

Oracle® Real-Time Decisions

インストレーションおよび管理ガイド

Oracle Real-Time Decisions インストレーションおよび管理ガイド, リリース 2.2

部品番号: E05253-01

原本名: Oracle Real-Time Decisions Installation and Administration of Oracle RTD, Version 2.2

Copyright © 2003, 2007, Oracle. All rights reserved.

制限付権利の説明

このプログラム（ソフトウェアおよびドキュメントを含む）には、オラクル社およびその関連会社に所有権のある情報が含まれています。このプログラムの使用または開示は、オラクル社およびその関連会社との契約に記載された制約条件に従うものとします。著作権、特許権およびその他の知的財産権と工業所有権に関する法律により保護されています。

独立して作成された他のソフトウェアとの互換性を得るために必要な場合、もしくは法律によって規定される場合を除き、このプログラムのリバース・エンジニアリング、逆アセンブル、逆コンパイル等は禁止されています。

このドキュメントの情報は、予告なしに変更される場合があります。オラクル社およびその関連会社は、このドキュメントに誤りが無いことの保証は致し兼ねます。これらのプログラムのライセンス契約で許諾されている場合を除き、プログラムを形式、手段（電子的または機械的）、目的に関係なく、複製または転用することはできません。

このプログラムが米国政府機関、もしくは米国政府機関に代わってこのプログラムをライセンスまたは使用する者に提供される場合は、次の注意が適用されます。

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

このプログラムは、核、航空産業、大量輸送、医療あるいはその他の危険が伴うアプリケーションへの用途を目的としておりません。このプログラムをかかるとして使用する際、上述のアプリケーションを安全に使用するために、適切な安全装置、バックアップ、冗長性（redundancy）、その他の対策を講じることは使用者の責任となります。万が一かかるプログラムの使用に起因して損害が発生いたしましても、オラクル社およびその関連会社は一切責任を負いかねます。

Oracle、JD Edwards、PeopleSoft、Siebel は米国 Oracle Corporation およびその子会社、関連会社の登録商標です。その他の名称は、他社の商標の可能性がありまます。

このプログラムは、第三者の Web サイトへリンクし、第三者のコンテンツ、製品、サービスへアクセスすることがあります。オラクル社およびその関連会社は第三者の Web サイトで提供されるコンテンツについては、一切の責任を負いかねます。当該コンテンツの利用は、お客様の責任になります。第三者の製品またはサービスを購入する場合は、第三者と直接の取引となります。オラクル社およびその関連会社は、第三者の製品およびサービスの品質、契約の履行（製品またはサービスの提供、保証義務を含む）に関しては責任を負いかねます。また、第三者との取引により損失や損害が発生いたしましても、オラクル社およびその関連会社は一切の責任を負いかねます。

インストールおよび管理ガイド

1	インストール前の準備	10
1.1	Oracle Real-Time Decisions のインストールについて	10
1.2	データベース・サーバーのインストール.....	11
1.3	Java Development Kit (JDK) のインストール	11
1.3.1	スタンドアロン OC4J へのインストール	11
1.3.2	Oracle Application Server へのインストール.....	12
1.3.3	WebSphere へのインストール	12
1.3.4	WebLogic へのインストール	12
1.4	アプリケーション・サーバーのインストール.....	13
1.4.1	Oracle RTD 用の OC4J のインストール.....	13
1.4.2	Oracle RTD 用の WebSphere のインストール.....	15
1.4.3	Oracle RTD 用の WebLogic のインストール	15
1.5	インストール前のその他のタスク	16
2	Real-Time Decision Server のインストール.....	17
2.1	Oracle Real-Time Decisions のファイルのインストール	17
2.1.1	単一の Windows コンピュータへの Oracle RTD のインストール.....	17
2.1.2	UNIX への Real-Time Decision Server のインストール.....	18
2.1.3	Oracle RTD のランタイム環境について	18
2.2	Oracle RTD Database の作成と初期化.....	19
2.2.1	データベースの作成: Microsoft SQL Server	19
2.2.2	データベースの作成: Oracle Database.....	20
2.2.3	データベースの作成: IBM DB2	20
2.2.4	SDDDBTool による Oracle RTD Database の初期化.....	20
2.3	CrossSell サンプル・データの移入 (オプション)	21

2.4	Oracle Real-Time Decisions での SSL の使用	22
3	Oracle Real-Time Decisions 用の OC4J の構成	24
3.1	サーバー・プロパティの構成	24
3.1.1	スタンドアロン OC4J のサーバー・プロパティの構成	24
3.1.2	Oracle Application Server のサーバー・プロパティの構成	26
3.2	Oracle RTD Database の JDBC プロバイダの作成	28
3.3	OC4J への Oracle Real-Time Decisions アプリケーションのインストール	30
3.4	Oracle Real-Time Decisions の起動	30
3.5	OC4J からの Oracle Real-Time Decisions アプリケーションのアンインストール	31
3.6	Real-Time Decision Server の SSL の構成（推奨）	31
3.6.1	SSL 構成のテスト	33
3.7	OC4J に対する JConsole の設定	34
3.8	Oracle Real-Time Decisions のポート番号の変更	36
3.8.1	OC4J でのポート番号の変更	36
3.8.2	Oracle RTD でのポート番号の変更	36
4	Oracle Real-Time Decisions 用の WebSphere の構成	37
4.1	サーバー・プロパティの構成	37
4.2	Oracle RTD Database の JDBC プロバイダの作成	40
4.3	WebSphere への Oracle Real-Time Decisions アプリケーションのインストール	42
4.4	Oracle Real-Time Decisions の起動	43
4.5	WebSphere からの Oracle Real-Time Decisions アプリケーションのアンインストール	43
4.6	Real-Time Decision Server の SSL の構成（推奨）	44
4.6.1	SSL 構成のテスト	47
4.7	WebSphere に対する JConsole の設定	47
4.7.1	WebSphere のブートストラップ・ポート番号の確認	47
4.7.2	JConsole のバッチ・ファイルの設定	48

4.7.3	JConsole ユーザーの作成.....	49
4.8	Oracle Real-Time Decisions のポート番号の変更.....	50
4.8.1	WebSphere でのポート番号の変更.....	50
4.8.2	Oracle RTD でのポート番号の変更.....	50
5	Oracle Real-Time Decisions 用の WebLogic の構成.....	52
5.1	サーバー・プロパティの構成.....	52
5.2	Oracle RTD Database の JDBC プロバイダの作成.....	56
5.3	WebLogic への Oracle Real-Time Decisions アプリケーションのインストール.....	57
5.4	Oracle Real-Time Decisions の起動.....	58
5.5	WebLogic からの Oracle Real-Time Decisions アプリケーションのアンインストール.....	59
5.6	Real-Time Decision Server の SSL の構成（推奨）.....	59
5.6.1	SSL 構成のテスト.....	61
5.7	WebLogic に対する JConsole の設定.....	62
5.8	Oracle Real-Time Decisions のポート番号の変更.....	63
5.8.1	WebLogic でのポート番号の変更.....	63
5.8.2	Oracle RTD でのポート番号の変更.....	64
6	Oracle Real-Time Decisions の認証の構成.....	65
6.1	Oracle Real-Time Decisions のセキュリティの概要.....	65
6.1.1	Oracle RTD のデフォルトのアカウントおよびパスワード.....	66
6.1.2	Oracle RTD の構成ファイルおよびログ・ファイル.....	66
6.1.3	Oracle RTD の Cookie.....	69
6.2	Windows Authentication の構成.....	69
6.3	Oracle RTD Platform Authentication の構成.....	71
6.4	Oracle Siebel Object Manager Authentication の構成.....	71
6.4.1	アプリケーション・サーバー用の jar ファイルへのアクセスの設定.....	72
6.4.2	Oracle RTD での認証の有効化.....	73

6.5	ユーザーおよびグループへの権限の割当て	74
6.6	認証タイプの変更	75
7	Oracle Real-Time Decisions のデータ・アクセスの構成	76
7.1	OC4J での追加の JDBC データソースの作成	76
7.1.1	データソース用の JDBC jar ファイルへのアクセスの設定	76
7.1.2	OC4J での接続プールとデータソースの作成	77
7.1.3	Oracle RTD への新しいデータソースの追加	79
7.2	WebSphere での追加の JDBC データソースの作成	80
7.2.1	WebSphere でのデータ・プロバイダとデータソースの作成	80
7.2.2	Oracle RTD への新しいデータソースの追加	83
7.2.3	WebSphere でのリソース参照の更新	84
7.2.4	別のデータソースの追加	85
7.3	WebLogic での追加の JDBC データソースの構成	85
7.3.1	データソース用の JDBC jar ファイルへのパスの設定	85
7.3.2	WebLogic でのデータソースの作成	87
7.3.3	Oracle RTD への新しいデータソースの追加	89
7.4	新しいエンタープライズ・データソースのテスト	90
8	追加の構成設定とクライアント・ツールの起動	92
8.1	Decision Center のブラウザの構成	92
8.2	Real-Time Decision Server のロギングの構成	92
8.3	Oracle Real-Time Decisions クライアント・ツールへのアクセス	93
8.3.1	Decision Studio へのアクセス	93
8.3.2	Decision Center へのアクセス	93
8.3.3	Load Generator へのアクセス	93
9	Oracle Real-Time Decisions の本番環境へのデプロイ	94
9.1	オール・イン・ワン構成	94

9.2	フル・デプロイ構成	94
10	JMX を使用した Oracle Real-Time Decisions の管理	96
10.1	JConsole へのアクセス	97
10.2	JMX MBean の操作と属性	97
10.3	Oracle Real-Time Decisions のクラスタ・レベルの管理	98
10.3.1	「OracleRTD」 → 「SDManagement」 → 「SDClusterPropertyManager」 について	98
10.3.2	「OracleRTD」 → 「SDClusterPropertyManager」 → 「Misc」 について	98
10.3.3	「OracleRTD」 → 「SDClusterPropertyManager」 → 「Cluster」 について	100
10.3.4	「OracleRTD」 → 「SDClusterPropertyManager」 → 「Deployment」 について	102
10.3.5	「OracleRTD」 → 「SDClusterPropertyManager」 → 「Alert Service」 について	102
10.4	Oracle Real-Time Decisions のメンバー・レベルの管理	102
10.4.1	「OracleRTD」 → 「SDManagement」 → 「SDLogging」 について	102
10.4.2	「OracleRTD」 → 「SDPropertyManager」 → 「Performance Monitoring」 について	103
10.4.3	「OracleRTD」 → 「SDPropertyManager」 → 「Misc」 について	103
10.4.4	「OracleRTD」 → 「Server」 → 「DecisionService」 について	104
10.5	Oracle Real-Time Decisions のセキュリティの管理	105
10.5.1	「OracleRTD」 → 「SDManagement」 → 「SecurityManager」 について	105
10.5.2	「OracleRTD」 → 「SDClusterPropertyManager」 → 「Security Properties」 について	108
10.5.3	「OracleRTD」 → 「SDManagement」 → 「SecurityAuthenticator」 について	109
10.6	インライン・サービスの管理	110
10.6.1	「OracleRTD」 → 「SDManagement」 → 「InlineServiceManager」 について	110
10.6.2	「OracleRTD」 → 「InlineServiceManager」 → 「インライン・サービス」 → 「デプロイ状態」 → 「フラグ」 について	110
10.7	デプロイ状態の管理	111
10.7.1	「OracleRTD」 → 「SDManagement」 → 「DeploymentStates」 について	111
10.7.2	「OracleRTD」 → 「Deployment States」 → 「状態」 について	112

10.8	ラーニング・サービスの管理	112
10.8.1	「OracleRTD」 → 「Server」 → 「LearningSevice」 について	112
10.8.2	「OracleRTD」 → 「Learning Server」 → 「スタディ」 について	112
10.8.3	「OracleRTD」 → 「Study」 → 「スタディ名」 → 「モデル」 について	113
10.9	アラート・サービスの管理	114
10.9.1	「OracleRTD」 → 「SDClusterPropertyManager」 → 「Alert Service」 について	114

はじめに

Oracle Real-Time Decisions (Oracle RTD) は、適応型のエンタープライズ・ソフトウェア・ソリューションの開発を可能にします。こうした適応型のソリューションは、ビジネス・プロセス・トランザクションが実行されるにつれて継続的に自己学習し、ルールと予測モデルに従って、各トランザクションをリアルタイムに最適化します。

このドキュメントについて

このドキュメントには、Oracle RTD のインストールと管理に関する情報が記載されています。このドキュメントでは、Oracle RTD のインストールおよび構成方法、セキュリティ目的での認証の設定方法、および Oracle RTD MBeans の管理方法について説明します。

対象読者

このドキュメントは、Oracle RTD の管理者を対象としています。Oracle RTD の管理者には、エンタープライズレベルのアプリケーションのインストールおよび管理方法に関する実際的な運用知識と、JMX コンソールの使用方法に関する知識が必要です。

このドキュメントの構成

このドキュメントの内容は次のとおりです。

第1章「インストール前の準備」では、インストール前に実行が必要なタスクについて説明しています。

第2章「Real-Time Decision Server のインストール」では、Real-Time Decision Server および Oracle RTD クライアントのファイルをインストールする手順と、Oracle RTD Database を作成および初期化する手順について説明しています。

第3章「Oracle Real-Time Decisions 用の OC4J の構成」では、Oracle Containers for J2EE (OC4J) に Oracle RTD をインストールする方法について説明しています。

第4章「Oracle Real-Time Decisions 用の WebSphere の構成」では、WebSphere に Oracle RTD をインストールする方法について説明しています。

第5章「Oracle Real-Time Decisions 用の WebLogic の構成」では、WebLogic に Oracle RTD をインストールする方法について説明しています。

第6章「Oracle Real-Time Decisions の認証の構成」では、サポートされている認証方法の設定について説明しています。




第7章「Oracle Real-Time Decisions のデータ・アクセスの構成」では、JDBC データソースを設定する方法について説明しています。

第8章「追加の構成設定とクライアント・ツールの起動」では、Oracle RTD クライアントおよびサーバーに対するその他の構成要件について説明しています。

第9章「Oracle Real-Time Decisions の本番環境へのデプロイ」では、本番環境にデプロイするシナリオについて説明しています。

第10章「JMXを使用した Oracle Real-Time Decisions の管理」では、Oracle RTD の管理に使用する JMX MBeans について説明しています。

表記規則

規則	説明
固定幅 フォント	ソース・コードおよびプログラム出力を示します。
太字	ラベル、タブ、メニューなどのユーザー・インタフェースを示します。
イタリック	ユーザー固有の値を示します。
'引用符'	ユーザーによる入力が必要な箇所を示します。
	タスクの実行に役立つ追加情報を示します。
	その項目に関する追加情報を示します。
	データの損失またはエラーが生じる可能性のあるアクションを示します。

パス名に関する注意

このドキュメントでは、複数のプラットフォームで実行可能な手順には、パス名にスラッシュ (/) が使用されています。UNIX システムでパスを入力する場合は、指定されているとおりに入力できます。Windows システムでパスを入力する場合は、スラッシュ (/) を円記号 (¥) に変更する必要があります。

このドキュメントでは、Windows オペレーティング・システム固有の手順または例には、パス名に円記号 (¥) が使用されています。

1 インストール前の準備

Oracle RTD は Windows と UNIX 両方のプラットフォームでサポートされ、Oracle Containers for J2EE (OC4J)、BEA WebLogic、IBM WebSphere の各アプリケーション・サーバーで実行されます。サポートされるプラットフォームの詳細は、『System Requirements and Supported Platforms for Oracle Real-Time Decisions』を参照してください。

注意: Oracle RTD のインストール先として OC4J を計画する場合は、スタンドアロン・バージョンの OC4J と Oracle Application Server に同梱されたバージョンの OC4J のどちらでも可能です。

この章の内容は次のとおりです。

- 1.1 Oracle Real-Time Decisions のインストールについて
- 1.2 データベース・サーバーのインストール
- 1.3 Java Development Kit (JDK) のインストール
- 1.4 アプリケーション・サーバーのインストール
- 1.5 インストール前のその他のタスク

1.1 Oracle Real-Time Decisions のインストールについて

Oracle RTD は、リアルタイムの意思決定エンジン・ソフトウェア・プラットフォームであり、次のコンポーネントで構成されています。

- Real-Time Decision Server
- Decision Studio
- Decision Center
- Load Generator
- Administration (JMX)

Real-Time Decision Server は、Windows と UNIX どちらのプラットフォームにもインストールできます。Decision Studio と Load Generator はクライアント・コンポーネントで、Windows でのみ実行できます。

Real-Time Decision Server では、Decision Studio に作成されるインライン・サービス用のオブジェクトとメタデータを格納するために、Oracle RTD Database という専用のデータベースが必要です。

Oracle RTD をインストールするには、次のタスクを実行する必要があります。

- 1 データベース・サーバー、Java Development Kit およびアプリケーション・サーバーをインストールします。
- 2 Real-Time Decision Server および Oracle RTD クライアントのファイルをインストールして、データベースを作成および初期化します。

- 3 Oracle RTD Database のデータソースを構成します。
- 4 アプリケーション・サーバーの管理コンソールなどのツールを使用して、Real-Time Decision Server をデプロイおよび構成します。
- 5 認証設定を構成します。
- 6 エンタープライズ・データソースへのアクセスを設定します。

1.2 データベース・サーバーのインストール

Oracle RTD Database として使用するデータベース・サーバーをインストールします。このデータベースは、Oracle Database、Microsoft SQL Server、IBM DB2 のいずれかであることが必要です。データベースのインストールの詳細は、各データベースのドキュメントを参照してください。サポートされるデータベースのバージョンの詳細は、『System Requirements and Supported Platforms for Oracle Real-Time Decisions』を参照してください。

次の点に注意してください。

- SQL Server を使用する場合は、データベース・サーバーをデフォルト（無名）のインスタンスとしてインストールする必要があります。
- Oracle Database を使用する場合は、`user_lock` パッケージをインストールする必要があります。オプションの Oracle パッケージのインストールの詳細は、Oracle Database のドキュメントを参照してください。
- データベース・サーバーは、Oracle RTD をインストールする予定のコンピュータまたは別のコンピュータにインストールできます。

1.3 Java Development Kit (JDK) のインストール

一部の構成では、Oracle RTD のインストールに進む前に、Java Development Kit (JDK) をダウンロードしてインストールする必要があります。この後の各項で、JDK のインストールの必要性の有無およびインストールが必要な JDK バージョンについて、構成別にその詳細を説明します。

JDK のインストールが完了したら、`JAVA_HOME` というシステム環境変数を作成し、JDK のインストール場所のフルパス名に値を設定します。

1.3.1 スタンドアロン OC4J へのインストール

Windows のスタンドアロン OC4J に Real-Time Decision Server をインストールする場合は、Oracle RTD をインストールする前に、Sun JDK バージョン 1.5.0.8 以降をインストールする必要があります。

Solaris または **Linux** のスタンドアロン OC4J に Real-Time Decision Server をインストールする場合は、Oracle RTD をインストールする前に、Sun JDK バージョン 1.5.0.8 以降をインストールする必要があります。その後で、Oracle RTD クライアント・ツール (Decision Studio、Load Generator、SDDDBTool など) をインストールする Windows コンピュータに、Sun JDK バージョン 1.5.0.8 以降をインストールします。

HP-UX のスタンドアロン OC4J に Real-Time Decision Server をインストールする場合は、Oracle RTD をインストールする前に、HP-UX JDK 5.0.05 をインストールする必要があります。その後で、Oracle RTD クライアント・ツール (Decision Studio、Load Generator、SDDDBTool など) をインストールする Windows コンピュータに、Sun JDK バージョン 1.5.0.8 以降をインストールします。

AIX ベースのシステムのスタンドアロン OC4J に Real-Time Decision Server をインストールする場合は、Oracle RTD をインストールする前に、IBM Java 5 SR2 をインストールする必要があります。その後で、Oracle RTD クライアント・ツール (Decision Studio、Load Generator、SDDDBTool など) をインストールする Windows コンピュータに、Sun JDK バージョン 1.5.0.8 以降をインストールします。

1.3.2 Oracle Application Server へのインストール

Windows の Oracle Application Server に同梱されたバージョンの OC4J に Real-Time Decision Server をインストールする場合は、JDK を個別にダウンロードする必要はありません。Oracle Application Server には、専用の JDK が同梱されています。ただし、`JAVA_HOME` システム環境変数を設定し、Oracle Application Server が使用する JDK のインストール場所のフルパス名に値を設定する必要があります。

Solaris、Linux、HP-UX または AIX ベースのシステムの Oracle Application Server に同梱されたバージョンの OC4J に Real-Time Decision Server をインストールする場合は、JDK を個別にダウンロードする必要はありません。Oracle Application Server には、専用の JDK が同梱されています。ただし、Oracle RTD クライアント・ツール (Decision Studio、Load Generator、SDDDBTool など) をインストールする Windows コンピュータに、Sun JDK バージョン 1.5.0.8 以降をインストールする必要があります。

1.3.3 WebSphere へのインストール

Windows の WebSphere に Real-Time Decision Server をインストールする場合は、JDK を個別にダウンロードする必要はありません。WebSphere には、専用の JDK が同梱されています。ただし、`JAVA_HOME` システム環境変数を設定し、WebSphere が使用する JDK のインストール場所のフルパス名に値を設定する必要があります。

UNIX (Solaris、Linux、HP-UX または AIX ベースのシステム) の WebSphere に Real-Time Decision Server をインストールする場合は、JDK を個別にダウンロードする必要はありません。WebSphere には、専用の JDK が同梱されています。ただし、Oracle RTD クライアント・ツール (Decision Studio、Load Generator、SDDDBTool など) をインストールする Windows コンピュータに、Sun JDK バージョン 1.5.0.8 以降をインストールする必要があります。

注意: Linux 以外の WebSphere にインストールする場合は、WebSphere Fix Pack 3 をインストールして、WebSphere に同梱された JDK を更新する必要があります。詳細は、第1.4.2項を参照してください。

1.3.4 WebLogic へのインストール

Windows の WebLogic に Real-Time Decision Server をインストールする場合は、JDK を個別にダウンロードする必要はありません。WebLogic には、専用の JDK が同梱されています。ただし、`JAVA_HOME` システム環境変数を設定し、WebLogic が使用する JDK のインストール場所のフルパス名に値を設定する必要があります。

Solaris または **Linux** の WebLogic に Real-Time Decision Server をインストールする場合は、JDK を個別にダウンロードする必要はありません。WebLogic には、専用の JDK が同梱されています。ただし、Oracle RTD クライアント・ツール (Decision Studio、Load Generator、SDDDBTool など) をインストールする Windows コンピュータに、Sun JDK バージョン 1.5.0.8 以降をインストールする必要があります。

HP-UX の WebLogic に Real-Time Decision Server をインストールする場合は、Oracle RTD をインストールする前に、HP-UX JDK 5.0.05 をインストールする必要があります。その後で、Oracle RTD クライアント・ツール (Decision Studio、Load Generator、SDDDBTool など) をインストールする Windows コンピュータに、Sun JDK バージョン 1.5.0.8 以降をインストールします。

AIX ベースのシステム の WebLogic に Real-Time Decision Server をインストールする場合は、Oracle RTD をインストールする前に、IBM Java 5 SR2 をインストールする必要があります。その後で、Oracle RTD クライアント・ツール (Decision Studio、Load Generator、SDDDBTool など) をインストールする Windows コンピュータに、Sun JDK バージョン 1.5.0.8 以降をインストールします。



注意: UNIX システムに WebLogic をインストールする場合は、次に示すように、PATH オペレーティング・システム環境変数の先頭に、適切な JDK の bin ディレクトリを追加する必要があります。

```
PATH=JAVA_HOME/bin:$PATH
export PATH
```

1.4 アプリケーション・サーバーのインストール

Real-Time Decision Server は、OC4J (Oracle Application Server のコンポーネント)、WebSphere、WebLogic の各アプリケーション・サーバーで実行されます。サポートされるアプリケーション・サーバーのバージョンの詳細は、『System Requirements and Supported Platforms for Oracle Real-Time Decisions』を参照してください。

1.4.1 Oracle RTD 用の OC4J のインストール

スタンドアロン・バージョンの OC4J と Oracle Application Server に同梱されたバージョンの OC4J のどちらでもインストールできます。スタンドアロン・バージョンの OC4J には、インストーラの付属しない小さなパッケージに、シングル・サーバーが用意されています。Oracle Application Server に同梱されたバージョンの OC4J では、フルスイートなインストーラが付属し、J2EE クラスターリングがサポートされます。

1.4.1.1 スタンドアロン OC4 のインストール

- 1 Real-Time Decision Server を実行するコンピュータに、スタンドアロン OC4J をインストールします。スタンドアロン OC4J のインストールの詳細は、『Oracle Containers for J2EE 構成および管理ガイド』を参照してください。

注意: このドキュメントでは、スタンドアロン OC4J インスタンスのインストール先ディレクトリを `OC4J_HOME` と記述しています。たとえば、スタンドアロン OC4J ファイルを `C:\Program Files` に解凍した場合、`OC4J_HOME` は `C:\Program Files\OC4J` になります。

- 2 ファイル `OC4J_HOME\bin\oc4j.cmd` (UNIX の場合は `OC4J_HOME/bin/oc4j`) のバックアップ・コピーを作成します。
- 3 `OC4J_HOME\bin\oc4j.cmd` (UNIX の場合は `OC4J_HOME/bin/oc4j`) を開いて編集します。ファイルの先頭に、次の行を追加します。

```
set ORACLE_HOME=your_OC4J_HOME
```

次に例を示します。

```
set ORACLE_HOME=C:\OracleBI\OC4J
```

保存してファイルを閉じます。

- 4 `oc4j` スクリプトを使用して OC4J を起動します。これを行うには、コマンド・プロンプトを開き、`OC4J_HOME\bin` (UNIX の場合は `OC4J_HOME/bin`) に移動して `oc4j -start` と入力します。
- 5 要求されたら、新しいパスワードを入力および確認します。このパスワードは、OC4J インスタンスの管理アカウントである `oc4jadmin` アカウントのパスワードとなります。このパスワード入力が必要なのは OC4J の初回起動時のみで、以降の OC4J 起動時には入力は不要です。
- 6 `oc4j` スクリプトを使用して OC4J を停止します。これを行うには、コマンド・プロンプトを開き、`OC4J_HOME\bin` (UNIX の場合は `OC4J_HOME/bin`) に移動して次のコマンドを発行します。OC4J が使用する ORMI ポート (デフォルトでは 23791) と、`oc4jadmin` アカウントのパスワードを入力する必要があります。

```
oc4j -shutdown -port oc4jOrmiPort -password oc4jadmin_password
```

次に例を示します。

```
oc4j -shutdown -port 23791 -password mypwd
```

1.4.1.2 Oracle Application Server のインストール

Real-Time Decision Server を実行するコンピュータに、Oracle Application Server をインストールします。Oracle Application Server のインストールの詳細は、Oracle Application Server のインストール・ガイドを参照してください。

注意: このドキュメントでは、Oracle Application Server のインストール先ディレクトリを `ORACLE_AS_HOME` と記述しています。たとえば、Oracle Application Server を `C:\Oracle` にインストールした場合、`ORACLE_AS_HOME` は `C:\Oracle\10.1.3.1\OracleAS_1` のようになります。

1.4.2 Oracle RTD 用の WebSphere のインストール

- 1 Real-Time Decision Server を実行するコンピュータに、WebSphere をインストールします。WebSphere のインストールの詳細は、WebSphere のドキュメントを参照してください。インストール時には、ローカル・コンピュータに管理ユーザーとしてログインする必要があります。

注意: このドキュメントでは、WebSphere のインストール先ディレクトリを `WEBSPHHERE_HOME` と記述しています。たとえば、Windows では、デフォルトの `WEBSPHHERE_HOME` は `C:\Program Files\IBM\WebSphere` です。

- 2 WebSphere のインストールを V6.1.0 Fix Pack 3 に更新します。必要な更新ファイルは、<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg24013830> でダウンロードできます。

Linux 以外のすべてのプラットフォームでは、**AppServer** と **Java SDK** 両方のコンポーネントに対する更新をインストールする必要があります。Linux プラットフォームでは、**AppServer** コンポーネントに対する更新のみをインストールし、**Java SDK** に対する更新はインストールしないでください。Linux に **Java SDK** の更新をインストールした場合、WebSphere サーバーが起動しないことがあります。

1.4.3 Oracle RTD 用の WebLogic のインストール

- 1 UNIX に WebLogic をインストールする場合は、`PATH` オペレーティング・システム変数の先頭に、適切な JDK の `JAVA_HOME/bin` ディレクトリを追加する必要があります。WebLogic に対する JDK の要件の詳細は、第1.3.4項「WebLogic へのインストール」を参照してください。

- 2 Real-Time Decision Server を実行するコンピュータに、WebLogic をインストールします。WebLogic のインストールの詳細は、WebLogic のドキュメントを参照してください。

注意: このドキュメントでは、WebLogic のインストール先ディレクトリを `BEA_HOME` と記述しています。たとえば、WebLogic を `C:\` にインストールした場合、`BEA_HOME` は `C:\bea` のようになります。

- 3 WebLogic を Windows にインストールした場合は、Oracle RTD 用に使用する WebLogic ドメインを設定するときに、Configuration Wizard の「Configure Server Start Mode and JDK」画面で「Sun SDK」を選択する必要があります。「JRockit SDK」は選択しないでください。

WebLogic ドメインの作成の詳細は、WebLogic のドキュメントを参照してください。WebLogic に対する JDK の要件の詳細は、第1.3.4項「WebLogic へのインストール」を参照してください。

- 4 既存の WebLogic ドメインを Oracle RTD 用に使用する場合、それが不適切な JDK をポイントしているときは、正しい JDK が使用されるように WebLogic の構成を変更する必要があります。

1.5 インストール前のその他のタスク

Oracle RTD のインストールを開始する前に、開いているアプリケーションをすべて終了する必要があります。特に、Oracle RTD のデフォルトのポートであるポート 8080 を使用するアプリケーションとプロセスは、すべて停止してください。

