ属性の詳細については、157 ページの「ユーザ情報」を参照してください。

   - グローバルセッション属性に基づいて条件を変更するには、[**グローバルセッション**] をクリックして、以下の手順に従います。

     1. [**グローバルセッションの属性名**] フィールドで、グローバルセッション属性の名前を入力します。

2. [**変数値**] フィールドで、グローバル属性の新しい値を入力します。
3. [**適用**] をクリックします。

- 要求ヘッダーに基づいて条件を変更するには、[**要求**] をクリックして、以下の手順に従います。

  1. 式を選択して、[**サブミット**] をクリックします。
  2. [**条件**] フィールドで、式を完成します。

  たとえば、任意の HTTP GET 要求を一致させるとします。[**条件**] フィールドにはウィザードによって `#{request.method}` が表示されます。残りを「`#{request.method} == "GET"`」と入力します。

- 現在のリソースまたは現在のリソースの親を照合する条件を変更するには、[**親リソース**] をクリックして、以下の手順に従います。

  1. リソースの子を検索するには、リソースの名前をクリックします。
  2. リソースを選択するには、リソースの左にあるオプションボタンをクリックします。
  3. [**適用**] をクリックします。

- 現在のリソースタイプを照合する条件を変更するには、[**現在のリソースタイプ**] をクリックし、リソースタイプを選択して、以下の手順に従います。

  1. リソースの子を検索するには、リソースの名前をクリックします。
  2. リソースを選択するには、リソースの左にあるオプションボタンをクリックします。
  3. [**適用**] をクリックします。

- リソースプロパティおよびその値に関する条件を変更するには、[**リソースプロパティ**] をクリックして、以下の手順に従います。

  1. [**プロパティ名**] フィールドに、プロパティ名を入力します。
  2. [**プロパティ値**] フィールドに、プロパティの値を入力します。
  3. [**適用**] をクリックします。

- [**結果**] フィールドを必要に応じて変更します。

4. [**ルールの更新**] をクリックします。

ルールのコピー

既存のルールと類似したルールを作成する場合、既存のルールのコピーを作成することで行えます。

ルールをコピーするには

1. システム管理者として次のように選択します。[**管理**] > [ユーザインタフェース] > [*rule-type* ルールの管理] > [ルールの表示]

   rule-type は、コピーするルールを表します。

2. コピーするルールの をクリックして、[ルールのコピー] を選択します。

3. 新規ルールの名前を入力します。

4. 新規ルールの説明を入力します (オプション)。

5. [**ルールのコピー**] をクリックします。

   365 ページの「ルールの変更」で説明しているように、新規ルールを変更できます。

ルールの評価順序の管理

ユーザがサーバリソースを要求すると、サーバではルールを使用して、要求を満たす方法を決定します。たとえば、ページのルックアンドフィールが、ユーザが会社の従業員であるか顧客であるかに応じて決まるとします。サーバでは、スキンルールを評価して、次の順序で適用するスキンを決定します。

1. 複数のスキンルールがある場合、ユーザに一致する最初のルールに関連付けられたスキンが適用されます。
2. ユーザに一致するルールがないか、スキンルールがない場合、ユーザのプロファイルページの [**ユーザ設定**] タブで割り当てられたデフォルトのスキンが適用されます。
3. プロファイルページでスキンが割り当てられていない場合、サーバのデフォルトのスキンが適用されます。

ルールの評価順序の変更

複数のスキンルールがある場合、ルールが評価される順序を決定できます。

ルールが評価される順序を変更するには

1. システム管理者として次のように選択します。[**管理**] > [ユーザインタフェース]
2. 管理するルールセットの名前をクリックします。
3. [**ルール評価順序の変更**] をクリックします。
4. ルールの順序を変更するには、必要に応じてルールを上下に移動します。

   リストのルールが最初に検索され、2 番目以降のルールが順次検索されます。

5. [**更新**] をクリックします。

ルールの削除

ルールを削除するには、以下の手順に従います。

ルールを削除するには

1. システム管理者として次のように選択します。[**管理**] > [ユーザインタフェース] > [*rule-type* ルールの管理] > [ルールの表示]

   rule-type は、削除するルールを表します。

2. 削除するルールの をクリックして、[ルールの削除] を選択します。

スキンルールの管理

スキンとは、My webMethods Server ユーザインタフェースのルックアンドフィールを定義する、インストール可能な My webMethods Server コンポーネントです。HTML コンテンツのイメージ、フォント、色、およびその他の微妙なスタイル設定可能な要素はスキンによって変わりますが、HTML コンテンツの機能が変わることはありません。

開発者は、新規カスタムスキンを作成してさまざまな機能を実行します。次のようなタスクがあります。

- 企業、パートナ、または部門のロゴを使用してサーバのブランドを設定
- 企業カラー、パートナカラー、または部門カラーでの色体系を調整

開発者は [スキン管理] ページを使用してスキンを作成し、必要に応じてカスタマイズします。スキンをカスタマイズする方法の詳細については、441 ページの「スキンのカスタマイズ」を参照してください。

My webMethods Server には、特定のユーザ、グループ、またはリソースに対して表示されるスキンを決定するパーソナライゼーションルールを設定するためのさまざまな方法が用意されています。特定のユーザに特定のスキンを明示的に割り当てることも、さまざまな条件に基づいてスキンを動的に割り当てるルールを設定することもできます。

[スキンルールの管理] ページを使用すると、ユーザ、グループ、または役割が使用するスキンを決定するルールを定義できます。システム管理者はこのページで、各ユーザがログインするたびにそのユーザについて評価されるルールの一覧を作成、変更、削除したり、評価順序を変更したりできます。

スキンルールの作成

新規スキンルールを作成するには、以下の手順に従います。

新規スキンルールを作成するには

1. システム管理者として次のように選択します。[**管理**] > [ユーザインタフェース] > [スキンルールの管理] > [新規ルールの作成]

2. [**名前**] フィールドに、ルールの名前を入力します。

3. [**説明**] フィールドに、新規ルールの説明を入力します (オプション)。

4. 作成中にルールを無効にしない場合は、[**有効**] を選択したままにします。

5. 個々のユーザ単位で条件を追加するには、[**現在のユーザ**] をクリックして、以下の手順に従います。

   a. [**キーワード**] フィールドで、検索するユーザを表すキーワードを入力し、[**検索**] をクリックします。

   b. 1 つ以上のユーザを [**選択済み**] ボックスに移動し、[**適用**] をクリックします。

6. グループまたは役割のメンバーシップに基づいて条件を追加するには、[**グループ/役割メンバーシップ**] をクリックして、以下の手順に従います。

   a. [**検索対象**] で、[**グループ**]、または [**役割**] オプションを選択します。

   b. [**キーワード**] フィールドで、検索するグループまたは役割を表すキーワードを入力し、[**検索**] をクリックします。

   c. 1 つ以上のグループまたは役割を [**選択済み**] ボックスに移動し、[**適用**] をクリックします。

7. ユーザ属性に基づいて条件を追加するには、[**ユーザ属性の選択**] リストの [**ユーザ属性**] をクリックし、ユーザ属性を選択して、[**適用**] をクリックします。

   このリスト内のユーザ属性の詳細については、157 ページの「ユーザ情報」を参照してください。

8. グローバルセッション属性に基づいて条件を追加するには、[**グローバルセッション**] をクリックして、以下の手順に従います。

   a. [**グローバルセッションの属性名**] フィールドで、グローバルセッション属性の名前を入力します。

   b. [**変数値**] フィールドで、グローバルセッション属性の値を入力します。

   c. [**適用**] をクリックします。

9. 要求ヘッダーに基づいて条件を追加するには、[**要求**] をクリックして、以下の手順に従います。

   a. 式を選択して、[**サブミット**] をクリックします。

   b. [**条件**] フィールドで、式を完成します。

   たとえば、任意の HTTP GET 要求を一致させるとします。[**条件**] フィールドにはウィザードによって `#{request.method}` が表示されます。残りの情報はユーザが入力します。

   ```
   #{request.method} == "GET"
   ```

10. 現在のリソースまたは現在のリソースの親を照合する条件を追加するには、[**親リソース**] をクリックして、以下の手順に従います。

    a. リソースの子を検索するには、リソースの名前をクリックします。

    b. リソースを選択するには、リソースの左にあるオプションボタンをクリックします。

    c. [**適用**] をクリックします。

11. 現在のリソースタイプを照合する条件を追加するには、[**現在のリソースタイプ**] をクリックし、リソースタイプを選択して、以下の手順に従います。

    a. リソースの子を検索するには、リソースの名前をクリックします。

    b. リソースを選択するには、リソースの左にあるオプションボタンをクリックします。

    c. [**適用**] をクリックします。

12. リソースプロパティおよびその値に関する条件を追加するには、[**リソースプロパティ**] をクリックして、以下の手順に従います。

    a. [**プロパティ名**] フィールドに、プロパティ名を入力します。

    b. [**プロパティ値**] フィールドに、プロパティの値を入力します。

    c. [**適用**] をクリックします。

    たとえば、PDF 形式のファイルを一致させたい場合、プロパティ名は `mimeType`、プロパティ値は `pdf` になります。

13. [**結果**] リストで、ターゲットスキンを選択します。

14. [**ルールの作成**] をクリックします。

シェルルールの管理

シェルは、My webMethods Server のインストール可能なコンポーネントです。シェルセグメントは、My webMethods Server のヘッダー、フッター、およびポートレットタイトルバーを生成する特殊な種類のページです。

通常のポートレットはページのプライマリコンテンツを生成しますが、シェルはそのプライマリコンテンツをフレーム化する構造を提供します。バナー、標準リンク、検索ボックスなどの一般的な Web ページ要素がシェルに表示されます。

開発者は、新規カスタムシェルを作成して、次のようなさまざまな機能を実行します。

- ページバナーの下に他社の Web サイトへのリンク行を追加
- デフォルトの検索ボックスを、企業カタログを検索するボックスに変更
- すべてのページに左側のナビゲーションバーを追加

特定のユーザの要求に対して使用するシェルを決定するための条件を多数設定できます。My webMethods Server には、特定のユーザ、グループ、またはリソースに対して表示されるシェルを決定するパーソナライゼーションルールを設定するためのさまざまな方法が用意されています。

注:
スキンと異なり、特定のユーザに特定のシェルを明示的に割り当てることはできません。さまざまな条件に基づいて動的にシェルを割り当てるルールを使用する必要があります。

[シェルルールの管理] ページを使用すると、ユーザ、グループ、または役割が使用するシェルを決定するルールを定義できます。システム管理者はこのページで、各ユーザがログインするたびにそのユーザについて評価されるルールを作成、変更、削除したり、ルールリストの評価順序を変更したりできます。

シェルルールの作成

新規シェルルールを作成するには、以下の手順に従います。

新規シェルルールを作成するには

1. システム管理者として次のように選択します。[**管理**] > [ユーザインタフェース] > [シェルルールの管理] > [新規ルールの作成]

2. [**名前**] フィールドに、ルールの名前を入力します。

   例: `folder-thumbnail view` (イメージファイルの場合)

3. [**説明**] フィールドに、新規ルールの説明を入力します (オプション)。

4. 作成中にルールを無効にしない場合は、[**有効**] を選択したままにします。

5. 個々のユーザ単位で条件を追加するには、[**現在のユーザ**] をクリックして、以下の手順に従います。

   a. [**キーワード**] フィールドで、検索するユーザを表すキーワードを入力し、[**検索**] をクリックします。

   b. 1 つ以上のユーザを [**選択済み**] ボックスに移動し、[**適用**] をクリックします。

6. グループまたは役割のメンバーシップに基づいて条件を追加するには、[**グループ/役割メンバーシップ**] をクリックして、以下の手順に従います。

   a. [**検索対象**] で、[**グループ**]、または [**役割**] オプションを選択します。

   b. [**キーワード**] フィールドで、検索するグループまたは役割を表すキーワードを入力し、[**検索**] をクリックします。

   c. 1 つ以上のグループまたは役割を [**選択済み**] ボックスに移動し、[**適用**] をクリックします。

7. ユーザ属性に基づいて条件を追加するには、[**ユーザ属性の選択**] リストの [**ユーザ属性**] をクリックし、ユーザ属性を選択して、[**適用**] をクリックします。

   このリスト内のユーザ属性の詳細については、157 ページの「ユーザ情報」を参照してください。

8. グローバルセッション属性に基づいて条件を追加するには、[**グローバルセッション**] をクリックして、以下の手順に従います。

   a. [**グローバルセッションの属性名**] フィールドで、グローバルセッション属性の名前を入力します。

   b. [**変数値**] フィールドで、グローバルセッション属性の値を入力します。

   c. [**適用**] をクリックします。

9. 要求ヘッダーに基づいて条件を追加するには、[**要求**] をクリックして、以下の手順に従います。

   a. 式を選択して、[**サブミット**] をクリックします。

   b. [**条件**] フィールドで、式を完成します。

   たとえば、任意の HTTP GET 要求を一致させるとします。[**条件**] フィールドにはウィザードによって `#{request.method}` が表示されます。残りの情報はユーザが入力します。

```
#{request.method} == "GET"
```

10. 現在のリソースまたは現在のリソースの親を照合する条件を追加するには、[**親リソース**] をクリックして、以下の手順に従います。

    a. リソースの子を検索するには、リソースの名前をクリックします。

    b. リソースを選択するには、リソースの左にあるオプションボタンをクリックします。

    c. [**適用**] をクリックします。

11. 現在のリソースタイプを照合する条件を追加するには、[**現在のリソースタイプ**] をクリックし、リソースタイプを選択して、以下の手順に従います。

    a. リソースの子を検索するには、リソースの名前をクリックします。

    b. リソースを選択するには、リソースの左にあるオプションボタンをクリックします。

    c. [**適用**] をクリックします。

12. リソースプロパティおよびその値に関する条件を追加するには、[**リソースプロパティ**] をクリックして、以下の手順に従います。

    a. [**プロパティ名**] フィールドに、プロパティ名を入力します。

    b. [**プロパティ値**] フィールドに、プロパティの値を入力します。

    c. [**適用**] をクリックします。

    たとえば、PDF 形式のファイルを一致させたい場合、プロパティ名は `mimeType`、プロパティ値は `pdf` になります。

13. [**結果**] リストで、ターゲットシェルを選択します。

    選択するレンダラは、次の手順で定義する評価条件に一致するすべてのサーバオブジェクトに適用されます。たとえば、[thumbnails] レンダラは、サーバにパブリッシュされるイメージのサムネイルビューを表示する際に役立ちます。

14. [**ルールの作成**] をクリックします。

要求に対するシェルの設定

要求に対する特定のシェルを設定するには、開発者がサーバリソースへのリンクを作成し、そのリンクにシェルパラメータを追加します。シェルパラメータ値は、ターゲットシェルに対するエイリアスである必要があります。

たとえば、エクストラネットシェルを持つパブリックフォルダへのリンクの URL が /folder.public?shell=shell.extranet であるとします。ユーザがリンクをたどると、指定されたエ

クストラネットシェルでフレーム化されたパブリックフォルダが表示されます。リンクにシェルパラメータもない場合は、ユーザがパブリックフォルダページから別のリンクをクリックすると、直前に使用していたシェルに戻ります。

セッション用シェルの設定

セッション用の特定のシェル (セッションの持続中は設定が保持されるため**スティッキー**シェルと呼ばれることもあります) を設定するには、開発者が forceShell コマンドへのリンクを作成します。シェルを設定しておくと、このコマンドは returnUrl パラメータを受け取り、ユーザは指定された URL にリダイレクトされます。

たとえば、スティッキーエクストラネットシェルを持つパブリックフォルダへのリンクの URL が /?command=forceShell&shellURI=shell.extranet&returnUrl=folder.public であるとします。ユーザがリンクをたどると、エクストラネットシェルでフレーム化されたパブリックフォルダが表示されます。ユーザがパブリックフォルダページから別のリンクをクリックすると、(リンクにシェルパラメータがなければ) エクストラネットシェルでフレーム化されたページが表示されます。

20 共通ディレクトリサービス API の使用

- 共通ディレクトリサービス API によるユーザ情報の管理 376
- 共通ディレクトリサービス API について 377
- 前提条件 377
- CDS のバージョンの相互運用性について 378
- CDS コード例 379

共通ディレクトリサービス APIによるユーザ情報の管理

My webMethods Server では、ユーザとグループの管理のための複数のディレクトリサービスオプションが利用できます。

- My webMethods システムディレクトリ。これは、内部 My webMethods Server ユーザディレクトリであり、My webMethods Server のすべてのインストールで、デフォルトで使用できます。このディレクトリサーバ内の情報には、My webMethods ユーザインタフェースおよび共通ディレクトリサービス (CDS) API を使用してアクセスできます。読み取りアクセスと書き込みアクセスの両方が可能です。
- LDAP (Lightweight Directory Access Protocol)。My webMethods では、1 つ以上の外部 LDAP ユーザディレクトリを定義できます。サポートされるディレクトリサーバ製品のリストについては、『*System Requirements for Software AG Products*』を参照してください。このディレクトリサーバ内の情報には、My webMethods ユーザインタフェースおよび CDS API を使用してアクセスできます。読み取りアクセスのみが可能です。
- データベース。My webMethods Server では、データベースディレクトリに対してユーザを認証することもできます。データベースディレクトリは、RDBMS テーブルとこれらのテーブルにアクセスするための SQL 設定のセットです。カスタム認証モジュールを実装して、データベースディレクトリに対する認証を拡張できます。このディレクトリサーバ内の情報には、My webMethods ユーザインタフェースおよび CDS API を使用してアクセスできます。読み取りアクセスのみが可能です。

My webMethods Server とそこで実行されるアプリケーションおよびサービスでは、これらのディレクトリ内にあるユーザ情報にアクセスできます。My webMethods Server にアクセスしてこのデータを使用する外部アプリケーションおよびサービスを設定できます。

ディレクトリサービスでユーザおよびグループを使用する以外に、役割情報にアクセスして保持できます。役割情報は My webMethods Server データベース内に個別に保持されます。

たとえば、次の作業が可能です。

- 他のスイートアプリケーション (webMethods Integration Server など) を設定して、Integration Server 固有のユーザディレクトリではなく、上記のユーザディレクトリオプションからユーザを認証します。
- ビジネスプロセスでプロセスステップを設定し、Java サービスを呼び出して、ディレクトリサービスからユーザ属性を取得するか My webMethods Server データベースから役割のメンバーシップを取得し、そのデータをプロセスパイプラインに渡します。
- Java サービスを設定して、プログラムでユーザを役割に割り当てます。

CDS API では、以下がサポートされています。

- ユーザ、グループおよび役割の検索と検出。
- 大規模ディレクトリの LDAP 検索制御のサポート。
- システムディレクトリでのユーザおよびグループの作成と更新。その他の外部ディレクトリはすべて読み取り専用です。

- システムディレクトリからのユーザおよびグループの削除。
- 役割の My webMethods Server での作成、更新および削除。
- LDAP およびデータベースディレクトリからのカスタム属性の読み取り。
- ユーザ、グループおよび役割のカスタムプロファイル属性 (つまり、外部ディレクトリによって管理されない属性) の読み取りと書き込み。

共通ディレクトリサービス API の詳細については、『*webMethods CAF and My webMethods Server Java API Reference*』の com.webmethods.sc.directory パッケージと com.webmethods.sc.mws パッケージを参照してください。

共通ディレクトリサービス API について

同じ My webMethods Server データベースおよび Universal Messaging サーバに接続されている場合、CDS は My webMethods Server クラスタインスタンスのように動作し、クラスタ全体をわたるすべての分散キャッシュに参加します。システムディレクトリユーザが更新されると、この変更はすべての CDS インスタンスによって数分で検出されます。

ディレクトリサービスは、My webMethods Server で、SysAdmin または Administrator として My webMethods にログインして定義および管理します。ディレクトリサービスの使用の詳細については、114 ページの「ディレクトリサービスの管理」および116 ページの「外部ディレクトリサービスの管理」を参照してください。

CDS API を使用する以外に、My webMethods インタフェースを使用して、すべてのディレクトリ管理アクティビティを実行できます。

- ユーザ、グループおよび役割管理。詳細については、154 ページの「ユーザとグループの管理について」を参照してください。
- 認証管理。詳細については、257 ページの「セキュリティの管理」を参照してください。
- 属性プロバイダの実装。詳細については、247 ページの「属性プロバイダ」を参照してください。

前提条件

CDS API を使用する前に、以下のアクションを実行する必要があります。

- webMethods Database Component Configurator を使用して、My webMethods Server のデータベーステーブルを作成します。通常、これはインストールの直後に実行します。
- My webMethods Server インスタンスを作成および初期化します。
- My webMethods Server データベーススキーマへの JDBC 接続によって CDS を初期化します。詳細については、378 ページの「共通ディレクトリサービスの初期化」を参照してください。
- Universal Messaging サーバを、My webMethods の [クラスタ管理] ページを使って JMS プロバイダとして設定します。これは、My webMethods Server をインストールした後、ある

いはMy webMethods Server コマンドラインユーティリティまたは Software AG Command Central を使用して新しいインスタンスを作成したときに設定します。

クラスパスの考慮事項

スタンドアロンアプリケーションで CDS を使用するには、正しい JDBC 接続情報を指定するとともに、CDS を実行している外部アプリケーションのクラスパスにすべての CDS .jar ファイルが存在する必要があります。これらの .jar ファイルを使用できるようにするには、*Software AG_directory* /common/lib および /common/lib/ext のすべての .jar ファイルを含めてください (My webMethods Server および Integration Server の標準インストールを想定しています)。

共通ディレクトリサービスの初期化

CDS API に My webMethods Server 内から (たとえば CAF アプリケーションから) または Integration Server からアクセスする場合、CDS は既に初期化されており、追加のアクションは必要ありません。CDS API を My webMethods Server にアクセスする外部アプリケーションまたはサービスから使用する場合は、CDS API を外部アプリケーションまたはサービスから明示的に初期化する必要があります。

共通ディレクトリサービスは、com.webmethods.sc.mws.MWSLibrary.init() 静的メソッドを呼び出すことによって初期化します。入力パラメータは、Java システムプロパティとして予期されており、My webMethods Server データベーススキーマへの JDBC 接続 URL を指定する必要があります。My webMethods Server へのリモートコールバックを使用する代わりに、CDS API はこのスキーマに接続して、すべての設定およびプリンシパル情報を読み取ります。

次に、MWSLibrary クラスを使用して CDS を初期化する方法を示すサンプルコードを示します。

```
System.setProperty(MWSLibrary.SYSTEM_PROP_DB_DRIVER,
"com.wm.dd.jdbc.sqlserver.SQLServerDriver"); // JDBC Driver Class
```

```
System.setProperty(MWSLibrary.SYSTEM_PROP_DB_URL,
"jdbc:wm:sqlserver://localhost:1433;DatabaseName=webm82_dev");
    // JDBC Connection URL
```

```
System.setProperty(MWSLibrary.SYSTEM_PROP_DB_USER, "webm82_dev");
    // DB username
```

```
System.setProperty(MWSLibrary.SYSTEM_PROP_DB_PASSWORD, "password");
    // DB
password
```

```
MWSLibrary.init();
```

CDS のバージョンの相互運用性について

My webMethods Server は、ホストしている Integration Server インスタンスが新しいリリースのものである場合に、新しいバージョンの共通ディレクトリサービスコンポーネントと通信して、ユーザ情報を交換することができます。たとえば、バージョン 10.3 の My webMethods Server インスタンスは、バージョン 10.5 の Integration Server インストールから CDS コンポーネントと通信できます。バージョン互換モードは、デフォルトでは無効です。My webMethods

Server と CDS のバージョンの相互運用性を有効にする方法の詳細については、379 ページの「My webMethods Server でのバージョンの相互運用性の有効化」を参照してください。

My webMethods Server でのバージョンの相互運用性の有効化

1. `getconfig` コマンドを使用して、My webMethods Server の `cluster.xml` ファイルを取り出します。

2. テキストエディタで `cluster.xml` ファイルを開き、次のように、`<Cluster>` 要素の `useInteropMode` 属性を `true` に設定します。

```
<Cluster clusterId="ID" frontEndSecurePort="-1"
frontEndUrl="http://localhost:8585" useInteropMode="true">
```

3. ファイルを保存して `putconfig` コマンドを実行し、データベースに保存します。

4. `cluster.xml` ファイルのローカルコピーをノードから削除します。

   `cluster.xml` ファイルの使用の詳細については、228 ページの「cluster.xml ファイルの使用」を参照してください。

5. 『*webMethods Integration Server Administrator's Guide*』の説明に従って、Integration Server でセントラルユーザ管理を設定します。

CDS コード例

すべての役割のリスト

```
IDirectorySession session =
DirectorySystemFactory.getDirectorySystem().createSession();
```

```
List roles = session.listRoles();
```

```
for (IDirectoryRole role: roles) {
```

```
    String roleID = role.getID();
```

```
    String roleName = role.getName();
```

```
    String roleDN = role.getDN();
```

```
}
```

名前によるユーザのルックアップとすべての属性のフェッチ

```
IDirectorySession session =
DirectorySystemFactory.getDirectorySystem().createSession();
```

```
IDirectoryUser user = (IDirectoryUser) session.lookupPrincipalByName
```

```
   ("user1", IDirectoryPrincipal.TYPE_USER);
```

```
Map attributes = user.getAllAttributes();
```

ユーザの認証

```
IDirectorySession session =
DirectorySystemFactory.getDirectorySystem().createSession();
```

```
IDirectoryUser user = session.authenticateUser("username", "password");
```

静的役割の作成とメンバーとしてのユーザの追加

```
IDirectorySession session =
DirectorySystemFactory.getDirectorySystem().createSession();
```

```
IDirectoryUser user = (IDirectoryUser) session.lookupPrincipalByName
    ("user1", IDirectoryPrincipal.TYPE_USER);
```

```
IDirectoryRole role = session.createRole(IDirectoryRole.STATIC_ROLE_TYPE,
    "roleName", Collections.EMPTY_MAP);
```

```
session.addPrincipalToRole(user.getID(), role.getID());
```

21 My webMethods Server からの Mobile 通知の送信

■ モバイルデバイスへのプッシュ通知の送信 382

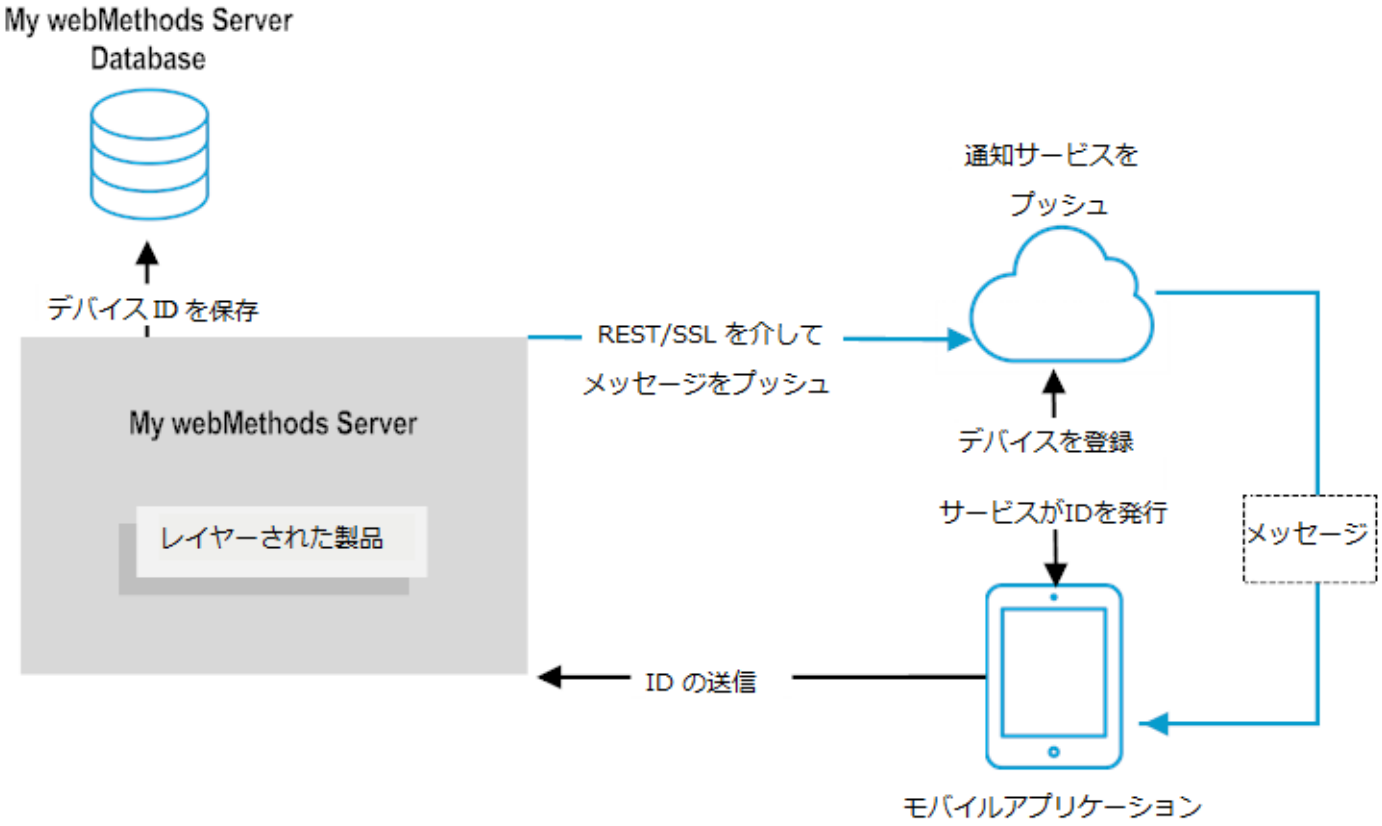
■ My webMethods Server でのプッシュ通知の設定 383

モバイルデバイスへのプッシュ通知の送信

モバイルデバイス上のユーザーにイベントのプッシュ通知を送信するように My webMethods Server を設定できます。

My webMethods Server は、クライアントアプリケーションにメッセージを送信するために、モバイル の OS に応じて異なるプッシュ通知サービスプロバイダを使用します。Android アプリケーションのデフォルト通知プロバイダは Google Cloud Messaging で、Apple のデフォルト通知プロバイダは iStore です。My webMethods Server からプッシュ通知メッセージを送信する前に、モバイルアプリケーションをそれぞれのサービスプロバイダに登録する必要があります。

