

Oracle® Sensor Edge Server

ガイド

10g (10.1.3.1.0)

部品番号 : B31872-01

2006 年 12 月

Oracle Sensor Edge Server ガイド, 10g (10.1.3.1.0)

部品番号 : B31872-01

原本名 : Oracle Sensor Edge Server Guide, 10g (10.1.3)

原本部品番号 : B28979-01

原本著者 : John Bassett

原本協力者 : Robin Clark, Joseph Garcia, Samuelson Rehman, Anit Chakraborty, James Chase, Ron Caneel, Greg Grisco

Copyright © 2004, 2006 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 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えます。

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。

目次

はじめに	vii
対象読者	viii
ドキュメントのアクセシビリティについて	viii
表記規則	viii
サポートおよびサービス	ix
1 インストール	
ハードウェア要件	1-2
ソフトウェア要件	1-2
OC4J のインストール	1-3
Oracle Sensor Edge Server のインストール、または OC4J 10g リリース 2 (10.1.2) インスタンスからのアップグレード	1-3
OC4J 10g リリース 3 (10.1.3) インスタンスからの Oracle Sensor Edge Server のインストール	1-4
インストール・ログ・ファイルの調査	1-5
Oracle Sensor Edge Server のアンインストール	1-5
インストールの確認	1-5
Sensor Edge Server コンソールの表示	1-5
データベース接続のテスト	1-6
Enterprise Manager を使用したデータベース接続のテスト	1-7
Oracle Sensor Edge Mobile のインストール	1-8
Pocket PC デバイスでの Oracle Sensor Edge Mobile のインストールと起動	1-8
デフォルトのデバイス構成の変更	1-10
RFID タグの読取り	1-10
バーコード・データの読取り	1-10
Sensor Edge Mobile サービスの停止	1-10
Oracle Sensor Edge Mobile エミュレータのインストール	1-10
リリース 3 (10.1.3) のセンサー・データ・リポジトリおよびセンサー・データ・ストリームの 手動構成	1-11
センサー・データ・リポジトリの手動配置	1-11
センサー・データ・リポジトリへの Oracle Sensor Edge Server の接続	1-12
センサー・データ・ストリームの手動配置	1-13
センサー・データ・ストリームへの Oracle Sensor Edge Server の接続	1-14
リリース 2 (10.1.2) の既存のセンサー・データ・リポジトリへの接続	1-15
リリース 2 (10.1.2) の既存のセンサー・データ・ストリームへの接続	1-15

2 Oracle Sensor Edge Server の概要

リリース 3 (10.1.3) の新機能	2-2
Oracle Sensor Edge Server コンソール	2-2
Oracle Sensor Edge Mobile	2-3
Oracle Application Server Control を使用した拡張管理	2-3
向上したパフォーマンス	2-3
拡張されたセンサー・データ・リポジトリ	2-3
センサー・データ・ストリーム	2-4
EPC コンプライアンス統合	2-4
トランスポート・レイヤー	2-4
拡張されたセキュリティ	2-4
デバイス・コントローラの廃止	2-4
Oracle Sensor Edge Server の概略	2-4
センサー・データの収集	2-5
センサー・データのフィルタリング	2-5
センサー・データのディスパッチ	2-5
センサー・データ・アーカイブおよびルール	2-6
Sensor Server とデバイス管理	2-6
Sensor Edge Mobile	2-6
SES コンソール	2-6
Oracle Sensor Edge Server のアーキテクチャ	2-6
デバイス・ドライバ	2-7
デバイス・グループ	2-7
ローカル処理	2-7
イベント・プロセッサ	2-7
ドライバ・マネージャ	2-7
Oracle Sensor Edge Mobile のアーキテクチャ	2-8
デバイス・ドライバのサポート	2-9
Oracle Sensor Edge Mobile の管理	2-9
サンプル・コードおよびデモ・アプリケーション	2-9
配置に関する考慮事項	2-10
ネットワーク特性の確認	2-10
データ・センター環境の特定	2-10
リーダーおよびセンサーの場所の確認	2-10
Edge Server の場所の選択	2-10
Oracle Sensor Edge Server およびセンサー・データ・リポジトリの考慮事項	2-11

3 Oracle Sensor Edge Server の管理

Oracle Sensor Edge サービス管理の概要	3-2
Oracle Sensor Edge Server インスタンスの管理	3-3
他の Oracle Sensor Edge Server インスタンスへのアクセス	3-4
Oracle Sensor Edge Server インスタンスのエントリの作成	3-5
Oracle Sensor Edge Server インスタンスのエントリの編集	3-5
Oracle Sensor Edge Server インスタンスのパフォーマンスの監視	3-6
イベント・データのキューのクリア	3-6
Oracle Sensor Edge Server インスタンスの一般情報の設定	3-7
Oracle Sensor Edge Server インスタンスのディスパッチャの設定	3-8

ディスペッチャ、ドライバおよびフィルタの表示	3-9
Oracle Sensor Edge Server で使用するデバイスおよびフィルタの設定	3-9
Oracle Sensor Edge Server のデバイス・グループの表示	3-10
デバイス・グループの作成	3-11
デバイス・グループへのフィルタの追加	3-17
デバイス・グループの編集	3-18
デバイス・グループの名前変更	3-18
デバイス・グループで使用するデバイスおよびフィルタの更新	3-18
デバイス・グループに割り当てられているデバイスの起動と停止	3-18
デバイス・グループの削除	3-18
Oracle Sensor Edge Server インスタンスの起動と停止	3-19
opmnctl を使用した Oracle Sensor Edge Server インスタンスの停止と起動	3-19
OracleAS Enterprise Manager を使用した Oracle Sensor Edge Server インスタンスの再起動	3-19
各デバイスの起動と停止	3-21
フィルタの管理	3-21
フィルタ・インスタンスの優先度の設定	3-22
デバイスまたはデバイス・グループのフィルタ・インスタンスの管理	3-22
イベント・データの監視	3-24
イベント・データの表示	3-24
各イベントの表示	3-24
未処理のイベント・データの表示	3-29
ログ情報の表示	3-30
処理済のイベント・データの表示	3-31
タグ ID によるイベントの検索	3-31
デバイス名によるイベントの検索	3-32
タグ ID およびデバイス名の再検索	3-32
拡張検索の作成	3-33
Oracle Sensor Edge Server インスタンスの拡張機能の追加	3-34
拡張アーカイブ・ファイル	3-35
拡張アーカイブ・ファイルのパッケージ化	3-37
拡張機能のアップロード	3-37
拡張クラスの階層	3-38
拡張機能の実装	3-39
拡張機能のコンテキスト	3-39
インスタンスに関する情報の取得	3-39
インスタンスのランタイム・コンテキストへのアクセス	3-39
インスタンスのパラメータの管理	3-40
カスタム・パラメータの公開	3-40
パラメータ値の取得	3-40

4 センサー・データ・リポジトリの使用

センサー・データ・リポジトリの概要	4-2
リレーショナル表	4-2
センサー・データ・リポジトリ内のリレーショナル・ビュー	4-2
センサー・データ・リポジトリの PL/SQL パッケージ	4-3
リポジトリに対する操作と問合せ	4-3
リポジトリの作成と削除	4-3
リポジトリへの測定値の保存	4-3

アーカイブの間合せ	4-4
スキーマ・リファレンス	4-4
表	4-4
EDG_CAP_TAB 表	4-4
EDG_CTXT_REL_TAB 表	4-5
EDG_CTXT_TAB 表	4-5
EDG_DEVICE_TAB 表	4-6
EDG_DIAG_TAB 表	4-6
EDG_EVENT_INFO_TAB 表	4-7
EDG_EVENT_TAB 表	4-8
EDG_LOG 表	4-8
EDG_TAG_TAB 表	4-9
ビュー	4-9
EDG_CAP	4-10
EDG_CTXT	4-10
EDG_CTXT_REL	4-10
EDG_CTXT_REL_NAME_VW	4-11
EDG_DEVICE	4-11
EDG_DEV_CAP_VW	4-12
EDG_DEV_DIAG_VW	4-13
EDG_DEV_EVENT_VW	4-13
EDG_DEV_LAST_DIAG_VW	4-14
EDG_DEV_LAST_OBSV_VW	4-15
EDG_DIAG	4-16
EDG_EVENT	4-17
EDG_EVENT_INFO	4-17
EDG_EVENT_VW	4-18
EDG_TAG	4-19
EDG_TAG_LAST_DEV_VW	4-19
EDG_TAG_PATH_VW	4-20
PL/SQL プログラミング・インタフェース	4-22
EDG_SDA パッケージ	4-22

5 Oracle Sensor Edge Mobile

Oracle Sensor Edge Mobile の概要	5-2
アプリケーションへの Sensor Edge Mobile の接続	5-3
ディスペッチャとドライバの構成	5-3
キーボード・ディスペッチャの構成	5-4
DestinationApplication パラメータの定義	5-5
RFIDReadMacro パラメータの定義	5-6
BarcodeReadMacro パラメータの定義	5-6
キー・シーケンス・マクロ・パラメータの定義	5-6
キー・シーケンス・マクロの作成	5-7
キー・シーケンス・マクロによる繰返し要素処理の有効化	5-7
特殊な制御シーケンスのキー・マクロ要素キー	5-8
制御キーまたはデータ配置のためのキーボード・マクロ要素	5-9
デバイス・ステータスの確認	5-10
ActiveX アプリケーション・インタフェース	5-11
オブジェクト宣言	5-11
rfid_read()	5-12

rfid_write()	5-12
rfid_kill()	5-12
barcode_read()	5-12
set_trigger_rfid_read()	5-12
set_trigger_barcode_read()	5-13
process_instruction()	5-13
is_supported()	5-14
測定イベントの処理	5-14
廃止された Activex アプリケーション・インタフェース	5-16
Sensor Edge Mobile の管理	5-16
国際化	5-18

6 デバイス、フィルタ・インスタンスおよびディスパッチャの構成

デバイス、フィルタ・インスタンスおよびディスパッチャの構成の概要	6-2
デバイスおよびディスパッチャの URI パラメータの設定	6-2
デバイスの構成	6-4
Alien リーダー・ドライバ・ベースのデバイスの構成	6-6
Alien リーダー・ドライバにより生成される測定イベント	6-6
Alien リーダー・ドライバにより受け入れられる命令イベント	6-7
AnimationDriver のインスタンスの構成	6-7
BarcodeDriver ベースのデバイスの構成	6-9
BarcodeDriver により返される RFID 測定イベント	6-9
ConsoleDriver のインスタンスの構成	6-9
Edge Echo ドライバ・ベースのインスタンスの構成	6-9
Edge Simulator ドライバ・ベースのデバイスの構成	6-10
HtmlDriver インスタンスの構成	6-12
HtmlDriver によりサポートされるイベント	6-13
Intermec BRI ドライバによりサポートされるデバイスの構成	6-14
Intermec BRI ドライバにより生成される測定イベント	6-15
Intermec BRI ドライバにより受け入れられる命令イベント	6-15
Intermec リーダー・ドライバ・ベースのデバイスの構成	6-16
LpmlDriver ベースのデバイスの構成	6-16
LpmlDriver により生成されるプリンタ・レスポンス測定イベント	6-17
LpmlDriver により受け入れられる命令イベント	6-17
Matrics ドライバ・ベースのデバイスの構成	6-17
Matrics ドライバにより生成される測定イベント	6-18
Matrics ドライバにより受け入れられる命令イベント	6-18
PatliteDriver ベースのデバイスの構成	6-19
Prolite ドライバ・ベースのデバイスの構成	6-19
Samsys ドライバ・ベースのデバイスの構成	6-19
Samsys ドライバにより生成される測定イベント	6-20
Samsys ドライバにより受け入れられる命令イベント	6-20
Simple Audio ドライバ・インスタンスの構成	6-21
Simple Audio ドライバによりサポートされるオーディオ・イベント	6-21
Tyco リーダー・ドライバ・ベースのデバイスの構成	6-22
Tyco リーダー・ドライバにより生成される測定イベント	6-23
Tyco リーダー・ドライバにより受け入れられる命令イベント	6-23

フィルタ・インスタンスの構成	6-24
Check Tag ID フィルタの構成	6-24
Cross-Reader Redundant フィルタの使用	6-25
Debug フィルタの使用	6-25
JavaScript フィルタの構成	6-26
Movement フィルタの構成	6-27
Pallet Pass Thru フィルタの構成	6-27
Pallet Shelf フィルタの構成	6-28
Pallet Shelf フィルタが生成するイベント	6-29
Pass フィルタの構成	6-30
Polygon フィルタの構成	6-31
Regex フィルタの定義	6-31
Shelf フィルタの構成	6-32
Shelf フィルタが生成するイベント	6-32
Oracle Sensor Edge Server インスタンス用のディスパッチャの管理	6-33
Web サービスにイベント・メッセージを送信するためのディスパッチャの構成	6-34
HTTP を使用してイベント・メッセージを送信するためのディスパッチャの構成	6-34
PML ディスパッチャの構成	6-34
テンプレート・ディスパッチャの構成	6-35
ALEDispatcher の構成	6-35
Null ディスパッチャの使用	6-35
Oracle Streams を使用するためのエッジ・ディスパッチャの構成	6-35

用語集

索引

はじめに

『Oracle Sensor Edge Server ガイド』では、Oracle Sensor Edge Server 製品について、およびそのインストールと構成について説明します。Sensor Edge Server アプリケーションの開発の詳細は、OTN にある Sensor Edge Server のドキュメントを参照してください。

対象読者

このマニュアルは、Oracle Sensor Edge Server を使用する顧客を対象としています。

ドキュメントのアクセシビリティについて

オラクル社は、障害のあるお客様にもオラクル社の製品、サービスおよびサポート・ドキュメントを簡単にご利用いただけることを目標としています。オラクル社のドキュメントには、ユーザーが障害支援技術を使用して情報を利用できる機能が組み込まれています。HTML 形式のドキュメントで用意されており、障害のあるお客様が簡単にアクセスできるようにマークアップされています。標準規格は改善されつつあります。オラクル社はドキュメントをすべてのお客様がご利用できるように、市場をリードする他の技術ベンダーと積極的に連携して技術的な問題に対応しています。オラクル社のアクセシビリティについての詳細情報は、Oracle Accessibility Program の Web サイト <http://www.oracle.com/accessibility/> を参照してください。

ドキュメント内のサンプル・コードのアクセシビリティについて

スクリーン・リーダーは、ドキュメント内のサンプル・コードを正確に読めない場合があります。コード表記規則では閉じ括弧だけを行に記述する必要があります。しかし JAWS は括弧だけの行を読まない場合があります。

外部 Web サイトのドキュメントのアクセシビリティについて

このドキュメントにはオラクル社およびその関連会社が所有または管理しない Web サイトへのリンクが含まれている場合があります。オラクル社およびその関連会社は、それらの Web サイトのアクセシビリティに関しての評価や言及は行っておりません。

Oracle サポート・サービスへの TTY アクセス

アメリカ国内では、Oracle サポート・サービスへ 24 時間年中無休でテキスト電話 (TTY) アクセスが提供されています。TTY サポートについては、(800)446-2398 にお電話ください。

表記規則

このマニュアルでは次の表記規則を使用します。

規則	意味
太字	太字は、操作に関連する Graphical User Interface 要素、または本文中で定義されている用語および用語集に記載されている用語を示します。
イタリック	イタリックは、ユーザーが特定の値を指定するプレースホルダ変数を示します。
固定幅フォント	固定幅フォントは、段落内のコマンド、URL、サンプル内のコード、画面に表示されるテキスト、または入力するテキストを示します。

サポートおよびサービス

次の各項に、各サービスに接続するための URL を記載します。

Oracle サポート・サービス

オラクル製品サポートの購入方法、および Oracle サポート・サービスへの連絡方法の詳細は、次の URL を参照してください。

<http://www.oracle.co.jp/support/>

製品マニュアル

製品のマニュアルは、次の URL にあります。

<http://otn.oracle.co.jp/document/>

研修およびトレーニング

研修に関する情報とスケジュールは、次の URL で入手できます。

<http://www.oracle.co.jp/education/>

その他の情報

オラクル製品やサービスに関するその他の情報については、次の URL から参照してください。

<http://www.oracle.co.jp>

<http://otn.oracle.co.jp>

注意： ドキュメント内に記載されている URL や参照ドキュメントには、Oracle Corporation が提供する英語の情報も含まれています。日本語版の情報については、前述の URL を参照してください。

インストール

この章では、Oracle Sensor Edge Server のインストールについて説明します。この章の内容は次のとおりです。

- 「ハードウェア要件」
- 「ソフトウェア要件」
- 「OC4J のインストール」
- 「Oracle Sensor Edge Server のインストール、または OC4J 10g リリース 2 (10.1.2) インスタンスからのアップグレード」
- 「OC4J 10g リリース 3 (10.1.3) インスタンスからの Oracle Sensor Edge Server のインストール」
- 「インストールの確認」
- 「Oracle Sensor Edge Mobile のインストール」
- 「リリース 3 (10.1.3) のセンサー・データ・リポジトリおよびセンサー・データ・ストリームの手動構成」
- 「リリース 2 (10.1.2) の既存のセンサー・データ・リポジトリへの接続」
- 「リリース 2 (10.1.2) の既存のセンサー・データ・ストリームへの接続」

ハードウェア要件

Oracle Sensor Edge Server のハードウェア要件は、Oracle Application Server のハードウェア要件と同じです。具体的なハードウェア要件については、ご使用のプラットフォームに対応する『Oracle Application Server のインストール・ガイド』を参照してください。

Oracle Sensor Edge Mobile には、Pocket PC 2003 以上のデバイスが必要です。Oracle Sensor Edge Mobile には、RFID およびバーコードに対応する Symbol 9000G および Intermec IP3 デバイス用のデバイス・ドライバが含まれています。Sensor Edge Mobile には、150KB 未満のプログラム記憶域と 1MB 未満のメモリーが必要です。

メモリーおよびディスク領域の最小要件を表 1-1 に示します。

表 1-1 メモリーおよびディスク領域の最小要件

	(OC4J および Oracle Sensor Edge Server)	Oracle Sensor Edge Server
メモリー	512MB	128MB
ディスク領域	550MB	150MB
一時ディレクトリ領域	256MB	256MB
スワップ領域	1.5GB	1.5GB

ソフトウェア要件

Oracle Sensor Edge Server では次のソフトウェアが必要です。

- UNIX、Linux、Windows 2000、Windows XP またはそれ以上
- センサー・データ・ストリームおよびセンサー・データ・リポジトリには、エンタープライズ・バージョンの Oracle Database (リリース 9.2.0.6 以上) が必要です。Oracle Database 10g リリース 2 が推奨されます。
- センサー・データ・ストリームをインストールして使用するには、Oracle Database が ARCHIVELOG モードで実行されており、自動アーカイブが有効になっている必要があります。ARCHIVELOG モードに変更する手順、および自動アーカイブを有効にする手順については、該当するバージョンの『Oracle Database 管理者ガイド』を参照してください。これらの手順は使用している Oracle Database のバージョンによって異なるため、必ずご使用のバージョンに対応する『Oracle Database 管理者ガイド』を参照してください。
- Sun Java2 SDK、Standard Edition、バージョン 1.4.2 以上
- Oracle Containers for J2EE (OC4J) 10g リリース 3 (10.1.3) 以上

注意： OC4J 10g リリース 2 (10.1.2) インスタンスとの下位互換性のために、次のものがが必要です。

リリース 2 (10.1.2.x.x) の Companion CD OUI (先に OC4J のインストールが必要)

リリース 2 (10.1.2.x.x) スタンドアロン・バージョン (oc4j_extended.zip をディレクトリに解凍してインストール)。後述の OC4J のインストールを参照してください。

- J2EE Server and Process Management リリース 3 (10.1.3) (Oracle HTTP Server も Web Cache もなし)。
- その他のソフトウェア要件はすべて、Oracle Application Server のソフトウェア要件と同じです。

また、センサー・データ・リポジトリおよびセンサー・データ・ストリームには UTF-8 へのエンコーディング・セットが必要です。

Oracle Sensor Edge Mobile のソフトウェア要件は次のとおりです。

- Pocket PC 2003 以上
- Pocket IE 3.0.3 以上のブラウザ

OC4J のインストール

Oracle Sensor Edge Server をインストールする前に、OC4J をインストールする必要があります。以前のリリースでは、OC4J は Oracle Sensor Edge Server インストールの一部としてパッケージ化されていました。リリース 3 (10.1.3) では、OC4J のアップグレードを容易にするために、OC4J は Oracle Sensor Edge Server インストールの一部に含まれていません。このため、Oracle Sensor Edge Server をインストールする前に OC4J をインストールする必要があります。

Oracle Application Server のインストール時に OC4J をインストールできます。Oracle Application Server 用の Oracle Universal Installer で、「**拡張インストール・モード**」を選択し、「**J2EE Server およびプロセス管理**」を選択する必要があります。Oracle Application Server のインストール方法の詳細は、ご使用のプラットフォームに対応する Oracle Application Server のインストール・ガイドを参照してください。

OC4J のインストールを終了した後、Oracle Sensor Edge Server のインストールを開始できます。

下位互換性のためにリリース 2 (10.1.2) をインストールする場合は、次のインストール手順を実行します。

1. リリース 2 (10.1.2.x.x) の Companion CD OUI をインストールした後（または oc4j_extended.zip を解凍した後）、ORACLE_HOME/j2ee/home に移動します。
2. `java -jar oc4j.jar -install` を実行します。
3. パスワードを要求されたら、OC4J インスタンスの管理者パスワードを入力します。これは管理 UI へのログインに使用するパスワードで、Oracle Sensor Edge Server のインストールにも必要です。

Oracle Sensor Edge Server のインストール、または OC4J 10g リリース 2 (10.1.2) インスタンスからのアップグレード

OC4J 10g リリース 2 (10.1.2) インスタンスがすでにインストールされている場合、「[OC4J 10g リリース 3 \(10.1.3\) インスタンスからの Oracle Sensor Edge Server のインストール](#)」の手順に従って Oracle Sensor Edge Server をインストールできます。インストール中、Oracle Universal Installer によって OC4J 管理者パスワードを作成するように要求されます。これは、OC4J のインストールおよび管理に使用するパスワードです。

旧バージョンの Oracle Sensor Edge Server がすでにインストールされている場合、それをアップグレードすることはできません。「[OC4J 10g リリース 3 \(10.1.3\) インスタンスからの Oracle Sensor Edge Server のインストール](#)」の手順に従って、新しいバージョンをインストールする必要があります。その後、新しいバージョンの Oracle Sensor Edge Server 設定を手動で再作成する必要があります。

OC4J 10g リリース 3 (10.1.3) インスタンスからの Oracle Sensor Edge Server のインストール

OC4J 10g リリース 3 (10.1.3) インスタンスから Oracle Sensor Edge Server をインストールするには、次のようにします。

1. Companion CD を挿入し、Oracle Universal Installer を起動します。
2. 「ようこそ」画面が表示されたら、「次へ」をクリックします。
3. 「ファイルの場所の指定」画面の「インストール先」に、インストールに使用する名前および Oracle Sensor Edge Server のインストール先パスを入力します。Oracle Sensor Edge Server では OC4J と同じ `ORACLE_HOME` を使用する必要があります。「次へ」をクリックします。
4. 「インストール・タイプの選択」画面で「Sensor Edge Server」を選択し、「次へ」をクリックします。
5. 「J2EE コンテナ (OC4J) パスワード」画面で OC4J 管理者のパスワードを入力し、「次へ」をクリックします。これは、OC4J をインストールしたときに定義した管理者パスワードです。
6. Sensor Edge Server をセンサー・データ・リポジトリに接続する場合は、「Sensor Data Repository インストール」画面で「はい」を選択します。「はい」を選択した場合は、新しいセンサー・データ・リポジトリが作成されるか、既存のセンサー・データ・リポジトリに接続できるようになります。「いいえ」を選択した場合は、[10](#)に進みます。「いいえ」を選択し、後でセンサー・データ・リポジトリを使用する場合は、「[リリース 2 \(10.1.2\) の既存のセンサー・データ・リポジトリへの接続](#)」または「[リリース 3 \(10.1.3\) のセンサー・データ・リポジトリおよびセンサー・データ・ストリームの手動構成](#)」の手順に従って、手動でセンサー・データ・リポジトリを構成できます。選択した後、「次へ」をクリックします。
7. 「Sensor Data Repository の構成」画面で、新しいセンサー・データ・リポジトリに接続するか、既存のセンサー・データ・リポジトリに接続するかを選択し、「次へ」をクリックします。
8. 「SDR データベース接続情報」画面で、センサー・データ・リポジトリの必要な情報を入力し、「次へ」をクリックします。
9. 「Sensor Data Repository パスワード」画面で、センサー・データ・リポジトリ・スキーマに使用するパスワードを指定し、「次へ」をクリックします。(この画面は、[7](#)で既存のセンサー・データ・リポジトリに接続するように選択した場合は表示されません。)スキーマはデータベース内の `edge` ユーザーの下にインストールする必要があります。
10. Sensor Edge Server をセンサー・データ・ストリームに接続する場合は、「Sensor Data Streams インストール」画面で「はい」を選択します。「はい」を選択した場合は、新しいセンサー・データ・ストリーム・スキーマを作成するか、既存のセンサー・データ・ストリーム・スキーマに接続できるようになります。「いいえ」を選択した場合は、[14](#)に進みます。「いいえ」を選択し、後でセンサー・データ・ストリームを使用する場合は、「[リリース 2 \(10.1.2\) の既存のセンサー・データ・ストリームへの接続](#)」または「[リリース 3 \(10.1.3\) のセンサー・データ・リポジトリおよびセンサー・データ・ストリームの手動構成](#)」の手順に従って、手動でセンサー・データ・ストリームを構成できます。選択した後、「次へ」をクリックします。
11. 「既存の SDS に接続」画面で、新しいセンサー・データ・ストリーム・スキーマを作成するか、既存のセンサー・データ・ストリーム・スキーマに接続するかを選択し、「次へ」をクリックします。
12. 「Sensor Data Streams の構成」画面で、データベース SYS パスワードおよび TNS 接続文字列を入力し、「次へ」をクリックします。
13. 「Sensor Data Streams パスワード」画面で、センサー・データ・ストリームに使用するパスワードを指定し、「次へ」をクリックします。(この画面は、[11](#)で既存のセンサー・データ・ストリーム・スキーマに接続するように選択した場合は表示されません。)スキーマはデータベース内の `edge` ユーザーの下にインストールする必要があります。

14. 「サマリー」画面でインストール・オプションを確認し、「インストール」をクリックします。
15. 「コンフィギュレーション・アシスタント」画面でコンフィギュレーション・アシスタントが完了するのを待ち、「次へ」をクリックします。
16. インストールの完了画面で「終了」をクリックします。

Oracle Sensor Edge Server の「ようこそ」画面が表示されます。

インストール・ログ・ファイルの調査

インストール中にエラーが発生した場合、`oracle_home/edge` (`createedgeuser_SDR.log`、`deploySDR.log`、`createedgeuser_SDS.log`) 内の Oracle Sensor Edge Server ログ・ファイルおよび Oracle Universal Installer ログ・ファイルを調べることができます。

Oracle Sensor Edge Server のアンインストール

Oracle Sensor Edge Server をアンインストールするには、Oracle Universal Installer を使用する必要があります。Oracle Universal Installer の「ようこそ」画面で、「製品の削除」をクリックしてアンインストールします。

センサー・データ・リポジトリまたはセンサー・データ・ストリームをアンインストールすると、スキーマを配置するときに使用された一時ファイルが削除されますが、スキーマそのものはデータベースから削除されません。

インストールの確認

インストールが正常に完了したことを確認するには、次の操作を実行します。

- Sensor Edge Server (SES) コンソールの画面を表示し、ナビゲートします。
- データベース接続をテストします。

Sensor Edge Server コンソールの表示

SES コンソールを表示するには、次のようにします。

1. HTTP サービスがリスニングするポート番号を指定します (たとえば、8888)。ポート番号がわからない場合は、次の操作を実行します。
 1. コマンド・ウィンドウを開き、ディレクトリを `oracle_home/opmn/bin` に変更します。
 2. `opmnctl status -l` と入力します。HTTP サービスのポート番号が表示されます。

注意: OC4J リリース 10.1.2 インスタンスのポート番号を特定するには、次のようにします。

1. `ORACLE_HOME/j2ee/home/config` に移動します。
2. `http-web-site.xml` ファイルを開きます。
3. `<web-site>` 要素に `port` という名前の属性が含まれます。ポート番号はこの属性によって定義されています。

詳細は、『Oracle Application Server Containers for J2EE ユーザーズ・ガイド』を参照してください。

2. ブラウザ・ウィンドウを開き、Sensor Edge Server コンソールの URL を `http://localhost:port_number/edge/` という形式で入力します。ここで、`port_number` は http サービスのポート番号です。

- Oracle Sensor Edge Server ログイン・ページで Sensor Edge Server 管理者のユーザー名およびパスワードを入力します。デフォルトのユーザー名は `oc4jadmin` です。パスワードには、(OC4J をインストールしたときに指定した) OC4J 管理者のパスワードを入力します。

データベース接続のテスト

データベース接続をテストして、インストールが正常に完了したことを確認できます。SQL*Plus を使用してセンサー・データ・リポジトリに接続するには、次の操作を実行します。

- (`/network/admin/` 内の) `tnsnames.ora` を構成して、データベースの TNS 接続文字列を設定します。たとえば、次のように構成します。

```
BAR.US.ORACLE.COM =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = foo.us.oracle.com) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = BAR)
    )
  )
```

- スキーマに接続します。必ず、`sda_usr` のユーザー名 `edge` を使用してください。たとえば、ユーザー名が `edge` でパスワードが `sda_pwd` の場合は、次のように入力します。

```
sqlplus edge/sdr_pwd@BAR
```

- イベントが発生しているかどうかを調べるには、次のように入力して、イベントの発生とともにビュー `edg_event_vw` の行数が増加しているか確認します。

```
select count(*) from edg_event_vw
```

イベントがリポジトリに送信されると、イベントの数は増加します。何も構成されていないクリーン・インストールの場合でも、エッジ・サーバー起動イベントなどのイベントがいくつか送信されているはずですが。

この他に、(デバイス関連のエラーおよびステータス・メッセージを確認するための) `edg_diag_tab` および (イベントの処理中に発生した内部エラーを確認するための) `edg_log` の 2 つの表を診断目的で使用できます。

SQL*Plus を使用してスキーマに接続するには、次の操作を実行します。

- `tnsnames.ora` を構成して、データベースの TNS 接続文字列を設定します。たとえば、次のように構成します。

```
BAR.US.ORACLE.COM =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = foo.us.oracle.com) (PORT = 1521)))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = BAR)
    )
  )
```

- スキーマに接続します。必ず、`sds_usr` のユーザー名 `edge` を使用してください。たとえば、ユーザー名が `edge` でパスワードが `sds_pwd` の場合は、次のように入力します。

```
sqlplus edge/sds_pwd@BAR
```

`sds_pwd` は、インストール中に入力したセンサー・データ・ストリーム用のパスワードです。

- イベントが発生しているかどうかを調べるには、次のように入力して、イベントの発生とともに `sda_observations` の行数が増加しているか確認します。

```
select count(*) from sda_observations
```

イベントがリポジトリに送信されると、イベントの数は増加します。

Enterprise Manager を使用したデータベース接続のテスト

OC4J リリース 10.1.3 の場合、センサー・データ・リポジトリへの JDBC 接続をテストするには、次のようにして OC4J Enterprise Manager を使用して JNDI リソースを調べます。

1. OC4J 管理者およびパスワードを入力して、OracleAS Enterprise Manager (http://localhost:oc4j_web_port_number/em/) にログインします。
2. 「メンバー」タブをクリックし、(デフォルト・ホームごとに) OC4J インスタンスをクリックします。
3. 「管理」を選択します。
4. 必要な場合は、「サービス」を展開します。
5. 「JDBC リソース」の「タスクに移動」アイコンをクリックします。「JDBC リソース」ページが表示されます。
6. 「edge/SensorDataRepositoryDS」の「接続テスト」アイコンをクリックします。
7. 「続行」をクリックします。
8. 「SQL 文」フィールドに `select count (*) from edg_event_vw` と入力して、イベントが発生していること、およびこれらのイベントの発生とともにビュー `edg_event_vw` の行数が増加していることを確認します。

イベントがセンサー・データ・リポジトリに送信されると、イベントの数は増加します。何も構成されていないクリーン・インストールの場合でも、Oracle Sensor Edge Server 起動イベントなどのイベントがいくつか送信されているはずです。

イベントが発生していること、およびイベントの発生とともに `sda_observations` の行数が増加していることを確認するには、「SQL 文」フィールドに `select count (*) from sda_observations` と入力します。

9. 「テスト」をクリックします。「確認」の下に接続ステータス（「グループに属するすべての OC4J インスタンスで、"edge/SensorDataRepositoryDS" への接続が正常に確立されました。」）が表示されます。接続が確立されなかった場合は、エラーが表示されます。

注意：OC4J リリース 10.1.3 の場合は、OC4J Enterprise Manager から直接 OC4J インスタンスをテストできます。また、SES コンソールを使用して、Oracle Sensor Edge Server インスタンスのログ・レベルを「すべて」に設定しておくと、起動時にログを調べて正常に接続されたことを確認できます。「ログの表示」ページに、センサー・データ・リポジトリからのエラーが表示され、問題が示されます。ログ・レベルの設定方法の詳細は、「[Oracle Sensor Edge Server インスタンスの一般情報の設定](#)」および「[ログ情報の表示](#)」を参照してください。

OC4J リリース 10.1.2 の場合は、Enterprise Manager を使用して接続をテストできません。ただし、起動時にセンサー・データ・リポジトリからのエラーがログに報告されなければ、接続は正常に完了しています。Oracle Sensor Edge Server のログ・レベルを「すべて」に設定した場合は、SES コンソールの「ログの表示」ページに SQL エラーが表示されます。

Oracle Sensor Edge Mobile のインストール

Oracle Sensor Edge Mobile をインストールするには、次のようにします。

1. Companion CD を挿入し、Oracle Universal Installer を起動します。
2. 「ようこそ」画面が表示されたら、「次へ」をクリックします。
3. 「ファイルの場所の指定」画面の「インストール先」に、インストールに使用する名前および Oracle Sensor Edge Mobile のインストール先パスを入力します。Oracle Sensor Edge Mobile では OC4J と同じ `ORACLE_HOME` を使用する必要があります。「次へ」をクリックします。
4. 「インストール・タイプの選択」画面で「Sensor Edge Mobile」を選択し、「次へ」をクリックします。
5. 「サマリー」画面でインストール・オプションを確認し、「インストール」をクリックします。

インストーラにより次の .CAB ファイルが `oracle_home/edge` に配置されます。

- OracleEdgeMobile.PPC420_StrongARM-XScale.CAB: ARM または XScale デバイス用のインストーラ。Pocket PC デバイスでインストーラを実行するには、「[Pocket PC デバイスでの Oracle Sensor Edge Mobile のインストールと起動](#)」の手順に従ってください。
- OracleEdgeMobile.PPC420_i686.CAB: Windows 2000 または XP コンピュータとともに使用する Pocket PC エミュレータ。これにより、使用可能な Pocket PC デバイスがない場合にコード・テストを実行できます。エミュレータをインストールするには、「[Oracle Sensor Edge Mobile エミュレータのインストール](#)」で説明された手順に従ってください。

Pocket PC デバイスでの Oracle Sensor Edge Mobile のインストールと起動

Pocket PC に Oracle Sensor Edge Mobile をインストールするには、次のようにします。

1. ActiveSync を使用して、(`oracle_home/edge` 内の)
OracleEdgeMobile.PPC420_StrongARM-XScale.CAB を Pocket PC デバイスにロードし、このファイルをクリックしてインストールします。
2. エミュレータ画面から Sensor Edge Mobile バックグラウンド・サービスを起動するには、「Start」→「Programs」を選択し、「Start EdgeMobile Service」ショートカットをクリックします。バックグラウンド・サービスが実行されます。

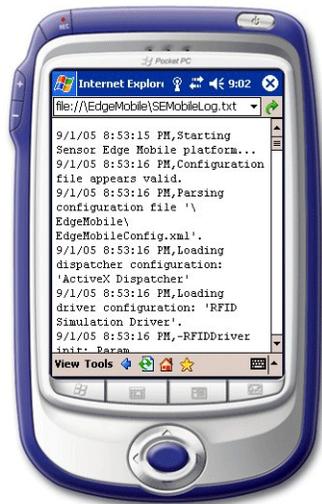
3. バックグラウンド・サービスが実行されていることを確認するには、「File Explorer」ウィンドウを開き、¥EdgeMobile¥html_samples ディレクトリに移動し、admin.html をクリックします。「Sensor Edge Mobile Administration」ページ (図 1-1) が表示されます。

図 1-1 「Sensor Edge Mobile Administration」ページ



4. 「System status」が「Running」になっていることを確認します。「System status」が「Running」でない場合は、「View log」をクリックしてログ・ファイル内のエラーを調べます。ログ・ファイルを図 1-2 に示します。

図 1-2 Sensor Edge Mobile のログ・ファイル



デフォルトのデバイス構成の変更

デフォルトでは、Symbol 900G ドライバが Pocket PC デバイスにロードされます。この他に、次のドライバが提供されています。

- EdgeMobileConfig_INTERMEC.xml: Intermec ドライバの構成ファイル
- EdgeMobileConfig_SYMBOL.xml: Symbol ドライバの構成ファイル
- EdgeMobileConfig_SIMULATOR.xml: Simulator ドライバの構成ファイル
- EdgeMobileConfig_KEYBOARD.xml: キーボード・ディスプレイパッチャの構成ファイル

別のドライバをロードするには、適切なドライバ構成ファイルの名前を EdgeMobileConfig.xml に変更して、Sensor Edge Mobile サービスを再起動します。

RFID タグの読取り

RFID タグを読み取るには、¥EdgeMobile¥html_samples ディレクトリに移動し、ファイル rfid_ops.html を開きます。シミュレーション・ドライバを使用している場合は、実際のデバイス・ドライバを使用している場合と同じようにタグが生成されます。シミュレートされた値または実際のタグが「RFID data」フィールドに表示されます。

バーコード・データの読取り

バーコード・データを読み取るには、¥EdgeMobile¥html_samples ディレクトリに移動し、ファイル barcode_ops.html を開きます。シミュレートされた値または実際のバーコードが「Barcode data」フィールドに表示されます。

Sensor Edge Mobile サービスの停止

Sensor Edge Mobile サービスを停止するには、次のようにします。

1. ¥EdgeMobile¥html_samples ディレクトリに移動し、ファイル admin.html を開きます。
2. 「Shutdown System」をクリックします。
3. 「Confirmation」ダイアログ・ボックスで「OK」をクリックします。
4. 「System status」が「Stopped」になっていることを確認します。

Oracle Sensor Edge Mobile エミュレータのインストール

Oracle Sensor Edge Mobile で使用するアプリケーションを開発する場合、Windows 2000 または Windows XP コンピュータ上で Sensor Edge Mobile エミュレータを実行できます。エミュレータを使用すると、使用可能な Pocket PC デバイスがない場合にコードをテストできます。

Oracle Sensor Edge Mobile エミュレータをインストールするには、次のようにします。

1. Oracle Sensor Edge Mobile をまだインストールしていない場合は、「[Oracle Sensor Edge Mobile のインストール](#)」の説明に従ってインストールします。
2. <http://www.microsoft.com/downloads/details.aspx?familyid=9996b314-0364-4623-9ede-0b5fbb133652&displaylang=en> から Windows Mobile 2003 ベースの Pocket PC 用の SDK をダウンロードします。
3. SDK を Windows 2000 または Windows XP コンピュータにインストールします。Pocket PC エミュレータはこの SDK の一部です。
4. Windows の「スタート」メニューから Pocket PC エミュレータを起動するには、「Microsoft Pocket PC 2003 SDK」→「Pocket PC Emulator」を選択します。
5. エミュレータが起動されたら、「Emulator」→「Folder Sharing」を選択します。
6. 「Browse」をクリックしてコンピュータ上の既存のフォルダを参照し、「OK」をクリックします。
7. 「Folder Sharing」ダイアログ・ボックスで「OK」をクリックします。

8. エミュレータ・ウィンドウで、「Start」 → 「Programs」 を選択します。
9. 「File Explorer」 アイコンをクリックします。
10. 「File Explorer」 で 「My Device」 を選択します。
11. 手順 6 で共有フォルダにしたフォルダを選択します。
12. OracleEdgeMobile.PPC420_i686.CAB ファイルを共有フォルダにロードし、エミュレータでこれをクリックしてインストールします。このファイルは `oracle_home\edge` 内にあります。

リリース 3 (10.1.3) のセンサー・データ・リポジトリおよびセンサー・データ・ストリームの手動構成

リリース 3 (10.1.3) のセンサー・データ・リポジトリまたはセンサー・データ・ストリームを使用するが、Sensor Edge Server をインストールしたときにこれらを構成していない場合は、この項で説明する手順を使用して手動で配置できます。この項の内容は次のとおりです。

- 「センサー・データ・リポジトリの手動配置」
- 「センサー・データ・リポジトリへの Oracle Sensor Edge Server の接続」
- 「センサー・データ・ストリームの手動配置」
- 「センサー・データ・ストリームへの Oracle Sensor Edge Server の接続」

センサー・データ・リポジトリの手動配置

センサー・データ・リポジトリの配置に必要な内容はすべてセンサー・データ・ストリームに含まれているため、センサー・データ・ストリームがすでに配置されている場合は、この項をスキップできます。

センサー・データ・リポジトリを手動で配置するには、次の手順を使用します。

1. センサー・データ・リポジトリ・スキーマを格納するデータベース内に `edge` ユーザー (スキーマ) を作成します。作成するには、SQL*Plus を使用して `system` としてログインし、`oracle_home/edge/stage/sql/10.1.3` 内の `create_edg_sda_user.sql` スクリプトを実行します。
2. 入力値を要求されたら、`edge` スキーマに使用するパスワードを入力します。
3. 選択したパスワードを使用して `edge` ユーザーとして SQL*Plus にログインし、`oracle_home/edge/stage/sql/10.1.3` 内の `edg_sda_with_edgeuser.sql` スクリプトを実行します。
4. 「センサー・データ・リポジトリへの Oracle Sensor Edge Server の接続」で説明されている手順に従います。

センサー・データ・リポジトリへの Oracle Sensor Edge Server の接続

Sensor Edge Server をセンサー・データ・リポジトリに手動で接続するには、次の手順を使用します。

1. OC4J が実行されていることを確認します。OC4J を起動するには、コマンド・ウィンドウを開き、ディレクトリを `oracle_home/opmn/bin` に変更します。次に `opmnctl startall` と入力します。

注意: OC4J リリース 10.1.2 の場合、OC4J を起動するには、`ORACLE_HOME/j2ee/home` に移動してから `Java_HOME/bin/java -jar oc4j.jar` を実行します。JDK 1.4.2 もマシンにインストールされている必要があります。