また、Google Desktop など、リアルタイムに索引付けを行うソフトウェアが実行されている場合は、Oracle RTD のインストールを開始する前にすべて停止してください。Real-Time Decision Server を実行中のコンピュータでリアルタイムの索引付けソフトウェアが実行されていると、Oracle RTD のパフォーマンスが低下する可能性があります。

2 Real-Time Decision Server のインストール

Real-Time Decision Server は、インストールしたアプリケーション・サーバーが常駐するコンピュータにインストールする必要があります。

この章の内容は次のとおりです。

- 2.1 Oracle Real-Time Decisions のファイルのインストール
- 2.2 Oracle RTD Database の作成と初期化
- 2.3 CrossSell サンプル・データの移入 (オプション)
- 2.4 Oracle Real-Time Decisions での SSL の使用

2.1 Oracle Real-Time Decisions のファイルのインストール

Real-Time Decision Server は、Windows と UNIX のどちらでも実行できます。Decision Studio、Load Generator、SDDDBTool など、Oracle RTD クライアント・ツールは、Windows でのみ実行できます。

2.1.1 単一の Windows コンピュータへの Oracle RTD のインストール

単一の Windows コンピュータに Oracle RTD のコンポーネントをすべてインストールするには、次の手順を実行します。

- 1 Oracle RTD のファイルをインストールするディレクトリに、適切な Oracle RTD アーカイブ・ファイルを解凍します。使用しているオペレーティング・システムとアプリケーション・サーバー・タイプに対応したアーカイブ・ファイルを、間違いなく選択してください。

Oracle RTD のインストール DVD では、Oracle RTD アーカイブ・ファイルには、オペレーティング・システム・プラットフォームとアプリケーション・サーバー・タイプに応じて、異なる名前が付いています。たとえば、Windows の WebLogic 用の Oracle RTD アーカイブ・ファイルの名前は `rtd_2.2_WL_win.zip` です。

ファイルを展開すると、`OracleBI¥RTD` というディレクトリが作成され、Oracle RTD のファイルがすべてそこに格納されます。このディレクトリが `RTD_HOME` になります。

- 2 WebLogic を使用している場合、Oracle RTD アプリケーションの `.ear` ファイルは、先に展開しておかないと、WebLogic アプリケーション・サーバーにデプロイできません。これを行うには、次の手順を実行します。
 - a. `RTD_HOME¥package` に移動して、`RTD.ear` ファイルの内容の格納に使用するディレクトリを `expanded` という名前で作成します。
 - b. WinZip を使用して、`RTD_HOME¥package¥expanded` ディレクトリに `RTD_HOME¥package¥RTD.ear` ファイルを展開します。

2.1.2 UNIX への Real-Time Decision Server のインストール

Real-Time Decision Server を UNIX コンピュータにインストールし、Oracle RTD のコンポーネントを Windows コンピュータにインストールするには、次の手順を実行します。

- 1 UNIX に Real-Time Decision Server をインストールするには、`cpio` を使用して、Real-Time Decision Server のファイルをインストールする UNIX コンピュータのディレクトリに、適切なアーカイブ・ファイルを展開します。使用しているオペレーティング・システムとアプリケーション・サーバー・タイプに対応したアーカイブ・ファイルを、間違いなく選択してください。

Oracle RTD のインストール DVD では、Oracle RTD アーカイブ・ファイルには、オペレーティング・システム・プラットフォームとアプリケーション・サーバー・タイプに応じて、異なった名前が付いています。たとえば、UNIX の WebSphere 用の Oracle RTD アーカイブ・ファイルの名前は `rtd_2.2_WAS_unix.cpio` です。

ファイルを展開すると、`OracleBI/RTD` というディレクトリが作成され、Real-Time Decision Server のファイルがすべてそこに格納されます。このディレクトリが、UNIX コンピュータの `RTD_HOME` になります。

- 2 WebLogic を使用している場合、Oracle RTD アプリケーションの `.ear` ファイルは、先に展開しておかないと、WebLogic アプリケーション・サーバーにデプロイできません。これを行うには、次の手順を実行します。
 - a. `RTD_HOME/package` に移動して、`RTD.ear` ファイルの内容の格納に使用するディレクトリを `expanded` という名前で作成します。
 - b. `jar` ユーティリティを使用して、`RTD_HOME/package/expanded` ディレクトリに `RTD_HOME/package/RTD.ear` ファイルを展開します。
- 3 Windows に Oracle RTD クライアント・ツールをインストールするには、Oracle RTD クライアント・ツールをインストールするディレクトリに、適切な Oracle RTD アーカイブ・ファイルを解凍します。使用しているオペレーティング・システムとアプリケーション・サーバー・タイプに対応したアーカイブ・ファイルを、間違いなく選択してください。

Oracle RTD のインストール DVD では、Oracle RTD アーカイブ・ファイルには、オペレーティング・システム・プラットフォームとアプリケーション・サーバー・タイプに応じて、異なった名前が付いています。たとえば、Windows の OC4J (および Oracle Application Server) 用の Oracle RTD アーカイブ・ファイルの名前は `RTD_win_oc4j_2.2.zip` です。

ファイルを展開すると、`OracleBI¥RTD` というディレクトリが作成され、Oracle RTD のファイルがすべてそこに格納されます。このディレクトリが、Windows コンピュータの `RTD_HOME` になります。

2.1.3 Oracle RTD のランタイム環境について

このドキュメントでは、`RTD_HOME` は、Oracle RTD のファイルがインストールされたディレクトリを表します。ディレクトリ構造は Oracle RTD のファイルの展開時に作成されるため、このパスの最後の部分は必ず `OracleBI/RTD` になります。たとえば、Windows の `C:¥` に RTD ファイルを展開した場合、`RTD_HOME` は `C:¥OracleBI¥RTD` になります。

Oracle RTD のログの表示など、Oracle RTD の一部の管理タスクでは、`RTD_RUNTIME_HOME` の場所の指定が必要になります。`RTD_RUNTIME_HOME` の場所は、使用するアプリケーション・サーバーによって異なります。

- スタンドアロン OC4J を使用する場合、`RTD_RUNTIME_HOME` は `OC4J_HOME/j2ee/home/applications/OracleRTD` です。
- Oracle Application Server を使用する場合、`RTD_RUNTIME_HOME` は `ORACLE_AS_HOME/j2ee/oc4j_instance/applications/OracleRTD` です。

注意: `oc4j_instance` という値は、Oracle RTD がデプロイされている特定の OC4J インスタンスを表します。たとえば、Oracle RTD がデフォルトの OC4J インスタンスにデプロイされている場合、このパスは `ORACLE_AS_HOME/j2ee/home/config/default-web-site.xml` になります。

- WebSphere を使用する場合、`RTD_RUNTIME_HOME` は `WEBSPHERE_HOME/AppServer/profiles/profile_name/installedApps/cell_name/OracleRTD.ear` です。
- WebLogic を使用する場合、`RTD_RUNTIME_HOME` は `RTD.ear` ファイルの内容を格納するディレクトリです。たとえば、WebLogic を使用していて `RTD_HOME` が `C:\¥OracleBI¥RTD` の場合、`RTD_RUNTIME_HOME` は `C:\¥OracleBI¥RTD¥package¥expanded` です。

2.2 Oracle RTD Database の作成と初期化

Oracle RTD のファイルのインストールが完了したら、Oracle RTD Database を作成および初期化する必要があります。

Oracle RTD には、管理ユーザーとランタイム・ユーザーという 2 種類のデータベース・ユーザーが必要です。管理ユーザーは、テーブルおよびストアド・プロシージャを作成できる必要があります。Oracle RTD の初回構成時にデータベースを初期化するときにのみ使用されます。Oracle RTD の実行時に管理ユーザーが使用されることはありません。データベースのランタイム・ユーザーは、実行時に Oracle RTD によって、データベースの情報へのアクセスに使用されます。

この後の各項では、データベースの作成についてデータベース・タイプ別に説明し、さらに Oracle RTD Database の初期化について説明します。

2.2.1 データベースの作成: Microsoft SQL Server

SQL Server のインストール後に、SQL Server Enterprise Manager を使用して、新しいデータベースを作成します。

Oracle RTD Database として SQL Server を使用するには、管理ユーザーとランタイム・ユーザーという 2 種類のデータベース・ユーザーを、このデータベースに割り当てる必要があります。管理ユーザーには、サーバー・ロールの `System Administrator` が必要です。ランタイム・ユーザーには、データベース・ロールの `db_ddladmin`、`db_reader` および `db_writer` が必要です。データベースの作成およびユーザーの割当ての詳細は、Microsoft SQL Server のドキュメントを参照してください。

SDDBToolの実行を計画しているコンピュータには、SQL Server Client Tools をインストールする必要があります。SDDBToolの詳細は、第2.2.4項を参照してください。

2.2.2 データベースの作成: Oracle Database

Oracle RTD Database として Oracle Database を使用するには、ネット・サービス名と、有効なユーザー名およびパスワードを指定する必要があります。ネット・サービスの設定方法の詳細は、Oracle Database のドキュメントを参照してください。

Oracle Database の場合は、管理ユーザーとランタイム・ユーザーを同じユーザーにする必要があります。このユーザーには、データベース・ロールの Connect および Resource が必要です。Resource ロールは、データベースの初期化後に取り消すことができます。Resource ロールを取り消す場合は、ランタイム・ユーザー用の表領域の割当て制限を、データベースのポリシーに応じて適切なレベルに設定するか unlimited に設定します。

SDDBToolの実行を計画しているコンピュータには、Oracle Administrator Client をインストールする必要があります。SDDBToolの詳細は、第2.2.4項を参照してください。

2.2.3 データベースの作成: IBM DB2

DB2 Server のインストール後に、DB2 Control Center を使用して、新しいデータベースを作成します。このデータベースのコード・セットは、UTF-8 に設定する必要があります。

Oracle RTD Database として DB2 を使用するには、管理ユーザーとランタイム・ユーザーという2種類のデータベース・ユーザーを指定する必要があります。管理ユーザーには、作成したデータベースに対する Database Administrator 権限が必要です。ランタイム・ユーザーには、Connect to database 権限が必要です。

データベースおよびユーザーの作成の詳細は、DB2 のドキュメントを参照してください。

SDDBToolの実行を計画しているコンピュータには、DB2 Administrative Client をインストールする必要があります。SDDBToolの詳細は、次の項を参照してください。

2.2.4 SDDBTool による Oracle RTD Database の初期化

Oracle RTD Windows クライアントのアプリケーション SDDBTool を使用して、Oracle RTD Database を初期化します。SDDBTool を使用してデータベースを初期化する前に、Real-Time Decision Server が実行中でないことを確認してください。

SDDBTool を使用して Oracle RTD Database を初期化するには、次の手順を実行します。

- 1 データベース・サーバーが実行中であることを確認します。
- 2 Oracle RTD Database として Oracle Database を使用している場合は、Oracle RTD クライアント・ツールをインストールしたコンピュータで、ファイル `RTD_HOME¥scripts¥sql¥Oracle¥SDTablespaceMap.txt` を編集します。このファイルによって、Oracle 表領域への Real-Time Decision Server のテーブルの割当てを制御できます。詳細は、ファイルに記載されている指示を参照してください。ファイルを編集したら、ファイルを保存して閉じます。

- 3 Oracle RTD クライアント・ツールをインストールしたコンピュータで、
`RTD_HOME¥scripts¥SDDDBTool.cmd` を実行して SDDDBTool を起動します。次にデータベースのタイプを選択します。
- 4 次のデータベース設定を入力します。
 - データベースのホスト
 - データベースのポート
 - データベース名（Oracle Database の場合は、データベース名でなく SID を入力します）
 - データベースのランタイム・ユーザー
 - データベースの管理ユーザー
 - データベースの管理ユーザーのパスワード



注意: ここで指定するデータベースの管理ユーザーには、データベースに対してテーブルおよびストアド・プロシージャを作成する権限が必要です。ランタイム・ユーザーは、実行時にシステム・データへのアクセスに使用します。

Oracle Database の場合は、データベースのランタイム・ユーザーと管理ユーザーは同じユーザーになります。

- 5 「Next」をクリックします。
- 6 「Initialize」または「Upgrade」を選択します。「Initialize」を選択すると、Oracle RTD の実行に必要なデータおよびデータソースが作成されます。「Upgrade」を選択すると、以前のバージョンから現行バージョンにアップグレードされます。

Oracle RTD Database に Oracle Database を使用しているときに、サンプルの CrossSell インライン・サービスを使用する予定がない場合は、データベースの初期化の完了後に、データベース・ユーザーから Resource ロールを取り消すことができます。サンプルの CrossSell インライン・サービスの詳細は、次の項を参照してください。

2.3 CrossSell サンプル・データの移入（オプション）

Oracle RTD には、CrossSell というサンプルのインライン・サービスが付属しています。このサンプル・インライン・サービスを使用するには、Oracle RTD Database に CrossSellCustomers と CrossSellResponses という 2 つのテーブルを作成して、データを移入する必要があります。これを行うには、Oracle RTD クライアント・ツールをインストールした Windows コンピュータで、スクリプト InitAppDB を実行します。InitAppDB は、サンプル・インライン・サービスと同じ場所にあります。コマンド・プロンプトを使用して、データベースのタイプに該当するスクリプトを実行します。

- Oracle RTD Database に SQL Server を使用している場合は、
`RTD_HOME¥examples¥CrossSell¥etc¥data¥SQLServer¥InitAppDB` を実行します。

- Oracle RTD Database に Oracle Database を使用している場合は、
`RTD_HOME¥examples¥CrossSell¥etc¥data¥Oracle¥InitAppDB` を実行します。
- Oracle RTD Database に DB2 を使用している場合は、
`RTD_HOME¥examples¥CrossSell¥etc¥data¥DB2¥InitAppDB` を実行します。

このスクリプトには、次のパラメータが必要です。

```
InitAppDB RTD_HOME db_host db_port db_name db_runtime_user
db_admin_user db_admin_password
```

各パラメータの内容は次のとおりです。

`RTD_HOME` は、Oracle RTD のファイルがインストールされているディレクトリのフルパスです。

`db_host` は、データベース・サーバーをホスティングするコンピュータの名前です。

`db_port` は、データベースのポート番号です。

`db_name` はデータベースの名前です。Oracle Database の場合は SID です。

`db_runtime_user` は、システムのランタイム・ユーザーのユーザー名です。

`db_admin_user` は、データベースに対してテーブルおよびストアド・プロシージャを作成する権限を持つユーザーの名前です。

`db_admin_password` は、管理ユーザーのパスワードです。

注意: Oracle Database の場合は、`db_runtime_user` と `db_admin_user` は同じユーザーになります。

Oracle Database を Oracle RTD Database に使用する場合は、`InitAppDB` スクリプトの実行後に、データベース・ユーザーから Resource ロールを取り消すことができます。

2.4 Oracle Real-Time Decisions での SSL の使用

Oracle RTD へのクライアント接続に SSL を使用し（推奨）、そこで Oracle RTD のデフォルトのキーストアおよびトラストストアを使用する場合は、キーストアとトラストストアのパスワードを最初に変更する必要があります。これを行うには、次の手順を実行します。

- 1 次の `keytool` コマンドを実行して、キーストアのパスワードを変更します。

```
JAVA_HOME/bin/keytool -storepasswd -new new_password -keystore
RTD_HOME/etc/ssl/sdserver.keystore
```

`RTD_HOME` は、実際の Oracle RTD のインストール・パスに置き換えてください。
`new_password` の箇所には、Oracle RTD のキーストアに使用する新しいパスワードを指定します。

キーストアの既存のパスワードが要求されたら、`tc-ssl` と入力します。

- 2 次の `keytool` コマンドを実行して、トラストストアのパスワードを変更します。

```
JAVA_HOME/bin/keytool -storepasswd -new new_password -keystore  
RTD_HOME/etc/ssl/sdtrust.store
```

`RTD_HOME` は、実際の Oracle RTD のインストール・パスに置き換えてください。
`new_password` の箇所には、Oracle RTD のトラストストアに使用する新しいパスワードを指定します。

トラストストアの既存のパスワードが要求されたら、`tc-ssl` と入力します。

- 3 次の `keytool` コマンドを実行して、サーバーの秘密鍵のパスワードを変更します。

```
JAVA_HOME/bin/keytool -keypasswd -alias tc-ssl -keypass tc-ssl  
-new new_password -keystore RTD_HOME/etc/ssl/sdserver.keystore
```

`RTD_HOME` は、実際の Oracle RTD のインストール・パスに置き換えてください。
`new_password` の箇所には、サーバーの秘密鍵に使用するパスワードを指定します。

キーストアのパスワードが要求されたら、手順 1 で指定したキーストアのパスワードを入力します。

3 Oracle Real-Time Decisions 用の OC4J の構成

Oracle RTD は、Oracle Application Server の Oracle Containers for J2EE (OC4J) コンポーネントで実行されます。スタンドアロン・バージョンの OC4J と Oracle Application Server に同梱されたバージョンの OC4J のどちらでも使用できます。

Oracle RTD は、UNIX と Windows どちらのプラットフォームの OC4J でもサポートされます。次の手順では、OC4J に Real-Time Decision Server をインストールする方法について説明します。



注意: Real-Time Decision Server は UNIX でも Windows でも実行されますが、Oracle RTD クライアント・ツールは、Windows プラットフォームでしか実行できません。

この章の内容は次のとおりです。

- 3.1 サーバー・プロパティの構成
- 3.2 Oracle RTD Database の JDBC プロバイダの作成
- 3.3 OC4J への Oracle Real-Time Decisions アプリケーションのインストール
- 3.4 Oracle Real-Time Decisions の起動
- 3.5 OC4J からの Oracle Real-Time Decisions アプリケーションのアンインストール
- 3.6 Real-Time Decision Server の SSL の構成 (推奨)
- 3.7 OC4J に対する JConsole の設定
- 3.8 Oracle Real-Time Decisions のポート番号の変更

3.1 サーバー・プロパティの構成

この後の項の手順を実行して、Oracle RTD に対して OC4J を構成します。作業を開始する前に、OC4J が稼動していないことを確認してください。

3.1.1 スタンドアロン OC4J のサーバー・プロパティの構成

- 1 使用しているオペレーティング・システムに対応する手順を実行します。
 - a. **Windows の場合:** ファイル `OC4J_HOME¥bin¥oc4j.cmd` を探し、それを開いて編集します。:oc4j 行の後に、次の 4 行を新しく挿入します。
`jmx_remote_port` には、JConsole のポート番号として使用する値を入力します (例、12345)。

```
set JVMARGS=%JVMARGS% -Dcom.sun.management.jmxremote=true
set JVMARGS=%JVMARGS%
-Dcom.sun.management.jmxremote.port=jmx_remote_port
set JVMARGS=%JVMARGS% -Dcom.sun.management.jmxremote.
authenticate=false
set JVMARGS=%JVMARGS% -Dcom.sun.management.jmxremote.ssl=false
```

保存してファイルを閉じます。

ヒント: メモリー不足のようなエラーを減らすには、この時点で JVM メモリーを増やしておきます。これを行うには、oc4j.cmd ファイルに次のような行を追加します。

```
set JVMARGS=%OC4J_JVM_ARGS% -Xms512M -Xmx1024M
```

- b. **UNIX の場合:** ファイル `OC4J_HOME/bin/oc4j` を探し、それを開いて編集します。echo "Starting OC4J from \$J2EE_HOME ..." 行の後に、次の 4 行を新しく挿入します。jmx_remote_port には、JConsole のポート番号として使用する値を入力します (例、12345)。

```
JVMARGS=$JVMARGS -Dcom.sun.management.jmxremote=true
JVMARGS=$JVMARGS -Dcom.sun.management.jmxremote.
port=jmx_remote_port
JVMARGS=$JVMARGS -Dcom.sun.management.jmxremote.
authenticate=false
JVMARGS=$JVMARGS -Dcom.sun.management.jmxremote.ssl=false
```

AIX ベースのシステムに OC4J をインストールした場合は、次の行も追加します。

```
JVMARGS=$JVMARGS -Djava.net.preferIPv4Stack=true
```

保存してファイルを閉じます。

注意: この手順を実行するには、`OC4J_HOME/bin/oc4j` ファイルに対する実行権限の付与が必要になる場合があります。

ヒント: メモリー不足のようなエラーを減らすには、この時点で JVM メモリーを増やしておきます。これを行うには、oc4j ファイルに次のような行を追加します。

```
OC4J_JVM_ARGS=$OC4J_JVM_ARGS -Xms512M -Xmx1024M
```

- 2 ファイル `OC4J_HOME/j2ee/home/config/default-web-site.xml` を探し、その名前を `rtd-web-site.xml` に変更して同じ場所にコピーします。名前変更後のファイルの内容を次のように変更します。

<web-site>タグで、port 属性を 8080 に設定します。

<web-site>タグで、display-name 属性を OC4J 10g (10.1.3) RTD Web Site に設定します。

<web-app>タグをすべて削除します。

<access-log>タグで、path 属性を ../log/rtd-web-access.log に設定します。

次に例を示します。

```
<?xml version="1.0"?>
<web-site xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://xmlns.oracle.com/oracleas/schema
/web-site-10_0.xsd" port="8080" display-name="OC4J 10g (10.1.3) RTD Web
Site" schema-major-version="10" schema-minor-version="0" >
<default-web-app application="default" name="defaultWebApp"
root="/j2ee" />
<access-log path="../log/rtd-web-access.log" split="day" />
</web-site>
```

保存してファイルを閉じます。

- 3 ファイル `OC4J_HOME/j2ee/home/config/server.xml` を探し、それを開いて編集します。`<web-site default="true" path="../default-web-site.xml" />`の後に、次の行を追加します。

```
<web-site path="../rtd-web-site.xml" />
```

保存してファイルを閉じます。

- 4 Oracle RTD Database に SQL Server と DB2 のいずれかを使用する場合は、次のようにして、データベースの JDBC jar ファイルを OC4J の `applib` ディレクトリにコピーします。
 - a. **SQL Server の場合:** ディレクトリ `RTD_HOME/lib/jdbc/`にあるファイル `sqljdbc.jar` を、ディレクトリ `OC4J_HOME/j2ee/home/applib` にコピーします。
 - b. **DB2 の場合:** `RTD_HOME/lib/jdbc/`にあるファイル `db2jcc.jar` および `db2jcc_license_cu.jar` を、ディレクトリ `OC4J_HOME/j2ee/home/applib` にコピーします。

3.1.2 Oracle Application Server のサーバー・プロパティの構成

- 1 ファイル `ORACLE_AS_HOME/opmn/conf/opmn.xml` を探し、それを開いて編集します。`default_group`、`oc4j_instance` (たとえば `home`)、`start-parameters` の各タグの下にある、`java-options` タグの `value` 属性の最後に次の行を追加します。

```
-Dcom.sun.management.jmxremote=true -Dcom.sun.management.jmxremote.
port=jmx_remote_port -Dcom.sun.management.jmxremote.authenticate=
false -Dcom.sun.management.jmxremote.ssl=false
```

これらのテキストは必ず 1 行で入力してください。このドキュメントからテキストをコピーする場合は、改行があったらすべて削除します。`jmx_remote_port` には、JConsole のポート番号として使用する値を入力します (例、12345)。

次に例を示します。

```
<ias-component id="default_group">
  <process-type id="home" module-id="OC4J" status="enabled">
    <module-data>
      <category id="start-parameters">
        <data id="java-options" value="...-Dcom.sun.management.
jmxremote=true -Dcom.sun.management.jmxremote.port=12345 -Dcom.sun.
management.jmxremote.authenticate=false -Dcom.sun.management.
jmxremote.ssl=false "/>
      </category>
    </module-data>
  </process-type>
</ias-component>
```

注意: 前述の例では、`-Dcom.sun.management.jmxremote=true` の前に表示される...は、この例では紹介しないその他の Java オプションのパラメータを表します。

AIX ベースのシステムに Oracle Application Server をインストールした場合は、次の Java オプションも追加します。

```
-Djava.net.preferIPv4Stack=true
```

注意: この Java オプションと前述のオプションの間には、必ず空白を挿入してください。

保存してファイルを閉じます。

- 2 ファイル `ORACLE_AS_HOME/j2ee/oc4j_instance/config/default-web-site.xml` を探し、その名前を `rtd-web-site.xml` に変更して同じ場所にコピーします。

注意: `oc4j_instance` という値は、Oracle RTD をデプロイする予定の特定の OC4J インスタンスを表します。たとえば、Oracle RTD をデフォルトの OC4J インスタンスにデプロイする予定の場合、このパスは

`ORACLE_AS_HOME/j2ee/home/config/default-web-site.xml` になります。

- 3 名前変更したファイルの内容を次のように変更します。

`<web-site>` タグで、`port` 属性を 8080 に設定します。

`<web-site>` タグで、`display-name` 属性を `OC4J 10g (10.1.3) RTD Web Site` に設定します。

`<web-app>` タグをすべて削除します。

`<access-log>` タグで、`path` 属性を `../log/rtd-web-access.log` に設定します。

次に例を示します。

```
<?xml version="1.0"?>
<web-site xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://xmlns.oracle.com/oracleas/schema
/web-site-10_0.xsd" port="8080" display-name="OC4J 10g (10.1.3) RTD Web
Site" schema-major-version="10" schema-minor-version="0" >
<default-web-app application="default" name="defaultWebApp"
root="/j2ee" />
<access-log path="../log/rtd-web-access.log" split="day" />
</web-site>
```

保存してファイルを閉じます。

- 4 ファイル `ORACLE_AS_HOME/j2ee/oc4j_instance/config/server.xml` を探し、それを開いて編集します。`<web-site default="true" path="./default-web-site.xml" />`の後に、次の行を追加します。

```
<web-site path="./rtd-web-site.xml" />
```

保存してファイルを閉じます。

- 5 Oracle RTD Database に SQL Server と DB2 のいずれかを使用する場合は、次のようにして、データベースの JDBC jar ファイルを OC4J の `applib` ディレクトリにコピーします。
 - a. **SQL Server の場合:** ディレクトリ `RTD_HOME/lib/jdbc/`にあるファイル `sqljdbc.jar` を、ディレクトリ `ORACLE_AS_HOME/j2ee/oc4j_instance/applib` にコピーします。
 - b. **DB2 の場合:** `RTD_HOME/lib/jdbc/`にあるファイル `db2jcc.jar` および `db2jcc_license_cu.jar` を、ディレクトリ `ORACLE_AS_HOME/j2ee/oc4j_instance/applib` にコピーします。

3.2 Oracle RTD Database の JDBC プロバイダの作成

Application Server Control という OC4J の管理コンソールを使用して、Oracle RTD Database に JDBC プロバイダを作成します。Application Server Control の使用方法の詳細は、OC4J のドキュメントを参照してください。

先に進む前に、OC4J が稼動していないことを確認してください。

- 1 OC4J サーバーを起動します。スタンドアロン・バージョンの OC4J を使用している場合は、コマンド・プロンプトを開き、`OC4J_HOME/bin` に移動して、コマンド `oc4j -start` を実行します。OC4J を Oracle Application Server の一部として使用している場合は、`opmnctl` を使用して OC4J を管理します。詳細は、『Oracle Containers for J2EE 構成および管理ガイド』を参照してください。
- 2 `oc4jadmin` ユーザーとして Application Server Control にログインします。Application Server Control には、`http://oc4j_host:port/em` からアクセスできます。OC4J のスタンドアロン・バージョンでは、通常、ポート番号は 8888 です。
- 3 「管理」タブに進み、「サービス」の下の「JDBC リソース」タスクをクリックします。「接続プール」の下の「作成」をクリックして、Oracle RTD Database 用の新しい接続プールを作成します。

注意: OC4J を Oracle Application Server の一部として使用している場合は、最初に「グループ」の下の「ホーム」をクリックしてから、「管理」タブに進みます。
- 4 「接続プールの作成 - アプリケーション」ページで、「アプリケーション」に「デフォルト」が選択され、「接続プール・タイプ」に「新規接続プール」が選択されていることを確認します。「続行」をクリックします。
- 5 「接続プールの作成」ページで、「名前」に「RTDConnectionPool」と入力します。

- 6 「コネクション・ファクトリ・クラス」に、次のいずれかを入力します。
 - **SQL Server の場合:** `com.microsoft.sqlserver.jdbc.SQLServerDriver`
 - **Oracle Database の場合:** `oracle.jdbc.driver.OracleDriver`
 - **DB2 の場合:** `com.ibm.db2.jcc.DB2Driver`

注意: Oracle RTD Database に Oracle Database を使用する場合は、「コネクション・ファクトリ・クラス」に `oracle.jdbc.driver.OracleDriver` を入力する必要があります。デフォルト値の `oracle.jdbc.pool.OracleDataSource` は使用しないでください。
- 7 「JDBC URL」に、次のいずれかを入力します。
 - **SQL Server の場合:**
`jdbc:sqlserver://db_host:db_port;databaseName=db_name`
 - **Oracle Database の場合:** `jdbc:oracle:thin:@db_host:db_port:sid`
 - **DB2 の場合:** `jdbc:db2://db_host:db_port/db_name`