モバイルアプリケーションを実行しているデバイスがそれぞれのプッシュ通知プロバイダに登録されると、プロバイダはデバイスの登録 ID を発行します。デバイスは、登録 ID をサブスクライブするイベントの詳細と共に My webMethods Server に送信します。My webMethods Server は、デバイス ID とイベントサブスクリプションの詳細をサーバデータベースに格納します。関連するイベントが発生した場合は、My webMethods Server はデバイス登録 ID を使用して、通知サービスプロバイダにメッセージをプッシュします。プッシュ通知サービスプロバイダは、登録されたデバイスにメッセージを送信します。モバイルアプリケーションユーザが複数のデバイスから接続する場合は、My webMethods Server は登録されている各デバイスに通知を送信します。



My webMethods Server は REST を使用して Google Cloud Messaging に通知をプッシュし、SSL を使用して iStore に通知をプッシュします。

My webMethods Server は、モバイルデバイス上の Task Engine ユーザに、Mobile Business Console イベントのプッシュ通知を送信するためのデフォルト実装を提供します。この実装では、ユーザまたはユーザが属する役割やグループが割り当てられるか、タスクが委任されるたびに、My webMethods Server は通知を送信します。カスタムモバイルアプリケーションのプッシュ通知を送信するには、次の手順で行う必要があります。

- 通知を送信するイベントタイプのカスタムイベントハンドラを実装します。詳細については、*webMethods CAF and OpenUI Development Help*を参照してください。
- My webMethods Server の `publishNotification` API を介して通知をプッシュするカスタムサービスを作成します。詳細については、*webMethods CAF and My webMethods Server Java API Reference*を参照してください。
- プッシュ通知サービスにアプリケーションを登録します。プッシュ通知サービスプロバイダに登録する方法の詳細については、プロバイダのマニュアルを参照してください。
- 383 ページの「My webMethods Server でのプッシュ通知の設定」で説明されているように、My webMethods Server でモバイルプッシュ通知オプションを設定します。

My webMethods Server でのプッシュ通知の設定

My webMethods Server でのプッシュ通知は、モバイルアプリケーションを各オペレーティングシステムのプッシュ通知サービスに登録した後で設定します。

1. sysadmin として、[**フォルダ**] > [管理用フォルダ] > [Administration Dashboard] > [設定] > [モバイルプッシュ通知管理] に移動します。
2. [**新規アプリケーションの追加**] をクリックします。
3. [**モバイルアプリケーション設定**] ページで、モバイルアプリケーションの名前とアプリケーションの詳細を指定します。
   - Android アプリケーションの場合

次の表に、Android アプリケーションに必要なプロパティを示します。

フィールド	説明
[エンドポイント URL]	Google Cloud Messaging に登録した送信 API のエンドポイント。
[OAuth]	Google Cloud Messaging に登録されているアプリケーションの OAuth 認証トークン。

   - iOS アプリケーションの場合

次の表に、iOS アプリケーションに必要なプロパティを示します。

フィールド	説明
[ホスト名]	iStore を使用して登録した Apple プッシュ通知サービスの URL。
[ポート番号]	iStore プッシュ通知サービスとの SSL 通信に使用するポート番号。デフォルトのポート番号は 2195 です。
[認証のパス]	iStore に登録した SSL 認証の場所への完全パス。
[パスワード]	iStore への接続に使用するパスワード。

4. [**サブミット**] をクリックします。

IV サーバページの開発

22 My webMethods Server でのページの管理 387
23 My webMethods Server でのワークスペースの管理 419
24 スキンのカスタマイズ 441
25 My webMethods Server でのシェルの使用 469

22 My webMethods Server でのページの管理

- ページの開発について 388
- カスタムフォルダおよびカスタムページについて 388
- My webMethods ナビゲーションのカスタマイズについて 407
- My webMethods のルックアンドフィールのカスタマイズについて 412
- My webMethods への簡単なフロントエンドページの作成 416
- シングルサインオン用のリンクの作成 417

ページの開発について

My webMethods Server では簡単にカスタムページを作成して、My webMethods Server でカスタムページを Web ページとして表示できます。詳細については、388 ページの「カスタムフォルダおよびカスタムページについて」を参照してください。

また、My webMethods Server には、以下の表に示すように、My webMethods ユーザインタフェースをカスタマイズできる柔軟なアーキテクチャが備えられています。

アクティビティ	参照先
以下の操作によって My webMethods ナビゲーションパネルを更新する ■ ナビゲーションパネルに選択肢を追加する ■ ナビゲーションパネルから選択肢を削除する ■ ナビゲーションパネルの標準的なタブおよびセクションを非表示にする ■ ナビゲーションパネルを完全に置換する	407 ページの「My webMethods ナビゲーションのカスタマイズについて」
以下の操作によって My webMethods のルックアンドフィールを更新する ■ ロゴイメージを置換する ■ 色体系を変更する ■ カスタムルックアンドフィールを適用する	412 ページの「My webMethods のルックアンドフィールのカスタマイズについて」
My webMethods アプリケーションページへの代替アクセス方法を作成する、つまりユーザが必要な少数の My webMethods ページのみへのリンクを提供する簡単なフロントエンドページを作成します。	416 ページの「My webMethods への簡単なフロントエンドページの作成」

さらに、上記の要領で、My webMethods Server で実行する完全なカスタムアプリケーションを作成することもできます。

カスタムフォルダおよびカスタムページについて

My webMethods Server によって Web ページとして表示されるカスタムフォルダおよびカスタムページを作成できます。My webMethods Server においては、これらは両方とも機能的には同等です。つまり、My webMethods Server では、どちらの内容も Web ページ上のポートレットとして表示されます。使用方法に基づいて、フォルダを使用するか、またはページを使用するかを決定します。他のフォルダおよびページを保持するコンテナが必要な場合は、フォルダを使用します。情報を表示する必要がある場合は、ページを使用します。

長期間使用する、永続的な内容のページが必要な場合は、カスタムページを作成します。永続的な内容ではなく、作業領域として使用できるページが必要な場合は、ワークスペースを使用することをお勧めします。詳細については、419 ページの「My webMethods Server でのワークスペースの管理」を参照してください。

ページにコンテンツを追加するには、ページにポートレットをドラッグアンドドロップします。ポートレットは、任意の場所への配置、サイズの変更、プロパティの設定が可能です。ワイヤリングを使用して、ページを動的にすることができます。

- あるポートレットのプロパティを別のポートレットのプロパティにワイヤリングできます。ワイヤリングにより、ソースポートレットのプロパティが設定されると、その変更内容が自動的に宛先ポートレットに反映されます。
- カスタムページにアクセスするユーザについての情報をポートレットプロパティにワイヤリングできます。たとえば、天気を表示するポートレットでは、ユーザプロファイルの郵便番号属性を天気のポートレットの郵便番号プロパティにワイヤリングできます。このようにワイヤリングすると、ユーザがページを表示したときに、My webMethods Server では、ユーザプロファイルの郵便番号を使用して自動的に天気ポートレットの郵便番号プロパティが設定されて、ユーザの現地の天気が表示されます。

カスタムページの作成

SysAdmin ユーザアカウントまたは Admin Role のメンバーである別のユーザアカウントを使用して My webMethods にログインし、ポータルページを作成します。ページを作成するには、ページ編集モードに切り替える必要があります。ページ編集モードでは、システム管理者ユーザインタフェースが変更されて、左側に [**ツール**] タブが、右側にカスタムページが表示されます。[**ツール**] タブには、カスタムページに追加できるポートレットが表示されます。

カスタムページを作成するには

1. システム管理者として、新規ページの配置先のフォルダに移動します。使用するフォルダが存在しない場合は、フォルダを作成します。
   - My webMethods アプリケーションのナビゲーションにページを追加する場合は、以下の階層内の場所に移動します。

     [フォルダ] > [My webMethods アプリケーション] > [ファブリックタスク]
   - 他の分類にページを追加する場合は、以下のいずれかの階層内の場所を使用することをお勧めします。
     - [フォルダ] (ルートフォルダ)
     - **[フォルダ] > [パブリックフォルダ]**
2. フォルダのタイトルバーで、 をクリックして [新規] > [ページ] を選択します。
3. [新規ページ] ウィンドウで、以下の手順に従います。
   a. [**名前**] フィールドに、新規ページの名前を入力します。

b. [**説明**] フィールドに、新規ページの説明を入力します (オプション)。

c. [**作成**] をクリックします。

4. ページのリンクをクリックして、作成した新規ページを開きます。

5. ページのタイトルバーで、をクリックして [ページを編集] を選択し、ページ編集モードに切り替えます。

6. ページのプロパティを設定します。詳細については、392 ページの「ページのプロパティ」および 392 ページの「ページのプロパティの設定」を参照してください。

7. ページレイアウトを定義します。手順については、『393 ページの「ページレイアウトの制御」』を参照してください。

8. 以下の表に示すタスクを実行して、ページを作成します。

タスク	参照先
[**ツール**] タブからページキャンバスにポートレットをドラッグして、ポートレットを追加する	397 ページの「ページへのポートレットの追加」
ページ上でポートレットを配置する	398 ページの「ページ上でのポートレットの配置」
ポートレットのプロパティを設定する	402 ページの「ポートレットプロパティの変更」
オプションで、ポートレットのエイリアスを設定する	403 ページの「ポートレットエイリアスの管理」
オプションで、ポートレットをワイヤリングする	404 ページの「ポートレット間でのプロパティのワイヤリング」、405 ページの「プリンシパル属性からポートレットプロパティへのワイヤリング」

9. ページのタイトルバーで、[**保存**] をクリックします。

新規ページを以下のように使用可能にできます。

- ユーザがページに直接アクセスできるように、ページの URL を提供します。URL は、ページにエイリアスを割り当てることによって単純なものにします。たとえば、ページに「MyCustomPage」というエイリアスを割り当てると、ユーザは以下の URL を入力できます。*host_name* は My webMethods Server のホスト名であり、*port_number* はポート番号です。

  http://www.*host_name:port_number*/MyCustomPage

エイリアスの割り当て方法の詳細については、392 ページの「ページのプロパティの設定」を参照してください。

- ページを My webMethods のナビゲーションに追加して、ユーザがナビゲーションからページを選択できるようにします。詳細については、407 ページの「My webMethods のナビゲーションへの選択肢の追加」を参照してください。

既存のページの編集

最初にページを作成した後、いつでもページをページ編集モードで開いて、追加の変更を加えることができます。

既存のページを編集するには

1. システム管理者として、編集するページに移動して、そのページを開きます。
2. ページのタイトルバーで、をクリックして [ページを編集] を選択し、ページ編集モードに切り替えます。
3. ページ編集モードで、以下のいずれかを実行します。

次の表は、既存のページの編集オプションと、各オプションに関する詳細情報の参照先を示しています。

タスク	参照先
ページのプロパティを更新する (ページの名前を変更するなど)	392 ページの「ページのプロパティ」、392 ページの「ページのプロパティの設定」
ページレイアウトを変更する	393 ページの「ページレイアウトの制御」
ページにポートレットを追加する	397 ページの「ページへのポートレットの追加」
ページからポートレットを削除する	398 ページの「ページからのポートレットの削除」
ページ上でポートレットを再配置する	398 ページの「ページ上でのポートレットの配置」
ポートレットのプロパティを更新する	402 ページの「ポートレットプロパティの変更」
ポートレットのエイリアスを設定する	403 ページの「ポートレットエイリアスの管理」
ポートレットをワイヤリングする	404 ページの「ポートレット間でのプロパティのワイヤリング」、405 ページの「プリンシパル属性からポートレットプロパティへのワイヤリング」

4. ページのタイトルバーで、[**保存**] をクリックします。

ページのプロパティ

ページの [全般] タブには、次の表に示すようなプロパティが表示されます。

プロパティ	説明
[名前]	ページの名前。
[説明]	ページの説明 (オプション)。
[キーワード]	独自の用途に使用するためにページに割り当てるキーワード (オプション)。デフォルトでは、My webMethods Server はキーワードを使用する機能を備えていません。ただし、キーワードを利用するカスタムコードを記述できます。たとえば、キーワードに基づいて検索を行うカスタム検索コードを作成できます。
[所有者]	ページの所有者。所有者のプロファイルを表示するには、リンクをクリックします。
[作成日時]	ページが作成された日時。
[更新日時]	ページが最後に更新された日時。
[エイリアス]	ページに割り当てられたエイリアス。

プロパティの設定方法の手順については、392 ページの「ページのプロパティの設定」を参照してください。[レイアウト] タブの詳細については、393 ページの「ページレイアウトの制御」を参照してください。

ページのプロパティの設定

ページのプロパティを設定するには、以下の手順に従います。プロパティの詳細については、392 ページの「ページのプロパティ」を参照してください。

ページのプロパティを設定するには

1. システム管理者として、プロパティを設定するページに移動して、そのページを開きます。
2. ページのタイトルバーで、をクリックして [ページを編集] を選択し、ページ編集モードに切り替えます。
3. [プロパティ] をクリックします。
4. [**全般**] タブで次を指定します。

以下の表は、[全般] タブで使用可能なページプロパティと、その機能を示しています。

タスク	プロパティ	アクション
ページの名前を変更する	**[名前]**	[**名前**] フィールドに新しい名前を入力します。
ページの説明を変更または追加する	**[説明]**	[**説明**] に説明を入力します。
ページにキーワードを割り当てる	**[キーワード]**	[**キーワード**] フィールドに、1 つ以上のキーワードをカンマで区切って入力します。
ページにエイリアスを割り当てる	**[エイリアス]**	**エイリアスを追加するには、以下の手順に従います。** a. [**追加**] をクリックします。 b. テキストボックスに、追加するエイリアス名を入力します。 c. [**OK**] をクリックします。 **エイリアスを更新するには、以下の手順に従います。** a. 変更するエイリアスを選択します。 b. [**編集**] をクリックします。 c. テキストボックスに、更新後のエイリアス名を入力します。 d. [**OK**] をクリックします。 **エイリアスを削除するには、以下の手順に従います。** a. 削除するエイリアスを選択します。 b. [**削除**] をクリックします。

5. [**OK**] をクリックします。

6. ページのタイトルバーで、[**保存**] をクリックします。

ページレイアウトの制御

ページ内でのポートレットの配置方法を定義するには、ページレイアウトを定義します。レイアウトには、以下の 2 つのタイプがあります。

- **列レイアウト**。このレイアウトでは、ページ内のすべてのポートレットが列内に整列されます。列内でポートレットを重ね合わせることはできません。ページに定義できる列は 1～4 です。新規ページのデフォルトは 2 列のレイアウトです。

  各列には単一の行があります。[**行**] ツールを列にドラッグすることで、列に行を追加できます。行の使用方法の詳細については、396 ページの「列レイアウト使用時の行の追加」および 397 ページの「列レイアウト使用時の行の削除」を参照してください。

- **フリーフォームレイアウト**。このレイアウトでは、ページ内の任意の場所にポートレットを配置できます。ポートレットを部分的に重ね合わせたり、互いに完全に重ねたりすることもできます。

ページへのポートレットの追加の詳細については、397 ページの「ページへのポートレットの追加」を参照してください。ページ内のポートレットの配置方法の詳細については、398 ページの「ページ上でのポートレットの配置」を参照してください。

ページのレイアウトを定義するには

1. システム管理者として、レイアウトを定義するページに移動して、そのページを開きます。
2. ページのタイトルバーで、 をクリックして [ページを編集] を選択し、ページ編集モードに切り替えます。
3. [プロパティ] をクリックします。
4. [**レイアウト**] タブを選択します。
5. [**表示形式**] リストを使用して、列、行、ポートレットの枠、および非表示のポートレットタイトルバーを表示するかどうかを指定します。
   - 枠または非表示のタイトルバーを表示しない場合は、[**エンドユーザ**] を選択します。
   - 枠および非表示のタイトルバーを表示する場合は、[**エキスパートユーザ**] を選択します。

     枠があると、ポートレットを配置するときに役立ちます。また、ポートレットのプロパティを設定してタイトルバーを非表示にする場合は、エキスパートユーザとしてページを表示し、タイトルバーを再表示して、ポートレットを移動したり、プロパティにアクセスしたりするなど、ポートレットに対してアクションを実行します。

     注:
     エキスパートとしての表示設定は、一時的です。編集対象ページを離れてからそのページに戻ると、再度エンドユーザとして表示されます。

注:
ページ上部の [**エキスパートとして表示**] チェックボックスは、[**表示形式**] プロパティと同じ目的で使用できます。[**表示形式**] プロパティで [**エキスパートユーザ**] を選択した場合、プロパティを保存すると、My webMethods Server によって [**エキスパートとして表示**] チェックボックスが自動的にオンになります。同様に、[**表示形式**] プロパティで [**エンドユーザ**] を選択した場合、プロパティを保存すると、My webMethods Server によって [**エキスパートとして表示**] チェックボックスが自動的にオフになります。

6. ページ上のポートレットのユーザによる再配置およびサイズ変更を許可するかどうかに基づいて、[**編集可能なキャンバス**] プロパティを設定します。

   - ページ上のポートレットのユーザによる再配置およびサイズ変更を許可する場合は、[**編集可能なキャンバス**] チェックボックスをオンにします。
   - ページ上のポートレットのユーザによる再配置およびサイズ変更を禁止する場合は、[**編集可能なキャンバス**] チェックボックスをオフにします。

     [**編集可能なキャンバス**] チェックボックスをオフにすると、エンドユーザがページの使用中に誤ってレイアウトを変更することを防止できます。ページに対してフリーフォームレイアウトを選択すると、ユーザは、ポートレットをクリックするだけで誤ってページを変更する危険性があります。

   注:
   [**編集可能なキャンバス**] チェックボックスをオンにすると、ユーザがレイアウトを変更できます。ユーザに編集権限が与えられていない場合、ユーザはレイアウトを変更できますが、編集内容の保存はできません。ただし、[**編集可能なキャンバス**] チェックボックスをクリアすると、すべてのユーザがレイアウトを変更できなくなります。

7. [**列**] リストから、ページで使用するレイアウトを選択します。

8. [**1 列**]、[**2 列**]、[**3 列**]、または [**4 列**] を選択した場合、レイアウトの各列に対して追加のプロパティを設定できます。

   a. オプションで、[**属性**] フィールドに、独自の用途に使用するための列の属性を指定します。標準の My webMethods Server は、属性を使用する機能を提供しません。ただし、属性を利用するカスタムコードは記述できます。

   b. [**幅**] フィールドに、列で使用する幅をページに対するパーセンテージで入力します。パーセンテージは、デフォルトで、各列の幅が均等なサイズになるように設定されます。

   c. サーバにおいて、ポートレット内の長いテキスト行を列サイズに収まるように折り返す必要がある場合は、[**行の折り返し**] チェックボックスをオンにします。長い行を折り返さない場合は、チェックボックスをオフにします。

      長い行を折り返すと、ポートレットを列にうまく収めることができます。

   d. [**水平配置**] リストから、ポートレットの列内での水平方向の配置方法を選択します。デフォルトで、ポートレットは、各列内で左揃えで配置されます。

   e. [**垂直配置**] リストから、ポートレットの列内での垂直方向の配置方法を選択します。デフォルトで、ポートレットは、各列内で上揃えで配置されます。

   f. 列に CSS クラスを適用するには、[**CSS クラス**] フィールドに、先頭のピリオドを省略してクラスの名前を入力します。たとえば、「.nav」クラスの場合は「nav」と入力します。

ページに含まれる、以下のいずれかの CSS スタイルシートによって定義される CSS クラスを指定します。

- 現在のスキンで使用されている
- ページコンテンツまたは現在のシェル内のカスタムポートレットに含まれている

g. 列にスタイルを適用するには、[**CSS スタイル**] フィールドに、CSS ファイル内で使用可能な任意のスタイルを入力します。たとえば、「`border: 1pt dashed red`」と入力すると、赤の破線が列の枠として表示されます。

h. 列に背景画像を適用するには、[**スキン背景画像**] フィールドに、現在のスキンのイメージファイルの名前を入力します。

スキンプロパティの名前を指定します。たとえば、スキンのメインロゴイメージを使用するには、「`images/logo.gif`」と入力します。([**Administration Dashboard**] > [ユーザインタフェース] > [スキン管理] ポートレット経由で) スキンエディタにアクセスし、[画像] ページでイメージを検索して、イメージのスキンプロパティ名を指定できます。スキンプロパティの「images/」の後には、プロパティ名が続きます (「images/logo.gif」など)。

注:
現在のスキンに含まれていないイメージを適用するには、[**CSS スタイル**] フィールドで標準の CSS 背景画像プロパティを指定します。[**CSS スタイル**] フィールドでプロパティを指定して、背景を繰り返すか、中央に配置するか、ページと共にスクロールするかどうかなど、背景の表示方法を制御することもできます。

9. [**OK**] をクリックします。

10. ページのタイトルバーで、[**保存**] をクリックします。

列レイアウト使用時の行の追加

ページのレイアウトに列レイアウトを使用するように設定すると、デフォルトで、ページの各列には 1 つの行が設定されます。列に追加するポートレットは、列に指定した垂直配置方法に基づいて、垂直方向に配置されます。ただし、1 つ以上の列に対して行を追加することができます。

ページの列に行を追加するには

1. システム管理者として、行を追加するページに移動して、そのページを開きます。

2. ページのタイトルバーで、 をクリックして [ページを編集] を選択し、ページ編集モードに切り替えます。

3. [**エキスパートとして表示**] チェックボックスをオンにして、ページに追加する行の枠およびタイトルバーが表示されるようにします。

4. [**ツール**] タブで、[**レイアウト**] アイテムを展開して、[**行**] ツールを表示します。

5. 行を追加する列に [**行**] ツールをドラッグします。

   マウスボタンを離した場合に行が配置される場所を示す赤いボックスが、カーソル位置の下に表示されます。

6. [**保存**] をクリックします。

列レイアウト使用時の行の削除

レイアウト内の列に行を追加した後、その行が不要になった場合は、行を削除できます。

列から行を削除するには

1. システム管理者として、行を削除するページに移動して、そのページを開きます。

2. ページのタイトルバーで、 をクリックして [ページを編集] を選択し、ページ編集モードに切り替えます。

3. [**エキスパートとして表示**] チェックボックスをオンにして、ページ上の行の枠およびタイトルバーが表示されるようにします。

4. 削除する行内に、保持するポートレットが含まれている場合は、そのポートレットを行からページ上の他の場所にドラッグします。

5. 行のタイトルバーで、 をクリックします。

6. [**保存**] をクリックします。

ページへのポートレットの追加

ページには、任意の数のポートレットを追加できます。同じポートレットを複数回追加することもできます。ページ上でポートレットを配置できる場所は、ページレイアウトに応じて異なります。詳細については、393 ページの「ページレイアウトの制御」と398 ページの「ページ上でのポートレットの配置」を参照してください。

ページにポートレットを追加するには

1. システム管理者として、ポートレットを追加するページに移動して、そのページを開きます。

2. ページのタイトルバーで、 をクリックして [ページを編集] を選択し、ページ編集モードに切り替えます。

ページ編集モードでは、左側の [**ツール**] タブに、ページに追加できるポートレットが表示されます。[**ツール**] タブに表示されるポートレットの多くは、『*My webMethods Server Portlet Reference*』で説明されています。

3. ポートレットを配置しやすいように列、行およびポートレットの枠を表示する場合は、[**エキスパートとして表示**] チェックボックスをオンにします。枠を表示する必要がなくなったら、チェックボックスをオフにします。

4. [**ツール**] タブで、追加するポートレットを選択して、ページにドラッグします。

   マウスボタンを離した場合にポートレットが配置される場所を示す赤いボックスが、カーソル位置の下に表示されます。

5. [**保存**] をクリックします。

ページからのポートレットの削除

ページからポートレットを削除するには、以下の手順に従います。

ページからポートレットを削除するには

1. システム管理者として、ポートレットを削除するページに移動して、そのページを開きます。

2. ページのタイトルバーで、 をクリックして [ページを編集] を選択し、ページ編集モードに切り替えます。

3. 削除するポートレットのタイトルバーが非表示になっている場合は、[**エキスパートとして表示**] チェックボックスをオンにして、タイトルバーを表示します。

4. 削除するポートレットのタイトルバーの をクリックし、[削除] を選択します。

5. [保存] をクリックします。

ページ上でのポートレットの配置

ページ上でポートレットを配置できる場所は、ページレイアウトに応じて異なります。

- 列レイアウト ([3 列] など) を使用すると、レイアウトによって、ポートレットが列内で強制的に整列されます。列内でのポートレットの上下の移動、および列間でのポートレットの移動は可能です。
- フリーフォームレイアウトを使用すると、ページ内の任意の場所にポートレットを移動できます。ポートレットは整列する必要がなく、相互に重なり合うことができます。

ページレイアウトの定義方法の詳細については、393 ページの「ページレイアウトの制御」を参照してください。

ページ上でポートレットを配置するには

1. システム管理者として、更新するページに移動して、そのページを開きます。
2. ページのタイトルバーで、をクリックして [ページを編集] を選択し、ページ編集モードに切り替えます。
3. 再配置するポートレットのタイトルバーが非表示になっている場合は、[**エキスパートとして表示**] チェックボックスをオンにして、タイトルバーを表示します。
4. 再配置するポートレットのタイトルバーの上にカーソルを移動して、カーソルを移動カーソルに変化させます。
5. ポートレットをクリックして、新しい場所にドラッグします。

   マウスボタンを離した場合にポートレットが配置される場所を示す赤いボックスが、カーソル位置の下に表示されます。
6. [**保存**] をクリックします。

ポートレットプロパティ

[全般] タブ

My webMethods Server は、通常表示モードおよびページ編集モードでポートレットプロパティを表示します。以下の表は、[全般] タブで使用可能なポートレットプロパティを示しています。

セクション	プロパティ	説明
[全般]	**[名前]**	ポートレットタイトルバーに表示されるポートレットの名前。
	[説明]	ポートレットの説明 (オプション)。
	[キーワード]	独自の用途に使用するためにポートレットに割り当てるキーワード (オプション)。デフォルトでは、My webMethods Server はキーワードを使用する機能を備えていません。ただし、キーワードを利用するカスタムコードを記述できます。
[KPI インスタンスの詳細] ページ	**[フルページ表示]**	ポートレット自体に移動し、My webMethods Server によってそのポートレットのみを含むページが表示される場合にポートレットで使用する表示方法。
	[ポートレット表示]	ポートレットを含むページに移動した場合にポートレットで使用する表示方法。
[保守]	**[所有者]**	ポートレットをページに追加したユーザのユーザ名。
	[作成日時]	ポートレットがページに追加された日時。

セクション	プロパティ	説明
	[更新日時]	ポートレットが最後に更新された日時。
	[エイリアス]	ポートレットに割り当てられたエイリアス (オプション)。詳細については、403 ページの「ポートレットエイリアスの管理」を参照してください。

[設定] タブ

[**設定**] タブには、ポートレットに固有であり、通常 My webMethods Server がポートレット内に表示する情報を定義するプロパティが含まれています。たとえば、[**HTML テキスト**] ツールでは、[**設定**] タブで、ポートレットに表示するテキストを指定できます。

[**設定**] タブを表示するには、ページ編集モードからプロパティを表示する必要があります。[**設定**] タブを使用しないポートレットもあります。

[レイアウト] タブ

[**レイアウト**] タブには、My webMethods Server によるポートレットの表示を制御するプロパティが含まれています。[**レイアウト**] タブを表示するには、ページ編集モードからプロパティを表示する必要があります。

以下の表は、一般的に [**レイアウト**] タブに含まれているプロパティを示しています。

セクション名	プロパティ	説明
[サイズ調整および配置]	**[幅]**	ポートレットの幅の指定。ポートレットの幅に使用するピクセル数を指定します。
	[高さ]	ポートレットの高さの指定。ポートレットの高さに使用するピクセル数を指定します。
	[自動配置]	ページのレンダリング時における My webMethods Server によるポートレットの配置方法を指定します。[**自動配置**] は次のいずれかに切り替わります。 ■ **オン**: My webMethods Server はポートレットを自動的に配置します。ポートレットと共に保存されている配置情報はすべて無視されます。 ■ **オフ**: My webMethods Server は、ポートレットと共に保存されている配置情報を使用して、ページでポートレットを配置する場所を決定します。このため、My webMethods Server は、ページが最後に保存されたときの場所を使用します。
[KPI インスタンスの詳細] ページ	**[タイトルバー]**	My webMethods Server がポートレットのタイトルバーを表示するかどうか。タイトルバーを表示するには、

セクション名	プロパティ	説明
		チェックボックスをオンにします。タイトルバーを非表示にするには、チェックボックスをオフにします。 **注:** タイトルバーを非表示にした場合は、ページを編集するときに [**エキスパートとして表示**] を選択して一時的にタイトルバーを表示すると、ページ上のポートレットを操作できます。
	[枠]	My webMethods Server がポートレットの枠を表示するかどうか。枠を表示するには、チェックボックスをオンにします。枠を非表示にするには、チェックボックスをオフにします。 **注:** 枠を非表示にした場合は、ページを編集するときに [**エキスパートとして表示**] を選択して一時的に枠を表示すると、ポートレットの配置およびサイズ変更に役立ちます。
	[最小化]	My webMethods Server がポートレットを含むページを表示するとき、初期状態でポートレットを最小化して表示するかどうか。ユーザは、ポートレットを復元して、そのコンテンツを表示できます。
	[CSS クラス]	ポートレットに適用する CSS クラス。先頭のピリオドを省略して、クラスの名前を入力します。たとえば、「.nav」クラスの場合は「`nav`」と入力します。 ページに含まれる、以下のいずれかの CSS スタイルシートによって定義される CSS クラスを指定します。 ■ 現在のスキンで使用されている ■ ページコンテンツまたは現在のシェル内のカスタムポートレットに含まれている
	[CSS スタイル]	ポートレットに適用するスタイル。CSS ファイルで使用可能な任意のスタイルを指定します。たとえば、「`border: 1pt dashed red`」と入力すると、赤の破線がポートレットの枠として表示されます。
	[スキン背景画像]	ポートレットで使用する背景画像。現在のスキンのイメージファイルの名前を入力します。 スキンプロパティの名前を指定します。たとえば、スキンのメインロゴイメージを使用するには、「`images/logo.gif`」と入力します。([**Administration Dashboard**] > [ユーザインタフェース] > [スキン管理]

