



クイック検索

検索

Copyright © 2012-2017 intra-mart
CORPORATION

目次

- 1. 改訂情報
- 2. はじめに
 - 2.1. 本書の目的
 - 2.2. 前提条件
 - 2.3. 対象読者
 - 2.4. 用語解説
- 3. セットアップの流れ
- 4. ミドルウェアのセットアップ
 - 4.1. JDK
 - 4.2. データベース
 - 4.3. Web Application Server
 - 4.4. Web Server
 - 4.5. Apache Cassandra
 - 4.6. Apache Solr
- 5. WARファイルの作成
 - 5.1. プロジェクトの作成とモジュールの選択
 - 5.2. ユーザモジュール
 - 5.3. intra-mart Accel Platform の設定ファイル
 - 5.4. WARファイルの出力
 - 5.5. 静的ファイルの出力
- 6. Web Application Server の起動・停止
 - 6.1. Windows
 - 6.2. Linux
- 7. WARファイルのデプロイ
- 8. テナント環境セットアップ
 - 8.1. システム管理者情報
 - 8.2. テナント情報
 - 8.3. テナント環境情報
 - 8.4. テナント管理者情報
 - 8.5. LDAP連携・設定
 - 8.6. ログインセッション管理
 - 8.7. Apache Cassandra接続情報
 - 8.8. Apache Solr接続情報
 - 8.9. 登録
- 9. ライセンスの登録
- 10. アップデート・パッチの適用・モジュール構成の変更
 - 10.1. アップデート
 - 10.2. パッチ
 - 10.3. モジュール構成の変更
- 11. 付録
 - 11.1. Resinのクラスタリング
 - 11.2. 二重ログイン防止機能
 - 11.3. 統合Windows認証
 - 11.4. IM-Notice
 - 11.5. WARファイルによる複数テナント
 - 11.6. テナント解決機能
 - 11.7. ポート一覧
 - 11.8. IM-Juggling の応用
 - 11.9. intra-mart Accel Platform のチューニング
 - 11.10. バックアップ・リストア(復元)
 - 11.11. アンインストール
 - 11.12. サンプルデータの投入
 - 11.13. セットアップで困ったら・・・

改訂情報

変更年月日	変更内容
2012-10-01	初版
2012-11-01	第2版 下記を追加・変更しました 「データベースサーバ」を追加 「セットアップで困ったら・・・」を追加
2012-12-21	第3版 下記を追加・変更しました 「ベースURL」を追加 「認可ポリシー設定キャッシュ」を追加 「IM-Juggling を利用中にエラーが発生してしまう場合」を追加
2013-04-01	第4版 下記を追加・変更しました - intra-mart Accel Platform 2013 Spring (Climbing) のシステム要件に合わせて内容を変更 - テナント環境 の構築で「ポートレットの初期化」を不要のため削除 「LDAP認証設定ファイル(アカウントの認証にLDAP認証を利用する場合)」の説明を修正 「テナント環境セットアップ・サンプルデータセットアップに失敗した場合」の説明を修正 「外部メニュー連携」の説明を追加 「Resin でWARファイルのデプロイ中にエラーが発生する場合」の説明を修正 「データベースサーバ」にデータベースの権限に関するコラムを追加 「WARファイルのアンデプロイ」の説明を修正 「デプロイ直後の Web Application Server 起動時にエラーが発生する場合」を追加 「認可リソースグループ設定キャッシュ」の説明を追加 「メニュー側ルーティング設定キャッシュ」の説明を追加 「認可IPv4サブジェクト設定キャッシュ」の説明を追加 「グローバルナビキャッシュ」の説明を追加 「個人設定メニューキャッシュ」の説明を追加 「Windows サービスへの登録・削除」の説明を追加 「Linuxデーモンへの登録・削除」の説明を追加 「WARファイルの出力」にWARファイルに関するコラムを追加 「Apache HTTP Server」の設定方法を追加 「Apache Cassandra」の設定方法を追加 「プロジェクトの作成とモジュールの選択」にプロキシ設定に関するコラムを追加 「データベースサーバ」に複数テナントに関するコラムを追加 「データベースサーバ」にMicrosoft SQL Serverに関するコラムを追加 「モジュールのアップデート」追加しました。
2013-04-30	第5版 下記を追加・変更しました 「アップデート・パッチの適用・モジュール構成の変更」を追加 「プロジェクトの作成とモジュールの選択」にプロキシ設定に関する説明を変更
2013-07-01	第6版 下記を追加・変更しました 「Oracle Database」スキーマに付与する権限についてわかりづらい表現を改善 「テナント環境セットアップ」にコンテキストパスに関するコラムを追加 「コンソール起動・停止」に Windows Server 2012 利用時における注意点を追加 「Internet Information Services 7.5 が使用する .NET Framework バージョンの確認・変更方法」を追加 「.NET Framework のセットアップ」を追加 「統合Windows認証」に設定時における注意点を追加 「Resinの設定」にタイムゾーンの設定における注意点を追加 「Network」の設定方法を改善しました。 「WARファイルによる複数テナント」の設定方法を追加 「Linuxデーモンへの登録・削除」における実行ユーザの変更方法を追加