`db_host` は Oracle RTD Database を実行するサーバーの名前、`db_port` はデータベース・サーバーへの接続に使用するポート番号、`db_name` は Oracle RTD Database インスタンスの名前 (`rtd` など)、`sid` はサーバーで実行される Oracle Database インスタンスを指定する Oracle システム識別子です。
- 8 「資格証明」の下の「ユーザー名」に、Oracle RTD Database インスタンスに対してシステム管理権限を持つデータベース・ユーザーの名前を入力します。次に、対応するパスワードを指定します。クリアテキスト形式のパスワードを指定するか、または間接的パスワードを指定するかの詳細は、『Oracle Application Server Containers for J2EE セキュリティ・ガイド』を参照してください。
- 9 「終了」をクリックします。
- 10 「JDBC リソース」ページの「データソース」で、「作成」をクリックして新しいデータソースを定義します。
- 11 「データ・ソースの作成 - アプリケーションとタイプ」ページで、「アプリケーション」に「デフォルト」が選択され、「データ・ソース・タイプ」に「マネージド・データソース」が選択されていることを確認します。「続行」をクリックします。
- 12 「データ・ソースの作成 - 管理データ・ソース」ページで、「名前」に「RTD_DS」、「JNDI ロケーション」に「jdbc/SDDS」をそれぞれ入力し、「接続プール」で「RTDConnectionPool」を選択します。その他のオプションはデフォルトのままにします。「終了」をクリックします。
- 13 「JDBC リソース」ページの「データソース」テーブルで、RTD_DS データソースについて「接続テスト」をクリックします。次の手順に従って RTD_DS データソースをテストします。
 - a. Oracle RTD Database に Oracle Database を使用する場合は、デフォルトの設定を維持したまま「テスト」をクリックします。
 - b. Oracle RTD Database に SQL Server または DB2 を使用する場合は、「SQL 文」を「`select * from SDAPPS`」に変更して、「テスト」をクリックします。

接続が確立されなかった場合は、OC4J を再起動してデータソースを再度テストします。それでも接続に失敗する場合は、接続プールが正しく設定されているかどうか確認してください。

3.3 OC4J への Oracle Real-Time Decisions アプリケーションのインストール

Application Server Control を使用して、OC4J に Oracle RTD をインストールします。作業を開始する前に、OC4J が起動していることを確認してください。

- 1 `oc4jadmin` ユーザーとして Application Server Control にログインします。Application Server Control には、`http://oc4j_host:port/em` からアクセスできます。
- 2 OC4J のホーム・ページで、「**管理**」タブをクリックします。
注意: OC4J を Oracle Application Server の一部として使用している場合は、最初に「グループ」の下の「**ホーム**」をクリックしてから、「**アプリケーション**」タブに進みます。
- 3 「**デプロイ**」をクリックします。「デプロイ: アーカイブの選択」ページの「アーカイブ」で、アーカイブのある場所を参照し `RTD_HOME/package/RTD.ear` を指定します。次に、「デプロイ・プラン」で、「**新規デプロイ・プランを自動的に作成します。**」が選択されていることを確認し、「**次へ**」をクリックします。
- 4 「デプロイ: アプリケーション属性」ページで、「**アプリケーション名**」に「OracleRTD」と入力し、「**サイトへの Web モジュールのバインド**」に「`rtd-web-site`」を選択します。「**次へ**」をクリックします。
- 5 「デプロイ: デプロイ設定」ページで、「**デプロイ**」をクリックします。デプロイが完了するまで、数分かかる場合があります。URL に `http://server_name:8080/ui` を指定して Decision Center に進み、Oracle RTD が実行中かどうかをチェックします。

3.4 Oracle Real-Time Decisions の起動

OC4J を起動すると、Oracle RTD が自動的に起動されます。OC4J と無関係に Oracle RTD を起動するには、Application Server Control を次のように使用します。

- 1 `oc4jadmin` ユーザーとして Application Server Control にログインします。Application Server Control には、`http://oc4j_host:port/em` からアクセスできます。
- 2 「**アプリケーション**」タブに進みます。「アプリケーション」テーブルで、「**OracleRTD**」を選択し、「**起動**」をクリックします。「ステータス」列の上向きの緑の矢印は、アプリケーションが起動されていることを示します。
注意: OC4J を Oracle Application Server の一部として使用している場合は、最初に「グループ」の下の「**ホーム**」をクリックしてから、「**アプリケーション**」タブに進みます。

3.5 OC4JからのOracle Real-Time Decisions アプリケーションのアンインストール

Application Server Control を使用して、OC4J から Oracle RTD をアンインストールできます。作業を開始する前に、OC4J が起動していることを確認してください。

- 1 `oc4jadmin` ユーザーとして Application Server Control にログインします。
Application Server Control には、`http://oc4j_host:port/em` からアクセスできます。
- 2 「**アプリケーション**」タブに進みます。「アプリケーション」テーブルで、「**OracleRTD**」を選択し、「**停止**」をクリックします。「確認」ページで、「**はい**」をクリックします。「ステータス」列の下向きの赤の矢印は、アプリケーションが停止されていることを示します。
注意: OC4J を Oracle Application Server の一部として使用している場合は、最初に「グループ」の下の「**ホーム**」をクリックしてから、「**アプリケーション**」タブに進みます。
- 3 「**アンデプロイ**」をクリックし、「アプリケーションのアンデプロイ」ページで「**はい**」をクリックします。

この手順では、OC4J から Oracle RTD がアンインストールされますが、オペレーティング・システムから Oracle RTD のファイルが削除されるわけではありません。Oracle RTD のファイルは、手動で削除する必要があります。

3.6 Real-Time Decision Server の SSL の構成 (推奨)

この項に記述する手順に従って、Real-Time Decision Server へのすべてのクライアント接続に対して SSL を設定します。作業を開始する前に、第2.4項の手順に従って、Oracle RTD のデフォルトのキーストアのパスワードを変更したことを確認します。

注意: 独自のキーストアを使用する場合は、第2.4項の手順を実行する必要はありません。

- 1 スタンドアロン OC4J を使用する場合は、使用しているオペレーティング・システムに対応する手順を実行します。Oracle Application Server を使用する場合は、手順2に進みます。
 - a. **Windows の場合:** ファイル `OC4J_HOME\bin\oc4j.cmd` を探し、それを開いて編集します。各 `%JAVA_HOME%\bin\java` 行の前に、次の行を新しく挿入します。

```
cd %J2EE_HOME%
```
 - b. **UNIX の場合:** ファイル `OC4J_HOME/bin/oc4j.sh` を探し、それを開いて編集します。各 `$JAVA_HOME/bin/java` 行の前に、次の行を新しく挿入します。

```
cd $J2EE_HOME
```
- 2 スタンドアロン OC4J を使用する場合は、`OC4J_HOME/jlib` に移動します。Oracle Application Server を使用する場合は、`ORACLE_AS_HOME/jlib` に移動します。ファイル `jssl-1_1.jar` の名前を `jssl-1_1.jar.notused` に変更します。

- 3 スタンドアロン OC4J を使用する場合は、`OC4J_HOME/j2ee/home/config` に移動します。Oracle Application Server を使用する場合は、`ORACLE_AS_HOME/j2ee/oc4j_instance/config` に移動します。ファイル `rtd-web-site.xml` のコピーを作成し、コピーしたファイルの名前を `rtd-secure-web-site.xml` に変更します。
- 4 編集のために `rtd-secure-web-site.xml` を開き、次のように変更します。
 - a. `<web-site>` タグで、`port` 属性を 8443 に設定します。
 - b. `<web-site>` タグに `secure` 属性を追加して `true` に設定します。`secure` 属性が存在している場合は `true` に設定します。
 - c. `<web-site>` タグで、`display-name` 属性を OC4J 10g (10.1.3) RTD Secure Web Site に設定します。
 - d. 各 `<web-app>` タグに `shared` 属性を追加して `true` に設定します。`shared` 属性が存在している場合は `true` に設定します。
 - e. `<access-log>` タグで、`path` 属性を `../log/rtd-secure-web-access.log` に設定します。
 - f. `</web-site>` タグの直前に次の行を追加します。`RTD_HOME` は、実際の Oracle RTD のインストール・パスに置き換えてください。

```
<ssl-config keystore="RTD_HOME/etc/ssl/sdserver.keystore"
keystore-password="your_keystore_password"
needs-client-auth="false" />
```

次に例を示します。

```
<?xml version="1.0" ?>
<web-site xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://xmlns.oracle.com/oracleas/schema/web-site-10_0.xsd" port="8443" secure="true" display-name="OC4J 10g
(10.1.3) Secure Default Web Site" schema-major-version="10"
schema-minor-version="0" >

<default-web-app application="default" name="defaultWebApp" />
  <web-app application="OracleRTD" name="ls" load-on-startup="true"
root="/ls" shared="true" />
  <web-app application="OracleRTD" name="rtis" load-on-startup="true"
root="/rtis" shared="true" />
  <web-app application="OracleRTD" name="soap" load-on-startup="true"
root="/soap" shared="true" />
  <web-app application="OracleRTD" name="ui" load-on-startup="true"
root="/ui" shared="true" />
  <web-app application="OracleRTD" name="schema" load-on-startup="true"
root="/schema" shared="true" />
  <web-app application="OracleRTD" name="cactus" load-on-startup="true"
root="/cactus" shared="true" />
  <access-log path="../log/rtd-secure-web-access.log" split="day" />
<ssl-config keystore="RTD_HOME/etc/ssl/sdserver.keystore"
keystore-password="my_password" needs-client-auth="false" />
</web-site>
```

Oracle RTD のデフォルトのキーストアではなく独自のキーストアを使用する場合は、`RTD_HOME/etc/ssl/sdserver.keystore` の箇所に、独自のキーストアのパスを指定します。

- 5 ファイル `OC4J_HOME/j2ee/home/config/server.xml` を探し、それを開いて編集します。行 `<web-site default="true" path="./default-web-site.xml" />` の後に、次の行を新しく挿入します。
`<web-site path="./rtd-secure-web-site.xml" />`
- 6 OC4J を再起動します。



注意: 真にセキュアな環境を実現するには、通常の HTTP ポートを無効にして、すべてのクライアント接続が SSL ポートを經由するように設定する必要があります。これを行うには、次の手順を実行します。

1. JConsole を使用して、「OracleRTD」→「SDPropertyManager」→「Misc MBean」を選択し、**WebServerPort** 属性を HTTP ポートから SSL ポートに変更します。次に、**HTTPSEnabled** 属性を `true` に設定します。詳細は、第10.1項「JConsole へのアクセス」を参照してください。
2. アプリケーション・サーバーのツールを使用して、Web サーバーの HTTP ポートを無効にします。詳細は、スタンドアロン OC4J または Oracle Application Server のドキュメントを参照してください。

- 7 独自のキーストアおよびトラストストアを使用する場合は、次の追加手順を実行して、Decision Studio および Load Generator に対して SSL を有効にします。Oracle RTD のデフォルトのキーストアおよびトラストストアを使用する場合は、この手順を実行する必要はありません。
 - a. `RTD_HOME\%eclipse%\eclipse.ini` を開いて編集します。
 - b. `-Djava.net.ssl.trustStore="..\%etc%\ssl\%sdtrust.store"` という行を探します。
 - c. `..\%etc%\ssl\%sdtruststore` を、使用するトラストストア・ファイルのフルパスに置き換えます。
 - d. ファイルを保存して閉じます。
 - e. `RTD_HOME\%scripts%\sdexec.cmd` を開いて編集します。
 - f. ファイルの下部にある、先頭が `%SD_START%` になっている行を探します。ファイルの終わり近くにある、`-Djavax.net.ssl.trustStore="%SD_ROOT%\%etc%\ssl\%sdtrust.store"` という文字列を探します。
 - g. `%SD_ROOT%\%etc%\ssl\%sdtruststore` を、使用するトラストストア・ファイルのフルパスに置き換えます。
 - h. ファイルを保存して閉じます。

3.6.1 SSL 構成のテスト

SSL ポートが正しく機能していることを確認するには、URL に `http://server_name:ssl_port/ui` を指定して Decision Center に進みます。SSL ポートが正しく機能している場合は、ブラウザにログイン画面「Welcome to Decision Center」が表示されます。

Web ブラウザには、「Do you want to accept this certificate?」のようなメッセージが表示される場合もあります。このメッセージが生成されるのは、Oracle RTD のデフォルトのキーストアに同梱されている自己署名証明書について、ブラウザが認識していないことが理由です。この自己署名証明書は、開発環境とテスト環境には適していますが、本番環境には使用しないことをお勧めします。

本番環境では、自己署名証明書を、Verisign や Thawte のような信頼できる認証局 (CA) が発行する証明書に置き換えることをお勧めします。この証明書は、Sun 社のキーツール・ユーティリティで生成された証明書リクエストを CA に送信することによって発行されます。証明書リクエストの生成、および CA からキーストアへの証明書のインポートの手順は、<http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html> を参照してください。

3.7 OC4J に対する JConsole の設定

次の手順を実行して、JConsole 管理ツールを設定します。作業を開始する前に、OC4J が稼動していないことを確認してください。

- 1 スタンドアロン OC4J を使用する場合は、使用しているオペレーティング・システムに対応する手順を実行します。

- a. **Windows の場合:** ファイル `OC4J_HOME\bin\oc4j.cmd` を探し、それを開いて編集します。次の行を探します。

```
set JVMARGS=%JVMARGS% -Dcom.sun.management.jmxremote.  
authenticate=false
```

この行の値を `true` に変更します。保存してファイルを閉じます。

- b. **UNIX の場合:** ファイル `OC4J_HOME/bin/oc4j` を探し、それを開いて編集します。次の行を探します。

```
JVMARGS=$JVMARGS :-Dcom.sun.management.jmxremote.  
authenticate=false
```

この行の値を `true` に変更します。保存してファイルを閉じます。

- 2 Oracle Application Server を使用する場合は、ファイル `ORACLE_AS_HOME/opmn/conf/opmn.xml` を探し、それを開いて編集します。文字列 `-Dcom.sun.management.jmxremote.authenticate=false` を探し、その値を `true` に設定します。保存してファイルを閉じます。
- 3 OC4J が実行されているコンピュータで、JDK のインストール・ディレクトリを探します。OC4J によって使用されている JDK インストールに移動します。
 - a. スタンドアロン OC4J を使用する場合は、ファイル `JDK_HOME/jre/lib/management/jmxremote.password.template` のコピーを作成し、その名前を `jmxremote.password` に変更します。JDK のインストール場所がわからない場合は、オペレーティング・システム環境変数の `JAVA_HOME` をチェックします。
 - b. Oracle Application Server を使用する場合は、ファイル `ORACLE_AS_HOME/jdk/jre/lib/management/jmxremote.password.template` のコピーを作成し、その名前を `jmxremote.password` に変更します。

注意: Oracle Application Server が AIX ベースのシステムで実行されている場合は、第 1.3.2 項で説明しているように、ダウンロードされているのは IBM Java 5 SR2 JDK です。この構成の場合は、`ORACLE_AS_HOME` ではなく、IBM Java 5 SR2 のインストール・ディレクトリに移動します。次に、`jmx.remote.password.template` ファイルにナビゲートします。

- 4 `jmxremote.password` ファイルを開いて編集します。ファイルの末尾に進み、次の 2 行の前にある番号記号 (#) を削除します。

```
monitorRole QED
controlRole R&D
```

これらは、1 行で 1 組の資格証明を表しています。つまり、最初がユーザー名で、その後がそれに対応するパスワードです。新しいユーザー名およびパスワードを、別の行に追加することもできます。デフォルトのユーザー名である `monitorRole` と `controlRole` をそのまま使用する場合は、デフォルトのパスワードを新しいパスワードに変更してください。

デフォルトでは、ユーザー名 `monitorRole` には JConsole MBean の読み取り権限のみが付与され、ユーザー名 `controlRole` には JMX MBean の完全な制御権限が付与されています。この後の手順では、これらのデフォルト権限の変更方法について説明します。

- 5 資格証明ごとに権限を設定するには、同じディレクトリにある `jmxremote.access` ファイルを開いて編集します。次に、キーワードの `readonly` および `readwrite` を使用して、資格証明ごとにアクセス・レベルを指定します。次に例を示します。

```
monitorRole readonly
controlRole readwrite
```

追加した新しい資格証明に対する権限は、必ず `jmxremote.password` ファイルに追加してください。

- 6 `jmxremote.password` ファイルには、パスワードがクリアテキスト形式で含まれているため、このファイルに対するアクセス権限をファイル所有者に制限する必要があります。これを行うには、次のようにしてアクセス権限を変更します。

a. Windows の場合:

<http://java.sun.com/j2se/1.5.0/docs/guide/management/security-windows.html> に記載されている手順に従います。

b. UNIX の場合: コマンド `chmod 600 jmxremote.password` を実行します。

JConsole のセキュリティの詳細は、

<http://java.sun.com/j2se/1.5.0/docs/guide/management/agent.html#PasswordAccess> を参照してください。



重要: `jmxremote.password` ファイルに対するファイル・アクセス権限を変更する必要があります。この手順はスキップしないでください。

- 7 OC4J を起動します。

これで、JConsole を実行し、`jmxremote.password` ファイルに定義されたユーザー名およびパスワードを使用してログインできるようになります。詳細は、第10.1項「JConsole へのアクセス」を参照してください。

3.8 Oracle Real-Time Decisions のポート番号の変更

Oracle RTD アプリケーションのポート番号を変更するには、OC4J と Oracle RTD 両方でポート番号を変更する必要があります。

3.8.1 OC4J でのポート番号の変更

- 1 OC4J を停止します。
- 2 スタンドアロン OC4J を使用する場合は、ファイル `OC4J_HOME/j2ee/home/config/rtd-web-site.xml` を探し、それを開いて編集します。Oracle Application Server を使用する場合は、ファイル `ORACLE_AS_HOME/j2ee/oc4j_instance/config/rtd-web-site.xml` を探し、それを開いて編集します。
- 3 `<web-site>` タグで、`port` 属性を新しいポート番号に設定します。
- 4 Oracle RTD へのクライアント接続に SSL を使用する場合は、`rtd-secure-web-site.xml` に対して手順 2 と 3 を繰り返します。
- 5 OC4J を起動します。

3.8.2 Oracle RTD でのポート番号の変更

- 1 JDK のインストール・ディレクトリから `jconsole` を実行して、JConsole を起動します。詳細は、第10.1項「JConsole へのアクセス」を参照してください。
- 2 「Remote」タブをクリックして、次の内容を入力します。
 - a. 「Host」に、Real-Time Decision Server を実行しているコンピュータの名前を入力します。
 - b. 「Port」に、JMX のリモート・ポート番号（たとえば、12345）を入力します。
 - c. 「User Name」および「Password」に、JConsole の設定時に指定した資格証明を入力します。詳細は、第3.7項「OC4J に対する JConsole の設定」を参照してください。
 - d. 「Connect」をクリックします。
- 3 「MBeans」タブをクリックして、「Oracle RTD」→「SDPropertyManager」→「Misc」MBean に移動します。
- 4 「WebServerPort」を新しいポート番号に変更してから[Enter]を押します。
- 5 JConsole を終了します。
- 6 Oracle RTD を再起動します。

4 Oracle Real-Time Decisions 用の WebSphere の構成

Oracle RTD は、UNIX と Windows どちらのプラットフォームの IBM WebSphere アプリケーション・サーバーでもサポートされます。次の各項では、WebSphere に Real-Time Decision Server をインストールする方法について説明します。



注意: Real-Time Decision Server は UNIX でも Windows でも実行されますが、Oracle RTD クライアント・ツールは、Windows プラットフォームでしか実行できません。

この章の内容は次のとおりです。

- 4.1 サーバー・プロパティの構成
- 4.2 Oracle RTD Database の JDBC プロバイダの作成
- 4.3 WebSphere への Oracle Real-Time Decisions アプリケーションのインストール
- 4.4 Oracle Real-Time Decisions の起動
- 4.5 WebSphere からの Oracle Real-Time Decisions アプリケーションのアンインストール
- 4.6 Real-Time Decision Server の SSL の構成 (推奨)
- 4.7 WebSphere に対する JConsole の設定
- 4.8 Oracle Real-Time Decisions のポート番号の変更

4.1 サーバー・プロパティの構成

Integrated Solutions Console という WebSphere の管理コンソールを使用して、サーバー・プロパティを構成します。Integrated Solutions Console の使用方法の詳細は、WebSphere のドキュメントを参照してください。

- 1 WebSphere を起動します。Windows では、「スタート」→「プログラム」でサーバーを起動できます。UNIX では、
`WEBSPPHERE_HOME/AppServer/profiles/profile_name/bin` に移動して、次のコマンドを実行します。

```
startServer.sh app_server_name -username admin_user  
-password admin_password
```

次に例を示します。

```
startServer.sh AppServer1 -username admin -password mypswd
```

- 2 `http://websphere_host:port/ibm/console` から Integrated Solutions Console にアクセスします。ログイン・プロンプトで、管理者のユーザー名とパスワードを入力します。Windows では、「スタート」→「プログラム」から Integrated Solutions Console にアクセスすることもできます。

Integrated Solutions Console のポート番号が不明な場合は、
`WEBSPPHERE_HOME/AppServer/profiles/profile_name/config/cells/
host_name` にある `virtualhosts.xml` ファイルで確認できます。

- 3 左側のツリーで、「Servers」を開いて「Application servers」を選択します。
- 4 Oracle RTD を実行するアプリケーション・サーバーの名前（たとえば、「server1」）をクリックします。「Server Infrastructure」の下の「Java and Process Management」を開き、「Process Definition」を選択します。
- 5 「Additional Properties」の下の「Java Virtual Machine」をクリックします。
- 6 AIX ベースのシステムに WebSphere をインストールした場合は、次の手順に従って、汎用的な JVM 引数にパラメータを追加します。
 - a. 「General Properties」の下の「Generic JVM arguments」フィールドに、次の文字列を入力します。

```
-Djava.net.preferIPv4Stack=true
```

このフィールドにすでに値が入っている場合は、既存の値の後に空白を追加して、新しい文字列を追加します。

- b. 「OK」をクリックします。

「OK」をクリックすると、「Initial Heap Size」および「Maximum Heap Size」に値を入力する必要があるというエラーが表示される場合があります。このエラーが表示されたら、使用しているシステムでの必要に応じて、これらの値を設定します（たとえば、「Initial Heap Size」を 512 に設定し、「Maximum Heap Size」を 2048 に設定します）。
- 7 「Process Definition」ページで「Java Virtual Machine」をクリックして、「Additional Properties」の下の「Custom Properties」をクリックします。
- 8 「New」をクリックします。
- 9 「Name」に「`org.eclipse.emf.ecore.ERPackage.Registry.INSTANCE`」と入力します。
- 10 「Value」に「`com.sigmadynamics.emf.util.SDEMFRegistry`」と入力します。
- 11 「OK」をクリックして、「Save」をクリックします。
- 12 次のカスタム・プロパティを追加する手順に従って、JConsole MBean の表示を調整します。
 - a. 左側のツリーで、「Servers」を開いて「Application servers」を選択します。
 - b. Oracle RTD を実行するアプリケーション・サーバーの名前（たとえば、「server1」）をクリックします。「Server Infrastructure」の下の「Administration」を開き、「Administration Services」を選択します。

- c. 「Additional Properties」の下の「Custom Properties」をクリックして、「New」をクリックします。
- d. 「Name」に「com.ibm.websphere.mbeans.disableRouting」と入力します。
- e. 「Value」に「<on>OracleRTD:*</on>」と入力します。
- f. 「OK」をクリックして、「Save」をクリックします。

Oracle RTD アプリケーションのデフォルトのポートとして、WebSphere プロファイルのデフォルトの HTTP 転送ポートを使用する場合は、手順13と14をスキップして手順15に進みます。Oracle RTD に新しい HTTP 転送ポートを追加する場合は、手順13に進みます。

- 13 Oracle RTD に新しい HTTP 転送チェーンを追加するには、次の手順を実行します。
 - a. 左側のツリーで、「Servers」を開いて「Application servers」を選択します。
 - b. Oracle RTD を実行するアプリケーション・サーバーの名前（たとえば、「server1」）をクリックします。「Container Settings」の下の「Web container settings」を開き、「Web container transport chains」を選択します。
 - c. 「New」をクリックします。
 - d. 「Transport chain name」に「OracleRTD_chain」と入力します。
 - e. 「Transport chain template」で「WebContainer(templates/chains/webcontainer-chains.xml#Chain_1)」を選択します。「Next」をクリックします。
 - f. 「Port name」に「OracleRTD_port」などの値を入力します。
 - g. 「Host」は、デフォルト値*のままにします。
 - h. 「Port」に、Oracle RTD で使用するポート番号を入力します。デフォルトでは、Oracle RTD アプリケーションはポート 8080 で実行されます。
 - i. 「Next」をクリックし、「Finish」をクリックして、「Save」をクリックします。
- 14 手順13で新しい HTTP 転送ポートを追加した場合は、次の手順に従って、新しいホストのエイリアスを作成します。それ以外の場合は、手順 15に進みます。
 - a. 左側のツリーで、「Environment」を開いて「Virtual Hosts」を選択します。
 - b. 「default_host」をクリックします。「Additional Properties」の下の「Host Aliases」をクリックします。
 - c. 「New」をクリックします。「Host」はデフォルト値*のままにし、「Port」には Oracle RTD のポート番号を入力します。
 - d. 「OK」をクリックして、「Save」をクリックします。
- 15 Oracle RTD アプリケーションのポートとして 8080 を使用していない場合は、Oracle RTD でポート番号を更新する必要があります。詳細は、第4.8.2項「Oracle RTD でのポート番号の変更」を参照してください。第4.2項、第4.3項、第4.4項および第4.7項のタスクを完了するまでは、第4.8.2項で説明する手順を実行することはできません。

16 WebSphere を再起動します。

4.2 Oracle RTD Database の JDBC プロバイダの作成

Integrated Solutions Console を使用して、Oracle RTD Database に JDBC プロバイダを作成します。作業を開始する前に、WebSphere が起動していることを確認してください。

- 1 `http://websphere_host:port/ibm/console` から Integrated Solutions Console にアクセスします。ログイン・プロンプトで、管理者のユーザー名とパスワードを入力します。Windows では、「スタート」→「プログラム」から Integrated Solutions Console にアクセスすることもできます。
- 2 左側のツリーで、「Resources」を開き、「JDBC」を開いて「JDBC Providers」を選択します。
- 3 スコープが正しく（Node=host_nameNode_number に）設定されていることを確認して、「New」をクリックします。
- 4 「Database type」に、「User-defined」を選択します。
- 5 「Implementation class name」に、次のいずれかを入力します。
 - a. **SQL Server の場合:** `com.microsoft.sqlserver.jdbc.SQLServerConnectionPoolDataSource`
 - b. **Oracle Database の場合:** `oracle.jdbc.pool.OracleConnectionPoolDataSource`
 - c. **DB2 の場合:** `com.ibm.db2.jcc.DB2ConnectionPoolDataSource`
- 6 「Name」に「RTDDataProvider」と入力して、「Next」をクリックします。
- 7 「Class path」に、次のいずれかを入力します。
 - a. **SQL Server の場合:** `RTD_HOME/lib/jdbc/sqljdbc.jar`
 - b. **Oracle Database の場合:** `RTD_HOME/lib/jdbc/ojdbc14.jar`
 - c. **DB2 の場合:** `RTD_HOME/lib/jdbc/db2jcc.jar;RTD_HOME/lib/jdbc/db2jcc_license_cu.jar`

RTD_HOME は、実際の Oracle RTD のインストール・パスに置き換えてください。
- 8 「Next」をクリックし、「Finish」をクリックして、「Save」をクリックします。
- 9 「JDBC Providers」ページで、「RTDDataProvider」をクリックします。「Additional Properties」の下で「Data sources」をクリックします。
- 10 「New」をクリックし、「Component-managed authentication alias and XA authentication alias」の下で、「create a new J2C authentication alias」をクリックします。
- 11 「New」をクリックして、次の値を入力します。
 - a. 「Alias」に「RTDDS_auth」と入力します。
 - b. 「User ID」に、データベースのランタイム・ユーザーの名前を入力します。

- c. 「Password」に、そのデータベース・ユーザーに対応するパスワードを入力します。パスワードは空白にできません。
これらの値を入力したら、「OK」をクリックして、「Save」をクリックします。
- 12 ロケータ・リンクを使用して「Data sources」ページに戻り、「New」をクリックします。次の値を入力します。
- a. 「Data source name」に、「RTD_DS」などの値を入力します。
 - b. 「JNDI name」に「SDDS」と入力します。Oracle RTD がデータベースにアクセスできるようにするには、JNDI 名を SDDS にする必要があります。
 - c. 「Component-managed authentication alias and XA authentication」に、「RTDDS_auth」を選択します。
これらの値を入力したら、「Next」をクリックします。
- 13 「Summary」ページで「Finish」をクリックして、「Save」をクリックします。
- 14 作成したデータソースの名前（たとえば、「RTD_DS」）をクリックします。
- 15 Oracle RTD Database に SQL Server を使用する場合は、次の手順に従い、データソースのプロパティを設定します。
- a. 「Additional Properties」の下の「Custom properties」をクリックします。
 - b. 「Select All」アイコンをクリックし、「Delete」をクリックして既存のカスタム・プロパティをすべて削除します。
 - c. 「New」をクリックし、「Name」に「databaseName」と入力して、「Value」にデータベースの名前を入力します。「OK」をクリックします。
 - d. 「New」をクリックし、「Name」に「portNumber」と入力して、「Value」にデータベースのポート番号（通常は 1433）を入力します。「OK」をクリックします。
 - e. 「New」をクリックし、「Name」に「serverName」と入力して、「Value」にデータベース・サーバーの名前を入力します。「OK」をクリックします。
 - f. 「Save」をクリックします。
- 16 Oracle RTD Database に Oracle Database を使用する場合は、次の手順に従い、データソースのプロパティを設定します。
- a. ページの下部にスクロールして、Oracle データソースのプロパティ・セクションを探します。
 - b. 「URL」に「jdbc:oracle:thin:@db_host:db_port:sid」と入力します。
例: jdbc:oracle:thin:@dbhost.company.com:1521:orcl
 - c. 「OK」をクリックして、「Save」をクリックします。
- 17 Oracle RTD Database に DB2 を使用する場合は、次の手順に従い、データソースのプロパティを設定します。
- a. ページの下部にスクロールして、DB2 Universal データソースのプロパティ・セクションを探します。