セクション名	プロパティ	説明
		ポートレット経由で) スキンエディタにアクセスし、[画像] ページでイメージを検索して、イメージのスキンプロパティ名を指定できます。スキンプロパティの「images/」の後には、プロパティ名が続きます (「images/logo.gif」など)。 **注:** 現在のスキンに含まれていないイメージを適用するには、[**CSS スタイル**] フィールドで標準の CSS 背景画像プロパティを指定します。[**CSS スタイル**] フィールドでプロパティを指定して、背景を繰り返すか、中央に配置するか、ページと共にスクロールするかどうかなど、背景の表示方法を制御することもできます。

[メタデータ] タブ

[**メタデータ**] タブには、ポートレットに固有であり、通常 My webMethods Server がポートレット内に表示する情報を定義するプロパティが含まれています。

[**メタデータ**] タブを表示するには、ページ編集モードからプロパティを表示する必要があります。[**設定**] タブを使用しないポートレットもあります。

[ワイヤリング] タブ

[**ワイヤリング**] タブを使用して、プロパティの値をワイヤリングします。[**ワイヤリング**] タブを表示するには、ページ編集モードからプロパティを表示する必要があります。ワイヤリングプロパティの詳細については、404 ページの「ポートレット間でのプロパティのワイヤリング」および405 ページの「プリンシパル属性からポートレットプロパティへのワイヤリング」を参照してください。

ポートレットプロパティの変更

ポートレットのプロパティを設定して、ポートレットの表示設定を指定し、ポートレットの機能を定義します。ポートレットのプロパティは、ポートレットを含むページのプロパティとは独立して設定します。

ポートレットのプロパティを変更するには

1. システム管理者として、更新するポートレットを含むページに移動して、そのページを開きます。

2. ページのタイトルバーで、 をクリックして [ページを編集] を選択し、ページ編集モードに切り替えます。

3. 更新するポートレットのタイトルバーが非表示になっている場合は、[**エキスパートとして表示**] チェックボックスをオンにして、タイトルバーを表示します。

4. ポートレットのタイトルバーで、 をクリックして [プロパティ] を選択します。

5. 必要に応じてプロパティを変更します。プロパティの詳細については、399 ページの「ポートレットプロパティ」を参照してください。

6. ポートレットの下部で、[**適用**] をクリックします。

7. [**保存**] をクリックします。

ヒント:
更新するポートレットのタイトルバーが表示されている場合は、ページ編集モードに切り替えることなくポートレットのプロパティを更新できます。通常表示モードでポートレットのプロパティを表示するには、ポートレットのタイトルバーで をクリックし、[プロパティ] を選択します。

ポートレットエイリアスの管理

個々のポートレットにエイリアスを割り当てることができます。エイリアスは、ポートレットに付ける新しい名前またはより簡単な名前です。ポートレットにエイリアスを割り当てると、URL にエイリアスが指定された場合に My webMethods Server によってエイリアスが認識されて、ユーザが自動的にポートレットにリダイレクトされます。

エイリアスは、ポートレットに直接アクセスするために使用します。また、エイリアスは、ページ開発者が複数ポートレットから構成されるアプリケーションを作成する場合にも役立ちます。開発者は、エイリアスを使用して、ポートレット間の通信を実行できます。

ポートレットエイリアスを追加、編集または削除するには

1. システム管理者として、更新するポートレットを含むページに移動して、そのページを開きます。

2. ページのタイトルバーで、 をクリックして [ページを編集] を選択し、ページ編集モードに切り替えます。

3. 更新するポートレットのタイトルバーが非表示になっている場合は、[**エキスパートとして表示**] チェックボックスをオンにして、タイトルバーを表示します。

4. ポートレットのタイトルバーで、 をクリックして [プロパティ] を選択します。

5. エイリアスを追加するには、以下の手順に従います。

   a. [プロパティ] ページの [**エイリアス**] セクションで、[**追加**] をクリックします。

b. 追加するポートレットのエイリアスをテキストボックスに入力します。

c. [**OK**] をクリックします。

6. 既存のエイリアスを編集するには、以下の手順に従います。

a. [プロパティ] ページの [**エイリアス**] セクションで、更新するエイリアスを選択して、[**編集**] をクリックします。

b. 新しいポートレットのエイリアスをテキストボックスに入力します。

c. [**OK**] をクリックします。

7. エイリアスを削除するには、[プロパティ] ページの [**エイリアス**] セクションで、削除するエイリアスを選択して、[**削除**] をクリックします。

8. ページの下部で、[**適用**] をクリックします。

9. [**保存**] をクリックします。

ヒント:
更新するポートレットのタイトルバーが表示されている場合は、ページ編集モードに切り替えることなくポートレットのエイリアスを操作できます。通常表示モードでポートレットのプロパティを表示するには、ポートレットのタイトルバーで をクリックし、[プロパティ] を選択します。

ポートレット間でのプロパティのワイヤリング

ページ上の任意のポートレットの任意のプロパティを、別の任意のポートレットの任意のプロパティと接続 (ワイヤリング) できます。あるプロパティを別のプロパティとワイヤリングした場合、ページがレンダリングされると、宛先のプロパティの値がソースプロパティの値で自動的に設定されます。この機能を使用して、複数のポートレットから 1 つの複合アプリケーションを迅速に作成できます。

たとえば、1 つは検索フォーム、もう 1 つは検索結果の表示という 2 つのポートレットを含むページを作成する場合、検索フォームの値 (ソース値) を結果表示の入力値 (宛先値) とワイヤリングできます。ユーザが検索フォームになんらかの情報を入力してサブミットすると、サーバはこの検索の結果を使用して、結果表示を更新します。

ポートレット同士をワイヤリングするには

1. システム管理者として、ワイヤリングするポートレットを含むページに移動して、そのページを開きます。

2. ページのタイトルバーで、 をクリックして [ページを編集] を選択し、ページ編集モードに切り替えます。

3. ワイヤリングのソースとするポートレットとワイヤリングの宛先とするポートレットを決定します。

   宛先のポートレットプロパティは、ソースのポートレットプロパティのプロパティ値を受け取ります。

4. 宛先ポートレットのタイトルバーが非表示になっている場合は、[**エキスパートとして表示**] チェックボックスをオンにして、タイトルバーを表示します。

5. 宛先ポートレットのポートレットタイトルバーで、をクリックして [プロパティ] を選択します。

6. [**ワイヤリング**] タブをクリックします。

   [ワイヤリング] タブに、ワイヤリング可能な宛先ポートレットのプロパティのリストが表示されます。

7. ワイヤリングするターゲットプロパティの [**ソースポートレット**] 列で、ソースとして使用するポートレットを選択します。

8. [**ソースプロパティ**] 列で、使用するソースポートレットの特定のプロパティを指定します。

   a. [**参照**] をクリックします。

   b. 使用するプロパティをリストから選択します。

   c. [**選択**] をクリックします。

9. [**適用**] をクリックします。

10. [**保存**] をクリックします。

    ページのランタイム表示では、宛先ポートレットに、ソースポートレットに設定された値が表示されます。

ヒント:
宛先ポートレットのタイトルバーが表示されている場合は、ページ編集モードに切り替えることなくポートレットをワイヤリングできます。 ポートレットの [ワイヤリング] ページを表示するには、ポートレットタイトルバーでをクリックし、[ワイヤリング] を選択します。

プリンシパル属性からポートレットプロパティへのワイヤリング

ユーザとグループには、ポートレットにワイヤリングできるプリンシパル属性のセットがあります。たとえば、ユーザがページを表示する際に、あるポートレットが郵便番号を使用して特定の情報を表示するとします。郵便番号がプリンシパル属性プロバイダからワイヤリングによって提供された場合、郵便番号属性がディレクトリサービス内で変更されると、ポートレットは変更された属性値を使用します。

プリンシパル属性をポートレットにワイヤリングするには

1. システム管理者として、ワイヤリングするポートレットを含むページに移動して、そのページを開きます。

2. ページのタイトルバーで、 をクリックして [ページを編集] を選択し、ページ編集モードに切り替えます。

3. ポートレットのタイトルバーが非表示になっている場合は、[**エキスパートとして表示**] チェックボックスをオンにして、タイトルバーを表示します。

4. ポートレットのポートレットタイトルバーで、 をクリックして [プロパティ] を選択します。

5. [**ワイヤリング**] タブをクリックします。

6. ワイヤリングするターゲットプロパティの [**ソースポートレット**] 列で、[**その他**] を選択します。

   [ポータルリソースの選択] ウィンドウが開きます。

7. [**グローバルワイヤリングデータ**] をクリックします。

8. [**ユーザプロファイルのワイヤリング**] を [**選択された項目**] ボックスに移動し、[**選択**] をクリックします。

   [ポータルリソースの選択] ウィンドウが閉じます。

9. [**ソースプロパティ**] 列で、ソースとして使用する特定のプロパティを指定します。

   a. [**参照**] をクリックします。

      [ワイヤリングされたプロパティの選択] ウィンドウが開きます。

   b. 使用するプロパティをリストから選択します。

   c. [**選択**] をクリックします。

10. [**適用**] をクリックします。

11. [**保存**] をクリックします。

    ポートレットを含むページを表示するユーザに属する属性値を使用するようにポートレットがワイヤリングされます。

ヒント:

ポートレットのタイトルバーが表示されている場合は、ページ編集モードに切り替えることなくポートレットをワイヤリングできます。ポートレットの [ワイヤリング] ページを表示するには、ポートレットタイトルバーで をクリックし、[ワイヤリング] を選択します。

My webMethods ナビゲーションのカスタマイズについて

My webMethods のナビゲーションパネルを以下のようにカスタマイズできます。

- ナビゲーションパネルの [**アプリケーション**] セクションに選択肢を追加できます。以下のものを追加できます。
  - [**監視**] または [**管理**] サブセクションへのページの追加
  - [**アプリケーション**] セクションへのページの新規フォルダの追加

    フォルダは、[**アプリケーション**] セクション内の新規サブセクションとなります。フォルダに含まれているアイテムは、新規サブセクション内の選択肢となります。

  詳細については、407 ページの「My webMethods のナビゲーションへの選択肢の追加」を参照してください。
- ナビゲーションパネルの [**アプリケーション**] セクションの [**監視**] または [**管理**] サブセクションから選択肢を削除できます。詳細については、408 ページの「My webMethods ナビゲーションからの選択肢の削除」を参照してください。
- ナビゲーションパネルの標準的なタブおよびセクションを非表示にできます。

  [**ナビゲーション**] タブを完全に非表示にできます。[**ナビゲーション**] タブを表示する場合は、[**ナビゲーション**] タブにある [**アプリケーション**] または [**ワークスペース**] セクションを非表示にできます。[**ツール**] タブを完全に非表示にすることもできます。詳細については、409 ページの「My webMethods のナビゲーションの標準的なタブおよびセクションの非表示」を参照してください。
- My webMethods のナビゲーションの [**アプリケーション**] セクションをカスタム分類で完全に置換できます。詳細については、410 ページの「My webMethods アプリケーションのナビゲーションを独自の分類と置換」を参照してください。

My webMethods のナビゲーションへの選択肢の追加

システム管理者は、My webMethods のナビゲーションの [**アプリケーション**] セクションにカスタムのページやページのフォルダを追加できます。

My webMethods のナビゲーションの [アプリケーション] セクションに選択肢を追加するには

1. ナビゲーションパネルに追加するカスタムフォルダまたはカスタムページを作成します。詳細については、388 ページの「カスタムフォルダおよびカスタムページについて」と389 ページの「カスタムページの作成」を参照してください。

フォルダまたはページを作成する場合は、以下のいずれかの場所に保存します。

- ページを [**アプリケーション**] セクションの [**監視**] サブセクションに追加するには、以下の場所に追加します。

  [フォルダ] > [My webMethods アプリケーション] > [ファブリックタスク] > [監視]

- ページを [**アプリケーション**] セクションの [**管理**] サブセクションに追加するには、以下の場所に追加します。

  [フォルダ] > [My webMethods アプリケーション] > [ファブリックタスク] > [管理]

- フォルダを追加して、[**アプリケーション**] セクション内に新規のカスタムサブセクションを作成するには、以下の場所に追加します。

  [フォルダ] > [My webMethods アプリケーション] > [ファブリックタスク]

- カスタムサブセクションにページを追加するには、サブセクションとして使用しているフォルダにページを追加します。たとえば、「Custom Pages」という名前のフォルダを追加すると、以下の場所にページを追加できます。

  [フォルダ] > [My webMethods アプリケーション] > [ファブリックタスク] > [Custom Pages]

2. システム管理者として、ナビゲーションパネルに追加するフォルダまたはページに移動して、そのフォルダまたはページを開きます。

3. ページのタイトルバーで、 をクリックして [プロパティ] を選択します。

4. [**タスクフォルダ**] チェックボックスをオンにして、My webMethods Server が My webMethods のナビゲーションパネルにページを表示することを指定します。

5. ユーザがナビゲーションパネルからページに移動したときに、ページが My webMethods Server によって新しいタブに自動的に開かれるようにする場合は、[**開くことができる**] チェックボックスをオンにします。

6. [**適用**] をクリックして、[プロパティ] ページを閉じます。

他の My webMethods ページと同様に、ページに対してアクセス特権を設定できます。

My webMethods ナビゲーションからの選択肢の削除

システム管理者は、My webMethods のナビゲーションの [**アプリケーション**] セクションから、カスタムページ、カスタムフォルダ、およびデフォルトページを削除できます。

- カスタムページおよびカスタムフォルダは、以下のように削除できます。
  - [**フォルダ**] > [My webMethods アプリケーション] > [ファブリックタスク] 内の場所からカスタムページやカスタムフォルダを削除することによって、これらを完全に削除できます。

- ページまたはフォルダのプロパティで [**タスクフォルダ**] チェックボックスをオフにして、My webMethods においてページまたはフォルダを一時的に非表示にし、ナビゲーションにこれらが表示されないようにすることができます。

- デフォルトページを削除する場合は、ページを削除するのではなく、ページのプロパティで [**タスクフォルダ**] チェックボックスをオフにすることをお勧めします。
- デフォルトの [**監視**] または [**管理**] サブセクションのいずれかまたは両方を削除するには、[**フォルダ**] > [システム] > [シェルセクション] > [Noodle Shell 左側ナビゲーション] > [左側ナビゲーション] のプロパティを更新します。詳細については、409 ページの「My webMethods のナビゲーションの標準的なタブおよびセクションの非表示」を参照してください。

My webMethods のナビゲーションの [アプリケーション] セクションから選択肢を削除するには

1. システム管理者として、[**フォルダ**] > [My webMethods アプリケーション] > [ファブリックタスク] に移動して開きます。
2. カスタムフォルダまたはカスタムページを完全に削除するには、フォルダまたはページのタイトルバーで をクリックして削除します。
3. カスタムフォルダ、カスタムページまたはデフォルトページを非表示にするには、以下の手順に従います。
   a. フォルダまたはページのタイトルバーで をクリックし、[プロパティ] を選択して、フォルダまたはページのプロパティを開きます。
   b. [**タスクフォルダ**] チェックボックスをオフにします。
   c. [**適用**] をクリックします。

My webMethods のナビゲーションの標準的なタブおよびセクションの非表示

システム管理者は、My webMethods のナビゲーションに使用される左側ナビゲーションページのプロパティを設定して、以下の操作を実行できます。

- [**ナビゲーション**] タブまたは [**ツール**] タブを完全に非表示にできます。
- [**ナビゲーション**] タブに表示されている [**アプリケーション**] セクションまたは [**ワークスペース**] セクションのいずれかまたは両方を非表示にできます。

My webMethods のナビゲーションの標準的なタブまたはセクションを非表示にするには

1. システム管理者として、以下の場所に移動します。

[フォルダ] > [システム] > [シェルセクション] > [Noodle Shell 左側ナビゲーション] > [左側ナビゲーション]

2. [左側ナビゲーション] ページの行で をクリックし、[プロパティ] を選択します。

3. ページの [**ポートレット設定**] セクションで、以下のチェックボックスをオンまたはオフにして、タブとセクションを表示または非表示にします。

次の表に、[ポートレット設定] セクションで使用可能な表示オプションを示します。

チェックボックス	説明
[Show Workspaces]	[**ナビゲーション**] タブの [**ワークスペース**] セクションを表示または非表示にします。
[Show Applications]	[**ナビゲーション**] タブの [**アプリケーション**] セクションを表示または非表示にします。
[Show Tools]	[**ツール**] タブを表示または非表示にします。
[Show Navigate]	[**ナビゲーション**] タブを表示または非表示にします。

4. [**適用**] をクリックします。

My webMethods アプリケーションのナビゲーションを独自の分類と置換

システム管理者は、My webMethods のナビゲーションパネルの [**アプリケーション**] セクションをカスタム分類と完全に置換できます。

カスタム分類を作成するには、使用する分類を含む、代替アプリケーションルートページを作成します。その後、My webMethods がナビゲーションパネルとして使用している [左側ナビゲーション] ページのプロパティを、新しいアプリケーションルートを指すように設定します。

My webMethods の [アプリケーション] 分類をカスタム分類で置換するには

1. システム管理者として、アプリケーションルートとして使用する新しいページを作成します。以下のいずれかの場所にページを作成することをお勧めします。

   - フォルダ
   - **[フォルダ] > [パブリックフォルダ]**

   ページを作成するときに、ナビゲーションパネルで表示する名前をページに対して割り当てます。割り当てる名前は、[**アプリケーション**] セクションの新しい名前となります。

2. 新しいアプリケーションルートページにエイリアスを割り当てます。

a. 新しいアプリケーションルートページの行で をクリックし、[プロパティ] を選択します。

b. [**エイリアス**] フィールドで、[**追加**] をクリックします。

c. テキストボックスに、新しいアプリケーションルートに指定するエイリアスを割り当てて、[**OK**] をクリックします。

d. [**適用**] をクリックします。

3. 新しいアプリケーションルートページ内に分類を作成します。

- 分類にサブセクションを追加するには、コンテナとして使用するページまたはフォルダを追加します。追加したページまたはフォルダに、分類内のサブセクション名として使用する名前を割り当てます。
- 情報を表示するページを追加するには、ページを追加します。分類で表示する名前をページに対して割り当てます。

カスタムフォルダおよびカスタムページの作成の詳細については、388 ページの「カスタムフォルダおよびカスタムページについて」および 389 ページの「カスタムページの作成」を参照してください。

4. My webMethods Server が My webMethods のナビゲーションとして使用している [左側ナビゲーション] ページのプロパティを、新しいカスタムアプリケーションルートを指すように設定します。

a. 次の場所に移動します。

[フォルダ] > [システム] > [シェルセクション] > [Noodle Shell 左側ナビゲーション] > [左側ナビゲーション]

b. [左側ナビゲーション] ページの行で をクリックし、[プロパティ] を選択します。

c. [**Applications Root**] フィールドに、作成したアプリケーションルートページに割り当てたエイリアスの名前を入力します。

d. [**適用**] をクリックします。

bean の有効期限ポリシーの変更

デフォルトでは、My webMethods ユーザが前のフォルダ (タスク受信トレイなど) に戻ると、フォルダはリフレッシュされ、フォルダに対する保存されていない変更は失われます。システム管理者は、ユーザが既存のデータを含むタブに戻ったときにそれが表示されるように、Fabric フォルダまたはワークスペーステンプレートの bean の有効期限ポリシーを変更できます。ポリシーを設定すると、ユーザが My webMethods でそのフォルダまたはワークスペースを表示するときに常に適用されます。

bean の有効期限ポリシーを変更するには

1. システム管理者として、以下のいずれかを行います。
   - Fabric フォルダの場合、[**フォルダ**] > [My webMethods アプリケーション] > [ファブリックタスク] に移動します。

     bean の有効期限ポリシーを変更できるフォルダは、 で識別されます。
   - ワークスペーステンプレートの場合、[**フォルダ**] > [システム] > [テンプレート] > [ワークスペーステンプレート] > [デフォルトワークスペーステンプレート] に移動します。
2. Fabric フォルダのタイトルバーまたはデフォルトワークスペーステンプレートで、 をクリックし、[プロパティ] を選択します。
3. [**beanExpirePolicy**] フィールドに、「`do_not_expire`」と入力します。
4. [**適用**] をクリックします。

変更するフォルダごとに、この変更を個別に行います。

My webMethods のルックアンドフィールのカスタマイズについて

システム管理者は、My webMethods で使用されているスキンを更新することによって、My webMethods のルックアンドフィールを変更できます。My webMethods のルックアンドフィールを以下のようにカスタマイズできます。

- Software AG のロゴを自社のロゴなどに変更できます。
- My webMethods ユーザインタフェースで使用されている色を変更できます。
- カスタムのシェルおよびスキンを使用して、My webMethods のルックアンドフィールを完全にカスタマイズできます。

My webMethods ユーザインタフェースのロゴの置換

システム管理者は、My webMethods で使用されているスキンのイメージを更新して、My webMethods ユーザインタフェースで使用されているイメージを変更できます。インタフェース上部に表示される Software AG のロゴを変更するには、logo.gif イメージを更新します。スキンの詳細については、441 ページの「スキンのカスタマイズ」を参照してください。

My webMethods では、「Noodle - Twilight」スキンが使用されています。「Noodle - Twilight」スキンは更新しないで、このスキンのコピーを作成し、そのコピーを更新して、My webMethods で変更したスキンのコピーを使用するように設定することをお勧めします。

My webMethods ユーザインタフェースのロゴを置換するには

1. 「Noodle - Twilight」スキンのコピーを作成します。

   a. システム管理者として、[**フォルダ**] > [管理用フォルダ] > [Administration Dashboard] > [ユーザインタフェース] > [スキン管理] に移動します。

   b. ページの上部にある [新規スキンの作成] リンクをクリックします。

   c. [新規スキンの作成] ページで、次を指定します。

次の表に、新しいスキンを作成するときに指定する必要がある情報を示します。

フィールド	説明
[システム名]	文字、数字および下線のみを含む短い名前。My webMethods Server では、この名前が内部的に使用されます。
[表示名]	スキン名をユーザインタフェースに表示する場合に My webMethods Server で使用される説明的な名前。
[親スキン]	`Noodle - Twilight`

   d. [**保存**] をクリックします。

2. をクリックして [編集] を選択し、新しいスキンのプロパティを編集します。

3. [**画像**] タブをクリックします。

4. logo.gif イメージのイメージを更新します。手順については、448 ページの「スキン内のイメージの置き換え」を参照してください。

5. 更新したスキンを使用するように My webMethods を設定します。

   a. [**フォルダ**] > [管理用フォルダ] > [Administration Dashboard] > [ユーザインタフェース] > [スキンルールの管理] に移動します。

   b. [**My webMethods**] リンクをクリックして、My webMethods で使用されているスキンルールを開きます。

   c. [**ルールの変更**] タブの [**結果**] リストで、新しいスキンを選択します。

   d. [**ルールの更新**] をクリックします。

My webMethods ユーザインタフェースの表示色の変更

システム管理者は、My webMethods で使用されているスキンの色を更新することによって、My webMethods ユーザインタフェースで使用されている色を変更できます。スキンの詳細については、441 ページの「スキンのカスタマイズ」を参照してください。

My webMethods では、「Noodle - Twilight」スキンが使用されています。「Noodle - Twilight」スキンは更新しないで、このスキンのコピーを作成し、そのコピーを更新して、My webMethods で変更したスキンのコピーを使用するように設定することをお勧めします。

My webMethods ユーザインタフェースの色を変更するには

1. 「Noodle - Twilight」スキンのコピーを作成します。

   a. システム管理者として、[**フォルダ**] > [管理用フォルダ] > [Administration Dashboard] > [ユーザインタフェース] > [スキン管理] に移動します。

   b. ページの上部にある [**新規スキンの作成**] リンクをクリックします。

   c. 新しいスキンのフィールドに入力します。

次の表に、My webMethods の新しいスキンを作成するときに設定する必須フィールドを示します。

フィールド	説明
[**システム名**]	文字、数字および下線のみを含む短い名前。My webMethods Server では、この名前が内部的に使用されます。
[**表示名**]	スキン名をユーザインタフェースに表示する場合に My webMethods Server で使用される説明的な名前。
[**親スキン**]	`Noodle - Twilight`

   d. [**保存**] をクリックします。

2. をクリックして [編集] を選択し、新しいスキンのプロパティを編集します。

3. [**色**] タブをクリックします。

4. 色を更新します。手順については、449 ページの「カラーピッカーを使用した色の置き換え」を参照してください。

5. 更新したスキンを使用するように My webMethods を設定します。

   a. [**フォルダ**] > [管理用フォルダ] > [Administration Dashboard] > [ユーザインタフェース] > [スキンルールの管理] に移動します。

   b. [**My webMethods**] リンクをクリックして、My webMethods で使用されているスキンルールを開きます。

   c. [**ルールの変更**] タブの [**結果**] リストで、新しいスキンを選択します。

d. [**ルールの更新**] をクリックします。

My webMethods へのカスタムのスキンおよびシェルの適用

システム管理者は、カスタムのスキンおよびシェルを使用するように設定することによって、My webMethods のルックアンドフィールを完全に変更できます。

My webMethods にカスタムのスキンおよびシェルを適用するには

1. 使用するカスタムシェルを作成します。手順については、469 ページの「My webMethods Server でのシェルの使用」を参照してください。

2. 使用するカスタムスキンを作成します。手順については、441 ページの「スキンのカスタマイズ」を参照してください。

3. カスタムシェルを使用するように My webMethods を設定します。

   a. [**フォルダ**] > [管理用フォルダ] > [Administration Dashboard] > [ユーザインタフェース] > [シェルルールの管理] に移動します。

   b. [**My webMethods**] リンクをクリックして、My webMethods で使用されているシェルルールを開きます。

   c. [**ルールの変更**] タブの [**結果**] リストで、カスタムシェルの名前を選択します。

   d. [**ルールの更新**] をクリックします。

4. カスタムスキンを使用するように My webMethods を設定します。

   a. [**フォルダ**] > [管理用フォルダ] > [Administration Dashboard] > [ユーザインタフェース] > [スキンルールの管理] に移動します。

   b. [**My webMethods**] リンクをクリックして、My webMethods で使用されているスキンルールを開きます。

   c. [**ルールの変更**] タブの [**結果**] リストで、カスタムスキンの名前を選択します。

   d. [**ルールの更新**] をクリックします。

My webMethods への簡単なフロントエンドページの作成

少数の My webMethods ページのみにアクセスする必要があるユーザが存在する場合、そのユーザがこれらのページに簡単にアクセスできるようにするために、ユーザが必要とするページへのリンクのみを備えた簡単なページを作成できます。

フロントエンドページに含めるリンクでは、ページのエイリアスを使用した URL を使用できます。一部の My webMethods アプリケーションページには、デフォルトでエイリアスが割り当てられています。使用するアプリケーションページにエイリアスがない場合は、ページにエイリアスを割り当てることができます。エイリアスを割り当てると、My webMethods のナビゲーションパネルからページを選択する代わりに、ページのエイリアスを使用した URL を入力することによってページに直接移動できます。

My webMethods への簡単なフロントエンドページを作成するには

1. フロントエンドページに含めるアプリケーションページを特定します。

2. 各アプリケーションページに対して、そのページのエイリアスを特定するか、またはページにエイリアスがない場合はエイリアスを割り当てます。

   a. システム管理者として、以下の場所に移動します。

      [フォルダ] > [My webMethods アプリケーション] > [ファブリックタスク]

   b. さらに、必要なアプリケーションページに移動して、そのページを開きます。

   c. ページのタイトルバーで、 をクリックして [プロパティ] を選択します。

   d. 画面の [**エイリアス**] セクションで、ページのエイリアスを確認します。ページにエイリアスが存在しない場合は、エイリアスを割り当てます。

   e. [**適用**] をクリックして、[プロパティ] ページを閉じます。

3. アプリケーションページへのリンクを含む Web ページを作成します。以下に示す URL を使用します。変数の意味は以下のとおりです。

   - *host_name* および *port_number* は、My webMethods Server のホスト名とポート番号です。
   - *alias* は、アプリケーションページのエイリアス名です。

   http://*host_name*:*port_number*/*alias*

   たとえば、My webMethods Server のホスト名とポートが「mws.company.com:8585」であり、「webm.apps.workflow.inbox」というエイリアスを持つタスク受信トレイページにアクセスする必要がある場合は、以下の URL を使用します。

http://mws.company.com:8585/webm.apps.workflow.inbox

シングルサインオン用のリンクの作成

シングルサインオンとは、ユーザがあるアプリケーションにログインすると、別途ログインすることなく他のアプリケーションを使用できる機能を指します。My webMethods Server では、セキュリティ情報交換の XML ベースのフレームワークである SAML を介したシングルサインオンをサポートしています。

シングルサインオンを利用するには、ユーザがソースサーバとターゲットエンティティの両方で認識されている必要があります。ほとんどの場合、同じディレクトリサービスを使用することでユーザを共通に認識できます。シングルサインオンのターゲットとしてサーバを設定する方法の詳細については、282 ページの「My webMethods Server シングルサインオンの設定」を参照してください。

任意のページで、サーバなどの SAML ターゲットエンティティへのリンクを追加できます。ターゲットがソースサーバからの SAML アサーションを受け入れる場合は、既知のユーザがリンクをクリックしたときに、ログインクレデンシャルは要求されません。ターゲットエンティティがソースサーバからの SAML アサーションを受け入れない場合、またはユーザがターゲットエンティティで認識されていない場合は、ログインクレデンシャルが要求されることがあります。

SAML 仕様では、アーティファクトレシーバと呼ばれる媒介が、ターゲットの Web アプリケーションに代わって認証を実行できます。このような場合、SAML ソースは 2 つの URL を必要とします。一方はアーティファクトレシーバ用で、もう一方はターゲット Web アプリケーション用です。

編集権限のある任意のページに 1 つ以上の SAML リンクを配置できます。

ソースページに SAML リンクを作成するには

1. ページの右上で をクリックし、[ページを編集] を選択します。

2. [使用可能なポートレット] パネルの [ルート] リストで、[リンク] をクリックします。

3. [使用可能なポートレット] パネルの [リンク] リストで、[シングルサインオンリンク] ポートレットをドラッグし、ページ上のリンクを追加する位置にドロップします。