変更年月日	変更内容
2013-10-01	第7版 下記を追加・変更しました 「SAStruts版ポートレットが404エラーでアクセスできない場合」を追加 「Linux環境でWARファイルのデプロイ中にファイル入出力エラーが発生する場合」を追加 「Oracle WebLogic Server 12c (12.1.3) でテナント環境セットアップに失敗した場合」を追加 「プロジェクトの作成とモジュールの選択」の説明を修正 「DataSource」の説明を修正 「SAStruts用設定ファイル (SAStruts版ポートレットを利用する場合)」を追加 「ファイルのアップロードを制限する方法」を追加 「認可ポリシー設定キャッシュ」の説明を修正 「認可リソースグループ設定キャッシュ」の説明を修正 「メニュー側ルーティング設定キャッシュ」の説明を修正 「認可IPv4サブジェクト設定キャッシュ」の説明を修正 「グローバルナビキャッシュ」の説明を修正 「個人設定メニューキャッシュ」の説明を修正 「DataSourceマッピングの設定」に注意点を追加 「外部メニュー連携」に「メニュー設定画面での表示順序設定」に関する説明を追加 「Linuxデーモンへの登録」に「複数のIPアドレスが設定されている場合」に関する説明を追加 「Resinの設定」に作業ディレクトリ、変更検出に関する注意点を追加 「プロジェクトの作成とモジュールの選択」にTERASOLUNA Global Frameworkに関する説明を追加 「intra-mart Accel Platform の設定ファイル」にTERASOLUNA Global Frameworkに関する説明を追加 「TERASOLUNA Server Framework for Java (5.x) 用設定ファイル (シェードデータベースを利用する場合)」を追加 「TERASOLUNA Global Framework用設定ファイル (リポジトリ層にJPAを利用する場合)」を追加 「TERASOLUNA Global Framework用設定ファイル (リポジトリ層にMyBatisを利用する場合)」を追加 「設定ファイルの編集 (アップデートによる設定項目のメンテナンス)」を追加
2014-01-01	第8版 下記を追加・変更しました - 目次構成を改善しました。 - intra-mart Accel Platform 2013 Winter (Felicia) のシステム要件に合わせて内容を変更 「設定ファイルの編集 (アップデートによる設定項目のメンテナンス)」に「2013 Autumn(8.0.5) からアップデート」に関する説明を追加 「インポート処理結果ログの確認」にエラー原因の追跡についての説明を追記 「テキスト抽出設定 (solr-extractor-config.xml)」の説明を追加 「テナント環境セットアップ中にタイムアウトが発生した場合」を追加 「IM-Workflow 用設定ファイル (シーケンスオブジェクト探番機能を利用する場合)」を追加 「アップデート・パッチの適用・モジュール構成の変更」にCassandraのバージョンアップについての記載を追加 「Resin でWARファイルの再デプロイが正常にできない場合」を追加 「IMBox モジュールを外す方法」を追加 「iPadからアクセス時にクライアントタイプをPCとして扱う場合」を追加 「外部メニュー連携サービス for intra-mart WebPlatform v7.2」にIM-SecureSignOn for Accel Platform利用時の設定についての記載を追加 「アップデート・パッチの適用・モジュール構成の変更」に静的コンテンツ、テナント環境セットアップについて説明を追加 - 非推奨のため「Apache HTTP Server (mod_caucho.dll による設定)」を削除しました。 「Apache Solr」に「利用するJavaのバージョンについて」の説明を追加
2014-02-14	第9版 下記を追加・変更しました 「Internet Information Services (IIS)」の設定方法を改善 「統合Windows認証」の設定方法を変更