- b. 「Database name」に、データベースの名前を入力します。
 - c. 「Driver type」に「4」と入力します。
 - d. 「Server name」に、データベース・サーバーの名前を入力します。
 - e. 「Port number」に、データベースのポート番号を入力します。
 - f. 「OK」をクリックして、「Save」をクリックします。
- 18 WebSphere を再起動してから Integrated Solutions Console を再起動します。
- 19 「Resources」を開き、「JDBC」を開いて「Data Sources」を選択します。
- 「RTD_DS」を選択して「Test Configuration」をクリックします。接続に失敗した場合は、データソースの設定が正しいことを確認してから、データソースを再度テストします。

4.3 WebSphere への Oracle Real-Time Decisions アプリケーションのインストール

Integrated Solutions Console を使用して、WebSphere に Oracle RTD をインストールします。作業を開始する前に、WebSphere が起動していることを確認してください。

- 1 `http://websphere_host:port/ibm/console` から Integrated Solutions Console にアクセスします。ログイン・プロンプトで、管理者のユーザー名とパスワードを入力します。Windows では、「スタート」→「プログラム」から Integrated Solutions Console にアクセスすることもできます。
- 2 左側のツリーで、「Applications」を開いて「Enterprise Applications」を選択します。
- 3 `ivtApp` や `query` などのサンプル・アプリケーションを削除します。これを行うには、次の手順を実行します。
 - a. サンプル・アプリケーションを選択します。
 - b. 「Stop」をクリックします。
 - c. 「Uninstall」をクリックします。
 - d. 「Save」をクリックします。
- 4 「Install」をクリックします。
- 5 新しいアプリケーション・セクションの「Path」に「`RTD_HOME/package/RTD.ear`」と入力するか、このパスを指定します。
`RTD_HOME` は、実際の Oracle RTD のインストール・パスに置き換えてください。
- 6 「Next」をクリックし、もう一度「Next」をクリックして、さらに「Next」をクリックします。
- 7 「Map resource references to resources」の下で、「Use default method (many-to-one mapping)」を選択します。
- 8 前の項で作成した J2C 認証を選択します (`RTDDS_auth`)。「Select All」アイコンをクリックし、「Apply」をクリックします。

- 9 モジュールごとに、「Target Resource JNDI Name」を「SDDS」に設定します。
- 10 「Next」をクリックし、もう一度「Next」をクリックします。
- 11 「Finish」をクリックして、「Save」をクリックします。

4.4 Oracle Real-Time Decisions の起動

WebSphere に Oracle RTD をインストールしても、デフォルトでは Oracle RTD は起動されません。Integrated Solutions Console を使用して、Oracle RTD を起動できます。作業を開始する前に、WebSphere が起動していることを確認してください。

- 1 `http://websphere_host:port/ibm/console` から Integrated Solutions Console にアクセスします。ログイン・プロンプトで、管理者のユーザー名とパスワードを入力します。Windows では、「スタート」→「プログラム」から Integrated Solutions Console にアクセスすることもできます。
- 2 左側のツリーで、「Applications」を開いて「Enterprise Applications」をクリックします。
- 3 「OracleRTD」を選択して「Start」をクリックします。Oracle RTD が起動するまで、数分かかる場合があります。
- 4 URL に `http://server_name:8080/ui` を指定して Decision Center に進み、Oracle RTD が実行中かどうかをチェックします。

Oracle RTD を一度起動した後は、WebSphere を起動および停止すると、Oracle RTD は自動的に起動および停止されます。WebSphere と無関係に Oracle RTD を起動および停止するには、Integrated Solutions Console を使用します。

4.5 WebSphere からの Oracle Real-Time Decisions アプリケーションのアンインストール

Integrated Solutions Console を使用して、WebSphere から Oracle RTD をアンインストールできます。作業を開始する前に、WebSphere が起動していることを確認してください。

- 1 `http://websphere_host:port/ibm/console` から Integrated Solutions Console にアクセスします。ログイン・プロンプトで、管理者のユーザー名とパスワードを入力します。Windows では、「スタート」→「プログラム」から Integrated Solutions Console にアクセスすることもできます。
- 2 左側のツリーで、「Applications」を開いて「Enterprise Applications」を選択します。
- 3 「OracleRTD」を選択して「Stop」をクリックします。
- 4 「Uninstall」をクリックして、「Save」をクリックします。

この手順では、WebSphere から Oracle RTD がアンインストールされますが、オペレーティング・システムから Oracle RTD のファイルが削除されるわけではありません。Oracle RTD のファイルは、手動で削除する必要があります。

4.6 Real-Time Decision Server の SSL の構成 (推奨)


この項に記述する手順に従って、Real-Time Decision Server へのすべてのクライアント接続に対して SSL を設定します。作業を開始する前に、第2.4項の手順に従って、Oracle RTD のデフォルトのキーストアおよびトラストストアのパスワードを変更したことを確認します。また、WebSphere が起動されていることを確認してください。

注意: 独自のキーストアおよびトラストストアを使用する場合は、第2.4項の手順を実行する必要はありません。

- 1 `http://websphere_host:port/ibm/console` から Integrated Solutions Console にアクセスします。ログイン・プロンプトで、管理者のユーザー名とパスワードを入力します。Windows では、「スタート」→「プログラム」から Integrated Solutions Console にアクセスすることもできます。
- 2 左側のツリーで、「Security」を開き、「SSL certificate and key management」を選択します。
- 3 「Related Items」の下の「Key stores and certificates」をクリックします。
- 4 次の手順に従って、Oracle RTD のキーストアを作成します。
 - a. 「New」をクリックします。
 - b. 「Name」に「OracleRTD_KeyStore」と入力します。
 - c. 「Path」に「`RTD_HOME/etc/ssl/sdserver.keystore`」と入力します。また、Oracle RTD のデフォルトのキーストアを使用しない場合は、独自のキーストアのパスを入力します。
 - d. 「Password」および「Confirm password」に、キーストアのパスワードを入力します。Oracle RTD のデフォルトのキーストアを使用する場合は、第2.4項で作成したパスワードを入力します。
 - e. 「Type」に「JKS」と入力します。
 - f. 「OK」をクリックします。
- 5 次の手順に従って、Oracle RTD のトラストストアを作成します。
 - a. 「Key stores and certificates」ページで「New」をクリックします。
 - b. 「Name」に「OracleRTD_TrustStore」と入力します。
 - c. 「Path」に「`RTD_HOME/etc/ssl/sdtrust.store`」と入力します。また、Oracle RTD のデフォルトのトラストストアを使用しない場合は、独自のトラストストアのパスを入力します。
 - d. 「Password」および「Confirm password」に、トラストストアのパスワードを入力します。Oracle RTD のデフォルトのトラストストアを使用する場合は、第2.4項で作成したパスワードを入力します。
 - e. 「Type」に「JKS」と入力します。
 - f. 「OK」をクリックします。

- 6 「SSL certificate and key management」ページに戻り、次の手順に従って、Oracle RTDのSSL構成を作成します。
 - a. 「Related Items」の下の「SSL configurations」をクリックします。
 - b. 「New」をクリックします。
 - c. 「Name」に「OracleRTD_SSL」と入力します。
 - d. 「Transport chain name」で「OracleRTD_TrustStore」を選択します。
 - e. 「Keystore name」で「OracleRTD_KeyStore」を選択します。
 - f. 「Get certificate aliases」をクリックします。
 - g. 「OK」をクリックします。
- 7 次の手順に従って、転送チェーンのHTTPSポートを設定します。
 - a. 左側のツリーで、「Servers」を開いて「Application servers」を選択します。
 - b. Oracle RTDをインストールしたアプリケーション・サーバーの名前（たとえば、「server1」）をクリックします。
 - c. 「Container Settings」の下の「Web Container Settings」を開き、「Web container transport chains」を選択します。
 - d. 「WCInboundDefaultSecure」をクリックします。
 - e. 「Transport Channels」の下のTCPインバウンド・チャンネルの名前をクリックします（たとえば、「TCP inbound channel (TCP 4)」）。
 - f. 「Related Items」の下の「Ports」をクリックします。
 - g. 「WC_defaulthost_secure」をクリックします。
 - h. 「Port」を8443に変更します。
 - i. 「OK」をクリックして、「Save」をクリックします。
- 8 次の手順に従って、仮想ホストのHTTPSポートを設定します。
 - a. 左側のツリーで、「Environment」を開いて「Virtual Hosts」を選択します。
 - b. 「default_host」をクリックします。
 - c. 「Additional Properties」の下の「Host Aliases」をクリックします。
 - d. ポート「9443」で「*」をクリックします。
 - e. 「Port」を8443に変更します。
 - f. 「OK」をクリックします。
- 9 次の手順に従って、手順6で作成したOracle RTDのSSLの構成を使用するように、HTTPSポートを設定します。
 - a. 左側のツリーで、「Security」を開き、「SSL certificate and key management」を選択します。
 - b. 「Configuration settings」の下の「Manage endpoint security configurations」をクリックします。

- c. 「Local Topology」の下で「Inbound」→「*cell_name*」→「nodes」→「*node_name*」→「servers」→「*server_name*」と開いて、「WC_defaulthost_secure」をクリックします。
 - d. 「Specific SSL configuration for this endpoint」の下で「Override inherited values」を選択して、「SSL configuration」で「OracleRTD_SSL」を選択します。
 - e. 「Update certificate alias list」をクリックします。
 - f. 「Certificate alias in key store」で、キーストアのパスワードを選択します。
 - g. 「Apply」をクリックします。
- 10 「Save」をクリックします。
- 11 WebSphere を再起動します。



注意: 真にセキュアな環境を実現するには、通常の HTTP ポートを無効にして、すべてのクライアント接続が SSL ポートを經由するように設定する必要があります。これを行うには、次の手順を実行します。

1. JConsole を使用して、「OracleRTD」→「SDPropertyManager」→「Misc MBean」を選択し、**WebServerPort** 属性を HTTP ポートから SSL ポートに変更します。次に、**HTTPSEnabled** 属性を `true` に設定します。詳細は、第10.1項「JConsole へのアクセス」を参照してください。
2. アプリケーション・サーバーのツールを使用して、Web サーバーの HTTP ポートを無効にします。詳細は、WebSphere のドキュメントを参照してください。

- 12 独自のキーストアおよびトラストストアを使用する場合は、次の追加手順を実行して、Decision Studio および Load Generator に対して SSL を有効にします。Oracle RTD のデフォルトのキーストアおよびトラストストアを使用する場合は、この手順を実行する必要はありません。
- a. `RTD_HOME\ eclipse \ eclipse . ini` を開いて編集します。
 - b. `-Djava.net.ssl.trustStore="..\etc\ssl\sdtrust.store"` という行を探します。
 - c. `..\etc\ssl\sdtruststore` を、使用するトラストストア・ファイルのフルパスに置き換えます。
 - d. ファイルを保存して閉じます。
 - e. `RTD_HOME\ scripts \ sdexec . cmd` を開いて編集します。
 - f. ファイルの下部にある、先頭が `%SD_START%` になっている行を探します。ファイルの終わり近くにある、`-Djavax.net.ssl.trustStore="%SD_ROOT%\etc\ssl\sdtrust.store"` という文字列を探します。
 - g. `%SD_ROOT%\etc\ssl\sdtruststore` を、使用するトラストストア・ファイルのフルパスに置き換えます。
 - h. ファイルを保存して閉じます。

4.6.1 SSL 構成のテスト

SSL ポートが正しく機能していることを確認するには、URL に `http://server_name:ssl_port/ui` を指定して Decision Center に進みます。SSL ポートが正しく機能している場合は、ブラウザにログイン画面「Welcome to Decision Center」が表示されます。

Web ブラウザには、「Do you want to accept this certificate?」のようなメッセージが表示される場合もあります。このメッセージが生成されるのは、Oracle RTD のデフォルトのキーストアに同梱されている自己署名証明書について、ブラウザが認識していないことが理由です。この自己署名証明書は、開発環境とテスト環境には適していますが、本番環境には使用しないことをお勧めします。

本番環境では、自己署名証明書を、Verisign や Thawte のような信頼できる認証局 (CA) が発行する証明書に置き換えることをお勧めします。この証明書は、Sun 社のキーツール・ユーティリティで生成された証明書リクエストを CA に送信することによって発行されます。証明書リクエストの生成、および CA からキーストアへの証明書のインポートの手順は、<http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html> を参照してください。

4.7 WebSphere に対する JConsole の設定

この後の項の手順を実行して、JConsole 管理ツールを設定します。

4.7.1 WebSphere のブートストラップ・ポート番号の確認

WebSphere のブートストラップ・ポート番号は、JConsole にアクセスするために、JMX リモート・ポートとして使用されます。JConsole へのアクセスを設定するために、ポートの値を知っておく必要があります。次の手順に従って、ブートストラップ・ポート番号の値を探し、必要に応じてそれを変更します。

- 1 `http://websphere_host:port/ibm/console` から Integrated Solutions Console にアクセスします。ログイン・プロンプトで、管理者のユーザー名とパスワードを入力します。Windows では、「スタート」→「プログラム」から Integrated Solutions Console にアクセスすることもできます。
- 2 左側のツリーで、「Servers」を開いて「Application servers」を選択します。
- 3 Oracle RTD が実行されているアプリケーション・サーバーの名前（たとえば、「server1」）をクリックします。「Communications」の下の「Ports」をクリックします。
- 4 「Ports」テーブルの「BOOTSTRAP_ADDRESS」エントリを探すと、ブートストラップ・ポートの値を見つけることができます。この値をメモして、以降の項で JConsole ポートにそれを使用することも、このブートストラップ・ポート番号を変更することもできます。ブートストラップ・ポート番号を変更するには、次の手順を実行します。
 - a. 「BOOTSTRAP_ADDRESS」をクリックします。
 - b. 「Port」に、JConsole で使用するポート番号を入力します（たとえば、12345）。
 - c. 「OK」をクリックして、「Save」をクリックします。

d. WebSphere を再起動します。

4.7.2 JConsole のバッチ・ファイルの設定

Windows オペレーティング・システムの場合は、startJConsole.bat というバッチ・ファイルを作成して、次の内容を含めます。

```
set WAS_HOME=WEBSPPHERE_HOME¥AppServer
set USER_HOME=WEBSPPHERE_HOME¥AppServer¥profiles¥profile_name
set WAS_HOST=localhost
set WAS_BOOTSTRAP_PORT=jmx_remote_port

"%WAS_HOME%¥java¥bin¥jconsole"
-J-Djava.class.path="%WAS_HOME%¥runtimes¥com.ibm.ws.admin.client_6.1.0.jar;%WAS_HOME%¥java¥lib¥tools.jar"
-J-Dcom.ibm.CORBA.ConfigURL=file:%USER_HOME%¥properties¥sas.client.props
-J-Dcom.ibm.SSL.ConfigURL=file:%USER_HOME%¥properties¥ssl.client.props
service:jmx:iiop://%WAS_HOST%:%WAS_BOOTSTRAP_PORT%/jndi/JMXConnector
```

jmx_remote_port には、JConsole のポート番号（たとえば、12345）を入力します。WebSphere では、JMX リモート・ポートは常に、WebSphere のブートストラップ・ポートと一致しています。*WEBSPPHERE_HOME* は、実際の WebSphere のインストール・パスに置き換え、*profile_name* は、実際の WebSphere プロファイルの名前（たとえば、AppSrv01）に置き換えてください。

ヒント: ファイルの最後の JConsole コマンド (“%WAS_HOME%¥java¥bin¥jconsole...”) で、すべてが 1 行に記述されていることを必ず確認してください。

Linux または AIX ベースのシステムの場合は、startJConsole.sh というバッチ・ファイルを作成して、次の内容を含めます。

```
#!/bin/sh
WAS_HOME=WEBSPPHERE_HOME/AppServer
USER_HOME=WEBSPPHERE_HOME/AppServer/profiles/profile_name
WAS_HOST=localhost
WAS_BOOTSTRAP_PORT=jmx_remote_port

$WAS_HOME/java/bin/jconsole
-J-Djava.class.path=$WAS_HOME/runtimes/com.ibm.ws.admin.client_6.1.0.jar:
$WAS_HOME/java/lib/tools.jar
-J-Dcom.ibm.CORBA.ConfigURL=file:$USER_HOME/properties/sas.client.props
-J-Dcom.ibm.SSL.ConfigURL=file:$USER_HOME/properties/ssl.client.props
service:jmx:iiop://$WAS_HOST:$WAS_BOOTSTRAP_PORT/jndi/JMXConnector
```

jmx_remote_port には、JConsole のポート番号（たとえば、12345）を入力します。WebSphere では、JMX リモート・ポートは常に、WebSphere のブートストラップ・ポートと一致しています。*WEBSPPHERE_HOME* は、実際の WebSphere のインストール・パスに置き換え、*profile_name* は、実際の WebSphere プロファイルの名前（たとえば、AppSrv01）に置き換えてください。

ヒント: ファイルの最後の JConsole コマンド (\$WAS_HOME/java/bin/jconsole...) で、すべてが 1 行に記述されていることを必ず確認してください。また、startJConsole.sh ファイルに、適切な実行権限が設定されていることを確認してください。

Solaris オペレーティング・システムの場合は、startJConsole.sh というバッチ・ファイルを作成して、次の内容を含めます。

```
#!/bin/sh
WAS_HOME=WEBSPHERE_HOME/AppServer
USER_HOME=WEBSPHERE_HOME/AppServer/profiles/profile_name
WAS_HOST=localhost
WAS_BOOTSTRAP_PORT=jmx_remote_port

$WAS_HOME/java/bin/jconsole
-J-Djava.class.path=$WAS_HOME/runtimes/com.ibm.ws.admin.client_6.1.0.jar:
$WAS_HOME/java/lib/tools.jar:$WAS_HOME/java/lib/jconsole.jar
-J-Dcom.ibm.CORBA.ConfigURL=file:$USER_HOME/properties/sas.client.props
-J-Dcom.ibm.SSL.ConfigURL=file:$USER_HOME/properties/ssl.client.props
service:jmx:iiop://$WAS_HOST:$WAS_BOOTSTRAP_PORT/jndi/JMXConnector
```

jmx_remote_port には、JConsole のポート番号（たとえば、12345）を入力します。WebSphere では、JMX リモート・ポートは常に、WebSphere のブートストラップ・ポートと一致しています。*WEBSPHERE_HOME* は、実際の WebSphere のインストール・パスに置き換え、*profile_name* は、実際の WebSphere プロファイルの名前（たとえば、AppSrv01）に置き換えてください。

ヒント: ファイルの最後の JConsole コマンド（*\$WAS_HOME/java/bin/jconsole...*）で、すべてが 1 行に記述されていることを必ず確認してください。また、startJConsole.sh ファイルに、適切な実行権限が設定されていることを確認してください。

4.7.3 JConsole ユーザーの作成

- 1 http://websphere_host:port/ibm/console から Integrated Solutions Console にアクセスします。ログイン・プロンプトで、管理者のユーザー名とパスワードを入力します。Windows では、「スタート」→「プログラム」から Integrated Solutions Console にアクセスすることもできます。
- 2 左側のツリーで、「Users and Groups」を開き、「Manage Users」を選択します。
- 3 「Create」をクリックして、作成する JConsole ユーザーの情報を入力します。たとえば、「User ID」に「jmx_admin」、「First name」に「JMX」、「Last name」に「Admin」と入力して、パスワードを入力します。
- 4 「Create」をクリックします。
- 5 左側のツリーで、「Users and Groups」を開き、「Administrative User Roles」を選択します。
- 6 「Add」をクリックして、次の値を入力します。
 - a. 「User」に、手順 3 で入力したユーザー ID を入力します。
 - b. 「Role(s)」で「Administrator」を選択します。
- 7 「OK」をクリックして、「Save」をクリックします。

これで、JConsole を実行し、作成したユーザー名およびパスワードを使用してログインできるようになります。詳細は、第 10.1 項「JConsole へのアクセス」を参照してください。

4.8 Oracle Real-Time Decisions のポート番号の変更

Oracle RTD アプリケーションのポート番号を変更するには、WebSphere と Oracle RTD 両方でポート番号を変更する必要があります。

4.8.1 WebSphere でのポート番号の変更

- 1 `http://websphere_host:port/ibm/console` から Integrated Solutions Console にアクセスします。ログイン・プロンプトで、管理者のユーザー名とパスワードを入力します。Windows では、「スタート」→「プログラム」から Integrated Solutions Console にアクセスすることもできます。
- 2 左側のツリーで、「Servers」を開いて「Application server」を選択します。
- 3 Oracle RTD が実行されているアプリケーション・サーバーの名前（たとえば、「server1」）をクリックします。「Container Settings」の下の「Web container settings」を開き、「Web container transport chains」を選択します。
- 4 変更するポート番号に対応する転送チェーンの名前をクリックします。たとえば、Oracle RTD の新しい転送チェーンを設定する場合は、「OracleRTD_chain」をクリックします。Oracle RTD アプリケーションがこのアプリケーション・サーバー・プロファイルにデフォルトの転送チェーンを使用している場合は、「WCInboundDefault」をクリックします。
- 5 TCP インバウンド・チャネルのリンクをクリックします（たとえば、「TCP inbound channel (TCP 6)」）。
- 6 「Related Items」の下の「Ports」をクリックします。
- 7 変更するポート番号の名前（たとえば、「OracleRTD_port」や「WC_defaulthost」）をクリックします。
- 8 「Port」に、Oracle RTD で使用する新しいポート番号を入力します。
- 9 「OK」をクリックして、「Save」をクリックします。
- 10 左側のツリーで、「Environment」を開いて「Virtual Hosts」を選択します。
- 11 「default_host」をクリックします。「Additional Properties」の下の「Host Aliases」をクリックします。
- 12 Oracle RTD ポートの前の値に対応する「*」リンクをクリックします。
- 13 「Port」に、Oracle RTD で使用する新しいポート番号を入力します。
- 14 「OK」をクリックして、「Save」をクリックします。
- 15 WebSphere を再起動します。

4.8.2 Oracle RTD でのポート番号の変更

- 1 WebSphere がインストールされているコンピュータで、第4.7.2項「JConsole のバッチ・ファイルの設定」で作成したバッチ・スクリプトを実行します。
- 2 第4.7.3項「JConsole ユーザーの作成」で作成した JConsole ユーザー（たとえば、`jmx_admin`）としてログインします。
- 3 「MBeans」タブをクリックして、「Oracle RTD」→「SDPropertyManager」→「Misc」MBean に移動します。

- 4 「WebServerPort」を、Oracle RTD アプリケーションの新しいポート番号に変更してから[Enter]を押します。
- 5 JConsole を終了します。
- 6 Oracle RTD を再起動します。

5 Oracle Real-Time Decisions 用の WebLogic の構成

Oracle RTD は、UNIX と Windows どちらのプラットフォームの BEA WebLogic アプリケーション・サーバーでもサポートされます。次の各項では、WebLogic に Real-Time Decision Server をインストールする方法について説明します。



注意: Real-Time Decision Server は UNIX でも Windows でも実行されますが、Oracle RTD クライアント・ツールは、Windows プラットフォームでしか実行できません。

この章の内容は次のとおりです。

- 5.1 サーバー・プロパティの構成
- 5.2 Oracle RTD Database の JDBC プロバイダの作成
- 5.3 WebLogic への Oracle Real-Time Decisions アプリケーションのインストール
- 5.4 Oracle Real-Time Decisions の起動
- 5.5 WebLogic からの Oracle Real-Time Decisions アプリケーションのアンインストール
- 5.6 Real-Time Decision Server の SSL の構成 (推奨)
- 5.7 WebLogic に対する JConsole の設定
- 5.8 Oracle Real-Time Decisions のポート番号の変更

5.1 サーバー・プロパティの構成

WebLogic Server Administration Console という WebLogic の管理コンソールを使用して、サーバー・プロパティを構成します。WebLogic Server Administration Console の使用方法の詳細は、WebLogic のドキュメントを参照してください。

- 1 次の手順に従って、WebLogic を起動します。
 - **Windows の場合:** `BEA_HOME/user_projects/domains/domain_name/bin/startWebLogic.cmd` の WebLogic 起動スクリプトを使用します。
 - **UNIX の場合:** `BEA_HOME/user_projects/domains/domain_name/bin/startWebLogic.sh` の WebLogic 起動スクリプトを使用します。



注意: AIX ベースのシステムで WebLogic を実行している場合、WebLogic を起動する前に次の手順を実行する必要があります。

1. `BEA_HOME/Weblogic92/common/bin/commEnv.sh` ファイルを開いて編集します。
2. 次の行を追加します。
`JAVA_OPTIONS="$JAVA_OPTIONS -Dcom.sun.xml.name.space.QName.useCompatibleSerialVersionUID=1.0"`
3. ファイルを保存して閉じます。

詳細は、http://e-docs.bea.com/platform/suppconfigs/configs/ibm_aix/ibm_aix53_92.html#97122 で、AIX ベースのシステムで WebLogic を実行したときの既知の問題に関する記述を参照してください。

- 2 `http://weblogic_host:port/console` から、Oracle RTD ドメインの WebLogic Server Administration Console にアクセスします。ログイン・プロンプトで、管理者のユーザー名とパスワードを入力します。Windows では、「スタート」→「プログラム」→「BEA Products」→「User Projects」→「*domain_name*」→「Admin Server Console」から WebLogic Server Administration Console にアクセスすることもできます。
- 3 次の手順に従って、Oracle RTD アプリケーションのポートの新しいチャネルを作成します。
 - a. 左側のツリーで、「Environment」を開いて「Servers」を選択します。
 - b. Oracle RTD をデプロイする予定のサーバーの名前（たとえば、「AdminServer」）をクリックします。
 - c. 「Protocols」タブをクリックして、「Channels」サブタブをクリックします。
 - d. 「New」をクリックします。「New」ボタンを有効にするには、最初に「Lock & Edit」をクリックする必要があります。
 - e. 「Name」に「OracleRTD_port」などの値を入力します。「Protocol」で「t3」（デフォルト）を選択して、「Next」をクリックします。
 - f. 「Listen Port」および「External Listen Port」に、Oracle RTD で使用するポート番号を入力します。デフォルトでは、Oracle RTD アプリケーションはポート 8080 で実行されます。
 - g. 「Finish」をクリックします。
 - h. Oracle RTD アプリケーションのポートとして 8080 を使用していない場合は、Oracle RTD でポート番号を更新する必要があります。詳細は、第 5.8.2 項「Oracle RTD でのポート番号の変更」を参照してください。第 5.2 項、第 5.3 項および第 5.7 項のタスクを完了するまでは、第 5.8.2 項で説明する手順を実行することはできません。

4 WebLogic を停止します。

- **Windows** では、新しいコマンド・ウィンドウを開いて、次のコマンドを実行します。

```
BEA_HOME\user_projects\domains\domain_name\bin\stopWebLogic.cmd
```

- **UNIX** では、新しいターミナル・ウィンドウを開いて、次のコマンドを実行します。

```
BEA_HOME/user_projects/domains/domain_name/bin/stopWebLogic.sh
```

5 Windows に WebLogic をインストールした場合は、次の手順を実行します。

- a. ファイル `BEA_HOME\user_projects\domains\domain_name\bin\startWebLogic.cmd` のバックアップ・コピーを作成します。
- b. `startWebLogic.cmd` を開いて編集します。ファイルの先頭に、次の行を新しく挿入します。

```
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dorg.eclipse.emf.ecore.EPackage.Registry.INSTANCE=com.sigmadynamics.emf.util.SDEMFRegistry
set JAVA_OPTIONS=%JAVA_OPTIONS% -Djavax.xml.parsers.SAXParserFactory=com.sun.org.apache.xerces.internal.jaxp.SAXParserFactoryImpl
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.sun.management.jmxremote=true
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.sun.management.jmxremote.port=your_jmx_remote_port
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.sun.management.jmxremote.authenticate=false
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.sun.management.jmxremote.ssl=false
```

`your_jmx_remote_port` には、JConsole のポート番号（たとえば、12345）を入力します。

- c. 前の項で説明した `JAVA_OPTIONS` に加えて、JDBC jar ファイルへのパスを `classpath` の設定に追加する必要があります。jar ファイルの名前は、Oracle RTD Database に使用するデータベースのタイプによって異なります。

SQL Server の場合、次の行を追加します。

```
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;RTD_HOME\lib\jdbc\sqljdbc.jar;
```

Oracle Database の場合、次の行を追加します。

```
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;RTD_HOME\lib\jdbc\ojdbc14.jar;
```

DB2 の場合、次の 2 行を追加します。

```
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;RTD_HOME\lib\jdbc\db2jcc.jar;
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;RTD_HOME\lib\jdbc\db2jcc_license_cu.jar;
```

RTD_HOME は、実際の Oracle RTD のインストール・パスに置き換えてください。

- d. ファイルを保存して閉じます。
- 6 UNIXに WebLogic をインストールした場合は、次の手順を実行します。
- a. ファイル *BEA_HOME/user_projects/domains/domain_name/bin/startWebLogic.sh* のバックアップ・コピーを作成します。
 - b. *startWebLogic.sh* を開いて編集します。ファイルの先頭に、次の行を新しく挿入します。

```
JAVA_OPTIONS="{JAVA_OPTIONS} -Dorg.eclipse.emf.ecore.EPackage.Registry.INSTANCE=com.sigmadynamics.emf.util.SDEMFRegistry"
JAVA_OPTIONS="{JAVA_OPTIONS} -Djava.awt.headless=true"
JAVA_OPTIONS="{JAVA_OPTIONS} -Djavax.xml.parsers.SAXParserFactory=com.sun.org.apache.xerces.internal.jaxp.SAXParserFactoryImpl"
JAVA_OPTIONS="{JAVA_OPTIONS} -Dcom.sun.management.jmxremote=true"
JAVA_OPTIONS="{JAVA_OPTIONS} -Dcom.sun.management.jmxremote.port=your_jmx_remote_port"
JAVA_OPTIONS="{JAVA_OPTIONS} -Dcom.sun.management.jmxremote.authenticate=false"
JAVA_OPTIONS="{JAVA_OPTIONS} -Dcom.sun.management.jmxremote.ssl=false"
SAVE_JAVA_OPTIONS=""
```

jmx_remote_port には、JConsole のポート番号（たとえば、12345）を入力します。

AIX ベースのシステムに WebLogic をインストールした場合は、*SAVE_JAVA_OPTIONS=""* の前に次の行を追加します。

```
JAVA_OPTIONS="{JAVA_OPTIONS} -Djava.net.preferIPv4Stack=true"
```