   カーソルを有効なページ位置に移動するたびに、カーソル位置の下に赤いボックスが表示されます。このボックスは、マウスボタンを離した場合にポートレットが配置される場所を示します。

4. ページ制御領域の左側で、[**保存**] をクリックします。

5. シングルサインオンポートレットのタイトルバーの右端で、 をクリックし、[プロパティ] を選択します。

6. [プロパティ] で次を指定します。

次の表に、シングルサインオンリンクのポートレットを設定するために指定するプロパティを示します。

プロパティ	説明
[名前]	[シングルサインオンリンク] をリンクの内容に合ったテキストで置換します。
[SAML 認証 URL]	ターゲットコンピュータ上のリソースの URL を入力します。サーバ上の任意のページをターゲットにできます。SAML アーティファクトレシーバを介して Web アプリケーションに接続している場合は、このフィールドをアーティファクトレシーバ URL 用に使用します。
[POST または GET の使用]	ターゲットコンピュータにデータを渡す方法を決定します。 POST　HTTP 要求の本文でデータを渡します。デフォルトである POST は、シングルサインオンデータに対して推奨される方法です。 [GET を使用]　URL に付加した疑問符の後のストリングとしてデータを渡します。
[アーティファクトパラメータ名]	別のサーバまたは他の webMethods 製品との SAML 接続である場合は、デフォルト値の「SAMLart」を変更しないでください。サードパーティソースへの SAML 接続である場合は、サードパーティアプリケーションで使用されているアーティファクトパラメータ名を入力します。
[アプリケーションターゲット URL]	SAML アーティファクトレシーバの URL を [SAML 認証 URL] フィールドに入力した場合は、Web アプリケーションの URL を入力します。それ以外の場合は、このフィールドを空白のままにします。

7. [**適用**] をクリックします。

23 My webMethods Server でのワークスペースの管理

- ワークスペースについて 420
- ワークスペースの管理タスク 420
- ワークスペース開発用のエキスパートユーザ機能 424
- [ワークスペース管理] ページから実行できるワークスペースアクション 430
- My webMethods ツールナビゲーションについて 438

ワークスペースについて

ワークスペースは、特定の目的のためにユーザが作業領域として作成および使用するページです。 ユーザは、ワークスペースにポートレットをドラッグして、ワークスペースのコンテンツを作成します。たとえば、ユーザがある問題を解決する必要がある場合に、その問題に関する情報収集用のワークスペースを作成することができます。この場合は、問題に関連するワークスペースに添付ファイルツールを追加して、そのワークスペースにファイルを添付します。問題を示す画面ショットがある場合は、イメージツールをワークスペースに追加することができます。

他のユーザとワークスペースを共有して、複数のユーザが共同で作業することもできます。たとえば、添付ファイルツールをワークスペースに追加すると、ワークスペースを共有するユーザは、ファイルのアップロードと共有を実行できます。ワークスペースの所有者は、ワークスペースを共有する他のユーザが実行できるアクションを決定する権限を設定します。

ワークスペースは必要に応じていくつでも作成できます。ワークスペース作成のきっかけとなった問題が解決されたなどの理由により、ワークスペースが不要になったときは、そのワークスペースを単に削除するだけです。

ワークスペースの作成や共有およびワークスペースへのポートレットの追加など、ワークスペースの基本的な使用方法の詳細については、『*Working with My webMethods*』を参照してください。

システム管理者および My webMethods 管理者は、ユーザのワークスペース機能を制限または拡張できます。詳細については、420 ページの「ワークスペースの管理タスク」を参照してください。

- 機能を制限するには、管理者がユーザの権限を設定して、標準で許可されている機能を拒否します。たとえば、ユーザが新しいワークスペースを作成する機能特権を拒否できます。
- 機能を拡張するには、管理者がユーザの権限を設定して、標準で許可されていない機能を許可します。 たとえば、次の権限を付与できます。
  - ユーザをエキスパートユーザにする機能特権。エキスパートユーザは、ワークスペースの開発をサポートする追加プロパティおよびメニューアクションにアクセスできます。詳細については、424 ページの「ワークスペース開発用のエキスパートユーザ機能」を参照してください。
  - [ナビゲーション] > [アプリケーション] > [管理] > [システム全体] > [ワークスペース管理] ページへのアクセス権。ユーザは、このページからワークスペースに対してアクションを実行できます。詳細については、430 ページの「[ワークスペース管理] ページから実行できるワークスペースアクション」を参照してください。

さらに、システム管理者は My webMethods ナビゲーションの [**ツール**] タブをカスタマイズできます。詳細については、438 ページの「My webMethods ツールナビゲーションについて」を参照してください。

ワークスペースの管理タスク

システム管理者および My webMethods 管理者は、次のタスクを実行して、ユーザのワークスペース機能を拡張または制限できます。

- ユーザに対する [ナビゲーション] > [アプリケーション] > [管理] > [システム全体] > [ワークスペース管理] ページへのアクセス権の付与。このアクセス権があると、ユーザはワークスペースの検索およびワークスペースに対するアクションを実行できます。詳細については、421 ページの「ユーザに対する [ワークスペース管理] ページへのアクセスの許可」を参照してください。
- ワークスペースの表示および作成など、ユーザによる基本的なワークスペースアクション実行の禁止。詳細については、422 ページの「ワークスペース機能特権」と423 ページの「ユーザが実行可能なワークスペース機能の制御」を参照してください。
- ユーザに対するエキスパートワークスペース開発特権の付与。この特権は、ワークスペースの作成時に使用できる追加機能をユーザに提供します。詳細については、422 ページの「ワークスペース機能特権」、423 ページの「ユーザが実行可能なワークスペース機能の制御」、424 ページの「ワークスペース開発用のエキスパートユーザ機能」を参照してください。
- すべてのユーザの My webMethods ナビゲーションから [**ワークスペース**] セクションを削除する。詳細については、409 ページの「My webMethods のナビゲーションの標準的なタブおよびセクションの非表示」を参照してください。
- My webMethods ナビゲーションの [**ツール**] タブに表示される分類のカスタマイズ。詳細については、439 ページの「My webMethods ワークスペースツールのカスタマイズ」を参照してください。

システム管理者がアクションを実行して、ユーザのワークスペースを修正または削除する必要がある場合に、[ナビゲーション] > [アプリケーション] > [管理] > [システム全体] > [ワークスペース管理] ページ経由ではこの操作を実行できないときは、ユーザのパーソナルフォルダにあるワークスペースにアクセスして、問題の修正を試みることができます。My webMethods Server は、ワークスペースを作成したユーザのパーソナルフォルダに、それぞれのワークスペースを保存します。たとえば、「jsmith」というユーザがワークスペースを作成した場合、システム管理者のユーザインタフェースを使用して、[フォルダ] > [ユーザ] > [jsmith のルートフォルダ] > [ワークスペース] で、このワークスペースに移動できます。

ユーザに対する [ワークスペース管理] ページへのアクセスの許可

My webMethods には、[ナビゲーション] > [アプリケーション] > [管理] > [システム全体] > [ワークスペース管理] ページがあります。このページを使用すると、ユーザがワークスペースに対するアクションを実行できます。詳細については、430 ページの「[ワークスペース管理] ページから実行できるワークスペースアクション」を参照してください。

デフォルトでは、My webMethods 管理者のみが [ワークスペース管理] ページにアクセスできます。エンドユーザはアクセスできません。ただし、エンドユーザは、My webMethods ナビゲーションの [**ワークスペース**] セクションにある右クリックメニューおよび開いているワークスペースにあるタブのメニューから、同じアクションの多くを実行できます。[ワークスペース管理] ページ経由でのみ使用可能なワークスペースアクションは、次のとおりです。

- ワークスペースの検索
- ファイルへのワークスペースのエクスポート
- ファイルにエクスポート済みのワークスペースのインポート

システム管理者および My webMethods 管理者は権限を割り当てて、ユーザ、グループ、および役割に [ワークスペース管理] ページへのアクセスを許可できます。

ユーザに [ワークスペース管理] ページへのアクセスを許可するには

1. [権限の管理] ページに移動します。

   - My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [システム全体] > [権限の管理]
   - システム管理者として次のように選択します。[**フォルダ**] > [管理用フォルダ] > [Administration Dashboard] > [設定] > [権限の管理]

2. [権限の管理] ページを使用して、[ワークスペース管理] ページへのアクセス権を付与およびアクセス特権を設定するユーザ、グループ、および役割を選択します。

   特権を設定するときは、[**権限**] ツリーで以下を選択します。

   [アクセス特権] > [管理] > [システム全体] > [ワークスペース管理]

   [権限の管理] ページを使用してアクセス特権を設定する方法の詳細については、179 ページの「アクセス特権および機能特権の管理」を参照してください。

ワークスペース機能特権

ワークスペース機能特権は、ユーザがワークスペースに対して実行できるアクションを管理します。システム管理者および My webMethods 管理者は、ユーザ、グループ、および役割のデフォルト権限を変更できます。詳細については、423 ページの「ユーザが実行可能なワークスペース機能の制御」を参照してください。

次の表に、My webMethods で使用可能なワークスペース機能特権とそのデフォルト設定を示します

機能特権	デフォルト設定	説明
ワークスペースの表示	許可済み	ユーザがワークスペースを表示できるかどうかを制御します。 ユーザがワークスペースの表示機能特権を拒否されている場合、My webMethods Server は My webMethods ナビゲーションの [**ナビゲーション**] タブに [**ワークスペース**] セクションを表示しません。 **注:** ワークスペースの表示機能特権が拒否されているときでも、ユーザが [ワークスペース管理] ページにアクセスできる場合は、ワークスペースを検索して開くことができます。詳細については、421 ページの「ユーザに対する [ワークスペース管理] ページへのアクセスの許

機能特権	デフォルト設定	説明
		可」と433ページの「ワークスペースを開く」を参照してください。
ワークスペースの編集	許可済み	ユーザがワークスペースにポートレットを追加できるかどうかを制御します。 ユーザがワークスペースの編集機能特権を拒否されている場合、My webMethods Server は My webMethods ナビゲーションの [**ツール**] タブを表示しません。 **注:** ワークスペースの編集機能特権が拒否されている場合でも、ユーザはワークスペースプロパティの設定、ワークスペース内でのポートレットの再配置、ポートレットプロパティの設定、およびワークスペースからのポートレットの削除は実行できます。
ワークスペースの作成	許可済み	ユーザが新しいワークスペースを作成できるかどうかを制御します。 ユーザがワークスペースの作成機能特権を拒否されている場合、My webMethods Server は My webMethods ナビゲーションに [**新規**] タブを表示しません。
ワークスペースのインポート	許可済み	ユーザがファイルにエクスポート済みのワークスペースをインポートできるかどうかを制御します。 ユーザがワークスペースのインポート機能特権を拒否されている場合、My webMethods Server は [ワークスペース管理] ページの [**インポート**] ボタンを表示しません。
エキスパートワークスペース開発	拒否	ワークスペースおよびページの作成時に使用できる追加機能をユーザに提供します。エキスパートワークスペース開発機能特権を持つユーザは、追加プロパティおよびメニューアクションにアクセスできます。詳細については、424ページの「ワークスペース開発用のエキスパートユーザ機能」を参照してください。

ユーザが実行可能なワークスペース機能の制御

システム管理者および My webMethods 管理者が、ユーザ、グループ、および役割に 422 ページの「ワークスペース機能特権」で説明する機能特権を付与または拒否するには、以下の手順に従います。

ユーザにワークスペース機能特権を付与または拒否するには

1. [権限の管理] ページに移動します。

- My webMethods で次のように選択します。[**ナビゲーション**] > [アプリケーション] > [管理] > [システム全体] > [権限の管理]
- システム管理者として次のように選択します。[**フォルダ**] > [管理用フォルダ] > [Administration Dashboard] > [設定] > [権限の管理]

2. [権限の管理] ページを使用して、使用および機能特権を付与または拒否するユーザ、グループ、および役割を選択します。

   ワークスペース機能特権を付与または拒否するには、[**権限**] ツリーの次の部分に移動し、422 ページの「ワークスペース機能特権」の説明にある特権を付与または拒否します。

   [機能特権] > [全般]

   [権限の管理] ページを使用して機能特権を設定する方法の詳細については、179 ページの「アクセス特権および機能特権の管理」を参照してください。

ワークスペース開発用のエキスパートユーザ機能

エキスパートユーザは、エキスパートワークスペース開発機能特権を割り当てられているユーザです。 管理者がこの機能特権を割り当てる方法の詳細については、423 ページの「ユーザが実行可能なワークスペース機能の制御」を参照してください。

エキスパートユーザは、次の追加プロパティおよびメニューアクションにアクセスできます。

- **ワークスペースプロパティ** ワークスペースのレイアウトを効率的に管理します。詳細については、425 ページの「エキスパートユーザ用のワークスペースプロパティ」を参照してください。
- **ワークスペース内のポートレットのプロパティ** ポートレットのコンテンツおよびワークスペースでのポートレットの表示方法を効率的に管理します。たとえば、エキスパートユーザは、ポートレットのタイトルバーが表示されないようにポートレットを設定できます。詳細については、428 ページの「エキスパートユーザ用のポートレットプロパティ」を参照してください。
- ポートレットメニューアクション
  - My webMethods Server がワークスペースのレンダリング時にポートレットを配置するよう指定する
  - ポートレットの権限を設定する

  詳細については、430 ページの「エキスパートユーザ用のポートレットメニューオプション」を参照してください。

注:
ワークスペースだけではなく、通常のページを使用する場合でも、エキスパートユーザには、これらの追加プロパティおよびポートレットメニューオプションが表示されます。

エキスパートユーザ用のワークスペースプロパティ

[全般] タブ

[**全般**] タブのワークスペースプロパティは、ユーザのエキスパートワークスペース開発特権の有無に関係なく、まったく同じです。これらのワークスペースプロパティの詳細については、『*Working with My webMethods*』を参照してください。

[レイアウト] タブ

エキスパートワークスペース開発特権の有無に関係なく、ユーザは [**レイアウト**] タブを使用できますが、一部のプロパティはエキスパートユーザ専用です。

次の表に、[**レイアウト**] タブのワークスペースプロパティを示します。

プロパティ	使用可能ユーザ	説明
[表示形式]	エキスパートユーザ	このプロパティは、列、行、およびポートレットの枠と非表示のポートレットタイトルバーを表示するかどうかを示します。 ■ 枠または非表示のタイトルバーを表示しない場合は、[**エンドユーザ**] を選択します。 ■ 枠および非表示のタイトルバーを表示する場合は、[**エキスパートユーザ**] を選択します。 枠があると、ポートレットを配置するときに役立ちます。また、ポートレットのプロパティを設定してタイトルバーを非表示にする場合は、エキスパートユーザとしてページを表示し、タイトルバーを再表示して、ポートレットを移動したり、プロパティにアクセスしたりするなど、ポートレットに対してアクションを実行します。
[編集可能なキャンバス]	エキスパートユーザ	このプロパティは、ワークスペースを表示するユーザが、ポートレットの再配置とサイズ変更の両方またはいずれか一方を実行できるようにするかを示します。 ユーザがワークスペースの使用中に、誤ってレイアウトを変更しないようにするには、[**編集可能なキャンバス**] チェックボックスをオフにします。ワークスペースにフリーフォームレイアウトを選択すると、ユーザがポートレットをクリックしただけで、誤ってページが変更される可能性があります。 **注:**

プロパティ	使用可能ユーザ	説明
		[**編集可能なキャンバス**] チェックボックスをオンにすると、ユーザがレイアウトを変更できます。ユーザがワークスペースの編集権限を拒否されている場合でもレイアウトの変更はできますが、編集内容の保存はできません。ただし、[**編集可能なキャンバス**] チェックボックスをクリアすると、すべてのユーザがレイアウトを変更できなくなります。
[列]	すべてのユーザ	このプロパティでは、ワークスペースに使用するレイアウトを指定します。レイアウトは、ワークスペース内でポートレットを配置する方法の定義です。以下のいずれかのレイアウトを定義できます。 ■ **列レイアウト**。このレイアウトでは、ワークスペース内のすべてのポートレットが列内に整列されます。列内でポートレットを重ね合わせることはできません。ワークスペースに定義できる列は 1～4 です。 ■ **フリーフォームレイアウト**。このレイアウトでは、ワークスペース内の任意の場所にポートレットを配置できます。ポートレットを部分的に重ね合わせたり、互いに完全に重ねたりすることもできます。これは新規ワークスペースのデフォルトです。
[属性]	エキスパートユーザ	列レイアウトを使用する場合、このプロパティで作成者のみが使用できる列の属性を指定します。標準の My webMethods Server は、属性を使用する機能を提供しません。ただし、属性を利用するカスタムコードは記述できます。
[幅]	すべてのユーザ	列レイアウトを使用する場合、このプロパティで列の幅に使用するワークスペースの割合を指定します。
[行の折り返し]	すべてのユーザ	列レイアウトを使用する場合、このプロパティでポートレット内の長いテキスト行を折り返して列サイズに合わせるかどうかを指定します。
[水平配置]	すべてのユーザ	列レイアウトを使用する場合、このプロパティで列内のポートレットを横揃えする方法を指定します。
[垂直配置]	すべてのユーザ	列レイアウトを使用する場合、このプロパティで列内のポートレットを縦揃えする方法を指定します。
[CSS クラス]	エキスパートユーザ	列レイアウトを使用する場合、このプロパティで列に適用する CSS クラスを指定します。先頭のピリオ

プロパティ	使用可能ユーザ	説明
		ドを省略して、クラスの名前を入力します。たとえば、「.nav」クラスの場合は「nav」と入力します。 ワークスペースに含まれる、以下のいずれかの CSS スタイルシートによって定義される CSS クラスを指定します。 ■ 現在のスキンで使用されている ■ ワークスペースコンテンツまたは現在のシェル内のカスタムポートレットに含まれている
[CSS スタイル]	エキスパートユーザ	列レイアウトを使用する場合、このプロパティで列に適用するスタイルを指定します。CSS ファイルで使用可能な任意のスタイルを指定します。たとえば、「`border: 1pt dashed red`」と入力すると、赤の破線がポートレットの枠として表示されます。
[スキン背景画像]	エキスパートユーザ	列レイアウトを使用する場合、このプロパティで列の背景画像として使用する、現在のスキンのイメージを指定します。 スキンプロパティの名前を入力します。たとえば、スキンのメインロゴイメージを使用するには、「`images/logo.gif`」と入力します。システム管理者は、([Administration Dashboard] > [ユーザインタフェース] > [スキン管理] ポートレット経由で) スキンエディタにアクセスし、[画像] ページでイメージを検索して、イメージのスキンプロパティ名を指定できます。スキンプロパティの「images/」の後には、プロパティ名が続きます (「images/logo.gif」など)。 **注:** 現在のスキンに含まれていないイメージを適用するには、[**CSS スタイル**] フィールドで標準の CSS 背景画像プロパティを指定します。[**CSS スタイル**] フィールドでプロパティを指定して、背景を繰り返すか、中央に配置するか、ページと共にスクロールするかどうかなど、背景の表示方法を制御することもできます。

エキスパートユーザ用のポートレットプロパティ

[全般] タブ

[**全般**] タブのポートレットプロパティは、ユーザのエキスパートワークスペース開発特権の有無に関係なく、まったく同じです。これらのワークスペースプロパティの詳細については、『*Working with My webMethods*』を参照してください。

[設定] タブ

[**設定**] タブは、ユーザのエキスパートワークスペース開発特権の有無に関係なく使用可能です。[**設定**] タブがあるのは一部のポートレットのみです。

[**設定**] タブのプロパティは各ポートレットに固有であり、通常は My webMethods Server がポートレットに表示する情報を定義します。たとえば、[HTML テキストツール] の場合、[**設定**] タブでポートレットに表示されるテキストを指定できます。

[レイアウト] タブ

[**レイアウト**] タブは、エキスパートワークスペース開発特権を持つユーザのみ使用可能です。次の表は、[**レイアウト**] タブのポートレットプロパティについて説明しています。。

プロパティ	説明
[属性]	作成者のみが使用する属性。標準の My webMethods Server は、属性を使用する機能を提供しません。ただし、属性を利用するカスタムコードは記述できます。
[幅]	ポートレットの幅の指定。ポートレットの幅に使用するピクセル数を指定します。
[高さ]	ポートレットの高さの指定。ポートレットの高さに使用するピクセル数を指定します。
[タイトルバー]	My webMethods Server がポートレットのタイトルバーを表示するかどうか。タイトルバーを表示するには、チェックボックスをオンにします。タイトルバーを非表示にするには、チェックボックスをオフにします。 **注:** タイトルバーを非表示にしている場合は、ワークスペースプロパティの [**レイアウト**] タブ上にある [**表示形式**] プロパティを使用してタイトルバーを一時的に表示すると、ワークスペースのポートレットを操作できます。
[枠]	My webMethods Server がポートレットの枠を表示するかどうか。枠を表示するには、チェックボックスをオンにします。枠を非表示にするには、チェックボックスをオフにします。 **注:**

プロパティ	説明
	タイトルバーを非表示にしている場合は、ワークスペースプロパティの [**レイアウト**] タブ上にある [**表示形式**] プロパティを使用してタイトルバーを一時的に表示すると、ワークスペースのポートレットを操作できます。
[最小化]	My webMethods Server がポートレットを含むワークスペースを表示するとき、初期状態でポートレットを最小化して表示するかどうか。ユーザは、ポートレットを復元して、そのコンテンツを表示できます。
[CSS クラス]	ポートレットに適用する CSS クラス。先頭のピリオドを省略して、クラスの名前を入力します。たとえば、「.nav」クラスの場合は「`nav`」と入力します。 ワークスペースに含まれる、以下のいずれかの CSS スタイルシートによって定義される CSS クラスを指定します。 ■ 現在のスキンで使用されている ■ ワークスペースコンテンツまたは現在のシェル内のカスタムポートレットに含まれている
[CSS スタイル]	ポートレットに適用するスタイル。CSS ファイルで使用可能な任意のスタイルを指定します。たとえば、「`border: 1pt dashed red`」と入力すると、赤の破線がポートレットの枠として表示されます。
[スキン背景画像]	ポートレットの背景画像として使用する、現在のスキンのイメージ。 スキンプロパティの名前を入力します。たとえば、スキンのメインロゴイメージを使用するには、「`images/logo.gif`」と入力します。エキスパートユーザは、使用可能なプロパティの情報をシステム管理者に確認できます。システム管理者は、([Administration Dashboard] > [ユーザインタフェース] > [スキン管理] ポートレット経由で) スキンエディタにアクセスし、[画像] ページでイメージを検索して、イメージのスキンプロパティ名を指定できます。スキンプロパティの「images/」の後には、プロパティ名が続きます (「images/logo.gif」など)。 **注:** 現在のスキンに含まれていないイメージを適用するには、[**CSS スタイル**] フィールドで標準の CSS 背景画像プロパティを指定します。[**CSS スタイル**] フィールドでプロパティを指定して、背景を繰り返すか、中央に配置するか、ページと共にスクロールするかどうかなど、背景の表示方法を制御することもできます。

[メタデータ] タブ

[**メタデータ**] タブは、エキスパートワークスペース開発特権を持つユーザのみ使用可能です。[**メタデータ**] タブのプロパティは各ポートレットに固有であり、通常は My webMethods Server がポートレットに表示する情報を定義します。

エキスパートユーザ用のポートレットメニューオプション

エキスパートユーザが追加のメニューアクションにアクセスするには、ポートレットのタイトルバーにある ▽ をクリックします。次の表に、エキスパートワークスペース開発特権を持つユーザのみが使用可能なポートレットメニューアクションを示します。

アクション	説明
[自動配置]	My webMethods Server がワークスペースをレンダリングする場合の、ポートレットの配置方法を定義します。[**自動配置**] は次のいずれかに切り替わります。 ■ **オン**: My webMethods Server はポートレットを自動的に配置します。ポートレットと共に保存されている配置情報はすべて無視されます。 ■ **オフ**: My webMethods Server は、ポートレットと共に保存されている配置情報を使用して、ワークスペースでポートレットを配置する場所を決定します。このため、My webMethods Server は、ワークスペースが最後に保存されたときの場所を使用します。
[権限]	ユーザとページのポートレットとのやりとりに関する権限を設定します。権限およびその設定方法の詳細については、180 ページの「個々のリソースの権限の管理」と182 ページの「セキュリティ領域の使用」を参照してください。

[ワークスペース管理] ページから実行できるワークスペースアクション

[ナビゲーション] > [アプリケーション] > [管理] > [システム全体] > [ワークスペース管理] ページを使用して、ワークスペースに対するアクションを実行できます。デフォルトでは、My webMethods 管理者のみが [ワークスペース管理] ページにアクセスできます。My webMethods 管理者は、[ワークスペース管理] ページへのアクセス権を他のユーザに付与できます。詳細については、421 ページの「ユーザに対する [ワークスペース管理] ページへのアクセスの許可」を参照してください。

ワークスペースのキーワード検索の実行

この検索を使用して、ワークスペースの名前、説明またはキーワードに含まれるテキストを指定することにより、ワークスペースを検索します。詳細な検索条件を指定する場合は、431 ページの「ワークスペースの高度な検索の実行」を参照してください。

自分が所有者であるワークスペースまたは表示のみ以上の権限がある共有ワークスペースを検索できます。My webMethods 管理者は、どのユーザが所有するワークスペースでも検索できます。

注:

ワークスペースの共有の詳細については、435 ページの「ワークスペースの共有」を参照してください。ワークスペース権限の詳細については、*Working with My webMethods*を参照してください。

ワークスペースのキーワード検索を実行するには

1. My webMethods 管理者: [**ナビゲーション**] > [アプリケーション] > [管理] > [システム全体] > [ワークスペース管理] を選択します。

2. [**キーワード**] 検索タブがまだ表示されていない場合は、このタブをクリックします。

   ヒント:
   [**高度な検索**] タブの使用方法の詳細については、431 ページの「ワークスペースの高度な検索の実行」を参照してください。[**保存済み**] タブおよび [**オプション**] タブの使用方法の詳細については、『*Working with My webMethods*』を参照してください。

3. テキストボックスに、検索するワークスペースの名前に含まれる 1 つ以上のストリングを入力します。

   キーワードの指定方法および特殊文字の使用方法の詳細については、『*Working with My webMethods*』を参照してください。

4. [**検索**] をクリックします。

   My webMethods Server は、([検索] パネルの下の)[結果] パネルに検索結果を表示します。

ワークスペースの高度な検索の実行

詳細な条件を指定してワークスペースを検索するには、高度な検索を使用します。

自分が所有者であるワークスペースまたは表示のみ以上の権限がある共有ワークスペースを検索できます。My webMethods 管理者は、どのユーザが所有するワークスペースでも検索できます。

注:
ワークスペースの共有の詳細については、435 ページの「ワークスペースの共有」を参照してください。ワークスペース権限の詳細については、*Working with My webMethods*を参照してください。

ワークスペースの高度な検索を実行するには

1. My webMethods 管理者: [**ナビゲーション**] > [アプリケーション] > [管理] > [システム全体] > [ワークスペース管理] を選択します。

2. [**高度な検索**] タブがまだ表示されていない場合は、このタブをクリックします。

   ヒント:

[**キーワード**] タブの使用方法の詳細については、430ページの「ワークスペースのキーワード検索の実行」を参照してください。[**保存済み**] タブおよび [**オプション**] タブの使用方法の詳細については、『*Working with My webMethods*』を参照してください。

3. オプションとして、[**キーワード**] フィールドで、検索するワークスペースの名前、説明またはキーワードに含まれるテキストを入力します。

4. 詳細な条件を指定するには、使用する各条件を指定します。

   a. リストから、使用する以下の条件を選択します。

次の表に、使用可能な検索条件と、各条件の設定方法を示します。

条件	設定
[名前]	テキストボックスに、検索するワークスペースの名前に含まれるストリングを入力します。
[説明]	テキストボックスに、検索するワークスペースの説明に含まれるストリングを入力します。
[キーワード]	テキストボックスに、検索するワークスペースのキーワードに含まれるストリングを入力します。
[エイリアス]	テキストボックスに、検索に使用するワークスペースに割り当てられたエイリアスに含まれるストリングを入力します。
[タイプ]	検索するワークスペースのタイプをリストから選択します。
[所有者]	a. [**参照**] をクリックします。 b. [ユーザの選択] ウィンドウを使用して、検索するワークスペースの所有者であるユーザを検索して選択します。ユーザの検索方法の詳細については、『*Working with My webMethods*』を参照してください。 c. [**適用**] をクリックします。
[作成日時]	[**日付範囲**] リストで、指定した日付範囲内に作成されたワークスペースの検索に使用する日付範囲を選択します。
[更新日時]	[**日付範囲**] リストで、指定した日付範囲内に変更されたワークスペースの検索に使用する日付範囲を選択します。

   b. 別の条件を追加するには、[追加] を選択してこの手順を繰り返します。

5. [**検索条件**] リストで、以下を選択します。

   - My webMethods Server で、指定したすべての条件を満たすワークスペースを検索する場合は [**AND**]

- My webMethods Server で、指定した条件のいずれかを満たすワークスペースを検索する場合は [**OR**]

6. [**検索**] をクリックします。

ワークスペースを開く

ワークスペースを表示するには、新しいタブでワークスペースを開きます。

ワークスペースを開くには

1. 開くワークスペースを検索します。手順については、『430 ページの「ワークスペースのキーワード検索の実行」』を参照してください。

2. 検索結果で、開くワークスペースの行の をクリックして [新しいタブで開く] を選択します。

ナビゲーションへのワークスペースの追加

My webMethods ナビゲーションの [**ワークスペース**] セクションにはワークスペースを追加できます。ワークスペースを追加すると、ナビゲーションで名前をクリックするだけでワークスペースを開くことができます。

ワークスペースを追加する場合は、ワークスペースを表示するフォルダを選択します。デフォルトでは、ナビゲーションの [**ワークスペース**] セクションには [**最近の追加分**] および [**マイワークスペース**] の 2 つのフォルダがあります。フォルダはさらに追加することができます。ワークスペースフォルダをさらに追加する方法およびナビゲーションからワークスペースを削除する方法の詳細については、『*Working with My webMethods*』を参照してください。

ナビゲーションにワークスペースを追加するには

1. 使用するワークスペースを検索します。詳細については、430 ページの「ワークスペースのキーワード検索の実行」または431 ページの「ワークスペースの高度な検索の実行」を参照してください。