変更年月日	変更内容
2014-04-01	第10版 下記を追加・変更しました ■ 目次構成を改善しました。 ■ 「外部メニュー連携」のメニュープロバイダ情報の説明を修正、および複数テナント運用時のメニュープロバイダの設定についての説明を追加 ■ 「自動ログイン機能を利用する場合」を追加 ■ 「統合Windows認証」の設定方法を変更 ■ 「intra-mart Accel Platform の設定ファイル」にWARファイル展開時の注意点を追加 ■ 「intra-mart Accel Platform の設定ファイル」に「IM-Workflow システム設定」を追加 ■ 「プロジェクトの新規作成」に エディション毎のモジュールの選択方法を追加 ■ 「セットアップ実行結果ログの確認」のログ出力先情報を変更 ■ 「IMBox」に「intra-mart Accel Platform 2014 Spring(Granada)からの変更点」を追加 ■ 「統合Windows認証環境でWebサービスを利用する」を追加 ■ 「統合Windows認証環境で外部ソフトウェア連携機能を利用する」を追加 ■ 「認可IPv4サブジェクト設定キャッシュ」にキャッシュサイズの計算式を追加 ■ 「付録」に「テナント解決機能」を追加 ■ 「付録」に「Resinのクラスタリング」を追加 ■ 「付録」に「ポート一覧」を追加 ■ 「付録」に「IM-Juggling の応用」を追加 ■ 「付録」に「intra-mart Accel Platform のチューニング」を追加 ■ 「認可リソース閉塞情報キャッシュ」を追加
2014-05-01	第11版 下記を追加・変更しました ■ 「統合Windows認証」に「統合Windows認証機能を無効化するには」を追加
2014-08-01	第12版 下記を追加・変更しました ■ 「設定ファイルの編集(アップデートによる設定項目のメンテナンス)」の「2013 Winter(8.0.6) からアップデート」に「データベース検索時ケーブフラグ」に関する説明を追加 ■ 「設定ファイルの編集(アップデートによる設定項目のメンテナンス)」に「2014 Spring(8.0.7) からアップデート」に関する説明を追加 ■ 「付録」に「WARファイルに含まれるモジュール情報・ショートモジュールIDの一覧を確認する方法」を追加 ■ 「付録」に「初回アクセス時に「[E.IWP.ADMIN.CONTEXT.10004] Tenant ID cannot be resolved.」が発生します。」を追加 ■ 「Apache Solr」の記載を「Solr管理者ガイド」-「Solrのセットアップ」に移動 ■ 「iAP-iWP間SSO連携(IM-HybridSSO)」の説明を追加 ■ 「外部メニュー連携」に IM-HybridSSO についての説明を追加
2014-09-01	第13版 下記を追加・変更しました。 ■ 「iAP-iWP間SSO連携(IM-HybridSSO)」に iWP patch8 適用時の注意喚起を追加。
2014-12-01	第14版 下記を追加・変更しました。 ■ 「ログインセッション管理」の設定方法を追加 ■ 「テナント環境情報」に「グローバルナビ最大表示数」を追加 ■ 「統合Windows認証」に「統合Windows認証機能の認証失敗時に通常のログイン機能を利用するには」を追加 ■ 「設定ファイルの編集(アップデートによる設定項目のメンテナンス)」に「2014 Summer(8.0.8) からアップデート」に関する説明を追加 ■ 「ヘルプド롭ダウンキャッシュ」の説明を追加 ■ 「WARファイル作成時の実行環境の変更」の説明を追加 ■ 「IM-Notice」の設定方法を追加 ■ 「Internet Information Services(IIS)」に「OAuth認証モジュールを利用する場合の追加設定」を追加