2. ブラウザ・ウィンドウを開き、Oracle Enterprise Manager の URL を `http://localhost:oc4j_web_port_number/em/` という形式で入力します。OC4J Web ポートに Apache リスナーが存在している場合は、Apache がリスニングしているポートを使用する必要がありますので注意してください。
3. Oracle Universal Installer で入力した OC4J パスワードを使用して、Oracle Enterprise Manager にログインします。
4. OC4J のホームページに移動し、「**管理**」をクリックします。
5. 「JMX」までスクロール・ダウンし、「**システム MBean ブラウザ**」の隣のアイコンをクリックし、`oc4j` → `J2EE Server` → `standalone` → `J2EE Application` → `default` に移動します。
6. 「**操作**」をクリックし、「**createNativeDataSource**」をクリックします。この画面を使用して、センサー・データ・リポジトリ・スキーマがインストールされているデータベースに接続するためのデータソースを作成します。
7. 次の値を入力します。
 - `dataSourceName`: `edge/SensorDataRepositoryDS`。
 - `ユーザー`: `edge`。
 - `パスワード`: `create_edg_user.sql` スクリプトを実行したときに選択したパスワードを入力します。
 - `jndiLocation`: `edge/SensorDataRepositoryDS`。
 - `loginTimeout`: `20`。
 - `dataSourceClass`: `oracle.jdbc.pool.OracleDataSource`。
 - `URL`: データベースへの jdbc URL を入力します。URL は、`jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=yourhost.us.oracle.com)(PORT=1521))) (CONNECT_DATA=(SERVICE_NAME=your_service_name)))` のようになります。
8. Sensor Edge Server 管理コンソールにログインします。
9. 最初のページの「**アーカイブを使用**」で「はい」を選択します。
10. OC4J を再起動します。

センサー・データ・ストリームの手動配置

センサー・データ・ストリームを手動で配置するには、Oracle Database が ARCHIVELOG モードで実行されており、自動アーカイブが有効になっている必要があります。ARCHIVELOG モードに変更する手順、および自動アーカイブを有効にする手順については、該当するバージョンの『Oracle Database 管理者ガイド』を参照してください。これらの手順は使用している Oracle Database のバージョンによって異なるため、必ずご使用のバージョンに対応する『Oracle Database 管理者ガイド』を参照してください。

すでにセンサー・データ・リポジトリを作成した場合、センサー・データ・ストリームを手動で配置するには、次の手順を使用します。

1. sqlplus /nolog を実行することにより、SQL*Plus を使用して sysdba としてデータベースに接続します。
2. sysdba として sqlplus connect sys/your_pwd@your_db を実行します。
3. oracle_home/edge/stage/sql/10.1.3 内のスクリプト grant_edg_user.sql を実行します。
4. 「センサー・データ・ストリームへの Oracle Sensor Edge Server の接続」で説明されている手順に従います。

まだセンサー・データ・リポジトリを作成していない場合、センサー・データ・ストリームを手動で配置するには、次の手順を使用します。

1. oracle_home/edge/stage/sql/10.1.3 内のスクリプト create_edg_user.sql を実行します。要求されたら、edge スキーマに使用するパスワードを入力します。
2. sqlplus connect edge/your_edge_user_pwd@your_db を実行します。
3. oracle_home/edge/stage/sql/10.1.3 内のスクリプト edg_create_streams.sql を実行します。

edg_create_streams.sql の実行中に、次のエラーが発生することがあります。

```
drop role edg_user
*
ERROR at line 1:
ORA-01919: role 'EDG_USER' does not exist
```

このエラーは、このデータベース・インスタンスに初めて edge スキーマがインストールされたことを示すものであるため、無視してかまいません。

4. 「センサー・データ・ストリームへの Oracle Sensor Edge Server の接続」で説明されている手順に従います。

注意: Oracle9i データベース・インスタンスにセンサー・データ・ストリームをインストールすると、次のエラーが発生します。

```
BEGIN DBMS_STREAMS_AUTH.GRANT_ADMIN_PRIVILEGE(grantee => 'edge'); END;

*
ERROR at line 1:
ORA-06550: line 1, column 7:
PLS-00201: identifier 'DBMS_STREAMS_AUTH.GRANT_ADMIN_PRIVILEGE' must be
declared
ORA-06550: line 1, column 7:
PL/SQL: Statement ignored
```

これは Oracle9i に存在しない Oracle 10g プロシージャであるため、このエラーは無視してかまいません。

センサー・データ・ストリームへの Oracle Sensor Edge Server の接続

Sensor Edge Server をセンサー・データ・ストリームに手動で接続するには、次の手順を使用します。

1. OC4J が実行されていることを確認します。OC4J を起動するには、コマンド・ウィンドウを開き、ディレクトリを `oracle_home/opmn/bin` に変更します。次に `opmnctl startall` と入力します。

注意： OC4J リリース 10.1.2 の場合、OC4J を起動するには、`ORACLE_HOME/j2ee/home` に移動してから `Java_HOME/bin/java -jar oc4j.jar` を実行します。JDK 1.4.2 もマシンにインストールされている必要があります。

2. ブラウザ・ウィンドウを開き、Oracle Enterprise Manager の URL を `http://localhost:oc4j_web_port_number/em/` という形式で入力します。OC4J Web ポートに Apache リスナーが存在している場合は、Apache がリスニングしているポートを使用する必要がありますので注意してください。
3. Oracle Universal Installer で入力した OC4J パスワードを使用して、Oracle Enterprise Manager にログインします。
4. OC4J のホームページに移動し、「**管理**」をクリックします。
5. 「JMX」までスクロール・ダウンし、「**システム MBean ブラウザ**」の隣のアイコンをクリックし、`oc4j` → `J2EE Server` → `standalone` → `J2EE Application` → `default` に移動します。
6. 「**操作**」をクリックし、「**createNativeDataSource**」をクリックします。この画面を使用して、センサー・データ・ストリーム・スキーマがインストールされているデータベースに接続するためのデータソースを作成します。
7. 次の値を入力します。
 - `dataSourceName`: `edge/StreamsDS`。
 - ユーザー : `edge`。
 - パスワード : `create_edg_user.sql` スクリプトを実行したときに選択したパスワードを入力します。
 - `jndiLocation`: `edge/StreamsDS`。
 - `loginTimeout`: 20。
 - `dataSourceClass`: `oracle.jdbc.pool.OracleDataSource`。
 - URL: データベースへの jdbc URL を入力します。URL は、`jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=yourhost.us.oracle.com)(PORT=1521))) (CONNECT_DATA=(SERVICE_NAME=your_service_name)))` のようになります。
8. ストリーム・ディスパッチャを使用するように Sensor Edge Server を設定するには、Sensor Edge Server の管理コンソールにログインします。
9. 「**ディスパッチャの変更**」をクリックし、**Streams ディスパッチャ V2** を選択します。
10. OC4J を再起動します。ストリーム・ディスパッチャにより、Sensor Edge Server スキーマへの接続が試行されます。詳細は、「[Oracle Sensor Edge Server インスタンスのディスパッチャの設定](#)」および「[Oracle Sensor Edge Server インスタンスの起動と停止](#)」を参照してください。

リリース 2 (10.1.2) の既存のセンサー・データ・リポジトリへの接続

センサー・データ・リポジトリを使用するが、Sensor Edge Server をインストールしたときに構成していない場合、次の手順を使用して手動でセンサー・データ・リポジトリに接続できます。

1. テキスト・エディタを使用して、`data-sources.xml` ファイルを開きます。このファイルは `oracle_home/j2ee/home/config/` 内にあります。
2. 次の xml エントリを追加します。

```
<data-source
  class="com.evermind.sql.DriverManagerDataSource"
  name="edge/SensorDataRepositoryDS"
  location="edge/SensorDataRepositoryDS"
  xa-location="edge/SensorDataRepositoryDSXA"
  ejb-location="edge/SensorDataRepositoryDS"
  connection-driver="oracle.jdbc.driver.OracleDriver"
  username="edge"
  password="<password for sdr>"
  url="<jdbc url for database holding SDR>"
  inactivity-timeout="100"
/>
```

3. `data-sources.xml` ファイルを保存して閉じます。

リリース 2 (10.1.2) の既存のセンサー・データ・ストリームへの接続

センサー・データ・ストリームを使用するには、Oracle Database が ARCHIVELOG モードで実行されており、自動アーカイブが有効になっている必要があります。ARCHIVELOG モードに変更する手順、および自動アーカイブを有効にする手順については、該当するバージョンの『Oracle Database 管理者ガイド』を参照してください。

センサー・データ・ストリームを使用するが、Sensor Edge Server をインストールしたときに構成していない場合、次の手順を使用して手動でセンサー・データ・ストリームに接続できます。

1. テキスト・エディタを使用して、`data-sources.xml` ファイルを開きます。このファイルは `oracle_home/j2ee/home/config/` 内にあります。
2. 次の xml エントリを追加します。

```
<data-source
  class="com.evermind.sql.DriverManagerDataSource"
  name="edge/StreamsDS"
  location="edge/StreamsDS"
  xa-location="edge/StreamsDSXA"
  ejb-location="edge/StreamsDS"
  connection-driver="oracle.jdbc.driver.OracleDriver"
  username="edge"
  password="<password for sds>"
  url="<jdbc url for database holding SDS>"
  inactivity-timeout="100"
/>
```

3. `data-sources.xml` ファイルを保存して閉じます。

Oracle Sensor Edge Server の概要

Oracle Sensor Edge Server は、センサーおよびその他のタイプのコマンドやレスポンス表示装置をアプリケーションと統合する中間層コンポーネントです。Oracle Sensor Edge Server を使用すると、企業はセンサー・デバイスから受信した情報を IT インフラストラクチャおよびビジネス・アプリケーションに統合することができます。

Oracle Sensor Edge Server は、センサー・デバイスからイベント・データを受信した後、共通のデータ形式に変換し、フィルタリングして不要な情報を除外することにより、このデータを標準化します。イベント・データは、標準化されたイベント・メッセージとしてデータベース・アプリケーションに送信されます。

この章では、Oracle Sensor Edge Server の概要を示します。この章の内容は次のとおりです。

- 「リリース 3 (10.1.3) の新機能」
- 「Oracle Sensor Edge Server の概略」
- 「Oracle Sensor Edge Server のアーキテクチャ」
- 「Oracle Sensor Edge Mobile のアーキテクチャ」
- 「配置に関する考慮事項」

リリース 3 (10.1.3) の新機能

この項では、Oracle Sensor Edge Server 10g リリース 3 (10.1.3) の新機能および拡張機能を要約して説明します。

リリース 3 (10.1.3) で追加された新機能および拡張機能は次のとおりです。

- Sensor Edge Server (SES) コンソール
- Oracle Sensor Edge Mobile
- Oracle Application Server Control を使用した拡張管理
- 向上したパフォーマンス
- 拡張されたセンサー・データ・リポジトリ
- センサー・データ・ストリーム
- 電子製品コード (EPC) コンプライアンス統合
- トランSPORT・レイヤー
- 拡張されたセキュリティ
- デバイス・コントローラの廃止

Oracle Sensor Edge Server コンソール

Oracle Sensor Edge Server の管理、監視、および構成は、新しい Sensor Edge Server コンソール (図 2-1) を使用して実行します。コンソールに備わっているツールを使用すると、配置およびメンテナンスのコストを低減できます。コンソールを使用すると、次のことを実行できます。

- 一般的なサーバー設定の構成
- サーバー・イベント統計の監視
- ディスパッチャ、フィルタ、ドライバなどの拡張機能の表示および管理
- グループ (デバイスの論理的な集まり) の管理
- 新しいデバイスの構成
- サーバー・インスタンス・パラメータの変更

図 2-1 Oracle Sensor Edge Server コンソール

The screenshot displays the Oracle Sensor Edge Server console interface. At the top, there is a navigation bar with tabs for Configuration, Monitor Events, Rules, and View Log. A warning message states: "The EdgeServer instance will need to be restarted for recent changes to take effect." Below this, the "Edge Server Configuration: MyEdgeServer123" page is shown. It includes a tree view on the left with folders for Available Extensions, Groups, Unassigned, and JimGroup. The main content area is divided into two sections: "General Settings" and "Usage Statistics".

General Settings

Server Name:	MyEdgeServer123
Site Name:	MySite
Internal Queue:	persist
Log Level:	notify
Shutdown Timeout:	10000

Usage Statistics

Total Events:	22 (82/sec)
Events Received:	92 (1/sec)
Events Generated:	97/sec
Events Sent:	35/sec
Backlog Events:	74/sec
Queued Events:	63

Current Dispatcher: Streams Dispatcher

Parameter	Type	Value
Status	STRING	NA
ExtensionName	STRING	Streams Dispatcher
uri	STRING	
username	STRING	

次のような一般的な作業を実行できます。

- サーバー・インスタンスの名前変更
- サーバーのサイト名の変更
- 内部キュー・タイプ、ログ・レベル、停止タイムアウトなどのインスタンス・パラメータの変更
- ディスパッチャの構成または変更
- グループ管理（新しいグループの作成、グループの名前変更または削除、グループへのフィルタの追加、新しいグループ・フィルタの構成、グループへのデバイスの追加など）の実行
- 新しいデバイスの構成
- デバイスのフィルタの追加または構成

Oracle Sensor Edge Mobile

Oracle Sensor Edge Mobile は、倉庫や工場内のモバイル・ワーカー向けに設計されたもので、Pocket PC 2003 以上のプラットフォームで実行される携帯 RFID リーダー上で動作します。Sensor Edge Mobile には Intermec デバイスおよび Symbol デバイスのサポートが付属しており、ActiveX コンポーネントを介して制御できます。Sensor Edge Mobile は様々なタイプのセンサーおよびデバイスとのインタフェースとして機能し、測定イベントをアプリケーションに送信します。

Oracle Application Server Control を使用した拡張管理

このリリースでは、Java Management Extensions (JMX) インタフェースが Application Server の管理コンソールと統合されたことで、アクセス可能性が拡大し、管理が容易になりました。Application Server に統合された管理作業には、起動、停止、プロセス管理およびリカバリなどがあります。これらの拡張機能により、配置およびメンテナンスのコストを低減できます。

向上したパフォーマンス

多くのセンサー対応アプリケーションで要求されるリアルタイム・パフォーマンスを実現するには、スケーラビリティと待機時間が非常に重要な要素になります。このリリースの Oracle Sensor Edge Server では、データベース・レイヤーとアプリケーション・サーバー・レイヤーを最適化することでスケーラビリティを向上させています。

拡張されたセンサー・データ・リポジトリ

このリリースでは、センサー・データ・リポジトリ (SDR) は、EPC およびセンサー・データを格納するように設計されています。データベース・スキーマは、Sensor Edge Server からのステータス情報と診断情報を格納し、センサー・データの単一リポジトリとして機能します。

SDR は、Oracle Sensor Edge Server から取得されたセンサー・イベント (RFID、温度、場所など) がアーカイブされるデータベース・スキーマです。イベントは SDR に格納されるまでに標準化されるため、レポート作成などのビジネス・インテリジェンスが容易になります。Oracle Sensor Edge Server でイベント・データの格納やイベント・アーカイブの間合せに使用できるように、SDR を有効化し、構成することができます。通常は、複数の Sensor Edge Server が 1 つの SDR を共有できます。

SDR には、センサー・イベント・データの格納と取得を簡易化するためのデータベース表、ビューおよび PL/SQL パッケージのセットが備わっています。

センサー・データ・ストリーム

Oracle Sensor Edge Server はディスパッチャと呼ばれるプラグインに基づいて、アプリケーションおよびイベント配信システムとの通信を行います。Streams ディスパッチャは、Oracle Streams のテクノロジーを使用してイベントを送受信するプラグインです。

EPC コンプライアンス統合

このリリースには、以前は別途提供されていた EPC コンプライアンス機能が含まれています。この機能によって、顧客は EPC の最小要件に簡単に準拠できます。また、エンタープライズ全体にわたって適用できる非常にスケーラブルなインフラストラクチャが提供されます。

トランスポート・レイヤー

トランスポート・レイヤーにより、Oracle Sensor Edge Server アプリケーションの開発およびメンテナンスが簡単になります。トランスポート・レイヤーにより、デバイスの実装およびアップグレードを簡易化する拡張可能な開発プラットフォームが提供されます。また、トランスポート・レイヤーによりプラットフォーム依存性が解決されるため、複数のオペレーティング・システムでデバイス・ドライバを使用することができます。

拡張されたセキュリティ

Oracle Sensor Edge Server では、Java Naming and Directory Interface (JNDI) を使用することでセキュリティを拡張しています。JNDI により、アプリケーションをディレクトリ固有のセキュリティ・システムと組み合わせて使用することができます。

デバイス・コントローラの廃止

デバイス・コントローラは不要になったため、このリリースの Oracle Sensor Edge Server では削除されました。これまでデバイス・コントローラを使用していた次のデバイスは、デバイス・コントローラなしで動作できるようになります。

- Patlite
- Intermec

Oracle Sensor Edge Server の概略

Oracle Sensor Edge Server は、情報インフラストラクチャと統合されたセンサーのパフォーマンスの管理および監視を行います。Sensor Edge Server は、センサー情報の収集とフィルタリングを行い、ローカルなセンサー・イベント処理を実行します。Oracle Sensor Edge Server は、中央のアプリケーションおよびデータベースにセキュアかつ信頼性の高い方法でイベント・データをディスパッチして戻します。

たとえば、RFID リーダーは、電子シリアル番号 (ESN) またはメモリーが埋め込まれた小型のトランスポンダを使用して、1 つ以上の周波数で ID を発信します。トランスポンダは、他の情報を自身のメモリーに格納でき、見通し内で接続されていない場合でも、離れた場所からの読取りまたは書込みが可能です。Oracle Sensor Edge Server はすべてのリーダーとのインタフェースとして機能し、標準化されたデータをアプリケーション・サーバーに戻します。

Oracle Sensor Edge Server には次の機能が備わっています。

- センサー・データの収集
- センサー・データのフィルタリング
- センサー・データのディスパッチ
- センサー・データ・アーカイブおよびルール
- Sensor Server とデバイス管理

- Sensor Edge Mobile
- SES コンソール

センサー・データの収集

Oracle Sensor Edge Server には、Radio Frequency Identification (RFID) リーダー、プリンタ、温度、運動、気圧、場所などのセンサー・ソースや、積層表示灯、メッセージ・ボード、サウンド・システムなどのレスポンス・デバイスと統合される拡張可能なドライバ・アーキテクチャが備わっています。また、Sensor Edge Server アーキテクチャへのプラグインとして機能するカスタム・コンポーネントも実装できます。標準のドライバ・インタフェースを Sensor Edge Server に実装したり、Sensor Edge Server 構成インタフェースを介してセンサー・ハードウェアのカスタム機能を公開することができます。

センサー・データのフィルタリング

Sensor Edge Server に接続されたセンサーからのデータ・ストリーミングは多様な形式で受信され、不要な情報や冗長な情報も含まれます。Sensor Edge Server は、センサー・データがエンタープライズ・アプリケーションに渡される前に、各センサーおよびセンサー・グループからデータをフィルタリングすることで、データ・クレンジングを実行します。このプロセスにより、センサー・データは一貫した形式に標準化されます。センサーのグループは、フィルタリング目的で1つの論理エンティティとして処理することができます。

センサー・データのディスパッチ

ディスパッチャは、アプリケーションとの双方向通信を可能にするプラグインです。Oracle Sensor Edge Server の主要な出力は、フィルタリングされたデータ・イベントです。これらのデータ・イベントは、最小化および標準化された形式で提供されます。データ・イベントは、サポートされている即時利用可能な方法の1つを使用して配信できます。

- Streams/AQ: Oracle Streams を介したディスパッチャは、最も堅牢で柔軟性の高いデータ転送方法です。ルールベースのプロセスおよびエージェントベースのテクノロジーが完全にサポートされるのは、このディスパッチャ方法のみです。
- Java Messaging Services (JMS) : JMS は、J2EE コンポーネントの通信に使用される標準のメッセージング API です。Oracle Sensor Edge Server には JMS ディスパッチャが用意されており、これによってユーザーはイベントを JMS トピックに取り次ぐことができます。
- Web サービス
- HTTP ポスト
- その他の転送方法は、カスタム・イベント・ディスパッチャを記述することでサポート可能になります。
- EPC PML (Physical Markup Language)
- アプリケーション・レベル・イベント (ALE)

センサー・データ・アーカイブおよびルール

データのフィルタリングおよびディスパッチが完了すると、Oracle Sensor Edge Server は Oracle 10g データベースを使用してセンサー・データのアーカイブを実行します。Sensor Edge Server を介して取り込まれたセンサー・データを格納するために調整したセンサー・データ・スキーマを使用できます。このアーカイブにより、エンタープライズ全体でセンサー・データを一様に格納でき、データ分析用のリポジトリが提供されます。

Sensor Server とデバイス管理

Oracle Sensor Edge Server では、センサー・インフラストラクチャのメンテナンス・コストを削減するため、1つの中央の場所からデバイスの管理と監視を行います。

Sensor Edge Mobile

Oracle Sensor Edge Mobile は、RFID リーダーやバーコード・リーダーなどの携帯情報端末上で動作する Pocket PC アプリケーションです。Sensor Edge Mobile は様々なタイプのセンサーおよびデバイスとのインタフェースとして機能し、測定イベントをアプリケーションに送信します。詳細は、「[Oracle Sensor Edge Mobile のアーキテクチャ](#)」を参照してください。

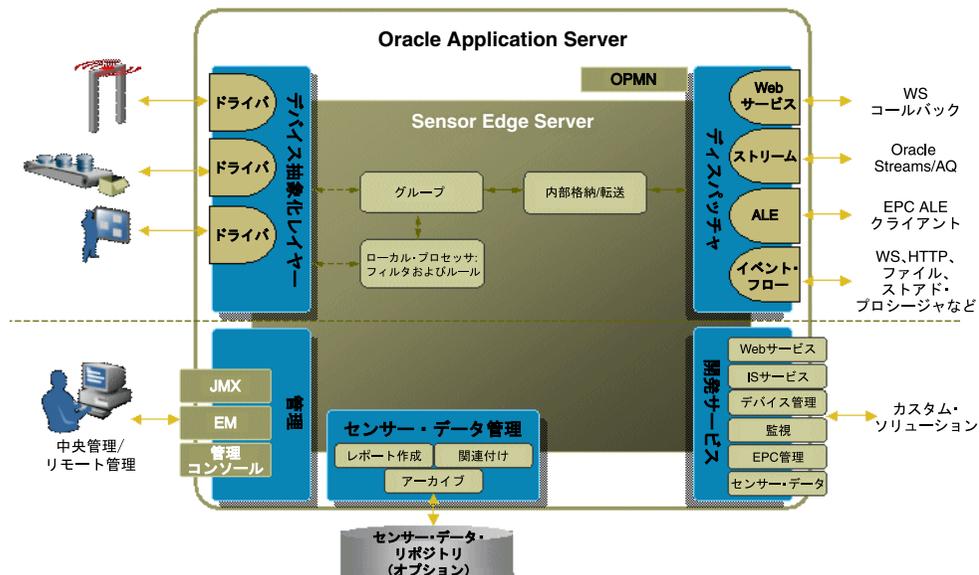
SES コンソール

Oracle Sensor Edge Server の管理、構成および監視は、SES コンソールを使用して実行します。詳細は、「[Oracle Sensor Edge Server コンソール](#)」を参照してください。

Oracle Sensor Edge Server のアーキテクチャ

この項では、Oracle Sensor Edge Server アーキテクチャの主要な部分について説明します。Oracle Sensor Edge Server アーキテクチャの概要を図 2-2 に示します。

図 2-2 Oracle Sensor Edge Server のアーキテクチャ



デバイス・ドライバ

デバイス・ドライバはセンサーと通信し、着信データを標準の形式に標準化します。デバイス・ドライバは、ニーズに合わせてカスタマイズまたは構築できます。

デバイス・グループ

デバイス・グループを使用すると、管理者はデバイスを論理的にグループ化し、より効率的にデバイスを管理できます。1つの Oracle Sensor Edge Server に対して、1つまたは多数のデバイス・グループをインスタンス化できます。各デバイス・グループには、その管理対象となるデバイス・ドライバが割り当てられます。

ローカル処理

ローカル処理（フィルタおよびルール）では、不要なイベントや下位のイベントが削除されず、フィルタおよびルールは、着信データをソートするために使用できます。各デバイスに適用することも、デバイス・グループに適用することもできます。

イベント・プロセッサ

イベント・プロセッサは、イベント配信用の中央処理エンジンです。イベント・プロセッサによって、（起動時の）ドライバ・マネージャやイベント・ディスパッチャなど、残りのコンポーネントがロードされます。内部では、ロードするイベント・ディスパッチャを確認するために構成が読み取られます。プロセスは起動時に開始されます。

ドライバ・マネージャ

ドライバ・マネージャは、デバイス・グループとデバイス・ドライバのライフ・サイクルをロードおよび管理します。1つの Oracle Sensor Edge Server につきドライバ・マネージャのインスタンスは1つのみです。ドライバ・マネージャは、デバイス・ドライバが構成情報を取得できるように、アクセス可能なコンテキストをデバイス・ドライバに提供します。次にデバイス・グループをロードし、デバイス・ドライバにデバイス・グループをバインドします。ドライバ・マネージャは内部でスレッドを保持しませんが、サーバーが動作している間は、イベント・プロセッサ・インスタンスによりドライバ・マネージャのインスタンスが保持されます。

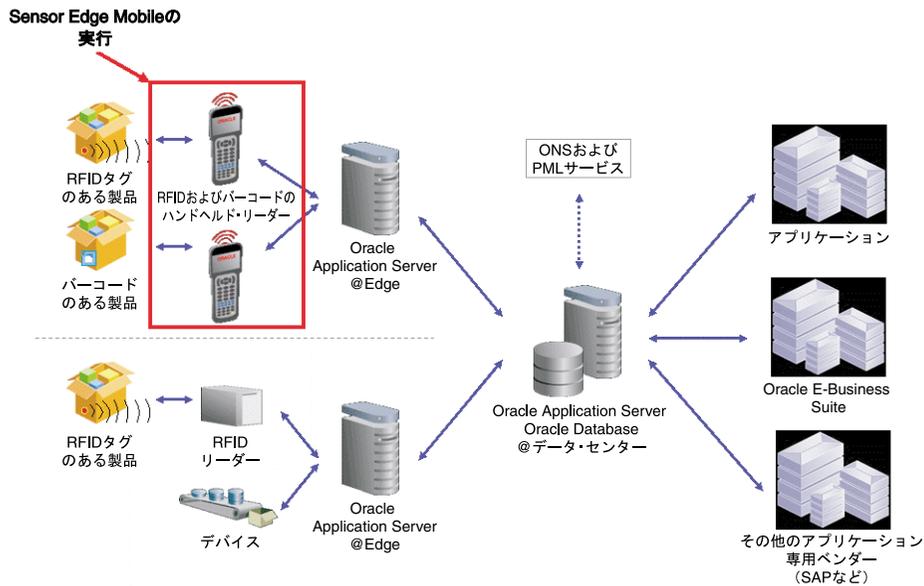
1つの Oracle Sensor Edge Server につき、イベント・プロセッサとドライバ・マネージャはそれぞれ1つずつです。各ドライバ・マネージャは多数のデバイス・グループをロードします。また、各デバイス・グループは任意の数のデバイス・ドライバを持つことができます。ただし、1つのデバイス・ドライバは1つのデバイス・グループにのみ属することができます。

Oracle Sensor Edge Mobile のアーキテクチャ

Oracle Sensor Edge Mobile は、デバイスのモバイル・センサー・データ収集を管理し、デバイス上で実行されているアプリケーションと通信したり、デバイスのブラウザを介してリモート通信を行うプラットフォームです。通信は双方向です。イベントはプラットフォームを介してデバイスからアプリケーションに渡され、命令はプラットフォームを介してアプリケーションからハードウェア・デバイス・ドライバに渡されます。

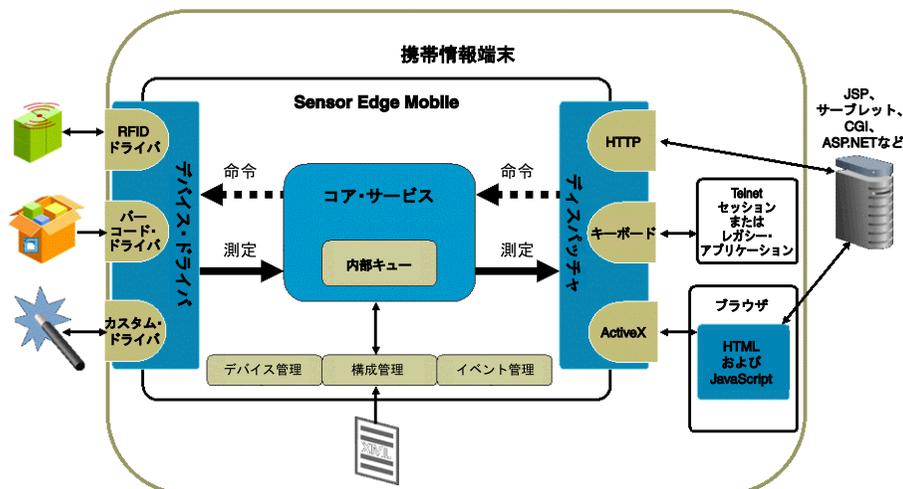
Oracle Sensor Edge Mobile のアーキテクチャの概要を図 2-3 に示します。Sensor Edge Mobile は、完全に携帯情報端末上で稼働し、他のアプリケーションや外部のサービスと通信できます。また、完全にオフラインで実行して、データを収集し、後でこのデータをアプリケーションと同期化することもできます。

図 2-3 Sensor Edge Mobile のアーキテクチャの概要



Oracle Sensor Edge Mobile プラットフォームをよく見ると、デバイス・ドライバにより収集された測定値は、いったん内部キューに格納されてから、ディスパッチャにより処理されることがわかります。Sensor Edge Mobile のコンポーネントを図 2-4 に示します。

図 2-4 Sensor Edge Mobile のコンポーネント



ディスプレイパッチャは、ローカルの ActiveX コントロールを使用して、文字をキーボード・バッファに送信することにより（またはその他ほとんどあらゆる通信方法によって）、アプリケーションと通信します。同時にアクティブにできるディスプレイパッチャは1つのみですが、そのディスプレイパッチャは複数のクライアント・デバイスおよび対応するドライバと同時に通信できます。

Sensor Edge Mobile のコア・コンポーネントは、ドライバ・マネージャ、イベント・マネージャおよび構成マネージャです。これらのコンポーネントが起動されると、メイン・サービスによって構成マネージャが呼び出され、構成ファイルが読み取られます。次に構成済のディスプレイパッチャが起動され、指定されている構成パラメータが渡されます。

ドライバ・マネージャは、ドライバをロードし、そのライフサイクルを管理します。ドライバ・マネージャは構成マネージャを呼び出して、ロードする必要のあるドライバと、インスタンス化時にドライバに対して使用可能にするパラメータを判別します。次に、それらのドライバをロードしてインスタンス化します。

Oracle Sensor Edge Mobile には、イベント・マネージャとドライバ・マネージャが1つずつのみ存在します。ドライバ・マネージャは多数のドライバをロードできます。

デバイス・ドライバのサポート

Oracle Sensor Edge Mobile では、Symbol 9000-G デバイスでの次の操作がサポートされています。

- RFID 読取り
- RFID 書込み
- RFID 中断
- バーコード読取り

Oracle Sensor Edge Mobile では、Color 700 Pocket PC を使用する Intermec IP3 での次の操作がサポートされています。

- RFID 読取り
- バーコード読取り

Oracle Sensor Edge Mobile の管理

Oracle Sensor Edge Mobile は、Oracle Sensor Edge Server の Web インタフェースを介して管理できます。次の管理作業を実行できます。

- 全体的なサービス・ステータスの監視
- 管理ログの監視
- ドライバ統計の確認
- サービスの停止

サンプル・コードおよびデモ・アプリケーション

Oracle Sensor Edge Mobile には、次の例が示されたサンプルの HTML および JavaScript コードが含まれています。

- RFID タグの読取りおよび書込み
- バーコードの読取り
- システム・ステータス・ページ
- サービス管理ページ

サンプル・コードは、API の使用方法を示し、インストールおよび構成の直後に Oracle Sensor Edge Mobile を実行する方法を示しています。

配置に関する考慮事項

Oracle のセンサー・サービス・アーキテクチャは、工業のサプライ・チェーン環境や、商用アプリケーションと政府機関アプリケーション用の特殊なマルチセンサー環境において直面する問題に対処するものです。これらの多様なニーズに応えるため、オラクル社はデュアル・エッジの配置構成を提供しています。各エッジが固有のフットプリントを持ち、集中型と連合型のいずれのサイト配置シナリオもサポートできる柔軟性が備わっています。

配置またはパイロットに選択する構成を決定する際は、次の点を考慮すると効果的です。

ネットワーク特性の確認

配置を計画する際は、まず、施設内のネットワーク帯域幅と、他の配信センターおよびデータ・センターへのイントラネット接続を特定します。通常は、次の点について考慮します。

- ウェアハウス内の接続: 通常は直接のイーサネットまたは Wireless 802.xx によって提供されます。接続停止時間を予測する必要があります。
- ウェアハウスからウェアハウスまたはデータ・センターへの接続: ほとんどの場合、この接続は ISDN 回線 1 本のみになります。

データ・センター環境の特定

通常は、データ・センターに関して次の点を考慮します。

- 環境: ウェアハウス内のハードウェアは、通常耐久性が高くなっています（バーコード・リーダー、システム・コントローラなど）。操作上の要件（たとえば、ドック・ドアを開くなど）によって、環境の制御が制限されることもあります。
- サポート・スタッフ: 現地または遠隔の IT サポート・スタッフが対応できるかどうか。

リーダーおよびセンサーの場所の確認

たとえば、RFID デバイスを配置する際、通常は、顧客側の制約（積上げと梱包の場所、搬送ラインなど）と、RF サーベイにより示された RFID ハードウェアが最小の干渉で最も効率的に機能する場所との間で得失評価を行う必要があります。

また、リーダーおよびセンサー・デバイスの接続要件（シリアル、ワイヤレス、ダイレクト・イーサネットなど）を特定する必要があります。

Edge Server の場所の選択

前述の項目を特定した後、データ・センターごとに Oracle Sensor Edge Server の設置場所を設計して文書化します。Edge Server の数および設置場所は通常、次の要素によって決定されます。

- 各設置場所におけるデバイスの数と、それらのインターフェース要件
- Edge Server の電源およびネットワーク接続の利用
- 環境の考慮事項（たとえば、外気に晒されている、工業設備など）

Oracle Sensor Edge Server およびセンサー・データ・リポジトリの考慮事項

ウェアハウスまたは配信センターにサーバー室が存在し、サポート・スタッフがいない場合は、センサー・データ・リポジトリ (SDR) インフラストラクチャをローカルに使用可能にすることを考慮します。SDR インフラストラクチャをローカルに使用可能にすると、業務上の複雑な意思決定をローカルに行ってリアルタイムで実施する必要がある場合、応答時間を短縮することができます。Oracle Sensor Edge Server にはデータ取得およびフィルタリングの機能が豊富に備わっていますが、ウェアハウスまたは配信センターにおける設置面積は小さくて済みます。SDR の詳細は、第 4 章「センサー・データ・リポジトリの使用」を参照してください。

また、ウェアハウスから中央のデータ・センターへのネットワーク接続が最小限しかない場合は、待機時間によって、スキャンされたデータに対してアラートや意思決定を実施できないことがあります。このような場合にも、SDR インフラストラクチャをローカルに使用可能にする必要があります。

通常、トランザクションを他のデータ・センターまたは中央のデータ・センターと同期化するため、データベース・レベルでレプリケーションを実装します。

ほとんどの顧客は、データベースおよびアプリケーション・サーバーを中央のデータ・センターに設置します。このシナリオでは、ウェアハウスまたは配信センターにはサポート・スタッフは少数しかいないか、あるいはまったくいません。ウェアハウスには Sensor Edge Server のみが設置され、データベースおよびアプリケーション・サーバーの機能を備えた中央データ・センターへの接続があります。

集中型構成においてはネットワーク帯域幅が重要であるため、フィルタリング済データに対して処理を実行する場合は、この帯域幅を重視する必要があります。データ・フィルタリング、例外、および基本的なトリガーはローカルに実行できます。ローカル・ウェアハウス・システムから返されたリモート・インスタンスおよび意思決定に対しては、複雑なルール・セット、製品相関、および交差型の問合せをすべて実行することができます。

制御システムの統合が必要な場合は、ローカル処理フィルタまたは SDR をローカルに設置して、ウェアハウスの自動操作に必要な機能およびパフォーマンスをローカルに提供する必要があります。

Oracle Sensor Edge Server の管理

この章では、Sensor Edge Server コンソール (SES コンソール) を使用して Oracle Sensor Edge Server インスタンスを管理および監視する方法について説明します。この章の内容は次のとおりです。

- 「Oracle Sensor Edge サービス管理の概要」
- 「Oracle Sensor Edge Server インスタンスの管理」
- 「他の Oracle Sensor Edge Server インスタンスへのアクセス」
- 「Oracle Sensor Edge Server インスタンスのパフォーマンスの監視」
- 「Oracle Sensor Edge Server インスタンスの一般情報の設定」
- 「Oracle Sensor Edge Server インスタンスのディスパッチャの設定」
- 「ディスパッチャ、ドライバおよびフィルタの表示」
- 「Oracle Sensor Edge Server で使用するデバイスおよびフィルタの設定」
- 「Oracle Sensor Edge Server インスタンスの起動と停止」
- 「各デバイスの起動と停止」
- 「フィルタの管理」
- 「イベント・データの監視」
- 「未処理のイベント・データの表示」
- 「ログ情報の表示」
- 「処理済のイベント・データの表示」
- 「Oracle Sensor Edge Server インスタンスの拡張機能の追加」

Oracle Sensor Edge サービス管理の概要

Oracle Sensor Edge Server を使用すると、センサー・デバイスまたはアプリケーションからイベント・データを受信し、受信したデータを共通のデータ形式に標準化した後、フィルタを使用して余分な情報を除外することで、エンタープライズはセンサーからの情報を IT インフラストラクチャおよびビジネス・アプリケーションに取り込むことができます。イベント・データは標準化されたイベント・メッセージとなり、ディスパッチャを使用してエッジ・クライアントに送信されます。Oracle Sensor Edge Server のディスパッチャの構成に応じて、Oracle Sensor Edge Server クライアントは Web サービス、HTTP、EPC PML、ALE Web サービスまたはデータベース・ストリームを使用してイベント・メッセージを受信します。メッセージは常にイベントによって受渡しされます。

SES コンソールでは、Oracle Sensor Edge Server インスタンスを管理および監視できます。SES コンソールのタブと、そのタブで実行できる操作を表 3-1 に示します。

表 3-1 SES コンソールで実行できる操作

タブ	操作
メイン	次の操作があります。 <ul style="list-style-type: none"> ■ Oracle Sensor Edge Server インスタンスのディスパッチャの設定 ■ Oracle Sensor Edge Server で使用するデバイスおよびフィルタの設定 ■ 各デバイスの起動と停止 ■ OracleAS Enterprise Manager を使用した Oracle Sensor Edge Server インスタンスの再起動 ■ 他の Oracle Sensor Edge Server インスタンスへのアクセス
監視イベント	未処理のイベント・データの表示
ログの表示	ログ情報の表示
イベント・レポート	次の操作があります。 <ul style="list-style-type: none"> ■ デバイス名によるイベントの検索 ■ タグ ID によるイベントの検索 ■ 拡張検索の作成

Oracle Sensor Edge Server インスタンスの管理

OC4J 管理者の名前およびパスワードを使用して SES コンソールにログインすると、コンソールではデフォルトで「構成」タブのメイン・ページが表示されます。このページには、現在の Oracle Sensor Edge Server インスタンスの使用状況、その基本構成、およびその現行ディスパッチャが全体的に表示されます。このページから、「サーバー」や「サイト名」パラメータなどの基本構成を編集したり、別のディスパッチャ方法を選択することができます。Oracle Edge Sensor Server インスタンスの基本構成の詳細は、「[Oracle Sensor Edge Server インスタンスの一般情報の設定](#)」を参照してください。

注意：最初に OC4J を起動する必要があります。「[Oracle Sensor Edge Server インスタンスの起動と停止](#)」を参照してください。

図 3-1 メイン・ページ (表示の一部)

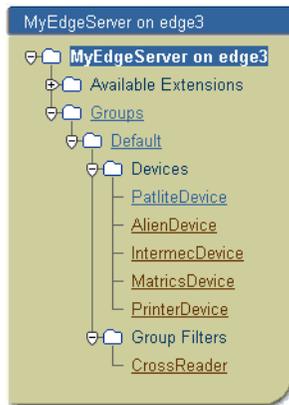
The screenshot shows the Oracle Sensor Edge Server configuration interface. On the left is a tree view under 'MyEdgeServer on edge3' with sub-items like 'Available Extensions', 'Groups', 'Default', 'Devices', and 'Group Filters'. The main area is titled 'Edge Server Configuration: MyEdgeServer on edge3'. It includes a warning message: 'The EdgeServer instance will need to be restarted for recent changes to take effect. You can restart by using the Enterprise Manager page.' Below this are 'General Settings' with fields for Server Name (MyEdgeServer on edge3), Site Name (edge-test3), Internal Queue (persist), Log Level (error), Use Archive (Yes), and Shutdown Timeout (10000). To the right is a 'Usage Statistics' box showing 0 events received, 1 generated, 0 sent, and 1 queued. At the bottom, a table shows the current dispatcher is 'Streams DispatcherV2' with version 1.0.