2. ナビゲーションにワークスペースを追加するには、以下のいずれかを行います。

   - 検索結果で、ナビゲーションに追加する各ワークスペースの隣のチェックボックスをオンにして、[ナビゲーションへの追加] をクリックします。

   - ナビゲーションに追加するワークスペースの行で、 をクリックして [ナビゲーションへの追加] を選択します。

3. [ナビゲーションへの追加] ウィンドウで、ワークスペースを追加するフォルダを選択します。

4. [**適用**] をクリックします。

ワークスペースの削除

不要となったワークスペースは、削除することができます。ワークスペースを削除する前にそのバックアップを保存する場合は、まずファイルにワークスペースをエクスポートします。手順については、『436 ページの「ワークスペースのエクスポート」』を参照してください。

ワークスペースを削除するには

1. 削除するワークスペースを検索します。詳細については、430 ページの「ワークスペースのキーワード検索の実行」または431 ページの「ワークスペースの高度な検索の実行」を参照してください。

2. ワークスペースを共有するユーザに削除を通知する場合は、[**ワークスペースの削除時にコラボレータに通知**] チェックボックスをオンにします。

3. ワークスペースを削除するには、以下のいずれかを行います。

   - 検索結果で、削除する各ワークスペースの隣のチェックボックスをオンにして、[**削除**] をクリックします。
   - 削除するワークスペースの行で、 をクリックして [ワークスペースの削除] を選択します。

ワークスペースの名前の変更

ワークスペースの名前を変更するには、以下の手順に従います。

ワークスペースの名前を変更するには

1. 名前を変更するワークスペースを検索します。詳細については、430 ページの「ワークスペースのキーワード検索の実行」または431 ページの「ワークスペースの高度な検索の実行」を参照してください。

2. 検索結果で、名前を変更するワークスペースの行の をクリックして [ワークスペース名の変更] を選択します。

3. [ワークスペース名の変更] ウィンドウで、[**新しい名前**] フィールドに新しい名前を入力します。

4. [**適用**] をクリックします。

ワークスペースの共有

ワークスペースは、他のユーザ、グループ、または役割と共有できます。ワークスペースの共有および共有時に割り当て可能な権限の詳細については、『*Working with My webMethods*』を参照してください。

ワークスペースを共有するには

1. 共有するワークスペースを検索します。詳細については、430 ページの「ワークスペースのキーワード検索の実行」または431 ページの「ワークスペースの高度な検索の実行」を参照してください。

2. 検索結果で、共有するワークスペースの行の をクリックして [ワークスペースの共有] を選択します。

3. [ワークスペースの共有] ウィンドウで、[**追加**] をクリックします。

4. [プリンシパルの選択] ウィンドウを使用して、ワークスペースを共有するユーザ、グループ、および役割を検索して選択します。終了したら、[**適用**] をクリックして [プリンシパルの選択] ウィンドウを閉じます。[プリンシパルの選択] ウィンドウを使用して、ユーザ、グループ、および役割を検索する方法の詳細については、『*Working with My webMethods*』を参照してください。

   選択したユーザ、グループおよび役割ごとに、My webMethods によって [ワークスペースの共有] ウィンドウに行が追加されます。

5. 選択したユーザ、グループ、および役割ごとに、そのユーザ、グループまたは役割に割り当てる権限を [**権限**] 列のリストから選択します。割り当て可能な権限の詳細については、『*Working with My webMethods*』を参照してください。

6. この変更によって影響を受けるユーザに通知する場合は、[**ワークスペースの共有の設定時または解除時にコラボレータに通知**] チェックボックスをオンにします。

7. [**適用**] をクリックします。

ワークスペースの共有の解除

ユーザ、グループまたは役割とのワークスペースの共有が不要になった場合は、以下の手順に従います。

ワークスペースの共有を解除するには

1. 共有が不要になったワークスペースを検索します。詳細については、430 ページの「ワークスペースのキーワード検索の実行」または431 ページの「ワークスペースの高度な検索の実行」を参照してください。

2. 検索結果で、共有を解除するワークスペースの行の をクリックして [ワークスペースの共有] を選択します。

3. [ワークスペースの共有] ウィンドウで、ワークスペースへのアクセス権の共有が不要になったユーザ、グループ、および役割のチェックボックスをオンにします。

4. この変更によって影響を受けるユーザに通知する場合は、[**ワークスペースの共有の設定時または解除時にコラボレータに通知**] チェックボックスをオンにします。

5. [**削除**] をクリックしてから、[**適用**] をクリックします。

ワークスペースのプロパティの設定

ワークスペースの全般プロパティを表示および設定するには、以下の手順に従います。

ワークスペースプロパティを設定するには

1. 使用するワークスペースを検索します。詳細については、430 ページの「ワークスペースのキーワード検索の実行」または431 ページの「ワークスペースの高度な検索の実行」を参照してください。

2. 検索結果で、プロパティを設定するワークスペースの行の をクリックして [プロパティ] を選択します。

3. [ワークスペースプロパティ] ウィンドウで、全般プロパティを更新します。

   全般プロパティの詳細については、『*Working with My webMethods*』を参照してください。

4. [**適用**] をクリックします。

ワークスペースのエクスポート

ワークスペースをエクスポートすると、ワークスペースのコピーをファイルに保存できます。その後、ワークスペースを同じまたは別の My webMethods Server にインポートできます。

My webMethods Server 間でワークスペースを移行する場合は、エクスポート/インポート機能を使用できます。また、ワークスペースを削除する前にエクスポートして、コピーを保存する場合もあります。将来、そのワークスペースが再度必要になった場合は、ファイルからインポートすることができます。

ワークスペースをエクスポートするには

1. エクスポートするワークスペースを検索します。詳細については、430 ページの「ワークスペースのキーワード検索の実行」または431 ページの「ワークスペースの高度な検索の実行」を参照してください。

2. 検索結果で、エクスポートする各ワークスペースの隣のチェックボックスをオンにして、[**エクスポート**] をクリックします。

3. [ワークスペースのエクスポート] ウィンドウで、エクスポートされたワークスペースを含むファイルのファイル名を指定します。

4. ワークスペースの共有方法に関する情報も含めて、後でワークスペースをインポートしたときに情報が設定されるようにする場合は、[**共有設定を含める**] チェックボックスをオンにします。

5. [**適用**] をクリックします。

6. [ファイルのダウンロード] ウィンドウで、[**保存**] をクリックします。

7. [名前を付けて保存] ウィンドウで、エクスポートされるワークスペースファイルを保存する場所を選択して、[**保存**] をクリックします。

ワークスペースのインポート

ファイルにエクスポート済みのワークスペースがある場合は、そのワークスペースを My webMethods Server にインポートできます。

My webMethods Server は、インポートを実行したユーザのパーソナルフォルダにワークスペースをインポートします。複数のユーザが同じワークスペースをインポートできます。My webMethods Server は、ワークスペースがインポートされるたびに、すべてのコンテンツを含むワークスペースのコピーを作成し、各ユーザのパーソナルフォルダに格納します。ユーザのパーソナルフォルダは [フォルダ] > [ユーザ] 内にあります。システム管理者のユーザインタフェースを使用すると、このフォルダを表示できます。

ワークスペースをインポートするには

1. My webMethods 管理者: [**ナビゲーション**] > [アプリケーション] > [管理] > [システム全体] > [ワークスペース管理] を選択します。

2. [結果] パネルで、[**インポート**] をクリックします。

3. [ワークスペースのインポート] ウィンドウで、[**参照**] をクリックします。

4. [アップロードするファイルの選択] ウィンドウで、インポートするワークスペースを含む .cdp ファイルの場所に移動してこのファイルを選択し、[**開く**] をクリックします。

5. [ワークスペースのインポート] ウィンドウで、[**適用**] をクリックします。

My webMethods ツールナビゲーションについて

ワークスペースにコンテンツを追加するには、ワークスペースにツールを追加します。My webMethods ナビゲーションの [**ツール**] タブには、追加できるツールが表示されます。ワークスペースキャンバスにツールをドラッグすると、そのツールが追加されます。ワークスペースにコンテンツを追加する方法の詳細については、『*Working with My webMethods*』を参照してください。

ワークスペースにツールを追加すると、My webMethods によって、追加したツールに固有のワークスペースに新しいポートレットが追加されます。標準では、いくつかのツールがエンドユーザに用意されています。詳細については、438 ページの「デフォルトで使用可能なワークスペースツール」を参照してください。

システム管理者は、既存のツール分類をカスタム分類で置き換えることができます。カスタム分類では、追加ツールを登録したり、一部の標準ツールを削除したりできます。詳細については、439 ページの「My webMethods ワークスペースツールのカスタマイズ」を参照してください。

デフォルトで使用可能なワークスペースツール

My webMethods には、ワークスペースに組み込むエンドユーザ向けのツールが備わっています。これらのツールは、My webMethods ナビゲーションパネルにある [**ツール**] タブの [**ワークスペースツール**] セクションで使用できます。ワークスペースツールとその動作の説明については、『*Working with My webMethods*』を参照してください。

次の表に、デフォルトで使用可能なワークスペースツールを示します。

ツール	使用目的
[添付ファイル]	ワークスペースにファイルを添付できるようにします。
[ブックマーク]	他のワークスペースおよび Web サイトにリンクを追加できるようにします。
[ディレクトリブラウザ]	他の My webMethods ユーザの識別情報を検索できるようにします。
[HTML テキスト]	ワークスペースに書式設定済みテキストを追加できるようにします。
[イメージ]	ワークスペースに画像を追加できるようにします。
[メモ]	ワークスペースに書式設定されていない単純なテキストを追加できるようにします。
[ユーザカレンダー]	ワークスペースにユーザのカレンダーを追加できるようにします。

注:
sysadmin としてログインすると、[ワークスペースツール] リストをカスタマイズすることができます。またアップグレードされた My webMethods Server のインストールを使用する場合、一部のツールが新しいバージョンで廃止または削除されていることがあります。

My webMethods ワークスペースツールのカスタマイズ

システム管理者は、My webMethods ツールナビゲーションを更新して、標準の My webMethods ツールを変更したり、カスタム分類で完全に置き換えたりすることができます。

カスタマイズでは、ページの編集時にシステム管理者のユーザインタフェースの [**ツール**] タブに表示されるデフォルトのツールを含む、既存のツールを使用できます。システム管理者の使用可能なページ編集ツールの多くは、『*My webMethods Server Portlet Reference*』で説明しています。自分で作成したカスタムツールを追加して、My webMethods Server に展開することもできます。

注:
ツールはポートレットとなります。ポートレットは Software AG Designer を使用して作成できます。

My webMethods ワークスペースツールナビゲーションをカスタマイズするには

1. システム管理者として、カスタム分類を格納する新しいツールルートフォルダを作成します。[フォルダ] > [システム] > [パレットレジストリ] に新規フォルダを追加します。

2. ツール分類にセクションが必要な場合は、カスタムツールルートフォルダ内にフォルダを作成します。各フォルダに割り当てた名前が、My webMethods ナビゲーションの [**ツール**] タブのセクション名になります。

3. 既存のツールを分類に追加するには、[Administration Dashboard] > [コンテンツ] > [パブリッシュ] を使用して、フォルダ、フォーム、リンクまたはポートレットのインスタンスをカスタムツール分類にパブリッシュします。

   たとえば、ページ編集モードで、My webMethods Server システム管理者が使用可能な既存のツールをパブリッシュできます。

4. カスタムツールを追加する場合は、カスタムツールを作成して、カスタムツールルートフォルダ内のフォルダに保存します。

   ツールは任意のポートレットに設定することができます。Software AG Designer を使用してカスタムツールを作成し、My webMethods Server に展開することができます。

5. My webMethods Server が My webMethods ナビゲーション用に使用する [左側ナビゲーション] ページのプロパティを設定して、このページを新しいカスタムツールルートフォルダに指定します。

   a. [フォルダ] > [システム] > [シェルセクション] > [Noodle Shell 左側ナビゲーション] > [左側ナビゲーション] に移動します。

   b. [左側ナビゲーション] ページの行で をクリックし、[プロパティ] を選択します。

c. [**Tools Root**] フィールドで、カスタムツールルートフォルダに割り当てた名前を入力します。

> **注:**
> [**Tools Root**] フィールドで指定する名前は、[フォルダ] > [システム] > [パレットレジストリ] にあるフォルダまたはページとする必要があります。

d. [**適用**] をクリックします。

24 スキンのカスタマイズ

- スキンとは 442
- 新規スキンの作成および変更 444
- [スキン管理] ページの使用 447
- スキンパッケージの構成 457

スキンとは

ページのルックアンドフィールはスキンを使用してカプセル化されています。特定のユーザまたはグループにスキンを関連付けることができます。この場合、そのユーザまたはグループには、このスキンを使用したページのコンテンツが表示されます。スキンは、特定のフォルダ階層と関連付けることができます。その階層内のページを表示したすべてのユーザに、そのスキンを使用したページが表示されます。

スキンをカスタマイズすると、次のように外観を変更できます。

- 自社のロゴを付加してページをブランド化する
- 自社の企業カラーに合うように色体系を変更する
- 社内で使用しているルックアンドフィールと同様のルックアンドフィールを採用する

システム管理者は、My webMethods Server の [スキン管理] ページを使用して、スキンプロパティ (イメージ、色およびフォント) の簡単なカスタマイズを行うことができます。より高度なカスタマイズを行うには、スキンが定義されているコンポーネントを直接編集する必要があります。

My webMethods Server には、特定のユーザ、グループ、またはリソースに対して表示されるスキンを決定するパーソナライゼーションルールを設定するためのさまざまな方法が用意されています。368 ページの「スキンルールの管理」 を参照してください。

スキンの継承方法

スキンは、親スキンからプロパティを継承することができます。カスタムスキンを作成する場合は、最初に既存のスキンのコピーを作成します。継承では、カスタムスキンとその親スキンとの違いを指定するだけです。

スキンの継承は、次のルールに従います。

- スキンは親スキンを 1 つのみ持つことができますが、その親スキンもまた親を持つことができます。このように、スキンは、プロパティの継承元となる祖先のリストを持つことができます。
- 親スキンは、そのプロパティを継承する子スキンをいくつでも持つことができます。
- スキンごとに skin.properties.xml ファイルを持ち、このファイルにスキンが親スキンから継承しないプロパティをリストします。さらに、このスキンに子スキンがある場合、子スキンはこれらの変更されたプロパティを継承します。
- skin.properties.xml ファイルには、スキンの親を識別する「parent」 プロパティを含めることもできます。
- skin.properties.xml ファイルのプロパティに加えて、子スキンは親スキンのスタイルシートおよび HTML フラグメントも継承します。

- skin.properties.xml ファイルで親スキンが識別されていない場合、スキンは skin.default エイリアスで識別される My webMethods Server デフォルトスキンのプロパティを継承します。

カスタマイズの程度の選択

スキンをカスタマイズする場合の推奨方法は、既存のスキンを継承する新しいスキンを作成してから、その新しいスキンを変更することです。My webMethods Server でインストールされているスキンは変更しないでください。新しいスキンを作成してから、サーバの外部で編集できるようにスキンをエクスポートすると、そのスキンを使用するすべてのサーバにスキンを展開できます。

次の方法の 1 つ以上を使用して、単純なカスタマイズから複雑なカスタマイズまでを実行できます。

- フォント、色およびイメージの変更

  [スキン管理] ページを使用すると、カスタムスキンを作成および削除したり、イメージ、色およびフォントの使用法を変更できます。これは、スキンをエクスポートしたり、スキンパッケージを構成するファイルを編集したりしないで使用できる唯一のカスタマイズ方法です。447 ページの「[スキン管理] ページの使用」を参照してください。

  より広範な変更を行う予定の場合も、カスタムスキンを作成し、[スキン管理] ページでフォント、色またはイメージを変更してから、さらに編集できるようにスキンをエクスポートすることができます。

- [スキン管理] ページでは管理できないスキンプロパティを変更するには、extended.css ファイル (464 ページの「カスケーディングスタイルシートの定義」) を使用します。

  スキンパッケージ内の既存のプロパティを少し変更する場合には、このカスタマイズ方法が適切です。extended.css ファイルを使用するには、スキンの skin.properties ファイル内で定義する必要があります。

- 既存プロパティの値は変更するが、カスタムスキンに新規プロパティを追加しないという場合は、skin.properties.xml ファイル (458 ページの「スキンプロパティファイル」) を使用します。

  たとえば、既に親スキンに定義されているフォントまたは色の値を変更できます。この場合、親スキンのプロパティは skin.properties.xml ファイルで上書きされますが、親スキンの CSS は引き続き使用されます。

- 新規 CSS クラスをカスタムスキンに追加するには、extended.csi ファイル (動的 CSS ファイル) (464 ページの「カスケーディングスタイルシートの定義」) を使用します。extended.csi ファイルを使用するには、スキンの skin.properties ファイル内で定義する必要があります。

  新規スタイルをカスタムスキンに追加する場合は、extended.csi ファイルを使用して新規クラスを作成するのが最も簡単な方法となります。

CSS および動的 CSS ファイルの定義については、次を参照してください。 463 ページの「カスケーディングスタイルシート」

既存のスキンから新規スキンを作成するには、444 ページの「新規スキンの作成および変更」を参照してください。

編集できるようにコンピュータにスキンをエクスポートするには、445 ページの「コンピュータへのスキンのエクスポート」を参照してください。

変更箇所を識別する方法

カスタムスキンパッケージに対する変更が、[スキン管理] ページで実行できる変更 (フォント、色およびイメージ) よりも広範な場合、スキンパッケージに存在する何百というプロパティの中から変更するプロパティまたは CSS クラスを識別する必要があります。これを行う方法の 1 つに、Mozilla Firefox 用の Firebug アドオンのようなブラウザ開発者ツールを使用する方法があります。

ブラウザで My webMethods Server を開いて、確認対象のスキンパッケージが使用されているページに移動します。開発者ツールを使用すると、ユーザインタフェースを調べて、表示に使用されている CSS クラスを特定することができます。これを特定したうえで、カスタムスキンで extended.css または extended.csi ファイルを使用することによって、親スキンの値を上書きします。

新規スキンの作成および変更

システム管理者として、[スキン管理] ページを使用してスキンを作成できます。新規スキンの作成後、[スキン管理] ページを使用するか、またはスキンパッケージを手動で編集して、新規スキンをカスタマイズすることができます。

新規スキンを作成するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザインタフェース] > [スキン管理] > [新規スキンの作成]

2. [**システム名**] フィールドに、文字、数字、および下線のみを使用して、短い名前を入力します。

   この名前はサーバで内部的に使用されます。

3. [**表示名**] フィールドに、ユーザに対して表示するスキンタイトルを入力します。

   この名前については文字の制限はありません。

4. [**親スキン**] リストからスキンを選択します。新規スキンの未指定のプロパティはすべてこのスキンから継承されます。

   デフォルトでは、システムのデフォルトスキンが選択されています。

5. [**作成**] をクリックします。

新規スキンのすべてのプロパティ (色、フォント、およびイメージ) は、初期状態ではこの親から継承されます。新規のヘッダーロゴを追加するなど、1 つのプロパティだけを変更したり、まったく異なる色、フォント、およびイメージを使用して斬新なルックアンドフィールを新しく作成するなどして、新規スキンを変更できます。

新規スキンを作成した後、次の 1 つ以上の作業を行って新規スキンをカスタマイズできます。

- 447 ページの「[スキン管理] ページの使用」の説明に従って、イメージ、色およびフォントを変更します。

  注:
  手動によるいずれかの編集方法を使用する前に、[スキン管理] ページで、計画されていた変更を行う必要があります。

- 445 ページの「コンピュータへのスキンのエクスポート」の説明に従って、手動で編集するためにスキンをエクスポートします。
- 446 ページの「Designer を使用したスキンパッケージの変更方法」の説明に従って、手動で編集するためにスキンを Designer にインポートします。

例:

親スキン Pearls から新規スキンを作成するとします。

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザインタフェース] > [スキン管理] > [新規スキンの作成]
2. [**システム名**] フィールドに、「`wm_skin_carbon`」のような名前を入力します。
3. [**表示名**] フィールドに、スキンタイトル「`Carbon Sailing Skin`」を入力します。
4. [**親スキン**] リストから、「Pearls」スキンを選択します。

   このスキンが、新規スキンが未指定のプロパティを継承する際の継承元になります。
5. [**作成**] をクリックします。

コンピュータへのスキンのエクスポート

[スキン管理] ページを使用してスキンパッケージを変更する場合は (447 ページの「[スキン管理] ページの使用」)、手動で編集する前に変更を行う必要があります。

システム管理者として、[スキン管理] ページを使用して、手動で編集できるようにスキンパッケージをコンピュータにエクスポートできます。スキンパッケージは、.skin ファイル拡張子の付いた圧縮ファイルです。

ヒント:
スキンパッケージを手動で編集するもう 1 つの方法は、スキンを Designer にインポートすることです。446 ページの「Designer を使用したスキンパッケージの変更方法」を参照してください。

スキンをエクスポートするには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザインタフェース] > [スキン管理] > [新規スキンの作成]

2. エクスポートするスキンのツールアイコン をクリックし、[エクスポート] を選択します。

3. [**ファイルの保存**] オプションを選択し、[**OK**] をクリックします。

   My webMethods Server ではブラウザのデフォルトのダウンロード先にファイルがダウンロードされます。

4. 手動で編集できるように、スキンパッケージを解凍します。

Designer を使用したスキンパッケージの変更方法

[スキン管理] ページを使用してスキンパッケージを変更する場合は、ファイルをエクスポートして手動で編集する前に、変更を行う必要があります。

カスタムスキンパッケージ内のファイルを変更する方法の 1 つに、スキンを Software AG Designer にインポートして [UI 開発] パースペクティブで変更を行う方法があります。スキンパッケージを直接 My webMethods Server からエクスポートし、CSS、動的 CSS または XML ファイルを編集してから、スキンパッケージを解凍しないでサーバにスキンを再度展開することができます。

注:
ここでは、Software AG Designer での作業の詳細は説明しません。Designer の MWS 管理ビューおよびサーバビューでの作業の詳細については、Composite Application Framework オンラインヘルプを参照してください。

Designer でカスタムスキンパッケージを変更するには

1. [UI 開発] パースペクティブの MWS 管理ビューで、カスタムスキンパッケージが含まれる My webMethods Server のインスタンスに接続します (データプロバイダを作成)。

   MWS 管理ビューにはフォルダおよびその他の My webMethods Server アセットのツリー表示が示されます。

2. Skins フォルダを展開し、カスタムスキンパッケージを右クリックして、[**インポート/エクスポート**] > [アセットをプロジェクトに抽出] をクリックします。

3. アセットをプロジェクトに抽出ウィザードで、[**新規**] をクリックします。

4. [**プロジェクト名**] フィールドに、MWS 管理ビューに表示されているスキンパッケージの名前を正確に入力して、[**終了**] を 2 回クリックします。

   Designer では MWS スキンプロジェクトが作成され、スキンパッケージのアセットがこのプロジェクトに抽出されます。

5. ナビゲータビューまたはプロジェクトエクスプローラビューを使用して、スキンプロジェクト内にある CSS、動的 CSS または XML ファイルを探し、テキストエディタでファイルを開きます。

6. スキンパッケージを My webMethods Server のインスタンスに展開するには、サーバビューを使用してサーバに接続してからスキンパッケージをパブリッシュします。

   スキンパッケージをテストするには、カスタムスキンを使用するようトリガーするスキンルールを作成する必要があります。368 ページの「スキンルールの管理」を参照してください。

   注:
   Designer の Preview Server を使用したスキンパッケージのテストは行わないでください。そのサーバインスタンスには、スキンを適切に表示するために必要なスキンルールがありません。

この方法を使用すると、徐々に変更を加えて、確認のために定期的にサーバに結果をパブリッシュすることができます。カスタムスキンにテストページを関連付けると、他のユーザに影響を与えずに設計作業を実行できます。

[スキン管理] ページの使用

My webMethods Server の [スキン管理] ページを使用すると、カスタムスキンを作成および削除したり、イメージ、色およびフォントの使用法を変更できます。[スキン管理] ページを使用するには、システム管理者からページへのアクセス権を付与してもらうか、システム管理者としてログインする必要があります。

[スキン管理] ページでできるカスタマイズよりもさらに広範なカスタマイズを行う場合でも、変更前にこのページでカスタムスキンを作成しておくと便利な場合があります。

My webMethods Serverイメージ、色またはフォントへの変更が、スキンパッケージに含まれる skin.properties.xml ファイルに適用されます。

スキンの削除

システム管理者として、[スキン管理] ページを使用してスキンを削除できます。

スキンを削除するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザインタフェース] > [スキン管理]

2. スキンを削除するには、ツールアイコン をクリックして、[削除] を選択します。

スキン内のイメージの置き換え

ページのスキンには、ページの外観と構造を整える、ロゴなどのイメージが含まれています。[スキン管理] ページを使用して、スキン内のイメージを、ローカルドライブ、既存のスキン、または Web サイトからのイメージで置き換えることができます。

スキン内のイメージを置き換えるには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザインタフェース] > [スキン管理]
2. スキンを変更するには、 をクリックして、[編集] を選択します。
3. [**画像**] をクリックします。
4. 置換するイメージのスキンプロパティをクリックします。

   プロパティが赤のボックスでハイライト表示されます。
5. イメージのソースに基づいて、以下のいずれかを行います。
   - ローカルドライブ - [ピッカー] パネルで [ファイルの選択] をクリックします。新規イメージを参照し、[**開く**] をクリックして、[**アップロード**] をクリックします。イメージを置換するには、 をクリックします。
   - 既存のスキンの一部 - [-- パレットを選択 --] リストで、イメージのコピー元のスキンを選択します。新しいイメージを選択して、 をクリックします。
   - Web サイト上 - [-- パレットを選択 --] リストで、[URL] を選択し、Web サイトの URL を入力して、[OK] をクリックします。新しいイメージを選択して、 をクリックします。
6. [**プレビュー**] をクリックします。

   この変更がスキンに与える影響がプレビュー表示されます。

   ヒント:
   [マイフォルダ]、[マイホームページ]、または [パブリックフォルダ] 以外のページのプレビューについては、456 ページの「サーバ上の他の場所にあるページのプレビュー」を参照してください。
7. [プレビュー] ウィンドウを閉じて、[**保存**] をクリックします。

例:

Carbon Sailing スキンがインストール済みで、webMethods ロゴをローカルファイルシステムに存在する Carbon Sailing ロゴ (carbon-logo.png) に変更するとします。

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザインタフェース] > [スキン管理]
2. 「Carbon Sailing Skin」の をクリックして、[編集] を選択します。
3. [**画像**] をクリックします。
4. [**logo.gif**] スキンプロパティをクリックします。

   プロパティが赤のボックスでハイライト表示されます。
5. [**ピッカー**] パネルで [**ファイルの選択**] をクリックして、carbon-logo.png イメージを参照し、[**開く**] をクリックしてから [**アップロード**] をクリックします。

   新規イメージが [**ピッカー**] パネルに表示されます。
6. イメージを置換するには、 をクリックします。
7. 同様の方法で、[**banner-bg.gif**] プロパティを選択し、carbon-bg.jpg ファイルでイメージを置換します。

   この結果の組み合わせでバナーが完成します。
8. [**プレビュー**] パネルで、リストから [**マイホームページ**] を選択して [**プレビュー**] をクリックします。
9. [プレビュー] ウィンドウを閉じて、[**保存**] をクリックします。

カラーピッカーを使用した色の置き換え

スキンには、ページのさまざまな部分に影響を与える、さまざまな色設定が数多く含まれています。設定を変更するには、スキンエディタで色を選択して、スキンプロパティに色を適用し、変更をプレビューします。

カラーピッカーを使用してスキンプロパティの色を置き換えるには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザインタフェース] > [スキン管理]
2. スキンを変更するには、ツールアイコン をクリックして、[編集] を選択します。
3. [**色**] をクリックします。
4. 色を置き換えるスキンプロパティをクリックします。

   スキンプロパティが赤のボックスでハイライト表示されます。
5. [**ピッカー**] パネルで、置き換えに使用する色をクリックします。

   選択した色が [**ピッカー**] パネルの下部の水平バーに表示され、上部の [#] フィールドには、16 進数の値が表示されます。

ヒント:
置き換えに使用する色の16進数の値がわかっているときには、[#] フィールドに直接値を入力できます。

6. [**スクラッチパッド**] パネルに色を保存するには、[**ピッカー**] パネルのすぐ下にある下向き矢印をクリックします。

ヒント:
前景または背景で使用するため、ある色の濃淡を変えたい場合は、[**スクラッチパッド**] パネルに色を保存し、[**ピッカー**] パネルの右端の垂直バーで、元の色の上側または下側をクリックします。複数の色を [**スクラッチパッド**] に保存します。

7. スキンプロパティの色を設定するには、[**ピッカー**] パネルまたは [**スクラッチパッド**] パネルのいずれかから ⇦ をクリックします。

8. ページの下部で、[**プレビュー**] をクリックします。

   この変更がスキンに与える影響がプレビュー表示されます。

ヒント:
[マイフォルダ]、[マイホームページ]、または [パブリックフォルダ] というラベルのフォルダに属していないページの場合は、456 ページの「サーバ上の他の場所にあるページのプレビュー」を参照してください。

9. [プレビュー] ウィンドウを閉じます。

10. [**保存**] をクリックします。

例:

444 ページの「新規スキンの作成および変更」の説明に従って Carbon Sailing スキンを作成してあり、ボタンに使用する色を変更するとします。

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザインタフェース] > [スキン管理]

2. スキンを変更するには、 をクリックして、[編集] を選択します。

3. [**色**] をクリックします。

4. [**button - button text**] スキンプロパティをクリックします。

5. [**ピッカー**] パネルの [**色**] フィールドで 「FFFFFF」と入力して、デフォルト色 (親スキンが Pearls の場合は BUTTON) を置換します。

6. [ピッカー] パネルで ⇦ をクリックします。

   [**継承**] 列のチェックボックスがオフになり、このスキンプロパティを親スキンから継承しなくなったことが示されます。

7. [**button-bg - button background**] スキンプロパティをクリックします。
8. [**ピッカー**] パネルで「9EB6C7」と入力して、デフォルト色を置換します。
9. [ピッカー] パネルで ⇦ をクリックします。
10. 同様に、残りのボタンの色のプロパティを次の値で置換します。