変更年月日	変更内容
2015-04-01	第15版 下記を追加・変更しました。 ■ 「ライセンスの登録」を追加 ■ 「統合Windows認証」に Internet Information Services (IIS) が必須である旨の注意を追加 ■ 「統合Windows認証」に 利用できるブラウザの説明を追加 ■ 「インターネットに接続できない環境で IM-Juggling を利用する場合」を追加 ■ 「プロキシ設定が必要な環境で IM-Juggling を利用する場合」を追加 ■ 「アップデート・パッチの適用・モジュール構成の変更」に「モジュール構成の変更」を追加 ■ 「バックアップ・リストア(復元)」に各対象ファイルの詳細と、復元手順の注意を追加 ■ 「Resin でWARファイルのデプロイ中にエラーが発生する場合」に「dependency-check-interval」の値の単位について説明を追加 ■ 「PostgreSQL」に PostgreSQL 9.4 を追加 ■ 「ベースURL」の説明を修正 ■ 「設定ファイルの編集(アップデートによる設定項目のメンテナンス)」に「2014 Winter(8.0.9) からアップデート」に関する説明を追加 ■ 「IM-Juggling を利用せず、CUIでWARファイルを作成する方法」に静的ファイルの出力方法を追記 ■ 「IM-ContentsSearch」の「Solrサーバ接続設定(solr-config.xml)」にポート番号に関する説明を追加 ■ Terasoluna Global FrameworkをTerasoluna Server Framework for Java (5.x)に変更 ■ 「Terasoluna Global Framework用設定ファイル (リポジトリ層にJPAを利用する場合)」を削除 ■ 「Terasoluna Global Framework用設定ファイル (リポジトリ層にMyBatisを利用する場合)」を削除 ■ 「Terasoluna Server Framework for Java (5.x) 用設定ファイル (リポジトリ層にMyBatis3を利用する場合)」を追加 ■ 「Terasoluna Server Framework for Java (5.x) on Accel Platform を使用する場合の設定」を追加 ■ 「Locale」の システムロケール変更に関する説明を修正

はじめに

本書の目的

本書では intra-mart Accel Platform のセットアップ手順について説明します。

前提条件

- リリースノートに記載されているシステム要件を満たしている必要があります。
詳細は「[リリースノート](#)」を参照してください。
- 本書は Resin に関するセットアップ内容となっています。



コラム

WebSphere Application Server 8.5.5 については、「[セットアップガイド for WebSphere](#)」を参照してください。
Oracle WebLogic Server 12c (12.1.3) については、「[セットアップガイド for WebLogic](#)」を参照してください。

- intra-mart Accel Platform で利用するポート番号については、「[付録](#)」 - 「[ポート一覧](#)」を参照してください。

対象読者

本書は、運用環境を想定したセットアップについて説明しています。

intra-mart Accel Platform を体験されたい方、はじめて intra-mart Accel Platform をセットアップされる方は、intra-mart Accel Platform の最小構成 (スタンドアローン) での簡易的なセットアップについて説明している「[クイック セットアップガイド](#)」を参照してください。