Parameter	Type	Value
ExtensionName <small>Name of this extension</small>	STRING	Streams DispatcherV2
Version <small>Version of this extension</small>	STRING	1.0

Oracle Sensor Edge Server インスタンスの基本構成の詳細は、「[Oracle Sensor Edge Server インスタンスの一般情報の設定](#)」を参照してください。ディスパッチャの詳細は、「[デバイス、フィルタ・インスタンスおよびディスパッチャの構成](#)」を参照してください。

SES コンソールの各ページに表示されるナビゲーション・ツリー (図 3-2) では、現在の Oracle Sensor Edge Server インスタンスおよびリポジトリ内の拡張機能 (フィルタ、ドライバおよびディスパッチャ) を構成できます。ツリーでは現在の Oracle Sensor Edge Server の構成ページが階層編成され、最上位ノードに現在の Oracle Sensor Edge Server の名前が表示されます。ノードの隣にあるプラス (+) 記号またはマイナス (-) 記号をクリックすると、項目を表示したり、非表示にすることができます。(図 3-2 の「グループ」のように) ツリーで下線の付いた項目は、変更できます。下線の付いた項目をクリックすると、選択した項目を変更および管理できるプロパティ・ページにアクセスできます。その他の項目 (下線の付いていない項目) はタイトルであり、変更できません。

図 3-2 ナビゲーション・ツリー



ツリーからアクセスしたページでは、次の操作を実行できます。

- Oracle Sensor Edge Server インスタンスのパフォーマンスの監視
- Oracle Sensor Edge Server インスタンスの一般情報の設定
- ディスパッチャ、ドライバおよびフィルタの表示
- Oracle Sensor Edge Server で使用するデバイスおよびフィルタの設定
- 各デバイスの起動と停止

他の Oracle Sensor Edge Server インスタンスへのアクセス

SES コンソールでは、「エッジ・サーバー・インスタンス・リスト」ページを使用して、同じセンサー・データ・リポジトリに接続されている他の Oracle Sensor Edge Server インスタンスにアクセスできます。このページにアクセスするには、「他のサーバー」アイコン (図 3-3) を使用します。

このページの「エッジ・サーバー・インスタンス・リスト」表には、同じセンサー・データ・リポジトリに接続されている Oracle Sensor Edge Server のエントリが一覧表示されます。これらのエントリにより、他のサーバー・インスタンスを特定したり、他のサーバー・インスタンスにアクセスすることができます。この表を使用すると、別の Oracle Sensor Server インスタンスにアクセスしたり、他の Oracle Sensor Server インスタンスの実行ステータスを表示することができます。また、他の Oracle Sensor Server インスタンスのエントリの編集および削除もできます。

別の Oracle Sensor Server インスタンスにアクセスするには、「エッジ・サーバー・インスタンス・リスト」表の「サーバー名」列のインスタンス名をクリックし、表示されるログイン・ページに OC4J の管理者の名前とパスワードを入力します。

表のメモ帳のアイコンをクリックすると、選択した Oracle Sensor Edge Server インスタンスのエントリを編集できます。ゴミ箱のアイコンをクリックすると、エントリを削除できます。ページの「新規エッジ・サーバー・インスタンス」セクション内のフィールドを使用すると、Oracle Sensor Edge Server インスタンスのエントリを作成できます。このエントリはインスタンスの名前と場所で構成されます (「Oracle Sensor Edge Server インスタンスのエントリを作成」を参照)。

注意： Oracle Sensor Edge Server のライブ・インスタンスが起動されるとそれ自身のエントリが作成されるため、Oracle Sensor Edge Server インスタンスのエントリを作成するように要求されることはありません。

図 3-3 「他のサーバー」アイコン



Oracle Sensor Edge Server インスタンスのエントリの作成

Oracle Sensor Edge Server インスタンスのエントリを作成するには、次のようにします。

1. 「他のサーバー」アイコン (図 3-3) をクリックします。「エッジ・サーバー・インスタンス・リスト」ページが表示されます。このページには、同じセンサー・データ・リポジトリに接続されている Oracle Sensor Edge Server インスタンスの表が表示されます。
2. Oracle Sensor Edge Server インスタンスの名前を入力します。
3. Oracle Sensor Edge Server インスタンスを指す URL を入力します。次の形式で URL を入力します。

```
http://<hostname/ip address>:<port>/edge
```

<port> の値は、Oracle Application Server Containers for J2EE (OC4J) がリスニングするポートです。これは、Oracle Sensor Edge Server インスタンスが実行されているマシン上に存在します。OC4J リリース 10.1.3 インスタンスの場合、ポート値を見つけるには、`oracle_home/opmn/bin` に移動してから次のコマンドを実行します。

```
opmnctl status - l
```

OC4J リリース 10.1.2 インスタンスの場合、OC4J リスナー・ポートは通常 8888 です。OC4J リリース 10.1.2 の詳細は、『Oracle Application Server Containers for J2EE ユーザーズ・ガイド』および「OC4J のインストール」を参照してください。

4. 「エントリの作成」をクリックします。新しい Oracle Sensor Edge Server インスタンスが「エッジ・サーバー・インスタンス・リスト」表に表示されます。

Oracle Sensor Edge Server インスタンスのエントリの編集

エントリを編集するには、まず、選択した Oracle Sensor Edge Server エントリのメモ帳アイコンをクリックします。編集ページが表示され、選択したエントリに設定されている情報が「サーバー名」および「URL」フィールドに表示されます。次のいずれか（または両方）の操作を実行します。

- エントリの新しい名前を入力します。
- エントリの URL を変更します。

変更をコミットする場合は「エントリの更新」をクリックし、エントリを元の状態に戻す場合は「取消」をクリックします。

Oracle Sensor Edge Server インスタンスのパフォーマンスの監視

Sensor Edge Server インスタンス・ページの「利用状況統計」セクション (図 3-1) に、Oracle Sensor Edge Server インスタンスの次のパフォーマンス・メトリックが一覧表示されます。

- 受信されたイベント : Oracle Sensor Edge Server インスタンスによって受信されたインバウンド命令イベントの数。インバウンド・イベントの詳細は、「[イベント・データの監視](#)」を参照してください。
- 生成されたイベント : Oracle Sensor Edge Server インスタンスから送信されたアウトバウンド・イベントの数。アウトバウンド・イベントの詳細は、「[イベント・データの監視](#)」を参照してください。
- 送信されたイベント : Oracle Sensor Edge Server インスタンスによりディスパッチされたイベントの合計数。
- キューに入れられたイベント : 現在キューで待機しているイベントの数。

イベント・データのキューのクリア

「[キューの消去](#)」をクリックすると、システム内のキューに格納されているイベントがすべて削除されます (また、「[キューに入れられたイベント](#)」に表示される数値が 0 に設定されます)。この機能を使用して、ディスパッチャを設定および起動する前に、または Oracle Sensor Edge Server インスタンスを再起動する前に、バックアップされている古いイベント・メッセージをページします。

警告: 「[キューの消去](#)」をクリックした後で、ページしたイベント・データをリカバリすることはできません。

Oracle Sensor Edge Server インスタンスの一般情報の設定

メイン・ページの「一般設定」セクションでは、Oracle Sensor Edge Server インスタンスの基本情報（表 3-2）を編集できます。「変更の保存」をクリックして、更新を Oracle Sensor Edge Server にコミットします。

表 3-2 一般設定

パラメータ	説明
サーバー名	Oracle Sensor Edge Server インスタンスの名前。
サイト名	Oracle Sensor Edge Server のサイト名。このパラメータは、複数の Oracle Sensor Edge Server インスタンスを論理的に区別するためのグループ化メカニズムである。
内部キュー	<p>リーダー（またはデバイス）から送信されたイベント・メッセージは、内部キューに格納される。ディスパッチャはこのキューからイベントを取り出してディスパッチする。たとえば、ストリーム・ディスパッチャはこのキューからイベント・メッセージをデータベースにディスパッチする。</p> <p>次の各オプションを使用すると、イベント・メッセージがディスパッチされる前にどのように保持されるかを制御することによりデータを保護し、イベント・データを保護できる。</p> <p>次のオプションがある。</p> <ul style="list-style-type: none"> ■ 永続：「永続」モードを選択すると、イベント・メッセージはディスクに格納されてから、ディスク上でディスパッチャにより収集される。これらのイベント・メッセージをディスクに格納すると、Oracle Sensor Edge Server インスタンスがクラッシュした場合にイベント・メッセージが失われることがない。 ■ メモリー：データベースまたは接続のいずれかが遅い場合、このオプションを選択すると、デバイスがイベント・メッセージを生成する速度がディスパッチできる速度よりも速いときにイベント・メッセージがバックアップされない。「メモリー」を選択すると、Oracle Sensor Edge Server インスタンスはこれらのメッセージをメモリー内に保持する。この場合、メッセージはディスクに書き込まれないため、生成されるオーバーヘッドは少ないが、Oracle Sensor Edge Server インスタンスがクラッシュした場合はイベント・メッセージが失われる。
ログ・レベル	<p>ログ・ファイルに書き込まれるメッセージの重大度レベルを設定する、次のエラー記録オプションのリスト。</p> <ul style="list-style-type: none"> ■ エラー ■ 警告 ■ 通知 ■ モニター ■ デバッグ <p>ログ・ファイルに書き込まれるエラー・メッセージは、選択したログ・レベルのみでなく、選択したログ・レベルよりも重大度の高いログ・レベルも反映する。選択したレベルにより、「ログの表示」タブに表示されるデータが決定される。「ログ情報の表示」も参照。</p>
アーカイブを使用	このオプションを選択すると、イベントがセンサー・データ・リポジトリに保存される。
停止タイムアウト	Oracle Sensor Edge Server が、正常に機能していないスレッドを停止するまでに待機する時間（ミリ秒）を入力する。

Oracle Sensor Edge Server インスタンスのディスパッチャの設定

メイン・ページの「ディスパッチャの変更」(図 3-4) をクリックすると、Oracle Sensor Edge Server で使用されるディスパッチャを変更できます。「Oracle Sensor Edge Server インスタンス用のディスパッチャの管理」も参照してください。

図 3-4 ディスパッチャの変更

Current Dispatcher: Streams DispatcherV2 [Status: NA]

Parameter	Type	Value
ExtensionName <small>Name of this extension</small>	STRING	Streams DispatcherV2
Version <small>Version of this extension</small>	STRING	1.0
threadnum <small>The number of worker threads to dispatch messages.</small>	STRING	<input type="text" value="2"/>
LastMessage <small>Last error or display message</small>	STRING	Start

表示される「検索と選択」ページ(図 3-5) から、リモート Web サービス、Oracle Streams などの方法を使用して (クライアント・アプリケーションには HTTP を使用して) イベント・メッセージを送信するエッジ・ディスパッチャを選択できます。

図 3-5 新しいディスパッチャの選択

Search and Select: Change current dispatcher

Once you select an appropriate dispatcher, be sure to configure it correctly.

To change dispatchers, choose a dispatcher from the list below. Only one dispatcher may be configured at any time.

Search

Search

Results

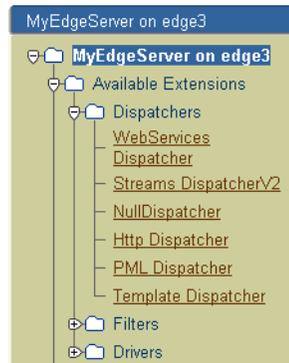
Select Dispatcher	Description
<input type="radio"/> WebServices Dispatcher	WebServices Dispatcher
<input type="radio"/> Streams DispatcherV2	Streams DispatcherV2
<input type="radio"/> NullDispatcher	Dispatcher that does nothing
<input checked="" type="radio"/> Http Dispatcher	Http Dispatcher
<input type="radio"/> PML Dispatcher	PML Dispatcher
<input type="radio"/> Template Dispatcher	Template Dispatcher

Oracle Sensor Edge Server インスタンスで同時に使用されるディスパッチャは1つのみです。ディスパッチャを現行ディスパッチャとして割り当てた後、Oracle Sensor Edge Server インスタンスを再起動する必要があります。「Oracle Sensor Edge Server インスタンスの起動と停止」を参照してください。

ディスパッチャ、ドライバおよびフィルタの表示

ツリー内の「使用可能な拡張機能」フォルダ (図 3-6) を展開すると、Oracle Sensor Edge Server インスタンスに使用可能なディスパッチャ、フィルタおよびドライバが表示されます。ディスパッチャは、Oracle Sensor Edge Server インスタンスに送信されたイベントをディスパッチ・レイヤーまたは直接アプリケーションに転送します。ドライバを使用すると、(リーダーなどの) デバイスと Oracle Sensor Edge Server インスタンスと間の通信が可能になります。また、通常、(重複イベントなどの) 不要なイベントを削除するか、1 つ以上のイベントを削除して上位レベルのイベントに変換するフィルタも使用可能になります。

図 3-6 「使用可能な拡張機能」フォルダ



ドライバなどの拡張機能をクリックすると、選択したドライバのデフォルト・プロパティが表示されます。パラメータの値セットは読み取り専用であり、これらのオブジェクトの現在の構成を表しているわけではありません。Oracle Sensor Edge Server インスタンスでは拡張機能のライブ・インスタンスは使用されないため、これらのパラメータを構成することはできません。かわりに、Oracle Sensor Edge Server インスタンスにより、「使用可能な拡張機能」フォルダに示されたフィルタおよびドライバのインスタンスを使用してイベント・データの読み取りとクレンジングが実行されます。これらのフィルタ・インスタンスおよびドライバ・インスタンス (デバイスと呼ばれる) を作成するには、デバイス・グループを作成する必要があります。デバイス・グループの作成の詳細は、「デバイス、フィルタ・インスタンスおよびディスパッチャの構成」を参照してください。Sensor Edge Server のディスパッチャの設定の詳細は、「Oracle Sensor Edge Server インスタンスのディスパッチャの設定」を参照してください。

Oracle Sensor Edge Server で使用するデバイスおよびフィルタの設定

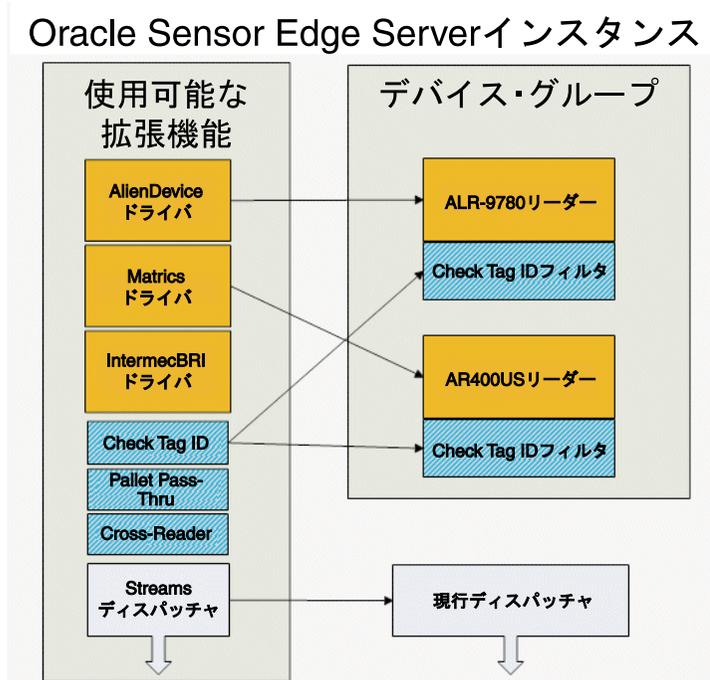
Oracle Sensor Edge Server インスタンスによるイベント・データの受信、フィルタリングおよびディスパッチを可能にするには、まずデバイス・グループを作成する必要があります。デバイス・グループは、デバイス (ドライバのインスタンス) の論理グループと、これらのデバイスに関連付けられたフィルタで構成されます。1 つの Oracle Sensor Edge Server インスタンスに対して、1 つまたは多数のデバイス・グループをインスタンス化できます。各デバイス・グループは、それに属するすべてのデバイス (およびそれに関連付けられたフィルタ) の処理を行います。

デバイス・グループにより、1 つ以上のデバイスを論理デバイス (グループ) に編成します。デバイス・グループを使用して、処理するデバイスを 1 つのイベント・ストリームとして関連付けます。たとえば、ウェアハウスのすべての出口に設置された全デバイスで構成される、Warehouse Exits という名前のデバイス・グループを作成 (また必要な場合、このグループにフィルタ・インスタンスを追加) したとします。この場合、生成されるイベントはすべて、複数のデバイスからでなく 1 つの論理デバイスから生成されたデータとして表示されます。

(前述の Warehouse Exits デバイス・グループの場合のように) デバイスを 1 つの論理的な管理単位として扱う必要がある場合、管理のためにデバイスをグループ化できます。あるいは、デバイスにより実行されるフィルタリング・タイプに基づいてデバイスをグループ化することもできます (図 3-7)。たとえば、リーダー間フィルタリングによってデバイスをグループ化する場合は、関連するデバイス・グループを作成し、そのグループにフィルタをアタッチすること

ができます。詳細は、「[デバイスまたはデバイス・グループのフィルタ・インスタンスの管理](#)」を参照してください。

図 3-7 フィルタによってグループ化されたリーダー・デバイス



Oracle Sensor Edge Server のデバイス・グループの表示

「グループ」フォルダをクリックすると、「グループ管理」ページ (図 3-8) が表示され、デバイス・グループおよびそれぞれのフィルタが一覧表示されます。また、このページにはグループごとのデバイスの実行ステータスも一覧表示されます。さらに、このページでは新しいデバイス・グループを作成することもできます (「[デバイス・グループの作成](#)」を参照)。特定のデバイス・グループの構成を表示するには、「グループ」フォルダを展開し、適切なノードを選択して、選択したデバイス・グループの「グループの構成」ページを表示します。デフォルトのデバイス・グループは、「未割当て」という名前の特殊な予約済グループです。「グループの構成」ページの詳細は、「[デバイス・グループの編集](#)」を参照してください。

図 3-8 デバイス・グループの構成

The EdgeServer instance will need to be restarted for recent changes to take effect. You can restart by using the [Enterprise Manager](#) page.

Group Management

This top level group management page lists all of the sensor devices and group-level filters configured on this Sensor Edge or filters, or to perform tasks on existing devices or filters, navigate to each group using the tree control on the left.

Groups and Devices

Group Name	Device/Filter Name	Status
Default	PatliteDevice	⚠ DOWN
Default	AlienDevice	⚠ DOWN
Default	CrossReader	NA
Default	IntermecDevice	⚠ DOWN
Default	MatricsDevice	⚠ DOWN
Default	PrinterDevice	⚠ DOWN

Create New Group

Group Name:

デバイス・グループの作成

Oracle Sensor Edge Server インスタンスをデバイスおよびフィルタに接続する場合、最初にデバイス・グループを作成します。デバイス・グループを作成した後は、デバイス・グループにデバイス（使用可能なドライバのインスタンス）を移入してから、フィルタ・インスタンスを個々のデバイス（またはデバイス・グループ全体）にアタッチします。新しいデバイス・グループを作成するには、次のようにします。

1. 「グループ管理」 ページで、「グループ名」 フィールドにデバイス・グループの名前を入力し、「新規グループの作成」 をクリックします。新しいデバイス・グループに対する「グループの構成」 ページが表示されます。

図 3-9 「グループの構成」 ページ

The EdgeServer instance will need to be restarted for recent changes to take effect. You can restart by using the [Enterprise Manager](#) page.

Configure Group: JMBGroup

This page lists all of the devices and filters configured in this group. To configure filters for a single device, use the tree control to navigate to that devices.

Devices

☑ TIP To delete a device from this group, or access the configuration for a device, use the tree control to the left.

Device Name	Status

Configured Filters

Order	Delete	Name	Move

Update Group

Group Name:

Event Collect Time: (in milliseconds)

2. 「新規デバイスの追加」をクリックして、デバイス・グループのデバイス（ドライバ・インスタンス）を作成します。「検索と選択」ページが表示され、リポジトリ内のドライバが一覧表示されます。

図 3-10 デバイスのドライバの選択

Search and Select: Add new device

Once you select an appropriate driver, be sure to configure it correctly.

To add a new device, choose a driver from the list below.

Search

Search

Results

Select Driver	Description
<input type="radio"/> Samsys Driver	Samsys Driver
<input type="radio"/> BarcodeDriver	BarcodeDriver
<input type="radio"/> Tyco Reader Driver	Tyco Reader Driver
<input checked="" type="radio"/> Alien Reader Driver	Alien Reader Driver
<input type="radio"/> Intermec BRI Driver	Intermec BRI Driver
<input type="radio"/> Simple Audio Driver	Simple Audio Driver
<input type="radio"/> Matrics Driver	Matrics Driver
<input type="radio"/> HtmlDriver	HtmlDriver
<input type="radio"/> Console Driver	Console Driver
<input type="radio"/> Intermec Reader Driver	Intermec Reader Driver
<input type="radio"/> Prolite Driver	Prolite Driver
<input type="radio"/> PatliteDriver	PatliteDriver
<input type="radio"/> AnimationDriver	AnimationDriver
<input type="radio"/> Edge Echo Driver	Edge Echo Driver
<input type="radio"/> LpmlDriver	LpmlDriver
<input type="radio"/> Edge Simulator Driver	Edge Simulator Driver

New device name:

select_device.gif はデバイスの「検索と選択」ページを示しています。

ヒント: ドライバを見つけるには、「検索」フィールドにドライバの名前（またはドライバ名の一部）を入力して「実行」をクリックします。

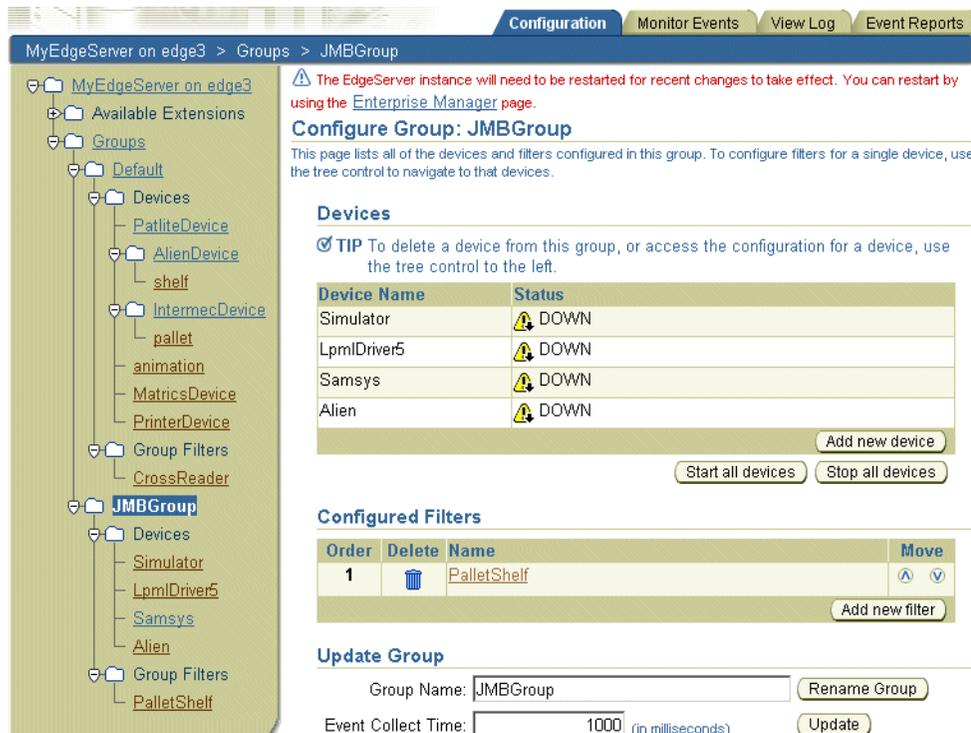
- a. デバイスのドライバを選択します。Oracle Sensor Edge Server でサポートされているドライバを表 3-3 に示します。これらのドライバの詳細、これらのドライバからデバイスを作成する場合に必要な構成パラメータ、およびこれらのドライバでサポートされているモデルについては、第 6 章「デバイス、フィルタ・インスタンスおよびデバイスパッチャの構成」を参照してください。

表 3-3 サポートされているドライバ

リーダー	インジケータ、通知および表示	プリンタ	その他
リーダーをサポートするドライバには次のものがある。	通知および表示をサポートするドライバには次のものがある。	LpmlDriver (「LpmlDriver ベースのデバイスの構成」を参照)	Edge Echo ドライバ (「Edge Echo ドライバ・ベースのインスタンスの構成」を参照)
Alien リーダー・ドライバ (「Alien リーダー・ドライバ・ベースのデバイスの構成」を参照)	AnimationDriver (「AnimationDriver のインスタンスの構成」を参照)		
BarcodeDriver (「BarcodeDriver ベースのデバイスの構成」を参照)	ConsoleDriver (「ConsoleDriver のインスタンスの構成」を参照)		
Intermec BRI ドライバ (「Intermec BRI ドライバによりサポートされるデバイスの構成」を参照)	Edge Simulator ドライバ (「Edge Simulator ドライバ・ベースのデバイスの構成」を参照)		
Intermec リーダー・ドライバ (「Intermec リーダー・ドライバ・ベースのデバイスの構成」を参照)	PatliteDriver (「PatliteDriver ベースのデバイスの構成」を参照)		
Matrics ドライバ (「Matrics ドライバ・ベースのデバイスの構成」を参照)	Prolite ドライバ (「Prolite ドライバ・ベースのデバイスの構成」を参照)		
Samsys ドライバ (「Samsys ドライバ・ベースのデバイスの構成」を参照)			
Tyco リーダー・ドライバ (「Tyco リーダー・ドライバ・ベースのデバイスの構成」を参照)			

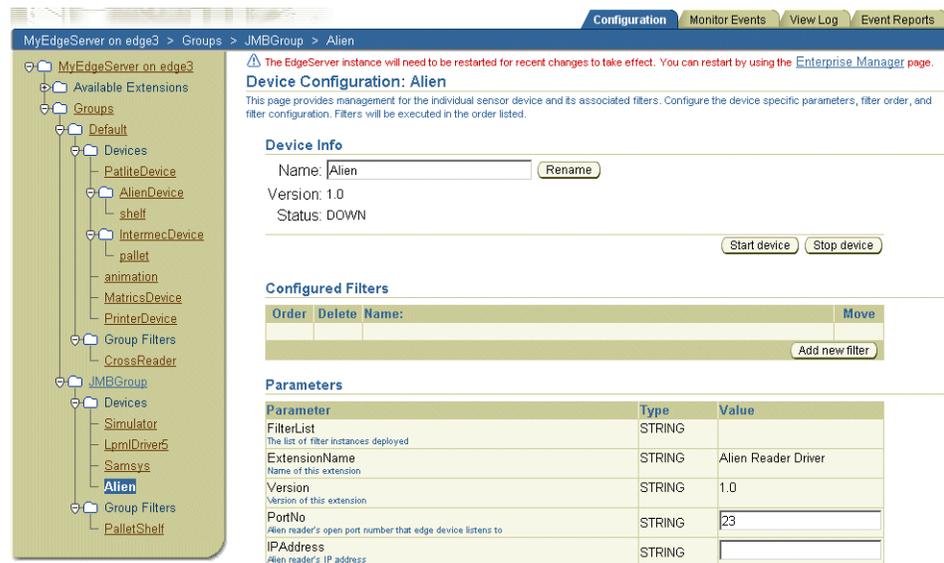
- b. 必要な場合、「新規デバイス名」フィールドにデバイスの名前を入力します。デバイスに名前を割り当てない場合、Oracle Sensor Edge Server によりデフォルト名が割り当てられます。
- c. 「選択」をクリックします。「グループの構成」ページが再び表示され、「デバイス」セクションにデバイスが一覧表示されます。また、ナビゲーション・ツリーにもデバイスが表示されます。

図 3-11 デバイス・グループに追加されたデバイス



3. デバイスをツリーから選択することにより、デバイスの構成を開始します。「デバイス構成」ページが表示され (図 3-12)、ドライバに固有のパラメータが表示されます。
 - a. パラメータを定義します。ドライバ・パラメータの詳細は、「デバイスの構成」を参照してください。
 - b. 「変更の保存」をクリックします。

図 3-12 デバイス・グループのデバイスの構成



4. 「新規フィルタの追加」をクリックして、デバイスにフィルタ（フィルタ・インスタンス）を追加します。「検索と選択」ページが表示され（図 3-13）、リポジトリ内のフィルタが一覧表示されます。Oracle Edge Sensor Server に組み込まれているデバイス・レベルおよびグループ・レベルのフィルタを表 3-5 に示します。

図 3-13 フィルタの選択

Search and Select: Add new filter

Once you select an appropriate filter, be sure to configure it correctly.

To add a new filter instance, choose a filter from the list below.

Search

Search

Results

Select Filter	Description
<input type="radio"/> Debug Filter	Debug Filter
<input type="radio"/> Pass Filter	Pass Filter
<input type="radio"/> MovementFilter	distance filter for rtls system
<input type="radio"/> Shelf Filter	Shelf Filter
<input type="radio"/> Cross Reader Filter	Cross Reader Filter
<input type="radio"/> JavaScriptFilter	JavaScript Filter
<input type="radio"/> RegexFilter	Regular Expression Filter
<input type="radio"/> Check Tag Filter	Check Tag Filter
<input checked="" type="radio"/> Pallet Shelf Filter	Pallet Shelf Filter
<input type="radio"/> Pallet Pass Filter	Pallet Pass Filter

New filter name:

- a. デバイスのフィルタ・インスタンスを選択します。

ヒント: フィルタを見つけるには、「検索」フィールドにフィルタの名前（またはフィルタ名の一部）を入力して「実行」をクリックします。

- b. 必要な場合、「新規フィルタ名」フィールドにフィルタの名前を入力します。フィルタ・インスタンスに名前を割り当てない場合、Oracle Sensor Edge Server によりデフォルト名が割り当てられます。
- c. 「選択」をクリックします。選択したフィルタが、デバイスの「構成」ページの「構成済フィルタ」セクションに表示されます（図 3-14）。また、デバイスの下のツリーにも表示されます。

図 3-14 デバイスへのフィルタ・インスタンスの追加

The screenshot displays the configuration page for a device named 'Alien'. On the left, a navigation tree shows the hierarchy: MyEdgeServer on edge3 > Groups > JMBGroup > Alien. The main content area includes a warning: 'The EdgeServer instance will need to be restarted for recent changes to take effect. You can restart by using the Enterprise Manager page.' Below this is the 'Device Configuration: Alien' section, which provides management for the device and its filters. The 'Device Info' section shows the device name as 'Alien', version as '1.0', and status as 'DOWN'. There are 'Start device' and 'Stop device' buttons. The 'Configured Filters' section contains a table with two filters: 'AlienPalletFilter' (Order 1) and 'AlienPalletShelfFilter' (Order 2). The 'Parameters' section shows a table with parameters like 'FilterList' and 'ExtensionName'.

Device Info

Name:

Version: 1.0

Status: DOWN

Configured Filters

Order	Delete	Name:	Move
1	<input type="button" value="Delete"/>	AlienPalletFilter	<input type="button" value="Up"/> <input type="button" value="Down"/>
2	<input type="button" value="Delete"/>	AlienPalletShelfFilter	<input type="button" value="Up"/> <input type="button" value="Down"/>

Parameters

Parameter	Type	Value
FilterList <small>The list of filter instances deployed</small>	STRING	AlienPalletFilter, AlienPalletShelfFilter
ExtensionName <small>Name of this extension</small>	STRING	Alien Reader Driver

5. 「デバイス構成」ページの「構成済フィルタ」セクションの表またはツリーからフィルタ・インスタンスを選択することにより、フィルタ・インスタンスの構成を開始します。「フィルタ構成」ページが表示され (図 3-15)、選択したフィルタのパラメータが表示されます。
 - a. 必要な場合は、フィルタ・インスタンスの名前を変更します。
 - b. フィルタ・パラメータを定義します。フィルタ・パラメータの詳細は、「フィルタ・インスタンスの構成」を参照してください。
 - c. 「変更の保存」をクリックします。

注意: デバイス・グループの作成、編集または削除を完了した後、Oracle Sensor Edge Server をいったん停止してから再起動する必要があります。詳細は、「Oracle Sensor Edge Server インスタンスの起動と停止」を参照してください。

図 3-15 デバイスのフィルタ・インスタンスの作成

The screenshot shows the configuration page for the 'AlienPalletFilter'. The breadcrumb navigation is 'MyEdgeServer on edge3 > Groups > JMBGroup > Alien > AlienPalletFilter'. A warning message at the top states: 'The EdgeServer instance will need to be restarted for recent changes to take effect. You can restart by using the Enterprise Manager page.' The page title is 'Filter Configuration: AlienPalletFilter'. Below the title, it says: 'This page allows you to configure the parameters for your filter. Edit the values below and click 'save' to commit your changes. You can return to the parent device by using the tree control on the left.'

Filter Info

Name:

Version: 1.0

Parameters

Parameter	Type	Value
ExtensionName <small>Name of this extension</small>	STRING	Pallet Shelf Filter
Version <small>Version of this extension</small>	STRING	1.0
ExitEventThresholdTime <small>Time elapsed in milliseconds since a tag has been seen last time</small>	STRING	<input type="text" value="3000"/>
EventCollectControlTime <small>Time elapsed in milliseconds since a new tag has been detected last time</small>	STRING	<input type="text" value="1500"/>

デバイス・グループへのフィルタの追加

必要な場合、次の手順でデバイス・グループにフィルタを割り当てます。

1. ツリーからデバイス・グループを選択します。「グループの構成」ページが表示されます。
2. 「新規フィルタの追加」をクリックします。「検索と選択」ページが表示され (図 3-13)、リポジトリ内のフィルタが一覧表示されます。Oracle Edge Sensor Server のデバイス・レベルおよびグループ・レベルのフィルタを表 3-5 に示します。
3. デバイス・グループのフィルタ・インスタンスを選択します。

ヒント: フィルタを見つけるには、「検索」フィールドにフィルタの名前 (またはフィルタ名の一部) を入力して「実行」をクリックします。

4. 必要な場合、「新規フィルタ名」フィールドにフィルタの名前を入力します。フィルタ・インスタンスに名前を割り当てない場合、Oracle Sensor Edge Server によりデフォルト名が割り当てられます。
5. 「選択」をクリックします。「グループの構成」ページの「構成済フィルタ」セクションと、選択したデバイス・グループの「グループ・フィルタ」の下に、フィルタが表示されます。
6. 「グループの構成」ページの「構成済フィルタ」セクションの表またはツリーからフィルタ・インスタンスを選択することにより、フィルタ・インスタンスの構成を開始します。「フィルタ構成」ページが表示され (図 3-15)、選択したフィルタのパラメータが表示されます。
7. 必要な場合は、フィルタ・インスタンスの名前を変更します。
8. フィルタ・パラメータを定義します。フィルタ・パラメータの詳細は、「フィルタ・インスタンスの構成」を参照してください。
9. 「変更の保存」をクリックします。

デバイス・グループの編集

「グループの構成」ページ (図 3-8) では、デバイス・グループのプロパティを編集できます。このページは、ナビゲーション・ツリー (図 3-2) 内のデバイス・グループをクリックすると表示され、次の操作を実行できます。

- デバイス・グループの名前変更
- デバイス・グループで使用するデバイスおよびフィルタの更新
- デバイス・グループに割り当てられているデバイスの起動と停止
- デバイス・グループの削除

デバイス・グループの名前変更

デバイス・グループの名前を変更するには、次のようにします。

1. 「グループ名」フィールドに、デバイス・グループの新しい名前を入力します。
2. 「グループの名前変更」をクリックします。新しいグループ名がツリー内に表示されます。

デバイス・グループで使用するデバイスおよびフィルタの更新

デバイスおよびフィルタ・インスタンスをデバイス・グループに追加するには、「新規デバイスの追加」および「新規フィルタの追加」ボタンを使用します (「デバイス・グループの作成」を参照)。デバイスおよびフィルタ・インスタンスの割当てを完了した後、「更新」をクリックします。

デバイス・グループに割り当てられているデバイスの起動と停止

新しいデバイスやフィルタ・インスタンスの追加などの操作を実行する場合は、そのたびにデバイス・グループをいったん停止して起動する必要があります。デバイス・グループに属するデバイスを起動および停止するには、「すべてのデバイスの起動」ボタンおよび「すべてのデバイスの停止」ボタンを使用します。

デバイス・グループの削除

「グループの削除」ボタンを使用すると、Oracle Sensor Edge Server からデバイス・グループを削除できます。デバイス・グループを削除すると、そのグループに対して構成されているデバイスおよびフィルタ・インスタンスもすべて削除されます。デバイス・グループを削除した後、変更を有効にするには、Oracle Sensor Edge Server 全部を再起動する必要があります。

Oracle Sensor Edge Server インスタンスの起動と停止

Oracle Sensor Edge Server インスタンスを変更した後、Oracle Sensor Edge Server を再起動して変更をコミットする必要があります。Oracle Sensor Edge Server をいったん停止して再起動するには、次のいずれかの方法を使用します。

- [opmnctl](#) を使用した Oracle Sensor Edge Server インスタンスの停止と起動
- OracleAS Enterprise Manager を使用した Oracle Sensor Edge Server インスタンスの再起動

opmnctl を使用した Oracle Sensor Edge Server インスタンスの停止と起動

Oracle Sensor Edge Server の一般的な設定やディスパッチャを変更したり、デバイス・グループを変更した後で、Oracle Sensor Edge Server を停止して再起動するには、`opmnctl shutdown` および `opmnctl startall` コマンドを使用します。詳細は、『Oracle Process Manager and Notification Server 管理者ガイド』を参照してください。

OracleAS Enterprise Manager を使用した Oracle Sensor Edge Server インスタンスの再起動

Oracle Sensor Edge Server インスタンスを変更した場合、SES コンソールに、OracleAS Enterprise Manager を使用して Oracle Sensor Edge Server を再起動するように通知するメッセージ (図 3-16) が表示されます。このメッセージには、OracleAS Enterprise Manager へのリンクが設定されています。

図 3-16 Oracle Enterprise Manager の再起動メッセージ



Oracle Sensor Edge Server インスタンスを再起動するには、次のようにします。

1. メッセージ (図 3-16) 内の「Enterprise Manager」リンクをクリックします。OracleAS Enterprise Manager のログイン・ページが表示されます。
2. OC4J のユーザー名およびパスワードを入力すると、「クラスタ・トポロジ」ページが表示されます。
3. ホームページ (図 3-17) に移動します。
4. 「アプリケーション」をクリックします。
5. 「edge」を選択します。
6. 「再起動」をクリックします。

注意：無効な `.jms.xml` に関するエラー・メッセージは、通常、OC4J の異常終了、OC4J のクラッシュ、または OC4J を実行しているサーバーの IP アドレスの変更によって発生します。

異常終了の後に OC4J JMS サーバーの起動に問題が発生した場合は、他の OC4J JMS サーバーが稼働中で同じ永続性ファイルを使用していないかをまず確認してください。ORACLE_HOME/j2ee/instance_name/persistence ディレクトリにロック・ファイルがあれば削除し、サーバーを再起動します。

問題が解決しない場合は、`.jms.xml` ファイルが有効であることを確認してください。

それでも問題が解決しない場合は、`persistence` ディレクトリから `.jms.state` ファイルを削除して、再起動してください。このファイルを削除すると、トランザクション情報が失われる可能性があります。『Oracle Containers for J2EE サービス・ガイド』の第3章「Oracle Enterprise Messaging Service (OEMS)」、リソース・プロバイダに関する項の異常終了の説明も参照してください。

図 3-17 OracleAS Enterprise Manager のホームページ

The screenshot shows the OracleAS Enterprise Manager 10g Application Server Control page. The breadcrumb trail is: Cluster Topology > Application Server: edge.edge-test1.us.oracle.com > OC4J: home. The page title is 'Application Server Control' with links for Setup, Logs, Help, and Logout. The page was refreshed on Dec 6, 2005 9:21:34 AM PST. The navigation tabs include Home, Applications (selected), Web Services, Performance, and Administration. A message states: 'This page shows the J2EE applications and application components (EJB Modules, WAR Modules, Resource Adapter Modules) deployed to this OC4J instance.' The 'View' dropdown is set to 'Applications'. Action buttons include Start, Stop, Restart, Undeploy, Redeploy, and Deploy. A table lists the deployed applications:

Select	Name	Status	Start Time	Active Requests	Request Processing Time (seconds)	Active EJB Methods	Application Defined MBeans
<input type="radio"/>	default	↑	Dec 2, 2005 4:05:40 PM PST	0	0.00	0	
<input type="radio"/>	ascontrol	↑	Dec 2, 2005 4:05:40 PM PST	1	0.00	0	
<input type="radio"/>	bc4j	↑	Dec 2, 2005 4:05:40 PM PST	0	0.00	0	
<input checked="" type="radio"/>	edge	↑	Dec 2, 2005 4:05:40 PM PST	0	0.00	0	

注意：OC4J リリース 10.1.2 を再起動するには、SES コンソールで OC4J をいったん停止してから、`Java_HOME/bin/java -jar oc4j.jar` を使用して OC4J を再び起動します。

各デバイスの起動と停止

デバイスを更新するには、最初にデバイスを停止する必要があります。デバイスを停止または起動するには、「デバイス構成」ページ (図 3-12) から「**デバイスの起動**」および「**デバイスの停止**」ボタンを使用します。このページには、デバイスのステータスも表示されます。ステータス・メッセージ (表 3-4 を参照) に、ドライバの現在の状態、またはクラッシュ前のドライバの状態が説明されます。

表 3-4 デバイス・ステータスのメッセージ

ステータス	定義
インスタンス化済	デバイスの初期化が正常に完了 (doInit() エントリ・ポイントを通過)。
初期化の失敗	初期化に失敗し、デバイスが停止 (doInit() メソッドが失敗)。
開始	デバイスが起動 (doStart() メソッドに入った)。
起動に失敗しました	起動に失敗 (start() メソッドが失敗)。
停止	デバイスが停止。
停止に失敗しました	デバイスの停止またはリソースのクリーンアップに失敗。
終了	デバイスが処理を完了。固定のタスク・セットがあり、タスクを完了すると通常自動的に停止するデバイス (たとえば、シミュレータなどは、シミュレータ・ファイルの処理を完了すると自動的に停止する) で使用されるステータス。
接続	デバイスは接続を試行中。
接続に失敗しました	デバイスが接続に失敗。
設定	デバイスはリソースおよび接続を設定中。
設定の失敗	設定中にデバイスに障害が発生。
実行	デバイスは実行中。
実行の失敗	実行中にデバイスに障害が発生。
無効	デバイスは無効化。

フィルタの管理

フィルタは、特定のデバイスまたはデバイス・グループのいずれかにアタッチできます。**Cross-Reader Redundant** フィルタなどの一部のフィルタは、グループ・レベル・フィルタとして記述されているため、デバイス・グループにのみアタッチできます。デバイス・グループ・フィルタを使用すると、エッジ・デバイスに配信される前のイベントをフィルタリングできます。デバイス・グループ専用で記述されているフィルタもあれば、デバイス・レベルのフィルタリング専用で記述されていて、特定のデバイスにアタッチされている場合にのみ機能するようなフィルタもあります。

「デバイス構成」の「構成済フィルタ」表 (図 3-11) および「グループの構成」(図 3-14) ページでは、フィルタ・インスタンスの追加、削除または並替えを行うことができます。「**新規フィルタの追加**」ボタンを使用すると、他のフィルタ・インスタンスをデバイスまたはデバイス・グループに追加できます。詳細は、「**デバイス・グループの作成**」および「**デバイス・グループへのフィルタの追加**」を参照してください。表のごみ箱アイコンをクリックすると、デバイスまたはデバイス・グループからフィルタ・インスタンスを削除できます。表の矢印を使用すると、フィルタ・インスタンスの優先度を設定できます。

フィルタ・インスタンスの優先度の設定

フィルタ・インスタンスの順序は、データのフィルタリングの結果を決定します。たとえば、IDをイベント配列にグループ化し、それを1つの項目として処理するグループ・フィルタと、特定のタグ TagXYZ を持つイベントをフィルタリングで除外するタグ・フィルタが、デバイスまたはデバイス・グループに割り当てられているとします。この場合、タグ・フィルタをグループ・フィルタより先に適用すると、タグ・フィルタによって TagXYZ を持つイベントが除外された後で、Oracle Sensor Edge Server は、検出された順序に基づいてグループ化されたイベントを受信します。フィルタの順序を逆にする（つまり、グループ・フィルタをタグ・フィルタより前に置く）と、タグ・フィルタのフィルタリングでは、TagXYZ を持つイベントではなくグループ・イベントのみを調べるため、何も除外されません。

「構成済フィルタ」表 (図 3-14) で、フィルタ・インスタンスを選択し、矢印を使用して上下に移動することにより、フィルタ・インスタンスを並替えることができます。

デバイスまたはデバイス・グループのフィルタ・インスタンスの管理

フィルタ・インスタンスは、インスタンス化されたフィルタ・オブジェクトです。フィルタがデバイス（またはデバイス・グループ）に適用されている場合は、デバイスまたはデバイス・グループがフィルタを使用できるようにフィルタ・インスタンスが作成されます。

独自のフィルタを開発してアップロードすることもできますが（「Oracle Sensor Edge Server インスタンスの拡張機能の追加」を参照）、Oracle Sensor Edge Server には複数のフィルタが付属しています（表 3-5 を参照）。

表 3-5 Oracle Sensor Edge Server に組み込まれているフィルタ

フィルタ名	機能	デバイス・グループに適用されるか。(グループ・レベルのフィルタリングのサポート)	デバイスに適用されるか。(デバイス・レベルのフィルタリングのサポート)
Check Tag ID フィルタ	指定された間隔中にデバイスがタグを読み取っているかどうかを確認する診断ツール。「Check Tag ID フィルタの構成」も参照。	×	○
Cross-Reader Redundant フィルタ	デバイス・グループに属するデバイスから送信された冗長イベントをブロック。「Cross-Reader Redundant フィルタの使用」も参照。	○	×
Debug フィルタ	システムを通過するイベントを追跡し、これらのイベントをログ・ファイルに書き込む。「Debug フィルタの使用」も参照。	○	○
JavaScript フィルタ	スクリプト言語でフィルタ・ロジックを記述できる。ソース・スクリプトの変更内容は動的にロードされるため、サーバーやその他のコンポーネントを再起動する必要がない。このフィルタは、Mozilla Rhino (http://www.mozilla.org/rhino/) など、スクリプトを実行する外部のスクリプト化エンジンに依存する。「JavaScript フィルタの構成」も参照。	○	○
Movement フィルタ	Real Time Location System 測定を使用して、エラーや干渉による急増や急激な運動変化を平坦にすることにより移動追跡を円滑化。「Movement フィルタの構成」も参照。	×	○

表 3-5 Oracle Sensor Edge Server に組み込まれているフィルタ (続き)

フィルタ名	機能	デバイス・グループに適用されるか。(グループ・レベルのフィルタリングのサポート)	デバイスに適用されるか。(デバイス・レベルのフィルタリングのサポート)
Pallet Pass-Thru フィルタ	コンテナまたはパレットにあるアイテムのすべてのタグ ID を確認できるようにする。「 Pallet Pass Thru フィルタの構成 」も参照。	×	○
Pallet Shelf フィルタ	新しいコンテナまたはパレットがデバイス・リーダーの読取り範囲またはゲートに入ったこと、または出たことを通知するイベントを送信。「 Pallet Shelf フィルタの構成 」も参照。	×	○
Pass フィルタ	タグがデバイスのゲートウェイまたは伝達範囲を通過したことをアプリケーションに通知。このフィルタはイベントのブロックも行うため、デバイスによってタグが検出されると、複製イベントではなく、1つのイベントのみが生成される。「 Pass フィルタの構成 」も参照。	×	○
Polygon フィルタ	Real Time Location System デバイスによってレポートされたすべての移動測定をフィルタリングで除外し、タグが定義済ポリゴンを出入りしたときのみイベントを生成。ポリゴンは、頂点とカッコを定義する x,y 座標のセットを使用して定義する。たとえば、((x,y), (x,y), ...), (...), ... のように定義する。「 Polygon フィルタの構成 」も参照。	×	○
Regex フィルタ	ストリームを削除するか、ストリームの通過を許可するためのタグを検索する正規表現検索を実行。このフィルタを使用すると、フィルタのパターン・セットを定義してイベントの任意の範囲を検索できる。フィルタにより検索基準の一致が見つかったら、イベントがシステムを通過することが許可される。一致が見つからない場合、そのイベントがフィルタリングで除外される。「 Regex フィルタの定義 」も参照。	○	○
Shelf フィルタ	アイテムがデバイス・リーダーの読取り範囲またはゲートに入ったこと、または出たことを通知。「 Shelf フィルタの構成 」も参照。	×	○

イベント・データの監視

デバイスおよびフィルタ・インスタンスは、発生した内容を説明するイベント・メッセージを使用して通信します。たとえば、デバイスは次のようなメッセージを送信して、起動したことを他のコンポーネントに通知します。このようなイベント・メッセージは、デバイスまたはフィルタ・インスタンスの各タイプに固有です。Oracle Sensor Edge Server 内部では、どのデバイスもイベントを直接送受信できます。一般に、コンポーネント間のイベント・フローの方向は次の2つです。

- **インバウンド**: デバイスおよびフィルタ・インスタンスがイベントを現行ディスパッチャに送信します。
- **アウトバウンド**: アプリケーションがイベントをデバイスおよびフィルタ・インスタンスに送信します。これらのイベントは、デバイスにコマンドを送信することが多いため、命令イベントと呼ばれることもあります。

イベント・データの表示

SES コンソールでは、「監視イベント」および「イベント・レポート」タブに表示されるイベント・データにより、Oracle Sensor Edge Server の状態を確認できます。「監視イベント」タブ (図 3-20) には、現在キューに格納されているデータが表示されます。また、「イベント・レポート」タブには、センサー・データ・リポジトリに格納されているデータが表示されます。

「監視イベント」タブと「イベント・レポート」タブに表示されるインバウンドおよびアウトバウンドのフィールドを表 3-6 に説明します。

表 3-6 インバウンドおよびアウトバウンドのイベント・データ

トピック	説明
タイプ	イベント・タイプのテキスト表示。「RFID 監視」、「RTLS 監視」、「温度」などのタイプがある。
説明	イベント・サブタイプのテキスト表示。
デバイス名	イベントを生成したデバイスの名前。
データ	イベントのペイロード。
時間	イベントが生成された時刻。

各イベントの表示

SES コンソールでは、「イベント・タイプ」、「サブタイプ」、「関連 ID」フィールドなど、個々のインバウンド・イベントまたはアウトバウンド・イベントに関する詳細情報を表示できます。この情報にアクセスするには、「詳細」アイコンをクリックします。

図 3-18 「詳細」アイコン



「イベントの詳細」ページが表示され、選択したイベントのメタデータおよびペイロードの情報が表示されます。

図 3-19 「イベントの詳細」 ページ

ORACLE
Sensor Edge Server

other servers logout help

Configuration Monitor Events View Log Event Reports

MyEdgeServer > Event Details

Event Details

This page lists all details for an individual event. Use Go Back button to return to the previous page.