次の表に、各スキンプロパティに指定する値を示します。

プロパティ	値
[button-border - button border]	668899
[button-border-light - button border]	BBDDEE
[button-hover - button text]	FFFFFF
[button-hover-bg - button background]	4C7499
[button-hover-border - button border]	395874
[button-hover-border-light - button border]	ABC9D8
[button-disabled - disabled button text]	C5C7C7
[button-disabled-bg - disabled button background]	FFFFFF
[button-disabled-border - disabled button border]	C5C7C7

11. ページの下部で、[**プレビュー**] をクリックします。

次の表は、変更されたボタンのプロパティのプレビュー結果を示しています。

デフォルトのボタン:	
新しい色:	
新しいホバリングの色:	
最終的なボタン:	

最終的なボタンには452 ページの「ピッカーを使用したフォントの置換」で行うフォント変更が含まれます。

12. [**保存**] をクリックします。

スキンまたは Web サイトからの色による置き換え

既存のスキンまたは Web サイトからの色の値を使用して、スキンプロパティの色を置き換えることができます。

スキンまたは Web サイトからの色を使用してスキンプロパティの色を置き換えるには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザインタフェース] > [スキン管理]
2. スキンを変更するには、ツールアイコン をクリックして、[編集] を選択します。
3. [**色**] をクリックします。
4. 色を置き換えるスキンプロパティをクリックします。

   スキンプロパティが赤のボックスでハイライト表示されます。
5. 色のソースに基づいて、以下のいずれかを行います。
   - 既存のスキンの一部 - [-- パレットを選択 --] リストで、色のコピー元のスキンを選択します。新しいイメージを選択して、 をクリックします。
   - Web サイト上 - [-- パレットを選択 --] リストで、[URL] を選択し、Web サイトの URL を入力して、[OK] をクリックします。新しい色を選択して、 をクリックします。
6. ページの下部の [**プレビュー**] という見出しの下の [**プレビュー**] をクリックします。

   この変更がスキンに与える影響がプレビュー表示されます。

   ヒント:
   [マイフォルダ]、[マイホームページ]、または [パブリックフォルダ] というラベルのフォルダに属していないページの場合は、456 ページの「サーバ上の他の場所にあるページのプレビュー」を参照してください。
7. [プレビュー] ウィンドウを閉じます。
8. [**保存**] をクリックします。

ピッカーを使用したフォントの置換

スキンには、ページのさまざまな部分に影響を与える、さまざまなフォント設定が数多く含まれています。設定を変更するには、スキンエディタでフォント設定を選択して、スキンプロパティに設定を適用し、変更をプレビューします。

注:
使用されるフォントプロパティは、スキンごとに異なる可能性があります。以下のサンプル手順で説明するプロパティが、すべてのスキンで使用可能であるとは限りません。

ピッカーを使用してフォント設定を置換するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザインタフェース] > [スキン管理]

2. スキンを変更するには、 をクリックして、[編集] を選択します。

3. [**フォント**] をクリックします。

4. フォント設定を置換するスキンプロパティをクリックします。

   スキンプロパティが赤のボックスでハイライト表示されます。

5. [**ピッカー**] パネルで、置き換えに使用する色をクリックします。

   次の表に、フォント値に加えることができる変更を示します。

フォント設定	選択項目
Font-family	フォントのファミリー。最初に 10 個の選択肢から選択できます。この選択により、コンピュータで使用可能なフォントをブラウザで使用することができます。
Font-size	相対サイズ値のリストから選択するか、または数値を指定します。
Font-style	太字、斜体または下線を選択します。
[大文字表記]	大文字表記スタイルのリストから選択します。
Text-spacing	文字間隔、単語間隔および行間に使用する値を指定します。

6. スキンプロパティのフォント値を設定するには、[**ピッカー**] パネルまたは [**スクラッチパッド**] パネルのいずれかから ⇦ をクリックします。

7. ページの下部で、[**プレビュー**] をクリックします。

   この変更がスキンに与える影響がプレビュー表示されます。

   ヒント:
   [マイフォルダ]、[マイホームページ]、または [パブリックフォルダ] というラベルのフォルダに属していないページの場合は、456 ページの「サーバ上の他の場所にあるページのプレビュー」を参照してください。

8. [プレビュー] ウィンドウを閉じます。

9. [**保存**] をクリックします。

例:

444 ページの「新規スキンの作成および変更」の説明に従って Carbon Sailing スキンを作成してあり、ボタンに使用するフォント値を変更するとします。

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザインタフェース] > [スキン管理]
2. スキンを変更するには、 をクリックして、[編集] を選択します。
3. [**フォント**] をクリックします。
4. [**button - button text**] スキンプロパティをクリックします。
5. [**フォントファミリー**] の [**第 1 候補**] リストから [**Tahoma**] フォントを選択します。
6. [**第 1 候補**] リストから [**sans-serif**] を選択します。
7. [**フォントスタイル**] から [**太字**] を選択します。
8. [ピッカー] パネルで をクリックします。

   [**継承**] 列のチェックボックスがオフになり、このスキンプロパティを親スキンから継承しなくなったことが示されます。
9. ページの下部で、[**プレビュー**] をクリックします。

次の表は、変更されたボタンのプロパティのプレビュー結果を示しています。

デフォルトのボタン:	Search
新しいフォント値:	Search
最終的なボタン:	Search

最終的なボタンには449 ページの「カラーピッカーを使用した色の置き換え」で行う色の変更が含まれます。

10. [**保存**] をクリックします。

Web サイトからのフォントの置換

スキンで使用されるフォントファミリーを制御するには、自分自身でスタイルを設計するか、別の Web サイトで使用されているフォントファミリーのセットを選択することができます。以下の手順では、別のサイトからファミリーを取得して使用する場合のシナリオを説明します。

注:
使用されるフォントプロパティは、スキンごとに異なる可能性があります。以下のサンプル手順で説明するプロパティが、すべてのスキンで使用可能であるとは限りません。

スキンで使用するフォントファミリーを別の Web サイトで使用されているフォントファミリーで置き換えるには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザインタフェース] > [スキン管理]

2. スキンを変更するには、 をクリックして、[編集] を選択します。

3. [**フォント**] をクリックします。

4. [-- **パレットを選択** --] リストで、[**URL**] を選択し、Web サイトの URL を入力して、[**OK**] をクリックします。

   Web サイトから導出されたフォントファミリーのリストがパレットに表示されます。

5. ページの右下にあるパレットで、Webサイトの通常の本文テキストを表すと思われる行を見つけて、これを選択します。

   このアクションの結果として、[**ピッカー**] パネルに複数のフォントファミリーが表示されます。表示されない場合は、通常の本文テキストと思われる別の行をパレットから選択してください。

6. パレットのすぐ上にある をクリックします。

   このアクションにより、後で使用できるように、フォントの情報が [**スクラッチパッド**] パネルに保存されます。

7. [**スクラッチパッド**] パネルで作成した行で、編集フィールドでテキストを選択して、「**my regular text**」など、意味のある名前に変更します。

8. ページの左側の [**スキンプロパティ**] リストで、[**regular**] スキンプロパティを選択します。

   編集用に選択されたことを示して、[**regular**] スキンプロパティが赤のボックスでハイライト表示されます。

9. [**regular**] スキンプロパティを設定するには、[**スクラッチパッド**] パネルの をクリックします。

   このアクションにより、[**regular**] スキンプロパティが、[**スクラッチパッド**] パネルで選択された値に設定されます。

10. ページの左側の [**スキンプロパティ**] リストで、[**bold**] スキンプロパティを選択します。

    [**bold**] スキンプロパティが赤のボックスでハイライト表示されます。

11. [**スクラッチパッド**] パネルで、「**my regular text**」を選択します。

    「**my regular text**」スクラッチパッドアイテムは [**スクラッチパッド**] パネルで既にアクティブになっているため (濃い赤の枠が周囲に表示されています)、このアイテムを再度選択すると、これがページのアクティブアイテムになり (明るい赤の枠が周囲に表示されます)、「**my regular text**」フォントの情報が、[**ピッカー**] パネルにロードされます。

12. [**ピッカー**] パネルの [**フォントスタイル**] 領域で、[**太字**] チェックボックスをオンにします。

13. [ピッカー] パネルで をクリックします。

このアクションにより、[**bold**] スキンプロパティが、[**ピッカー**] パネルの中の値で設定されます。

14. [**スキンプロパティ**] リストで、[**small**] スキンプロパティをクリックします。

15. [**スクラッチパッド**] パネルで、「**my regular text**」を選択します。

16. [**ピッカー**] パネルの [**フォントサイズ**] 領域で、以下のいずれかを行います。

- [相対サイズ] を選択して、リスト内の値を既存のサイズよりも小さいサイズに変更します。
- [数値サイズ] を選択して、フィールド内の数値を編集し、既存のサイズの 2 分の 1 または 3 分の 1 のサイズに変更します。

17. [**ピッカー**] パネルのすぐ左にある矢印をクリックします。

18. [**スキンプロパティ**] リストで、[**medium**] スキンプロパティを選択します。

19. [**スクラッチパッド**] パネルで、「**my regular text**」を選択します。

20. ページの下部の [**プレビュー**] リストで、[**パブリックフォルダ**] を選択して、[**プレビュー**] をクリックします。

    この変更がスキンに与える影響がプレビュー表示されます。

    ヒント:
    [マイフォルダ]、[マイホームページ]、または [パブリックフォルダ] というラベルのフォルダに属していないページの場合は、456 ページの「サーバ上の他の場所にあるページのプレビュー」を参照してください。

21. [プレビュー] ウィンドウを閉じます。

22. [**保存**] をクリックします。

サーバ上の他の場所にあるページのプレビュー

[マイフォルダ]、[マイホームページ]、または [パブリックフォルダ] 以外のページをスキンエディタでプレビューできます。

サーバ上の他の場所にあるページをプレビューするには

1. [スキン管理] でスキンプロパティを変更します。

2. ページの下部の [**プレビュー**] リストで、[**URL**] を選択します。

3. 新規のブラウザウィンドウを開き、サーバに移動して、プレビューするページまで移動します。

4. ブラウザの [**アドレス**] バーで URL を選択して、Ctrl キーを押しながら C キーを押して、URL をコピーします。

5. 最初のブラウザから [スクリプト プロンプト] ダイアログに戻り、Ctrl キーを押しながら V キーを押して、URL を貼り付けます。

6. [**OK**] をクリックします。

7. [**プレビュー**] をクリックします。

スキンパッケージの構成

編集に使用可能なスキンパッケージを作成する方法は 2 つあります。

- 445 ページの「コンピュータへのスキンのエクスポート」
- 446 ページの「Designer を使用したスキンパッケージの変更方法」

編集用に公開されるスキンパッケージには、パッケージを構成する次のファイルおよびディレクトリがあります。

- skin.properties.xml ファイルでは、スキンで実装されるコンポーネントが示されます。[スキン管理] ページを使用するとフォント、色およびイメージは変更できますが、他のスキンコンポーネントについては、スキンプロパティファイルを直接編集する必要があります。詳細については、458 ページの「スキンプロパティファイル」を参照してください。

  My webMethods Server で新規スキンを作成してからまだ何も変更していない場合、ファイルに含まれているプロパティは全プロパティの継承元である親スキンのみということがあります。

- skinDeploy.xml ファイルには、スキンパッケージの展開情報が含まれます。詳細については、462 ページの「スキン展開ファイル」を参照してください。

- css ディレクトリ。カスケーディングスタイルシート (CSS) では、スキンを構成するプロパティが示されます。スキン用に生成される CSS は、他のスタイルシートファイルによる変更時に skin.properties.xml ファイルから導出されます。詳細については、463 ページの「カスケーディングスタイルシート」を参照してください。

  My webMethods Server で新規スキンを作成してからまだ何も変更していない場合、すべてのプロパティが親スキンから継承されるため、ディレクトリにカスケーディングスタイルシートは含まれていません。スタイルシートを追加または変更する場合は、このディレクトリにスタイルシートを配置する必要があります。

- components ディレクトリ。このディレクトリには、スキンパッケージの一部として変更する可能性の少ない、独自のファイルが含まれます。

- images ディレクトリ。スキンで使用されるイメージを追加または変更する場合は、このディレクトリにイメージを配置する必要があります。

スキンプロパティファイル

skin.properties.xml ファイルでは、スキンで実装されるコンポーネントが示されます。[スキン管理] ページを使用して行ったスキンの変更は (447 ページの「[スキン管理] ページの使用」)、そのスキン用の skin.properties.xml ファイルに反映されます。スキン用の base.csi ファイルは、My webMethods Server で base.css ファイルを生成するときに使用できるように、skin.properties.xml ファイル内のプロパティを参照します。

スキンプロパティファイルの重要性

スキン用の base.csi ファイルは、skin.properties.xml ファイルに定義されているプロパティを参照します。My webMethods Server は base.csi ファイルを使用して、スキンの表示方法を管理する base.css ファイルを生成します。この強力なメカニズムにより、スキンごとに新しいスタイルシートを再作成しなくても、新しい skin.properties.xml ファイルにイメージ、フォントおよび色を指定することで、共通の祖先に基づいて新規スキンを作成できます。

次の例では、Pearls スキンが親スキンになっています。カスタムスキンでは、CSS (.css または .csi) を変更する必要はありません。

イメージの置換

カスタムスキンの skin.properties.xml ファイルに含まれる `images/banner-bg.gif` プロパティの値を変更することによって、そのカスタムスキンのバナー背景イメージを変更できます。

次の表に、イメージを置換するときに変更するプロパティを示します。

ファイル	コンテンツ

親の skin.properties.xml	`<property>`
	`  <name>images/banner-bg.gif</name>`
	`  <value>images/dot.gif</value>`
	`  <description>banner background</description>`
	`</property>`

親の base.csi	`background-image: url(@skin images/banner-bg.gif;);`

子の skin.properties.xml	`<property>`
	`  <name>images/banner-bg.gif</name>`
	`  <value>images/carbon-bg.jpg</value>`
	`</property>`

461 ページの「1 つのイメージを別のイメージと置換する方法」も参照してください。

フォントのグローバルな置換

カスタムスキンの skin.properties.xml ファイルに含まれる `fonts/regular` プロパティの値を変更することによって、そのカスタムスキンで使用されるフォントをグローバルに変更できます。

次の表に、フォントを置換するときに変更するプロパティを示します。

ファイル	コンテンツ
親の skin.properties.xml	<property> <name>fonts/regular</name> <value>font-family: tahoma, sans-serif; font-size: 0.7em;</value> <description>standard text</description> </property>
親の base.csi	@skin fonts/regular;
子の skin.properties.xml	<property> <name>fonts/regular</name> <value>font-family: Georgia, 'Times New Roman', Times, serif; font-size: 1em; font-size: 0.7em; </value> <description>standard text</description> </property>

色のグローバルな置換

カスタムスキン内で使用される色をグローバルに変更できます。この例では、colors/section-body-border プロパティで行の枠線の色を制御しています。カスタムスキン用の skin.properties.xml ファイルでの 1 つの変更は、base.csi ファイル内の 7 つのプロパティに影響します。

次の表に、色を置換するときに変更するプロパティを示します。

ファイル	コンテンツ
親の skin.properties.xml	<property> <name>colors/section-body-border</name> <value>#ccc</value> <description>row border</description> </property>

親の base.csi	.tbl {
	  border: @skin sizes/section-body-border;
	  solid @skin colors/section-body-border;;
	  .
	  .
	}

子の skin.properties.xml	<property> <name>colors/section-body-border</name> <value>#dfe6ec</value> <description>row border</description> </property>

461 ページの「色の変更方法」も参照してください。

スキンプロパティファイル内のエントリの作成

skins.properties.xml ファイルを手動で編集する場合、そのスキンのいずれかの祖先から継承されるコンポーネントを含める必要はありません。必要なのは、継承されたコンポーネントに対する変更内容を含めることだけです。スキンを表すプロパティを確認する良い方法は、445 ページの「コンピュータへのスキンのエクスポート」で説明した方法を使用して親スキンパッケージをエクスポートし、スキンパッケージを解凍して、その skin.properties.xml ファイルを調べることです。

注:
skins.properties.xml ファイル内のスキンプロパティの変更には、CSS の専門知識が必要です。

skin.properties.xml ファイルはプロパティ要素で構成され、それぞれがスキンのコンポーネントを示します。プロパティ要素のフォーマットは、次のとおりです。

```
<property>
    <name>property_name</name>
    <value>property_value</value>
    <description>optional_description</description>
</property>
```

プロパティ要素には次のガイドラインが適用されます。

- 値が名前と同じである場合は、値を省略できます。
- 説明はオプションです。
- 標準の XML コマンド構文を使用して、コメントを追加できます。

```
<!-- This is a comment -->
```

スキンコンポーネントに、複数のプロパティを関連付けることができます。たとえば、My webMethods Server ページの上部に表示されるバナーには、その外観を決定する次のような 20 個以上のプロパティがあります。

背景の色、位置およびパディング

リンクテキスト、選択済みリンクの色、フォントおよびウェイト (ホバリング特性を含む)

イメージとその位置

スキンを表すプロパティを確認するには、445 ページの「コンピュータへのスキンのエクスポート」で説明した方法を使用して pearls スキンパッケージをエクスポートし、スキンパッケージを解凍して、skin.properties.xml ファイルを調べます。次の例に、プロパティを変更する方法を示します。

親スキンの指定方法

カスタムスキンを簡単に開発するには、カスタムスキンがプロパティを継承する際の継承元になる親スキンを指定する必要があります。1 つのスキンが持つことのできる親スキンは 1 つのみです。次の例では、skin.properties.xml ファイルで、カスタムスキンの親として pearls スキンパッケージを指定しています。

```
<!-- parent skin; all unspecified properties are inherited -->
<property>
    <name>parent</name>
    <value>skin.wm_skin_pearls</value>
    <description>parent skin</description>
</property>
```

1 つのイメージを別のイメージと置換する方法

1 つのイメージを別のイメージと置換するには、最初に新規イメージのコピーをスキンパッケージの images ディレクトリに移動する必要があります。その後、skins.properties.xml ファイルに、イメージのプロパティを記載します。

たとえば、バナーの左側に表示されるロゴイメージを変更して、ページをブランド再生するとします。新規イメージの名前は my_logo.png です。

```
 <!-- images -->

<property>
    <name>images/logo.gif</name>
    <value>images/my_logo.png</value>
    <description>header logo</description>
</property>
```

色の変更方法

449 ページの「カラーピッカーを使用した色の置き換え」のトピックでは、[スキン管理] ページで変更されるいくつかの色が示されています。これと同じ変更が、skin.properties.xml ファイルでは次のようになります。

```
<property>
    <name>colors/button</name>
    <value>#FFFFFF</value>
    <description>button text</description>
</property>
<property>
    <name>colors/button-bg</name>
    <value>#9EB6C7</value>
    <description>button background</description>
</property>
<property>
    <name>colors/button-border</name>
    <value>#668899</value>
    <description>button border</description>
</property>
<property>
    <name>colors/button-border-light</name>
    <value>#BBDDEE</value>
    <description>button border</description>
</property>
<property>
    <name>colors/button-hover</name>
    <value>#FFFFFF</value>
    <description>button text</description>
</property>
<property>
```

```
    <name>colors/button-hover-bg</name>
    <value>#4C7499</value>
    <description>button background</description>
</property>
<property>
    <name>colors/button-hover-border</name>
    <value>#395874</value>
    <description>button border</description>
</property>
<property>
    <name>colors/button-hover-border-light</name>
    <value>#ABC9D8</value>
    <description>button border</description>
</property>
<property>
    <name>colors/button-disabled</name>
    <value>#C5C7C7</value>
    <description>disabled button text</description>
</property>
<property>
    <name>colors/button-disabled-bg</name>
    <value>#FFFFFF</value>
    <description>disabled button background</description>
</property>
<property>
    <name>colors/button-disabled-border</name>
    <value>#C5C7C7</value>
    <description>disabled button border</description>
</property>
```

カスケーディングスタイルシートの追加方法

skin.properties.xml ファイルに複数のスタイルシートを追加できます。詳細については、466 ページの「スキンパッケージへのスタイルシートの追加」を参照してください。

スキン展開ファイル

skinDeploy.xml ファイルには、スキンパッケージの展開情報が含まれます。この展開情報全体が、root wm_xt_skin 要素の属性として格納されます。

注:
新規に作成するカスタムスキンパッケージのためにこのファイルを変更する必要はありません。

次の表に、skinDeploy.xml ファイルの wm_xt_skin 要素の属性を示します。

属性	説明
name	スキンパッケージの名前。
alias	My webMethods Server で使用されるスキンのエイリアスのカンマ区切りリスト。
description	ユーザインタフェースに表示される際のスキンの表示名。

属性	説明
version	スキンのバージョン番号。スキンパッケージを変更してリリースする場合は、マイナーバージョン番号を増やす必要があります。
cssPreview	ユーザインタフェースで html 要素のスタイルをデモ表示する、CSS スタイル宣言のセミコロン区切りリスト。
serversideResourcePath	レガシー属性。この属性の値は、常に `/ui/skins/skin-name/` の形式をとる必要があります。*skin-name* はスキンのシステム名です。
clientsideResourcePath	レガシー属性。この属性の値は、常に `/ui/skins/skin-name/` の形式をとる必要があります。*skin-name* はスキンのシステム名です。

例

このマニュアルの他の場所で例として使用されている Carbon Sailing スキンパッケージの skinDeploy.xml ファイルは、次のようになっています。

```
<wm_xt_skin name="wm_skin_carbon"
    alias="skin.wm_skin_carbon,skin.carbon"
    description="Carbon Sailing Skin"
    version="1.0"
    cssPreview="color:#fff; background-color:#6f6f60;"
    serversideResourcePath="/ui/skins/wm_skin_carbon/"
    clientsideResourcePath="/ui/skins/wm_skin_carbon/">
</wm_xt_skin>
```

カスケーディングスタイルシート

注:
ここで説明するファイルの変更には、CSS の専門知識が必要です。

My webMethods Server でページが表示されると、そのページのスキンは、スキンを構成するすべてのプロパティが記載されているカスケーディングスタイルシート (CSS) を使用します。スキンパッケージをサーバに展開したとき、およびサーバが起動するたびに、My webMethods Server によってこの基本の CSS が生成されます。生成された CSS ファイルは、次のような複数のソースから提供されるプロパティに基づいています。

- 親スキンの階層から継承されたプロパティ
- スキンパッケージに含まれる skins.properties.xml ファイル
- スキンパッケージ内の 1 つ以上の CSS または動的 CSS ファイル

My webMethods Server は動的 CSS ファイルを使用して、skin.properties.xml ファイル内のプロパティ値を参照し、スキンパッケージ用に生成される base.css ファイルにその値を配置します。465 ページの「動的 CSS ファイルの動作」を参照してください。

カスケーディングスタイルシートの定義

base.css
スキンパッケージのプライマリ CSS。base.css ファイルは、ほとんどの場合、生成された CSS ファイルとしてしか存在せず、スキンパッケージまたはその親に含まれる他のソースによって定義されます。base.css は、1 つ以上の拡張 CSS または動的 CSS ファイルで変更できます。カスタムスキンパッケージに base.css ファイルを作成することはお勧めしていませんが、これを行う場合は、skin.properties.xml ファイルでこのファイルを定義します。base.css ファイルと base.csi ファイルの両方を同じカスタムスキンパッケージで定義することはできません。

extended.css
スキンパッケージのセカンダリ CSS。拡張 CSS ファイルのエントリは、base.css ファイルのエントリよりも優先されます。複数の拡張 CSS ファイルを用意して、ブラウザ固有の CSS 値を提供するといったことができます。拡張 CSS ファイルにそれぞれ異なる名前を付けて、skin.properties.xml ファイルで定義します。

base.csi
skin.properties.xml ファイル (465 ページの「動的 CSS ファイルの動作」) 内のプロパティを参照する動的 CSS ファイル。base.csi ファイルの結果として、生成された base.css ファイルには skin.properties.xml ファイルからのプロパティが含まれます。トップレベルのスキンパッケージ (親がないスキンパッケージ) の skin.properties.xml ファイルには、スキンを構成するすべてのプロパティが記載されます。スキンパッケージ内に base.csi ファイルがない場合、親スキンから継承されます。base.csi ファイルは、skin.properties.xml ファイルで定義します。base.css ファイルと base.csi ファイルの両方を同じカスタムスキンパッケージで定義することはできません。

extended.csi
スキンパッケージのセカンダリの動的 CSS ファイル (465 ページの「動的 CSS ファイルの動作」)。extended.csi ファイルのエントリは、base.csi ファイルのエントリよりも優先されます。複数の extended.csi ファイルを用意して、ブラウザ固有の CSS 値を提供するといったことができます。extended.csi ファイルにそれぞれ異なる名前を付けて、skin.properties.xml ファイルで定義します。

変更するスタイルシートのタイプを選択する方法

使用するスタイルシートを決定する際には、次のガイドラインを使用します。

- 基本となる親スキンとあまり違わないカスタムスキンに選択的な変更を加えるには、extended.css ファイルを使用します。extended.css ファイルは、次のファイルよりも優先されます。
  - 親スキンパッケージ内の base.css または base.csi ファイル

  カスタムスキン用の extended.csi ファイルを定義する場合、そこに含まれるクラスは、親スキンの base.css または base.csi ファイル内の対応するクラスのみを上書きします。
- カスタムスキンパッケージまたはその親の skin.properties.xml ファイルに対する選択的な参照を変更するには、extended.csi ファイルを使用します。extended.csi ファイルは、次のファイルよりも優先されます。
  - 親スキンパッケージ内の base .csi ファイル

- カスタムスキンパッケージ内の base.csi ファイル (作成してある場合)

  新規スタイルをカスタムスキンに追加する場合は、extended.csi ファイルを使用して新規クラスを作成するのが最も簡単な方法となります。

  extended.css ファイルは、skin.properties.xml ファイルで定義する代わりに、インポートステートメントを使用して extended.csi ファイルに含めることができます。たとえば、以下のステートメントでは、スキンの base.css の生成時に、2 つの CSS ファイルが含められます。

```
@import url(@skin css/general.css;);"
@import url(@skin css/dialog_styles.css;);
```

- カスタムスキンパッケージ内の skin.properties ファイルに対して、参照一式の変更を必要とする広範な変更を行う場合は、base.csi ファイルを使用します。親スキンパッケージがないスキンパッケージには base.csi ファイルが必要です。

  カスタムスキン用の base.csi ファイルを定義すると、親スキン内の base.css または base.csi ファイルは完全に上書きされます。カスタムの base.csi ファイルに親スキン用に定義したクラスのサブセットが含まれる場合、生成される base.css ファイルにはそのクラスのサブセットのみが含まれます。

動的 CSS ファイルの動作

My webMethods Server は動的 CSS ファイル内のスキンプロパティ参照を skin.properties.xml ファイル内のプロパティ値で置換して、スキンパッケージ用に生成される base.css ファイルにその値を含めます。トップレベルのスキンパッケージ (親がないスキンパッケージ) には、通常、base.css ファイルはなく、代わりに base.csi ファイルを使用して、生成される base.css ファイルを作成します。

カスタムスキンパッケージで使用すると、この機能により、新しい skin.properties.xml ファイルにスキンプロパティを指定するだけで、共通の祖先に基づく新規スキンを作成できるようになります。skin.properties.xml ファイルに対して、動的 CSS ファイルの変更を必要としない単純な変更を行う場合は、親スキンから継承される base.csi ファイルを使用できます。さらに広範な変更を行う場合は、extended.csi ファイルを作成して、変更または追加されたプロパティを適切に参照することが必要になる場合があります。

注:
ここで説明するファイルの変更には、CSS の専門知識が必要です。

動的 CSS ファイルは、アットルール (@規則) を使用して、ローカルの skin.properties.xml ファイル内のプロパティまたは親スキンから継承されるプロパティを参照します。たとえば、base.csi ファイルに次のルールが含まれるとします。

```
body {
      @skin fonts/regular;
      color: @skin colors/text;;
      background-color: @skin colors/text-bg;;
      direction: @skin dir/text;;}
```

さらに、skin.properties.xml ファイルに 次のプロパティ定義が含まれるとします。

```
<property>
  <name>fonts/regular</name>
```

```
  <value>font-family: trebuchet; font-size: 20px;</value>
</property>
<property>
  <name>colors/text</name>
  <value>#000</value>
</property>
<property>
  <name>colors/text-bg</name>
  <value>#fff</value>
</property>
<property>
  <name>dir/text</name>
  <value>ltr</value>
</property>
```

生成される base.css ファイルには、次のルール が含まれます。

```
body {
      font-family: trebuchet; font-size: 20px;
      color: #000;
      background-color: #fff;
      direction: ltr;
}
```

スキンパッケージで使用するための base.csi または extended.csi ファイルを作成する場合、466 ページの「スキンパッケージへのスタイルシートの追加」で説明しているように、skin.properties.xml ファイルでこれらのファイルを宣言する必要があります。

スキンパッケージへのスタイルシートの追加

CSS および動的 CSS ファイルを使用して、スキンパッケージ用に生成される base.css ファイルを変更できます。CSS または動的 CSS ファイルをスキンパッケージに追加するには、カスタムスキンパッケージ用の skin.properties.xml ファイルでこれらのファイルを定義します。個々のスタイルシートは、次の形式を持つプロパティ要素で定義します。

```
<property>
   <name>css/name.css</name>
   <value>css/name.extension</value>
   <description>optional description</description>
</property>
```

<value> 要素のファイル名が <name> 要素のファイル名と同一の場合は、<value> 要素を省略できます。

skin.properties.xml ファイルにスタイルシートのプロパティ要素が含まれていない場合、親スキンのスタイルシートが継承されます。

注:
スタイルシートを追加または変更した場合は、スキンパッケージの css ディレクトリにスタイルシートを配置する必要があります。

例: base.css

注:

カスタムスキンパッケージで使用するための base.css ファイルを作成することはお勧めしません。ただし、後に続く例と比較するために、ここの構文を使用することができます。

skin.properties.xml ファイルに記載できる base.css 要素は 1 つのみです。次の例では、<value> 要素が <name> 要素と同一のため、<value> 要素を省略することができます。

```
<property>
   <name>css/base.css</name>
   <value>css/base.css</value>
</property>
```