- c. 前の項で説明した *JAVA_OPTIONS* に加えて、JDBC jar ファイルへのパスを *classpath* の設定に追加する必要があります。jar ファイルの名前は、Oracle RTD Database に使用するデータベースのタイプによって異なります。

SQL Server の場合、次の行を追加します。

```
EXT_PRE_CLASSPATH="{EXT_PRE_CLASSPATH}:RTD_HOME/lib/jdbc/sqljdbc.jar"
```

Oracle Database の場合、次の行を追加します。

```
EXT_PRE_CLASSPATH="{EXT_PRE_CLASSPATH}:RTD_HOME/lib/jdbc/ojdbc14.jar"
```

DB2 の場合、次の 2 行を追加します。

```
EXT_PRE_CLASSPATH="{EXT_PRE_CLASSPATH}:RTD_HOME/lib/jdbc/db2jcc.jar"
EXT_PRE_CLASSPATH="{EXT_PRE_CLASSPATH}:RTD_HOME/lib/jdbc/db2jcc_license_cu.jar"
```

RTD_HOME は、実際の Oracle RTD のインストール・パスに置き換えてください。

- d. ファイルを保存して閉じます。
- 7 変更した起動スクリプトを使用して、次の手順に従って、WebLogic を起動します。
 - **Windows の場合:** `BEA_HOME/user_projects/domains/domain_name/bin/startWebLogic.cmd`
 - **UNIX の場合:** `BEA_HOME/user_projects/domains/domain_name/bin/startWebLogic.sh`

5.2 Oracle RTD Database の JDBC プロバイダの作成

WebLogic Server Administration Console を使用して、Oracle RTD Database に JDBC プロバイダを作成します。作業を開始する前に、WebLogic が起動していることを確認してください。

- 1 `http://weblogic_host:port/console` から、Oracle RTD ドメインの WebLogic Server Administration Console にアクセスします。ログイン・プロンプトで、管理者のユーザー名とパスワードを入力します。Windows では、「スタート」→「プログラム」→「BEA Products」→「User Projects」→「*domain_name*」→「Admin Server Console」から WebLogic Server Administration Console にアクセスすることもできます。
- 2 左側のツリーで、「Services」を開き、「JDBC」を開いて「Data Sources」を選択します。
- 3 「New」をクリックします。「New」ボタンを有効にするには、最初に「Lock & Edit」をクリックする必要があります。
- 4 「JDBC Data Source Properties」ページで、次の手順を実行します。
 - a. 「Name」に「RTD_DS」と入力します。
 - b. 「JNDI Name」に「SDDS」と入力します。
 - c. 「Database Type」で「Other」を選択します。
 - d. 「Database Driver」が「Other」に設定されていることを確認し、「Next」をクリックします。
- 5 「Transaction Options」ページで「Supports Global Transactions」の選択を解除して、「Next」をクリックします。
- 6 「Connection Properties」ページで、次の手順を実行します。
 - a. 「Database Name」に、第2.2項で作成した Oracle RTD Database の名前を入力します。
 - b. 「Host Name」に、データベース・サーバーをホスティングするコンピュータの名前を入力します。
 - c. 「Port」に、データベースへの接続に使用されるデータベース・サーバーのポート番号を入力します（SQL Server の場合は 1433、Oracle Database の場合は 1521、DB2 の場合は 50000）。
 - d. 「Database User Name」に、データベースのランタイム・ユーザーの名前を入力します。

- e. 「Password」に、データベースのランタイム・ユーザーのパスワードを入力して、「Next」をクリックします。
- 7 「Test Database Connection」ページの「Driver Class Name」に、接続プールの物理データベース接続を作成するために使用される JDBC ドライバ・クラスの完全なパッケージ名を入力します（このドライバ・クラスは、デプロイ先のサーバーの classpath に設定されている必要があります）。
 - **SQL Server:** `com.microsoft.sqlserver.jdbc.SQLServerDriver`
 - **Oracle Database:** `oracle.jdbc.driver.OracleDriver`
 - **DB2:** `com.ibm.db2.jcc.DB2Driver`
- 8 「URL」に、接続先のデータベースの URL を入力します。URL の書式は、データベースのタイプによって異なります。
 - **SQL Server:** `jdbc:sqlserver://db_host:db_port`
 - **Oracle Database:** `jdbc:oracle:thin:@db_host:db_port:sid`
 - **DB2:** `jdbc:db2://db_host:db_port/db_name`
- 9 「Properties」フィールドに、JDBC ドライバに必要なプロパティとその値を入力します。指定する必要があるプロパティは、データベースのタイプによって異なります。
 - **SQL Server** の場合、プロパティ `user=db_user_name` および `DatabaseName=db_name` を個別の行に入力します。
 - **Oracle Database** の場合、プロパティ `user=db_user_name` を入力します。
 - **DB2** の場合、プロパティ `user=db_user_name` を入力します。
- 10 ページの下部にスクロールして、「Test Table」に「SDAPPS」と入力します。
- 11 「Test Configuration」をクリックします。テストに失敗した場合は、戻って設定を確認します。テストが成功した場合は、「Next」をクリックします。
- 12 データソースを使用可能にするサーバー（たとえば、「AdminServer」）を選択します。この手順は、データソースの構成を完了する前に実行する必要があります。
- 13 「Finish」をクリックします。
- 14 「Activate Changes」をクリックします。

5.3 WebLogic への Oracle Real-Time Decisions アプリケーションのインストール

WebLogic Server Administration Console を使用して、WebLogic に Oracle RTD をインストールします。作業を開始する前に、WebLogic が起動していることを確認してください。

- 1 `http://weblogic_host:port/console` から、Oracle RTD ドメインの WebLogic Server Administration Console にアクセスします。ログイン・プロンプトで、管理者のユーザー名とパスワードを入力します。Windows では、「スタート」→「プログラム」→「BEA Products」→「User Projects」→「`domain_name`」→「Admin Server Console」から WebLogic Server Administration Console にアクセスすることもできます。

- 2 左側のツリーで、「Deployments」をクリックします。
- 3 「Install」をクリックします。「Install」ボタンを有効にするには、最初に「Lock & Edit」をクリックする必要があります。
- 4 RTD.ear ファイルを展開したファイル・パス（たとえば、*RTD_HOME/package/expanded*）を選択して、「Next」をクリックします。第2.1項で、RTD.ear ファイルは展開されています。
- 5 「Install this deployment as an application」を選択して、「Next」をクリックします。
- 6 「Optional Settings」ページで、「Name」に「OracleRTD」と入力します。「Next」をクリックします。
- 7 設定を確認して「Finish」をクリックします。
- 8 「Save」をクリックして、「Activate Changes」をクリックします。
- 9 WebLogic を再起動します。
- 10 再度 WebLogic Server Administration Console にアクセスします。
- 11 左側のツリーで、「Deployments」をクリックします。
- 12 デプロイのリストで、アプリケーション OracleRTD の「State」列が「Active」になっていることを確認します。ステータスが「Prepared」になっている場合は、Oracle RTD を起動する必要があります。問題が発生した場合、WebLogic のログでエラーをチェックし、問題を解決して WebLogic を再起動します。
- 13 URL に `http://server_name:8080/ui` を指定して Decision Center に進み、Oracle RTD が実行中かどうかをチェックします。Decision Center のログイン・ページが表示されます。

Decision Center では、認証がまだ有効になっていません。認証を有効にするには、第6章「Oracle Real-Time Decisions の認証の構成」を参照してください。

5.4 Oracle Real-Time Decisions の起動

WebLogic に Oracle RTD をインストールしても、デフォルトでは Oracle RTD は起動されません。WebLogic Server Administration Console を使用して、Oracle RTD を起動できます。作業を開始する前に、WebLogic が起動していることを確認してください。

- 1 `http://weblogic_host:port/console` から、Oracle RTD ドメインの WebLogic Server Administration Console にアクセスします。ログイン・プロンプトで、管理者のユーザー名とパスワードを入力します。Windows では、「スタート」→「プログラム」→「BEA Products」→「User Projects」→「*domain_name*」→「Admin Server Console」から WebLogic Server Administration Console にアクセスすることもできます。
- 2 左側のツリーで、「Deployments」をクリックします。
- 3 「OracleRTD」を選択して「Start」をクリックします。最初に「Lock & Edit」をクリックする必要があります。

Oracle RTD を一度起動した後は、WebLogic を起動および停止すると、Oracle RTD は自動的に起動および停止されます。WebLogic と無関係に Oracle RTD を起動および停止するには、WebLogic Server Administration Console を使用します。

5.5 WebLogic からの Oracle Real-Time Decisions アプリケーションのアンインストール

WebLogic Server Administration Console を使用して、WebLogic から Oracle RTD をアンインストールできます。作業を開始する前に、WebLogic が起動していることを確認してください。

- 1 `http://weblogic_host:port/console` から、Oracle RTD ドメインの WebLogic Server Administration Console にアクセスします。ログイン・プロンプトで、管理者のユーザー名とパスワードを入力します。Windows では、「スタート」→「プログラム」→「BEA Products」→「User Projects」→「*domain_name*」→「Admin Server Console」から WebLogic Server Administration Console にアクセスすることもできます。
- 2 左側のツリーで、「Deployments」をクリックします。
- 3 「OracleRTD」を選択して、「Stop」→「Force Stop Now」をクリックします。アクションの確認が要求されたら、「Yes」をクリックします。
- 4 「OracleRTD」を選択して「Delete」をクリックします。アクションの確認が要求されたら、「Yes」をクリックします。
- 5 「Activate Changes」をクリックします。

この手順では、WebLogic から Oracle RTD がアンインストールされますが、オペレーティング・システムから Oracle RTD のファイルが削除されるわけではありません。Oracle RTD のファイルは、手動で削除する必要があります。

5.6 Real-Time Decision Server の SSL の構成（推奨）

この項に記述する手順に従って、Real-Time Decision Server へのすべてのクライアント接続に対して SSL を設定します。作業を開始する前に、第2.4項の手順に従って、Oracle RTD のデフォルトのキーストアおよびトラストストアのパスワードを変更したことを確認します。また、WebLogic が起動されていることを確認してください。

注意: 独自のキーストアおよびトラストストアを使用する場合は、第2.4項の手順を実行する必要はありません。

- 1 `http://weblogic_host:port/console` から、Oracle RTD ドメインの WebLogic Server Administration Console にアクセスします。ログイン・プロンプトで、管理者のユーザー名とパスワードを入力します。Windows では、「スタート」→「プログラム」→「BEA Products」→「User Projects」→「*domain_name*」→「Admin Server Console」から WebLogic Server Administration Console にアクセスすることもできます。

- 2 左側のツリーで、「Environment」を開いて「Servers」を選択します。
- 3 Administration Server の名前（通常は「AdminServer」）をクリックします。最初に「Lock & Edit」をクリックする必要があります。
- 4 「General」タブで「SSL Listen Port Enabled」を選択し、「SSL Listen Port」に「8443」と入力します。
- 5 「Keystores」タブをクリックして、次の値を入力します。
 - a. 「Keystores」で「Custom Identity and Custom Trust」を選択します。
 - b. 「Custom Identity Keystore」に「`RTD_HOME/etc/ssl/sdserver.keystore`」と入力します。また、Oracle RTD のデフォルトのキーストアを使用しない場合は、独自のキーストアのパスを入力します。
 - c. 「Custom Identity Keystore Type」に「JKS」と入力します。
 - d. 「Custom Identity Keystore Passphrase」および「Confirm Custom Identity Keystore Passphrase」に、キーストアのパスワードを入力します。Oracle RTD のデフォルトのキーストアを使用する場合は、第2.4項で作成したパスワードを入力します。
 - e. 「Custom Trust Keystore」に「`RTD_HOME/etc/ssl/sdtrust.store`」と入力します。また、Oracle RTD のデフォルトのトラストストアを使用しない場合は、独自のトラストストアのパスを入力します。
 - f. 「Custom Trust Keystore Type」に「JKS」と入力します。
 - g. 「Custom Trust Keystore Passphrase」および「Confirm Custom Trust Keystore Passphrase」に、トラストストアのパスワードを入力します。Oracle RTD のデフォルトのトラストストアを使用する場合は、第2.4項で作成したパスワードを入力します。
- 6 「SSL」タブをクリックして、次の値を入力します。
 - a. 「Identity and Trust Locations」で「Keystores」を選択します。
 - b. 「Private Key Alias」、「Private Key Passphrase」および「Confirm Private Key Passphrase」に、キーストアのパスワードを入力します。Oracle RTD のデフォルトのキーストアを使用する場合は、第2.4項で作成したパスワードを入力します。
- 7 「Save」をクリックして、「Activate Changes」をクリックします。
- 8 WebLogic を再起動します。



注意: 真にセキュアな環境を実現するには、通常の HTTP ポートを無効にして、すべてのクライアント接続が SSL ポートを經由するように設定する必要があります。これを行うには、次の手順を実行します。

1. JConsole を使用して、「OracleRTD」→「SDPropertyManager」→「Misc MBean」を選択し、**WebServerPort** 属性を HTTP ポートから SSL ポートに変更します。次に、**HTTPSEnabled** 属性を `true` に設定します。詳細は、第10.1項「JConsole へのアクセス」を参照してください。

2. アプリケーション・サーバーのツールを使用して、Web サーバーの HTTP ポートを無効にします。詳細は、WebLogic のドキュメントを参照してください。

- 9 独自のキーストアおよびトラストストアを使用する場合は、次の追加手順を実行して、Decision Studio および Load Generator に対して SSL を有効にします。Oracle RTD のデフォルトのキーストアおよびトラストストアを使用する場合は、この手順を実行する必要はありません。
 - a. `RTD_HOME\ eclipse\ eclipse.ini` を開いて編集します。
 - b. `-Djava.net.ssl.trustStore=..\etc\ssl\sdtrust.store` という行を探します。
 - c. `..\etc\ssl\sdtruststore` を、使用するトラストストア・ファイルのフルパスに置き換えます。
 - d. ファイルを保存して閉じます。
 - e. `RTD_HOME\ scripts\ sdexec.cmd` を開いて編集します。
 - f. ファイルの下部にある、先頭が `%SD_START%` になっている行を探します。ファイルの終わり近くにある、`-Djavax.net.ssl.trustStore=%SD_ROOT%\etc\ssl\sdtrust.store` という文字列を探します。
 - g. `%SD_ROOT%\etc\ssl\sdtruststore` を、使用するトラストストア・ファイルのフルパスに置き換えます。
 - h. ファイルを保存して閉じます。

5.6.1 SSL 構成のテスト

SSL ポートが正しく機能していることを確認するには、URL に `http://server_name:ssl_port/ui` を指定して Decision Center に進みます。SSL ポートが正しく機能している場合は、ブラウザにログイン画面「Welcome to Decision Center」が表示されます。

Web ブラウザには、「Do you want to accept this certificate?」のようなメッセージが表示される場合があります。このメッセージが生成されるのは、Oracle RTD のデフォルトのキーストアに同梱されている自己署名証明書について、ブラウザが認識していないことが理由です。この自己署名証明書は、開発環境とテスト環境には適していますが、本番環境には使用しないことをお勧めします。

本番環境では、自己署名証明書を、Verisign や Thawte のような信頼できる認証局 (CA) が発行する証明書に置き換えることをお勧めします。この証明書は、Sun 社のキーツール・ユーティリティで生成された証明書リクエストを CA に送信することによって発行されます。証明書リクエストの生成、および CA からキーストアへの証明書のインポートの手順は、<http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html> を参照してください。

5.7 WebLogic に対する JConsole の設定

次の手順を実行して、JConsole 管理ツールを設定します。作業を開始する前に、WebLogic が稼動していないことを確認してください。

- 1 使用しているオペレーティング・システムに対応する手順を実行します。

- a. Windows に WebLogic をインストールした場合は、
`BEA_HOME\user_projects\domains\domain_name\bin\startWebLogic.cmd` を開いて編集します。次の行を探します。

```
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.sun.management.jmxremote.  
authenticate=false
```

この行の値を `true` に変更します。保存してファイルを閉じます。

- b. UNIX に WebLogic をインストールした場合は、`startWebLogic.sh` を開いて編集します。次の行を探します。

```
JAVA_OPTIONS="{JAVA_OPTIONS} -Dcom.sun.management.jmxremote.  
authenticate=false"
```

この行の値を `true` に変更します。保存してファイルを閉じます。

- 2 `BEA_HOME/jdkversion/jre/lib/management` に移動します。そこで、`jmxremote.password.template` ファイルのコピーを作成し、その名前を `jmxremote.password` に変更します。

注意: WebLogic が HP-UX または AIX ベースのシステムで実行されている場合は、`JDK_HOME/jre/lib/management` に移動します。`JDK_HOME` は、WebLogic アプリケーション・サーバーで使用される JDK のインストール・ディレクトリです。これらのプラットフォームで WebLogic に使用する JDK の詳細は、第 1.3.4 項を参照してください。

- 3 `jmxremote.password` ファイルを開いて編集します。ファイルの末尾に進み、次の 2 行の前にある番号記号 (#) を削除します。

```
monitorRole QED  
controlRole R&D
```

これらは、1 行で 1 組の資格証明を表しています。つまり、最初がユーザー名で、その後がそれに対応するパスワードです。ユーザー名 `monitorRole` には JConsole MBean の読取り権限のみが付与され、ユーザー名 `controlRole` には JMX MBean の完全な制御権限が付与されています。パスワードを新しいパスワードに変更してください。新しいユーザー名およびパスワードを、別の行に追加することもできます。

- 4 また、資格証明ごとに権限を設定することもできます。これを行うには、同じディレクトリにある `jmxremote.access` ファイルを開いて編集します。次に、キーワードの `readonly` および `readwrite` を使用して、資格証明ごとにアクセス・レベルを指定します。次に例を示します。

```
monitorRole readonly
controlRole readwrite
```

追加した新しい資格証明は、必ず `jmxremote.password` ファイルに追加してください。

- 5 `jmxremote.password` ファイルには、パスワードがクリアテキスト形式で含まれているため、このファイルに対するアクセス権限をファイル所有者に制限する必要があります。これを行うには、次のようにしてアクセス権限を変更します。

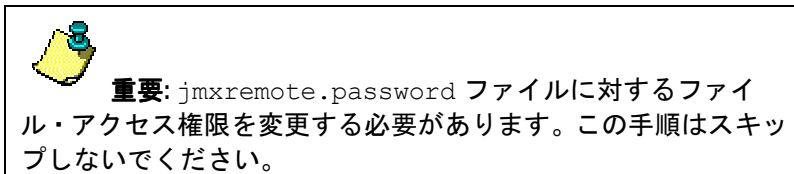
- a. **Windows の場合:**

<http://java.sun.com/j2se/1.5.0/docs/guide/management/security-windows.html> に記載されている手順に従います。

- b. **UNIX の場合:** コマンド `chmod 600 jmxremote.password` を実行します。

JConsole のセキュリティの詳細は、

<http://java.sun.com/j2se/1.5.0/docs/guide/management/agent.html#PasswordAccess> を参照してください。



- 6 WebLogic を起動します。

これで、JConsole を実行し、`jmxremote.password` ファイルに定義されたユーザー名およびパスワードを使用してログインできるようになります。詳細は、第10.1項「JConsole へのアクセス」を参照してください。

5.8 Oracle Real-Time Decisions のポート番号の変更

Oracle RTD アプリケーションのポート番号を変更するには、WebLogic と Oracle RTD 両方でポート番号を変更する必要があります。

5.8.1 WebLogic でのポート番号の変更

- 1 `http://weblogic_host:port/console` から、Oracle RTD ドメインの WebLogic Server Administration Console にアクセスします。ログイン・プロンプトで、管理者のユーザー名とパスワードを入力します。Windows では、「スタート」→「プログラム」→「BEA Products」→「User Projects」→「`domain_name`」→「Admin Server Console」から WebLogic Server Administration Console にアクセスすることもできます。

- 2 左側のツリーで、「Environment」を開いて「Servers」を選択します。
- 3 Oracle RTD が実行されているサーバーの名前（たとえば、「AdminServer」）をクリックします。
- 4 「Protocols」タブをクリックして、「Channels」サブタブをクリックします。
- 5 Oracle RTD のチャンネルの名前（たとえば、「OracleRTD_port」）をクリックします。
- 6 「Enabled」を選択解除します。最初に「Lock & Edit」をクリックする必要があります。
- 7 「Listen Port」に、Oracle RTD で使用する新しいポート番号を入力します。
- 8 「Enabled」を選択します。
- 9 「Save」をクリックして、「Activate Changes」をクリックします。
- 10 WebLogic を再起動します。

5.8.2 Oracle RTD でのポート番号の変更

- 1 JDK のインストール・ディレクトリから `jconsole` を実行して、JConsole を起動します。詳細は、第10.1項「JConsole へのアクセス」を参照してください。
- 2 「Remote」タブをクリックして、次の内容を入力します。
 - a. 「Host」に、Real-Time Decision Server を実行しているコンピュータの名前を入力します。
 - b. 「Port」に、JMX のリモート・ポート番号（たとえば、12345）を入力します。
 - c. 「User Name」および「Password」に、JConsole の設定時に指定した資格証明を入力します。詳細は、第5.7項「WebLogic に対する JConsole の設定」を参照してください。
 - d. 「Connect」をクリックします。
- 3 「MBeans」タブをクリックして、「Oracle RTD」→「SDPropertyManager」→「Misc」MBean に移動します。
- 4 「WebServerPort」を新しいポート番号に変更してから[Enter]を押します。
- 5 JConsole を終了します。
- 6 Oracle RTD を再起動します。

6 Oracle Real-Time Decisions の認証の構成

Oracle RTD は、サーバー（クラスタ）とインライン・サービスの両方のレベルで認証が行われます。Oracle RTD で使用可能な認証は、Windows Authentication、Oracle RTD Platform Authentication、Oracle Siebel Object Manager Authentication の 3 種類です。デフォルトでは、認証は有効になっていません。この章の指示に従って、Oracle RTD の認証を有効にして構成します。

この章の内容は次のとおりです。

- 6.1 Oracle Real-Time Decisions のセキュリティの概要
- 6.2 Windows Authentication の構成
- 6.3 Oracle RTD Platform Authentication の構成
- 6.4 Oracle Siebel Object Manager Authentication の構成
- 6.5 ユーザーおよびグループへの権限の割当て
- 6.6 認証タイプの変更

6.1 Oracle Real-Time Decisions のセキュリティの概要

Oracle RTD のセキュリティには、サーバー全体のセキュリティ、インライン・サービス（アプリケーション・レベル）のセキュリティ、Decision Center のパースペクティブ（ビュー）での追加レベルのセキュリティの 3 つのレベルがあります。また、各データソースには、データベースのユーザー名およびパスワードに基づく追加のセキュリティ・レベルがあります。

サーバー全体のセキュリティは、SecurityManager MBean を使用して構成します。JConsole を使用して、Oracle RTD ユーザーに権限コードを設定できます。詳細は、第10.5項「Oracle Real-Time Decisions のセキュリティの管理」を参照してください。

インライン・サービスのセキュリティは、個々のインライン・サービスに対して構成します。インライン・サービスのセキュリティは、Decision Studio を使用して構成します。詳細は、『Oracle Real-Time Decisions Decision Studio リファレンス・ガイド』を参照してください。

パースペクティブレベルのセキュリティも、Decision Studio で構成します。詳細は、『Oracle Real-Time Decisions Decision Studio リファレンス・ガイド』を参照してください。

重要なのは、3 つのセキュリティ・レベルをすべて構成するということです。そうしないと、問題が発生する可能性があります。たとえば、あるユーザーが Decision Center でデプロイするためのサーバー全体の権限を持っていても、インライン・サービスおよびパースペクティブのレベルでデプロイするための正しい権限がない場合は、そのユーザーはインライン・サービスでの表示および編集ができません。

この項では、次の追加的なセキュリティ項目について説明します。

- 6.1.1 Oracle RTD のデフォルトのアカウントおよびパスワード
- 6.1.2 Oracle RTD の構成ファイルおよびログ・ファイル
- 6.1.3 Oracle RTD の Cookie

6.1.1 Oracle RTD のデフォルトのアカウントおよびパスワード

デフォルトでは、Oracle RTD には次のデフォルトのアカウントおよびパスワードがあります。

- OC4J または WebLogic を使用している場合は、第3.7項および第5.7項で説明しているように、`jmxremote.password` ファイルを使用して JConsole の認証を設定します。このファイルには、`monitorRole` と `controlRole` という、2つのデフォルトのアカウントが含まれています。これらのアカウントのデフォルトのパスワードは、それぞれ `QED` と `R&D` です。
- Oracle RTD へのクライアント接続に SSL を使用する場合は、キーストア、トラストストア、およびサーバーの秘密鍵のデフォルトのパスワードは `tc-ssl` です。Oracle RTD のデフォルトのキーストアおよびトラストストアを使用する場合は、Oracle RTD に SSL を使用する前に、これらのパスワードを変更する必要があります。詳細は、第2.4項を参照してください。

6.1.2 Oracle RTD の構成ファイルおよびログ・ファイル

次の各項では、Oracle RTD の構成ファイルおよびログ・ファイルの名前と場所について、アプリケーション・サーバーのタイプ別に説明します。これらのファイル間で情報の整合性を維持するには、オペレーティング・システムなどのツールを使用して、ファイルが含まれる親ディレクトリに対して適切なファイル権限を設定し、侵入者がアクセスできないようにする必要があります。

6.1.2.1 Oracle RTD の構成ファイルおよびログ・ファイル (スタンドアロン OC4J)

構成ファイル:

`OC4J_HOME/j2ee/home/applications/OracleRTD/etc/sdconfig-cluster.xml`

`OC4J_HOME/j2ee/home/applications/OracleRTD/etc/sdconfig-machine.xml`

`OC4J_HOME/j2ee/home/applications/OracleRTD/etc/sdconfig-tmpl.xml`

`OC4J_HOME/j2ee/home/applications/OracleRTD/etc/sdlog4j.xml`

ログ・ファイル:

`OC4J_HOME/j2ee/home/application/OracleRTD/log/server.log`

`OC4J_HOME/j2ee/home/application-deployments/OracleRTD/application.log`

`OC4J_HOME/j2ee/home/log/server.log`

`OC4J_HOME/j2ee/home/log/oc4j/log.xml`

Oracle RTD の `server.log` (太字表記) は、Oracle RTD の主要なログであり、問題をトラブルシューティングするときに主として使用するログです。

6.1.2.2 Oracle RTD の構成ファイルおよびログ・ファイル (Oracle Application Server)

構成ファイル:

```
ORACLE_AS_HOME/j2ee/oc4j_instance/applications/OracleRTD/etc/  
sdconfig-cluster.xml
```

```
ORACLE_AS_HOME/j2ee/oc4j_instance/applications/OracleRTD/etc/  
sdconfig-machine.xml
```

```
ORACLE_AS_HOME/j2ee/oc4j_instance/applications/OracleRTD/etc/  
sdconfig-tmpl.xml
```

```
ORACLE_AS_HOME/j2ee/oc4j_instance/applications/OracleRTD/etc/sdlog4j.xml
```

ログ・ファイル:

```
ORACLE_AS_HOME/j2ee/oc4j_instance/application-deployments/  
OracleRTD/home_default_group_1/application.log
```

```
ORACLE_AS_HOME/j2ee/oc4j_instance/applications/OracleRTD/log/server.log
```

```
ORACLE_AS_HOME/j2ee/oc4j_instance/og/home_default_group_1/server.log
```

```
ORACLE_AS_HOME/j2ee/oc4j_instance/log/home_default_group_1/oc4j/log.xml
```

```
ORACLE_AS_HOME/opmn/logs/opmn.log
```

```
ORACLE_AS_HOME/opmn/logs/default_group~home~default_group~1.log
```

Oracle RTD の `server.log` (太字表記) は、Oracle RTD の主要なログであり、問題をトラブルシューティングするときに主として使用するログです。

6.1.2.3 Oracle RTD の構成ファイルおよびログ・ファイル (WebSphere)

構成ファイル:

```
WEBSPHERE_HOME/AppServer/profiles/profile_name/installedApps/host_node_  
cell_name/OracleRTD.ear/etc/sdconfig-cluster.xml
```

```
WEBSPHERE_HOME/AppServer/profiles/profile_name/installedApps/host_node_  
cell_name/OracleRTD.ear/etc/sdconfig-machine.xml
```

```
WEBSPHERE_HOME/AppServer/profiles/profile_name/installedApps/host_node_  
cell_name/OracleRTD.ear/etc/sdconfig-tmpl.xml
```

```
WEBSPHERE_HOME/AppServer/profiles/profile_name/installedApps/host_node_  
cell_name/OracleRTD.ear/etc/sdlog4j.xml
```

ログ・ファイル:

WEBSPHERE_HOME/AppServer/profiles/*profile_name*/logs/server1/startServer.log

WEBSPHERE_HOME/AppServer/profiles/*profile_name*/logs/server1/stopServer.log

WEBSPHERE_HOME/AppServer/profiles/*profile_name*/logs/server1/SystemOut.log

WEBSPHERE_HOME/AppServer/profiles/*profile_name*/logs/server1/SystemErr.log

WEBSPHERE_HOME/AppServer/profiles/*profile_name*/logs/fdcc/server1/
server1_*.txt (error logs)

***WEBSPHERE_HOME*/AppServer/profiles/*profile_name*/installedApps/*host_node_cell_name*/OracleRTD.ear/log/server.log**

Oracle RTD の *server.log* (太字表記) は、Oracle RTD の主要なログであり、問題をトラブルシューティングするときに主として使用するログです。