Metadata

Event ID:	Description: Instruction (subtype=2)
Device Name: Sim	Type: 100
Source Name:	Subtype: 2
Site Name:	Correlation ID:

Payload

Tag ID:

Data:

Timestamp: Nov 17, 2005

Go Back

Copyright (c) 1996, 2005, Oracle. All rights reserved.

メタデータ

「イベントの詳細」画面の「メタデータ」セクションに、次のルーティング情報およびイベントの発生場所に関連する情報が一覧表示されます。

- イベント ID

未処理のイベント・データと、センサー・データ・リポジトリ内に格納されているイベント・データでは、「イベント ID」の意味が異なります。未処理のデータ（図 3-20 の「監視イベント」タブの詳細ページに表示されるデータ）の場合、「イベント ID」はイベントをトリガーしたタグ ID です。「イベント・レポート」ページ（図 3-22、図 3-23 および図 3-24）に表示される、センサー・データ・リポジトリに格納されているイベント・データでは、「イベント ID」フィールドはイベントを一意に識別します。

- デバイス名

イベントを生成した拡張機能（デバイスまたはフィルタ・インスタンス）あるいはアプリケーションの名前です。

- ソース名

このフィールドは、イベントの発信元を識別します。このフィールドはオプションで、クライアントによって設定されます。

- サイト名

最初にメッセージを生成したサイトです。

- タイプ

生成されたイベントのタイプに対応する数値です。イベント・タイプは次のように分類されます。

- 0 ~ 99: システム・メッセージ
- 100 ~ 199: デバイスへの汎用命令
- 200 ~ 299: デバイスからの測定
- 500 ~ 599: カスタム・メッセージ（未登録）

「タイプ」フィールドに表示される登録済のイベント・タイプを表 3-7 に示します。

- サブタイプ
イベントのサブタイプに対応する数値です。「サブタイプ」フィールドに表示される値を表 3-7 に示します。
- 説明
イベント・タイプおよびそのイベント・サブタイプのテキスト説明です。たとえば、イベント・サブタイプが 200（「RFID 監視」）でそのサブタイプが 2（タグが読取り範囲を出た）の場合、「説明」フィールドに「RFID フィールド外」と表示されます。イベント・タイプが 200 でイベント・サブタイプが 9（明示的なメッセージがない）場合、「説明」フィールドには、イベント・タイプ 200 のメッセージ（「RFID 監視」）が表示され、その後にイベント・サブタイプが「RFID 監視」（サブタイプ 9）として表示されます。「説明」フィールドに表示されるテキストを表 3-7 に示します。
- 関連 ID
このイベント・スレッドを識別する一意の ID です。関連 ID は、特定のクライアントへのメッセージ・レスポンス（デバイスが機能しているかどうかの確認など）に使用されます。クライアントから返されたメッセージ・レスポンスはすべて同じ ID を持ちます。この ID はクライアントにより設定されます。送信イベント・メッセージと受信イベント・メッセージはこの ID によって関連付けられるため、この ID をデバイスのパラメータとして使用することはできません。このフィールドはオプションです。

表 3-7 登録済のイベント・タイプおよび関連するサブタイプ

イベント・タイプ	説明	(イベント・レポートに表示される) タイプの説明	サブタイプ
0	不明。0 の値は、イベントまたはシステム内部イベントが正しくないことを示す。	システム・イベント	N/A
1	命令またはコマンド。メッセージ・イベント。	インストラクション・リターン・コード	次のサブタイプがある。 <ul style="list-style-type: none"> ■ 1 および 2: エラー・メッセージ ■ 3: サポートされない機能 ■ 4: Oracle Sensor Edge Server 起動メッセージ ■ 5: Oracle Sensor Edge Server 停止メッセージ ■ 10: 通知メッセージ
100	汎用命令	インストラクション	次のサブタイプがある。 <ul style="list-style-type: none"> ■ 0: デバイス・ステータスの取得 ■ 1: デバイスの起動 ■ 2: デバイスの停止
101	RFID 命令	インストラクション書込み	次のサブタイプがある。 <ul style="list-style-type: none"> ■ 0: タグへの書込み ■ 1: タグの破壊 ■ 2: 電界強度の取得 ■ 3: タグ・ペイロードの読取り

表 3-7 登録済のイベント・タイプおよび関連するサブタイプ (続き)

イベント・タイプ	説明	(イベント・レポートに表示される) タイプの説明	サブタイプ
102	プリンタ命令	インストラクション印刷	次のサブタイプがある。 <ul style="list-style-type: none"> ■ 1: LPML を使用して印刷。このサブタイプは、「イベントの詳細」ページに「インストラクション印刷ラベル」として表示される。 ■ 2: RAW で印刷 (ダイレクト・ペイロード)。このサブタイプは、「イベントの詳細」ページに「インストラクション印刷 RAW」として表示される。
103	積層表示灯命令	インジケータ	次のサブタイプがある。 <ul style="list-style-type: none"> ■ 1: 積層表示灯 XML コマンド。このサブタイプは、「イベントの詳細」ページに「インジケータの再生」として表示される。 ■ 2: 「イベントの詳細」ページに「インジケータのレスポンス」として表示される。 ■ 3: 「イベントの詳細」ページに「インジケータの消去」として表示される。
104	オーディオ	オーディオ	1: 「イベントの詳細」ページに「オーディオ・インストラクション」として表示される。
200	RFID	RFID 監視	次のサブタイプがある。 <ul style="list-style-type: none"> ■ 0: 正常。 ■ 1: タグが読取り範囲に入った。「イベントの詳細」ページに「RFID 監視」として表示される。 ■ 2: タグが読取り範囲を出た。「イベントの詳細」ページに「RFID フィールド外」として表示される。 ■ 3: タグが読取り範囲を通過。「イベントの詳細」ページに「RFID パス」として表示される。 ■ 4: タグ・グループが読取り範囲に入った。「イベントの詳細」ページに「フィールド内の RFID パレット」として表示される。 ■ 5: タグ・グループが読取り範囲を出た。「イベントの詳細」ページに「フィールド外の RFID パレット」として表示される。 ■ 6: タグ・グループが読取り範囲を通過。「イベントの詳細」ページに「RFID パレット・パス」として表示される。 ■ 7: コンテナ・イベント。「イベントの詳細」ページに「RFID コンテナ」として表示される。

表 3-7 登録済のイベント・タイプおよび関連するサブタイプ (続き)

イベント・タイプ	説明	(イベント・レポートに表示される) タイプの説明	サブタイプ
201	RTLS	RTLS	次のサブタイプがある。 <ul style="list-style-type: none"> ■ 1: 「イベントの詳細」 ページに「RTLS 監視」として表示される。 ■ 2: 「イベントの詳細」 ページに「ポリゴン内の RTLS」として表示される。 ■ 3: 「イベントの詳細」 ページに「ポリゴン外の RTLS」として表示される。 ■ 4: 「イベントの詳細」 ページに「RTLS 移動済」として表示される。
202	物理的接触	物理的な接続	1: 「イベントの詳細」 ページに「物理的な切断」として表示される。
203	温度	温度	次のサブタイプがある。 <ul style="list-style-type: none"> ■ 1: 「イベントの詳細」 ページに「温度の読取り」として表示される。 ■ 2: 「イベントの詳細」 ページに「温度の変更」として表示される。
204	湿度	湿度	次のサブタイプがある。 <ul style="list-style-type: none"> ■ 1: 「イベントの詳細」 ページに「湿度の読取り」として表示される。 ■ 2: 「イベントの詳細」 ページに「湿度の変更」として表示される。
205	重み	重み	1: 「イベントの詳細」 ページに「重みの読取り」として表示される。
206	不正変更	改ざん	1: 「イベントの詳細」 ページに「改ざん」として表示される。
208	メッセージ・ボード	メッセージ・ボード	N/A
209	PLC	自動化	N/A
210	プリンタ・レスポンス	プリンタのレスポンス	2: 印刷が正常に完了。「イベントの詳細」 ページに「印刷成功」として表示される。
211	有害センサー	有害	N/A
212	バーコード	バーコード	1: 「イベントの詳細」 ページに「バーコード読取り」として表示される。
213	放射	放射	N/A

ペイロード

「イベントの詳細」 ページの「ペイロード」 セクションには、次のフィールドが表示されます。

- タグ ID
このイベントで説明された項目の ID です。このフィールドのテキスト値により、イベント命令に対するタグ（読取りまたはターゲット）が識別されます。
- データ
タグ・データ（またはイベントのペイロード）です。このフィールドはオプションです。
- タイムスタンプ
イベントが生成された日時です。

未処理のイベント・データの表示

SES コンソールの「監視イベント」タブから、このイベント・データをリアルタイムで表示できます。「監視イベント」タブをクリックすると、「監視イベント」ページ (図 3-20) が表示されます。

図 3-20 「監視イベント」ページ

The screenshot shows the Oracle Sensor Edge Server interface. At the top, there's the Oracle logo and 'Sensor Edge Server' text. Below that, there are navigation tabs: 'Configuration', 'Monitor Events' (selected), 'View Log', and 'Event R'. The main content area is titled 'Monitor Events' and contains the following information:

This page monitors events that have recently moved through the Edge Server queue.
 The Inbound Queue displays events from the application and the Outbound Queue displays events from the device drivers.
 TIP This page will only function properly if the log level is set to 'monitor' or higher.

Inbound Queue

Details	Type	Description	ID	Device Name	Data	Time
	Instruction (subtype=2)			Sim		Nov 17, 2005
	Instruction (subtype=1)			Sim		Nov 17, 2005
	Instruction (subtype=2)			Sim		Nov 17, 2005
	Instruction (subtype=2)			Sim		Nov 17, 2005
	Instruction (subtype=2)			Sim		Nov 17, 2005
	Instr write (subtype=1)		1	Sim	data	Nov 17, 2005
	Instr write (subtype=1)		1	Sim	data2	Nov 17, 2005

Outbound Queue

Details	Type	Description	ID	Device Name	Data	Time
	RFID Observation		95001C0001000000	Simulator1	sim1-0-0-Thu Nov 17 15:37:38 PST 2005	Nov 17, 2005
	RFID Outfield		95001C0001800000	Simulator2	sim2-0-2-Thu Nov 17 15:37:39 PST 2005	Nov 17, 2005

「監視イベント」ページでは、キューを通過したインバウンド・イベント・データとアウトバウンド・イベント・データをすべて表示できます。このページに表示されるイベント・データはまだ処理されていません。つまり、センサー・データ・リポジトリに送信されていません。

ヒント: 「監視イベント」ページに表示されるイベント・データは動的であり、頻繁に変更されます。データが静的のままであると、デバイスがイベントの送信や命令イベントの受信を実行できなくなることがあるためです。

ログ情報の表示

「ログの表示」ページ (図 3-21) は、「ログの表示」タブからアクセスします。このページには、メイン・ページ (図 3-1) の「ログ・レベル」リストから選択した重大度レベルに従ってログに書き込まれたデータが表示されます。たとえば、メイン・ページから「エラー」を選択した場合、「ログの表示」ページにはエラー・レベルのメッセージのみが表示されます。メイン・ページから「すべて」を選択した場合、「ログの表示」ページには記録されたすべてのメッセージが表示されます。

図 3-21 「ログの表示」ページ



「ログの表示」ページを使用すると、日付によりログ・ファイルを選択したり、このページに表示する行数を設定できます。

ログ・データを表示するには、次のようにします。

1. 「ログ・ファイル」リストからログ・ファイルの日付を選択します。
2. 必要な場合、「フィルタ」リストからログ・レベルを選択します。
3. 必要な場合、表示する行数を「行」フィールドに入力します。
4. 「ログ・データの取得」をクリックします。

処理済のイベント・データの表示

SES コンソールでは、「イベント・レポート」タブからアクセスしたページを使用して、処理が終わってセンサー・データ・リポジトリに格納されているイベントを追跡できます。これらのページでは、タグ ID、デバイス名、またはイベントがキューを通過した期間に基づいてデータを取得できます。イベント・タイプやイベント・サブタイプなどの検索基準を含む問合せを作成すると、タグ ID およびデバイス名を再検索できます。

「イベント・レポート」タブをクリックすると、デフォルトで「タグの表示」ページ (図 3-22) が表示されます。

図 3-22 「タグの表示」ページ

The screenshot shows the Oracle Sensor Edge Server interface. At the top, there are navigation tabs: Configuration, Monitor Events, View Log, and Event Reports. The 'View Tags' page is active. It includes a search criteria section with a 'TIP' to leave fields blank. The search fields are: Tag ID (with a partial 'f' entered), Start Date (11/30/05), End Date (11/30/05), Start Time (12:00:00), and End Time (13:00:00). Below the search fields are 'Advanced Search' and 'Fetch Results' buttons. The 'Results' section shows a table with the following columns: Details, Type, Description, Device Name, Time, and Time Delta. The table is currently empty.

「タグの表示」ページのナビゲーション・ツリーを使用すると、他のタイプのイベント検索も選択できます。ツリーは、「イベント・レポート」タブからアクセスしたすべてのページに表示されます。

タグ ID によるイベントの検索

タグ ID によりイベントを取得するには、次のようにします。

1. タグ ID の一部を入力します。(これは like パターンです。)
2. 開始日を入力するか、日時エディタを使用して開始日を選択します。この条件はオプションです。このフィールドを空白のままにしておくと、この条件が検索から除外されます。
3. 開始時刻を (*hh:mm:ss* のように) 入力します。この条件はオプションです。このフィールドを空白のままにしておくと、この条件が検索から除外されます。
4. 終了日を入力するか、日時エディタを使用して終了日を選択します。この条件はオプションです。このフィールドを空白のままにしておくと、この条件が検索から除外されます。
5. 終了時刻を (*hh:mm:ss* のように) 入力します。この条件はオプションです。このフィールドを空白のままにしておくと、この条件が検索から除外されます。

ヒント: 「終了時間」フィールド内をクリックすると、「開始日」フィールドに入力した値が「終了日」フィールドに表示されます。

6. 「フェッチ結果」をクリックします。「結果」表に検索結果が表示されます。「詳細」アイコンをクリックすると、特定のイベントを表示できます。詳細は、「タグ ID およびデバイス名の再検索」を参照してください。検索に条件を追加するには、「拡張検索」をクリックします。

デバイス名によるイベントの検索

デバイス名によりイベントを取得するには、次のようにします。

1. デバイス名（またはデバイス名の一部）を入力します。これは like パターンです。
2. 開始日を入力するか、日時エディタを使用して開始日を選択します。この条件はオプションです。このフィールドを空白のままにしておくと、この条件が検索から除外されます。
3. 開始時刻を (hh:mm:ss のように) 入力します。この条件はオプションです。このフィールドを空白のままにしておくと、この条件が検索から除外されます。
4. 終了日を入力するか、日時エディタを使用して終了日を選択します。この条件はオプションです。このフィールドを空白のままにしておくと、この条件が検索から除外されます。
5. 終了時刻を (hh:mm:ss のように) 入力します。この条件はオプションです。このフィールドを空白のままにしておくと、この条件が検索から除外されます。

ヒント: 「終了時間」フィールド内をクリックすると、「開始日」フィールドに入力した値が「終了日」フィールドに表示されます。

6. 「フェッチ結果」をクリックします。「結果」表に検索結果が表示されます。「詳細」アイコンをクリックすると、特定のイベントを表示できます。詳細は、「[タグ ID およびデバイス名の再検索](#)」を参照してください。検索に条件を追加するには、「[拡張検索](#)」をクリックします。

タグ ID およびデバイス名の再検索

「タグの表示」および「デバイスの表示」ページの「[拡張検索](#)」ボタンを使用すると、問合せ文を作成して検索結果を絞り込むことができます。

デバイス名またはタグ ID の検索に問合せ文を追加するには、次のようにします。

1. タグ ID またはデバイス名の検索基準を入力します。必要な場合、検索の時刻および日付の制約を追加します。
2. 「[拡張検索](#)」をクリックします。「[拡張検索](#)」ページが表示され、タグ ID またはデバイス名に対して入力した検索基準が表示されます。

図 3-23 (タグ検索の)「拡張検索」ページ

The screenshot shows the Oracle Sensor Edge Server interface. At the top, there's a navigation bar with 'Configuration', 'Monitor Events', 'View Log', and 'Event Reports'. Below that, the 'Advanced Search' page is displayed. It includes a 'Search Criteria' table and an 'Add Statement' section.

Delete	Statement
	Tag ID is like ff
	and Time is greater than or equal to 11/30/05 12:00:00
	and Time is less than or equal to 11/30/05 13:00:00
	and Type is equal to 200
	and Subtype is equal to 5

Add Statement
Use the area below to build your report search criteria, then click the Fetch Results button.

Relational
 Select a query field
 Select an operator
 Input query value

3. 次の手順で問合せ文を作成します。
 - 「問合せフィールドの選択」リストを使用して、イベントのメタデータ（「イベント ID」、「デバイス名」、「ソース名」、「サイト名」、「タイプ」、「サブタイプ」、「関連 ID」など）、およびペイロード（「タグ ID」、「データ」および「時間」）を選択します。
 - 「問合せフィールドの選択」リストから選択した検索値を「問合せ値の入力」フィールドに入力した値と比較するための演算子を選択します。次のオプションがあります。
 - 次と等しい
 - 次と等しくない
 - 次より小さい
 - 次以下
 - 次より大きい
 - 次以上
 - 類似
 - 「問合せ値の入力」フィールドに、「問合せフィールドの選択」で選択したオプションに関連する値を入力します。たとえば、「問合せフィールドの選択」で「タイプ」を選択した場合、「問合せ値の入力」フィールドに 200 を入力します。
4. 「文の追加」をクリックします。「検索基準」セクションに問合せ文が表示されます。
5. 「フェッチ結果」をクリックします。検索文を使用してセンサー・データ・リポジトリの問合せが実行されます。要求されたデータが、ページの「結果」セクションに表示されます。データは、「イベント ID」および「データ」によって昇順と降順でソートできます。各イベントを表示するには、「詳細」アイコンをクリックします。詳細は、「各イベントの表示」を参照してください。

拡張検索の作成

「拡張検索」ページでは、イベント・データのメタデータおよびペイロード情報を使用してイベント・データを取得する文で構成された問合せを作成することにより、特定のイベント・データを取得できます。

図 3-24 「拡張検索」ページ

Reports > Advanced Search

[View Tags](#)
[View Device](#)
[Advanced Search](#)

Advanced Search

These report pages allows you to view a collection of events from the Sensor Data Archive.
This advanced search page allows you to generate a report using one or more search statements.

Search Criteria

Delete	Statement
	Tag ID is like f
	and Time is greater than or equal to 11/15/05 12
	and Time is less than or equal to 11/15/05 13
	and Type is equal to 200
	and Subtype is equal to 1

Add Statement

Use the area below to build your report search criteria, then click the Fetch Results button.

Relational Select a query field Select an operator Input query value

Results

Previous 1-10 of 791 Next 10

Details	Event ID	Type Description	Data
	1195361	RFID Observation	LPN1223
	1195362	RFID Observation	LPN1223

イベント・データに対する特定の検索基準を作成するには、次のようにします。

1. 「問合せフィールドの選択」リストを使用して、イベントのメタデータ（「イベント ID」、[「デバイス名」](#)、「ソース名」、「サイト名」、「タイプ」、「サブタイプ」、「[「関連 ID」](#) など）、およびペイロード（[「タグ ID」](#)、「データ」および「時間」）を選択します。
2. 「問合せフィールドの選択」リストから選択した検索値を「問合せ値の入力」フィールドに入力した値と比較するための演算子を選択します。次のオプションがあります。
 - 次と等しい
 - 次と等しくない
 - 次より小さい
 - 次以下
 - 次より大きい
 - 次以上
 - 類似
3. 「問合せ値の入力」フィールドに、「問合せフィールドの選択」で選択したオプションに関連する値を入力します。たとえば、「問合せフィールドの選択」で「タイプ」を選択した場合、「問合せ値の入力」フィールドに 200 を入力します。

ヒント： 問合せを完成して「[文の追加](#)」をクリックすると、「[リレーショナル](#)」リストが表示されます。このリストを使用すると、複合検索条件（AND、OR、NOT）を使用して問合せ文を結合できます。問合せ文を削除するには、ごみ箱アイコンを使用します。

4. 「[文の追加](#)」をクリックします。「[検索基準](#)」セクションに検索文が表示されます。
5. 必要な場合、さらに文を追加します。
6. 「[フェッチ結果](#)」をクリックします。選択した検索文を使用してセンサー・データ・リポジトリの問合せが実行されます。結果がページの「[結果](#)」セクションに表示されます。データは、「[イベント ID](#)」および「[データ](#)」によって昇順と降順でソートできます。各イベントを表示するには、「[詳細](#)」アイコンをクリックします。詳細は、「[各イベントの表示](#)」を参照してください。

Oracle Sensor Edge Server インスタンスの拡張機能の追加

拡張機能とは、拡張アーカイブ・ファイルにコンポーネントをパッケージ化することによって Oracle Sensor Edge Server にアップロードする、カスタムビルドのドライバ、ディスパッチャまたはフィルタです。拡張アーカイブ・ファイルは、ドライバ、フィルタまたはディスパッチャのすべてのクラス・ファイルとネイティブ・バイナリ、ならびにプロパティ・ファイルや静的データ・ファイルを含む JAR ファイルです。また、拡張アーカイブには、拡張アーカイブ記述子ファイル（拡張機能のロードと管理に関する Oracle Sensor Edge Server への命令を含む XML ファイル）が含まれます。

注意： <IsExtensionMonitorEnabled> の要素コンテンツを true に設定すると、Oracle Sensor Edge Server に拡張機能を動的にアップロードできます。Oracle Sensor Edge Server を再起動する必要はありません。ただし、拡張機能で作成されたインスタンスを Oracle Sensor Edge Server が使用するには、説明された手順に従って Oracle Sensor Edge Server を再起動する必要があります。

拡張アーカイブ・ファイル

ドライバ、ディスパッチャ、フィルタなどのカスタム拡張機能をアップロードするには、その前に拡張ファイルを拡張アーカイブにパッケージ化する必要があります。拡張アーカイブには、すべての拡張機能のバイナリ、起動データおよび構成情報が含まれます。各拡張アーカイブには1つの拡張機能の実装のみが含まれ、実行時にロードされます。拡張アーカイブには次のディレクトリが含まれます。

- [Meta-INF](#)
- [classes](#)
- [lib](#)

Meta-INF

このディレクトリには、アーカイブに関するメタ情報が含まれます。このディレクトリに拡張アーカイブ記述子ファイルを含める必要があります。拡張アーカイブ記述子ファイルは、拡張機能をロードおよび管理するために、Oracle Sensor Edge Server に必要な情報を含む META-INF ディレクトリ内の XML ファイルです。

拡張アーカイブ記述子ファイルは `ext.xml` という名前です。Loop Back Filter というフィルタ拡張機能の拡張アーカイブ記述子ファイル (`ext.xml`) を例 3-1 に示します。

例 3-1 フィルタ拡張機能の拡張アーカイブ記述子ファイル

```
<?xml version="1.0"?>
<Extension>
<name>Loop Back Filter</name>
<version>1.0</version>
<className>oracle.edge.impl.filter.LoopBackFilter</className>
<type>Filter</type>
<Parameters>
  <Parameter name="TagID" defaultValue="" description="The Invalid Tag ID">
    <valueType type="string"/>
  </Parameter>
  <Parameter name="LightStackName" defaultValue="stack1" description="The Light
Stack Instance Name">
    <valueType type="string"/>
  </Parameter>
</Parameters>
```

`ext.xml` 用の DTD を例 3-2 に簡略化して示します。この DTD の要素は表 3-8 で説明します。

例 3-2 拡張アーカイブ記述子ファイルの DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Extension (name, version, className, type, Parameters)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT version (#PCDATA)>
<!ELEMENT className (#PCDATA)>
<!ELEMENT type (#PCDATA)>
<!ELEMENT Parameters (Parameter+)>
<!ELEMENT Parameter (valueType)>
<!ATTLIST Parameter
  name CDATA #REQUIRED
  displayName CDATA #IMPLIED
  defaultValue CDATA #IMPLIED
  description CDATA #IMPLIED
  encrypted (true|false) #IMPLIED
  isClearText (true|false) #IMPLIED
  required (true|false) #IMPLIED>
<!ELEMENT valueType EMPTY>
<!ATTLIST valueType
  type (int | string | double | boolean) #REQUIRED>
```

表 3-8 拡張アーカイブ記述子ファイルの要素と DTD

要素	属性またはテキスト	説明
Extension		拡張機能のプロパティを定義。
name	#text	拡張機能の名前。
type	#text	ドライバ、フィルタまたはディスパッチャなどの拡張機能のタイプ。大 / 小文字の区別はないが、テキストに余分なスペースや特殊文字を含めないようにする。予約された値は、Device、Filter、Dispatcher。
version	#text	拡張機能のバージョン番号を示すテキスト。
className	#text	ドライバをロードし、インスタンス化するクラスの名前。標準の拡張機能インタフェースの 1 つを実装するエン트리・クラス。完全修飾されたクラス名にするためにパッケージ名を含める必要がある。
Parameters	(<Parameter> 要素の親。)	拡張機能がアップロードされた後にユーザーが編集できるパラメータ。
Parameter	次の属性がある。 <ul style="list-style-type: none"> ■ name ■ displayName ■ defaultValue ■ encrypted ■ isClearText ■ required 	<ul style="list-style-type: none"> ■ name: パラメータの名前。 ■ displayName: パラメータの表示名。 ■ defaultValue: パラメータのデフォルト値。 ■ encrypted: 値がクリアテキスト形式で格納されないようにパラメータ値を暗号化するかどうかを指定。 ■ isClearText: デフォルト値（およびパラメータ・インスタンスの値）をクリアテキスト形式にリセットできるようにする。 encrypted パラメータが true に設定されている場合、クリアテキスト形式が読み取られ、次回 Oracle Sensor Edge Server を起動すると暗号化形式に設定される。 ■ required: パラメータ値が必須かどうかを指定。
valueType	type	パラメータのタイプ（次のいずれかの値を指定できる） <ul style="list-style-type: none"> ■ int: パラメータが 32 ビット符号付き整数の場合。 ■ string: 可変長文字列の場合。 ■ double: 倍精度の数値の場合。 ■ boolean: ブール値（true または false）の場合。

classes

このディレクトリには、すべてのクラス・ファイル、ネイティブ・バイナリ、関連ファイルまたは静的データが含まれます。JAR ファイルにパッケージ化されたクラス・ファイルは、このディレクトリの最上位に展開する必要があります。このリリースでは、JAR ライブラリのローディングはサポートされません。

lib

拡張アーカイブ・ファイルには、lib ディレクトリも含まれます。このディレクトリではサード・パーティのライブラリを指定します。Alien デバイス・ドライバの拡張アーカイブ・ファイルを例 3-3 に示します。lib ディレクトリに、Alien デバイス・ドライバに固有のライブラリ Gateway.jar が含まれています。

例 3-3 Alien デバイス・ドライバの拡張アーカイブ・ファイル

```
meta-inf/ext.xml
meta-inf/Manifest.mf
classes/oracle/edge/impl/driver/AlienReader.class
lib/Gateway.jar
```

拡張アーカイブ・ファイルのパッケージ化

拡張アーカイブ・ファイルのパッケージ化するには、次のようにします。

1. サンドボックス・ディレクトリを作成します。このディレクトリを JAR ソース・ディレクトリとして使用します。
2. このディレクトリの最上位に、META-INF および classes ディレクトリを作成します。
3. すべてのクラス・ファイルとプロパティ・ファイル（ある場合）を classes ディレクトリにコピーします。META-INF ディレクトリに、ext.xml（拡張アーカイブ記述子ファイル）を作成します。
4. ファイルをアーカイブします。JDK に含まれている JAR ツール、またはその他の標準的な圧縮ユーティリティを使用できます。サンドボックスの最上位ディレクトリから JAR ツールを実行します。たとえば、`jar cvmf test.jar` を実行すると、サンドボックス・ディレクトリのファイルが `test.jar` にアーカイブされます。これで、`test.jar` を Oracle Sensor Edge Server にアップロードできます。META-INF および classes ディレクトリをサンドボックス・ディレクトリの一部としてアーカイブしないでください。たとえば、コマンド `c:/work> jar tvf test.jar` を実行すると、正しくアーカイブされた `test.jar` 内のファイルが次のように表示されます。

```
0 Thu Apr 08 14:36:56 PDT 2004 META-INF/
71 Thu Apr 08 14:36:56 PDT 2004 META-INF/ext.xml
0 Thu Apr 08 13:42:52 PDT 2004 classes/
0 Thu Apr 08 13:42:52 PDT 2004 classes/my/
0 Thu Apr 08 13:42:58 PDT 2004 classes/my/test.class
```

注意： META-INF および classes ディレクトリの前には、スラッシュやその他のディレクトリ識別子は表示されません。JAR にパス全体を含めると、Oracle Sensor Edge Server は拡張アーカイブ記述子ファイルやクラスを見つけれません。そのため、拡張機能をデプロイできなくなります。

拡張機能のアップロード

拡張機能をアップロードするには、次のようにします。

1. 「[Oracle Sensor Edge Server インスタンスの拡張機能の追加](#)」の説明に従って、ドライバ、フィルタまたはディスパッチャを拡張アーカイブ・ファイルにパッケージ化します。
2. この JAR ファイルを次の場所にコピーします。
ORACLE_HOME/j2ee/applications/edge/edge/extensions
3. OC4J インスタンスを再起動して、Oracle Sensor Edge Server を再起動します。拡張機能は使用可能なドライバ、フィルタまたはディスパッチャとしてツリーに表示され、Oracle Sensor Edge Server のデバイス、フィルタ・インスタンスまたは現行ディスパッチャとして構成できます。

ヒント： <IsExtensionMonitorEnabled> 要素が `edgeserver.xml` で `true` に設定されている場合は、JAR ファイルを ORACLE_HOME/edge/extensions にコピーするだけで済みます。稼働中の Oracle Sensor Edge Server が拡張機能を自動的に取得するため、停止と再起動の必要はありません。この方法で拡張機能を追加するとパフォーマンスが低下するため、開発インスタンス以外では使用しないことをお勧めします。

Oracle Application Server 10g リリース 3 (10.1.3) では、`edgeserver.xml` は使用されなくなり、下位互換性の目的のみで保持されています。

拡張クラスの階層

Oracle Sensor Edge Server のすべての拡張機能は次のように編成されています。

EdgeObject: 基本のルート・クラス（一意の識別子を含む）。

AbstractEdgeExtensionImpl: EdgeExtension インタフェースを実装します。

- AbstractDispatcherV2: すべてのディスパッチャのベース・クラス。
 - AbstractDispatcher: 基本ディスパッチャ動作を提供する AbstractDeviceV2 クラスを実装します。
 - * SimpleDispatcher: カスタム・ディスパッチャの構築が可能になります。

AbstractFilter: すべてのフィルタのベース・クラス。Filter インタフェースを実装し、フィルタのパフォーマンスを監視するイベントの生成に必要なフィルタの監視 API を定義します。

- SimpleFilter: デバイス・レベルとデバイス・グループ・レベルの両方のフィルタの記述が可能になります。

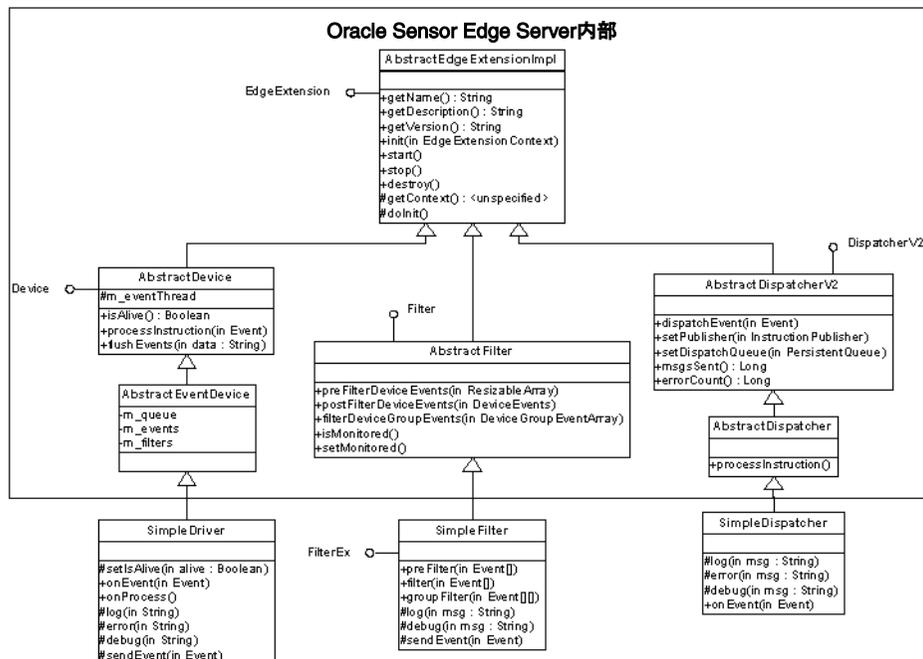
AbstractDevice: すべてのデバイスのベース・クラス。デバイスからデータを読み取り、読み取ったデータを処理するための独自のスレッドを提供します。

- AbstractEventDevice: AbstractDevice クラスの拡張クラス。デバイス・レベルのフィルタとの統合を提供し、イベントをイベント・プロセッサに伝播するために必要なメソッドを実装します。
 - SimpleDriver: ほとんどのカスタム・ドライバで必要とされる共通機能を提供します。Oracle Sensor Edge Server に付属のドライバの多くは、SimpleDriver から拡張されたものです。

注意: 拡張機能を作成するには、SimpleDriver、SimpleFilter または SimpleDispatcher クラスから拡張します。

拡張クラス階層を図 3-25 に示します。

図 3-25 エッジ拡張機能の階層



拡張機能の実装

`doInit()` メソッドが拡張機能を実装します。このコールにより実行時に拡張機能のインスタンスが初期化されます。

拡張機能のコンテキスト

実行時に拡張機能のインスタンスが作成されると、対応する Context が作成され、拡張機能で次の作業ができるようになります。

- ランタイム Context データを設定（または取得）します。
- Oracle Sensor Edge Server のその他の拡張機能を見つけ、通信します。
- Oracle Sensor Edge Server のシステム機能にアクセスします。
- インスタンスそのものに関する情報を取得します。

インスタンスに関する情報の取得 ベース・クラス `EdgeExtension` は、インスタンスがそれ自体に関する情報を取得するためのユーティリティ関数を提供します。次のメソッドがあります。

- `getContext()`
ランタイム・コンテキストを返します。
- `getName()`
拡張機能の名前を返します。
- `getDescription()`
拡張機能の説明を返します。
- `getVersion()`
拡張機能のバージョン文字列を返します。

インスタンスのランタイム・コンテキストへのアクセス インスタンスの Context オブジェクトを取得するには、次の文を使用します。

```
EdgeExtensionContext context = super.getContext();
```

メソッド・コール `getContext()` は、現行インスタンスの Context オブジェクトを返します。

インスタンスのパラメータの管理

拡張機能のインスタンスは、永続的なデータや構成を保持しません。インスタンスが初期化されると、実行時に構成データが渡されます。構成データはパラメータとして定義され、名前と値のペアで構成されます。各パラメータには一意の名前とオプションの値があります。

注意： このリリースの Oracle Sensor Edge Server では、値ツリーまたは値配列は直接サポートされません。スカラー型以外のデータを構成する場合は、データのアンマーシャリングをユーザー側で行ってください。

カスタム・パラメータの公開

拡張機能が特定の構成を持つことはよくあります。たとえば、ドライバにシリアル・ポート名、ボー・レート、IP アドレス、ポート番号、ログインおよびパスワードなどの構成パラメータが含まれることがあります。このようなパラメータは、ドライバがデバイスと通信できるように定義する必要があります。

ドライバ実装用のパラメータを公開するには、拡張アーカイブ記述子ファイルを変更する必要があります。シリアル・ポート名およびボー・レートの構成可能な 2 つのパラメータを `<Parameter>` 抽出タグに定義したデバイスを [例 3-4](#) に示します。

例 3-4 公開されたパラメータを持つ拡張アーカイブ記述子ファイル

```
<Extension>
  <name>My Driver</name>
  <type>Device</type>
  <className>my.testdriver</className>
  <Parameters>
    <Parameter name="port" displayName="Serial Port">
      <valueType type="string"/>
    </Parameter>
    <Parameter name="baud" displayName="Baud Rate">
      <valueType type="int"/>
    </Parameter>
  </Parameters>
</Extension>
```

パラメータ値の取得

拡張アーカイブ記述子ファイルの `<Parameter>` タグを定義すると、`EdgeExtensionConfigInfo` オブジェクトを使用してパラメータの値をフェッチできます。`<Parameter>` タグで定義された値は、`Context` オブジェクトを使用して取得されます ([例 3-5](#) を参照)。

例 3-5 コンテキスト・オブジェクトを使用したパラメータ値の取得