例: base.csi

親スキンの base.csi ファイルを上書きする広範な変更を行う場合に、新しい base.csi ファイルを作成して、カスタムスキンパッケージ内でそのファイルを定義することがあります。この場合、次に示すように、<name> 要素は `base.css` で、<value> 要素は `base.csi` になります。

```
<property>
   <name>css/base.css</name>
   <value>css/base.csi</value>
</property>
```

例: extended.css または extended.csi

extended.css および extended.csi ファイルのルールは、base.css のルールと同じです。

```
<property>
   <name>css/extended.css</name>
   <value>css/extended.css</value>
</property>
```

または

```
<property>
   <name>css/extended.css</name>
   <value>css/extended.csi</value>
</property>
```

例: 複数の extended.css または extended.csi ファイル

skin.properties.file には、同時に複数の extended.css または extended.csi ファイルのプロパティを含めることができます。この機能の用途の 1 つとして、ブラウザ固有のプロパティを提供するという使い方があります。

```
<property>
   <name>css/extended.css</name>
   <value>css/extended.csi</value>
</property>

<property>
   <name>css/ie6.css</name>
   <value>css/ie6.csi</value>
   <description>special ie6 stylesheet rules</description>
</property>
```

```
<property>
   <name>css/ie7.css</name>
   <value>css/ie7.csi</value>
   <description>special ie7 stylesheet rules</description>
</property>
```

25 My webMethods Server でのシェルの使用

- シェルとは 470
- 新規シェルの作成 470
- シェルの変更 471
- HTML <head> 要素への追加タグの挿入 472
- シェルセクションでのエイリアスの使用 472
- シェルの削除 473
- 空のシェルセクションの作成 473

シェルとは

My webMethods Server では、ページのヘッダーおよびフッターのコンテンツとレイアウトは、現在のシェルの [ヘッダー] および [フッター] フォルダのコンテンツとレイアウトから導出されます。ページがレンダリングされると、実際には、最大 5 つのフォルダ (シェルのヘッダー、シェルの左側ナビゲーション、要求されたページ、シェルの右側ナビゲーション、シェルのフッター) のコンテンツが同時に表示されます。要求されたページのコンテンツは個別のポートレットとして表示されますが、シェルセクションのコンテンツは、タイトルバー、枠、または追加のスペースなしでレンダリングされます。シェルセクションを構成する行および列は、CSS クラスおよび CSS スタイルを適用することも、その正確な寸法を指定してシェルのレイアウトをさらにカスタマイズすることもできます。

[タイトルバー] シェルセクションは、ページ上の各ポートレットにタイトルバーを適用します。タイトルバーには表示名およびポートレットを制御するボタンが含まれます。[タイトルバー] シェルセクションは、個々のポートレット単位で非表示にできます。

[シェル管理] ページを使用して、シェルを作成および変更できます。[シェル管理] ページを使用するには、システム管理者からポートレットへのアクセス権を付与してもらうか、システム管理者としてログインする必要があります。

新規シェルの作成

新規シェルを構築する最初のステップでは、既存のシェルを親として使用して、シェルを作成します。新規シェルのプロパティは、初期状態ではすべて親シェルから継承されます。これらのプロパティ (シェルセクション) には、ヘッダー、フッター、左側ナビゲーション、右側ナビゲーション、およびタイトルバーがあります。これらのセクションを、新規のカスタムシェルセクションで置き換えることができます。

新規シェルを作成するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザインタフェース] > [シェル管理] > [新規シェルの作成]

2. [**名前**] フィールドに、シェルの名前を入力します。

   この名前については文字の制限はありません。

3. [**説明**] フィールドに、シェルの説明を入力します (オプション)。

   この説明は、[シェル管理] ページのシェルのリストに表示されます。

4. [**親シェル**] リストからシェルを選択します。新規シェルの未指定のプロパティはすべてこのシェルから継承されます。

   デフォルトでは、システムのデフォルトシェルが選択されています。

5. [**作成**] をクリックします。

シェルの変更

親シェルから新規シェルを作成したら、そのシェルのセクションを個々に変更して、新規シェルを構築できます。

シェルを変更するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザインタフェース] > [シェル管理]

2. シェルを変更するには、 をクリックして、[プロパティ] を選択します。

3. シェルの表示名を変更する場合は、[**表示名**] フィールドに、シェルの新しい名前を入力します。

4. 各種のシェルセクションを継承するための親シェルを変更する場合は、[**親**] リストで親シェルを選択します。このリストには、現在サーバ上に存在するシェルが含まれています。

5. シェルセクションごとに、次の親またはソースのいずれかを指定します。

   - [継承] - [親シェル] フィールドで選択したシェルのシェルセクションを使用します。
   - [ポータルページ] - 既存のフォルダのコンテンツをシェルセクションで使用するには、[選択された項目] ボックスにフォルダを移動し、[選択] をクリックします。[タイトルバー] シェルセクションでは使用できません。
   - [ポートレット] - 既存のポートレットを使用するには、[選択された項目] ボックスにポートレットを移動し、[選択] をクリックします。[タイトルバー] シェルセクションでのみ使用可能です。

6. 編集するシェルセクションについて、次のいずれかを実行します。

   - 別のシェルから継承したシェルセクションを編集するには、[継承] を選択して [親からクローンを作成] をクリックします。継承されたシェルセクションに基づいて My webMethods Server がフォルダを作成します。
   - シェルセクションとして使用されている既存のフォルダを編集するには、[**ポータルページ**] オプションを選択して、ターゲットフォルダの名前が表示されていることを確認します。

   注:
   サーバ内でポートレットを編集することはできません。Software AG Designer を使用する必要があります。

7. [**編集**] をクリックします。

   シェルセクションを表すフォルダが編集モードで開きます。

8. 他のフォルダの場合と同様の手順で、シェルセクションを変更します。

9. [**保存**] をクリックします。

HTML <head> 要素への追加タグの挿入

シェルの <head> 要素に JavaScript ライブラリとスタイルシートを挿入できます。有効な HTML タグは <link>、<meta>、<script> および <style> です。

HTML <head> 要素に挿入するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザインタフェース] > [シェル管理]

2. シェルを変更するには、 をクリックして、[プロパティ] を選択します。

3. [**HTML の拡張タグ**] フィールドで、シェルにカスタムコードを追加する HTML タグを入力します。次に例を示します。

```
<link href="default.css" rel="stylesheet" type="text/css"
```

4. [**保存**] をクリックします。

シェルセクションでのエイリアスの使用

フォルダにエイリアスがある場合は、エイリアスを使用して、シェルセクションとして使用するフォルダを選択できます。フォルダのエイリアスの使用方法については、330 ページの「エイリアスの管理」を参照してください。

エイリアスを使用してシェルセクションを選択するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザインタフェース] > [シェル管理]

2. シェルを変更するには、 をクリックして、[編集] を選択します。

3. エイリアスを関連付けるシェルセクションで、[**エイリアスを使用**] をクリックします。

4. サーバリソースセレクタの [**エイリアス名**] フィールドで、このシェルセクションに使用するフォルダのエイリアスを入力します。

5. サーバがエイリアスを検索できることを確認するため、[**テスト**] をクリックします。

6. サーバがエイリアスを正しく解決した場合は、[**選択**] をクリックします。

7. 必要であれば、471 ページの「シェルの変更」の説明に従って、このフォルダのクローンを作成するか、フォルダを直接編集できます。

8. [**保存**] をクリックします。

シェルの削除

不要になったシェルは削除できます。

シェルを削除するには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザインタフェース] > [シェル管理]

2. シェルを削除するには、 をクリックして、[削除] を選択します。

空のシェルセクションの作成

シェルには、シェルセクションを構成する 4 つのフォルダが必ず含まれています。ただし、1 つ以上のシェルセクションを空にした (表示スペースを取らない) シェルの設計が必要になることも考えられます。ポートレットを含まず、それに関連するフォーマット情報を持たないシェルセクションは空です。My webMethods Server が提供するデフォルトのシェルでは、[左側ナビゲーション] および [右側ナビゲーション] シェルセクションは、空として表示されます。

[タイトルバー] シェルセクションを、他のシェルセクションのように編集することはできません。タイトルバーを非表示にするには、個々のポートレットの [**タイトルバー**] 属性を [**いいえ**] に設定する必要があります。

シェルセクション (タイトルバー以外) を空にするには

1. システム管理者として次のように選択します。[**Administration Dashboard**] > [ユーザインタフェース] > [シェル管理]

2. シェルを変更するには、 をクリックして、[編集] を選択します。

3. リソースセレクタの [**エイリアス名**] フィールドに、エイリアス「`shell.section.blank`」を入力します。

4. サーバがエイリアスを検索できることを確認するため、[**テスト**] をクリックします。

5. サーバがエイリアスを正しく解決した場合は、[**選択**] をクリックします。

6. [**保存**] をクリックします。

V Command Central を使用した My webMethods Server の管理

26 My webMethods Server 管理者ガイド 477

27 コマンドラインを使用する My webMethods Server の管理 493

28 My webMethods Server の認証 507

26 My webMethods Server 管理者ガイド

- My webMethods Server インスタンスの管理 478
- My webMethods Server ポートの設定 481
- ディレクトリサービスの設定 482
- My webMethods Server 電子メールの設定 489
- My webMethods Server 環境変数の設定 490
- My webMethods Server インスタンスの KPI の監視 491
- 信用できる認証を使用した My webMethods Server への接続 492

My webMethods Server インスタンスの管理

My webMethods Server インスタンスの作成

My webMethods Server Web ユーザインタフェースを使用して Command Central インスタンスを作成するには

1. Command Central ホーム画面で環境を選択して、[**Installations**] タブをクリックします。
2. インストールの名前をクリックして、インストールの [**Instances**] タブに移動します。
3. [Instances] タブで [Create instance] をクリックして、My webMethods Server を選択します。

   Command Central で [Create instance] ウィザードが表示されます。
4. [Create instance] ウィザードの [Specify Properties] 画面で、次の内容を指定します。
   - [**Instance Name**] - このフィールドは必須です。新規インスタンスに、マシン上のサーバインスタンスの中で一意となる名前を指定します。
   - [**Register Windows service/UNIX daemon for automatic startup**] - チェックされている場合、My webMethods Server は Windows マシンではサービスとして、UNIX マシンではデーモンとしてインストールされます。このチェックボックスは、デフォルトでオフになっています。
5. [**Database**] タブで、次を指定します。

次の表に、新しい My webMethods Server インスタンスを作成するときに指定するデータベース設定を示します。

フィールド	説明
Database type	My webMethods Server データベースのタイプ。デフォルト値は Oracle です。

JDBC URL	データベース接続の URL は次のとおりです。
	`jdbc:wm:database://hostname:port;`
	`databaseName=db_name;`
	`option=value;option=value`
	database は My webMethods Server データベースのタイプ、*hostname* はデータベースがインストールされているサーバのアドレス、*port* はデータベースの接続ポート、*db_name* は My webMethods Server が使用しているデータベースの名前です。*option=value* ペアで追加オプションを指定します (セミコロンで

フィールド	説明
	区切る)。有効なデータベースのタイプは、`oracle`、`db2`、`sqlserver`、`mysql`、`postgresql` です。 例: `jdbc:wm:sqlserver://localhost:1433;databaseName=WMDB`
Database user	データベースに接続するユーザの名前。
Password	データベース接続用のパスワード。
JDBC driver class	JDBC ドライバのドライバクラス名。Software AG が提供するものとは異なる JDBC ドライバを使用する場合は、このフィールドは必須です。

6. [**Ports**] タブで、次を指定します。

次の表に、新しい My webMethods Server インスタンスを作成するときに指定するポート設定を示します。

フィールド	説明
HTTP port	オプション。新規サーバインスタンスが受信待機する HTTP ポート。デフォルトのポート番号は `8585` です。
HTTPS port	オプション。新規サーバインスタンスが受信待機する HTTPS ポート。デフォルトのポート番号は `8586` です。
Debug port	オプション。新規サーバインスタンスの Java デバッグポート。デフォルトのポート番号は `10033` です。
JMX port	オプション。新規サーバインスタンスの JMX ポート。デフォルトのポート番号は `5002` です。

7. [**Create instance**] ウィザードの [**Cluster**] タブで、My webMethods Server クラスタ内のノードとして新規インスタンスを作成する場合、次の項目を指定します。
   - [**Cluster node name**] - クラスタノードの名前。クラスタに一意の名前を指定します。名前に空白を入れないでください。
   - [**JNDI provider URL**] - JMS プロバイダとして使用する Universal Messaging サーバの URL。
8. [**Next**] をクリックして設定の概要を確認し、 [**Finish**] をクリックして新規 My webMethods Server インスタンスを作成します。

My webMethods Server インスタンスの更新

サーバインスタンスの更新は、新しいコンポーネントをインストールした後、またはインスタンスの設定を変更した後に行います。例えば、インスタンスを My webMethods Server クラスタに追加したときなどです。

My webMethods Server Web ユーザインタフェースを使用して Command Central インスタンスを更新するには

1. インストールの [**Instances**] タブでインスタンスアイコンをクリックして、更新するインスタンスを選択します。

2. ギヤボタンのドロップダウンメニューから [**Update instance**] を選択します。

3. 必要に応じてインスタンスの設定を更新し、[**Next**] をクリックします。以下のフィールドを設定できます。

次の表に、新しい My webMethods Server インスタンスを更新するときに指定する設定を示します。

フィールド	説明
ノード名	オプション。My webMethods Server クラスタ内でインスタンスをホストするサーバノードの名前。既存のサーバノードの名前を変更するには、このフィールドを使用します。ノード名には空白を含めないでください。
JDBC URL	オプション。My webMethods Server データベースの接続 URL。
Database user	オプション。My webMethods Server データベースに接続するユーザの名前。
Password	オプション。My webMethods Server データベース接続用のパスワード。

4. インスタンスの設定を確認し、[**Finish**] をクリックします。

My webMethods Server インスタンスの削除

My webMethods Server Web ユーザインタフェースを使用して Command Central インスタンスを削除するには

1. インストールの [**Instances**] タブでインスタンスアイコンをクリックして、削除するインスタンスを選択します。

2. [**Remove**] をクリックして、インスタンスを削除します。

3. [**OK**] をクリックして、削除を確認します。

My webMethods Server インスタンス動作の一時停止と再開

Command Central Web ユーザインタフェースおよびコマンドラインインタフェースを使用して My webMethods Server インスタンスの動作を一時停止および再開することができます。Command Central から My webMethods Server インタフェースを一時停止すると、インスタンスは保守モードになります。インスタンスが保守モードのときは、My webMethods ログインページには通知が表示され、Administrator または sysadmin ユーザだけがログインできます。

コマンドラインインタフェースを使用して My webMethods Server インスタンスを一時停止または再開するための詳細な情報は、498 ページの「My webMethods Server-ENGINE のライフサイクルアクション」 を参照してください。

Command Central Web ユーザインタフェースを使用して My webMethods Server インスタンスを一時停止および再開するには

1. インストール製品の [**Instances**] タブで、一時停止または再開するインスタンスの状態アイコンをクリックします。

2. [**Lifecycle Actions**] ドロップダウンメニューで指定するアクションを選びます。

3. [**OK**] をクリックして、削除を確認します。

My webMethods Server ポートの設定

Command Central での My webMethods Server ポートの設定

My webMethods Server は、1 つ以上のポートでクライアントの要求を受信待機します。ポートがメッセージまたは要求を受信すると、My webMethods Server は適切なサービスを呼び出します。各ポートは、特定のプロトコルで動作するように設定されています。必要に応じて、1 つ以上の追加のポートで HTTP または HTTPS を関連付けることができます。デフォルトでは、My webMethods Server は HTTP が 8585 で事前設定されています。

MWS_default コンポーネントは、OSGi プロファイルです。My webMethods Server コンポーネントは、サーバインスタンスの標準的なプロファイルです。My webMethods Server コンポーネントの設定は編集できますが、追加や削除はできません。

Command Central Web ユーザインタフェースを使用して、次の手順で My webMethods Server ポートを設定します。

My webMethods Server ポートを設定するには

1. [Environments] パネルで、My webMethods Server インスタンスを表示する環境を選択します。

2. [**Instances**] タブを選択します。

3. 設定する My webMethods Server インスタンスを含む MWS_*instanceName* ノードを展開します。

4. 名前の列で **My webMethods Server** をクリックします。

5. [**Configuration**] タブを選択します。**My webMethods Server** が左側のペインで選択されていることを確認します。

6. ドロップダウンリストから [**Ports**] を選択します。

7. ポートを [**Test**] して、[**Save**] します。

ポート設定の編集

Command Central Web ユーザインタフェースを使用して、次の手順でサーバポート設定を変更します。

ポートを有効または無効にするには

1. [Environments] パネルから My webMethods Server 環境を設定して、 [**Instances**] タブから編集する My webMethods Server インスタンスをクリックします。

2. [**Configuration**] タブをクリックします。

3. ドロップダウンリストで [**Ports**] を選択します。

4. 編集するポートの数をクリックして、[**Edit**] をクリックします。ポートの設定が編集可能になります。ポートの設定に必要な変更を加えます。

5. 変更を [**Test**] して、[**Save**] します。

ディレクトリサービスの設定

My webMethods Server には、デフォルトのシステムユーザを使用する内部ディレクトリが含まれます。LDAP などの外部ディレクトリを使用している場合、そのディレクトリから外部ディレクトリサービスを介してユーザやグループ情報にアクセスできるように My webMethods Server を設定することができます。Command Central Web ユーザインタフェースを使用して、ディレクトリサービスを作成し、My webMethods Server を外部の LDAP、ADSI または ADAM サーバに接続できます。

外部ディレクトリサービスを設定するには

1. [Environments] パネルで、設定する My webMethods Server インスタンスの環境を選択します。
2. [Instances] タブを選択します。
3. My webMethods Server インスタンスの MWS_*instanceName* ノードを展開します。
4. **My webMethods Server** をクリックします。
5. [**Configuration**] タブを選択します。**My webMethods Server** が左側のペインで選択されていることを確認します。
6. ドロップダウンリストから[LDAP] を選択します。
7. をクリックして、次のオプションからディレクトリサービスのタイプを選択して追加します。
   - **LDAP** - Lightweight Directory Access Protocol。
   - **ADSI** - Active Directory Service Interfaces。
   - **ADAM** - Active Directory Application Mode。
8. ディレクトリサービスを設定するフォームを入力します。

   使用可能な設定の詳細については、483 ページの「LDAP、ADSI、ADAM ディレクトリサービスのプロパティ」を参照してください。
9. ディレクトリサービスの設定をテストして、保存します。

LDAP、ADSI、ADAM ディレクトリサービスのプロパティ

LDAP、ADSI、または ADAM タイプの外部ディレクトリサービスを設定する場合、以下のプロパティを設定できます。

[全般] セクション

次の表に、[全般] セクションで設定するディレクトリサービスのプロパティを示します。

プロパティ	説明
[名前]	必須。外部ディレクトリサービスを識別する名前。My webMethods Server は、この名前を使用してユーザインタフェース内で外部ディレクトリサービスを表示します。
[説明]	外部ディレクトリサービスに関する説明的なコメント。

プロパティ	説明
[キーワード]	外部ディレクトリサービスの検索に使用する 1 つ以上のキーワード。

[キャッシュ] セクション

次の表に、[キャッシュ] セクションで設定するディレクトリサービスのプロパティを示します。

プロパティ	説明
[キャッシュ容量]	必須。キャッシュに入れるデータベースクエリーの数。デフォルトは `1000` です。 キャッシュされたクエリーの数が指定された容量に達した場合、My webMethods Server はキャッシュエントリを最も古いエントリから順に削除します。
[キャッシュタイムアウト]	必須。キャッシュ容量を超えない限り、クエリーがキャッシュに残る時間。デフォルトは 1 時間です。 指定されたキャッシュ容量に達していなくても、キャッシュタイムアウトの期限が切れると、My webMethods Server はキャッシュエントリを削除します。

My webMethods Server はすべてのキャッシュをメモリに保存し、再起動されたときにすべてのキャッシュエントリを消去します。

[接続情報] セクション

次の表に、[接続情報] セクションで設定するディレクトリサービスのプロパティを示します。

プロパティ	説明
[サービスの有効化]	ディレクトリサービスを有効および無効にします。デフォルトは [はい (このサービスを有効にする)] です。
[接続エラーのしきい値]	必須。このサービスを無効にするまでに発生する最大接続エラー数です。デフォルトは `10` です。
[自動再接続]	接続エラーのしきい値に達した後にサービスが無効になった場合、またはネットワーク障害や定期メンテナンスのためにサーバへの接続が失われた場合に、ディレクトリサーバへの再接続を試行します。デフォルトでは有効になっています。
[自動再接続間隔]	次の再接続試行まで待機する期間 (秒)。デフォルトは `6` です。
[プロバイダ URL]	必須。次の構文を使用した外部ディレクトリサーバの URL。 `ldap://host_name:port_number`

プロパティ	説明
[基本 DN]	必須。ディレクトリサーバにクエリーを行う際に使用するルート識別名。たとえば、「`ou=mywebMethods,o=webmethods.com`」と入力します。
[ユーザ DN]	ユーザの検索やロードの際に使用する追加のユーザ DN。
[グループ DN]	ユーザの検索やロードの際に使用する追加のグループ DN。
[セキュリティプリンシパル]	必須。外部ディレクトリサーバへのログインに必要な識別名。
[セキュリティクレデンシャル]	必須。外部ディレクトリサーバへのログインに必要なパスワード。
[フェイルオーバー URL]	[プロバイダ URL] フィールドで指定したプライマリ LDAP サーバの障害の際に My webMethods Server がフェイルオーバーで使用する、別の LDAP サーバの URL。複数の場合はスペースで区切ります。
[検索タイムアウト]	必須。LDAP 検索クエリーが期限切れになるまでの最大時間(秒)デフォルトは `0` で、クエリーが期限切れになりません。 custom_wrapper.conf ファイルで接続タイムアウトを設定しない限り、My webMethods Server は LDAP サーバへの接続タイムアウトの定義に [検索タイムアウト] の値を使用します。LDAP サーバの接続タイムアウトの設定の詳細については、122 ページの「LDAP ディレクトリサービスの接続タイムアウトの設定」を参照してください。
[デフォルトワイルドカード検索を有効にする]	必須。ディレクトリ検索でワイルドカード文字の使用を有効または無効にします。デフォルトは [はい (デフォルトワイルドカード検索を有効化)] です。 ワイルドカード検索を無効にすると、大規模なサーバの場合にパフォーマンスが向上します。ワイルドカードを使用する場合、サーバでは内部インデックスが検索に使用されません。
[グループ全域ディレクトリサービスを有効にする]	必須。My webMethods Server で設定されたすべての外部ディレクトリサービスに対してグループメンバーシップのクエリーを行うかどうかを示します。このオプションを有効にすると、すべてのディレクトリサービスに対してグループメンバーシップのクエリーが行われるため、ログインパフォーマンスが著しく低下します。デフォルトは [いいえ、グループ全域ディレクトリサービス] です。 詳細については、123 ページの「ディレクトリサービス間のグループメンバーシップ」を参照してください。
[GroupQuickSearch を有効にする]	Active Directory で必須。再帰検索の代わりに 1 つのクエリーを使用している Active Directory ユーザのグループメンバーシップを特定するかどうかを示します。このオプションを有

プロパティ	説明
	効にすると、検索では 1 つのクエリーを使用するため、ログインパフォーマンスが向上します。ユーザは、Active Directory セキュリティグループまたは標準グループのいずれかに属している必要があります。デフォルトは [無効] です。
[Active Directory ドメイン URL]	Active Directory にのみ適用されます。複数の Active Directory サブドメイン URL をスペースで区切って指定します。

[高度なオブジェクトフィルタ] セクション

次の表に、[高度なオブジェクトフィルタ] セクションで設定するディレクトリサービスのプロパティを示します。

プロパティ	説明
[ユーザオブジェクトフィルタ]	ユーザの検索時にすべてのクエリーに対して My webMethods Server が適用する LDAP フィルタ。My webMethods Server で公開されるオブジェクトのタイプを制限する技術 LDAP クエリーを使用します。 **注:** My webMethods Server ディレクトリのデバッグログを調べて、クエリーが正しく動作していることを確認することをお勧めします。
[グループオブジェクトフィルタ]	グループの検索時にすべてのクエリーに対して My webMethods Server が適用する LDAP フィルタ。My webMethods Server で公開されるオブジェクトのタイプを制限する技術 LDAP クエリーを使用します。 **注:** My webMethods Server ディレクトリのデバッグログを調べて、クエリーが正しく動作していることを確認してください。
[ネストされたグループを使用する]	ネストされた LDAP グループ内での検索を有効または無効にします。デフォルト値は [いいえ (ネストされたグループを使用しない)] です。
[仮想リスト表示制御 (VLV) を使用する]	LDAP クエリーのオブジェクトのサブセットを取得するために、仮想リスト表示制御 (VLV) を使用するかどうかを指定します。デフォルト値は [いいえ (VLV 制御を使用しない)] です。LDAP サーバ制御の自動設定が無効になっている場合にのみ適用されます。
[サーバ側ページング制御を使用する]	LDAP クエリーの結果をページングするために、サーバ側ページング制御を使用するかどうかを指定します。デフォルト値は [いいえ (ページング制御を使用しない)] です。LDAP サー

プロパティ	説明
	バ制御の自動設定が無効になっている場合にのみ適用されます。
[サーバ側並べ替え制御を使用する]	LDAP クエリーの結果を特定の順番に並べ替えるために、サーバ側並べ替え制御を使用するかどうかを指定します。デフォルト値は [いいえ (並べ替え制御を使用しない)] です。LDAP サーバ制御の自動設定が無効になっている場合にのみ適用されます。
[サーバ側制御を自動設定]	My webMethods Server による LDAP サーバ制御の自動設定を有効または無効にします。デフォルト値は [はい (制御を自動設定)] です。

[ユーザ属性] セクション

次の表に、[ユーザ属性] セクションで設定するディレクトリサービスのプロパティを示します。

プロパティ	説明
[ユーザオブジェクトクラス]	必須。外部ディレクトリサービスのユーザオブジェクトクラス属性。デフォルトは「person」です。
[ユーザ ID]	必須。外部ディレクトリサービスのユーザ ID 属性。デフォルトは「uid」です。
[名]	必須。外部ディレクトリサービスの名属性。デフォルトは「sn」です。
[姓]	必須。外部ディレクトリサービスの姓属性。デフォルトは「givenName」です。
[氏名]	必須。外部ディレクトリサービスの氏名属性。デフォルトは「cn」です。
[電子メール]	必須。外部ディレクトリサービスの電子メールアドレス属性。デフォルトは「mail」です。
[パスワード]	必須。外部ディレクトリサービスのパスワード属性。
[ユーザが無効]	ユーザが無効化されていると識別する外部ディレクトリサービス内の属性の名前。デフォルトは「true」です。
[「ユーザが無効」値の正規表現]	外部ディレクトリサービスの「ユーザが無効」属性を検証するために使用される正規表現。
[UUID]	ユーザの汎用的に一意な識別子として使用される属性の名前。最大 128 文字の文字列 (たとえば「cn」または「email」) を指定できます。

プロパティ	説明
	注: 既存のディレクトリサービスの [**UUID**] の値を変更する場合、UserDirectory_UpdateUUID ユーティリティを実行して、ディレクトリサービスのユーザの UUID 値を更新する必要があります。 詳細については、124 ページの「ユーザへの Universally Unique Identifier (UUID) の設定」を参照してください。

[グループ属性] セクション

次の表に、[グループ属性] セクションで設定するディレクトリサービスのプロパティを示します。

プロパティ	説明
[グループオブジェクトクラス]	必須。外部ディレクトリサービスのグループオブジェクトクラス属性。デフォルトは「`groupofuniquenames`」です。
[グループ ID]	必須。外部ディレクトリサービスのグループ ID 属性。デフォルトは「`cn`」です。
[グループ名]	必須。外部ディレクトリサービスのグループ名属性。デフォルトは「`cn`」です。
[グループのメンバー]	必須。外部ディレクトリサービスのグループのメンバー属性。デフォルトは「`uniquemember`」です。
[グループ電子メール]	必須。外部ディレクトリサービスのグループ電子メール属性。デフォルトは「`mail`」です。

[接続プール] セクション

次の表に、[接続プール] セクションで設定するディレクトリサービスのプロパティを示します。

プロパティ	説明
[接続数の最小値]	外部ディレクトリサーバに対して常に開いておく最小接続数。デフォルトは `1` です。
[接続数の最大値]	外部ディレクトリサーバに対して常に開いておく最大接続数。デフォルトは `20` です。
[最大接続時間]	外部ディレクトリサーバへの接続を再利用するまで開いておく最大時間。LDAP 接続が検索プロセス中に開いておくように、この値はサーバによって LDAP 検索ごとにリセットされます。デフォルトは 10 分です。

プロパティ	説明
[クリーンアップの間隔]	期限切れのLDAP接続をクリーンアップする時間間隔。デフォルトは 1 分です。

注:
一部の LDAP 実装では、ページング Cookie は特定の LDAP 接続にバインドされます。最大接続数プロパティの値が同時LDAP検索を処理するのに十分に大きく、最大接続時間プロパティの値が指定した時間範囲内で検索を終了させるのに十分に長いことを確認します。

My webMethods Server 電子メールの設定

Command Central Web ユーザインタフェースを使用して、次の手順で My webMethods Server 電子メールを設定します。

電子メールを設定するには

1. [Environment] パネルから My webMethods Server 環境を設定して、 [**Instances**] タブからインスタンスをクリックします。
2. [**Configuration**] タブをクリックします。
3. ドロップダウンリストから [**Email**] を選択します。

   Command Central は My webMethods Server の SMTP サーバ設定を表示します。
4. [**Edit**] をクリックします。
5. [**Connection Basics**] で次を指定します。

次の表に、基本的な接続設定を示します。

フィールド	説明
Server Name	SMTP サーバのホスト名。たとえば、「smtp.server.com」と入力します。
Port	SMTP サーバのポート番号。
Sender Name	サーバが送信する電子メールメッセージの [**From**] フィールドで使用されるデフォルトの名前。
Sender Email	サーバが送信する電子メールメッセージの [**From**] フィールドで使用されるデフォルトの電子メールアドレス。