用語解説

Resin をインストールしたディレクトリを %RESIN_HOME% と略します。

Apache HTTP Server をインストールしたディレクトリを %APACHE_HOME% と略します。

Storage として使用するディレクトリを %STORAGE_PATH% と略します。

PublicStorage として使用するディレクトリを %PUBLIC_STORAGE_PATH% と略します。

Webサーバ利用時の静的コンテンツを配置するディレクトリを %WEB_PATH% と略します。

セットアップの流れ

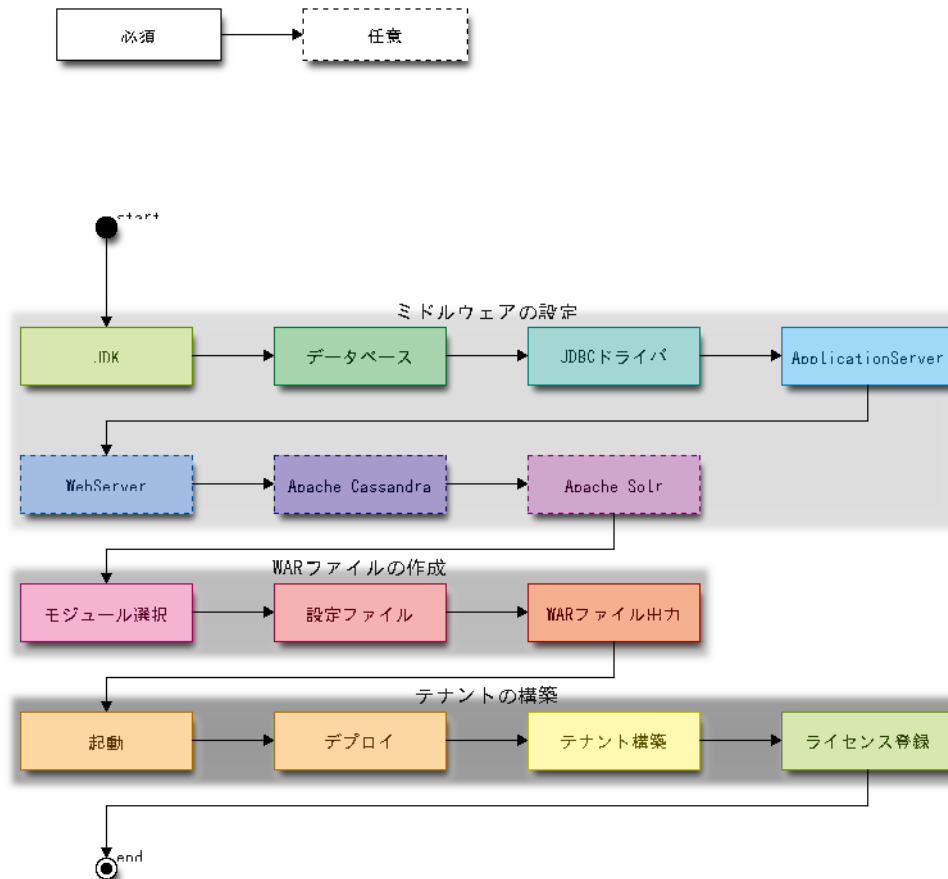
セットアップの流れは次の通りです。

各ステップごとのセットアップ手順は一覧のリンク先を参照してください。

凡例

必須・・・セットアップが必要な項目です。

任意・・・セットアップをスキップする事ができる項目です。



Name	Description
start	
JDK	JDK
データベース	データベースサーバ
JDBCドライバ	JDBCドライバ
ApplicationServer	Web Application Server
WebServer	Web Server
Apache Cassandra	Apache Cassandra
Apache Solr	Apache Solr
モジュール選択	WARファイルの作成 - プロジェクトの新規作成
設定ファイル	WARファイルの作成 - intra-mart Accel Platform の設定ファイル
WARファイル出力	WARファイルの作成 - WARファイルの出力
起動	Web Application Server の起動・停止
デプロイ	WARファイルのデプロイ
テナント構築	テナント環境セットアップ
ライセンス登録	ライセンスの登録
end	

ミドルウェアのセットアップ

intra-mart Accel Platform のセットアップに必要な以下のインストールおよび設定を行います。

JDK

intra-mart Accel Platform を利用するには、JDK が必要です。
Resin の場合、Oracle Javaを入手し、セットアップを行います。以下の入手先を参考にしてください。
Resin 以外の Web Application Server の場合、各製品のドキュメントを参照してください。