```
EdgeExtensionContext context = super.getContext();
ConfigParameter filenameParam = ct.getParameter( fileName );
```

`getParameter()` メソッドは、`ConfigParameter` オブジェクトを返します。`getParameter()` メソッドは、パラメータの値を返します。(例 3-5 では、`ConfigParameter` オブジェクトは `filenameParam` と呼ばれ、`getParameter()` メソッドが `fileName` というパラメータの値を返します。) ターゲット・パラメータの名前は、`ConfigParameter` オブジェクトに渡される必要があります。さらに、このパラメータの名前は、拡張アーカイブ記述子ファイルの `<Parameter>` 要素の `name` 属性に指定された名前に一致している必要があります。`ConfigParameter` オブジェクトを取得すると、パラメータの値を取得できます ([例 3-6](#) を参照)。

例 3-6 パラメータの値の取得

```
m_filename = filenameParam.getStringValue();
```

注意: getStringValue() メソッドは、パラメータの string 値を返します。パラメータの値が int の場合は、Integer オブジェクトを返す getIntegerValue() メソッドをコールします。

センサー・データ・リポジトリの使用

この章では、センサー・データ・リポジトリについて説明します。この章の内容は次のとおりです。

- 「センサー・データ・リポジトリの概要」
- 「スキーマ・リファレンス」

センサー・データ・リポジトリの概要

センサー・データ・リポジトリは、センサー・イベント・データを格納および問合せするためのデータベース表、ビューおよび PL/SQL パッケージの集合です。「[Oracle Sensor Edge Server およびセンサー・データ・リポジトリの考慮事項](#)」も参照してください。

リレーショナル表

センサー・データ・リポジトリのリレーショナル表には、実際のイベントの測定値とメタデータが格納されます。リレーショナル・ビューはこれらの表に基づいています。アプリケーションは、これらのリレーショナル・ビューおよびプログラミング・インタフェースを使用します。センサー・データ・リポジトリ内のリレーショナル表について表 4-1 に説明します。

表 4-1 センサー・データ・リポジトリ内のリレーショナル表

表名	目的
EDG_EVENT_TAB 表	ミドルウェアおよびアプリケーションからのイベントを格納する表
EDG_TAG_TAB 表	これまでに測定されたタグのキャッシュ・コピー
EDG_CAP_TAB 表	デバイスが送受信できるイベントの種類を定義するマッピング表
EDG_CTXT_TAB 表	コンテキストと包含のリレーションシップを定義するためのコンテキスト / 包含表
EDG_CTXT_REL_TAB 表	コンテキスト間のリレーションシップが定義された表
EDG_EVENT_INFO_TAB 表	イベント・タイプやサブタイプなど、イベントに関連する情報を格納するための表
EDG_DEVICE_TAB 表	デバイス表
EDG_DIAG_TAB 表	診断情報を格納する表
EDG_LOG 表	警告および内部エラーのログ表

センサー・データ・リポジトリ内のリレーショナル・ビュー

センサー・データ・リポジトリのビュー（表 4-2 を参照）は、格納されているデータの間合せに使用します。

表 4-2 センサー・データ・リポジトリ内のリレーショナル・ビュー

ビュー名	説明
EDG_CAP	デバイスの機能のビュー
EDG_CTXT	コンテキストの読取り専用ビュー（ビューは PL/SQL を使用して変更可能）
EDG_CTXT_REL	コンテキストのリレーションシップの読取り専用ビュー
EDG_CTXT_REL_NAME_VW	コンテキストとコンテキスト名のリレーションシップのビュー
EDG_DEVICE	デバイス表の読取り専用ビュー
EDG_DEV_CAP_VW	デバイス機能のビュー
EDG_DEV_DIAG_VW	デバイスの診断情報のビュー
EDG_DEV_EVENT_VW	デバイスにより取得されたイベントが表示されるビュー
EDG_DEV_LAST_DIAG_VW	最後の診断情報が表示されるビュー
EDG_DEV_LAST_OBSV_VW	デバイスにより実行された最新の測定のビュー
EDG_DIAG	診断表の読取り専用ビュー

表 4-2 センサー・データ・リポジトリ内のリレーショナル・ビュー（続き）

ビュー名	説明
EDG_EVENT	イベント表の読取り専用ビュー
EDG_EVENT_INFO	現在有効なイベント・メタデータのビュー
EDG_EVENT_VW	タイプおよびサブタイプが指定されたイベントのビュー
EDG_TAG	タグ画面の読取り専用ビュー
EDG_TAG_LAST_DEV_VW	タグを検出した最後のデバイスのビュー
EDG_TAG_PATH_VW	タグを検出したデバイスを基準にしたタグのパスのビュー

センサー・データ・リポジトリの PL/SQL パッケージ

センサー・データ・リポジトリにより定義された PL/SQL パッケージを使用すると、コンテキスト包含リレーションシップを操作したり、手動でイベントをキューに挿入できます。PL/SQL パッケージ内に指定されているプロシージャを表 4-3 で説明します。「PL/SQL プログラミング・インタフェース」も参照してください。

表 4-3 パッケージ内に指定されているプロシージャ

プロシージャ	説明
on_event	着信イベントを処理し、パーツを逆アセンブルして様々な表に挿入する方法を選択するプロシージャ
create_ctxt	新しいコンテキストを追加するプロシージャ
update_ctxt_rel	コンテキスト階層を更新するプロシージャ
update_ctxt	コンテキストを更新するプロシージャ
remove_ctxt	コンテキストを削除するプロシージャ

リポジトリに対する操作と問合せ

この項では、センサー・データ・リポジトリに対する次の操作と問合せについて説明します。

- リポジトリの作成と削除
- リポジトリへの測定値の保存
- アーカイブの問合せ

リポジトリの作成と削除

センサー・データ・リポジトリの作成と削除は、センサー・データ・リポジトリ・スキーマの作成と削除の方法と基本的に同じです。インストール・プロセスの一部としてリポジトリをインストールするように選択した場合、リポジトリは自動的に作成されます。それ以外の場合は、手動で SQL スクリプトを起動してリポジトリをインストールできます。サーバー側には、isArchived という名前のフラグがあります。サーバーがデータのアーカイブを開始するためには、このフラグがオンになっている必要があります。

リポジトリへの測定値の保存

Oracle Sensor Edge Server をアーカイブ・モードに設定した場合、イベントは自動的にセンサー・データ・リポジトリに送信され、Oracle Sensor Edge Server インスタンスに送信されるすべてのイベントはアーカイブされます。アプリケーションには、edg_sda パッケージ内の PL/SQL プロシージャを使用してイベントをリポジトリに手動でエンキューするというオプションがあります（「PL/SQL プログラミング・インタフェース」を参照）。

アーカイブの問合せ

表 4-1 および表 4-2 に説明されたビューとリレーショナル表を使用して、センサー・データ・リポジトリを問い合わせます。「スキーマ・リファレンス」も参照してください。

スキーマ・リファレンス

この項では、センサー・データ・リポジトリの表、ビューおよび PL/SQL プログラミング・インタフェースを一覧表示します。

表

センサー・データ・リポジトリには次の表が含まれます。

- [EDG_CAP_TAB](#) 表
- [EDG_CTXT_REL_TAB](#) 表
- [EDG_CTXT_TAB](#) 表
- [EDG_DEVICE_TAB](#) 表
- [EDG_DIAG_TAB](#) 表
- [EDG_EVENT_INFO_TAB](#) 表
- [EDG_EVENT_TAB](#) 表
- [EDG_LOG](#) 表
- [EDG_TAG_TAB](#) 表

EDG_CAP_TAB 表

デバイスが送受信できるイベントのタイプを定義するマッピング表です。

表 4-4 EDG_CAP_TAB 表

名前	データ型	NULL	デフォルト値	説明
OBJECT_ID	NUMBER(10)	不可	na	このマッピング・エントリの主キー
REF_DEVICE	NUMBER(10)	可	na	関連するデバイスへの参照
REF_EVENT_INFO	NUMBER(10)	可	na	関連するイベント情報への参照
SEND_OR_RECV	VARCHAR2(16)	可	'SEND'\$	デバイスがイベントを送受信できるかどうかを示すフラグ
CREATED_BY	VARCHAR2(256)	可	USER\$	このエントリを作成したユーザー
CREATED_TIME	TIMESTAMP(6)	可	SYSTIMESTAMP\$	エントリが作成された時刻

EDG_CTXT_REL_TAB 表

コンテキスト間のリレーションシップが定義された表です。

表 4-5 EDG_CTXT_REL_TAB 表

名前	データ型	NULL	デフォルト値	説明
OBJECT_ID	NUMBER(10)	不可	na	コンテキスト間のリレーションシップ・エントリの主キー
CID	NUMBER(10)	可	na	子コンテキストの ID
PID	NUMBER(10)	可	na	親コンテキストの ID
CREATED_BY	VARCHAR2(256)	可	USER	このエントリを作成したユーザー
CREATED_TIME	TIMESTAMP(6)	可	TIMESTAMP	エントリが作成された時刻
RETIRED_BY	VARCHAR2(256)	可	na	このエントリを無効にしたユーザー
RETIRED_TIME	TIMESTAMP(6)	可	na	エントリが無効化された時刻
IS_CURRENT	VARCHAR2(1)	可	na	エントリが現在のエントリかどうかを示すインジケータ
REF_NEXT_VER	NUMBER(10)	可	na	次のバージョンのリレーションシップへの参照 (コンテキストの包含履歴の再構成に役立つように追加される)

EDG_CTXT_TAB 表

コンテキストと包含のリレーションシップを定義するためのコンテキスト / 包含表です。

表 4-6 EDG_CTXT_TAB 表

名前	データ型	NULL	デフォルト値	説明
OBJECT_ID	NUMBER(10)	不可	na	コンテキスト・エントリの主キー
NAME	VARCHAR2(1024)	可	na	コンテキストの名前
DESCRIPTION	NUMBER(10)	可	na	コンテキストの説明
IS_DEFAULT	VARCHAR2(256)	可	'F\$'	コンテキストは汎用フラグ
CREATED_BY	TIMESTAMP(6)	可	USERS\$	エントリを作成したユーザー
CREATED_TIME	VARCHAR2(256)	可	SYSTIMESTAMP\$	エントリが作成された時刻
RETIRED_BY	TIMESTAMP(6)	可	na	エントリを無効化したユーザー
RETIRED_TIME	VARCHAR2(1)	可	na	エントリが無効化された時刻
IS_CURRENT	NUMBER(10)	可	'T\$'	現行エントリ・フラグ

EDG_DEVICE_TAB 表

デバイス表です。

表 4-7 EDG_DEVICE_TAB 表

番号	名前	データ型	NULL	デフォルト値
1: デバイスの主キー	OBJECT_ID	NUMBER(10)	不可	na
2: このデバイスを表すタグ ID (EPC コード)	TAG_ID	VARCHAR2(256)	可	na
3: このデバイス・オブジェクトの名前	NAME	VARCHAR2(256)	不可	na
4: このデバイスの説明	DESCRIPTION	VARCHAR2(1024)	可	na
5: このデバイスが存在するサイト名	SITE_NAME	VARCHAR2(256)	不可	na
6: このデバイスを作成したユーザー	CREATED_BY	VARCHAR2(256)	可	USER\$
7: このデバイス・エントリが作成された時刻	CREATED_TIME	TIMESTAMP(6)	可	SYSTIMESTAMP\$
8: デバイスの最後の診断ステータスを調べるための参照	LAST_STATUS	NUMBER(10)	可	na

EDG_DIAG_TAB 表

診断情報を格納するために使用される表です。

表 4-8 EDG_DIAG_TAB 表

番号	名前	データ型	NULL	デフォルト値
1: 診断エントリの主キー	OBJECT_ID	NUMBER(10)	不可	na
2: この診断エントリのデバイスへの参照	REF_DEVICE	NUMBER(10)	可	na
3: デバイス、サーバーまたはコンポーネントのステータス	STATUS	VARCHAR2(64)	不可	na
4: エラーまたは警告に関連付けられたメッセージ	MESSAGE	VARCHAR2(1024)	可	na
5: エラーのあるイベントが発生した時刻	TIME	TIMESTAMP(6)	可	na
6: この診断エントリを作成したユーザー	CREATED_BY	VARCHAR2(256)	可	USER\$
7: この診断エントリが作成された時刻	CREATED_TIME	TIMESTAMP(6)	可	SYSTIMESTAMP\$

EDG_EVENT_INFO_TAB 表

(タイプやサブタイプなどの) イベント情報を格納するために使用される表です。

表 4-9 EDG_EVENT_INFO_TAB 表

番号	名前	データ型	NULL	デフォルト値
1: イベント・エントリの主キー	OBJECT_ID	NUMBER(10)	不可	na
2: イベント・エントリの名前	NAME	VARCHAR2(256)	不可	na
3: このイベント情報エントリのタイプ	TYPE	NUMBER(5)	不可	na
4: このイベント情報エントリのサブタイプ	SUBTYPE	NUMBER(5)	可	-1\$
5: イベントがカスタム定義イベントか、それとも Oracle に登録されている Oracle 提供イベントかを示すフラグ	REGISTERED	VARCHAR2(1)	可	'F'\$
6: イベントの説明	DESCRIPTION	VARCHAR2(1024)	可	na
7: イベント内の ID フィールドの使用法パターン (ドライバと一致している必要あり)	ID_USAGE	VARCHAR2(1024)	可	na
8: イベント内のデータ・フィールドの使用法パターン (ドライバと一致している必要あり)	DATA_USAGE	VARCHAR2(1024)	可	na
このエントリを作成したユーザー	CREATED_BY	VARCHAR2(256)	可	USER\$
エントリが作成された時刻	CREATED_TIME	TIMESTAMP(6)	可	SYSTIMESTAMP\$
このエントリを無効化したユーザー	RETIRED_BY	VARCHAR2(256)	可	na
このエントリが無効化された時刻	RETIRED_TIME	TIMESTAMP(6)	可	na
エントリが有効かどうか	IS_CURRENT	VARCHAR2(1)	可	'T'\$

EDG_EVENT_TAB 表

ミドルウェアおよびアプリケーションからのイベント情報を格納するために使用される表です。

表 4-10 EDG_EVENT_TAB 表

番号	名前	データ型	NULL	デフォルト値
1: イベントの主キー	OBJECT_ID	NUMBER(10)	不可	na
2: タグのタグ ID (EPC コード)	TAG_ID	VARCHAR2(256)	可	na
3: イベントが発信されたデ バイス	REF_DEVICE	NUMBER(10)	可	na
4: イベントに関連するコン テキスト。コンテキストに 応じてアプリケーションに より設定される	REF_CTXT	NUMBER(10)	可	-\$
5: イベントのソース名	SOURCE_NAME	VARCHAR2(64)	可	'F'\$
6: イベント・メタデータへ の参照	REF_EVENT_ INFO	NUMBER(10)	可	na
7: 集約イベントの関係付け に使用される参照	CORRELATION_ ID	VARCHAR2(64)	可	na
8: イベントのデータ・ フィールド。デバイスのタ イプにより異なる。詳細は 「EDG_EVENTINFO」の 「ID_USAGE フィールド」 を参照	DATA	VARCHAR2(1024)	可	na
イベントが発生した時刻	TIME	TIMESTAMP(6)	可	na

EDG_LOG 表

警告および内部エラーを格納するために使用される表です。

表 4-11 EDG_LOG 表

番号	名前	データ型	NULL	デフォルト値
1: ログ・エントリの主 キー。	OBJECT_ID	NUMBER(10)	不可	na
2: ログ・レベル。使用可能 な値は、"N" (通知)、"W" (警告) および "E" (エ ラー)。	LOG_LEVEL	VARCHAR2(1)	可	na
3: エラーおよび警告用に記 録されるメッセージ。	MESSAGE	VARCHAR2(4000)	可	na
4: ログ・エントリを作成し たデータベース・ユー ザー。	CREATED_BY	VARCHAR2(64)	可	USER\$
5: データベース内にエント リが作成された時刻。	CREATED_TIME	TIMESTAMP(6)	可	SYSTIMESTAMP\$

EDG_TAG_TAB 表

これまでに測定されたタグのキャッシュ・コピーです。

表 4-12 EDG_TAG_TAB 表

番号	名前	データ型	NULL	デフォルト値
1: タグ・エントリの主キー	OBJECT_ID	NUMBER(10)	不可	na
2: タグ ID (EPC コード)	TAG_ID	VARCHAR2(256)	可	na
3: タグの最後の測定への参照	LAST_EVENT	NUMBER(10)	可	na
4: エントリを作成したユーザー	CREATED_BY	VARCHAR2(256)	可	USER\$
5: エントリが作成された時刻	CREATED_TIME	TIMESTAMP(6)	可	SYSTIMESTAMP\$

ビュー

センサー・データ・リポジトリには次のビューが含まれます。

- [EDG_CAP](#)
- [EDG_CTXT](#)
- [EDG_CTXT_REL](#)
- [EDG_CTXT_REL_NAME_VW](#)
- [EDG_DEVICE](#)
- [EDG_DEV_CAP_VW](#)
- [EDG_DEV_DIAG_VW](#)
- [EDG_DEV_EVENT_VW](#)
- [EDG_DEV_LAST_DIAG_VW](#)
- [EDG_DEV_LAST_OBSV_VW](#)
- [EDG_DIAG](#)
- [EDG_EVENT](#)
- [EDG_EVENT_INFO](#)
- [EDG_EVENT_VW](#)
- [EDG_TAG](#)
- [EDG_TAG_LAST_DEV_VW](#)
- [EDG_TAG_PATH_VW](#)

EDG_CAP

デバイスの機能のビューです。

表 4-13 EDG_CAP ビュー

ビュー	データ型	NULL
1 OBJECT_ID: デバイスとイベントの機能マッピング用の主キー	NUMBER(10)	不可
2 REF_DEVICE: 関連するデバイスへの参照	NUMBER(10)	可
3 REF_EVENT_INFO: 関連するイベント情報への参照	NUMBER(10)	可
4 SEND_OR_RECV: デバイスがイベントを送受信できるかどうかを示すフラグ	VARCHAR2(16)	可

```
SELECT object_id, ref_device, ref_event_info, send_or_recv
FROM edg_cap_tab
```

EDG_CTXT

コンテキストの読取り専用ビューです。ビューは、PL/SQL プロシージャを使用して変更できません。

表 4-14 EDG_CTXT ビュー

ビュー	データ型	NULL
1 OBJECT_ID: コンテキスト・エントリの主キー	NUMBER(10)	不可
2 NAME: コンテキストの名前	VARCHAR2(1024)	可
3 DESCRIPTION: コンテキストの説明	VARCHAR2(1024)	可
4 SEND_OR_RECV: コンテキストが汎用かどうか	VARCHAR2(16)	可

```
SELECT object_id, name, description, is_default
FROM edg_ctxt_tab
WHERE is_current = 'T'
WITH READ ONLY
```

EDG_CTXT_REL

コンテキストのリレーションシップの読取り専用ビューです。

表 4-15 EDG_CTXT_REL ビュー

ビュー	データ型	NULL
1 OBJECT_ID: コンテキスト間のリレーションシップ・エントリの主キー	NUMBER(10)	不可
2 CID: 子コンテキストの ID	NUMBER(10)	可
3 PID: 親コンテキストの ID	NUMBER(10)	可

```
SELECT object_id, cid, pid
FROM edg_ctxt_rel_tab
WHERE is_current = 'T'
WITH READ ONLY
```

EDG_CTXT_REL_NAME_VW

コンテキストのリレーションシップの読取り専用ビューです。

表 4-16 EDG_CTXT_REL_NAME_VW ビュー

ビュー	データ型	NULL
1 CHILD_ID: 子コンテキストの ID	NUMBER(10)	可
2 PARENT_ID: 親コンテキストの ID	NUMBER(10)	可
3 PARENT_NAME: 親コンテキストの名前	VARCHAR2(1024)	可
4 CHILD_NAME: 子コンテキストの名前	VARCHAR2(1024)	可

```
SELECT
rel.cid child_id,
rel.pid parent_id,
cP.name parent_name,
cC.name child_name
FROM edg_ctxt cP, edg_ctxt cC, edg_ctxt_rel rel
WHERE cP.object_id = rel.pid
AND cC.object_id = rel.cid
```

EDG_DEVICE

デバイス表の読取り専用ビューです。

表 4-17 EDG_DEVICE ビュー

番号	データ型	NULL
1 OBJECT_ID: デバイスの主キー	NUMBER(10)	不可
2 TAG_ID: デバイスを表すタグ ID (EPC コード)	VARCHAR2(256)	可
3 NAME: デバイス・オブジェクトの名前	VARCHAR2(256)	不可
4 DESCRIPTION: デバイスの説明	VARCHAR2(1024)	可
5 SITE_NAME: デバイスが存在するサイトの名前	VARCHAR2(256)	不可
6 CREATED_BY: デバイス・エントリを作成したユーザー	VARCHAR2(256)	可
7 CREATED_TIME: デバイス・エントリが作成された時刻	TIMESTAMP(6)	可
8 LAST_STATUS: デバイスの最後の診断ステータスを調べるための参照	NUMBER(10)	可

```

SELECT
"OBJECT_ID", "TAG_ID", "NAME", "DESCRIPTION", "SITE_NAME", "CREATED_BY", "CREATED
_TIME", "LAST_STATUS"
FROM edg_device_tab
WITH READ ONLY

```

EDG_DEV_CAP_VW

デバイス機能のビューです。

表 4-18 EDG_DEV_CAP_VW ビュー

番号	データ型	NULL
1 DEVICE_TAG_ID: デバイスを表すタグ ID (EPC コード)	VARCHAR2(256)	可
2 DEVICE_NAME: デバイス・オブジェクトの名前	VARCHAR2(256)	不可
3 DEVICE_DESC: デバイスの説明	VARCHAR2(64)	可
4 SITE_NAME: デバイスが存在するサイトの名前	VARCHAR2(1024)	不可
5 EVENT_TYPE: イベント情報エントリのタイプ	VARCHAR2(256)	不可
6 EVENT_SUBTYPE: イベント情報のサブタイプ	NUMBER(5)	可
7 EVENT_DESC: このイベント・タイプの説明	NUMBER(5)	可
8 SEND_OR_RECV: デバイスが送受信できるかどうかを示すフラグ	VARCHAR2(1024)	可

```

SELECT
dev.tag_id device_tag_id,
dev.name device_name,
dev.description device_desc,
dev.site_name site_name,
ei.type event_type,
ei.subtype event_subtype,ei.description event_desc,
cap.send_or_recv send_or_recv
FROM edg_cap cap, edg_device dev, edg_event_info ei
WHERE cap.ref_device=dev.object_id
AND cap.ref_event_info=ei.object_id

```

EDG_DEV_DIAG_VW

デバイスの診断情報が表示されるビューです。

表 4-19 EDG_DEV_DIAG_VW ビュー

ビュー	データ型	NULL
1 DEVICE_TAG_ID: デバイスを表すタグ ID (EPC コード)	VARCHAR2(256)	可
2 STATUS: デバイス、サーバーまたはコンポーネントのステータス	VARCHAR2(64)	可
3 MESSAGE: エラーまたは警告に関連付けられたメッセージ	VARCHAR2(1024)	可
4 TIME: エラー条件が発生した時刻	TIMESTAMP(6)	可
5 DEVICE_NAME: デバイス・オブジェクトの名前	VARCHAR2(256)	不可
6 DEVICE_DESC: デバイスの説明	VARCHAR2(1024)	可
7 SITE_NAME: デバイスが存在するサイトの名前	VARCHAR2(256)	不可

```

SELECT
dev.tag_id device_tag_id,
diag.status status,
diag.message message,
diag.time time,
dev.name
device_name,
dev.description device_desc,
dev.site_name site_name
FROM edg_diag diag, edg_device dev
WHERE dev.object_id=diag.ref_device

```

EDG_DEV_EVENT_VW

デバイスにより取得されたイベントが表示されるビューです。

表 4-20 EDG_DEV_EVENT_VW ビュー

ビュー	データ型	NULL
1 DEVICE_NAME: デバイス・オブジェクトの名前	VARCHAR2(256)	不可
2 DEVICE_TAG_ID: デバイスを表すタグ ID (EPC コード)	VARCHAR2(256)	可
3 DEVICE_DESC: デバイスの説明	VARCHAR2(1024)	可
4 SITE_NAME: デバイスが存在するサイトの名前	VARCHAR2(256)	不可
5 EVENT_ID: イベント・オブジェクトへの参照	NUMBER(10)	不可
6 EVENT_TAG_ID: このイベントのタグおよびデバイスを表すタグ ID (EPC コード)	VARCHAR2(256)	可
7 DEVICE_ID: デバイス・オブジェクトへの参照	NUMBER(10)	可

表 4-20 EDG_DEV_EVENT_VW ビュー (続き)

ビュー	データ型	NULL
8.8 EVENT_DATA: イベントのデータ・フィールド。デバイスのタイプによって異なる。詳細は「edge_event_info」の「id_usage フィールド」を参照	VARCHAR2(1024)	可
9 EVENT_TIME: イベントが発生した時刻	TIMESTAMP(6)	可
10 EVENT_CORRELATION_ID: 集約イベントの関係付けに使用される参照	VARCHAR2(64)	可
11 EVENT_NAME: イベントの種類の名前	VARCHAR2(256)	不可
12 EVENT_TYPE: イベント情報エントリのタイプ。たとえば、RFID 測定の場合は 200	NUMBER(5)	不可
13 EVENT_SUBTYPE: イベント情報エントリのサブタイプ。たとえば、IN_FIELD RFID 測定の場合は 1	NUMBER(5)	可
14 EVENT_DESC: イベントの種類の説明	VARCHAR2(1024)	可

```

SELECT
dev.name device_name,
dev.tag_id device_tag_id,
dev.description device_desc,
dev.site_name site_name,
diag.status status,
ev.*
FROM edg_device dev, edg_event_vw ev
WHERE ev.device_id=dev_object_id

```

EDG_DEV_LAST_DIAG_VW

最後の診断情報が表示されるビューです。

表 4-21 EDG_DEV_LAST_DIAG_VW ビュー

ビュー	データ型	NULL
1 DEVICE_TAG_ID: デバイスを表すタグ ID (EPC コード)	VARCHAR2(256)	可
2 STATUS: デバイス、サーバーまたはコンポーネントのステータス	VARCHAR2(64)	可
3 MESSAGE: エラーまたは警告に関連付けられたメッセージ	VARCHAR2(1024)	可
4 TIME: エラー条件が発生した時刻	TIMESTAMP(6)	可
5 DEVICE_NAME: デバイス・オブジェクトの名前	VARCHAR2(256)	不可
6 DEVICE_DESC: デバイスの説明	VARCHAR2(1024)	可
7 SITE_NAME: デバイスが存在するサイトの名前	VARCHAR2(256)	不可

```

SELECT
dev.tag_id device_tag_id,
diag.status status,
diag.message message,
diag.time time,
dev.name device_name,
dev.description device_desc,
dev.site_name site_name
FROM edg_diag diag, edg_device dev
WHERE dev.last_status = diag.object_id

```

EDG_DEV_LAST_OBSV_VW

デバイスにより実行された最後の測定が表示されるビューです。

表 4-22 EDG_DEV_LAST_OBSV_VW ビュー

ビュー	データ型	NULL
1 DEVICE_NAME: デバイス・オブジェクトの名前	VARCHAR2(256)	不可
2 DEVICE_TAG_ID: デバイスを表すタグ ID (EPC コード)	VARCHAR2(256)	可
3 DEVICE_DESC: デバイスの説明	VARCHAR2(1024)	可
4 SITE_NAME: デバイスが存在するサイトの名前	VARCHAR2(256)	可
5 EVENT_ID: イベント・オブジェクトへの参照	NUMBER(10)	不可
6 DEVICE_ID: デバイスを表すタグ ID (EPC コード)	VARCHAR2(256)	可
7 EVENT_TAG_ID: デバイス・オブジェクトへの参照	NUMBER(10)	可
8 EVENT_DATA: イベントのデータ・フィールド。デバイスによって異なる。 「edg_event_info」の「id_usage フィールド」を参照	VARCHAR2(1024)	可
9 EVENT_TIME: イベントが発生した時刻	VARCHAR2(1024)	可
10 EVENT_CORRELATION_ID: 集約イベントの関係付けに使用される参照	VARCHAR2(64)	可
11 EVENT_NAME: イベントの種類の名前	VARCHAR2(256)	可
12 EVENT_TYPE: イベント情報エントリのタイプ。たとえば、RFID 測定の場合は 200	NUMBER(5)	可
13 EVENT_SUBTYPE: イベント情報エントリのサブタイプ。たとえば、IN_FIELD RFID 測定の場合は 1	NUMBER(5)	可
14 EVENT_DESC: イベントの種類の説明	VARCHAR2(1024)	可

```

SELECT
device_name,
device_tag_id,
device_desc,
site_name,
event_id,
event_tag_id,
device_id,
event_data,
event_time,
event_correlation_id,
event_name,
event_type,
event_subtype,
event_desc
FROM (

```

EDG_DIAG

診断表の読取り専用ビューです。

表 4-23 EDG_DIAG ビュー

ビュー	データ型	NULL
1 OBJECT_ID: 診断エントリの主キー	NUMBER(10)	不可
2 REF_DEVICE: 関連付けられたデバイスへの参照。	NUMBER(10)	可
3 STATUS: デバイス、サーバーまたはコンポーネントのステータス	VARCHAR2(64)	可
4 MESSAGE: エラーまたは警告に関連付けられたメッセージ	VARCHAR2(1024)	可
5 TIME: エラー条件が発生した時刻	TIMESTAMP(6)	可
6 CREATED_BY: 診断エントリを作成したユーザー	VARCHAR2(256)	可
7 CREATED_TIME 診断エントリが作成された時刻	TIMESTAMP(6)	可

```

SELECT
"CONNECT_ID", "REF_DEVICE", "STATUS", "MESSAGE", "TIME", "CREATED_BY", "CREATED_TIME"
FROM edg_diag_tab
WITH READ ONLY

```

EDG_EVENT

イベント表の読取り専用ビューです。

表 4-24 EDG_EVENT ビュー

ビュー	データ型	NULL
1 OBJECT_ID: イベントの主キー	NUMBER(10)	不可
2 TAG_ID: イベントが作成されたタグおよびデバイスを表すタグ ID (EPC コード)	VARCHAR2(256)	可
3 REF_DEVICE: イベントが発信されたデバイス	NUMBER(10)	可
4 REF_CTXT: イベントに関連するコンテキスト。コンテキスト状況に応じてアプリケーションにより設定される	NUMBER(10)	可
5 SOURCE_NAME: イベントのソース名	VARCHAR2(64)	可
6 REF_EVENT_INFO: イベント・メタデータへの参照	NUMBER(10)	可
7 CORRELATION_ID: 集約イベントの関係付けに使用される参照	VARCHAR2(64)	可
8 DATA: イベントのデータ・フィールド。デバイスによって異なる。 詳細は「edg_event_info」の「id_usage」を参照	VARCHAR2(1024)	可
9 TIME: イベントが発生した時刻	TIMESTAMP(6)	可

```
SELECT
"OBJECT_ID", "TAG_ID", "REF_DEVICE", "REF_CTXT", "SOURCE_NAME", "REF_EVENT_INFO",
"CORRELATION_ID" "DATA", "TIME"
FROM edg_event_tab
WITH READ ONLY
```

EDG_EVENT_INFO

現在有効なイベント・メタデータのビューです。

表 4-25 EDG_EVENT_INFO ビュー

ビュー	データ型	NULL
1 OBJECT_ID: イベント情報エントリの主キー	NUMBER(10)	不可
2 NAME: イベント情報エントリの名前	VARCHAR2(256)	不可
3 TYPE: イベント情報エントリのタイプ	NUMBER(5)	不可
4 SUBTYPE: イベント情報エントリのサブタイプ	NUMBER(5)	不可
5 REGISTERED: イベント情報がカスタム定義イベント情報か、それとも Oracle に登録されている Oracle 提供イベント情報かを示すフ ラグ	VARCHAR2(1)	可
6 DESCRIPTION: イベントの種類の説明	VARCHAR2(1024)	可

表 4-25 EDG_EVENT_INFO ビュー (続き)

ビュー	データ型	NULL
7 ID_USAGE: (ミドルウェア側からの) イベント内の ID フィールドの使用 方法パターン。ドライバ実装と一致している必要がある	VARCHAR2(1024)	可
8 DATA_USAGE: (ミドルウェア側からの) イベント内のデータ・フィールドの 使用方法パターン。ドライバ実装と一致している必要がある	VARCHAR2(1024)	可

```
SELECT object_id, name, type, subtype, registered, description, id_usage, data_usage,
data_usage
FROM edg_event_tab
WHERE is_current='T'
```

EDG_EVENT_VW

タイプおよびサブタイプが指定されたイベントのビューです。

表 4-26 EDG_EVENT_VW ビュー

ビュー	データ型	NULL
1 EVENT_ID: イベントの主キー	NUMBER(10)	不可
2 EVENT_TAG_ID: イベントが作成されたタグおよびデバイスを表すタグ ID (EPC コード)	VARCHAR2(256)	可
3 DEVICE_ID: イベントが発信されたデバイス	NUMBER(10)	可
4 EVENT_DATA: イベントのデータ・フィールド。デバイスの種類によって異 なる。詳細は「edg_event_info」の「id_usage フィールド」 を参照	VARCHAR2(1024)	可
5 EVENT_TIME: イベントが発生した時刻	TIMESTAMP(6)	可
6 CORRELATION_ID: 集約イベントの関係付けに使用される参照	VARCHAR2(64)	可
7 EVENT_NAME: イベントの種類の名前	VARCHAR2(256)	不可
8 EVENT_TYPE: イベント情報エントリのタイプ。たとえば、RFID 測定の場合 は 200	NUMBER(5)	不可
9 EVENT_SUBTYPE: イベント情報エントリのサブタイプ。たとえば、IN_FIELD RFID 測定イベントの場合は 1	NUMBER(5)	可
10 EVENT_DESC: イベントの種類の説明	VARCHAR2(1024)	可

```
SELECT
ev.object_id event_id,
ev.tag_id event_tag_id,
ev.ref_device device_id,
ev.data event_data,
ev.time event_time,
ev.correlation_id event correlation_id,
md.name event_name,
md.type event_type,
```

```
md.subtype event_subtype,
md.description event_desc
FROM edg_event_tab ev, edg_event_info md
WHERE ev.ref_event_info=md.object_id
```

EDG_TAG

タグ画面の読取り専用ビューです。

表 4-27 EDG_TAG ビュー

ビュー	データ型	NULL
1 OBJECT_ID: タグ・エントリの主キー	NUMBER(10)	不可
2 TAG_ID: タグ ID (EPC コード)	VARCHAR2(256)	可
3 LAST_EVENT: タグの最後の測定への参照	NUMBER(10)	可

```
SELECT
object_id tag_id, last_event
FROM edg_tag_tab
WITH READ ONLY
```

EDG_TAG_LAST_DEV_VW

タグを検出した最後のデバイスのビューです。

表 4-28 EDG_TAG_LAST_DEV_VW ビュー

番号	データ型	NULL
1 DEVICE_NAME: デバイス・オブジェクトの名前	VARCHAR2(256)	不可
2 DEVICE_TAG_ID: デバイスを表すタグ ID (EPC コード)	VARCHAR2(256)	可
3 DEVICE_DESC: デバイスの説明	VARCHAR2(1024)	可
4 SITE_NAME: デバイスが存在するサイトの名前	VARCHAR2(256)	不可
5 EVENT_ID: イベント・オブジェクトへの参照	NUMBER(10)	不可
6 EVENT_TAG_ID: このイベントのタグおよびデバイスを表すタグ ID (EPC コード)	VARCHAR2(256)	可
7 DEVICE_ID: デバイス・オブジェクトへの参照	NUMBER(10)	可
8.8 EVENT_DATA: イベントのデータ・フィールド。デバイスのタイプによって異なる。詳細は「edge_event_info」の「id_usage フィールド」を参照	VARCHAR2(1024)	可
9 EVENT_TIME: イベントが発生した時刻	TIMESTAMP(6)	可
10 EVENT_CORRELATION_ID: 集約イベントの関係付けに使用される参照	VARCHAR2(64)	可
11 EVENT_NAME: イベントの種類の名前	VARCHAR2(256)	不可

表 4-28 EDG_TAG_LAST_DEV_VW ビュー (続き)

番号	データ型	NULL
12 EVENT_TYPE: イベント情報エントリのタイプ。たとえば、RFID 測定の場合は 200	NUMBER(5)	不可
13 EVENT_SUBTYPE: イベント情報エントリのサブタイプ。たとえば、IN_FIELD RFID 測定の場合は 1	NUMBER(5)	可
14 EVENT_DESC: イベントの種類の説明	VARCHAR2(1024)	可

```
SELECT
devEVENT.*
FROM edg_tag, edg_dev_event_vw devEVENT
WHERE tag.last_event=devEVENT.event_id
```

EDG_TAG_PATH_VW

タグを検出したデバイスを基準にしたタグのパスのビューです。

表 4-29 EDG_TAG_PATH_VW ビュー

ビュー	データ型	NULL
1 DEVICE_NAME: デバイス・オブジェクトの名前	VARCHAR2(256)	不可
2 DEVICE_TAG_ID: デバイスを表すタグ ID (EPC コード)	VARCHAR2(256)	可
3 DEVICE_DESC: デバイスの説明	VARCHAR2(1024)	可
4 SITE_NAME: デバイスが存在するサイトの名前	VARCHAR2(256)	不可
5 EVENT_ID: イベント・オブジェクトへの参照	NUMBER(10)	不可
6 EVENT_TAG_ID: このイベントのタグおよびデバイスを表すタグ ID (EPC コード)	VARCHAR2(256)	可
7 DEVICE_ID: デバイス・オブジェクトへの参照	NUMBER(10)	可
8. 8 EVENT_DATA: イベントのデータ・フィールド。デバイスのタイプによって異なる。詳細は「edge_event_info」の「id_usage フィールド」を参照	VARCHAR2(1024)	可
9 EVENT_TIME: イベントが発生した時刻	TIMESTAMP(6)	可
10 EVENT_CORRELATION_ID: 集約イベントの関係付けに使用される参照	VARCHAR2(64)	可
11 EVENT_NAME: イベントの種類の名前	VARCHAR2(256)	可
12 EVENT_TYPE: イベント情報エントリのタイプ。たとえば、RFID 測定の場合は 200	NUMBER(5)	可
13 EVENT_SUBTYPE: イベント情報エントリのサブタイプ。たとえば、IN_FIELD RFID 測定の場合は 1	NUMBER(5)	可

表 4-29 EDG_TAG_PATH_VW ビュー (続き)

ビュー	データ型	NULL
14 EVENT_DESC: イベントの種類の説明	VARCHAR2(1024)	可
15 NEXT_DEVICE_ID: 次のデバイス・オブジェクトへの参照	NUMBER	可
16 NEXT_DEVICE_NAME: 次のデバイス・オブジェクトの名前	VARCHAR2(256)	可
17 NEXT_DEVICE_TAG_ID: 次の時間を表すタグ ID (EPC コード)	VARCHAR2(256)	可
18 TIME_DIFF: デバイス間の移動にかかる時間	INTERVAL DAY() TO SECOND()	可

```

SELECT
device_name
.device_tag_id
.device_desc
.site_name
.event_id
.event_tag_id
.device_id
.event_data
.event_time
.event_correlation_id
.event_name
.event_type
.event_subtype
.event_desc
.next_device_id
.next_device_name
.next_device_tag_id
.(event_end_time-event_time) time_diff
FROM (
  SELECT edv.*,
    DECODE( LEAD(edv.device_id) OVER (PARTITION BY edv.event_tag_id ORDER BY
edv.event_time), edv.device_id,
0, 1) hop,
    LEAD(edv.device_id) OVER (PARTITION BY edv.event_tag_id ORDER BY edv.event_time)
next_device_id,
    LEAD(edv.device_name) OVER (PARTITION BY edv.event_tag_id ORDER BY edv.event_time)
next_device_name,
    LEAD(edv.device_TAG_ID) OVER (PARTITION BY edv.event_tag_id ORDER BY edv.event_time)
next_device_tag_id,
    LEAD(edv.event_time) OVER (PARTITION BY edv.event_tag_id ORDER BY edv.event_time)
event_end_time
FROM edg_dev_event_vw edv
) ev_path
WHERE hop=1

```

PL/SQL プログラミング・インタフェース

センサー・データ・リポジトリには次の PL/SQL パッケージが含まれます。

EDG_SDA パッケージ