6. [Advanced Setting] を展開して、次を指定します。

次の表に、高度な電子メール設定を示します。

フィールド	説明
SMTP Username	オプション。My webMethods Server が認証のために提示するユーザ名。SMTP サーバで認証が必要な場合に、ユーザ名を指定します。
SMTP Password	オプション。[**SMTP Username**] に関連付けられているパスワード。SMTP サーバで認証が必要な場合に、適切なパスワードを指定します。

7. [**Test**] をクリックし、電子メールの設定を [**Save**] します。

My webMethods Server 環境変数の設定

Command Central で My webMethods Server 変数を設定するときの考慮事項

My webMethods Server 環境変数を設定する場合、次の命名規則を考慮してください。

- 変数名と値の長さが 255 文字を超えないようにします。
- 変数名は大文字小文字を区別します。
- 変数名には、特殊文字の制限は適用されません。

My webMethods Server 変数の設定

Command Central ユーザインタフェースで My webMethods Server 環境変数を追加する場合、次のカスタムタイプのいずれかを選択します。

- **Default**:セキュアなフィールドなしで変数を作成します。
- **Secure**:値が My webMethods Server によって暗号化されているパスワード変数を作成します。

次の表に、Command Central ユーザインタフェースで My webMethods Server グローバル環境変数を追加または編集するときに入力するフィールドを示します。

フィールド	説明
Key	必須。My webMethods Server グローバル環境変数の名前。My webMethods Server は、環境変数の置換の実行時に、このキーを使用して環境変数を参照します。 **注:** 変数の編集時には、[**Key**] フィールドは無効になります。

フィールド	説明
Type	必須。My webMethods Server グローバル環境変数のデータタイプ。 **注:** 変数の編集時には、[**Type**] フィールドは無効になります。
Value	My webMethods Server グローバル環境変数の値。 セキュアな環境変数を追加または編集する場合、このフィールドで指定した値は非表示です。環境変数を保存すると、[**Value**] フィールドは空のままになります。

重要:
環境変数を追加または編集した後は、変数を使用している webMethods アプリケーションが自動的に再起動されます。

My webMethods Server インスタンスの KPI の監視

Command Central Web ユーザインタフェースで My webMethods Server インスタンスの KPI を監視するには

1. [Environments] パネルで、監視する環境を選択します。
2. [**Instances**] タブをクリックします。
3. 監視する My webMethods Server を選択します。
4. [**Overview**] タブをクリックします。

[**Dashboard**] の [**Monitoring**] セクションに、My webMethods Server インスタンスの KPI が表示されます。

次の表に、My webMethods Server が返す KPI を示します。

名前	マージナル値	クリティカル値	最大値
Number of user sessions	最大値 の 80%	最大値 の 95%	少なくとも 100 または最高水準点。 (最高水準点は到達したことのある最高値)
JDBC connection pool size (JDBC への接続の最大数)	最大値 の 80%	最大値 の 95%	設定どおり。

名前	マージナル値	クリティカル値	最大値
Average response time (ミリ秒)	最大値 の 50%	最大値 の 90%	少なくとも 10 秒または最高水準点。

信用できる認証を使用した My webMethods Server への接続

デフォルトでは、Command Central のバージョン 9.8 以上が My webMethods Server に接続する際に、SAML ベースの信頼できる認証を使用します。この機能を有効にする操作を実行する必要はありません。

ただし、My webMethods Server 管理 Web ユーザインタフェースで [**Assertion Parameter Name**] フィールドの値を変更する場合、My webMethods Server ランタイムコンポーネントを管理する Platform Manager の custom_wrapper.conf ファイルで追加の Java パラメータ MWS.SAML.paramname を設定する必要があります。MWS.SAML.paramname パラメータでは、変更した [Assertion Parameter Name] の値を指定します。アサーションパラメータ名の値の設定の詳細については、「*Administering My webMethods Server*」を参照してください。

Platform Manager で MWS.SAML.paramname パラメータを設定するには

1. *Software AG_directory* \profiles\SPM\configuration ディレクトリに移動し、custom_wrapper.conf ファイルをテキストエディタで開きます。

2. 次のパラメータを追加します。

   ```
   wrapper.java.additional.number=
   -DMWS.SAML.paramname=assertionParameterName
   ```

   number はファイルに既存しているパラメータの数で決まる一意のシーケンス番号、*assertionParameterName* は My webMethods Server 変更された管理ユーザインタフェース内の [**Assertion Parameter Name**] フィールドの値です。

3. ファイルを保存して、Platform Manager を再起動します。

27 コマンドラインを使用する My webMethods Server の管理

- My webMethods Server がサポートするコマンド 494
- My webMethods Server-ENGINE がサポートする設定タイプ 496
- My webMethods Server-ENGINE のライフサイクルアクション 498
- My webMethods Server インスタンスの管理 499
- My webMethods Server-ENGINE のランタイムの監視状態 504
- My webMethods Server ランタイムの監視状態 505

My webMethods Server がサポートするコマンド

次の表に、My webMethods Server がサポートするすべての Platform Manager コマンドと、各コマンドの参照先を示します。

コマンド	参照先
`sagcc create configuration data`	このコマンドの一般情報については、『*Software AG Command Central Help*』を参照してください。 このコマンドの使用に関する My webMethods Server 固有の情報については、496 ページの「My webMethods Server-ENGINE がサポートする設定タイプ」を参照してください。
`sagcc delete configuration data`	このコマンドの一般情報については、『*Software AG Command Central Help*』を参照してください。
`sagcc get configuration data`	このコマンドの一般情報については、『*Software AG Command Central Help*』を参照してください。 このコマンドの使用に関する My webMethods Server 固有の情報については、496 ページの「My webMethods Server-ENGINE がサポートする設定タイプ」を参照してください。
`sagcc update configuration data`	このコマンドの一般情報については、『*Software AG Command Central Help*』を参照してください。
`sagcc get configuration instances`	このコマンドの一般情報については、『*Software AG Command Central Help*』を参照してください。
`sagcc list configuration instances`	このコマンドの一般情報については、『*Software AG Command Central Help*』を参照してください。 このコマンドの使用に関する My webMethods Server 固有の情報については、496 ページの「My webMethods Server-ENGINE がサポートする設定タイプ」を参照してください。
`sagcc get configuration types`	このコマンドの一般情報については、『*Software AG Command Central Help*』を参照してください。 このコマンドの使用に関する My webMethods Server 固有の情報については、496 ページの「My webMethods Server-ENGINE がサポートする設定タイプ」を参照してください。
`sagcc list configuration types`	このコマンドの一般情報については、『*Software AG Command Central Help*』を参照してください。

コマンド	参照先
	このコマンドの使用に関する My webMethods Server 固有の情報については、496 ページの「My webMethods Server-ENGINE がサポートする設定タイプ」を参照してください。
`sagcc exec configuration validation create`	このコマンドの一般情報については、『*Software AG Command Central Help*』を参照してください。 このコマンドの使用に関する My webMethods Server 固有の情報については、496 ページの「My webMethods Server-ENGINE がサポートする設定タイプ」を参照してください。
`sagcc exec configuration validation delete`	このコマンドの一般情報については、『*Software AG Command Central Help*』を参照してください。
`sagcc exec configuration validation update`	このコマンドの一般情報については、『*Software AG Command Central Help*』を参照してください。 このコマンドの使用に関する My webMethods Server 固有の情報については、496 ページの「My webMethods Server-ENGINE がサポートする設定タイプ」を参照してください。
`sagcc create instances`	このコマンドの一般情報については、『*Software AG Command Central Help*』を参照してください。 このコマンドの使用に関する My webMethods Server 固有の情報については、499 ページの「My webMethods Server インスタンスの作成」を参照してください。
`sagcc delete instances`	このコマンドの一般情報については、『*Software AG Command Central Help*』を参照してください。
`sagcc list instances supported products`	このコマンドの一般情報については、『*Software AG Command Central Help*』を参照してください。
`sagcc update instances`	このコマンドの一般情報については、『*Software AG Command Central Help*』を参照してください。 このコマンドの使用に関する My webMethods Server 固有の情報については、503 ページの「My webMethods Server インスタンスの更新」を参照してください。
`sagcc get inventory components`	このコマンドの一般情報については、『*Software AG Command Central Help*』を参照してください。
`sagcc list inventory components`	このコマンドの一般情報については、『*Software AG Command Central Help*』を参照してください。

コマンド	参照先
`sagcc update inventory components`	このコマンドの一般情報については、『*Software AG Command Central Help*』を参照してください。
`sagcc exec lifecycle`	このコマンドの一般情報については、『*Software AG Command Central Help*』を参照してください。 このコマンドの使用に関する My webMethods Server 固有の情報については、498 ページの「My webMethods Server-ENGINE のライフサイクルアクション」を参照してください。
`sagcc get monitoring`	このコマンドの一般情報については、『*Software AG Command Central Help*』を参照してください。 このコマンドの使用に関する My webMethods Server 固有の情報については、以下を参照してください。 ■ 504 ページの「My webMethods Server-ENGINE のランタイムの監視状態」 ■ 505 ページの「My webMethods Server ランタイムの監視状態」

My webMethods Server-ENGINE がサポートする設定タイプ

My webMethods Server-ENGINE ランタイムコンポーネントは、次の表に示す設定タイプのインスタンスの作成をサポートします。

設定タイプ	設定対象
COMMON-ADMINUI	My webMethods Server への完全 URL。 ■ My webMethods Server HTTP ポートが有効な場合は、*hostname* に My webMethods Server ホスト名、*httpport* に My webMethods Server HTTP ポート番号を入れて次の形式を使用します。 http://*hostname*:*httpport* ■ My webMethods Server HTTPS ポートが有効で、HTTP ポートが有効ではない場合は、*hostname* に My webMethods Server ホスト名、*httpport* に My webMethods Server HTTPS ポート番号を入れて次の形式を使用します。 https://*hostname*:*httpsport*

設定タイプ	設定対象
COMMON-CLUSTER	My webMethods Server クラスタの設定。My webMethods Server クラスタのロードバランサ、JMS プロバイダ、およびノード役割を設定できます。指定するフィールドと値の詳細については、「*Administering My webMethods Server*」を参照してください。 **重要:** ノードを再起動した後にのみ、クラスタの設定の変更が反映されます。
COMMON-JDBC	My webMethods Server データベース接続のデフォルトの接続プール。コマンドラインインタフェースでプールを編集できますが、削除はできません。 また、My webMethods Server で実行されているカスタム Composite Application Framework (CAF) アプリケーションが使用する追加のカスタム JDBC プールを追加、更新、削除することができます。 **注:** コマンドラインインタフェースでこの設定タイプのインスタンスを管理できますが、Command Central ユーザインタフェースでは管理できません。
COMMON-KEYSTORES	My webMethods Server のキーストア。My webMethods Server が HTTPS ポートに使用するキーストアを編集して、独自のキーストアを指定できます。 **注:** コマンドラインインタフェースでこの設定タイプのインスタンスを管理できますが、Command Central ユーザインタフェースでは管理できません。
COMMON-LOCAL-USERS	My webMethods Server インスタンスの内部ユーザを管理するための設定。COMMON-LOCAL-USERS-*userId* は、各ユーザの詳細の設定をサポートします。指定するフィールドと値の詳細については、「*Administering My webMethods Server*」を参照してください。
COMMON-PORTS	My webMethods Server HTTP ポートと HTTPS ポート。 ポートを追加、編集、削除する場合、次の点に注意してください。 ■ 少なくとも HTTP ポートが 1 つまたは HTTPS ポートが 1 つ定義されていることを確認します。

設定タイプ	設定対象
	■ HTTP ポートが定義されている場合のみ、HTTPS ポートを削除できます。HTTPS ポートを削除しても、後で再度追加できます。 ■ HTTPS ポートが定義されている場合のみ、HTTP ポートを削除できます。HTTP ポートを削除しても、後で再度追加できます。 ■ ポート番号の編集に関しては制限がありません。 **注:** ポートが有効な場合は、Command Central ユーザインタフェースでもポートを管理できます。My webMethods Server クラスタ管理でポートを有効にして、My webMethods Server を再起動すると、ポートが Command Central ユーザインタフェースに表示されます。
COMMON-SMTP	電子メールメッセージを送信するための設定。
COMMON-TRUSTSTORES	My webMethods Server のトラストストア。My webMethods Server が HTTPS ポートに使用するトラストストアを編集して、独自のトラストストアを指定できます。 **注:** コマンドラインインタフェースでこの設定タイプのインスタンスを管理できますが、Command Central ユーザインタフェースでは管理できません。
COMMON-VARS	My webMethods Server のグローバル環境変数。

My webMethods Server-ENGINE のライフサイクルアクション

Software AG Command Central コマンドラインユーティリティまたは Web ユーザインタフェースを使用して、My webMethods Server のライフサイクル操作を実行できます。Web ユーザインタフェースの詳細については、481 ページの「My webMethods Server インスタンス動作の一時停止と再開」を参照してください。

`sagcc exec administration` コマンドの一般的な情報については、*Software AG Command Central Help*を参照してください。

以下の表に、My webMethods Server-ENGINE が `sagcc exec lifecycle` コマンドを使ってサポートするアクション、および、アクションの実行時に My webMethods Server ランタイムコンポーネントが返す状態を示します。

アクション	説明
pause	ランタイムコンポーネントを保守モードにします。成功した場合、ランタイム状態が NOT_READY に設定されます。
resume	ランタイムコンポーネントを保守モードから通常状態に戻します。成功した場合、ランタイム状態が ONLINE に設定されます。

My webMethods Server-ENGINE ランタイムコンポーネントが報告するランタイムの状態の詳細情報については、504 ページの「My webMethods Server-ENGINE のランタイムの監視状態」 を参照してください。

My webMethods Server インスタンスの管理

My webMethods Server インスタンスの作成

次の表に、Command Central インスタンス管理コマンドを使用して My webMethods Server インスタンスを作成するときに含めるパラメータを示します。

コマンド	パラメータ	説明
sagcc create instances	MwsProgramFiles	必須。インストール済み My webMethods Server 製品 ID。
	instance.name=*name*	必須。新規 My webMethods Server インスタンスの名前。デフォルト名は default です。
	fixRepository=*repoID*	オプションですが推奨されています。製品の初期インストール時に修正をインストールするために使用された修正リポジトリの ID を指定します。この ID は Command Central に登録されています。このパラメータを使用すると、Command Central は、作成される新しい製品インスタンスに必要な修正をこのリポジトリから再インストールします。 このパラメータを含めない場合、Command Central は新しい製品インスタンスに修正を適用しないことに注意してください。
	db.type=*type*	必須。新しいサーバインスタンスが使用するデータベースのタイプ。次のいずれかのデータベースを指定します。 ■ sqlserver または ms: Microsoft SQL Server ■ oracle: Oracle

コマンド	パラメータ	説明
		■ db2: DB2 ■ mysqlee - MySQL Enterprise Edition ■ mysqlce - MySQL Community Edition ■ postgresql - PostgreSQL
	db.url="*url*"	必須。データベースの接続 URL。*url* 値を二重引用符で囲みます。
	db.username=*name*	必須。My webMethods Server データベースに割り当てられたユーザ名。
	db.password=*password*	必須。My webMethods Server データベースユーザのパスワード。
	[install.service={true|false}]	オプション。My webMethods Server の新規インスタンスをアプリケーションとしてインストールするか、サービスとしてインストールするかを指定します。有効な値は次のとおりです。 ■ true - サービスとしてインストールします。 ■ false-アプリケーションとしてインストールします (デフォルト)。
	db.driver=*name*	オプション。My webMethods Server データベースへの接続に使用する JDBC ドライバーのクラス名。
	node.name=*name*	オプション。新規サーバインスタンスをホストするクラスタノードの名前。デフォルトの名前は「master」です。
	http.port=*port*	オプション。新規サーバインスタンスが受信待機するポート番号。デフォルトのポート番号は 8585 です。
	https.port=*port*	オプション。新規サーバインスタンスが受信待機する HTTPS ポート。値 0 を指定すると、リスナーが無効になります。
	debug.port=*port*	オプション。新規サーバインスタンスの Java デバッグポート。デフォルトのポート番号は 10033 です。
	jmx.port=*port*	オプション。新規サーバインスタンスの JMX ポート。デフォルトのポート番号は 5002 です。

コマンド	パラメータ	説明
	server.features={core|default|fabric|all}	オプション。新規サーバインスタンスに設定されたコンポーネント機能のセット。次のいずれかの値を指定します。 ■ core: Software AG Designer を使用した JSR 168 ポートレットの開発をサポートするために必要な最小限の機能セット。この値は、単一のスキンおよびシェルを持つ純粋なランタイムを示します。管理機能または設定機能はありません。 ■ default: My webMethods Server 機能の標準セット。 *Software AG_directory* \MWS\ components ディレクトリにある追加機能や開発ツールは含まれません。これがデフォルト値です。 ■ fabric: インストールされているすべての My webMethods Server ユーザインタフェースを含む My webMethods Server のデフォルト分類。 ■ all: 機能の標準セットおよび *Software AG_directory* \MWS\components ディレクトリにあるすべてのコンポーネント。
	components.include=*name*	オプション。新規サーバインスタンスに含める *Software AG_directory* \MWS\components ディレクトリ内のコンポーネント。
	components.exclude=*name*	オプション。新規サーバインスタンスから除外する *Software AG_directory* \MWS\components ディレクトリ内のコンポーネント。
	components.overwrite= {true|false}	オプション。 *Software AG_directory* \MWS\components ディレクトリ内のコンポーネントファイルより古い *Software AG_directory* \MWS\server*serverName*\deploy ディレクトリ内のコンポーネントファイルを上書きするかどうかを指定します。有効な値は次のとおりです。 ■ true - \deploy ディレクトリ内のコンポーネントファイルを上書きします。 ■ false - \deploy ディレクトリ内のコンポーネントファイルを上書きしません。これがデフォルト値です。

Command Central 実行する場合の例

- 次の値を使用して、エイリアス名「local」のインストールにインストール済み My webMethods Server に対して新規インスタンスを作成する例: インスタンス名: test1、データベースタイプ: sqlserver、データベースユーザ名: mws、データベースのパスワード: mws、データベース URL: jdbc:wm:sqlserver://localhost:1433;databaseName=testDB。

```
sagcc create instances local MwsProgramFiles  instance.name=test1
db.type=sqlserver db.username=mws db.password=mws
db.url="jdbc:wm:sqlserver://localhost:1433;databaseName=testDB"
```

- 次の値を使用して、エイリアス名「local」のインストールにインストール済み My webMethods Server に対して新規インスタンスを作成する例: インスタンス名: test1、データベースタイプ: sqlserver、データベースユーザ名: mws、データベースのパスワード: mws、データベース URL: jdbc:wm:sqlserver://localhost:1433;databaseName=testDB、データベースドライバ: com.wm.dd.jdbc.sqlserver.SQLServerDriver、 HTTP ポート: 9595、JMX ポート: 7000。

```
sagcc create instances local MwsProgramFiles  instance.name=test1
db.type=sqlserver db.username=mws db.password=mws
db.url="jdbc:wm:sqlserver://localhost:1433;databaseName=testDB"
db.driver=com.wm.dd.jdbc.sqlserver.SQLServerDriver http.port=9595
jmx.port=7000
```

- 次の値を使用して、ローカルインストールにインストール済みの My webMethods Server に対して新規インスタンスを作成し、インスタンスをサービスとして登録する例: インスタンス名: test1、HTTP ポート: 9595、JMX ポート: 7000。新規 My webMethods Server インスタンスは、次の値の外部データベースを使用します: データベースタイプ: sqlserver、データベースユーザ名: mws、データベースのパスワード: mws、データベース URL: jdbc:wm:sqlserver://localhost:1433;databaseName=testDB、データベースドライバ: com.wm.dd.jdbc.sqlserver.SQLServerDriver。

```
sagcc create instances local MwsProgramFiles instance.name=test1
db.type=sqlserver db.username=mws db.password=mws
db.url="jdbc:wm:sqlserver://localhost:1433;databaseName=testDB"
db.driver=com.wm.dd.jdbc.sqlserver.SQLServerDriver http.port=9595
jmx.port=7000 install.service=true
```

Platform Manager 実行する場合の例

- 次の値を使用して、インストール済みの My webMethods Server に対して新規インスタンスを作成する例: インスタンス名: test1、データベースタイプ: sqlserver、データベースユーザ名: mws、データベースのパスワード: mws、データベース URL: jdbc:wm:sqlserver://localhost:1433;databaseName=testDB。

```
sagcc create instances MwsProgramFiles  instance.name=test1
db.type=sqlserver db.username=mws db.password=mws
db.url="jdbc:wm:sqlserver://localhost:1433;databaseName=testDB"
```

- 次の値を使用して、インストール済みの My webMethods Server に対して新規インスタンスを作成する例: インスタンス名: test1、データベースタイプ: sqlserver、データベースユーザ名: mws、データベースのパスワード: mws、データベース URL: jdbc:wm:sqlserver://localhost:1433;databaseName=testDB、データベースドライバ: com.wm.dd.jdbc.sqlserver.SQLServerDriver、 HTTP ポート: 9595、JMX ポート: 7000。

```
sagcc create instances MwsProgramFiles  instance.name=test1
db.type=sqlserver db.username=mws db.password=mws
db.url="jdbc:wm:sqlserver://localhost:1433;databaseName=testDB"
db.driver=com.wm.dd.jdbc.sqlserver.SQLServerDriver http.port=9595
jmx.port=7000
```

My webMethods Server インスタンスの更新

次の表に、`sagcc update instances` コマンドを使用して My webMethods Server インスタンスを更新するときに含めるパラメータを示します。

パラメータ	説明
`MwsProgramFiles-`	必須。更新する My webMethods Server インスタンスの ID。たとえば `MwsProgramFiles-test1` です。
`components.overwrite={true|false}`	オプション。 *Software AG_directory* \MWS\components ディレクトリ内のコンポーネントファイルより古い *Software AG_directory* \MWS\server*serverName*\deploy ディレクトリ内のコンポーネントファイルを上書きするかどうかを指定します。有効な値は次のとおりです。 ■ `true` - \deploy ディレクトリ内のコンポーネントファイルを上書きします。 ■ `false` - \deploy ディレクトリ内のコンポーネントファイルを上書きしません。これがデフォルト値です。
`dont.update.classpath={true|false}`	オプション。生成された My webMethods Server クラスパスを更新するかどうかを指定します。有効な値は次のとおりです。 ■ `true` - サーバクラスパスを更新しません。 ■ `false` - サーバクラスパスを更新します。これがデフォルト値です。
`platformdir=`*`path`*	オプション。プラットフォームのインストールディレクトリへのパス。デフォルトパスは *Software AG_directory* です。
`node.name=`*`name`*	オプション。My webMethods Server インスタンスをホストするクラスタノードの名前。インスタンスを更新するときに、新しい値を指定してクラスタノードの名前を変更します。
`timeout=`*`number`*	オプション。コマンド実行の試みを終了する前に、コマンドの終了を待つ秒数を指定します。

`sagcc update instances` コマンドの使用方法の詳細については、*Software AG Command Central Help*を参照してください。

My webMethods Server インスタンスの削除

`sagcc delete instances` コマンドを使用して、My webMethods Server インスタンスを削除できます。`sagcc delete instances` コマンドの使用方法の詳細については、*Software AG Command Central Help*を参照してください。

My webMethods Server-ENGINE のランタイムの監視状態

次の表は、`sagcc get monitoring state` コマンドに応答して、My webMethods Server-ENGINE ランタイムコンポーネントが返すことができるランタイムの状態と、各状態の意味を示します。

ランタイムの状態	意味
ONLINE	My webMethods Server-ENGINE ランタイムコンポーネントは実行されています。 プロファイル JVM が実行されていて、My webMethods Server ポートが応答している場合、ランタイムコンポーネントは ONLINE を示します。
ERROR	何らかの障害のため My webMethods Server-ENGINE ランタイムコンポーネントは実行されておらず、再起動にも失敗しました。 My webMethods Server がデータベースに接続できない場合、または、コンポーネントのインストールが失敗した場合、ランタイムコンポーネントは ERROR を示します。
NOT_READY	My webMethods Server-ENGINE ランタイムコンポーネントの準備ができていません。 サーバが保守モードの場合、ランタイムコンポーネントは NOT_READY を示します。
STARTING	My webMethods Server-ENGINE ランタイムコンポーネントは起動中です。 最初にサーバが起動したとき、または、起動中に HTTP 503 状態が報告された場合に、ランタイムコンポーネントは STARTING を示します。
STOPPED	My webMethods Server-ENGINE ランタイムコンポーネントは停止しています。
UNKNOWN	My webMethods Server-ENGINE ランタイムコンポーネントの状態は判定できません。

My webMethods Server ランタイムの監視状態

`sagcc get monitoring state` コマンドに応答して、My webMethods Server はキーパフォーマンスインジケータ (KPI) に関する情報を表示します。次の表に、My webMethods Server で有効な KPI を示します。

KPI	説明
Number of user sessions	この KPI でアクティブなユーザセッションの数を監視して、数がクリティカル値に近づいた場合に修正処置を行えるようにします。 ■ マージナル値は 80 のアクティブなユーザセッションです。 ■ クリティカル値は 95 のアクティブなユーザセッションです。 ■ 最大値は 100 以上のアクティブなユーザセッションです。
Number of active connections in the JDBC poo	この KPI で JDBC プールのアクティブな接続の数を監視して、接続数がクリティカル値に近づいた場合に修正処置を行えるようにします。 ■ マージナル値は 90 のアクティブな接続です。 ■ クリティカル値は 95 のアクティブな接続です。 ■ 最大値は 100 以上のアクティブな接続です。
Average response times in milliseconds (ミリ秒)	この KPI で My webMethods Server の応答時間を監視して、応答時間がクリティカル値に近づいた場合に修正処置を行えるようにします。 ■ マージナル値は 5000 ミリ秒です。 ■ クリティカル値は 9000 ミリ秒です。 ■ 最大値は 10000 ミリ秒以上の応答時間です。

28 My webMethods Server の認証

- My webMethods Server の認証モードの変更 508
- Command Central で My webMethods Server の管理者ユーザパスワードを変更するには 508
- UNIX シェルスクリプトを使用した管理製品の接続クレデンシャルの変更 508
- 送信認証設定の確認 509
- Command Central を介した管理インタフェースへのアクセス 509

My webMethods Server の認証モードの変更

Command Central のデフォルトでは、信用できる認証をサポートするランタイムコンポーネントの認証モードは、[**Trusted**] に設定されています。

インスタンスの [Overview] タブで、[Authentication] フィールドの をクリックすることにより、[Authentication Mode] ダイアログボックスを使って認証モードを変更します。

インスタンスに認証モードを指定すると、主製品のインスタンスのすべての階層化された製品インスタンスに対しても、その認証モードが設定されます。ただし、My webMethods Server の OSGI プロファイルの認証モードを変更しても、OSGI プロファイルに属する My webMethods Server ランタイムコンポーネントの認証モードは変更されません。

基本認証を使用するには、ランタイムコンポーネントの認証モードを [**Fixed User**] に変更する必要があります。Command Central は固定ユーザに基本認証を使用して、Platform Manager と通信します。[**Fixed User**] 認証を使用すると、Platform Manager の認証クレデンシャルは固定されます。

Command Central で My webMethods Server の管理者ユーザパスワードを変更するには

Command Central に管理されている製品の管理者パスワードを、Command Central Web ユーザインタフェースで変更します。Command Central で管理製品の管理者パスワードを変更した後、送信クレデンシャルは自動的に更新されます。

Command Central で My webMethods Server の管理者ユーザパスワードを変更するには

1. Command Central の [Environments] パネルで、管理製品のインスタンスを含む環境を選択します。
2. インスタンスの表で、MWS-*instanceName* 下にある My webMethods Server コンポーネントを選択します (たとえば MWS-default)。
3. 設定タブで、[**Users**] を選択します。
4. ユーザページで、[**Administrator**] をクリックします。
5. [**Edit**] をクリックし、[**Administrator**] の新しいパスワードを指定します。

UNIX シェルスクリプトを使用した管理製品の接続クレデンシャルの変更

次のサンプル UNIX シェルスクリプトを使用して、Command Central が管理する製品コンポーネントの基本認証クレデンシャルを設定できます。

```
NODE_ALIAS=local
USERNAME=Administrator
PASSWORD=secret
RCID=integrationServer-default
# RCID=MwsProgramFiles-default
# RCID=Universal-Messaging-nirvana
# RCID=OSGI-CTP
# RCID=OSGI-InfraDC

sagcc get configuration data $NODE_ALIAS $RCID COMMON-LOCAL-USERS-Administrator

-o administrator.xml
sed "s,/>,><Password>${PASSWORD}</Password></User>,g" administrator.xml >

 administrator_new.xml
sagcc update configuration data $NODE_ALIAS $RCID COMMON-LOCAL-USERS

-Administrator -o administrator_new.xml

# verify connection
sagcc get monitoring runtimestatus $NODE_ALIAS $RCID -e ONLINE
```

送信認証設定の確認

次の手順で、Command Central が正しい送信認証設定で設定されていることを確認します。

Command Central が正しいユーザクレデンシャルで設定されていることを確認するには

1. Command Central で、製品コンポーネントの [**Overview**] タブを開き、 をクリックします。製品の状態が [**Online**] であり、JVM KPI が更新されていることを確認します。

2. [**Logs**] タブで、認証エラーの製品ログを確認します。

Command Central を介した管理インタフェースへのアクセス

Command Central で、SSO (Single Sign-On : シングルサインオン) は、インストール後の設定なしに管理リンクを使用して webMethods 製品を管理するよう設計されています。拡張設定タスクを実行する場合、製品のプライマリ管理インタフェースへのアクセスが必要なことがあります。Command Central では、管理された各製品の [Instances Overview] ページに管理インタフェースへのリンクが用意されています。たとえば、My webMethods Server インスタンスの [Overview] ページの My webMethods リンクをクリックすると、Command Central は対応する My webMethods の URL にブラウザをリダイレクトします。

重要:
ランドスケープを管理するには、インスタンスを *1* つのみCommand Central使用します。特定の Command Central インスタンスの Command Central Web ユーザインタフェースに別の Command Central インスタンスからアクセスすることはできません。

Software AG Common Platform ベースの製品のカスタム SSO および SAML 認証を生成し設定する方法の詳細については、Software AG のセキュリティインフラストラクチャのマニュアルを参照してください。