6.1.2.4 Oracle RTD の構成ファイルおよびログ・ファイル (WebLogic)

構成ファイル:

RTD_HOME/package/expanded/etc/sdconfig-cluster.xml

RTD_HOME/package/expanded/etc/sdconfig-machine.xml

RTD_HOME/package/expanded/etc/sdconfig-tmpl.xml

RTD_HOME/package/expanded/etc/sdlog4j.xml

ログ・ファイル:

BEA_HOME/user_projects/domains/*domain_name*/servers/*server_name*/logs/
server_name.log

BEA_HOME/user_projects/domains/*domain_name*/servers/*server_name*/logs/
profile_name.log

***RTD_HOME*/package/expanded/log/server.log**

Oracle RTD の *server.log* (太字表記) は、Oracle RTD の主要なログであり、問題をトラブルシューティングするときに主として使用するログです。

6.1.2.5 構成ファイルおよびログ・ファイル (Oracle RTD クライアント・ツール)

構成ファイル:

RTD_HOME/client/clientHttpEndpoints.properties

RTD_HOME/client/clientHttpsEndpoints.properties

ログ・ファイル:

`RTD_HOME/scripts/SDDBTool.log`

`RTD_HOME/log/client.log`

`RTD_HOME/log/loadgen.csv`

`RTD_HOME/log/server.log`

6.1.3 Oracle RTD の Cookie

Oracle RTD では、永続的でない Cookie を 1 つ生成し、ログイン・セッションの間、クライアント・コンピュータに格納します。この Cookie には個人を特定できる情報は含まれていませんが、Oracle RTD でユーザーがリクエストのたびにログインしなくてもいいように、セッション情報が格納されています。詳細は次のとおりです。

Cookie 名: `sdLoginTicket`

パス: `/ui`

説明: Oracle RTD のセッション・テーブル・インデックスを BASE64 でエンコードした文字列表現、およびユーザー資格証明の SHA-1 ハッシュとランダムな 4 バイトが含まれています。Real-Time Decision Server は、セッションを特定するためのセッション・テーブルを使用してチケットを検証し、セッションの有効期限が切れていないことを確認した後、セッションに同じハッシュが含まれていることを確認します。

6.2 Windows Authentication の構成

Real-Time Decision Server が稼動しているコンピュータで、この項の手順に従って、Oracle RTD に Windows Authentication を構成します。Oracle RTD に Windows Authentication を使用するには、Real-Time Decision Server が Windows コンピュータで稼動している必要があります。

- 1 <http://free.tagish.net/jaas/> から Tagishauth ユーティリティをダウンロードします。必ずバージョン 1.0.3 をダウンロードしてください。
- 2 `tagishauth.jar` を展開して、それを `RTD_HOME¥lib` に置きます。
- 3 `NTSystem.dll` を同じディレクトリに展開します。
- 4 Windows の PATH 環境変数を編集して、`NTSystem.dll` の場所 (`RTD_HOME¥lib`) を入れます。`RTD_HOME` は、実際の Oracle RTD のインストール・パスに置き換えてください。`NTSystem.dll` の場所は、システム PATH の最初に追加する必要があります。`NTSystem.dll` のフォルダの場所のみを追加します。実際のファイル名は含めないでください。次に例を示します。
`C:¥OracleBI¥RTD¥lib`
- 5 Real-Time Decision Server を OC4J で実行している場合は、ファイル `tagishauth.jar` を `RTD_HOME¥lib` から `OC4J_HOME¥j2ee¥home¥applib` にコピーします。その後で OC4J を再起動します。

- 6 Real-Time Decision Server を WebSphere で実行している場合は、次の手順を実行します。
 - a. `http://websphere_host:port/ibm/console` から Integrated Solutions Console にアクセスします。ログイン・プロンプトで、管理者のユーザー名とパスワードを入力します。Windows では、「スタート」→「プログラム」から Integrated Solutions Console にアクセスすることもできます。
 - b. 左側のツリーで、「Environment」を開いて「Shared Libraries」を選択します。
 - c. 「New」をクリックします。共有ライブラリの名前 (Tagish Lib など) を入力して、「Classpath」に「`RTD_HOME¥lib¥tagishauth.jar`」と入力します。`RTD_HOME` は、実際の Oracle RTD のインストール・パスに置き換えてください。「OK」をクリックします。
 - d. 左側のツリーで、「Applications」を開き、「Enterprise Applications」を開いて「OracleRTD」を選択します。
 - e. 画面の下部にスクロールして、「Additional Properties」を開いて「Libraries」をクリックします。
 - f. 「Add」をクリックします。ドロップダウン・リストから共有ライブラリの名前を選択して、「OK」をクリックします。
 - g. 「Save」をクリックします。
- 7 Real-Time Decision Server を WebLogic で実行している場合は、次の手順を実行します。
 - a. `BEA_HOME¥user_projects¥domains¥domain_name¥bin` に移動し、`startWebLogic_RT.D.cmd` ファイルを開いて編集します。
 - b. ファイルの先頭に、次の行を新しく追加します。


```
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;RTD_HOME¥lib¥tagishauth.jar
```

`RTD_HOME` は、実際の Oracle RTD のインストール・パスに置き換えてください。
 - c. ファイルを保存して閉じます。
- 8 Oracle RTD を起動します。
- 9 JConsole を起動してログインします。詳細は、第10.1項「JConsole へのアクセス」を参照してください。
- 10 「OracleRTD」→「SDClusterPropertyManager」→「Windows Authenticator Properties」MBean に移動します。ここで、`WindowsAuthenticatorDomain` プロパティを、ドメイン名またはコンピュータ名に設定します。
- 11 「OracleRTD」→「SDClusterPropertyManager」→「Security Properties」MBean に移動します。ここで、`AuthenticationProviderClass` プロパティを `com.sigmadynamics.server.security.WindowsAuthenticator` に設定し、`AuthenticationEnabled` プロパティを `true` に設定します。

Windows Authentication を使用する場合は、Windows オペレーティング・システムを使用して、Oracle RTD のユーザーおよびグループを作成します。次に、Oracle RTD の権限を、JConsole を使用するユーザーおよびグループに割り当てます。詳細は、第6.5項「ユーザーおよびグループへの権限の割当て」を参照してください。

6.3 Oracle RTD Platform Authentication の構成

この各項の手順に従って、Oracle RTD に Oracle RTD Platform Authentication を構成して、この認証タイプのユーザーとグループを管理します。

Oracle RTD Platform Authentication を設定するには、次の手順を実行します。

- 1 Oracle RTD を起動します。
- 2 JConsole を起動してログインします。詳細は、第10.1項「JConsole へのアクセス」を参照してください。
- 3 「OracleRTD」→「SDClusterPropertyManager」→「Security Properties」MBean に移動します。
- 4 **AuthenticationEnabled** プロパティを `true` に設定します。
- 5 **AuthenticationProviderClass** プロパティを `com.sigmadynamics.server.security.DBAuthenticator` に設定します。

Oracle RTD Platform Authentication のユーザーおよびグループを管理するには、次の手順を実行します。

- 1 Oracle RTD を起動します。
- 2 JConsole を起動してログインします。詳細は、第10.1項「JConsole へのアクセス」を参照してください。
- 3 「OracleRTD」→「SDManagement」→「SecurityManager」MBean に移動します。
- 4 新しい Oracle RTD ユーザーを作成するには、**createUser()**操作を使用します。ユーザーの作成にこの方法を使用するのは、Oracle RTD Platform Authentication を使用する場合のみです。
- 5 新しいセキュリティ・グループを作成するには、**createGroup()**操作を使用します。Oracle RTD グループにユーザーを追加するには、**addUserToGroup()**操作を使用します。

6.4 Oracle Siebel Object Manager Authentication の構成

次の各項では、Oracle RTD に Siebel Object Manager Authentication を構成する方法について説明します。

- 6.4.1 アプリケーション・サーバー用の jar ファイルへのアクセスの設定
- 6.4.2 Oracle RTD での認証の有効化

6.4.1 アプリケーション・サーバー用の jar ファイルへのアクセスの設定

次の各項の手順に従い、アプリケーション・サーバー用の Siebel jar ファイルへのアクセスを設定します。jar ファイルにアクセスを設定する手順を完了したら、第6.4.2項「Oracle RTD での認証の有効化」に進みます。

6.4.1.1 OC4J 用の jar ファイルへのアクセスの設定

- 1 ディレクトリ `SIEBEL_HOME%siebsrvr%CLASSES` で、`Siebel.jar`、`SiebelJI_Common.jar`、`SiebelJI_enu.jar` の各ファイルを探して、適切なディレクトリにコピーします。
 - スタンドアロン OC4J の場合: `OC4J_HOME/j2ee/home/applib`
 - Oracle Application Server の場合: `ORACLE_AS_HOME/j2ee/oc4j_instance/applib`

Siebel 7.5.3 では、3つの jar ファイルがすべて必要です。Siebel 7.7、7.8 および 8.0 では、`SiebelJI.jar` と `SiebelJI_enu.jar` のみが必要です。
- 2 OC4J を再起動します。

6.4.1.2 WebSphere 用の jar ファイルへのアクセスの設定

- 1 `http://websphere_host:port/ibm/console` から Integrated Solutions Console にアクセスします。ログイン・プロンプトで、管理者のユーザー名とパスワードを入力します。Windows では、「スタート」→「プログラム」から Integrated Solutions Console にアクセスすることもできます。
- 2 左側のツリーで、「Environment」を開いて「Shared Libraries」を選択します。右側で、「New」をクリックします。
- 3 「Name」に「Siebel Lib」と入力し、「Classpath」に次の値を入力します。

```
SIEBEL_HOME%siebsrvr%CLASSES%SiebelJI.jar;  
SIEBEL_HOME%siebsrvr%CLASSES%SiebelJI_common.jar;  
SIEBEL_HOME%siebsrvr%CLASSES%SiebelJI_enu.jar
```

`SIEBEL_HOME` は、実際の Siebel Server のインストール・パスに置き換えてください。

各 classpath コンポーネントは、別々の行に入力する必要があります。Siebel 7.5.3 では、3つの jar ファイルがすべて必要です。Siebel 7.7、7.8 および 8.0 では、`SiebelJI.jar` と `SiebelJI_enu.jar` のみが必要です。



Siebel Server が WebSphere とは別のコンピュータにインストールされている場合、これらの jar ファイルは、WebSphere がインストールされているコンピュータの場所にコピーする必要があります。Classpath の値は、WebSphere コンピュータ上の jar ファイルを指すようにします。

- 4 「OK」をクリックします。

- 5 左側のツリーで、「Applications」を開き、「Enterprise Applications」を選択して「OracleRTD」を選択します。
- 6 画面の下部で、「Additional Properties」をクリックして「Libraries」をクリックします。「Add」をクリックします。
- 7 ドロップダウン・リストから「Siebel Lib」を選択して、「OK」をクリックします。
- 8 「Save」をクリックします。


6.4.1.3 WebLogic 用の jar ファイルへのアクセスの設定

- 1 `BEA_HOME\user_projects\domains\domain_name\bin` に移動し、`startWebLogic.cmd` ファイルを開いて編集します。
- 2 ファイルの先頭に、次の行を新しく追加します。

```
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;SIEBEL_HOME%siebsrivr\CLASSES\SiebelJI.jar;
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;SIEBEL_HOME%siebsrivr\CLASSES\SiebelJI_common.jar;
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;SIEBEL_HOME%siebsrivr\CLASSES\SiebelJI_enu.jar;
```

`SIEBEL_HOME` は、実際の Siebel Server のインストール・パスに置き換えてください。

各 classpath コンポーネントは、別々の行に入力する必要があります。Siebel 7.5.3 では、3つの jar ファイルがすべて必要です。Siebel 7.7、7.8 および 8.0 では、`SiebelJI.jar` と `SiebelJI_enu.jar` のみが必要です。



注意: Siebel Server が WebLogic とは別のコンピュータにインストールされている場合、これらの jar ファイルは、WebLogic がインストールされているコンピュータの場所にコピーする必要があります。Classpath の値は、WebLogic コンピュータ上の jar ファイルを指すようにします。

- 3 ファイルを保存して閉じます。


6.4.2 Oracle RTD での認証の有効化

- 1 Oracle RTD を起動します。
- 2 JConsole を起動してログインします。詳細は、第10.1項「JConsole へのアクセス」を参照してください。
- 3 「OracleRTD」 → 「SDClusterPropertyManager」 → 「Security Properties」 MBean に移動します。
- 4 `AuthenticationEnabled` プロパティを `true` に設定します。
- 5 `AuthenticationProviderClass` プロパティを `com.sigmadynamics.server.security.SiebelAuthenticator` に設定します。

- 6 「OracleRTD」 → 「SDClusterPropertyManager」 → 「Siebel Authenticator Properties」 MBean に移動します。
- 7 「SiebelAdminPassword」 および 「SiebelAdminUser」 に、有効な Siebel Server ユーザーの資格証明を入力します。
- 8 「SiebelConnectString」 に、Siebel Server への有効な接続文字列を入力します。次に例を示します。

Siebel.TCPIP.None.None://111.22.33.44:2321/Siebel782/SMObjMgr_enc

接続文字列のフォーマットの詳細は、Siebel Server のドキュメントを参照してください。



注意: 本番システムで、より高度なセキュリティを実現するには、Siebel Object Manager への接続に RSA 暗号化を使用する必要があります。これを行うには、Siebel Server を RSA 用に構成し、その RSA 構成を反映させるために、「OracleRTD」 → 「SDClusterPropertyManager」 → 「Siebel Authenticator Properties」 MBean の SiebelConnectString 属性を更新します。詳細は、Siebel Server のドキュメントを参照してください。

Siebel Object Manager Authentication を使用する場合は、Oracle Siebel Server を使用して、Oracle RTD のユーザーおよびグループを作成します。次に、Oracle RTD の権限を、JConsole を使用するユーザーおよびグループに割り当てます。詳細は、第6.5項「ユーザーおよびグループへの権限の割当て」を参照してください。

6.5 ユーザーおよびグループへの権限の割当て

JConsole を使用して、ユーザーおよびグループに権限を割り当てます。Windows Authentication、Oracle RTD Platform Authentication、Siebel Object Manager Authentication のいずれを使用しているかに関係なく、次の手順で説明する方法を使用します。

- 1 Oracle RTD を起動します。
- 2 JConsole を起動してログインします。詳細は、第10.1項「JConsole へのアクセス」を参照してください。
- 3 「OracleRTD」 → 「SDManagement」 → 「SecurityManager」 MBean に移動します。
- 4 `assignPermission()` 操作を使用して、ユーザーまたはグループに権限を追加します。`assignPermission()`操作を使用するときには、ユーザーまたはグループの名前と権限コードを指定します。権限コードは次のとおりです。
 - 0: 管理者
 - 1: サービスを開く
 - 2: 読取り用サービスを開く
 - 3: Studio からサービスをデプロイする
 - 4: サービスをダウンロードする

- 6: Studio から統合点を実行する

また、`listPermissionCodes()`操作を使用して、使用可能な権限を一覧表示することもできます。

注意: 権限コード 5 (Load Generator から統合点を実行する) は、このリリースではサポートされていません。

6.6 認証タイプの変更

Oracle RTD では、Windows Authentication、Oracle RTD Platform Authentication または Siebel Object Manager Authentication を使用することができます。次の手順で JConsole を使用して、Oracle RTD で使用する認証のタイプを変更できます。

- 1 Oracle RTD を起動します。
- 2 JConsole を起動してログインします。詳細は、第10.1項「JConsole へのアクセス」を参照してください。
- 3 「Oracle RTD」 → 「SDClusterPropertyManager」 → 「Security Properties」 MBean に移動します。
- 4 `AuthenticationEnabled` が `true` に設定されていることを確認します。
- 5 必要に応じて、`AuthenticationProviderClass` を変更します。
 - **Windows Authentication:** `com.sigmadynamics.server.security.WindowsAuthenticator`
 - **Oracle RTD Platform Authentication:** `com.sigmadynamics.server.security.DBAuthenticator`
 - **Siebel Object Manager Authentication:** `com.sigmadynamics.server.security.SiebelAuthenticator`

選択する認証のタイプは、前の各項で説明した手順を使用して構成する必要があります。

認証タイプを変更したら、この認証タイプのユーザーおよびグループに、適切な権限を必ず割り当ててください。詳細は、第6.5項「ユーザーおよびグループへの権限の割当て」を参照してください。

7 Oracle Real-Time Decisions のデータ・アクセスの構成

Oracle RTD の外部データへのアクセスには、JDBC データソースが使用されます。データソースはアプリケーション・サーバーに固有のものであり、インライン・サービスのサプライヤとして使用される新しい JDBC データソースを識別するために使用されます。これらのデータソースは、RDBMS データベースであることも、ODBC で識別されるデータソースであることもあります。サポートされるエンタープライズ・データソースのバージョンの詳細は、『System Requirements and Supported Platforms for Oracle Real-Time Decisions』を参照してください。

この章の内容は次のとおりです。

- 7.1 OC4J での追加の JDBC データソースの作成
- 7.2 WebSphere での追加の JDBC データソースの作成
- 7.3 WebLogic での追加の JDBC データソースの構成
- 7.4 新しいエンタープライズ・データソースのテスト

7.1 OC4J での追加の JDBC データソースの作成

Real-Time Decision Server を OC4J で実行する場合は、この項の手順に従い、インライン・サービスが外部データにアクセスできるように JDBC データソースを構成します。

7.1.1 データソース用の JDBC jar ファイルへのアクセスの設定

この項の手順に従い、OC4J にデータソース用の JDBC jar ファイルへのアクセスを設定します。jar ファイルの名前は、データソースとして追加するデータベースのタイプによって異なります。

- 1 スタンドアロン OC4J を使用する場合は、適切な jar ファイルを `OC4J_HOME/j2ee/home/applib` にコピーします。Oracle Application Server を使用する場合は、適切な jar ファイルを `ORACLE_AS_HOME/j2ee/oc4j_instance/applib` にコピーします。次に示すデータソースに適した jar ファイルをコピーします。

- a. **SQL Server** の場合、ディレクトリ `RTD_HOME/lib/jdbc` にあるファイル `sqljdbc.jar` をコピーします。
- b. **DB2** の場合、`RTD_HOME/lib/jdbc` にあるファイル `db2jcc.jar` および `db2jcc_license_cu.jar` をコピーします。
- c. **Teradata** の場合、ファイル `terajdbc4.jar`、`tdgssjava.jar` および `tdgssconfig.jar` をコピーします。

これらのファイルは、Teradata の Web サイト (<http://www.teradata.com>) からダウンロードできます。jar ファイルのバージョンが 3.3.0.2 以上であることを確認してください。

- d. **Oracle BI EE** の場合、ファイル `bijdbc.jar` をコピーします。

このファイルは、ご使用の Oracle BI EE インストールにあります。

- 2 OC4J を再起動します。

Oracle Database または Oracle Siebel Analytics Server データソースを追加する場合は、jar ファイルをコピーする必要はありません。

7.1.2 OC4J での接続プールとデータソースの作成

Application Server Control を使用して、OC4J に接続プールとデータソースを作成できます。作業を開始する前に、OC4J が起動していることを確認してください。

- 1 oc4jadmin ユーザーとして Application Server Control にログインします。
Application Server Control には、`http://oc4j_host:port/em` からアクセスできます。OC4J のスタンドアロン・バージョンでは、通常、ポート番号は 8888 です。

- 2 「管理」タブをクリックします。「サービス」の下の「JDBC リソース」タスクをクリックします。「接続プール」の下の「作成」をクリックして、エンタープライズ・データベース用の新しい接続プールを作成します。

注意: OC4J を Oracle Application Server の一部として使用している場合は、最初に「グループ」の下の「ホーム」をクリックしてから、「管理」タブに進みます。

- 3 「接続プールの作成 - アプリケーション」ページで、「アプリケーション」に「デフォルト」が選択され、「接続プール・タイプ」に「新規接続プール」が選択されていることを確認します。「続行」をクリックします。

注意: 追加するエンタープライズ・データソースが Oracle RTD Database と同じタイプのデータベースである場合、「接続プール・タイプ」には「既存の接続プールから新規の接続プール」を選択します。次に、「既存の接続プール」リストから「RTDConnectionPool」を選択します。

- 4 「接続プールの作成」ページの「名前」に「db_name_pool」（たとえば、「mydatabase_pool」）を入力します。

- 5 「コネクション・ファクトリ・クラス」に、次のいずれかを入力します。

- **SQL Server:** `com.microsoft.sqlserver.jdbc.SQLServerDriver`
- **Oracle Database:** `oracle.jdbc.driver.OracleDriver`

注意: Oracle Database データソースを追加する場合は、「コネクション・ファクトリ・クラス」に `oracle.jdbc.driver.OracleDriver` を入力する必要があります。デフォルト値の `oracle.jdbc.pool.OracleDataSource` は使用しないでください。

- **DB2:** `com.ibm.db2.jcc.DB2Driver`
- **Teradata:** `com.ncr.teradata.TeraDriver`
- **Oracle BI EE:** `oracle.bi.jdbc.AnaJdbcDriver`
- **Siebel Analytics Server:** `sun.jdbc.odbc.JdbcOdbcDriver`

- 6 「JDBC URL」に、次のいずれかを入力します。

- **SQL Server:** `jdbc:sqlserver://db_host:db_port;databaseName=db_name`
- **Oracle Database:** `jdbc:oracle:thin:@db_host:db_port:sid`

- **DB2:** jdbc:db2://db_host:db_port/db_name
- **Teradata:** jdbc:teradata://db_host/DATABASE=db_name
- **Oracle BI EE:** jdbc:oraclebi://server_name:9703/
user=bi_user_name;password=bi_password;catalog=catalog_name;

注意: カタログ名はオプションです。

- Siebel Analytics Server: jdbc:odbc:SAS_DSN_name

注意: Siebel Analytics Server データソースを追加するには、最初に Siebel Analytics ODBC データソース名 (*SAS_DSN_name*) を作成して構成する必要があります。詳細は、『Siebel Analytics Server Administration Guide』を参照してください。Siebel Analytics Server データソース名には、空白を含めることはできません。

- 7 「資格証明」の下の「ユーザー名」に、エンタープライズ・インスタンスのランタイム権限を持つデータベース・ユーザーの名前を指定します。次に、対応するパスワードを指定します。クリアテキスト形式のパスワードを指定するか、または間接的パスワードを指定するかの詳細は、『Oracle Application Server Containers for J2EE セキュリティ・ガイド』を参照してください。

Oracle BI EE または Siebel Analytics Server データソースを追加する場合は、データベース・ユーザー名ではなく、Oracle BI EE または Siebel Analytics Server の資格証明を指定します。

- 8 「終了」をクリックします。
- 9 「JDBC リソース」ページの「データソース」で、「作成」をクリックして新しいデータソースを定義します。
- 10 「データ・ソースの作成 - アプリケーションとタイプ」ページで、「アプリケーション」に「デフォルト」が選択され、「データ・ソース・タイプ」に「マネージド・データソース」が選択されていることを確認します。「続行」をクリックします。

注意: 追加するエンタープライズ・データソースが Oracle RTD Database と同じタイプのデータベースである場合、「データ・ソース・タイプ」には「既存のデータソースから新規データソース」を選択します。次に、「既存のデータソース」リストから「SDDS」を選択します。

- 11 「データ・ソースの作成 - 管理データ・ソース」ページで、次の手順を実行します。
 - a. 「名前」に、データソースの名前（たとえば、データベースの名前）を入力します。
 - b. 「JNDI ロケーション」に、任意の *jndi_name* を入力します。指定した *jndi_name* は、Oracle RTD がデータソースを参照するために使用されます。
 - c. 「接続プール」に、「db_name_Pool」を選択します。
 - d. 「終了」をクリックします。

- 12 「JDBC リソース」ページの「データソース」テーブルで、追加したエンタープライズ・データソースについて「接続テスト」をクリックします。

注意: Oracle BI EE または Siebel Analytics Server データソースを追加する場合、接続テストは実行しないでください。

次の手順に従ってデータソースをテストします。

- a. Oracle Database データソースを追加する場合は、デフォルトの設定を維持したまま「テスト」をクリックします。
- b. SQL Server、DB2 または Teradata データソースを追加する場合は、「SQL 文」を「select * from *your_table_name*」 (*your_table_name* はデータベース内の既存のテーブル名) に変更して、「テスト」をクリックします。

接続が確立されなかった場合は、OC4J を再起動してデータソースを再度テストします。それでも接続に失敗する場合は、接続プールが正しく設定されているかどうか確認してください。

7.1.3 Oracle RTD への新しいデータソースの追加

OC4J で接続プールとデータソースを作成した後、Oracle RTD に新しいデータソースを追加する必要があります。そのためには、4 つの異なる XML ファイルにリソース参照を追加する必要があります。作業を開始する前に、OC4J が稼動していないことを確認してください。

- 1 Oracle RTD をスタンドアロン OC4J で実行する場合は、ディレクトリ `OC4J_HOME/j2ee/home/applications/OracleRTD` に移動します。Oracle RTD を Oracle Application Server で実行する場合は、`ORACLE_AS_HOME/j2ee/oc4j_instance/applications/OracleRTD` に移動します。

このディレクトリは、RTD.ear ファイルをアプリケーションとしてデプロイしたときにファイルが展開された場所です。

- a. ファイル `./rtis/WEB-INF/web.xml` を探し、それを開いて編集します。ファイルの下部にスクロールします。SDDS_RTIS のリソース参照を定義しているセクションをコピーして、既存のセクションの後ろに貼り付けます。コピーしたセクションで、文字列 SDDS を、第 7.1.2 項の手順 11 で入力した JNDI 名 (`jndi_name`) に置き換えます。次に例を示します。

```
<resource-ref id="jndi_name_RTIS">
<res-ref-name>jndi_name</res-ref-name>
<res-type>javax.sql.DataSource</res-type>
<res-auth>Container</res-auth>
<res-sharing-scope>Unshareable</res-sharing-scope>
</resource-ref>
```

変更を保存して、ファイルを閉じます。

- b. ファイル `./soap/WEB-INF/web.xml` を探し、それを開いて編集します。ファイルの下部にスクロールします。SDDS_Axis のリソース参照を定義しているセクションをコピーして、既存のセクションの後ろに貼り付けます。コピーしたセクションで、文字列 SDDS を、第 7.1.2 項の手順 11 で入力した JNDI 名 (`jndi_name`) に置き換えます。次に例を示します。

```
<resource-ref id="jndi_name_Axis">
<res-ref-name>jndi_name</res-ref-name>
<res-type>javax.sql.DataSource</res-type>
<res-auth>Container</res-auth>
<res-sharing-scope>Unshareable</res-sharing-scope>
</resource-ref>
```

変更を保存して、ファイルを閉じます。

- 2 ディレクトリ `OC4J_HOME/j2ee/home/application-deployments/` OracleRTD に移動します。この場合、`applications` ディレクトリではなく、`application-deployments` ディレクトリに移動することに注意してください。

- a. ファイル `./rtis/WEB-INF/orion-web.xml` を探し、それを開いて編集します。ファイルの下部にスクロールします。SDDS のリソース参照マッピングを定義している行をコピーして、既存の行の後ろに貼り付けます。コピーした行で、文字列 SDDS を、第7.1.2項の手順 11 で入力した JNDI 名 (`jndi_name`) に置き換えます。次に例を示します。

```
<resource-ref-mapping name="jndi_name" location="jndi_name"/>
```

変更を保存して、ファイルを閉じます。

- b. ファイル `./soap/WEB-INF/orion-web.xml` を探し、それを開いて編集します。ファイルの下部にスクロールします。SDDS のリソース参照マッピングを定義している行をコピーして、既存の行の後ろに貼り付けます。コピーした行で、文字列 SDDS を、第7.1.2項の手順 11 で入力した JNDI 名 (`jndi_name`) に置き換えます。次に例を示します。

```
<resource-ref-mapping name="jndi_name" location="jndi_name"/>
```

変更を保存して、ファイルを閉じます。

- 3 OC4J を起動します。

7.2 WebSphere での追加の JDBC データソースの作成

Real-Time Decision Server を WebSphere で実行する場合は、この項の手順に従い、インライン・サービスが外部データにアクセスできるように JDBC データソースを構成します。

7.2.1 WebSphere でのデータ・プロバイダとデータソースの作成

Integrated Solutions Console を使用して、WebSphere にデータ・プロバイダとデータソースを作成できます。作業を開始する前に、WebSphere が起動していることを確認してください。

- 1 追加するエンタープライズ・データソースが Teradata データソースである場合は、最初にファイル `terajdbc4.jar`、`tdgssjava.jar` および `tdgssconfig.jar` を `RTD_HOME/lib/jdbc` にコピーする必要があります。これらのファイルは、Teradata の Web サイト (<http://www.teradata.com>) からダウンロードできます。jar ファイルのバージョンが 3.3.0.2 以上であることを確認してください。

- 2 `http://websphere_host:port/ibm/console` から Integrated Solutions Console にアクセスします。ログイン・プロンプトで、管理者のユーザー名とパスワードを入力します。Windows では、「スタート」→「プログラム」から Integrated Solutions Console にアクセスすることもできます。
- 3 左側のツリーで、「Resources」を開き、「JDBC」を開いて「JDBC Providers」を選択します。
- 4 スcope が正しく (Node=host_nameNode_number に) 設定されていることを確認して、「New」をクリックします。
- 5 「Database type」に、「User-defined」を選択します。
- 6 「Implementation class name」に、次のいずれかを入力します。
 - a. **SQL Server:** `com.microsoft.sqlserver.jdbc.SQLServerConnectionPoolDataSource`
 - b. **Oracle Database:** `oracle.jdbc.pool.OracleConnectionPoolDataSource`
 - c. **DB2:** `com.ibm.db2.jcc.DB2ConnectionPoolDataSource`
 - d. **Teradata:** `com.ncr.teradata.TeraConnectionPoolDataSource`
- 7 「Name」に、このプロバイダの名前 (たとえば、データベースの名前) を入力します。任意の説明を入力することもできます (推奨)。「Next」をクリックします。
- 8 「Class path」に、次のいずれかを入力します。
 - a. **SQL Server:** `RTD_HOME/lib/jdbc/sqljdbc.jar`
 - b. **Oracle Database:** `RTD_HOME/lib/jdbc/ojdbc14.jar`
 - c. **DB2:** `RTD_HOME/lib/jdbc/db2jcc.jar;RTD_HOME/lib/jdbc/db2jcc_license_cu.jar`
 - d. **Teradata:** `RTD_HOME/lib/jdbc/terajdbc4.jar;RTD_HOME/lib/jdbc/tdgssjava.jar;RTD_HOME/lib/jdbc/tdgssconfig.jar`