```
1: PACKAGE EDG_SDA IS
2:
3: --- procedure to handle the incoming event
4: --- and sort out how to disassemble the parts
5: --- and put them into various tables.
6:
7: PROCEDURE on_event
8: (p_correlation_id IN edg_event_tab.correlation_id%TYPE
9: .p_source_name IN edg_event_tab.source_name%TYPE
10: .p_site_name IN edg_device_tab.site_name%TYPE
11: .p_device_name IN edg_device_tab.name%TYPE
12: .p_type IN edg_event_info_tab.type%TYPE
13: .p_subtype IN edg_event_info_tab.type%TYPE
14: .p_time IN edg_event_tab.time%TYPE
15: .p_id IN edg_event_tab_tag.id%TYPE
16: .p_data IN edg_event_tab.data%TYPE
17: );
18:
19: --- procedure to add a new context
20: PROCEDURE create_ctxt
21: (p_object_id) OUT edg_ctxt_tab.object_id%TYPE
22: .p_name IN edg_ctxt_tab.name%TYPE
23: .p_description IN edg_ctxt_tab.description%TYPE
24: .p_parent_ctxt_tab.object_id%TYPE
25: );
26:
27: ---procedure to add a new context
28: PROCEDURE create_ctxt
29: (p_name IN edg_ctxt_tab.name%TYPE
30: .p_description IN edg_ctxt_tab.description%TYPE
31: .p_parent_ctxt_id IN edg_ctxt_tab.object_id%TYPE
32: );
33:
34: --- procedure to update the context hierarchy
35: PROCEDURE update_ctxt_rel
36: (p_cid IN edg_ctxt_rel_tab.cid%TYPE
37: .p_pid IN edg_ctxt_rel_tab.pid%TYPE
38: );
39:
40: --- procedure to update the context
41: PROCEDURE update_ctxt
42: (p_object_id IN edg_ctxt_tab.object_id%TYPE
43: .p_name IN edg_ctxt_tab.name%TYPE
44: .p_description IN edg_ctxt_tab.description%TYPE
45: );
46:
47: --- procedure to remove a context
48: PROCEDURE remove_ctxt
49: (p_object_id IN edg_ctxt_tab.object_id%TYPE
50:
51: END EDG_SDA;
```

Oracle Sensor Edge Mobile

この章では Oracle Sensor Edge Mobile について説明します。この章の内容は次のとおりです。

- 「Oracle Sensor Edge Mobile の概要」
- 「ディスパッチャとドライバの構成」
- 「キーボード・ディスパッチャの構成」
- 「デバイス・ステータスの確認」
- 「ActiveX アプリケーション・インタフェース」
- 「Sensor Edge Mobile の管理」
- 「国際化」

Oracle Sensor Edge Mobile の概要

モバイルの Oracle Sensor Edge Server (Sensor Edge Mobile) は、バーコードの読取りと RFID タグの読書きを行うドライバで構成されたクライアント側 (Pocket PC 専用) アプリケーション (図 5-1) です。

図 5-1 携帯情報端末での「Sensor Edge Mobile Administration」ページ



アーキテクチャは、アプリケーションと直接通信できるディスパッチャが含まれるという点で、Oracle Sensor Edge Server のアーキテクチャと共通しています。たとえば、Sensor Edge Mobile には、ActiveX コントロールと対話するディスパッチャが含まれます。このため、JavaScript を使用して HTML で記述された管理ページによって、ドライバを制御し、返されたデータを表示または発行することができます。

Sensor Edge Mobile は、センサー・デバイスとアプリケーションの間に位置します。Sensor Edge Mobile は、様々なタイプのセンサーおよびデバイスとのインターフェースとして機能する一方で、フィルタリングされたデータ・イベントをアプリケーションに送信します。通信は双方向です。イベントはプラットフォームを介してデバイスからアプリケーションに渡され、命令はプラットフォームを介してアプリケーションからハードウェア・デバイス・ドライバに渡されます。

Sensor Edge Mobile サービスは、完全に携帯情報端末上で稼働し、他のアプリケーションや外部のサービスと通信できます。また、完全にオフラインで実行して、データを収集し、後から外部アプリケーションと同期化することもできます。イベントは、収集デバイスからドライバに送信されます。イベントは、ディスパッチャにより処理可能になるまで、キューによって収集されます。ディスパッチャは、ActiveX コントロールを介して、文字をキーボード・バッファに送信することにより (または他のほとんどあらゆる通信方法によって)、アプリケーションと通信できます。同時にアクティブにできるディスパッチャは1つのみですが、そのディスパッチャは複数のクライアント・デバイスおよび任意の数のドライバと同時に通信できます。

Sensor Edge Mobile のコード・コンポーネントには、ドライバ・マネージャ、イベント・マネージャおよび構成マネージャが含まれます。これらのコンポーネントが起動されると、メイン・サービスによって構成マネージャが呼び出され、構成ファイルが読み取られます。次に構成済みのディスパッチャが起動され、指定されている構成パラメータが渡されます。

ドライバ・マネージャは、ドライバをロードし、そのライフ・サイクルを管理します。ドライバ・マネージャは構成マネージャを呼び出して、ロードする必要のあるドライバと、インスタンス化時にドライバに対して使用可能にするパラメータを判別します。次に、それらのドライバをロードしてインスタンス化します。ドライバ・マネージャは内部でスレッドを保持しませんが、プラットフォームが実行されている間は、コア・インスタンスによりドライバ・マネージャのインスタンスが保持されます。

Sensor Edge Mobile には、イベント・マネージャとドライバ・マネージャが 1 つずつだけ存在します。ドライバ・マネージャは任意の数のドライバをロードできます。

アプリケーションへの Sensor Edge Mobile の接続

Sensor Edge Mobile の設計は、アプリケーション・プログラマがデバイスから発信されたイベントに簡単にアクセスし、デバイスに命令を発行できるようになっています。アプリケーションは、直接ディスパッチャと通信することも、(ActiveX コントロールなどの) 別のレイヤーを介してディスパッチャと通信することもできます。

アプリケーションは、次のような手段により、Sensor Edge Mobile とのインタフェースとして機能します。

- ActiveX コントロール。この場合、ActiveX コントロールが、バックグラウンドでサービス・プロセスとして実行されている Sensor Edge Mobile と通信します。
- キーボード・ディスパッチャ。この場合、指定されたアプリケーションのウィンドウにフォーカスが移り、ユーザーが入力したときと同じように文字がキーボード・バッファに入力されます。詳細は、「[キーボード・ディスパッチャの構成](#)」を参照してください。
- カスタム・ディスパッチャ。ディスパッチャは動的にロードされ、サード・パーティが特殊なディスパッチャを開発できるようにインタフェースが公開されます。

ディスパッチャとドライバの構成

ドライバと現行ディスパッチャは、Oracle Sensor Edge Mobile サービスの起動時に構成情報を取得します。この構成は、名前 / 値ペアで構成されるパラメータとして表されます。

ドライバと現行ディスパッチャは、Sensor Edge Mobile アプリケーションと同じディレクトリ内にある EdgeMobileCofig.xml (例 5-1) 内で構成します。このファイルでは、<CurrentDispatcher> パラメータ内のサービスにより使用される 1 つのディスパッチャと、<DeviceList> 要素内にロードされたドライバが記述されます。これらの要素は異なるパラメータ値を渡します。例 5-1 では、現行ディスパッチャとしてキーボード・ディスパッチャが設定され、(IP3Driver という名前の) Intemec IP3 ドライバ (このファイル内で定義されている唯一の <device> 要素) が設定され、必要なパラメータが渡されています。

例 5-1 EdgeMobileConfig.xml

```
<EdgeMobileConfig>
  <CurrentDispatcher>
    <Name>Keyboard Dispatcher</Name>
    <Version>1.0</Version>
    <Description>Keyboard Dispatcher</Description>
    <Library>KeyboardDispatcher.dll</Library>
    <Parameters>
      <Param name="DestinationApplication" value="Telnet"/>
      <Param name="RFIDReadMacro" value="TelnetMultipleRFID"/>
      <Param name="BarcodeReadMacro" value="TelnetMultipleBarcode"/>
      <Param name="TelnetSingle" type="KeySequenceMacro">
        <tab/><tab/>RFID<tab/>
        <data name="TagID"/>
        <tab/>
        <data name="TagData"/>
        <cr/>
      </Param>
      <Param name="TelnetMultipleRFID" type="KeySequenceMacro">
```

```

<tab/><tab/>RFID<tab/>
<Repeat>
  <data name="tag_id"/>
  <tab/>
  <data name="data"/>
  <Separator>
    <tab/><tab/>
  </Separator>
</Repeat>
<Cr/>
</Param>
<Param name="TelnetMultipleBarcode" type="KeySequenceMacro">
  <tab/><tab/>Barcode<tab/>
<Repeat>
  <data name="data"/>
  <Separator>
    <tab/><tab/>
  </Separator>
</Repeat>
<Cr/>
</Param>
</Parameters>
</CurrentDispatcher>
<DeviceList>
  <Device>
    <Name>IP3Driver</Name>
    <Version>1.1</Version>
    <Description>Intermec RFID IP3 driver</Description>
    <Library>IP3Driver.dll</Library>
    <Parameters>
    </Parameters>
  </Device>
</DeviceList>
</EdgeMobileConfig>

```

キーボード・ディスパッチャの構成

キーボード・ディスパッチャは、データを収集し、データをキーボード・バッファに入力し、キーボード・コードをマクロに入力します。このマクロにより、Telnet セッションや Web ブラウザなどのアプリケーションの実行中にユーザーの入力がシミュレーションされます。キーボード・ディスパッチャは次の操作を（次の順序で）実行します。

1. デバイス・ドライバからイベント・データを取得します。
2. 指定されたアプリケーションのウィンドウにフォーカスを移します。
3. アプリケーション・ウィンドウに文字が入力されると、データおよび KeySequenceMacro 言語で指定された制御文字を送信します。

ロードするディスパッチャおよびドライバを指定し、これらの拡張機能に構成パラメータを渡すには、EdgeMobileConfig.xml を変更します。Sensor Edge Mobile でキーボード・ディスパッチャを使用するには、例 5-2 に示すように、構成ファイルの <CurrentDispatcher> 要素としてキーボード・ディスパッチャを定義する必要があります。

例 5-2 現行ディスパッチャとしてのキーボード・ディスパッチャの構成

```

<CurrentDispatcher>
  <Name>Keyboard Dispatcher</Name>
  <Version>1.0</Version>
  <Description>Keyboard Dispatcher</Description>
  <Library>KeyboardDispatcher.dll</Library>
  <Parameters>
    <Param name="DestinationApplication" value="Telnet"/>
    <Param name="RFIDReadMacro" value="TelnetMultipleRFID"/>
  </Parameters>
</CurrentDispatcher>

```

```

<Param name="BarcodeReadMacro" value="TelnetMultipleBarcode"/>
<Param name="TelnetSingle" type="KeySequenceMacro">
  <tab/><tab/>RFID<tab/>
  <data name="tag_id"/>
  <tab/>
  <data name="data"/>
  <enter/>
</Param>
<Param name="TelnetMultipleRFID" type="KeySequenceMacro">
  <tab/><tab/>RFID<tab/>
  <Repeat>
    <data name="tag_id"/>
    <tab/>
    <data name="data"/>
    <Separator>
      <tab/><tab/>
    </Separator>
  </Repeat>
  <enter/>
</Param>
<Param name="TelnetMultipleBarcode" type="KeySequenceMacro">
  <tab/><tab/>Barcode<tab/>
  <Repeat>
    <data name="data"/>
    <Separator>
      <tab/><tab/>
    </Separator>
  </Repeat>
  <enter/>
</Param>
</Parameters>
</CurrentDispatcher>

```

DestinationApplication パラメータの定義

DestinationApplication パラメータは、キーボード・バッファにより Sensor Edge Mobile サービスからの文字データが送信される前にフォーカスを移すアプリケーションの名前を定義します。このパラメータの構文は次のとおりです。

```
<Param name="DestinationApplication" value="Telnet"/>
```

アプリケーションのパスではなく、アプリケーション名そのもの（例 5-1 では "Telnet"）を指定します。ほとんどのアプリケーションでは、アプリケーション・ウィンドウの最上部に、ドキュメント名とダッシュに続いてこの名前が表示されます（たとえば、Microsoft Word アプリケーションの場合は「Doc1.doc - Microsoft Word」）。

このようなアプリケーション・ウィンドウが複数開いている場合は、その中で最も前面にあるウィンドウ（"z-order" の最初のウィンドウ）が前面になり、アクティブになります。

RFIDReadMacro パラメータの定義

RFIDReadMacro パラメータは、RFID 読取りによる測定データに使用する構成ファイル内のキー・シーケンス・マクロを定義します。このパラメータにより、構成ファイル内のキー・シーケンス・マクロを多数定義できます。そのうちの1つを選択するには、このパラメータ内に目的のマクロ名を挿入します。このパラメータの構文は次のとおりです。

```
<Param name = "RFIDReadMacro" value = "TelnetMultipleRFID"/>
```

BarcodeReadMacro パラメータの定義

BarcodeReadMacro パラメータは、バーコード読取りによる測定データに使用する構成ファイル内のキー・シーケンス・マクロを定義します。このパラメータにより、構成ファイル内のキー・シーケンス・マクロを多数定義できます。そのうちの1つを選択するには、このパラメータ内に目的のマクロ名を挿入します。このパラメータの構文は次のとおりです。

```
<Param name="BarcodeReadMacro" value="TelnetMultipleBarcode"/>
```

キー・シーケンス・マクロ・パラメータの定義

キーボード・ディスプレイパッチャは、Sensor Edge Mobile により収集されたデータを、ユーザーにより入力されたときと同じように受信側アプリケーションに送信します。既存のアプリケーションに対してこれを設定するには、まず、ブラウザまたはアプリケーション・ウィンドウのフォーカスを、データを受信する特定の入力コントロールに移動します。たとえば、タブ文字を特定の回数送信すると、フォーカスが特定の入力フィールドに移動します。また、データを送信する（またはアクティブにする）ための [Enter] キー入力を生成することもできます。アプリケーションに必要な特定のキーストローク・シーケンスを指定するには、(例 5-3 に示すように) キー・シーケンス・マクロを定義する必要があります。

例 5-3 キー・シーケンス・マクロの定義

```
<Param name="TelnetSessionRFID" type="KeySequenceMacro">
  <tab/><tab/>RFID<sp/>Data<tab/>
  <data name="tag_id"/>
  <tab/>
  <data name="data"/>
  <enter/>
</Param>
```

例 5-2 では、パラメータに一意のユーザー定義名 ("TelnetSessionRFID") が指定されています。また、その属性 type を "KeySequenceMacro" として指定する必要があります。マクロ本体には、一連の制御文字、指定されたスペース (<sp/>)、および (ユーザーがキーボードで入力したときと同様に) データ値が送信される先を指定する 1 つ以上の <data> タグが含まれます。

たとえば、データ値が "tag id" の場合に Sensor Edge Mobile から送信される値が "123456789" で、RFID タグからの "data" の値が "0A0B0C0D0E0F" である場合、アプリケーションには次のシーケンスが送信されます。

```
¥t¥tRFID Data¥t123456789¥t0A0B0C0D0E0F¥r
```

ここで、¥t はシーケンス内のタブ文字を表し、¥r は改行 ([Enter]) 文字を表します。これにより、アプリケーション・ウィンドウの最初の入力フィールドにタブ移動し、"RFID Data" (スペース文字コード <sp/> が必要) が入力されます。次に TagID フィールドにタブ移動し、タグ ID が入力され、TagData フィールドを使用して同じルーチンが実行されます。最後に、改行 ([Enter]) が送信されます。改行は、ボタンのクリック、データの発行、またはデータの保存に対して使用できます。

XML 内の改行、タブおよびスペースはアプリケーションに送信されません。構成ファイル内のマクロのコンテンツは、不要な文字をアプリケーションに送信することなく、タブ移動および整形印刷できます。たとえば、例 5-4 では、"Macro1" と "Macro2" は同じ文字シーケンスを送信します。

例 5-4 文字シーケンスの送信

```
<Param name="Macro1" type="KeySequenceMacro">
  <data name="tag_id"/>
  <tab/>
  <data name="data"/>
  <enter/>
</Param>
<Param name="Macro2" type="KeySequenceMacro"><data name="tag_id"/><tab/><data
name="data"/><enter/></Param>
```

キー・シーケンス・マクロの作成

キー・シーケンス・マクロを作成するには、送信先アプリケーションの入力フィールドおよびデータ・フィールドの入力場所に移動するために必要なキーストロークを記録します。次に、例 5-4 に示すように、このシーケンスをキー・シーケンス・マクロとして記述します。最後に、"RFIDReadMacro" または "BarcodeReadMacro" パラメータを新しいキー・シーケンス・マクロの名前（例 5-5 では "MyMacro"）に設定します。

例 5-5 RFIDReadMacro を新しいキー・シーケンス・マクロの名前に設定

```
<Param name="RFIDReadMacro" value="MyMacro"/>
  <Param name="MyMacro" type="KeySequenceMacro">
    <data name="tag_id"/>
    <tab/>
    <data name="data"/>
    <enter/>
  </Param>
```

これで、RFID 読取りから発生したイベントによって、例 5-5 に示すキー・シーケンスが生成され、<data> 要素により指定された値が挿入されます。

キー・シーケンス・マクロによる繰返し要素処理の有効化

これまでに示した例では、RFID タグからの "tag_id" および "data" データ要素 1 セットのみを処理していました。このマクロを連続するこれらのデータ値の組に対しても有効にするには、繰返しの概念を定義する必要があります。<Repeat> および <Separator> マクロ要素を使用して、繰返すキー・シーケンス・マクロ部分を指定し、次の繰返しに入る前にアプリケーションの次の行またはフィールドにフォーカスを移すために送信する文字シーケンスを指定します。

例 5-6 <Repeat> 要素と <Separator> 要素の定義

```
<Param name="TelnetMultipleRFID" type="KeySequenceMacro">
  <tab/><tab/>RFID<tab/>
  <Repeat>
    <data name="tag_id"/>
    <tab/>
    <data name="data"/>
    <Separator>
      <tab/><tab/>
    </Separator>
  </Repeat>
  <enter/>
</Param>
```

例 5-6 には、<Repeat> および <Separator> マクロ要素が示されています。これらの要素により、各データ・セットで繰り返すマクロのセクションと、次のデータ・セットに入る前に送信するシーケンスを指定します。

データ測定に 3 つの組を指定する場合、"Tag_id" と "data" 値の組 ("Tag1"、"Data1")、("Tag2"、"Data2") および ("Tag3"、"Data3") を使用します。例 5-6 に示すマクロから生成されるシーケンスは、
`¥t¥tRFID¥tTag1¥tData1¥t¥tTag2¥tData2¥t¥tTag3¥tData3¥r` です。

このシーケンスを分解すると、<Repeat> の前の部分は次のデータを 1 回送信します。

`¥t¥tRFID¥t`

繰り返し部分 `<data name="tag_id"/><tab/><data name="data"/>` は次のデータを送信します。

`Tag1¥tData1 Tag2¥tData2 Tag3¥tData3`

<Separator> シーケンスで区切られた各部分は次のデータを送信します。

`¥t¥t ¥t¥t`

最後に、改行 ([Enter]) で次のデータを 1 回送信します。

`¥r`

<Separator> シーケンスはそれぞれの繰り返しシーケンスを区切りますが、最後の繰り返しシーケンスの後には続きません。この例では、最後のデータ値の組 (この場合は "Tag3¥tData3") の後に "¥t¥t" セパレータ・シーケンスは送信されません。

特殊な制御シーケンスのキー・マクロ要素キー アプリケーションで必要とされるキーボード・キー入力をシミュレーションするために、キーボード・シーケンス・マクロ内で特殊な制御シーケンスを使用できます (表 5-1 を参照)。

表 5-1 キーボード・キーの制御シーケンス

要素	送信されるキー・シーケンス
<space/>	[Space]
<sp/>	[Space]
<enter/>	[Enter]
<cr/>	[Enter]
<backspace/>	[Back Space]
<bs/>	[Back Space]
<tab/>	[Tab]
<backtab/>	[Shift]+[Tab]
<clear/>	[Clear]
<esc/>	[Esc]
<pageup/>	[Page Up]
<pagedown/>	[Page Down]
<end/>	[End]
<home/>	[Home]
<leftarrow/>	[←]
<uparrow/>	[↑]
<rightarrow/>	[→]

表 5-1 キーボード・キーの制御シーケンス (続き)

要素	送信されるキー・シーケンス
<downarrow/>	[↓]
<ins/>	[Insert]
	[Del]
<cancel/>	[Ctrl]+[Break]
<f1/>	[F1]
<f2/>	[F2]
<f3/>	[F3]
<f4/>	[F4]
<f5/>	[F5]
<f6/>	[F6]
<f7/>	[F7]
<f8/>	[F8]
<f9/>	[F9]
<f10/>	[F10]
<f11/>	[F11]
<f12/>	[F12]

例 5-6 に示すように、リテラル文字のシーケンスを送信するには、<sp/> および <tab/> のようにスペースまたはタブを明示的に入力する必要があります。これ以外のタブおよびスペースがマクロ内にある場合、XML の整形印刷フォーマットとして無視されます。これには別の理由もあります。タブおよびスペースを解析するための XML 標準では、要素間のこれらの「空白」すべてが保持されるわけではありません。空白が変更されると、マクロの処理も変更されて予測が困難になるためです。

制御キーまたはデータ配置のためのキーボード・マクロ要素 <control char="x"/> および <data name="tag_id"/> マクロ要素 (表 5-2 を参照) は、生成される制御キーまたはデータの入力場所を指定します。

表 5-2 キーボード・マクロ要素

要素	説明
<control char="x"/>	[Ctrl]+ 文字
<data name="tag_id"/>	名前付きデータ値

特殊な <control> 要素を指定すると、ユーザーが [Ctrl] キーを押しながら特定の文字を入力したときと同じように、制御シーケンスが生成されます。たとえば、[Ctrl]+[B] を生成するには、<control char="b"/> を指定します。

<data> 要素を指定すると、測定イベント内のデータの現在の値がフェッチされて現在の位置に挿入されます。特定の名前とその意味（たとえば、"tag_id" は RFID タグの Tag-ID を表すなど）は、名前と値の組としてアプリケーションに対して使用可能になります。<data> 要素の定義済の値を表 5-3 に示します。

表 5-3 <data> 要素の定義済の値

名前	説明
device_name	測定イベントを生成するデバイスの名前。
device_desc	測定イベントを生成するデバイスの説明。
timestamp	イベント作成のタイムスタンプ（テキスト表示形式）。
block_index	イベント・ブロック（グループ）内のこのイベントの索引。
block_count	このイベント・ブロック内のイベント数。
tag_id	RFID: 16 進文字列の RFID タグの TagID。データ・ペイロードは "data" 内で指定する。
data	RFID: 16 進文字列のデータ・ペイロード。バーコード: 16 進文字列のバーコード読取りデータ。
type	バーコード: バーコードのタイプ ("1d" または "2d")。
checksum	バーコード: 16 進文字列のチェックサム値。
encoding_format	バーコード: エンコーディング形式。

（「キー・シーケンス・マクロによる繰り返し要素処理の有効化」で説明されているように）<data> 要素を <Repeat> 要素と組み合わせると、読取りにより複数のイベントが返された場合にデータ値のシーケンスを反復することができます。"block_index" および "block_count" データ値の詳細は、「測定イベントの処理」を参照してください。

RFID 読取りで何もタグが読み取られなかった場合、"tag_id" および "data" 値が空の文字列（""）に設定されてイベントが返されます。

デバイス・ステータスの確認

デバイス・マネージャは、デバイスに ping を発行して、そのデバイスが動作していて応答可能であるかどうかを判断します。応答していないデバイスは、ユーザー操作なしで再起動されず。

デバイス・マネージャは、デバイスからイベントが返された最後の時刻を記録し、最後のイベントから長時間経過している場合は、ドライバに対して OS イベントに関する ping を発行します。ドライバ・インタフェースの規約では、デバイスが ping からのレスポンスという特殊なイベントに適切な時間内に応答していることが必要です。デバイスが構成可能な時間内に応答していない場合、デバイス・マネージャはデバイスに停止コマンドを発行します。一定時間が経過しても停止承認イベントによる応答がない場合、デバイス・マネージャはドライバの .dll をアンロードし、新しいデバイスをロードして初期化します。

ActiveX アプリケーション・インタフェース

この改訂済 ActiveX API では、多数の主要な新機能および旧ベータ・リリースから変更された機能があります。

- 設計者が測定データをフレキシブルに自在に処理または表示できるように、コントロールそのものは表示されません。
- 測定データ・イベントを受信するための関数を登録することにより、データは JavaScript に返されます。アプリケーションはこのデータを使用して入力フィールドを設定したり、その他任意の処理を実行できます。
- 新しい API では、現在構成されているドライバによりサポートされているデバイス・クラスのうちいずれにも同時にアクセスできます。たとえば、RFID データとバーコード・データを 1 つの HTML ページから読み取ることができます。

オブジェクト宣言

Oracle Sensor Edge Mobile クライアント・コントロールのオブジェクト宣言を例 5-7 に示します。この例では、"event_handler" パラメータにより、測定イベントが受信されるとコールされる JavaScript 関数の名前が登録されています。パラメータを指定しない場合は、デフォルトのイベント・ハンドラ関数 "handle_event ()" が試行されます。この関数が存在しない場合、エラーは返されず、イベント・ハンドラなしで処理が続行されます。

例 5-7 Sensor Edge Mobile クライアント・コントロールのオブジェクト宣言

```
<object id="SEMobileControl" name="SEMobileControl"
  classid="CLSID:049BE519-EE78-4AA5-8FC8-C5AE084CB26C"
  width="0"
  height="0">
  <param name="event_handler" value="handle_event"/>
<table><tr><td bgcolor="#FF0000">
<hr/>Unable to load Sensor Edge Mobile control.<hr/>
</td></tr></table>
</object>
```

Activex API には次のメソッドが含まれます。

- rfid_read()
- rfid_write()
- rfid_kill()
- barcode_read()
- set_trigger_rfid_read()
- set_trigger_barcode_read()
- process_instruction()
- is_supported()

rfid_read()

`rfid_read()` メソッドは、範囲内のすべてのタグに対する読取りを発行します。メソッド・シグネチャは次のとおりです。

```
var retVal = SEMobileControl.rfid_read( );
```

タグ読取りは、登録済イベント・ハンドラ・メソッドを 0 回以上コールバックすることにより返されます。読取りが正常に完了した場合、戻り値は 0 になります。なんらかの理由で失敗した場合は、0 以外の値が返されます。

rfid_write()

`rfid_write()` メソッドは、範囲内のすべてのタグに対する書込みを発行します。メソッド・シグネチャは次のとおりです。

```
var retVal = SEMobileControl.rfid_write( srcForm.srcData.value,  
                                         srcForm.srcPasscode.value );
```

この例では、パラメータは `srcForm` という形式の非表示変数です。`srcData` 変数は書き込むデータを表し、16 進文字列で表されます。`srcPasscode` には書込み用のパスコードが含まれます。書込みが正常に完了した場合、戻り値は 0 になります。なんらかの理由で失敗した場合は、0 以外の値が返されます。

rfid_kill()

`rfid_kill()` メソッドは、範囲内のすべてのタグに対する中断を発行します。メソッド・シグネチャは次のとおりです。

```
var retVal = SEMobileControl.rfid_kill( srcForm.srcTagId.value,  
                                        srcForm.srcPasscode.value );
```

`srcTagId` 変数は、中断するタグの TagID (16 進文字列) を表します。`srcPasscode` には中断用のパスコードが含まれます。中断が正常に完了した場合、戻り値は 0 になります。なんらかの理由で失敗した場合は、0 以外の値が返されます。

barcode_read()

`barcode_read()` メソッドは、デバイスにより指し示されたバーコードの読取りを発行します。メソッド・シグネチャは次のとおりです。

```
var retVal = SEMobileControl.barcode_read( );
```

バーコード読取り測定イベントは、登録済イベント・ハンドラ・メソッドをコールバックすることにより返されます。読取りが正常に完了した場合、戻り値は 0 になります。なんらかの理由で失敗した場合は、0 以外の値が返されます。

set_trigger_rfid_read()

`set_trigger_rfid_read()` メソッドは、ユーザーがデバイスに対してトリガーを引いたときに RFID データの読取りを実行する必要があることを Sensor Edge Mobile に指示します。メソッド・シグネチャは次のとおりです。

```
var retVal = SEMobileControl.set_trigger_rfid_read();
```

トリガーを引く操作に対して RFID 読取り操作を登録するためのコールが必要です。このコールが正常に完了した場合、戻り値は 0 になります。コールが失敗した場合、RFID デバイス・ドライバはロードされず、0 以外の値が返されます。

set_trigger_barcode_read()

set_trigger_barcode_read() メソッドは、ユーザーがデバイスに対してトリガーを引いたときにバーコード・データの読取りを実行する必要があることを Sensor Edge Mobile に指示します。メソッド・シグネチャは次のとおりです。

```
var retVal = SEMobileControl.set_trigger_barcode_read();
```

トリガーを引く操作に対してバーコード読取り操作を登録するためのコールが必要です。このコールが正常に完了した場合、戻り値は 0 になります。コールが失敗した場合、バーコード・デバイス・ドライバはロードされず、0 以外の値が戻されます。

process_instruction()

process_instruction メソッドは、ドライバによる命令を処理するための下位レベル・インタフェースです。このメソッドは、現在定義されている RFID およびバーコードの操作には必要ありませんが、このメソッドを使用可能にすると、カスタム・ドライバの使用時に命令レパトリを簡単に拡張できます。メソッド・シグネチャは次のとおりです。

```
var retVal=SEMobileControl.process_instruction(instruction, data);
```

instruction は、定義済定数のリストからの（文字列として表された）整数値です。

EdgeMobilehtml_samples では、SEMobileCore.js という名前のファイルです。次のようなスクリプト要素を使用して、このファイルの値を HTML ドキュメントに挿入できます。

```
<script type="text/javascript" src="SEMMobileCore.js"></script>
```

この場合、これらの定数を使用して RFID 読取りを開始するための命令は次のようになります。

```
var retVal=SEMobileControl.rfid_read();
```

この命令は、次の操作と同じ操作を実行します。

```
var retVal=SEMobileControl.rfid_read();
```

コマンドに使用可能な値は次のとおりです。

- シンボルの説明

RFID_READ_START: RFID 読取り。引数の値は無視されます。RFID_WRITE

DESCRIPTION: デバイス生成イベントの文字列。RFID_KILL: テキスト表示形式のイベント作成のタイムスタンプ。BARCODE_READ: イベント・ブロック（グループ）内のこのイベントの索引。REGISTER_TRIGGER0: このイベント・ブロック内のイベントの数。

TRIGGER_PULL0 RFID: 16 進文字列の RFID タグの TagID。データ・ペイロードは "data" 内に指定します。

バーコード読取りを開始するための命令は次のようになります。

```
var retVal = SEMobileControl.process_instruction(BARCODE_DEAD, "");
```

これは、次のメソッド・コールと同等です。

```
var retval = SEMobileControl.barcode_read();
```

これらの定数を使用して RFID 書込みを実行するための命令は次のようになります。

```
var retVal = SEMobileControl.process_instruction(RFID_WRITE, "0102030405060708,123");
```

書き込む 16 進値とパスコードとをカンマで区切ります。これは、次の命令と同等です。

```
var retVal = SEMobileControl.rfid_write("0102030405060708", "123");
```

RFID 中断命令は次のようになります。

```
var retval = SEMobileControl.process_instruction(RFID_KILL, "0102030405060708,123");
```

RFID 書込みの場合と同じように、書き込む 16 進値とパスコードとをカンマで区切ります。これは、次の命令と同等です。

```
var retval = SEMobileControl.rfid_kill("0102030405060708", "123");
```

最後に、特定の読取り操作に対してトリガーを引く操作を登録するための命令は次のようになります。

```
var retval = SEMobileControl.process_instruction(
REGISTER_TRIGGER0, RFID_READ);
```

これは、次の命令とまったく同じ操作を実行します。

```
var retval = SEMobileControl.set_trigger_rfid_read();
```

is_supported()

is_supported メソッドは、RFID_WRITE などの命令コードの 1 つの引数を持ち、この命令を処理できる構成済ドライバがあるかどうかに基づいて true または false を返します。メソッド・シグネチャは次のとおりです。

```
if (SEMobileControl.is_supported( RFID_KILL) { //RFID_KILL instruction supported by a
configured driver...}
```

instruction は、SEMobileCore.js ファイル内の定義済定数のリストからの（文字列として表された）整数値です。

測定イベントの処理

測定イベントが受信されると、ActiveX コントロールにより登録済の JavaScript イベント・ハンドラ関数がコールされます。イベント・ハンドラ・メソッド内では、イベント・オブジェクトは ActiveX コントロール・インスタンス名で参照されます。例 5-8 および例 5-9 では、"SEMobileControl" がコントロール名として使用されています。

イベント・データは、イベント・ハンドラ関数から JavaScript が返されるまで、ActiveX オブジェクトから使用可能です。すべてのイベント内に存在するデータはプロパティとして使用可能であり、特定のイベント・タイプに固有のその他のデータは検索により使用可能です。イベント・データを表 5-4 に示します。

表 5-4 使用可能なイベント・データ

名前	説明
device_name	測定イベントを生成するデバイスの名前。
device_desc	測定イベントを生成するデバイスの説明。
is_rfid_read	イベントが RFID 読取りイベントの場合は "true" を返す。
is_barcode_read	イベントがバーコード読取りイベントの場合は "true" を返す。
timestamp	イベント作成のタイムスタンプ。
block_index	イベント・ブロック（グループ）内のこのイベントの索引。
block_count	このイベント・ブロック内のイベント数。
type	イベント・オブジェクトのタイプ・コード。
subtype	イベント・タイプのサブタイプ・コード。

イベントのプロパティを使用してイベントに関するレポートを生成するイベント・ハンドラ関数を、例 5-8 に示します。

例 5-8 イベント・レポートを生成するためのイベント・ハンドラ関数

```
function handle_event ()
{
  if ( SEMobileControl.is_barcode_read ) {
    alert ("Barcode data read from " + SEMobileControl.device_name );
  }
  if ( SEMobileControl.is_rfid_read ) {
    alert ("RFID data read from " + SEMobileControl.device_name );
  }
}
```

これらのプロパティはすべて、イベント・ハンドラ関数が返されるまでの間のみ使用可能です。イベント・ハンドラ関数が返された時点で、Sensor Edge Management コントロールは別のイベントを持つハンドラを再びコールできるようになり、最後のイベントに関連付けられているデータは使用できなくなります。

RFID タグが読み取られると、返されたイベントはブロックの一部としてマークされます (1 つのタグの読取りが 1 つのイベントのブロックとして表されます)。これを反映するため、すべてのイベントはブロック索引と、それに関連付けられたブロック・カウントを持ちます。イベントがシングルトンの場合、ブロック・カウントと索引は "1" に設定されます。これに対し、ブロックに 3 つのイベントが含まれる場合、そのブロック内の 3 つのイベントのブロック索引とカウントの値は 1 と 3、2 と 3、および 3 と 3 のようになります。"block_index" プロパティと "block_count" プロパティにより、これらの値へのアクセスが許可されます。

イベントに関連付けられたデバイスのクラスに固有の他のデータが存在する場合もあります。これらのデータは名前と値の組として表され、次の data() メソッドを使用して検索可能です。data() プロパティを使用すると、特定のデバイス・クラスからのイベントに固有のデータ値を検索できます。デバイス・クラスごとに異なるデータ名が定義されています。現在のイベント内にデータ値が存在しない場合は、空の文字列 ("") が返されます。定義されているデータ名とそれに関連付けられているデバイス・クラスを表 5-5 に示します。

表 5-5 データ名

名前	説明
tag_id	RFID: 16 進文字列の RFID タグの TagID。データ・ペイロードは "data" 内で指定する。
data	RFID: 16 進文字列のデータ・ペイロード。バーコード: 16 進文字列のバーコード読取りデータ。
type	バーコード: バーコードのタイプ ("1d" または "2d")。
checksum	バーコード: 16 進文字列のチェックサム値。
encoding_format	バーコード: エンコーディング形式。

イベント・ハンドラが最初にデバイス・クラス (RFID または バーコード) を判別してから、`data()` を参照してデバイス・クラスに固有のデータ値をフェッチする方法を、例 5-9 に示します。

例 5-9 デバイス・クラスに固有のデータ値のフェッチ

```
function handle_event ()
{
  if ( SEMobileControl.is_barcode_read ) {
    alert ("Barcode data is " + SEMobileControl.data("data") );
  }
  if ( SEMobileControl.is_rfid_read ) {
    /* Note that "tag_id" is an RFID-specific event value */
    alert ("RFID tag ID is " + SEMobileControl.data("tag_id") );
  }
}
```

廃止された Activex アプリケーション・インタフェース

現在の Sensor Edge Mobile ActiveX コントロールには、次のようにしてアクセスできます。

- ドライバにより収集された測定データが含まれる HTML ページ上の可視コントロールとして使用。
- コントロールの API を使用。この場合、コントロールはページに表示されませんが、JavaScript の API が提供されます。
- 可視コントロールと使用中の API を組み合わせて使用。

この API では、特定の HTML ページから 1 つのデバイス・クラス (バーコードまたは RFID) へのアクセスが許可されるのみです。

注意： この API は、Oracle Sensor Edge Server 10g リリース 3 (10.1.3) では廃止されました。

Sensor Edge Mobile の管理

管理 HTML ページ (図 5-1) で、Oracle Sensor Edge Mobile を管理できます。このページを使用すると、次の操作を実行できます。

- 拡張機能の構成と管理

ドライバの場合、構成済のドライバとその状態 (初期化中、アクティブ、停止または応答なし) が HTML ページに示されます。リリース 3 (10.1.3) の Oracle Sensor Edge Mobile では、Color 700 Pocket PC を使用する Symbol 9000-G および Intermec IP3 がサポートされています。

Symbol 9000-G の場合、Sensor Edge Mobile では次の操作がサポートされています。

- [rfid_read\(\)](#)
- [rfid_write\(\)](#)
- [rfid_kill\(\)](#)
- [barcode_read\(\)](#)

Intermec IP3 の場合、Sensor Edge Mobile では次の操作がサポートされています。

- [rfid_read\(\)](#)
- [barcode_read\(\)](#)

ディスクパッチャの場合、構成済のドライバとその現在の状態 (初期化中、アクティブまたは停止) がページに表示されます。

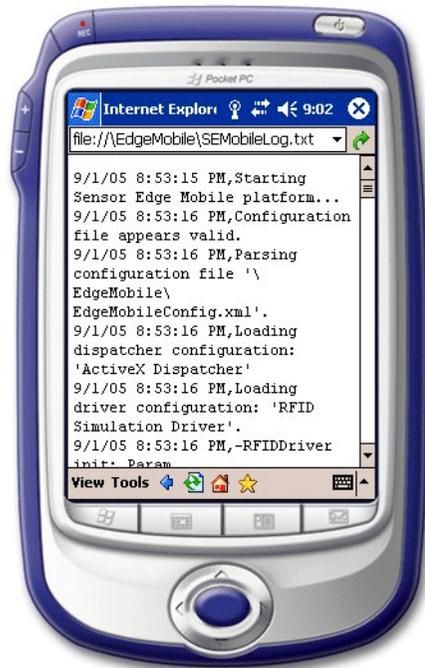
- 実行ステータスの監視
Sensor Edge Mobile のサービス・ステータス（初期化中、アクティブまたは停止）がページに示されます。
- サービス管理
「Shutdown Service」 ボタンがページに表示されます。ドライバの構成、ディスパッチャの変更、または XML 構成ファイル（EdgeMobileConfig.xml）の編集を完了した後は、このボタンを使用します。

図 5-2 Sensor Edge Mobile サービスの停止



- ログイン
ログ・エントリのリアルタイムの更新内容がページに表示されます。

図 5-3 システム・ログの表示



- パフォーマンス監視
ロードされたドライバごとに、測定データ・イベントの数と、各ドライバからの最後のイベントのタイムスタンプがページに表示されます。
- 構成
ページでは、構成 XML ファイル (EdgeMobileConfig.xml) を読み取り、必要に応じてファイルの構成を変更できます。

国際化

国際化のために、Sensor Edge Mobile 内のすべての文字列リテラルが .d11 の中に入っているため、各国で使用できる実行可能ファイルの出荷が可能です。

デバイス、フィルタ・インスタンスおよび ディスクパッチャの構成

この章の内容は次のとおりです。

- 「デバイス、フィルタ・インスタンスおよびディスクパッチャの構成の概要」
- 「デバイスの構成」
- 「フィルタ・インスタンスの構成」
- 「Oracle Sensor Edge Server インスタンス用のディスクパッチャの管理」

デバイス、フィルタ・インスタンスおよびディスパッチャの構成の概要

この章では、Oracle Sensor Edge Server で即時利用できる拡張機能（ドライバ、フィルタおよびディスパッチャ）と、これらのオブジェクトのインスタンスを作成する際（ディスパッチャの場合は、Oracle Sensor Edge Server インスタンスの現行ディスパッチャを設定する際）のパラメータの構成方法について説明します。これらの拡張機能は静的オブジェクトであるため、Oracle Sensor Edge Server インスタンスが拡張機能を使用してイベント・データを処理するためには、拡張機能のインスタンスを作成する必要があります。詳細は、「[Oracle Sensor Edge Server インスタンスのディスパッチャの設定](#)」および「[Oracle Sensor Edge Server で使用するデバイスおよびフィルタの設定](#)」を参照してください。

デバイスおよびディスパッチャの URI パラメータの設定

デバイスおよび現行ディスパッチャの構成の中には、URI パラメータの定義が必要なものもあります。このパラメータは、String 値を必要とし、使用するトランスポートとパラメータを [トランスポート・ライブラリ](#) に指示します。URI パラメータの値は、次の形式で入力します。

```
esc://<transport>?<param1>=<value1>&<param2>=<value2>&...
```

各要素の意味は次のとおりです。

- `<transport>` は、使用するトランスポートの名前です（`transports.xml` ファイル内の `transporttransport@name` の名前と同じ）。
- `<param1>` は、トランスポートのパラメータ値です。
- `<value>` はパラメータの値です。

パラメータは必要なだけいくつでも指定できます。たとえば、TCP/IP を使用してポート 9999 で `test.oracle.com` という名前のサーバーに接続する場合は、URI パラメータの値を次のように入力します。

```
esc://tcp&hostname=test.oracle.com&port=9999
```

9600 ボーでシリアル・ポート COM1 に接続するには、URI パラメータの値を次のように入力します。

```
esc://com?port=COM1&baud=9600
```

IP 1.2.3.4 で TCP ポート 9999 に接続するには、URI パラメータの値を次のように入力します。

```
esc://tcp?hostName=1.2.3.4&port=9999
```

出力をファイルに書き込んで別のファイルから読み取る（テストおよびロギングに便利）には、URI パラメータの値を次のように入力します。

```
esc://file?inputFileName=/tmp/myInFile.txt&outputFileName=/tmp/myOutFile.txt
```

URI パラメータに定義されている値は、トランスポート・タイプに固有です。サポートされている各トランスポート・タイプとそれに関連付けられているパラメータ値を表 6-1 に示します。

表 6-1 サポートされているトランスポート・タイプ

名前	説明	パラメータ値
com	シリアル通信ポート。	port はポートの名前 (COM1、COM2、tty1S など)。 baud はボー・レート (9600、38400、115200 など)。 例: esc://com?port=COM1&baud=9600
tcp	TCP/IP 接続。	hostname は接続先の IP アドレスまたはホスト名 (たとえば、127.0.0.1)。 port はポート番号。 timeout は、有効な接続を待機する数値 (秒) を設定するオプションのパラメータ。 例: esc://tcp?hostName=127.0.0.1&port=9999
stdout	標準の入出力ストリームを使用する。	N/A
file	入出力に使用するファイル。	outFileName は、出力ファイル名を指定するオプションのパラメータ。値を入力しない場合、出力は破棄される。 inFileName は、入力ファイル名を指定するオプションのパラメータ。値を入力しない場合、入力はない。 purgeOnClose は、boolean 値を必要とするオプションのパラメータ。このパラメータを true に設定すると、接続が閉じたときに入力ファイルがパージされる。このパラメータはデータをポーリングする場合に使用する。 例: esc://file?inputFileName=/tmp/myInFile.txt&outputFileName=/tmp/myOutFile.txt
http	HTTP を介して URL にポストする。	URL はポストする宛先の URL。 proxyServer は、使用するプロキシ・サーバーを指定するオプションのパラメータ。 proxyPort は、プロキシのポートを指定するオプションのパラメータ。
ftp	FTP を介してファイルをポストおよび取得する。	hostname は FTP サーバーの名前 user はユーザーの名前。 password はユーザーのパスワード。

ヒント: よく使用するパラメータを [transports.xml](#) ファイル内で定義することもできます。URI パラメータで定義された値はすべて、[transports.xml](#) ファイル内に定義されている値を上書きします。

デバイスの構成

Oracle Sensor Edge Server では、RFID リーダーをサポートするドライバが提供されています (表 6-2 を参照)。

表 6-2 RFID リーダー

ドライバ名	サポートされているデバイス
Alien リーダー・ドライバ	Alien NannoScanner (Model 915MHz、ePC Reader) など、Alien Technology 社の RFID リーダー全部。「 Alien リーダー・ドライバ・ベースのデバイスの構成 」を参照。
BarcodeDriver	シリアル接続またはネットワーク接続を介して標準の ASCII モードで動作するバーコード・リーダー。 「 BarcodeDriver ベースのデバイスの構成 」を参照。
Intermec BRI ドライバ	IF5 Fixed RFID Reader など、BRI (Basic Reader Interface) をサポートする Intermec の RFID リーダー全部。「 Intermec BRI ドライバによりサポートされるデバイスの構成 」を参照。
Intermec リーダー・ドライバ	次のデバイスがある。 <ul style="list-style-type: none"> ■ Intermec IDK (Model ITRF91501) ■ Intermec PCMCIA リーダー (Model ITR100) 「 Intermec リーダー・ドライバ・ベースのデバイスの構成 」を参照。
Matrics ドライバ	Symbol Technologies によりサポートされる固定リーダーには次のものがある。 <ul style="list-style-type: none"> ■ Matrics AR400 リーダー ■ Matrics SR400 リーダー ■ Matrics XR400 リーダー 「 Matrics ドライバ・ベースのデバイスの構成 」を参照。
Samsys ドライバ	SAMSys EPC UHF ロングレンジ・リーダー (Model MP9320 2.7)。 「 Samsys ドライバ・ベースのデバイスの構成 」を参照。
Tyco リーダー	(ThingMagic により駆動される) Tyco Sensormatic Agile2 リーダー。 「 Tyco リーダー・ドライバ・ベースのデバイスの構成 」を参照。

また、Oracle Sensor Edge Server では、プリンタ・デバイスをサポートするドライバと、表示および通知用のドライバも提供されています（表 6-3 を参照）。

表 6-3 プリンタ・ドライバおよび表示 / 通知ドライバ

ドライバ名	サポートされているデバイス
AnimationDriver	デバイス・シミュレーションに使用されるソフトウェア専用ドライバ。 「AnimationDriver のインスタンスの構成」を参照。
ConsoleDriver	デバイス・シミュレーションに使用されるソフトウェア専用ドライバ。 「ConsoleDriver のインスタンスの構成」を参照。
Edge Simulator ドライバ	デバイス・シミュレーションに使用されるソフトウェア専用ドライバ。 「Edge Simulator ドライバ・ベースのデバイスの構成」を参照。
LpmlDriver	次のプリンタ・デバイスがサポートされている。 <ul style="list-style-type: none"> ■ Zebra Technologies R110Xi プリンタ / エンコーダ ■ Intermec EasyCoder PM4i プリンタ ■ (LPML をサポートする) Sato Barcode Printer CL408e ■ Loftware Print Server 「LpmlDriver ベースのデバイスの構成」を参照。
PatliteDriver	Patlite 社（米国）の PHE-3FB PC 制御信号灯。Intermec および積層表示灯のデバイス・コントローラは、Oracle Technology Network (http://www.oracle.com/technology/) からダウンロード可能。 「PatliteDriver ベースのデバイスの構成」を参照。
Prolite ドライバ	Pro-Lite TruColorII LED Display (Model PL-M2014RV6)。 「Prolite ドライバ・ベースのデバイスの構成」を参照。

Oracle Sensor Edge Server には、Edge Echo ドライバも付属しています。Edge Echo ドライバは、物理デバイスを制御するのではなく、命令イベントを受信し、イベントをコピーして送り返します。このイベントには、Edge Echo ドライバ・インスタンスの構成に応じて、元の命令イベントと同じ属性がすべて含まれます。Oracle Sensor Edge Server はこのイベントを、デバイスにより生成された他のイベントと同じように処理します。つまり、イベントはフィルタリングされ、現行ディスパッチャによってディスパッチされます。「Edge Echo ドライバ・ベースのインスタンスの構成」も参照してください。

Alien リーダー・ドライバ・ベースのデバイスの構成

Alien リーダー・ドライバでは、Alien Technology 社の RFID リーダーすべてがサポートされません。

Alien リーダー・ドライバ・ベースのデバイスを構成するには、次のパラメータを定義します。

- **IPAddress:** デバイス・コントローラが実行されているマシンのホスト名または IP アドレス。デバイス・コントローラが Oracle Sensor Edge Server と同じマシンで実行されている場合は、127.0.0.1 と入力。
- **Port Number:** デバイスとの通信に使用するポート番号（デフォルトは 23）。
- **username** および **password**。
- **AntennaSeqIDList:** 各アンテナの識別子のリスト。
- **AntennaMappedDeviceNameList:** 各アンテナに関連付けられたマップ・デバイス名のリスト。

Alien リーダー・ドライバにより生成される測定イベント

測定イベントは、ドライバにより生成され、システムにディスパッチされるイベントです。Alien リーダー・ドライバによって生成されるイベントを表 6-4 に示します。

表 6-4 Alien リーダー・ドライバにより生成される測定イベント

タイプ	サブタイプ	タグ ID	データ	説明
200	各種	タグ ID	タグに格納されるデータ・フィールド（ペイロード）	汎用タグ測定イベント
1	0	NULL	"alive or dead"	データ・プロパティを使用してデバイスが応答しているかどうかを示す各イベント
1	0	NULL	"dis"	デバイス起動操作の正常完了を示すイベント
1	2	NULL	"dif"	デバイス起動操作の失敗を示すイベント
1	0	NULL	"dss"	デバイス停止操作の正常完了を示すイベント
1	2	NULL	"dsf"	デバイス停止操作の失敗を示すイベント
1	2	NULL	"drf"	イベント受信操作の失敗を示すイベント
1	0 または 2	NULL	<callResults> <callResult name="write-tag"> <code>0/1</code> <message/> </callResults>	タグ書込み操作の正常完了または失敗を示すイベント

Alien リーダー・ドライバにより受け入れられる命令イベント

アプリケーションは、特定のタスクの実行を指示する命令イベントをデバイスに送信します。Alien リーダー・ドライバにより受け入れられる命令イベントを表 6-5 に示します。

表 6-5 Alien リーダー・ドライバにより送信される命令イベント

タイプ	サブタイプ	タグ ID	データ	説明
101	0	NULL	<pre><methodCalls> <methodCall name="write-tag"> <params> <param name='tagid'>%TAGID%</param> <param name='tagdata'>%TAGDATA%</param> </params> </methodCall> </methodCalls></pre>	指定されたタグ ID で識別されるタグにデータを書き込む。

AnimationDriver のインスタンスの構成

AnimatorDriver ではデバイスがサポートされるのではなく、デバイスの対話型ソフトウェア・シミュレーションがサポートされます。このドライバの構成済インスタンスは、Oracle Sensor Edge Server からの入力またはイベントを受け入れ、構成ファイル (例 6-1) 内に定義されているアニメーション・シーケンスを画面上で実行します。

AnimationDriver インスタンスを構成するには、fileName パラメータを、構成ファイル (例 6-1 に示されている形式の .xml ファイル) へのフルパスまたは相対パスとして定義します。このファイルを使用すると、任意のタイプのイベントを受け入れて生成するように AnimationDriver インスタンスを構成できます。

例 6-1 AnimationDriver の構成ファイル

```
<animate>
  <window>... </window>
  <task>...</task>
</animate>
```

このファイルを構成するには、<window> 要素と <task> 要素を定義します。

<window> 要素の定義

<window> 要素を使用すると、ウィンドウを定義できます。各ウィンドウは、サーバーの表示で最上位のウィンドウとして表示されます。<window> 要素内には、タスクをトリガーするウィンドウ内のユーザー・インタフェース・コントロールをいくつでも定義できます。

<window> 要素は、ウィンドウとそのディメンションを定義する 3 つの属性 (name、width および height) で構成されます。

- name はウィンドウの一意的な名前前で、ウィンドウのタイトル・バーに表示されます。
- width はウィンドウの幅 (ピクセル) です。
- height はウィンドウの高さ (ピクセル) です。

ウィンドウを定義した後は、ウィンドウ内のコントロールを定義できます。例 6-2 は、1 つのプッシュ・ボタン・コントロールを含むウィンドウを示しています。AnimationDriver では、プッシュ・ボタン用のコントロール・タイプである button がサポートされています。

例 6-2 構成ファイル内でのウィンドウの定義

```
<window name='myWin' width='200' height='100'>
  <control type='button' name='Pass Thru' />
</window>
```

ユーザーがボタンをクリックしたときにトリガーされるタスクを定義するには、<task> 要素を使用します。

<task> 要素

タスクとは、コントロールまたはイベントによりトリガーされるアクションのセットです。

例 6-3 は、子要素 <when> および <action> を持つ <task> 要素を示しています。<when> 要素は、<action> 要素を実行する条件をタスクに指示します。複数のタスクが、重なった条件にマッピングされている場合もあります。

例 6-3 構成ファイル内でのタスクの定義

```
<task name='TaskA'>
  <when> ... </when>
  <action> ... </action>
</task>
```

<when> 要素を定義するには、次の条件を使用します。

- <onClick> は、ユーザーがウィンドウ内のコントロールをクリックしたことに一致します。この条件は 1 つの引数として、イベントのコントロール名を使用します。たとえば、<onClick name='Pass Thru' /> は、ユーザーが Pass Thru という名前のコントロール・ボタンをクリックしたことに一致します (例 6-2 を参照)。
- <onEvent> は、このイベントに対して定義されている基準に一致するイベントがドライバ・インスタンスにより受信された場合に起動されます。イベントには次の基準を定義できます。
 - type: イベントのタイプに一致します。
 - subtype: イベントのサブタイプに一致します。
 - id: ID の完全一致を検索します。
 - ids: ID が一致するイベントをリストから検索します。属性の値は、検索対象の ID をカンマで区切ったリストとして指定する必要があります。
 - data: イベントのデータ・フィールド内の一致を検索します。
- <onInterval> は定期的呼び出されます。この要素を定義するには、frequency 属性を指定します。この属性を設定するには、String 値または数値 (int) を使用して、<onInterval> を呼び出す繰り返し間隔を秒単位で定義します。たとえば、String 値を 1 回使用してこの属性を設定した場合、起動時に 1 回 <onInterval> が呼び出されるように設定されます。

タスクを起動する条件を定義した後は、<action> 要素内に、タスクにより実行される処理を定義できます。<action> 要素内では次の内容を定義できます。

- オーディオ・ファイルを再生する <audio>
- 特定の場所に画像を表示する
- フレームとフレームの時間をずらすための <delay>
- イベントをシステムに送信するための <send>

BarcodeDriver ベースのデバイスの構成

BarcodeDriver ベースのデバイスを構成するには、uri パラメータを積層表示灯への接続として定義します。「[デバイスおよびディスパッチャの URI パラメータの設定](#)」を参照してください。

BarcodeDriver により返される RFID 測定イベント

BarcodeDriver により返される RFID 測定イベントを表 6-6 に示します。

表 6-6 BarcodeDriver により返される RFID 測定イベント

タイプ	サブタイプ	方向	タイトル	説明
200	1	デバイスから (インバウンド)	ID Observed	新しいバーコードが読み取られ、デコードされる。新しいバーコードが測定されると、バーコードはデコードされてイベントのタグ ID フィールドに格納される。

ConsoleDriver のインスタンスの構成

ConsoleDriver は、受信したすべてのイベントをグラフィック・ウィンドウに表示するシミュレーション・ドライバです。これには、次のビューが含まれます。

- **Event List:** 起動後にデバイスが受信したすべてのイベントのリストが表示されます。
- **Details:** 「Event List」ビューに表示されたイベントをダブルクリックすると、そのイベントに固有の詳細がこのウィンドウに表示されます。
- **Send:** イベントをシステムに送信する場合は、このビューを選択します。イベントを送信する前に、TYPE、SUBTYPE および ID フィールドにデータが入力されている（フィールドが空でない）ことを確認してください。TIME フィールドには、現在のタイムスタンプが表示されます。

このドライバは、イベント・タイプおよびデータを認識しません。受信したイベントを表示し、「Send」ダイアログからのユーザー入力を送信するのみです。ユーザー・インタフェースを使用すると、イベント・データを入力し、実際のデバイスで生成された場合と同じように、そのデータを送信してディスパッチできます。このドライバは、システム内のイベント・フローの監視およびデバッグを行う場合に使用します。

Edge Echo ドライバ・ベースのインスタンスの構成

Edge Echo ドライバは、物理デバイスを制御するのではなく、命令イベントを受信し、イベントをコピーして送り返します。このイベントの属性は、現在の時刻を表示するために構成可能な Timestamp 属性を除いて、元の命令イベントの属性と同じです。Oracle Sensor Edge Server は、Edge Echo ドライバのインスタンスにより生成されたイベントを、デバイスにより生成された他のイベントと同じように処理します。つまり、イベントはフィルタリングされ、現行ディスパッチャによってディスパッチされます。「[イベント・データの表示](#)」も参照してください。

Edge Echo ドライバのインスタンスを構成するには、次のパラメータを定義します。

- **stampTime: true** を入力すると、Edge Echo ドライバ・インスタンスにより生成されるイベントの Timestamp 属性が現在の時刻に設定されます。
- **logFileName:** ログ・ファイルの場所を入力します。これは String 値です。

Edge Simulator ドライバ・ベースのデバイスの構成

Edge Simulator ドライバは、イベントを生成して実際のデバイスをシミュレーションします。通常は、Edge Simulator ドライバを使用して構成および配置設計をテストしますが、内部機能テストに使用して、システムでイベントがどのように処理されるかを確認することもできます。Edge Simulator ドライバは他のドライバと同じように機能しますが、物理デバイスに接続してイベントを読み取るのではなく、偽のイベントを生成するタイミングに関する命令を、入力ファイル（例 6-6 など）からパラメータとして受け取る点が異なります。これは、デバイスが起動すると（Oracle Sensor Edge Server の起動時にデバイスが起動すると）すぐに開始されません。

Edge Simulator ドライバ・ベースのデバイスを構成する場合、この入力ファイルの名前を入力することにより、デバイスの FILENAME パラメータを定義することが必要です。このパラメータは、次の命令を使用して偽のイベントを生成する方法を Edge Simulator ドライバ・ベースのデバイスに指示します。

```
<EventList>
```

<EventList> 要素はループを定義します。この要素は、その他のすべての命令をまとめるメイン・ブロックでもあります。<EventList> には repeat という属性が 1 つあり、これはループの制御に必要です。repeat の値は、0 から LONG_MAX の間の数値である必要があります。イベントを 1 回のみ生成する場合は、repeat 属性を 1 に設定します。repeat を n に設定すると、すべての命令が n 回ループされます。repeat を 0 に設定すると、ブロックが使用不可になり、パーサーはそのブロックをスキップします。

2 つのイベントを生成し、一時停止してから、さらに 2 つのイベントを生成するという一連の処理を 20 回繰り返す構文を例 6-4 に示します。

例 6-4 ループの定義

```
<EventList repeat='20'>
<Event> ... </Event>
<Event> ... </Event>
<EventInterval>...</ EventInterval>
<Event> ... </Event>
<Event> ... </Event>
</EventList>
```

<EventList> 要素内には、任意の数の命令を含めることができます。定義された順序で命令が実行されます。

```
<EventInterval>
```

<EventInterval> 要素は、次に進む前に一定の時間、一時停止するようにシミュレータに指示します。通常はデータ率を調節するために使用されます。次の命令を実行する前に待機する時間をミリ秒単位で定義する数値です。6-5 項は、各イベントの間に 0.5 秒、ループの間に 3 秒待機するようにシミュレータに命令する方法を示しています。

例 6-5 <EventInterval> 要素

```
<EventList repeat='20'>
  <Event> ... </Event>
  <EventInterval>500</ EventInterval>
  <Event> ... </Event>
  <EventInterval>500</ EventInterval>
  <Event> ... </Event>
  <EventInterval>3000</ EventInterval>
</EventList>
<Event>
```

<Event> 要素は、イベントを送信するようにシミュレータに指示します。子要素（表 6-7 を参照）はイベントのフィールドを制御します。

表 6-7 シミュレータのイベント要素

イベント・フィールド	値
<type>	イベントのタイプに対応する数値。
<subtype>	サブタイプの数値。たとえば、例 6-6 のサブタイプ値は一般的な命令イベント（特定のデバイスに処理を実行するように指示するために、アプリケーションまたはデバイスによって送信されたイベント）に対応する。例 6-6 では、1 という値がデバイスに指定されている。
<id>	このフィールドのテキスト値により、イベント命令に対するタグ（読取りまたはターゲット）が識別される。例 6-6 では、タグの <id> の値の 1 つが 03ffff045679 となっている。
<data>	タグ・データ。オプションのフィールド。
<deviceName>	イベントを生成するデバイスまたはアプリケーションの名前。<deviceName> により、イベントの生成時にシミュレータを別のデバイスのように見せることができる。

2 つのイベント・グループを含む入力ファイルを例 6-6 に示します。1 つ目のイベントは 1 回だけ実行され、2 つ目のイベントは 20 回実行されます。

例 6-6 シミュレータ入力ファイル

```
<EdgeEventSimulation>
  <EventList repeat='1'>
    <Event>
      <type>100</type>
      <subtype>1</subtype>
      <id>03ffff045679</id>
      <data>No Data</data>
      <deviceName>My Device</deviceName>
    </Event>
  </EventList>
  <EventInterval>500</ EventInterval>
  <Event>
    <type>100</type>
    <subtype>1</subtype>
    <id>03ffff045680</id>
    <data>No Data</data>
    <deviceName>My Device</deviceName>
  </Event>
  <EventInterval>3000</ EventInterval>
</EventList>

<EventList repeat='20'>
  <Event>
    <type>100</type>
    <subtype>1</subtype>
    <id>04ffff045679</id>
    <data>No Data</data>
    <deviceName>My Device</deviceName>
  </Event>
  <EventInterval>500</ EventInterval>
  <Event>
    <type>100</type>
    <subtype>1</subtype>
    <id>04ffff045680</id>
    <data>No Data</data>
    <deviceName>My Device</deviceName>
  </Event>
```

```

    </Event>
  </EventListener>
</EdgeEventSimulation>

```

イベント・タイプの形式は固定ですが、イベントのタイプに応じてフィールドを異なる意味にマッピングすることにより、イベント・タイプを拡張できます。

HtmlDriver インスタンスの構成

HtmlDriver は、Oracle Sensor Edge Server 画面にウィンドウを表示したり、リモート・サーバーに接続して HTML コンテンツを表示することを可能にする表示ドライバです。複雑なステータスやメッセージをグラフィック表示するためにデータをプッシュするアプリケーションには、HtmlDriver を使用します。

HtmlDriver には表示サーバーが含まれており (htmlDriver.jar に含まれる)、これを使用するとリモート・サーバー上の表示ウィンドウが使用可能になります。表示サーバーは、HtmlDriver インスタンスからのネットワーク接続を待機する Java プロセスとして実行されます。ネットワーク上には任意の数の表示サーバーを設定できます。表示サーバー・プロセスを実行するマシンの IP またはホスト名を覚えておく必要があります。同じマシン上で複数の表示サーバーを実行できます。ただし、各表示サーバーが別々のポートに割り当てられている必要があります。

注意： JDK 1.3 以上がマシンにインストールされており、java.exe がパスに存在している必要があります。

表示サーバーをマシンにインストールするには、次のようにします。

1. htmlDriver.jar をコピーします。
2. ファイルをローカル・ディレクトリに抽出します。たとえば、displayServer という名前のディレクトリを作成し、jar xf htmlDriver.jar を実行します。
3. classes ディレクトリに移動します (cd classes)。
4. 次のコマンドを使用して DisplayServer アプリケーションを実行します。

```
java oracle.edge.impl.driver.HtmlDriver <port> [title] [x] [y]
[width] [height] [showHistory]
```

各要素の意味は次のとおりです。

- <port> は、表示サーバーがリスニングする TCP/IP ポートを指定します。すでに使用されているポートは指定できません。これは必須パラメータです。
- [title] は、ウィンドウ名を指定するオプションのテキスト文字列です。
- [x] は、画面上の表示ウィンドウの x 位置です。
- [y] は、画面上の表示ウィンドウの y 位置です。
- [width] はウィンドウの幅です。0 に設定した場合、ウィンドウでは画面の幅が使用されます。
- [height] はウィンドウの高さです。0 に設定した場合、ウィンドウでは画面の高さが使用されます。
- [showHistory]: true に設定した場合、ウィンドウの一番上に小さなリストが表示され、以前に表示されたすべてのドキュメントが示されます。このウィンドウを使用すると、ユーザーは最後に表示されたドキュメントまで手動でスクロールできます。

たとえば、全画面ウィンドウで、ポート 8999 を使用してデスクトップ上で表示サーバーを実行する場合は、次のコマンドを実行します。

```
java oracle.edge.impl.driver.HtmlDriver 8999 "My Window" 0 0 0 0 false
```

注意: このコマンドには、スペースが付いた引用符文字列が含まれます。シェルのドキュメントを参照してください。

リモート・サーバーでの表示が使用可能になるように `HtmlDriver` インスタンスを構成するには、次のパラメータを定義します。

- `serverlist` パラメータは、このドライバ・インスタンスが接続できる表示サーバーをカンマで区切ったリストを入力することにより定義します。このパラメータの `String` 値の形式は次のとおりです。

```
<hostname or IP>:<port>,...
```

セキュリティを設定するには、各表示サーバーをリストする必要があります。

- 表示サーバーがリスニングする `TCP/IP` ポートを入力します。すでにマシン上で使用されているポートは指定できません。

`serverList` パラメータを定義しない場合、インスタンスは `Oracle Sensor Edge Server` のローカル・ウィンドウに表示されます。ローカル・ウィンドウでインスタンスを実行する場合、表示サーバーは必要ありません。

ローカル・ウィンドウでインスタンスを表示するには、次のようにします。

- まず `title` フィールドにウィンドウ名を入力することにより表示ウィンドウを定義してから、`ShowHistory` フィールドに `boolean` (`true` または `false`) を入力して表示対象ページのウィンドウを表示します。
- 次のパラメータに `int` 値を入力して、ウィンドウのディメンションを設定します。
- `x`: 画面上のウィンドウの `x` 位置 (ピクセル)。
- `y`: 画面上のウィンドウの `y` 位置 (ピクセル)。
- ウィンドウの幅および高さをピクセルで入力します。0 を入力すると、ウィンドウは画面の高さおよび幅に設定されます。

HtmlDriver によりサポートされるイベント

`HtmlDriver` によりサポートされるイベントを表 6-8 に示します。

表 6-8 `HtmlDriver` によりサポートされるイベント

タイプ	サブタイプ	方向	タイトル	説明
210	1	アプリケーションから (アウトバウンド)	Display	イベントの <code>DATA</code> フィールドに組み込まれた <code>HTML</code> ドキュメントを表示する。ID は 0 から始まる番号で、イベントの送信先のサーバーを定義する。これは、ドライバ・インスタンスの <code>serverList</code> パラメータ内で定義されているサーバーの索引である。
210	2	アプリケーションから (アウトバウンド)	Print LPML Directly	イベントの <code>DATA</code> フィールドに指定されている URL により参照される <code>HTML</code> ドキュメントを表示する。ID は 0 から始まる番号で、イベントの送信先のサーバーを定義する。これは、ドライバの <code>serverList</code> パラメータ内で定義されているサーバーの索引である。

Intermec BRI ドライバによりサポートされるデバイスの構成

Intermec BRI ドライバでは、BRI (Basic Reader Interface) をサポートし、シリアル接続またはネットワーク (TCP/IP) 接続を介して Oracle Sensor Edge Server と通信する Intermec 社のすべての RFID リーダーがサポートされます。Oracle Sensor Edge Server は、IF5 リーダーに対してテスト済です。

詳細は、<http://www.intermec.com> を参照してください。

Intermec BRI ドライバ・ベースのデバイスを構成するには、表 6-9 に示されたパラメータを定義する必要があります。

表 6-9 Intermec BRI ドライバ・ベースのデバイスのパラメータ値

パラメータ名	値のタイプ	説明
URI	String	接続タイプを示す接続識別子と接続パラメータについては、「 デバイスおよびディスプレイの URI パラメータの設定 」を参照。
Status Query Response Timeout	Int	ステータス問合せからのレスポンスを待機するためのタイムアウト。タイムアウトには、Status Query Interval パラメータに設定した値よりも小さい値を設定する必要があります。
Status Query Interval	Int	連続する 2 つのステータス問合せの間隔 (ミリ秒単位)。
AntennaIds	String	接続されているアンテナの識別子。アンテナが複数ある場合、複数の識別子カンマ (,) を使用して区切る。
DataFieldLength	Int	取得されるデータ・フィールドのサイズ (バイト)。0 は、データ・フィールドを取得する必要がないことを示す。
DataFieldValueType	String	データ・フィールドに対して次の 3 つのいずれかの値を指定する。 <ul style="list-style-type: none"> ■ Int (整数) ■ HexString (16 進文字列) ■ String (ASCII 文字列) このパラメータは、DataFieldLength パラメータが 0 より大きい場合のみ有効になる。
TagType	N/A	読み取るタグ・タイプを指定する。次のオプションがある。 <ul style="list-style-type: none"> ■ ISO6B/G1 ■ ISO6B/G2 ■ ISO6C ■ EPC/Class/Gen1 ■ EPC/Class1/Gen2 ■ Phillips/V1.19 ■ 混合

Intermec BRI ドライバにより生成される測定イベント

測定イベントは、ドライバにより生成され、システムにディスパッチされるイベントです。Intermec BRI ドライバによって生成されるイベントを表 6-10 に示します。

表 6-10 Intermec BRI ドライバにより生成されるイベント

タイプ	サブタイプ	タグ ID	データ	説明
200	各種	タグ ID	タグに格納されるデータ・フィールド (ペイロード)	汎用タグ測定イベント
1	0	NULL	"alive or dead"	データ・プロパティを使用してデバイスが応答しているかどうかを示すイベント
1	0	NULL	"dis"	デバイス起動操作の正常完了を示すイベント
1	2	NULL	"dif"	デバイス起動操作の失敗を示すイベント
1	0	NULL	"dss"	デバイス停止操作の正常完了を示すイベント
1	2	NULL	"dsf"	デバイス停止操作の失敗を示すイベント
1	2	NULL	"drf"	イベント受信操作の失敗を示すイベント
1	0 または 2	NULL	<callResults> <callResult name="write-tag"> <code>0/1</code> <message/> </callResults>	タグ書き込み操作の正常完了または失敗を示すイベント

Intermec BRI ドライバにより受け入れられる命令イベント

アプリケーションは、特定のタスクの実行を指示する命令イベントをデバイスに送信します。Intermec BRI ドライバにより受け入れられる命令イベントを表 6-11 に示します。

表 6-11 Intermec BRI ドライバにより送信される命令イベント

タイプ	サブタイプ	タグ ID	データ	説明
101	0	NULL	<methodCalls> <methodCall name="write-tag"> <params> <param name='tagid'>%TAGID%</param> <param name='tagdata'>%TAGDATA%</param> Name='tagdataVauleType' (String/HexString) </param> </params> </methodCall> </methodCalls>	指定されたタグ ID で識別されるタグにデータを書き込む。

Intermec リーダー・ドライバ・ベースのデバイスの構成

詳細は、<http://www.intermec.com> を参照してください。

要件

Intermec リーダー・ドライバ・ベースのデバイスを作成する場合、バンドル化されて Intermec リーダー・ドライバに付属している次のコンポーネントが必要です。

- IntelliTag IDK

IntelliTag IDK (IDK) は Intermec がサポートするソフトウェア・ライブラリとツールのセットです。Intermec デバイスとの通信ではこのライブラリを使用する方法だけが採用されており、Windows 32 プラットフォーム (Windows 2000 および Windows XP) でのみサポートされます。IntelliTag IDK は次のサイトから入手できます。

http://www.oracle.com/technology/products/iaswe/edge_server/extensions.html

- Intermec リーダー・ドライバ

Intermec リーダー・ドライバ・ベースのデバイスを構成する場合、次のパラメータを定義する必要があります。

- IPAddress をホスト名または IP アドレスに設定します。デバイスが Edge Server と同じマシンで実行されている場合は、127.0.0.1 と入力します。
- PortNo をポート番号に設定します。デフォルトは 6666 です。
- AntennaSeqIDList を各アンテナの識別子のリストに設定します。
- AntennaMappedDeviceNameList を、各アンテナに関連付けられたマップ・デバイス名のリストに設定します。

LpmlDriver ベースのデバイスの構成

LPML (Label Printing Markup Language) は、オラクル社が定義した XML ベースのプリンタ言語であり、ラベル・プリンタおよびプリンタ・サーバーのベンダーによってサポートされています。LpmlDriver では、LPML 言語およびバインディングをサポートするすべてのプリンタまたは印刷サーバー・システムがサポートされます。

LpmlDriver ベースのデバイスを構成するには、次のようにデバイスのパラメータを定義する必要があります。

- URI: トランスポート・レイヤーがプリンタに接続するために使用する文字列。「[デバイスおよびディスパッチャの URI パラメータの設定](#)」を参照してください。
- 「**Response Mode**」を選択した場合、デバイスは、通信先のプリンタで LPML レスポンス・メッセージがサポートされていると予測します。このため、デバイスは印刷ジョブごとにプリンタからのレスポンスを待機します。
- 「**Batch Mode**」を選択した場合、レスポンスを待機する前に、1 つの接続セッションで可能なかぎり多くのジョブを送信するようにプリンタに命令します。このオプションを使用すると、デバイスは、プリンタとの半 2 重通信モードではなくウィンドウ・アルゴリズムを使用できます。
- 「**Batch Mode**」を選択した場合は、ReadTimeout パラメータを、プリンタがレスポンスに失敗したとみなさずに待機する時間 (ミリ秒単位) に設定します。
- maxRetryCount: 印刷ジョブを再試行する回数。これは int 値です。
- PerJobDelay: 印刷ジョブと印刷ジョブの時間をずらす間隔 (秒)。これは int 値です。

LpmlDriver により生成されるプリンタ・レスポンス測定イベント

LpmlDriver により生成され、システムにディスパッチされるプリンタ・レスポンス測定イベントを表 6-12 に示します。

表 6-12 LpmlDriver により生成されるプリンタ・レスポンス測定イベント

タイプ	サブタイプ	タグ ID	データ	説明
210	2	使用されない	LPML レスポンス・メッセージ	印刷ジョブの結果を送信側に通知する、プリンタから (「Response Mode」が選択されている場合)、またはドライバそのものからのレスポンス・メッセージ。correlationId フィールドと sourceName フィールドは、元の印刷ジョブ・リクエストから使用される。

LpmlDriver により受け入れられる命令イベント

命令イベントはアプリケーションによってドライバに送信され、ドライバは特定のタスクを実行するように指示されます。LpmlDriver により受け入れられる命令イベントを表 6-13 に示します。

表 6-13 LpmlDriver により受け入れられる命令イベント

タイプ	サブタイプ	タグ ID	データ	説明
102	1	使用されない	LPML メッセージ	Data フィールドに指定された LPML メッセージを印刷する。

Matrics ドライバ・ベースのデバイスの構成

Matrics ドライバでは、シリアル接続またはネットワーク (TCP/IP) 接続を使用して Oracle Sensor Edge Server と通信する、Symbol Technologies が作成した Matrics リーダーがサポートされます。

Matrics ドライバ・ベースのデバイスを構成する場合、次のパラメータを定義する必要があります。

- URI: 接続タイプおよび接続識別子。「[デバイスおよびディスパッチャの URI パラメータの設定](#)」を参照してください。
- Status Query Interval: 連続するステータス問合せの時間間隔 (ミリ秒単位)。これは int 値です。
- Status Query Response Timeout: ステータス問合せからのレスポンスを待機するためのタイムアウトの時間 (ミリ秒単位)。この int 値は、Status Query Interval パラメータに設定した値よりも小さい値にする必要があります。
- AntennaIds: 接続されているアンテナ。複数のエントリはカンマ (,) を使用して区切ります。
- Node Address: RS485 に基づいて接続されているリーダーのノード・アドレス。このアドレスは、Matrics 管理コンソールでの値と一致している必要があります。

Matrics ドライバにより生成される測定イベント

測定イベントは、ドライバにより生成され、システムにディスパッチされるイベントです。Matrics ドライバによって生成されるイベントを表 6-14 に示します。

表 6-14 Matrics ドライバにより生成される測定イベント

タイプ	サブタイプ	タグ ID	データ	説明
200	各種	タグ ID	タグに格納されるデータ・フィールド (ペイロード)	汎用タグ測定イベント
1	0	NULL	"alive or dead"	データ・プロパティを使用してデバイスが応答しているかどうかを示すイベント
1	0	NULL	"dis"	デバイス起動操作の正常完了を示すイベント
1	2	NULL	"dif"	デバイス起動操作の失敗を示すイベント
1	0	NULL	"dss"	デバイス停止操作の正常完了を示すイベント
1	2	NULL	"dsf"	デバイス停止操作の失敗を示すイベント
1	2	NULL	"drf"	イベント受信操作の失敗を示すイベント
1	0 または 2	NULL	<callResults> <callResult name="write-tag"> <code>0/1</code> <message/> </callResults>	タグ書き込み操作の正常完了または失敗を示すイベント

Matrics ドライバにより受け入れられる命令イベント

アプリケーションは、特定のタスクの実行を指示する命令イベントをデバイスに送信します。Matrics ドライバにより受け入れられる命令イベントとその使用方法を表 6-15 に示します。

表 6-15 Matrics ドライバにより受け入れられる命令イベント

タイプ	サブタイプ	タグ ID	データ	説明
101	0	NULL	<methodCalls> <methodCall name="write-tag"> <params> <param name='tagid'>%TAGID%</param> <param name='tagdata'>%TAGDATA%</param> </params> </methodCall> </methodCalls>	指定されたタグ ID で識別されるタグにデータを書き込む。

PatliteDriver ベースのデバイスの構成

PatliteDriver によりサポートされる Patlite の積層表示灯シリーズは、イベントを生成しませんが、信号灯および信号として機能します。Patlite 積層表示灯およびツリーにイベントを送信すると、ライトが点灯するか、または一定間隔で点滅します。

PatliteDriver ベースのデバイスを構成するには、URI パラメータを定義します。「[デバイスおよびディスパッチャの URI パラメータの設定](#)」を参照してください。

Prolite ドライバ・ベースのデバイスの構成

Prolite ドライバ・ベースのデバイスを構成するには、URI パラメータを、接続 URI を入力することにより定義します。「[デバイスおよびディスパッチャの URI パラメータの設定](#)」を参照してください。

Samsys ドライバ・ベースのデバイスの構成

Samsys ドライバでは、シリアル接続またはネットワーク (TCP/IP) 接続を介して Oracle Sensor Edge Server と通信する SAMSys MP320 2.7 EPC リーダーがサポートされます。Samsys MP9320 2.7 EPC リーダーは、CHUMP (Cookie-handling UDP Message Protocol) をサポートしており、Oracle Edge Sensor Server に対してテスト済です。

Samsys ドライバ・ベースのデバイスを構成するには、次のパラメータを定義します。

- **Connection Identifier:** 接続タイプおよび接続識別子。「[デバイスおよびディスパッチャの URI パラメータの設定](#)」を参照してください。
- **Status Query Interval:** 連続するステータス問合せの時間間隔 (ミリ秒単位) を入力します。これは int 値です。
- **Status Query Response Timeout:** ステータス問合せからのレスポンスを待機するためのタイムアウトの時間。この int 値は、Status Query Interval パラメータに設定した値よりも小さい値にする必要があります。
- **Tag Scan Mode:** 自律型かポーリングかを指定します。自律型モードを選択すると、タグをスキャンするたびにデバイスに通知するように MP9320 2.7 EPC リーダーが設定されます。ポーリング・モードを選択すると、ドライバが `get-tag-list` コマンドを送信した場合にのみタグをスキャンするように MP9320 2.7 EPC リーダーが設定されます。
- **Tag Type:** デバイスが読み取るタグのタイプ (EPC1、EPC0、IS18gA、IS186B、STG など) を指定します。
- **AntennaIds:** 接続されているアンテナ。複数のエントリはカンマ (,) を使用して区切ります。

Samsys ドライバにより生成される測定イベント

測定イベントは、ドライバにより生成され、システムにディスパッチされるイベントです。Samsys ドライバによって生成されるイベントを表 6-16 に示します。

表 6-16 Samsys ドライバにより生成される測定イベント

タイプ	サブタイプ	タグ ID	データ	説明
200	各種	タグ ID	タグに格納されるデータ・フィールド (ペイロード)	汎用タグ測定イベント
1	0	NULL	"alive or dead"	データ・プロパティを使用してデバイスが応答しているかどうかを示す各イベント
1	0	NULL	"dis"	デバイス起動操作の正常完了を示すイベント
1	2	NULL	"dif"	デバイス起動操作の失敗を示すイベント
1	0	NULL	"dss"	デバイス停止操作の正常完了を示すイベント
1	2	NULL	"dsf"	デバイス停止操作の失敗を示すイベント
1	2	NULL	"drf"	イベント受信操作の失敗を示すイベント
1	0 または 2	NULL	<callResults> <callResult> <callResult name="write-tag"> <code>0/1</code> <message/> </callResult> </callResults>	タグ書き込み操作の正常完了または失敗を示すイベント

Samsys ドライバにより受け入れられる命令イベント

アプリケーションは、特定のタスクの実行を指示する命令イベントをデバイスに送信します。Samsys ドライバにより受け入れられる命令イベントを表 6-17 に示します。

表 6-17 Samsys ドライバにより受け入れられる命令イベント

タイプ	サブタイプ	タグ ID	データ	説明
101	0	NULL	<methodCalls> <methodCall name="write-tag"> <params> <param name='tagid'>%TAGID%</param> <param name='tagdata'>%TAGDATA%</param> </params> </methodCall> </methodCalls>	指定されたタグ ID で識別されるタグにデータを書き込む。

Simple Audio ドライバ・インスタンスの構成

SimpleAudioDriver は命令イベントを受け入れ、アプリケーションが Oracle Sensor Edge Server マシンのサウンド・カードを使用して .wav ファイルを再生できるようにします。セキュリティ上の理由から、(audioFiles パラメータ内で定義されている) 特定のディレクトリ内のファイルのみが再生されます。

注意: このドライバでは、標準の JRE または JDK に付属の `sun.audio` パッケージが使用されます。このパッケージでは、1つのマシンで複数のサウンド・カードを使用する場合など、特殊な設定を行う場合にのみ特殊な構成が必要です。このような場合は、Sun Microsystems 社の JDK ドキュメントを参照してください。

Simple Audio ドライバ・インスタンスを構成するには、オーディオ・ファイルのパス (String 値) を入力することにより、audioFiles パスのパラメータを定義します。パスは、最後にスラッシュの付かない絶対パスまたは相対パスです。パスはエッジ拡張クラス・ディレクトリを基準にしているため、相対パスを入力する場合は、このディレクトリから横断する必要があります。たとえば、Oracle Sensor Edge Server ホーム・パス (サンプル・レベル、config や persistent などのディレクトリのロケール) 上のオーディオ・ファイルを指定する場合、次のように入力します。

```
..¥..¥edge¥audio
```

この値により、Oracle Sensor Edge Server ホーム・パスの最上位の audio ディレクトリを使用するようにパラメータが設定されます。

ヒント: 絶対パスを使用するほうが、相対パスを使用するよりもセキュリティは強くなります。

Simple Audio ドライバによりサポートされるオーディオ・イベント

Simple Audio ドライバによりサポートされるイベント・タイプ 207 (オーディオ) を表 6-18 に示します。

表 6-18 イベント・タイプ 207 (オーディオ)

タイプ	サブタイプ	方向	タイトル	説明
207	1	アプリケーションから (アウトバウンド)	Play Audio File	Data フィールドの .xml ファイル内に定義されているオーディオ・ジョブを再生する。

DATA フィールドに組み込まれている XML ファイルは、標準の XML-RPC コール形式を使用します。その形式は例 6-7 に示すとおりです。

例 6-7 Data フィールドに組み込まれた XML ファイル

```
<methodCalls>
  <methodCall>
    <params>
      <param name='paramName'>paramValue</param>
    </params>
  </methodCall>
</methodCalls>
```

paramName はパラメータ名で、paramValue はその値です。この関数はタイプおよびサブタイプにより定義されるため、<methodName> は省略できます。paramName は、fileName および url を使用して定義できます。

セキュリティ上の理由で、fileName パラメータにより定義された値がオーディオ・ファイル名として使用されます。fileName の値に拡張子を含めることはできません。また、オーディオ・ファイルは .wav ファイルである必要があります。たとえば、(例 6-8 のように)

fileName パラメータの値を welcome として定義した場合、オーディオ・ファイル名は welcome.wav になります。

例 6-8 welcome.wav を再生する XML ファイル