RTD_HOME は、実際の Oracle RTD のインストール・パスに置き換えてください。
- 9 「Next」をクリックし、「Finish」をクリックして、「Save」をクリックします。
- 10 次の手順に従って、データソースの J2C 認証エントリを作成します。
 - a. 左側のツリーで、「Security」を開き、「Secure administration, applications, and infrastructure」を選択します。
 - b. 「Authentication」の下の「Java Authentication and Authorization Service」を開き、「J2C authentication data」を選択します。「New」をクリックします。
 - c. 「Alias」に、認証エントリとして識別可能な名前 (たとえば、`db_name_auth`) を入力します。
 - d. 「User ID」と「Password」に、エンタープライズ・データベースのユーザー名とパスワードを入力します。
 - e. 「OK」をクリックして、「Save」をクリックします。

- 11 左側のツリーで、「Resources」を開き、「JDBC」を開いて「JDBC Providers」を選択します。このデータベースに追加した JDBC プロバイダの名前をクリックします。
- 12 「Additional Properties」の下の「Data sources」をクリックします。
- 13 次の手順に従って、新しいデータソースを作成します。
 - a. 「New」をクリックします。
 - b. 「Data source name」に、データソースとして識別可能な名前（たとえば、`db_name_DS`）を入力します。
 - c. 「JNDI name」に、「Data source name」で指定した同じ値を入力します。この値は、データソース・オブジェクトでインポートを実行する際に Decision Studio に表示されます。
 - d. 「Component-managed authentication alias and XA authentication」に、手順 10 で作成した J2C 認証のエイリアスを選択します。
 - e. 「Next」をクリックし、「Finish」をクリックして、「Save」をクリックします。
- 14 追加したデータソースの名前をクリックします。
- 15 SQL Server データソースを追加する場合は、次の手順に従い、データソースのプロパティを設定します。
 - a. 「Additional Properties」の下の「Custom properties」をクリックします。
 - b. 「Select All」アイコンをクリックし、「Delete」をクリックして既存のカスタム・プロパティをすべて削除します。
 - c. 「New」をクリックし、「Name」に「databaseName」と入力して、「Value」にデータベースの名前を入力します。「OK」をクリックします。
 - d. 「New」をクリックし、「Name」に「portNumber」と入力して、「Value」にデータベースのポート番号（通常は 1433）を入力します。「OK」をクリックします。
 - e. 「New」をクリックし、「Name」に「serverName」と入力して、「Value」にデータベース・サーバーの名前を入力します。「OK」をクリックします。
 - f. 「Save」をクリックします。
- 16 Oracle Database データソースを追加する場合は、次の手順に従い、データソースのプロパティを設定します。
 - a. ページの下部にスクロールして、Oracle データソースのプロパティ・セクションを探します。
 - b. 「URL」に「`jdbc:oracle:thin:@db_host:db_port:sid`」と入力します。

例: `jdbc:oracle:thin:@dbhost.company.com:1521:orcl`
 - c. 「OK」をクリックして、「Save」をクリックします。

- 17 DB2 データソースを追加する場合は、次の手順に従い、データソースのプロパティを設定します。
 - a. ページの下部にスクロールして、DB2 Universal データソースのプロパティ・セクションを探します。
 - b. 「Database name」に、データベースの名前を入力します。
 - c. 「Driver type」に「4」と入力します。
 - d. 「Server name」に、データベース・サーバーの名前を入力します。
 - e. 「Port number」に、データベースのポート番号を入力します。
 - f. 「OK」をクリックして、「Save」をクリックします。
- 18 Teradata データソースを追加する場合は、次の手順に従い、データソースのプロパティを設定します。
 - a. 「Additional Properties」の下の「Custom properties」をクリックします。
 - b. 「DSName」をクリックし、「Value」に Teradata データベース・サーバー・ホストの名前を入力して、「OK」をクリックします。
 - c. 「databaseName」をクリックし、「Value」に Teradata データベースの名前を入力して、「OK」をクリックします。
 - d. 「Save」をクリックします。

7.2.2 Oracle RTD への新しいデータソースの追加

WebSphere でデータ・プロバイダとデータソースを作成した後、Oracle RTD に新しいデータソースを追加する必要があります。そのためには、次の手順に従って、2つの異なる web.xml ファイルにリソース参照を追加する必要があります。

- 1 次のファイルを開いて編集します。
`WEBSPPHERE_HOME/AppServer/profiles/profile_name/config/cells/server_name/applications/OraclerTD.ear/deployments/OraclerTD/rtis.war/WEB-INF/web.xml`
- 2 次の手順に従って、新しい<resource-ref>エントリを作成します。
 - a. Oracle RTD Database (SDDS) の既存の<resource-ref>エントリをコピーして、その直後に貼り付けます。
 - b. SDDS エントリと同様に、識別可能な値（末尾に_RTIS を付加）を入力して id 属性を変更します。id の値はファイル内で一意である必要があります。
 - c. 第7.2.1項の手順 13 で指定した JNDI 名を入力して、<res-ref-name>タグを変更します。

Teradata データソース用の新しい<resource-ref>エントリの例を次に示します。

```
<resource-ref id="TeradataDS_RTIS">
  <res-ref-name>TeradataDS</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
  <res-sharing-scope>Unshareable</res-sharing-scope>
</resource-ref>
```

- 3 新しいエントリの入力完了したら、ファイルを保存して閉じます。
- 4 次のファイルを開いて編集します。
`WEBSPHERE_HOME/AppServer/profiles/profile_name/config/cells/server_name/applications/OracleRTD.ear/deployments/OracleRTD/soap.war/WEB-INF/web.xml`
- 5 次の手順に従って、新しい<resource-ref>エントリを作成します。
 - a. Oracle RTD Database (SDDS) の既存の<resource-ref>エントリをコピーして、その直後に貼り付けます。
 - b. SDDS エントリと同様に、識別可能な値（末尾に `_Axis` を付加）を入力して `id` 属性を変更します。 `id` の値はファイル内で一意である必要があります。
 - c. 第7.2.1項の手順 13 で指定した JNDI 名を入力して、<res-ref-name>タグを変更します。

Teradata データソース用の新しい<resource-ref>エントリの例を次に示します。

```

<resource-ref id="TeradataDS_Axis">
  <res-ref-name>TeradataDS</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
  <res-sharing-scope>Unshareable</res-sharing-scope>
</resource-ref>

```

- 6 新しいエントリの入力完了したら、ファイルを保存して閉じます。

7.2.3 WebSphere でのリソース参照の更新

`web.xml` ファイルにリソース参照を追加した後、Integrated Solutions Console を使用して、WebSphere でリソース参照を構成する必要があります。作業を開始する前に、WebSphere が起動していることを確認してください。

- 1 `http://websphere_host:port/ibm/console` から Integrated Solutions Console にアクセスします。ログイン・プロンプトで、管理者のユーザー名とパスワードを入力します。Windows では、「スタート」→「プログラム」から Integrated Solutions Console にアクセスすることもできます。
- 2 左側のツリーで、「Applications」を開いて「Enterprise Applications」を選択します。「Oracle RTD」をクリックします。
- 3 「References」の下の「Resource references」をクリックします。
- 4 リソース参照が表示されるテーブルまでスクロール・ダウンし、`web.xml` ファイルに追加した 2 つのエントリを探します。
- 5 各エントリの「Target Resource JNDI Name」に、第7.2.1項の手順 13 で指定した JNDI 名を設定します。
- 6 両方のエントリを選択し、「Specify authentication method」にスクロール・アップして、「Use default method (many-to-one mapping)」を選択します。次に、「Authentication」データ・エントリに、第7.2.1項の手順 10 で作成した J2C 認証エントリを選択します。
- 7 「Apply」をクリックします。

- 8 テーブルの下部にスクロールして「OK」をクリックし、「Save」をクリックします。

7.2.4 別のデータソースの追加

同じタイプのデータソースを新たに追加する場合は、第7.2.1項の手順 10 以降を実行します。特定のデータベース・タイプに対して JDBC プロバイダをすでに作成している場合は、新たに JDBC プロバイダを作成する必要はありません。

異なるタイプのデータソースを新たに追加する場合は、すべての手順を繰り返す必要があります。

7.3 WebLogic での追加の JDBC データソースの構成

Real-Time Decision Server を WebLogic で実行する場合は、この項の手順に従い、インライン・サービスが外部データにアクセスできるように JDBC データソースを構成します。

7.3.1 データソース用の JDBC jar ファイルへのパスの設定

- 1 追加するエンタープライズ・データソースが Teradata データソースである場合は、ファイル `terajdbc4.jar`、`tdgssjava.jar` および `tdgssconfig.jar` を `RTD_HOME/lib/jdbc` にコピーします。これらのファイルは、Teradata の Web サイト (<http://www.teradata.com>) からダウンロードできます。jar ファイルのバージョンが 3.3.0.2 以上であることを確認してください。
- 2 追加するエンタープライズ・データソースが Oracle BI EE データソースである場合は、ファイル `bijdbc.jar` を `RTD_HOME/lib/jdbc` にコピーします。このファイルは、ご使用の Oracle BI EE インストールにあります。
- 3 WebLogic を Windows で実行する場合は、次の手順に従い、データソース用の JDBC jar ファイルへのパスを `classpath` の設定に追加します。
 - a. ファイル `BEA_HOME\user_projects\domains\domain_name\bin\startWebLogic.cmd` を開いて編集します。
 - b. ファイル上部に `classpath` を設定する適切な行を挿入して、データソース用の JDBC jar ファイルへのパスを追加します。jar ファイルの名前は、データソースとして追加するデータベースのタイプによって異なります。

SQL Server の場合、次の行を追加します。

```
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;RTD_HOME\lib\jdbc\sqljdbc.jar;
```

Oracle Database の場合、次の行を追加します。

```
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;RTD_HOME\lib\jdbc\ojdbc14.jar;
```

DB2 の場合、次の 2 行を追加します。

```
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;RTD_HOME¥lib¥jdbc¥
db2jcc.jar;
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;RTD_HOME¥lib¥jdbc¥
db2jcc_license_cu.jar;
```

Teradata の場合、次の 3 行を追加します。

```
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;RTD_HOME¥lib¥jdbc¥
terajdbc4.jar;
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;RTD_HOME¥lib¥jdbc¥
tdgssjava.jar;
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;RTD_HOME¥lib¥jdbc¥
tdgssconfig.jar;
```

Oracle BI EE の場合、次の行を追加します。

```
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;RTD_HOME¥lib¥jdbc¥
bijdbc.jar;
```

RTD_HOME は、実際の Oracle RTD のインストール・パスに置き換えてください。

注意: Siebel Analytics Server データソースを追加する場合は、行を挿入する必要はありません。

- c. ファイルを保存して閉じます。
 - d. WebLogic を再起動します。
- 4 WebLogic を UNIX で実行する場合は、次の手順に従い、データソース用の JDBC jar ファイルへのパスを classpath の設定に追加します。
- a. ファイル `BEA_HOME/user_projects/domains/domain_name/bin/startWebLogic.sh` を開いて編集します。
 - b. ファイル上部に classpath を設定する適切な行を挿入して、データソース用の JDBC jar ファイルへのパスを追加します。jar ファイルの名前は、データソースとして追加するデータベースのタイプによって異なります。

SQL Server の場合、次の行を追加します。

```
EXT_PRE_CLASSPATH="${EXT_PRE_CLASSPATH}:RTD_HOME/lib/jdbc/sqljdbc.
jar"
```

Oracle Database の場合、次の行を追加します。

```
EXT_PRE_CLASSPATH="${EXT_PRE_CLASSPATH}:RTD_HOME/lib/jdbc/ojdbc14.
jar"
```

DB2 の場合、次の 2 行を追加します。

```
EXT_PRE_CLASSPATH="${EXT_PRE_CLASSPATH}:RTD_HOME/lib/jdbc/db2jcc.
jar"
EXT_PRE_CLASSPATH="${EXT_PRE_CLASSPATH}:RTD_HOME/lib/jdbc/db2jcc_
license_cu.jar"
```

Teradata の場合、次の 3 行を追加します。

```
EXT_PRE_CLASSPATH="${EXT_PRE_CLASSPATH}:RTD_HOME/lib/jdbc/
terajdbc4.jar"
EXT_PRE_CLASSPATH="${EXT_PRE_CLASSPATH}:RTD_HOME/lib/jdbc/
tdgssjava.jar"
EXT_PRE_CLASSPATH="${EXT_PRE_CLASSPATH}:RTD_HOME/lib/jdbc/
tdgssconfig.jar"
```

Oracle BI EE の場合、次の行を追加します。

```
EXT_PRE_CLASSPATH="${EXT_PRE_CLASSPATH}:RTD_HOME/lib/jdbc/
bijdbc.jar"
```

RTD_HOME は、実際の Oracle RTD のインストール・パスに置き換えてください。

注意: Siebel Analytics Server データソースを追加する場合は、行を挿入する必要はありません。

- c. ファイルを保存して閉じます。
- d. WebLogic を再起動します。

7.3.2 WebLogic でのデータソースの作成

WebLogic Server Administration Console を使用して、WebLogic にデータソースを作成できます。作業を開始する前に、WebLogic が起動していることを確認してください。

- 1 `http://weblogic_host:port/console` から、Oracle RTD ドメインの WebLogic Server Administration Console にアクセスします。ログイン・プロンプトで、管理者のユーザー名とパスワードを入力します。Windows では、「スタート」→「プログラム」→「BEA Products」→「User Projects」→「*domain_name*」→「Admin Server Console」から WebLogic Server Administration Console にアクセスすることもできます。
- 2 左側のツリーで、「Services」を開き、「JDBC」を開いて「Data Sources」を選択します。
- 3 「New」をクリックします。「New」ボタンを有効にするには、最初に「Lock & Edit」をクリックする必要があります。
- 4 「JDBC Data Source Properties」ページで、次の手順を実行します。
 - a. 「Name」に、データソースとして識別可能な名前（たとえば、*db_name_DS*）を入力します。
 - b. 「JNDI name」に、「Name」で指定した同じ値を入力します。この値は、データソース・オブジェクトでインポートを実行する際に Decision Studio に表示されます。
 - c. 「Database Type」に「Other」を設定します。
 - d. 「Database Driver」が「Other」に設定されていることを確認し、「Next」をクリックします。
- 5 「Transaction Options」ページで「Supports Global Transactions」の選択を解除して、「Next」をクリックします。

- 6 「Connection Properties」 ページで、次の手順を実行します。
- a. 「Database Name」に、データベースの名前を入力します。Oracle BI EE データソースを追加する場合は、空でない文字列を入力します。このプロパティは Oracle BI EE では使用されないため、どのような値でもかまいません。Siebel Analytics Server データソースを追加する場合は、「SAS_DSN_name」と入力します。
注意: Siebel Analytics Server データソースを追加するには、最初に Siebel Analytics ODBC データソース名 (SAS_DSN_name) を作成して構成する必要があります。詳細は、『Siebel Analytics Server Administration Guide』を参照してください。Siebel Analytics Server データソース名には、空白を含めることはできません。
 - b. 「Host Name」に、データベース・サーバーをホスティングするコンピュータの名前を入力します。Oracle BI EE または Siebel Analytics Server データソースを追加する場合は、Oracle BI EE または Siebel Analytics Server をホスティングするコンピュータの名前を入力します。
 - c. 「Port」に、データベースへの接続に使用されるデータベース・サーバーのポート番号を入力します (SQL Server の場合は 1433、Oracle Database の場合は 1521、DB2 の場合は 50000、Oracle BI EE および Siebel Analytics Server の場合は 9703)。
 - d. 「Database User Name」に、データベースのランタイム・ユーザーの名前を入力します。Oracle BI EE または Siebel Analytics Server データソースを追加する場合は、Oracle BI EE または Siebel Analytics Server ユーザーの名前を入力します。
 - e. 「Password」に、データベースのランタイム・ユーザーのパスワードを入力します。Oracle BI EE または Siebel Analytics Server データソースを追加する場合は、Oracle BI EE または Siebel Analytics Server ユーザーのパスワードを入力します。「Next」をクリックします。
- 7 「Test Database Connection」 ページの「Driver Class Name」に、接続プールの物理データベース接続を作成するために使用される JDBC ドライバ・クラスの完全なパッケージ名を入力します (このドライバ・クラスは、デプロイ先のサーバーの classpath に設定されている必要があります)。
- **SQL Server:** com.microsoft.sqlserver.jdbc.SQLServerDriver
 - **Oracle Database:** oracle.jdbc.driver.OracleDriver
 - **DB2:** com.ibm.db2.jcc.DB2Driver
 - **Teradata:** com.ncr.teradata.TeraDriver
 - **Oracle BI EE:** oracle.bi.jdbc.AnaJdbcDriver
 - **Siebel Analytics Server:** sun.jdbc.odbc.JdbcOdbcDriver
- 8 「URL」に、接続先のデータベースの URL を入力します。URL の書式は、データソースのタイプによって異なります。
- **SQL Server:** jdbc:sqlserver://db_host:db_port
 - **Oracle Database:** jdbc:oracle:thin:@db_host:db_port:sid
 - **DB2:** jdbc:db2://db_host:db_port/db_name

- **Teradata:** jdbc:teradata://db_host/DATABASE=db_name
 - **Oracle BI EE:** jdbc:oraclebi:// server_name:9703/user=bi_user_name; password=bi_password; catalog=catalog_name;
注意: カタログ名はオプションです。
 - **Siebel Analytics Server:** jdbc:odbc:SAS_DSN_name
- 9 「Properties」フィールドに、JDBCドライバに必要なプロパティとその値を入力します。指定する必要があるプロパティは、データソースのタイプによって異なります。
- **SQL Server** の場合、プロパティ user=db_user_name および DatabaseName=db_name を個別の行に入力します。
 - **Oracle Database** の場合、プロパティ user=db_user_name を入力します。
 - **DB2** の場合、プロパティ user=db_user_name を入力します。
 - **Teradata** の場合、プロパティ username=db_user_name を入力します。
 - **Oracle BI EE** の場合、プロパティは必要ありません。「Properties」フィールドは空白のままにしておきます。
 - **Siebel Analytics Server** の場合、プロパティ user=SAS_user_name を入力します。
- 10 ページの下部にスクロールします。「Test Table」に、データベース内の既存のテーブル名を入力します。
- 注意:** Oracle BI EE または Siebel Analytics Server データソースを追加する場合、接続テストは実行しないでください。手順 12に進みます。
- 11 「Test Configuration」をクリックします。テストに失敗した場合は、戻って設定を確認します。テストが成功した場合は、「Next」をクリックします。
- 12 変更内容を反映させるサーバー（たとえば、「AdminServer」）を選択します。この手順は、データソースの構成を完了する前に実行する必要があります。
- 13 「Finish」をクリックします。
- 14 「Activate Changes」をクリックします。

7.3.3 Oracle RTD への新しいデータソースの追加

WebLogic でデータソースを作成した後、Oracle RTD に新しいデータソースを追加する必要があります。そのためには、次の手順に従って、soap.war および rtis.war アーカイブ内の web.xml ファイルにリソース参照を追加する必要があります。

- 1 インストール時に RTD.ear ファイルを展開したディレクトリ (`RTD_HOME/package/expanded`) に移動します。
- 2 `rtis.war` アーカイブを開いて `web.xml` を抽出し、`web.xml` を開いて編集します。ファイルの下部にスクロールします。`SDDS_RTIS` のリソース参照を定義しているセクションをコピーして、既存のセクションの後ろに貼り付けます。コピーしたセクションで、文字列 `SDDS` を、第7.3.2項の手順 4 で入力した JNDI 名に置き換えます。次に例を示します。

```
<resource-ref id="jndi_name_RTIS">
<res-ref-name>jndi_name</res-ref-name>
<res-type>javax.sql.DataSource</res-type>
<res-auth>Container</res-auth>
<res-sharing-scope>Unshareable</res-sharing-scope>
</resource-ref>
```

変更内容を保存してファイルを閉じ、ファイルを元の `rtis.war` に再びアーカイブします。

- 3 `soap.war` アーカイブを開いて `web.xml` を抽出し、`web.xml` を開いて編集します。ファイルの下部にスクロールします。`SDDS_Axis` のリソース参照を定義しているセクションをコピーして、既存のセクションの後ろに貼り付けます。コピーしたセクションで、文字列 `SDDS` を、第7.3.2項の手順4で入力した JNDI 名に置き換えます。次に例を示します。

```
<resource-ref id="jndi_name_Axis">
<res-ref-name>jndi_name</res-ref-name>
<res-type>javax.sql.DataSource</res-type>
<res-auth>Container</res-auth>
<res-sharing-scope>Unshareable</res-sharing-scope>
</resource-ref>
```

変更内容を保存してファイルを閉じ、ファイルを元の `soap.war` に再びアーカイブします。

7.4 新しいエンタープライズ・データソースのテスト

新しいエンタープライズ・データソースを追加した後、この項の手順に従って、データソースが正しく構成されていることを確認します。作業を開始する前に、Oracle RTD が起動していることを確認してください。

- 1 `RTD_HOME\ eclipse` で `eclipse.exe` を実行して、Decision Studio を起動します。次に、新しいインライン・サービスを作成するか、既存のインライン・サービスを開きます。詳細は、Decision Studio のヘルプを参照してください。
- 2 「Service Metadata」フォルダを開き、「Data Sources」を右クリックして「New SQL Data Source」を選択します。
- 3 「Display Label」にテストするデータソースの名前を入力して、「OK」をクリックします。
- 4 「Import」をクリックします。次に、「JDBC Data Source」ドロップダウン・リストから、テストするデータソースを選択します。テーブルとビューのリストが、そのデータソースのテーブルとビューの名前で更新されます。
- 5 特定のテーブルまたはビューを選択し、「Import」ダイアログ・ボックスの「Finish」をクリックします。「Data Source」エディタの「Output」テーブルに、利用可能な列のリストが表示されます。
- 6 任意の基本コードを記述し、実行時にそのテーブルから実際の行が取得されることを確認します。

前述の手順を完了できなかった場合は、データソースの構成設定を確認してから、もう一度実行してください。

8 追加の構成設定とクライアント・ツールの起動

この章では、Decision Center と Real-Time Decision Server の追加の構成設定について説明します。

この章の内容は次のとおりです。

- 8.1 Decision Center のブラウザの構成
- 8.2 Real-Time Decision Server のロギングの構成
- 8.3 Oracle Real-Time Decisions クライアント・ツールへのアクセス

8.1 Decision Center のブラウザの構成

Decision Center の Internet Explorer クライアント・ブラウザは、最適なパフォーマンスを得るために、次のように構成する必要があります。

- 1 Internet Explorer で、「ツール」→「インターネットオプション」を選択してオプションを設定します。
- 2 「詳細設定」タブで、「ショートカットを起動するためにウィンドウを再使用する」を選択解除します。
- 3 ブラウザで Cookie が有効になっていることを確認します。

8.2 Real-Time Decision Server のロギングの構成

「OracleRTD」→「SDManagement」→「SDLogging」 MBean のログ出力属性は、Log4J タイプのログ出力です。詳細は、Log4J の Web サイト (<http://logging.apache.org/log4j>) を参照してください。

ロギングの構成パラメータを変更する手順は次のとおりです。

- 1 Oracle RTD を起動します。
- 2 JConsole を起動してログインします。詳細は、第10.1項「JConsole へのアクセス」を参照してください。
- 3 「OracleRTD」→「SDManagement」→「SDLogging」 MBean に移動します。
- 4 必要に応じてロギング属性を変更します。
 - a. 属性 **Priority** および **InlineServicePriority** を **DEBUG**、**INFO**、**WARN** または **ERROR** に設定することにより、サーバー・ログに記録されるログ・メッセージのレベルを指定できます。
 - b. 属性 **LogFile** はサーバー・ログの場所を指定します。デフォルトでは、`RTD_HOME¥log¥server.log` に設定されています。
- 5 [Enter]を押します。変更内容を取り消す場合は、[Esc]キーを押します。

8.3 Oracle Real-Time Decisions クライアント・ツールへのアクセス

Decision Studio、Decision Center、Load Generator などの Oracle RTD クライアント・ツールを起動してアクセスするには、次の各項の手順を実行します。JConsole 管理ツールの起動方法の詳細は、第10.1 項「JConsole へのアクセス」を参照してください。

Decision Center にアクセスするには、Real-Time Decision Server を実行している必要があります。Decision Studio と Load Generator は、Real-Time Decision Server を実行していなくても、部分的に機能します。

8.3.1 Decision Studio へのアクセス

Decision Studio にアクセスするには、Oracle RTD クライアント・ツールをインストールしたクライアント・コンピュータで `RTD_HOME¥eclipse¥eclipse.exe` を実行します。

8.3.2 Decision Center へのアクセス

任意のコンピュータから Decision Center にアクセスするには、Web ブラウザを起動して、`http://server_name:8080/ui` に移動します。

8.3.3 Load Generator へのアクセス

Load Generator にアクセスするには、Oracle RTD クライアント・ツールをインストールしたクライアント・コンピュータで `RTD_HOME¥scripts¥loadgen.cmd` を実行します。

9 Oracle Real-Time Decisions の本番環境へのデプロイ

インライン・サービスをテストして本番環境にデプロイする準備ができたなら、インライン・サービスを1つ以上の本番用サーバーにデプロイします。システムのハードウェア要件は、オール・イン・ワン構成を選択したか、フル・デプロイ構成を選択したかによって異なります。システム構成は、特定の要件やアプリケーション特性に応じて、調整が必要になる場合があります。

この章の内容は次のとおりです。

- 9.1 オール・イン・ワン構成
- 9.2 フル・デプロイ構成

9.1 オール・イン・ワン構成

オール・イン・ワン構成は、コンセプトの実証やパイロットタイプの実装をサポートします。この構成を使用する目的は、長期的に本番環境のシステムをサポートすることではなく、Inline Intelligence ソリューションを導入するメリットを理解することにあります。この構成は、約 100 のアカウント・マネージャ、または約 50 のコール・センター・エージェントをサポートします。

Inline Intelligence™ プラットフォーム	最小構成
J2EE アプリケーション・サーバー	OC4J
OS	Windows
CPU 数	2
CPU	2GHz 以上
RAM	2GB
HD	100GB
データベース	SQL Server、Oracle Database または DB2

9.2 フル・デプロイ構成

フル・デプロイ構成は、本番環境で約 1,500 の B2B アカウント・マネージャまたは約 1,000 のコール・センター・エージェントをサポートします。システム構成を決定する要素は多数あります。たとえば、1秒当たりのピーク時リクエスト数、選択数、モデルの数とサイズ、データソースの数などがあります。プロフェッショナル・サービス・グループは、ビジネスと技術要件に最も適した構成の決定をサポートします。

サーバー	コンポーネント	最小構成
Decision Server (2)	CPU	2 X 2-CPU、2GHz 以上、2GB RAM
	J2EE アプリケーション・サーバー	OC4J、WebSphere または WebLogic
	OS	Windows、Solaris、AIX、HP-UX または Linux

サーバー	コンポーネント	最小構成
データベース・サーバー ¹	CPU	4-CPU、2GHz 以上、8GB RAM
	DBMS	SQL Server、Oracle Database または DB2
Learning Server ²	CPU	2-CPU、2GHz 以上、2GB RAM
	J2EE アプリケーション・サーバー	OC4J、WebSphere または WebLogic
	OS	Windows、Solaris、AIX、HP-UX または Linux
<p>注意:</p> <p>1: データベース・サーバーの CPU と Decision Server の CPU の比率はほぼ 1:1 です。</p> <p>2: Learning Server の CPU と Decision Server の CPU の比率はほぼ 1:4 です。</p>		

10 JMX を使用した Oracle Real-Time Decisions の管理

Oracle Real-Time Decisions では、J2EE 業界標準の Java Management Extensions (JMX) を使用して、管理サービス、デシジョン・サービス、ラーニング・サービス、およびデプロイされたインライン・サービスが管理されます。そのため、Java Development Kit 1.5.0.8 に同梱されている JConsole という JMX 管理ツールを使用して Oracle RTD を管理できます。また、サード・パーティによる JMX 実装を使用して Oracle RTD を管理することもできます。このマニュアルでは、JConsole を使用することを前提にしています。

JMX MBean は、ロギング、インライン・サービス構成、セキュリティ、ユーザーなど、Oracle RTD のあらゆる側面を管理します。Oracle RTD は、次の 3 つのサービスで構成されています。

管理サービス: Decision Studio を使用して作成される構成からインライン・サービスの生成を管理します。

デシジョン・サービス: インライン・サービスを実行し、エンタープライズ運用プロセスに統合します。

ラーニング・サービス: インライン・サービスの基盤となる、分析に基づくセルフラーニング・モデルを保持します。

Oracle RTD は、多くの場合、トランザクションの多い環境でパフォーマンスを向上させるために複数のサーバーやクラスタにデプロイされます。その場合、これらの各サービスでは、コード、トランザクション・データおよび構成を保持するためにリレーショナル・データベースが使用されます。

この章の内容は次のとおりです。

- 10.1 JConsole へのアクセス
- 10.2 JMX MBean の操作と属性
- 10.3 Oracle Real-Time Decisions のクラスタ・レベルの管理
- 10.4 Oracle Real-Time Decisions のメンバー・レベルの管理
- 10.5 Oracle Real-Time Decisions のセキュリティの管理
- 10.6 インライン・サービスの管理
- 10.7 デプロイ状態の管理
- 10.8 ラーニング・サービスの管理
- 10.9 アラート・サービスの管理