```
<methodCalls>
  <methodCall>
    <params>
      <param name=fileName'>welcome</param>
    </params>
  </methodCall>
</methodCalls>
```

url パラメータは、オーディオ・ファイルを指す URL です。

注意： fileName パラメータと url パラメータは相互に排他的です。つまり、同じコールに対して両方のパラメータを定義することはできません。両方のパラメータを定義した場合は、url パラメータのみが使用されます。

Tyco リーダー・ドライバ・ベースのデバイスの構成

Tyco リーダー・ドライバでは、シリアル接続またはネットワーク (TCP/IP) 接続を介して Oracle Sensor Edge Server と通信する、Sensormatic が作成したすべての Tyco リーダーがサポートされます。

Tyco リーダー・ドライバ・ベースのデバイスを構成する場合、次のパラメータを定義する必要があります。

- **Connection Identifier:** 接続タイプおよび接続識別子。「[デバイスおよびディスパッチャの URI パラメータの設定](#)」を参照してください。
- **Data Collection Timeout:** 物理デバイスに対してデータを収集するためのタイムアウト (ミリ秒単位)。これは int 値です。
- **Status Query Interval:** 連続するステータス問合せの時間間隔 (ミリ秒単位)。これは int 値です。

Tyco リーダー・ドライバにより生成される測定イベント

測定イベントは、ドライバにより生成され、システムにディスパッチされるイベントです。Tyco リーダー・ドライバによって生成されるイベントを表 6-19 に示します。

表 6-19 Tyco リーダー・ドライバにより生成される測定イベント

タイプ	サブタイプ	タグ ID	データ	説明
200	各種	タグ ID	タグに格納されるデータ・フィールド (ペイロード)	汎用タグ測定イベント
1	0	NULL	"alive or dead"	データ・プロパティを使用してデバイスが応答しているかどうかを示す各イベント
1	0	NULL	"dis"	デバイス起動操作の正常完了を示すイベント
1	2	NULL	"dif"	デバイス起動操作の失敗を示すイベント
1	0	NULL	"dss"	デバイス停止操作の正常完了を示すイベント
1	2	NULL	"dsf"	デバイス停止操作の失敗を示すイベント
1	2	NULL	"drf"	イベント受信操作の失敗を示すイベント
1	0 または 2	NULL	<pre><callResults> <callResult> <callResult name="write-tag"> <code>0/1</code> </callResult> </callResults></pre>	タグ書き込み操作の正常完了または失敗を示すイベント

Tyco リーダー・ドライバにより受け入れられる命令イベント

アプリケーションは、特定のタスクの実行を指示する命令イベントをデバイスに送信します。Tyco リーダー・ドライバにより受け入れられる命令イベントを表 6-20 に示します。

表 6-20 Tyco リーダー・ドライバにより受け入れられる RFID 命令イベント

タイプ	サブタイプ	タグ ID	データ	説明
101	0	NULL	<pre><methodCalls> <methodCall name="write-tag"> <params> <param name='tagid'>%TAGID%</param> <param name='tagdata'>%TAGDATA%</param> <param name='antennaID'>%ANTENNA_ID%</param> </params> <param name='protocol'>EPCO/CC915/ CC1356/ISO15693/ISO18000-6B</param> </params> </methodCall> </methodCalls></pre>	指定されたタグ ID で識別されるタグにデータを書き込む。

フィルタ・インスタンスの構成

次の各項では、組み込みフィルタでイベントおよびその構成パラメータを生成する方法について説明します。

- 「[Check Tag ID フィルタの構成](#)」
- 「[Cross-Reader Redundant フィルタの使用](#)」
- 「[Debug フィルタの使用](#)」
- 「[JavaScript フィルタの構成](#)」
- 「[Movement フィルタの構成](#)」
- 「[Pallet Pass Thru フィルタの構成](#)」
- 「[Pallet Shelf フィルタの構成](#)」
- 「[Pass フィルタの構成](#)」
- 「[Polygon フィルタの構成](#)」
- 「[Regex フィルタの定義](#)」
- 「[Shelf フィルタの構成](#)」

Check Tag ID フィルタの構成

チェック・タグは、デバイス（この場合はリーダー）がタグを読み取っているかどうかをテストするために使用する一般的なタグです。チェック・タグはリーダーの読取り範囲内に物理的に存在するため、必ず読み取られます。その他のタグがリーダーの読取り範囲を通過すると、それに連動してデバイスがチェック・タグも読み取ります。

Check Tag ID フィルタは、デバイスがチェック・タグを読み取っていることを定期的に確認します。このフィルタを使用すると、デバイスのステータス、対応するリーダーおよび使用されるアンテナを確認できます。Check Tag ID フィルタは診断目的のみで使用されるため、クライアント・デバイスにディスパッチするイベントは提供しません。かわりに、指定された時間内にデバイスがチェック・タグを読み取ったことを検出しなかった場合は、このフィルタがイベントを生成します。

注意： Check Tag ID フィルタはデバイスにのみ適用できます。

Check Tag ID フィルタのパラメータ（および関連する値）を表 6-21 に示します。

表 6-21 Check Tag ID フィルタのパラメータ

名前	値のタイプ	説明
Check Tag Id	String 値。	チェック・タグの ID。タグが読み取られているかどうかを確認するためにフィルタが検索する ID。
Tag Check Time Window	int 値。	フィルタが指定されたチェック・タグを検出できなかった場合にイベントが生成されるまでの時間（ミリ秒単位）。

Check Tag ID フィルタのパラメータを定義するには、(デバイスの読取り範囲内にある) チェック・タグ自体の ID を確認する必要があります。この ID を、Check Tag Id の String 値として入力します。Tag Check Time Window パラメータで定義された時間内に、指定された ID を持つチェック・タグをデバイスが読み取ったことを検知しないと、フィルタはイベントを生成します。生成されるイベントのシグネチャを表 6-22 に示します。

表 6-22 Check Tag ID フィルタのイベント・シグネチャ

イベント・フィールド	値
sourceName	このフィールドは、イベントの発信元を識別する。このフィールドはオプションで、値はクライアントによって設定される。
correlationId	このフィールドはクライアントによって設定され、特定のクライアントへのメッセージ・レスポンス (デバイスが機能しているかどうかのチェックなど) に使用される。同じクライアントによって返されたメッセージは同じ ID を持つ。オプションのフィールド。
siteName	このイベントを生成したサイトの名前。
deviceName	このイベントを生成したデバイスの名前。
time	フィルタがこのイベントを生成した時刻。
type	メッセージ・イベント (タイプ 1)。 注意: デバイスにいつ障害が発生したかを通知する場合は、アプリケーションをこのメッセージ・タイプにサブスクライブする必要がある。
subtype	エラー・レポート (サブタイプ 2)。
id	チェック・タグの ID (フィルタの Check Tag Id パラメータ内に定義されている値)。
data	追加のエラー・メッセージ (ある場合)。

Cross-Reader Redundant フィルタの使用

Cross-Reader Redundant フィルタは、デバイス・グループのデバイスから送信された冗長なイベントをブロックし、イベントは生成しません。同じタグ ID を持っているイベントが見つかった場合、このフィルタはイベントが冗長であるとみなします。フィルタリングは、ドライバ読取りサイクルに対応するウィンドウ時間中に発生します。

Cross-Reader Redundant フィルタは、グループ・レベルのフィルタリングにのみ使用できません。デバイスに適用すると、機能が実行されません。このフィルタには、構成するパラメータがありません。

Debug フィルタの使用

Debug フィルタは、システムを通過するイベントを追跡します。関連付けられたデバイスからイベントを受信すると、このフィルタはログ・ファイルにイベントを書き込みます。このフィルタのパラメータは、Event Output File というパラメータ 1 つです。このパラメータを定義するには、Debug フィルタがイベントを書き込むログ・ファイルのフルパスを入力します。(サーバー上でこのファイルを書き込み可能にする必要があります。) Debug フィルタの出力形式は次のとおりです。

```
"Devicename: <devicename> Type: <type> Subtype: <subtype> EventTime:
<time>TagIds:<tagid(,tagid)*>Data:<dat(,data)*>¥n"
```

各イベントは個別の行になります。各行は改行文字 (オペレーティング・システムによって LF または CRLF) で区切られます。<time> の値は、time(2) のコールによって返された long です。

このフィルタは、デバイスまたはデバイス・グループにのみアタッチできます。このフィルタには、構成するパラメータがありません。

JavaScript フィルタの構成

JavaScript フィルタを使用すると、スクリプト言語でフィルタ・ロジックを記述できます。ソース・スクリプトの変更内容は動的にロードされるため、サーバーやその他のコンポーネントを再起動する必要はありません。このフィルタは、Mozilla Rhino (<http://www.mozilla.org/rhino/>) など、ソース・スクリプトを実行する外部の JavaScript エンジンに依存しています。

JavaScript フィルタのパラメータは、`scriptFileName` というパラメータ 1 つです。このパラメータを定義するには、ドライバがロードおよび監視するソース・スクリプトの相対パスまたはフルパスの名前を入力します。

ソース・スクリプトの記述

JavaScript フィルタは、処理を実行するためにソース・スクリプト内で関数を検索します。関数は次のとおりです。

- `function filter(events)`: これを指定した場合、フィルタがイベントのフィルタリングを要求されると、この関数がコールされます。このメソッドは、フィルタが特定のデバイスにアタッチされている場合のみコールされます。イベントの配列が渡されます。この配列内のイベントの更新や削除、または新しい配列の作成が可能です。システムに処理を続行させるイベントの配列を返す必要があります。パラメータ・イベントは、エッジ・イベントの配列 (`oracle.edge.common.Event []`) です。
- `function groupFilter(events)`: このコールは、デバイス・グループがフィルタリングのためにフィルタをコールすると起動されます。このコールは、フィルタが 1 つのデバイスでなくデバイス・グループにアタッチされている場合にのみ使用されます。パラメータ・イベントは、エッジ・イベントの配列 (`oracle.edge.common.Event []`) です。
- `function instructionFilter(instruction)`: これは、システムが命令イベントをデバイスに送信しようとするときにコールされます。イベントは変更可能です。イベントをすべてまとめてブロックする場合は、NULL を返します。パラメータは 1 つのイベント (`oracle.edge.common.Event`) です。

前述のいずれかの関数を 1 つ記述することができます。その他の関数は無視されます。初期化されたブロックは、フィルタの起動時にコールされます。例 6-9 に示すソース・スクリプトでは、検出されたすべてのタグ ID がコンソールに出力されます。

例 6-9 ソース・スクリプト

```
function filter(events)
{
    for ( i=0; i<events.length; i++ )
    {
        id = events[i].getTagId();
        java.lang.System.out.println("Id:"+id )
    }
}
```

Movement フィルタの構成

Movement フィルタは、**Real Time Location System** 測定を使用して、エラーや干渉による急増や急な運動変化を平坦にすることにより移動追跡を円滑化します。Movement フィルタは、現在の位置と以前の位置との距離を計算します。この距離が距離しきい値を下回ると、現在のイベントがフィルタリングで除外されます。

Movement フィルタのパラメータは、Distance Threshold というパラメータ 1 つです。このパラメータを定義するには、レポートする必要がある最小移動を示す正の数 (int 値) を入力します。

Movement フィルタのシグネチャを表 6-23 に示します。

表 6-23 Movement フィルタのシグネチャ

イベント・フィールド	値
sourceName	このフィールドは、イベントの発信元を識別する。このフィールドはオプションで、値はクライアントによって設定される。
correlationId	このフィールドはクライアントによって設定され、特定のクライアントへのメッセージ・レスポンス (デバイスが機能しているかどうかのチェックなど) に使用される。同じクライアントによって返されたメッセージは同じ ID を持つ。オプションのフィールド。
siteName	このイベントを生成したサイトの名前。
deviceName	このイベントを生成したデバイスの名前。
time	イベントが生成された時刻。
type	測定イベント。範囲は 200 ~ 299。
subtype	RTLS Moved (サブタイプ 4)。
id	タグの ID。
data	タグのデータ・ペイロード。

Pallet Pass Thru フィルタの構成

Pallet Pass Thru フィルタは指定された時間内に受信したすべてのイベントを収集し、1 つのイベントとして送信します。パレットまたはコンテナがリーダー・デバイスのゲートまたは伝達範囲を通過すると、このフィルタがこれらすべてのタグに対して 1 つのイベントを生成します。このフィルタにより、コンテナまたはパレットが所有しているアイテムを確認できます。

Pallet Pass Thru フィルタには、次のパラメータが含まれます。

- [Exit Event Threshold Time](#)
- [Event Collect Control Time](#)

Exit Event Threshold Time

このパラメータを定義するには、デバイスが最後にタグを読み取ってから、デバイスの伝達範囲を出たとみなされるまでの時間 (int 値) をミリ秒単位で入力します。このパラメータには、デバイスがこれらのタグ検出中イベントをレポートする頻度 (リーダー・サイクル) を 50 ミリ秒から 2 秒の範囲で指定します。この頻度を大きすぎる値 (2 秒など) に設定すると、デバイスがタグを完全に見過ごす可能性があります。

Event Collect Control Time

このパラメータを定義するには、新しい読取りサイクルを開始する前に、デバイスがパレットまたはコンテナに含まれているタグの読取りサイクルを完了する時間（int 値）をミリ秒単位で入力します。この時間が経過すると読取りサイクルは終了し（デバイスがすべての新しいタグを読み取ったとみなされ）、Pallet Pass Thru フィルタは次のシグネチャ（表 6-24 を参照）を持つイベントを生成します。

表 6-24 Pallet Pass Thru イベントのシグネチャ

イベント・フィールド	値
sourceName	このフィールドは、イベントの発信元を識別する。このフィールドはオプションで、値はクライアントによって設定される。
correlationId	このフィールドはクライアントによって設定され、特定のクライアントへのメッセージ・レスポンス（デバイスが機能しているかどうかのチェックなど）に使用される。同じクライアントによって返されたメッセージは同じ ID を持つ。オプションのフィールド。
siteName	このイベントを生成したサイトの名前。
deviceName	このイベントを生成したデバイスの名前。
time	イベントが生成された時刻。
type	測定イベント。範囲は 200 ~ 299。
subtype	RFID Pallet Pass（サブタイプ 6）。
id	タグ ID のカンマ区切りのリスト。
data	データのカンマ区切りのリスト。

Pallet Shelf フィルタの構成

Pallet Shelf フィルタは指定された間隔内に受信したすべてのイベントを収集し、1 つのイベントとして送信します。このフィルタにより、多数のアイテムを持つ新しいコンテナまたはパレットがデバイス・リーダーの読取り範囲またはゲートを出入りするときに識別できます。

Pallet Shelf フィルタには、次のパラメータがあります。

- [Exit Event Threshold Time](#)
- [Event Collect Control Time](#)

Exit Event Threshold Time

このパラメータを定義するには、デバイスが最後にタグを読み取ってから、デバイスの伝達範囲を出たとみなされるまでの時間（int 値）をミリ秒単位で入力します。Pallet Shelf フィルタは、Exit Event Threshold Time パラメータで定義した間隔が経過してイベントが生成されないと、キャッシュを警告なしにクリアします。

Event Collect Control Time

このパラメータを定義するには、新しい読取りサイクルを開始する前に、デバイスがパレットまたはコンテナのタグの読取りサイクルを完了する時間（int 値）をミリ秒単位で入力します。この時間が経過すると読取りサイクルは終了し（デバイスがすべての新しいタグを読み込んだとみなされ）、Pallet Shelf フィルタはイベントを生成します。

Pallet Shelf フィルタが生成するイベント

Pallet Shelf フィルタは次の2つのイベントを生成します。

- [MULTIPLE IN FIELD イベント](#)
- [MULTIPLE OUT FIELD イベント](#)

MULTIPLE IN FIELD イベント

デバイスが最初にタグを検出すると、Pallet Shelf フィルタは MULTIPLE IN FIELD イベントを生成します。このイベントには、次のシグネチャ (表 6-25) が含まれます。

表 6-25 MULTIPLE IN FIELD イベントのシグネチャ

イベント・フィールド	値
sourceName	このフィールドは、イベントの発信元を識別する。このフィールドはオプションで、値はクライアントによって設定される。
correlationId	このフィールドはクライアントによって設定され、特定のクライアントへのメッセージ・レスポンス (デバイスが機能しているかどうかのチェックなど) に使用される。同じクライアントによって返されたメッセージは同じ ID を持つ。オプションのフィールド。
siteName	このイベントを生成したサイトの名前。
deviceName	このイベントを生成したパレットまたはコンテナを読み取るデバイスの名前。
time	Pallet Shelf フィルタがこのイベントを生成した時刻。
type	測定イベント。範囲は 200 ~ 299。
subtype	RFID Pallet In-Field (サブタイプ 4)。
id	タグ ID のカンマ区切りのリスト。
data	データのカンマ区切りのリスト。

MULTIPLE OUT FIELD イベント

Exit Event Threshold Time パラメータに定義した間隔が経過すると、Pallet Shelf フィルタは MULTIPLE OUT FIELD イベントを生成します。このイベントには、次のシグネチャ (表 6-26) が含まれます。

表 6-26 MULTIPLE OUT FIELD イベントのシグネチャ

イベント・フィールド	値
sourceName	このフィールドは、イベントの発信元を識別する。このフィールドはオプションで、値はクライアントによって設定される。
correlationId	このフィールドはクライアントによって設定され、特定のクライアントへのメッセージ・レスポンス (デバイスが機能しているかどうかのチェックなど) に使用される。同じクライアントによって返されたメッセージは同じ ID を持つ。オプションのフィールド。
siteName	このイベントを生成するサイトの名前。
deviceName	このイベントを生成したパレットまたはコンテナを読み取るデバイスの名前。
time	Pallet Shelf フィルタがこのイベントを生成した時刻。
type	測定イベント。範囲は 200 ~ 299。
subtype	RFID Pallet Out Filed (サブタイプ 5)。
id	タグ ID のカンマ区切りのリスト。
data	データのカンマ区切りのリスト。

Pass フィルタの構成

タグがデバイス・リーダーの伝達範囲またはゲートを通過すると、一連のタグ検出中イベントが生成されます。タグが伝達範囲に入ると、デバイスはこれらのイベントを定期的にレポートします。タグがリーダーの伝達範囲を出ると、レポートは停止します。

デバイス・リーダーが生成する一連のイベントをアプリケーションが必要とせず、タグがデバイスのゲートまたは伝達範囲を通過したことがわかればよいことも多々あります。Pass フィルタはそのような状況に適しており、すべてのタグ検出中イベントを、リーダー・デバイスの伝達範囲を通過する一意のタグごとに1つのイベントにまとめます。

Pass フィルタのパラメータは、Exit Event Threshold Time というパラメータ1つです。このパラメータを定義するには、デバイスが最後にタグを読み取ってから、デバイスの伝達範囲を出たとみなされるまでの時間 (int 値) をミリ秒単位で入力します。このパラメータには、デバイスがこれらのタグ検出中イベントをレポートする頻度 (リーダー・サイクル) を50ミリ秒から2秒の範囲で指定します。この頻度を大きすぎる値 (2秒など) に設定すると、デバイスがタグを完全に見過ごす可能性があります。

デバイスが最初にタグを読み取ると、Pass フィルタはタグの ID (タグ ID) をキャッシュし、そのタグ ID がキャッシュに読み取られた時間を記録すると、すぐにパススルー・イベントを送信します。フィルタは、このキャッシュされたタグ ID がその後に読み取られないようにブロックします。フィルタがデバイスから新しい読取りを受信するたびに、タグ ID をキャッシュに読み取った時刻を更新します。キャッシュ時刻と Exit Event Threshold Time に設定された値の合計が現在の時間より小さい場合、Pass フィルタはキャッシュからタグ ID を消去します。デバイスが次にこのタグを読み取ると、フィルタは新しいイベントとみなし、そのタグ ID をキャッシュして、新しいパススルー・イベントを送信します。

パススルー・イベントには、次のシグネチャ (表 6-27) が含まれます。

表 6-27 パススルー・イベントのシグネチャ

イベント・フィールド	値
sourceName	このフィールドは、イベントの発信元を識別する。このフィールドはオプションで、値はクライアントによって設定される。
correlationId	このフィールドはクライアントによって設定され、特定のクライアントへのメッセージ・レスポンス (デバイスが機能しているかどうかのチェックなど) に使用される。同じクライアントによって返されたメッセージは同じ ID を持つ。オプションのフィールド。
siteName	このイベントを生成したサイトの名前。
deviceName	このイベントを生成したデバイスの名前。
time	イベントが生成された時刻。
type	測定イベント。範囲は 200 ~ 299。
subtype	パススルー・イベント (サブタイプ 3)。
id	タグの ID。
data	タグのデータ・ペイロード。

Polygon フィルタの構成

Polygon フィルタは、**Real Time Location System** デバイスによってレポートされたすべての移動測定をフィルタリングで除外し、タグが定義済みのポリゴンに出入りしたときにのみイベントを生成します。ポリゴンは、頂点とカッコを定義する x,y 座標のセットを使用して定義します。たとえば、(x,y), (x,y), ..., (...), ... のように定義します。

Polygon フィルタのパラメータは、PolygonCoordinates というパラメータ 1 つです。このパラメータを定義するには、String 値を ((x1, y1), (x2, y2)...(xn, yn)) という形式で入力します。

Regex フィルタの定義

Regex フィルタは、ストリームを削除するか、ストリームの通過を許可するためのタグを検索する正規表現検索を実行します。このフィルタを使用すると、フィルタのパターン・セットを定義してイベントの任意の範囲を検索できます。フィルタにより検索基準の一致が見つかり、イベントがシステムを通過することが許可されます。一致が見つからない場合、そのイベントがフィルタリングで除外されます。RegexFilter は汎用フィルタであり、どのイベント・タイプに対しても使用できます。

Regex フィルタのパラメータは、allow というパラメータ 1 つです。このパラメータを定義するには、フィルタの通過を許可するイベントの正規表現 (String) を次の形式で入力します。

```
<field>~<pattern>&&<field>~<pattern>||...
```

ここで、<field> は、一致をフィルタリングするイベント・フィールドの名前です。このフィルタでは次のフィールドがサポートされます。

- TAGID
- DATA
- CORRELATIONID
- SOURCENAME
- TYPE
- SUBTYPE
- DEVICENAME
- SITENAME
- TIME

注意： TIME フィールドは、<pattern> と比較される前に time() long 値に変換されます。

<pattern> は標準の正規表現、&& は AND 接続語、|| は OR 接続語です。たとえば、イベント・タイプが 103 のすべてのイベントと、最初の 3 桁が 0FE に一致するタグを RegexFilter インスタンスにより検索する場合は、次のように入力します。

```
type~103&&tagid~^0FE
```

Shelf フィルタの構成

Shelf フィルタは、リーダーの読取り範囲内でタグが検出されたとき、またはタグが読取り範囲を離れたときにイベントを生成するデバイス・レベルのフィルタです。Pass フィルタと同様に、Shelf フィルタのパラメータも Exit Event Threshold Time というパラメータ 1 つです。このパラメータを定義するには、デバイスが最後にタグを読み取ってから、デバイスの伝達範囲を出たとみなされるまでの時間 (int 値) をミリ秒単位で入力します。ただし、Pass フィルタとは異なり、Shelf フィルタは Exit Event Threshold Time パラメータで定義した間隔が経過してイベントが生成されないと、キャッシュを警告なしにクリアします。

Shelf フィルタが生成するイベント

Shelf フィルタは次の 2 つのイベントを生成します。

- IN FIELD イベント
- OUT FIELD イベント

IN FIELD イベント

Shelf フィルタは、デバイスが最初にタグを検出すると、このイベントを生成します。このイベントには、次のシグネチャ (表 6-28) が含まれます。

表 6-28 IN FIELD イベントのシグネチャ

イベント・フィールド	値
sourceName	このフィールドは、イベントの発信元を識別する。このフィールドはオプションで、値はクライアントによって設定される。
correlationId	このフィールドはクライアントによって設定され、特定のクライアントへのメッセージ・レスポンス (デバイスが機能しているかどうかのチェックなど) に使用される。同じクライアントによって返されたメッセージは同じ ID を持つ。オプションのフィールド。
siteName	このイベントを生成したサイトの名前。
deviceName	このイベントを生成したデバイスの名前。
time	Shelf フィルタがこのイベントを生成した時刻。
type	測定イベント。測定イベントの範囲は 200 ~ 299。
subtype	IN-FIELD イベント (サブタイプ 2)。
id	タグの ID。
data	タグのデータ・ペイロード。

OUT FIELD イベント

Shelf フィルタは、Exit Event Threshold Time パラメータに定義した間隔が経過すると、このイベントを生成します。このイベントには、次のシグネチャ (表 6-29) が含まれます。

表 6-29 OUT FIELD イベントのシグネチャ

イベント・フィールド	値
sourceName	このフィールドは、イベントの発信元を識別する。このフィールドはオプションで、値はクライアントによって設定される。
correlationId	このフィールドはクライアントによって設定され、特定のクライアントへのメッセージ・レスポンス (デバイスが機能しているかどうかのチェックなど) に使用される。同じクライアントによって返されたメッセージは同じ ID を持つ。オプションのフィールド。
siteName	このイベントを生成したサイトの名前。
deviceName	このイベントを生成したデバイスの名前。
time	Shelf フィルタがこのイベントを生成した時刻。
type	測定イベント。測定イベントの範囲は 200 ~ 299。
subtype	OUT-FIELD イベント (サブタイプ 2)。
id	タグの ID。
data	タグのデータ・ペイロード。

デバイスが最初にタグを検出すると、Shelf フィルタはタグの ID をキャッシュし、IN FIELD イベントを生成します。この時点で、タグは各リーダー・サイクル中に読み取られます。これらのサイクル中にタグが読み取られない場合には、他のサイクル中に読み取られます。デバイスが、Event Exit Threshold Time パラメータで指定したよりも長い時間継続してタグを読み取らないと、フィルタはタグの ID をキャッシュから削除し、OUT FIELD イベントを生成します。デバイスの読取り範囲を出ると、デバイスはタグの読取りを停止します。

Oracle Sensor Edge Server インスタンス用のディスパッチャの管理

Oracle Sensor Edge Server の主要な出力は、フィルタリングされたデータ・イベントです。デバイスと各フィルタは、このイベント・データを標準化および最小化します。Oracle Sensor Edge Server では、次のディスパッチャ方法によりイベント・データを配信できます。

- Web サービス
- HTTP/ ポスト
- EPC PML
- アプリケーション・レベル・イベント (ALE)
- Oracle Streams/ アドバンスト・キューイング

これらのディスパッチャ方法の構成方法については、次の項で説明しています。

- 「Oracle Streams を使用するためのエッジ・ディスパッチャの構成」
- 「Web サービスにイベント・メッセージを送信するためのディスパッチャの構成」
- 「HTTP を使用してイベント・メッセージを送信するためのディスパッチャの構成」
- 「PML ディスパッチャの構成」
- 「テンプレート・ディスパッチャの構成」
- 「ALEDispatcher の構成」
- 「Null ディスパッチャの使用」

Web サービスにイベント・メッセージを送信するためのディスパッチャの構成

新しいメッセージを配信する必要がある場合、クライアント・デバイスまたはアプリケーションは、Oracle Sensor Edge Server が起動する SOAP コールを登録できます。

Web サービスを介してイベント・メッセージを配信するように WebService ディスパッチャを構成するには、クライアント・コールが記述された WSDL (Web Service Definition Language) ドキュメントのサービス URL を入力します。この URL は、Web サービスのエンドポイント (ポート) を指す必要があります。たとえば、`http://localhost:8888/wsdl/mytest.wsdl` と入力します。このドキュメントには、EdgeClientCallback の portType と、その子要素としてコール processEvent が含まれている必要があります。Oracle Sensor Edge Server は起動時に、この WSDL ドキュメント内に定義されているサービスへの接続およびバインドを試行します。

HTTP を使用してイベント・メッセージを送信するためのディスパッチャの構成

HTTP 1.0 を使用してクライアントにイベントをルーティングするようにディスパッチャを構成すると、Oracle Sensor Edge Server では各イベント・メッセージがクライアントに個別にポストされます。Oracle Sensor Edge Server ではこれらのポストが順番に実行されるため、1つのポストがブロックされると、それに続くすべてのポストもブロックされます。

HTTP を使用してイベントをルーティングするように Oracle Sensor Server を構成するには、次のパラメータを定義します。

- URL: Oracle Sensor Edge Server がディスパッチのときにイベント・メッセージをポストする宛先のサブレット、JSP または CGI の URL を入力します。このディスパッチャを構成するには、URL を次の形式で入力します。

```
http://hostname:port/serverPath
```

Oracle Sensor Edge Server が HTTP ディスパッチャを使用する場合は、クライアント・インタフェースは Oracle Sensor Edge Server にいつどのようにコールするのかを指示する必要があります。

- proxyServer: プロキシ・サーバーの IP またはホスト名を入力します。
- proxyPort: プロキシ・サーバーのポートを入力します。たとえば、80 と入力します。

PML ディスパッチャの構成

PML ディスパッチャは、HTTP、FTP、FILE などのプロトコルに対して **トランスポート・ライブラリ** を使用して EPC PML 形式でイベントをポストします。

注意: PML は、測定をレポートする場合に使用されるレガシー形式です。

PML ディスパッチャを使用するように Oracle Sensor Server を構成するには、宛先 URI を定義します。「[デバイスおよびディスパッチャの URI パラメータの設定](#)」を参照してください。

テンプレート・ディスパッチャの構成

テンプレート・ディスパッチャ (EventFlowDispatcher と呼ばれる) は、様々なソースからイベントを送受信し、eflow.xml 構成ファイルを使用して構成された有限状態マシンを介してイベントを処理します。このディスパッチャは、トランスポート・ライブラリを使用して、HTTP、POJ、Web サービスおよびダイレクト・データ・ストリームを介してデータを送受信します。

テンプレート・ディスパッチャを使用するように Oracle Sensor Server を構成するには、eflow.xml の場所を入力します。

ALEDispatcher の構成

ALEDispatcher を ALE Web サービス・インタフェースと組み合わせて使用すると、レポート仕様が満たされたときに ALE サブスクライバに通知することができます。

Null ディスパッチャの使用

Null ディスパッチャはデフォルトで作成され、渡されたすべてのイベントを破棄します。破棄されたイベントは保存またはスプールされません。このディスパッチャは、Oracle Sensor Edge Server がイベントをディスパッチしないようにする場合にのみ使用します。

Oracle Streams を使用するためのエッジ・ディスパッチャの構成

Oracle Streams およびアドバンスト・キューイングを使用するように Oracle Sensor Edge Server を構成すると、エッジ・ディスパッチャがイベント・メッセージを取得および配信する方法を制御できます。Web サービスや HTTP ディスパッチャ・オプションとは異なり、Oracle Streams ディスパッチャを使用してディスパッチされたイベント・メッセージは、エントリ・ポイントに直接取り次ぐ必要がありません。Oracle Streams ディスパッチャでは、ルールベースの処理およびエージェント・テクノロジーがサポートされます。また、Streams ディスパッチャでは UTF-8 エンコーディングのみがサポートされます。

ヒント:

- Oracle Streams により、1 つまたは多数のデータベースでデータ・ストリーム内のデータ、トランザクションおよびイベントを伝播し、管理できるようになるため、これは、用意されたディスパッチャ・オプションの中で最も柔軟性が高いオプションです。
- Oracle Streams ディスパッチャには、JDK 1.4.x が必要です。

イベント・メッセージは、ステージング・エリア (内部キュー) に置かれるデータです。このデータは、各種のクライアント・デバイスやアプリケーション (イベント・メッセージのコンシューマ) 用に様々な方法で集約できます。Oracle Streams をディスパッチャとして使用すると、Oracle Sensor Edge Server またはアプリケーションではなく、データベースのデータおよびイベント・レイヤーにより、どのイベントがいつ生成されるかが決まります。データおよびイベント・レイヤーは、各クライアント・デバイスまたはアプリケーションに適したイベント・メッセージを決定するルールベースの処理を行います。

イベント・メッセージが取得され、ステージング・キューに置かれると、イベント・メッセージ・データはルール評価ジョブによって処理できます。このジョブはステージング・キューからイベント・メッセージを取得し、Oracle Sensor Edge Server ルール・セットと比較します。各ルールには、ルールが当てはまる場合に実行されるアクションがあります。これらのアクションには、その他のアプリケーションで使用できるその他のキューに、イベント・メッセージを伝播するための PL/SQL コールバックが含まれます。

これらのルールベースのアクションに加え、ルール評価ジョブは、Oracle Sensor Edge Server やその他のソースからセンサー・データを受信するセンサー・データ・リポジトリ (SDR) をコールしてアプリケーションを起動します。SDR には、フィルタリングされたすべてのセンサー・イベントをシステムに格納するアーカイブ表のセットが含まれます。詳細は、第 4 章「センサー・データ・リポジトリの使用」を参照してください。

注意： ルールによって処理されていない生の未フィルタ処理イベント・データが必要なアプリケーションは、AQ 通知を使用してステー징・エリアに接続できます。

Streams ディスパッチャを構成するには、**threadnum** フィールドに、アウトバウンド測定値を処理するために作成する最大スレッド数を入力します。各スレッドが接続を保持するため、これは、アウトバウンド・イベント用に使用するセンサー・データ・リポジトリへの接続数でもあります（インバウンド・イベント用に使用する接続は別に存在します）。0 は無効な値です。「リリース 3 (10.1.3) のセンサー・データ・リポジトリおよびセンサー・データ・ストリームの手動構成」も参照してください。

用語集

Oracle Sensor Edge Server

すべてのリーダーとアプリケーション中間層の間にあるサーバー。すべてのリーダーとのインタフェースとして機能し、標準化されたデータをアプリケーション・サーバーに戻す。

Radio Frequency Identification (RFID)

電子シリアル番号 (ESN) またはメモリーが埋め込まれた小型のトランスポンダを使用して、1つ以上の周波数で ID を発信する。

Real Time Location System (RTLS)

電波を使用して、タグが付けられたアイテムのリアルタイムの位置情報を生成するテクノロジー。

transports.xml

使用可能なトランスポートを [トランスポート・ライブラリ](#) に通知する、edgeHome/config ディレクトリ内のファイル。たとえば、次のように構成されている。

```
<transports> <transport name="com" type="stream"
class="oracle.edge.tools.io.SerialConnection"> <params> <param name='port' value='com1'
/> <param name='driver' value='com.sun.comm.Win32Driver' /> </params> </transport>
<transport name="tcp" type="stream" class="oracle.edge.tools.io.TcpConnection" />
<transport name="file" type="stream" class="oracle.edge.tools.io.FileConnection" />
<transport name="stdout" type="stream" class="oracle.edge.tools.io.StdoutConnection" />
<transport name="http" type="stream" class="oracle.edge.tools.io.HttpConnection" >
<params> <param name='proxyServer' value='www-proxy.us.oracle.com' /> <param
name='proxyPort' value='80' /> </params> </transport>
</transports>
```

各 <transport> 要素は新しいトランスポート・タイプを定義する。<transport> タグごとに、名前 (URI 内で使用する、このトランスポートに一意の名前) とクラス (ロードするクラスを指定する) を定義する必要がある。このクラスは

oracle.edge.tools.io.Connection インタフェースを実装する必要がある。

また、<param> タグを使用してトランスポートのデフォルト・パラメータを指定することもできる。

アンテナ (antenna)

各タグには1つ以上のアンテナがある。通信リンクの反対側にあるリーダーも、アンテナを持っている必要がある。同時に複数のアンテナを使用するリーダーもある。アンテナは、プロトコル、周波数およびアプリケーションにより、表面に貼られた薄く細長いメタルから、数メートルの長さのゲート型のポータル・アンテナまで様々である。

イベント (event)

イベントとは、センサー・デバイスまたはアプリケーションから送信される、状態が変更されたことを通知するメッセージである。Sensor Edge Server は、これらのセンサー・デバイスやアプリケーションからデータを受信し、共通のデータ形式にすることでこれらのイベント・メッセージの内容を標準化し、フィルタリングして無関係な情報や不要なイベントを除外する。

タグ (tag)

(RFID タグともいう。) チップ、1 つ以上のアンテナおよび電源を含む単一のユニット。電池駆動または外部電源を使用する場合、アクティブ・タグという。電源が電磁誘導 (つまり、リモートで生成された電磁波を利用して電力を生成する光電効果を使用) の場合は、パッシブ・タグという。変更できないデータを保持するタグは、読取り専用タグという。「[チップ](#)」を参照。

チップ (chip)

タグでは、エンベデッド・メモリーを搭載したシリコン・チップが使用される。チップはワイヤレス・プロトコルおよびエンベデッド・メモリーへのアクセス機能を実装している。アクティブ・タグの場合は、単一のチップではなく、ボード全体を指す。「[タグ](#)」を参照。

デバイス (device)

エッジ・デバイスは、Radio-Frequency Identification (RFID) リーダー、ドライ接点、半導体レーザー、カラーセル、ロボット・ピッカー、またはケミカル・ライトやメッセージ・ボードのような表示機器など、センサーベースのアーキテクチャのエンド・ポイント。センサーは、特定の状態変化を測定するハードウェアまたはソフトウェアのエンド・ポイント。通常は、半導体レーザー光線が遮られた場合などの物理的な変化を測定するが、エッジ・コントローラ上で実行されているモニター・デーモンが終了した場合など、ソフトウェア内部で発生した変化を測定することもできる。また、センサーはソフトウェアの欠陥も測定する。デバイスはドライバのインスタンスである。

トランスポート・ライブラリ (Transport Library)

シリアル・ポートや TCP/IP などの通信チャンネルを抽象化するクラスのセット。トランスポート・ライブラリを使用すると、サード・パーティのプラグインを追加できる。

リーダー (reader)

タグとの読み書きを行う。通常、リーダーにはシリアル・インタフェースがあり、ホスト・コンピュータとの通信に使用される。このプロトコルで広く適用されている標準はない。タグに電波を送り、タグが戻した電波をデータに変換することにより、RFID タグに格納されたデータを取得する処理を読取りという。

リーダー読取り範囲 (reader field)

リーダーの処理対象の範囲。タグがリーダー読取り範囲の外にあると、電波を受信できず読取りできない。

索引

A

Activex API のメソッド, 5-11
 barcode_read(), 5-12
 is_supported(), 5-14
 process_instruction(), 5-13
 rfid_kill(), 5-12
 rfid_read(), 5-12
 rfid_write(), 5-12
 set_trigger_barcode_read(), 5-13
 set_trigger_rfid_read(), 5-12
Activex アプリケーション・インタフェース
 廃止, 5-16

B

BarcodeReadMacro パラメータ, 5-6

D

DestinationApplication
 定義, 5-5
 パラメータ, 5-5

E

EDG_CAP, 4-10
EDG_CAP_TAB, 4-4
EDG_CTXT, 4-10
EDG_CTXT_REL, 4-10
EDG_CTXT_REL_NAME_VW, 4-11
EDG_CTXT_REL_TAB, 4-5
EDG_CTXT_TAB, 4-5
EDG_DEV_CAP_VW, 4-12
EDG_DEV_DIAG_VW, 4-13
EDG_DEV_EVENT_VW, 4-13
EDG_DEV_LAST_DIAG_VW, 4-14
EDG_DEV_LAST_OBSV_VW, 4-15
EDG_DEVICE, 4-11
EDG_DEVICE_TAB, 4-6
EDG_DIAG, 4-16
EDG_DIAG_TAB, 4-6
EDG_EVENT, 4-17
EDG_EVENT_INFO, 4-17
EDG_EVENT_INFO_TAB, 4-7
EDG_EVENT_TAB, 4-8
EDG_EVENT_VW, 4-18
EDG_LOG, 4-8

EDG_TAG, 4-19
EDG_TAG_LAST_DEV_VW, 4-19
EDG_TAG_PATH_VW, 4-20
EDG_TAG_TAB, 4-9
EdgeMobileCofig.xml, 5-3
EPC コンプライアンス統合, 2-4

J

Java Naming and Directory Interface (JNDI), 2-4

O

OC4J
 インストール, 1-3
 インストール、リリース 2 (10.1.2) インスタンスか
 らのアップグレード, 1-3
 インストール、リリース 3 (10.1.3) インスタンスか
 らのアップグレード, 1-4
Oracle Application Server Control, 2-3
Oracle Sensor Edge Mobile
 Pocket PC でのインストールと起動, 1-8
 アーキテクチャ, 2-8
 アプリケーションへの接続, 5-3
 インストール, 1-8
 エミュレータのインストール, 1-10
 オブジェクト宣言, 5-11
 概要, 5-2
 管理, 2-9, 5-16
 国際化, 5-18
 サンプル・コードおよびデモ・アプリケーション,
 2-9
 説明, 2-6
 ディスクパッチャとドライバの構成, 5-3
 デバイス・ドライバのサポート, 2-9
 要件, 1-2
Oracle Sensor Edge Server
 opmnctl を使用した停止と起動, 3-19
 OracleAS Enterprise Manager を使用した停止と起動,
 3-19
 アーキテクチャ, 2-6
 アンインストール, 1-5
 インストール, 1-1
 エッジ・ディスクパッチャの管理, 6-33
 概要, 2-1
 概略, 2-4
 拡張されたセキュリティ, 2-4
 向上したパフォーマンス, 2-3

停止と起動, 3-19
トランスポート・レイヤー, 2-4
配置に関する考慮事項, 2-10
Oracle Sensor Edge Server アプリケーション
開発, vii
Oracle Sensor Edge Server コンソール, 2-2
説明, 2-6
Oracle Sensor Edge Server のインストール, 1-1

P

Pocket PC
RFID タグの読取り, 1-10
Sensor Edge Server のインストール, 1-8
デフォルトのデバイス構成の変更, 1-10
バーコード・データの読取り, 1-10
Pocket PC 2003, 1-2

S

Sensor Edge Mobile
停止, 1-10
Sensor Edge Server コンソール
表示, 1-5
Sensor Server とデバイス管理, 2-6

い

イベント・データ, 3-24
各イベント詳細の表示, 3-24
センサー・データ・リポジトリに格納されているイ
ベント・データの表示, 3-31
タグ ID によるデータの表示, 3-31
デバイス名によるデータの表示, 3-32
リアルタイムのイベント・データの表示, 3-29
イベント・ハンドラ, 5-14
インストール
拡張インストール・モード, 1-3
確認, 1-5
ログ・ファイル, 1-5

お

オブジェクト宣言, 5-11

か

拡張アーカイブ記述子, 3-35
拡張アーカイブ・ファイル, 3-34
アップロード, 3-37
構成, 3-35
パッケージ化, 3-37
拡張インストール・モード, 1-3
拡張機能
拡張アーカイブ・ファイル, 3-35
拡張アーカイブ・ファイルのアップロード, 3-37
拡張アーカイブ・ファイルのパッケージ化, 3-37

き

キー・シーケンス・マクロ
作成, 5-7
キー・シーケンス・マクロ・パラメータ, 5-6

キーボード・ディスパッチャ
構成, 5-4
規則
表記, viii

こ

このリリースの新機能, 2-2

し

手動の構成および配置, 1-11
処理
イベント, 2-7
ローカル, 2-7

す

スキーマ・リファレンス, 4-4

せ

センサー・データ・アーカイブおよびルール, 2-6
センサー・データ・ストリーム, 2-4
要件, 1-2
センサー・データの収集, 2-5
センサー・データのディスパッチ, 2-5
センサー・データのフィルタリング, 2-5
センサー・データ・リポジトリ
PL/SQL パッケージ, 4-22
拡張機能, 2-3
操作と問合せ, 4-3
要件, 1-2
リレーショナル表, 4-2
センサー・データ・リポジトリ / センサー・データ・ス
トリーム
手動の構成および配置, 1-11
センサー・データ・リポジトリの問合せ, 3-31
センサー・リポジトリ
PL/SQL パッケージ, 4-3
使用, 4-2
リレーショナル・ビュー, 4-2

そ

ソフトウェア要件, 1-2

た

対象読者, viii

て

ディスパッチャ
HTTP を使用してイベント・メッセージを送信, 6-34
Null, 6-35
Oracle Streams を使用するためのエッジ・ディスパ
ッチャの構成, 6-35
Web サービスにイベント・メッセージを送信, 6-34
現行ディスパッチャの設定, 3-8
データベース接続
テスト, 1-6
データベース接続のテスト, 1-6

デバイス
各デバイスの起動と停止, 3-21
グループとして停止して起動, 3-18
作成, 3-12
デバイス・レベルのフィルタの追加, 3-15
デバイス・グループ, 2-7
Oracle Sensor Edge Server インスタンスからの削除,
3-18
グループ・レベルのフィルタの追加, 3-17
作成, 3-11
フィルタ・インスタンスおよびデバイスの変更, 3-18
編集, 3-18
デバイス・コントローラ
廃止, 2-4
デバイス・ドライバ, 1-2, 2-7
Intermec IP3, 1-2
Symbol 9000G, 1-2
デバイス・マネージャ
デバイス・ステータスの確認, 5-10
電子シリアル番号 (ESN), 2-4

と

ドライバ
AlienDevice, 6-6
Edge Simulator, 6-10
Intermec, 6-16
Intermec BRI, 6-14
LPML プリンタ, 6-16
Matrics, 6-17
PatliteDriver, 6-19
Prolite ドライバ, 6-19
Samsys ドライバ, 6-19
Tyco, 6-22
事前に組み込まれている, 6-4
ドライバ・インスタンス
作成, 3-12
ドライバ・マネージャ, 2-7

は

ハードウェア要件, 1-2

ひ

表記規則, viii

ふ

フィルタ
Check Tag ID, 6-24
Cross-Reader Redundant, 6-25
Debug, 6-25
JavaScript フィルタ, 6-26
Movement フィルタ, 6-27
Pallet Pass Thru, 6-27
Pallet Shelf, 6-28
Pass, 6-30
Regex フィルタ, 6-31
Shelf, 6-32
組込みフィルタのパラメータの定義, 6-24
作成, 3-15
優先度の設定, 3-22

フィルタ・インスタンス
作成, 3-15

よ

要件

センサー・データ・ストリーム, 1-2
センサー・データ・リポジトリ, 1-2
ソフトウェア, 1-2
ハードウェア, 1-2

ろ

ログ・ファイル

表示, 3-30