10.1 JConsole へのアクセス

OC4J、Oracle Application Server または WebLogic の場合、JConsole にアクセスする手順は次のとおりです。

- 1 `JDK_HOME/bin/jconsole` を実行します。

例: `C:\Program Files\Java\jdk1.5.0_08\bin\jconsole.exe`

コンピュータに複数の JDK をインストールしている場合は、どの `JDK_HOME` を使用してもかまいません。任意の JDK から JConsole を実行できます。



注意: UNIX で JConsole を実行する場合、最初にオペレーティング・システムの環境変数 `PATH` の先頭に `JAVA_HOME/bin` を追加する必要があります。

- 2 「Remote」タブをクリックして、次の内容を入力します。
 - a. 「Host」に、Real-Time Decision Server を実行しているコンピュータの名前を入力します。
 - b. 「Port」に、JMX のリモート・ポート番号（たとえば、12345）を入力します。
 - c. 「User Name」および「Password」に、JConsole の設定時に指定した資格証明を入力します。詳細は、第3.7項「OC4J に対する JConsole の設定」および第5.7項「WebLogic に対する JConsole の設定」を参照してください。

WebSphere の場合、JConsole にアクセスする手順は次のとおりです。

- 1 WebSphere がインストールされているコンピュータで、第4.7.2項「JConsole のバッチ・ファイルの設定」で作成したバッチ・スクリプトを実行します。
- 2 第4.7.3項「JConsole ユーザーの作成」で作成した JConsole ユーザー（たとえば、`jmx_admin`）としてログインします。



注意: Oracle RTD を WebSphere で実行する場合、デフォルトでは、JConsole を WebSphere サーバーと同じコンピュータで実行する必要があります。この問題は、WebSphere の制限によって発生します。

10.2 JMX MBean の操作と属性

Oracle RTD MBean は、JConsole の「MBean」タブの「OracleRTD」フォルダからアクセスできます。これらの MBean を使用して、Oracle RTD の様々な面を管理できます。各 MBean は、情報および管理目的に使用できる属性と操作で構成されます。

SDCluster レベルの属性と操作は、クラスタ・レベルの機能を管理できます。MBean は階層構造になっています。SDCluster の属性には次のものがあります。

- Properties: SD クラスタのプロパティの構成。
- Members: クラスタのメンバー。各メンバーはこの属性によって表示されます。メンバーを使用して、ローカル・サーバー・レベルのプロパティを管理できます。
- Security: 様々な認証属性を管理できます。この属性の多くは、1つの認証タイプに固有のものであり、他の認証タイプには表示されません。
- InlineServiceManager: デプロイされたインライン・サービスを管理します。
- DeploymentStates: デプロイ状態の設定と並替えを実行できます。
- LearningService: ラーニング・サービスの属性を管理します。
- AlertService: アラート・サービスを管理します。

10.3 Oracle Real-Time Decisions のクラスタ・レベルの管理

クラスタ・レベルの管理は、サーバーのクラスタ全体に影響を与えるアイテムに使用します。サーバーを1つしか使用していない場合でも、クラスタ・レベルの管理はあります。

10.3.1 「OracleRTD」 → 「SDManagement」 → 「SDClusterPropertyManager」について

「SDManagement」 → 「SDClusterPropertyManager」 MBean には、次の操作があります。

`java.lang.Void restoreDefault()`

パラメータ: なし

目的: Oracle RTD をデフォルトのインストール設定に戻します。このコマンドをクラスタで実行すると、値がクラスタのデフォルト値に戻されます。

10.3.2 「OracleRTD」 → 「SDClusterPropertyManager」 → 「Misc」について

「SDClusterPropertyManager」 → 「Misc」 MBean には、次の属性があります。

属性	説明
DisableBatchDBOperations	データベースのバッチ操作を指定するブール・スイッチ。
AutoFlushTimeout	データベースの書き込みバッファの自動フラッシュを指定する時間間隔（秒）。端数がサポートされます。

属性	説明
WorkerThreadPoolSize	通常の統合点リクエストの処理ではなく、一般的なメンテナンス・アクティビティに使用されるスレッド数。メンテナンス・アクティビティには、モデルのメンテナンス、セッション時間の操作、およびタイムアウトしたリクエストの処理が含まれます。
IntegrationPointRequestIOFactor	統合点リクエストが、I/O 処理に費やす時間、またはこの VM の外部システムを待機する時間のパーセンテージ。
IntegrationPointMaxConcurrentJobs	現在実行している統合点リクエストの最大数。通常は 0 に設定する必要があります。この場合、値は次のように計算されます。Math.ceil は、次に大きい整数値に丸められることを意味します。 NumCPUs * Math.ceil(1/(1-DSRequestIOFactor)) + 5
IntegrationPointQueueSize	実行を待機する統合点リクエストの最大数。リクエストがこの値を超えると、サーバーで、「Server Too Busy」エラー・メッセージとともにリクエストが終了します。この属性は、サブレット・コンテナによって割り当てられるサブレット・スレッドの数から IntegrationPointMaxConcurrentJobs の構成値または計算値を引いた値以下に設定する必要があります。IntegrationPointMaxConcurrentJobs の計算値は、「SDConsole」→「Members」→「Decision Service」で確認できます。
IntegrationPointGuaranteedRequestTimeout	統合点リクエストに対して保証されるミリ秒単位のレスポンス時間（サービス・レベルの保証）。0 の場合、統合点リクエストにはタイムアウトが発生しません。これは、デバッグにのみ適しています。
DBOperationLogThresholdMilliSec	指定された時間のしきい値を超えるすべてのデータベース操作が記録されます。
DCOperationLogThresholdMilliSec	指定された時間のしきい値を超える Decision Center のすべてのリクエストが記録されます。
SystemDSName	データソースの JNDI 名。
ModelDSName	ラーニング・サービスによって使用されるデータソースの JNDI 名。
ArchivedModelCatalogRefreshInterval	アーカイブされたモデルのカタログのリフレッシュ時間間隔（秒）。カタログは Discovery Explorer によって使用されます。

属性	説明
ArchivedModelCacheTimeToLive	アーカイブされたモデルがメモリーに保存される最大時間（秒）。アーカイブされたモデルのキャッシュは、Discovery Explorer によって使用されます。
DSQueueThreadCount	DS リクエスト・キューをリスニングする JMS スレッドの数。
DSSessionIdleTimeoutMillisec	デシジョン・サービス・セッションのアイドル・タイムアウト（ミリ秒）。
DSManagesSessionAffinity	デシジョン・サービスはセッション・アフィニティを管理します。true に設定すると、デシジョン・サービスはアクティブなセッション・キーのマップを維持し、必要な場合は、統合点リクエストを、キーのセッションを所有しているクラスタ・ホストに転送します。ホストが1つのインストール、およびセッション・アフィニティがアプリケーション・サーバーまたは外部のロード・バランサによって完全に管理されているインストールでは、無効にする必要があります。
CustomLoginLogo	ログイン・ページのカスタム・イメージ。
CustomHeaderLogo	メイン・ページのカスタム・イメージ。
SMTPHost	SMTP メール・サーバーのホスト名。現在、アラート通知の送信にのみ使用されます。
SMTPPort	SMTP メール・サーバーのポート。
SMTPUser	SMTP 認証が必要な場合の SMTP メール・サーバーのユーザー名。
SMTPPassword	SMTP 認証が必要な場合の SMTP メール・サーバーのパスワード。

10.3.3 「OracleRTD」 → 「SDClusterPropertyManager」 → 「Cluster」 について

「SDClusterPropertyManager」 → 「Cluster」 MBean には、次の属性があります。

属性	説明
GenerateDSCookies	Decision Server HTTP Cookie を生成します。true に設定すると、Decision Server が統合点リクエストを HTTP セッションと関連付け、Web コンテナはコンテナ固有のセッション・アフィニティ Cookie を生成します。

属性	説明
JGroupProtocols	JGroups チャンネルに使用されるプロトコルとプロパティを定義します。
JGroupDSProtocols	Decision Server の転送チャンネルに使用される JGroups のプロトコルとプロパティを定義します。
SDGroupName	クラスタの名前。
LearningServiceInitialWait	最初にサーバーが起動したときに、ラーニング・サービスが起動されるまで待機する時間（ミリ秒）。
LearningServiceRestartWait	マシンに障害が発生した後、またはマシンがクラスタから削除された後、アラート・サービスが再起動されるまで待機する時間（ミリ秒）。
AlertServiceInitialWait	最初にサーバーが起動したときに、アラート・サービスが起動されるまで待機する時間（ミリ秒）。
AlertServiceRestartWait	マシンに障害が発生した後、またはマシンがクラスタから削除された後、アラート・サービスが再起動されるまで待機する時間（ミリ秒）。
OperationalDataCleanupPeriod	データベースの操作データ（選択履歴、統計、学習データ・ストレージ）のクリーンアップの時間間隔（端数の指定も可能）。
StatisticsCleanupChunkSize	古い統計レコードを削除する場合に使用するチャンク・サイズ。
ChoiceHistoryCleanupChunkSize	古い選択履歴レコードを削除する場合に使用するチャンク・サイズ。
LearningDataStorageCleanupChunkSize	古い学習データ・ストレージ・レコードを削除する場合に使用するチャンク・サイズ。
StatisticsCleanupThrottle	0.1～1の数字を指定します。より大きいスロットル値は、より高速な処理に対応します。
ChoiceHistoryCleanupThrottle	0.1～1の数字を指定します。より大きいスロットル値は、より高速な処理に対応します。
LearningDataStorageCleanupThrottle	0.1～1の数字を指定します。より大きいスロットル値は、より高速な処理に対応します。

10.3.4 「OracleRTD」 → 「SDClusterPropertyManager」 → 「Deployment」 について

「SDClusterPropertyManager」 → 「Deployment」 MBean には、次の属性があります。

属性	説明
AppPollingInterval	AppFactory が SDApps テーブルをポーリングして、新しいアプリケーションの存在を確認する頻度（秒）。

10.3.5 「OracleRTD」 → 「SDClusterPropertyManager」 → 「Alert Service」 について

「SDClusterPropertyManager」 → 「Alert Service」 MBean には、次の属性があります。

属性	説明
MailFrom	電子メールのアラート通知の送信元アドレス。
MailLogo	電子メールのアラート通知のヘッダーに表示するロゴ。
AlertHistoryCleanupIntervalMinutes	古いアラートの有効期限を確認するためにアラート履歴のアーカイブをスキャンする頻度。分単位で指定します。
AlertDispatcherSleepIntervalMinutes	アラート・サービスでアラートを評価する頻度。分単位で指定します。
AlertDispatcherErrorSleepIntervalMinutes	エラーが発生した場合に、アラート・サービスがアラートの評価を再実行する頻度。分単位で指定します。

10.4 Oracle Real-Time Decisions のメンバー・レベルの管理

10.4.1 「OracleRTD」 → 「SDManagement」 → 「SDLogging」 について

「SDManagement」 → 「SDLogging」 MBean には、次の属性があります。

属性	説明
Priority	com.sigmadynamics カテゴリの現在のロギング・プライオリティ。有効な値は、DEBUG、ERROR、INFO および WARN です。
InlineServicePriority	インライン・サービスの現在のロギング・プライオリティ。有効な値は、DEBUG、ERROR、INFO および WARN です。
LogFile	現在のログ・ファイルの場所。

ログ・ファイルに記録される値の詳細は、『Oracle Real-Time Decisions Decision Studio リファレンス・ガイド』のインライン・サービスのトラブルシューティングとデバッグに関する項を参照してください。

10.4.2 「OracleRTD」 → 「SDPropertyManager」 → 「Performance Monitoring」について

「SDPropertyManager」 → 「Performance Monitoring」 MBeanには、次の属性があります。

属性	説明
DSPerfCounterEnabled	DSパフォーマンス・カウンタの書き込みを有効にします。ただし、ファイルのサイズが大きくなりすぎるため、有効化したままにはしないでください。
DSPerfCounterAppend	trueの場合、パフォーマンス・データが既存のファイル（存在する場合）に付加されます。falseの場合、サーバーの再起動時に既存のファイルが上書きされます。
DSPerfCounterLogFile	DSパフォーマンス・カウントが定期的に付加されるタブ区切り CSV ファイル。MS Excelが利用可能な場合、インストールのetcディレクトリにあるds_perf.xlsを使用すると、見やすく表示されます。
DSPerfCounterLogInterval	DSパフォーマンス・カウントの更新間隔（ミリ秒）。

Performance Monitoring の使用方法の詳細は、『Oracle Real-Time Decisions Decision Studio リファレンス・ガイド』のPerformance Monitoringの使用に関する項を参照してください。

10.4.3 「OracleRTD」 → 「SDPropertyManager」 → 「Misc」について

「SDPropertyManager」 → 「Misc」 MBeanには、次の属性があります。

属性	説明
DecisionServiceEnabled	このインスタンスでデシジョン・サービスを実行するかどうかを指定します。
LearningServiceEnabled	このインスタンスでラーニング・サービスを実行するかどうかを指定します。
AlertServiceEnabled	このインスタンスでアラート・サービスを実行するかどうかを指定します。
WorkerThreadPoolSize	通常の統合点リクエストの処理ではなく、一般的なメンテナンス・アクティビティに使用されるスレッド数。メンテナンス・アクティビティには、モデルのメンテナンス、セッション時間の操作、およびタイムアウトしたリクエストの処理が含まれます。

属性	説明
NumCPUs	このホストの CPU 数。Windows では、ハイパースレッド・コンピュータの場合、「Windows タスク マネージャ」に表示されるプロセッサ数の半分を使用します。
IntegrationPointRequestIOFactor	統合点リクエストが、I/O 処理に費やす時間、またはこの VM の外部システムを待機する時間のパーセンテージ。
IntegrationPointMaxConcurrentJobs	現在実行している統合点リクエストの最大数。通常は 0 に設定する必要があります。この場合、値は次のように計算されます。Math.ceil は、次に大きい整数値に丸められることを意味します。NumCPUs * Math.ceil(1/(1-DSRequestIOFactor)) + 5
IntegrationPointQueueSize	実行を待機する統合点リクエストの最大数。リクエストがこの値を超えると、サーバーで、「Server Too Busy」エラー・メッセージとともにリクエストが終了します。この属性は、サブレット・コンテナによって割り当てられるサブレット・スレッドの数から IntegrationPointMaxConcurrentJobs の構成値または計算値を引いた値以下に設定する必要があります。IntegrationPointMaxConcurrentJobs の計算値は、「SDConsole」→「Members」→「Decision Service」で確認できます。
WebServerPort	Web サーバーのポート。プロパティの変更を有効にするには、サーバーを再起動する必要があります。このポートは、複数の RTD サーバー間の内部通信に使用されます。
HTTPSEnabled	Web サーバー・ポートで HTTPS を許可するかどうかを指定します。プロパティの変更を有効にするには、サーバーを再起動する必要があります。

10.4.4 「OracleRTD」 → 「Server」 → 「DecisionService」について

「Server」 → 「DecisionService」 MBean には、次の読み取り専用の属性があります。

属性	説明
MaxAllowedConcurrentRequests	同時に実行できるリクエストの最大数。
CurrentRequestsRunning	現在実行されているリクエストの数。
CurrentRequestsQueued	現在実行されずに、待機しているリクエストの数。
TotalRequests	サーバーの起動後に出現したリクエストの合計数。

属性	説明
TimedOutRequests	タイムアウトになったリクエストの合計数。
RequestsForwarded	クラスタ内の別のサーバーに転送されたリクエストの合計数。
RequestQueueCapacity	同時に待機できるリクエストの最大数。
RequestsQueued	実行されるまでに待機する必要があったリクエストの合計数。
PeakRequestsQueued	一度でも待機する必要があったリクエストの最大数。
RequestsWhenQueueFull	リクエストのキューが一杯であったため、拒否されたリクエストの合計数。
CurrentSessions	まだオープンしているデシジョン・サービス・セッションの数。
TotalSessions	作成されたデシジョン・サービス・セッションの合計数。

10.5 Oracle Real-Time Decisions のセキュリティの管理

セキュリティ管理の MBean は、クラスタの認証設定に影響を与えます。「Security Authenticator」で設定するタイプによって、異なるセキュリティ設定が表示されます。

10.5.1 「OracleRTD」 → 「SDManagement」 → 「SecurityManager」について

「SDManagement」 → 「SecurityManager」 MBean には、次の操作があります。

addGroupMember()

パラメータ: group Name: メンバーを追加するグループの名前。
username: 追加するユーザーの名前。

目的: ユーザーまたはグループをグループの直接のメンバーにします。

assignPermission()

パラメータ: userOrGroup: 権限を割り当てるユーザー名またはグループ名。
permCode: 権限のコード。

目的: 指定されたクラスタの権限を割り当てます。詳細は、第6.5項「ユーザーおよびグループへの権限の割当て」を参照してください。

createGroup()

パラメータ: groupName: 作成するグループの名前。
description: 新しいグループの説明。

目的: 新しいセキュリティ・グループを作成します。Oracle RTD Platform Authentication のみ。

createUser()

パラメータ: username: 作成するユーザーの名前。
description: 新しいユーザーの説明。
password: クリア・テキストでのユーザーのパスワード。

目的: 新しいユーザーを作成します。Oracle RTD Platform Authentication のみ。

deleteGroup()

パラメータ: groupName: 削除するグループの名前。

目的: セキュリティ・グループを削除します。Oracle RTD Platform Authentication と Siebel Object Manager Authorization のみ。

deleteUser()

パラメータ: username: 削除するユーザーの名前。

目的: 既存のユーザーを削除します。Oracle RTD Platform Authentication のみ。

listDirectGroupMembers()

パラメータ: groupName: ターゲット・グループの名前。
includeGroups: True または False。True の場合、メンバー・グループが含まれます。False の場合、ユーザーのみが含まれます。

目的: 指定されたグループのメンバーとしてユーザーおよびグループの名前を直接返します。

listDirectlyContainingGroups ()

パラメータ: userOrGroup: ターゲット・ユーザーまたはグループの名前。

目的: 直接のメンバーとして、指定されたユーザーまたはグループが含まれるグループの名前を返します。

listDirectPermissions()

パラメータ: userOrGroup: ターゲット・ユーザーまたはグループの名前。

目的: 特定のユーザーまたはグループに直接割り当てられた権限を返します。

listEffectivePermissions()

パラメータ: userOrGroup: ターゲット・ユーザーまたはグループの名前。

目的: 特定のユーザーまたはグループ、あるいはそれらが直接的または間接的に含まれるグループに直接割り当てられた権限を返します。

listEveryoneHavingDirectPermissions()

パラメータ: なし

目的: 権限が直接割り当てられたユーザーおよびグループの名前を返します。

listGroupNames()

パラメータ: なし

目的: 権限を割り当てることができるすべてのグループの名前を返します。

listPermissionCodes()

パラメータ: なし

目的: すべての使用可能な権限タイプのローカライズされた名前と整数コードを返します。このコードは、権限タイプの指定を必要とする様々な操作のオペランドとして使用されます。

listUserNames()

パラメータ: なし

目的: 権限を割り当てることができるすべてのユーザーの名前を返します。

removeAllDirectPermissions ()

パラメータ: userOrGroup: ターゲット・ユーザーまたはグループの名前。

目的: 指定されたユーザーまたはグループに直接割り当てられたすべての権限を削除します。

removeGroupMember()

パラメータ: containingGroupName: メンバー・グループを含むグループ。
memberGroupName: 別のグループのメンバーとして追加されるグループ。

目的: 包含するグループからユーザーまたはグループを削除します。

removePermission()

パラメータ: userOrGroup: ターゲット・ユーザーまたはグループの名前。
permCode: 削除する権限のコード。

目的: ユーザーまたはグループに直接割り当てられた権限を削除します。指定されたユーザーまたはグループが含まれるグループからは権限を削除しません。

renameGroup()

パラメータ: oldGroupName: グループの以前の名前。
newGroupName: グループの新しい名前。

目的: グループの名前を変更します。

renameUser()

パラメータ: oldUserName: ユーザーの以前の名前。
newUserName: ユーザーの新しい名前。

目的: ユーザーの名前を変更します。

setPassword()

パラメータ: username: ターゲット・ユーザーの名前。
password: クリア・テキストでの新しいパスワード。

目的: 特定のユーザーのパスワードを変更します。

10.5.2 「OracleRTD」 → 「SDClusterPropertyManager」 → 「Security Properties」について

「SDClusterPropertyManager」 → 「Security Properties」 MBean には、次の属性があります。

属性	説明
AuthenticationEnabled	false の場合、パスワードのチェックおよび権限のチェックが無効になります。
AuthenticatorLogsInvalidCredentials	true の場合、ログインの失敗が無効な資格証明によるものと判断されたときは、必ず Authenticator が短いメッセージをログに記録します。
AuthenticatorLogsFailures	true の場合、Authenticator は、例外スタック・トレースを含む、すべての失敗したログインに関する詳細をログに記録します。この詳細レベルは、セキュリティ・プロバイダによって、エラーの原因が構成の問題にあると判断された場合に無条件で表示されます。またこのフラグでは、プロバイダによって、エラーの原因が無効な資格証明にあると誤って判断されたときにも同様の詳細レベルが表示されます。

属性	説明
AuthenticationProviderClass	ログインやグループ・メンバーシップ・クエリーを実行し、オプションでユーザーとグループの定義を管理するクラスの名前。com.sigmadynamics.server.security.WindowsAuthenticator、com.sigmadynamics.server.security.DBAuthenticator または com.sigmadynamics.server.security.SiebelAuthenticator を使用します。

10.5.3 「OracleRTD」 → 「SDManagement」 → 「SecurityAuthenticator」 について

「SDManagement」 → 「SecurityAuthenticator」 MBean には、次の読取り専用の属性があります。

属性	説明
SecurityManager	セキュリティ・マネージャ。
ProviderClass	認証プロバイダ・クラス。

「SDManagement」 → 「Security Authenticator」 MBean には、次の操作があります。

java.lang.String loginUserPassword()

パラメータ: username: ユーザーの名前。
password: クリア・テキストでのユーザーのパスワード。

目的: ユーザー名とパスワードを使用してログインし、ログイン・チケットを取得します。

java.lang.String getUserName()

パラメータ: ticket: ログインから返されるチケット。

目的: ログイン・セッションに関連付けられたユーザー名を取得します。

java.lang.Boolean isUserInRole()

パラメータ: ticket: ログインから返されるチケット。
userOrGroupName: ID または包含関係がそれぞれテストされたユーザーまたはグループの名前。

目的: ログインしたユーザーが指定されたユーザーの場合、または指定されたグループに直接的または間接的に含まれる場合、true を返します。

java.lang.Void logout()

パラメータ: ticket: ログインから返されるチケット。

目的: ログイン・セッションを終了します。

10.6 インライン・サービスの管理

Inline Service Manager を使用して、クラスタにデプロイされているインライン・サービスを管理します。

10.6.1 「OracleRTD」 → 「SDManagement」 → 「InlineServiceManager」 について

デプロイされた各インライン・サービスは、InlineServiceManager MBean の下に表示されます。

「SDManagement」 → 「InlineServiceManager」 MBean には、次の操作があります。

removeAllServices ()

パラメータ: なし

目的: すべてのインライン・サービス (loaded、loadable、failed) を削除します。

10.6.2 「OracleRTD」 → 「InlineServiceManager」 → 「インライン・サービス」 → 「デプロイ状態」 → 「フラグ」について

InlineServiceManager MBean を表示するには、インライン・サービスの名前を選択し、デプロイ状態 (たとえば、Development、QA または Production) を選択して、特定のフラグ (Failed、Inactive、Loadable) を選択します。たとえば、「OracleRTD」 → 「InlineServiceManager」 → 「CrossSell」 → 「Production」 → 「Loadable」 を選択します。

特定のインライン・サービスの InlineServiceManager MBean には、次の属性があります。

属性	説明
ServiceId	サービス ID。
DeploymentState	デプロイ状態。
Flag	フラグ。
LockStatus	ロック状態。

特定のインライン・サービスの InlineServiceManager MBean には、次の操作があります。

操作名	パラメータ	目的
unlockService()	なし	このサービスのロックを解除します。
removeService()	なし	このサーバーのインライン・サービスを停止し、データベースからサービスを削除します。

操作名	パラメータ	目的
flushStatistics()	なし	このサービスのすべての統計をデータベースにフラッシュします。
deleteStatistics()	なし	データベースでこのサービスのすべての統計をフラッシュおよび削除します。
deleteChoiceHistory()	なし	データベースからこのサービスのすべての選択履歴を削除します。
deleteAllOperationalData()	なし	データベースからこのサービスのすべての操作データを削除します。これには、選択履歴、統計およびスタディが含まれます。
deleteStudy()	なし	このサービスのスタディを削除します。

10.7 デプロイ状態の管理

「OracleRTD」 → 「DeploymentStates」 MBean を表示すると、クラスタで利用可能なデプロイ状態のリストを表示できます。「OracleRTD」 → 「SDManagement」 → 「DeploymentStates」 MBean では、デプロイ状態の追加、編集、削除および並替えを実行できます。

10.7.1 「OracleRTD」 → 「SDManagement」 → 「DeploymentStates」 について

「SDManagement」 → 「DeploymentStates」 MBean には、次の属性があります。

属性	説明
StateObjectNames	サーバーで利用可能なすべてのデプロイ状態のリスト。

「SDManagement」 → 「Deployment States」 MBean には、次の操作があります。

addDeploymentState ()

パラメータ: Name: デプロイ状態の名前。
 AfterDeploymentState: 新しいデプロイ状態が作成される直前のデプロイ状態の名前（値が渡されていない場合、新しいデプロイ状態はリストの最初に作成されます）。
 AllowHotSwapping: インライン・サービスのホット・スワップを許可します。このデプロイ状態にホット・スワップを許可し、その状態でインライン・サービスを再デプロイすると、既存のセッションがクローズするかタイムアウトになるまで、既存のインライン・サービスはアクティブのままになります。新しくデプロイされたインライン・サービスには、新しいセッションが作成されます。

目的: 新しいデプロイ状態を作成します。

10.7.2 「OracleRTD」 → 「Deployment States」 → 「状態」について

特定のデプロイ状態の MBean を表示するには、「OracleRTD」 → 「Deployment States」を選択して、デプロイ状態（Development、QA、Production など）を選択します。

「Deployment States」 → 「状態」の各 MBean には、次の属性があります。

属性	説明
Name	デプロイ状態の名前。
Id	デプロイ状態の ID。
AllowHotSwapping	デシジョン・サービスのこのデプロイ状態でのインライン・サービスのホット・スワップを許可します。

「Deployment States」 → 「状態」の各 MBean には、次の操作があります。

`remove ()`

パラメータ: なし

目的: クラスタからデプロイ状態を削除します。

10.8 ラーニング・サービスの管理

クラスタのラーニング・サービスを管理することにより、ラーニング・モデルのステータスを確認し、そのモデルの保守を実行できます。

10.8.1 「OracleRTD」 → 「Server」 → 「LearningService」について

「Server」 → 「LearningService」 MBean には、次の属性があります。

属性	説明
Studies	Learning Server で実行されているすべてのスタディのリスト。スタディをクリックすると、そのスタディのモデルが表示されます。

10.8.2 「OracleRTD」 → 「Learning Server」 → 「スタディ」について

「Learning Server」 → 「スタディ」 MBean には、次の属性があります。

属性	説明
Models	このスタディに属しているモデル。

属性	説明
Name	スタディの名前。

「Learning Server」 → 「スタディ」 MBeanには、次の操作があります。

Delete()

パラメータ: なし

目的: このスタディを削除します。

10.8.3 「OracleRTD」 → 「Study」 → 「スタディ名」 → 「モデル」について

「Study」 → 「スタディ名」 → 「モデル」 MBeanには、次の属性があります。

属性	説明
Attributes	モデルの属性の名前。ここにリストされた名前は、インライン・サービスのセッションの属性名と一致します。
Name	モデルの名前。
TimeWindows	この特定のモデルによって学習された時間範囲のリスト。

「Study」 → 「スタディ名」 → 「モデル」 MBeanには、次の操作があります。

Delete()

パラメータ: なし

目的: このモデルを削除します。

DeleteAttributeValue()

パラメータ: AttributeName: 属性の名前。
Value: 削除する値。

目的: 属性の値について収集されたモデルのデータを消去します。

DeleteAttributeValueRange()

パラメータ: AttributeName: 属性の名前。
HighestValue: 削除する最も高い値。
LowestValue: 削除する最も低い値。

目的: 属性の値の範囲について収集されたモデルのデータを消去します。

StartNewTimeWindow()

パラメータ: なし

目的: 現在の時間枠を閉じて、新しい時間枠を開始します。この操作は、将来のモデル学習に悪影響を与える可能性があるため、本番環境では使用しないでください。

10.9 アラート・サービスの管理

「SDClusterPropertyManager」 → 「Alert Service」 MBean を使用すると、アラート・サービスに関連したパラメータを調整できます。

10.9.1 「OracleRTD」 → 「SDClusterPropertyManager」 → 「Alert Service」 について

「SDClusterPropertyManager」 → 「Alert Service」 MBean には、次の属性があります。

属性	説明
MailFrom	電子メールのアラート通知の送信元アドレス。
MailLogo	電子メールのアラート通知のヘッダーに表示するロゴ。
AlertHistoryCleanupIntervalMinutes	古いアラートの有効期限を確認するためにアラート履歴のアーカイブをスキャンする頻度。分単位で指定します。
AlertDispatcherSleepIntervalMinutes	アラート・サービスでアラートを評価する頻度。分単位で指定します。
AlertDispatcherErrorSleepIntervalMinutes	エラーが発生した場合に、アラート・サービスがアラートの評価を再実行する頻度。分単位で指定します。