Oracle Java (JDK)

<http://www.oracle.com/technetwork/java/javase/downloads/index.html> (English)

データベース

データベース のセットアップに必要な以下のインストールおよび設定を行います。

データベースサーバ

項目

- データベース のインストール
 - 各 データベース 作成時の注意点
 - Oracle Database
 - Microsoft SQL Server
 - IBM DB2
 - PostgreSQL
 - PostgreSQL のインストール
 - データベースとログインロールの作成

データベース のインストール

- intra-mart Accel Platform は、データベースが必要となります。
ご利用になる データベース をインストールしてください。インストールの詳細は各製品元のドキュメントを参照してください。
本書では、PostgreSQL (Windows版)を一例として説明します。



注意

データベース のエンコーディングは、UTF-8 で作成する必要があります。



注意

データベース に対してデータの登録・更新・削除を行う権限に加え、テーブルやVIEWを作成する権限も必要となります。
具体的な設定方法については、各データベースベンダから提供されているドキュメントを参照してください。



コラム

複数テナントを構築する場合、接続先のデータベースはインスタンス単位で分ける事を推奨します。

各 データベース 作成時の注意点

Oracle Database

- intra-mart Accel Platform を利用するには、最低限「CONNECT」・「RESOURCE」・「CREATE VIEW」・「UNLIMITED TABLESPACE」権限が付与されている必要があります。
その他利用するアプリケーション、エクステンションシリーズによって必要な権限があります。
詳細については、各製品のリリースノートおよび、セットアップガイドを参照してください。



注意

IM-Workflow を利用する場合、テナントデータベースとして接続するユーザのデフォルト表領域は、自動セグメント領域管理(ASSM)を有効にする必要があります。
IM-Workflow モジュールでは、2013 Summer にて トランザクションデータの保存先としてデータベースを選択出来るようになりました。
この対応で、テナント環境セットアップ時にBLOB データ型カラムを持つテーブルを作成しています。
対象の BLOB データ型カラムには、パフォーマンスの改善を目的として、SECUREFILE パラメータを指定しています。
この場合、対象のカラムは自動セグメント領域管理(ASSM)で管理される表領域でのみ作成できます。

Microsoft SQL Server

- 作成するデータベースの既定の照合順序は、データベース をインストールしたプラットフォーム(OS)の
コントロールパネル・[地域と言語のオプション]・[詳細設定] タブで Windows システム ロケール名を確認します。
既存の Windowsロケールでの照合順序設定に対応する照合順序指定子を指定します。
各Windows 照合順序は大文字小文字、アクセント、文字幅、かなの区別を定義する一連のサフィックスとして組み合わせることが可能となります。
運用される環境において、サフィックスの組み合わせを設定する必要があります。



コラム

intra-mart Accel Platform では、大文字・小文字を区別する設定でデータベースを作成することを推奨します。
推奨する設定値「Japanese_90_CS_AS_KS_WS」



注意

IM-Workflowを利用する場合、「テーブル スキャンによるロック待ち」が発生する可能性があります。
ロック待ちを回避するための方法のひとつとして「READ_COMMITTED_SNAPSHOT」の利用が挙げられます。
利用の是非を検討のうえ、必要に応じて利用してください。
※「テーブル スキャンによるロック待ち」ならびに「READ_COMMITTED_SNAPSHOT」については、
詳しくは Microsoft SQL Server のドキュメントを参照してください。

[READ_COMMITTED_SNAPSHOT の利用方法]

利用するには、以下のようにデータベースに対して READ_COMMITTED_SNAPSHOT を ON に設定する必要があります。

- ALTER DATABASE {data_base_name} SET READ_COMMITTED_SNAPSHOT ON;

※ データベースへの接続コネクションがすべて終了している状態で実行してください。

IBM DB2

- データベース作成時、「デフォルトのバッファ・プールと表スペースのページ・サイズ」に
「32k」を設定する必要があります。

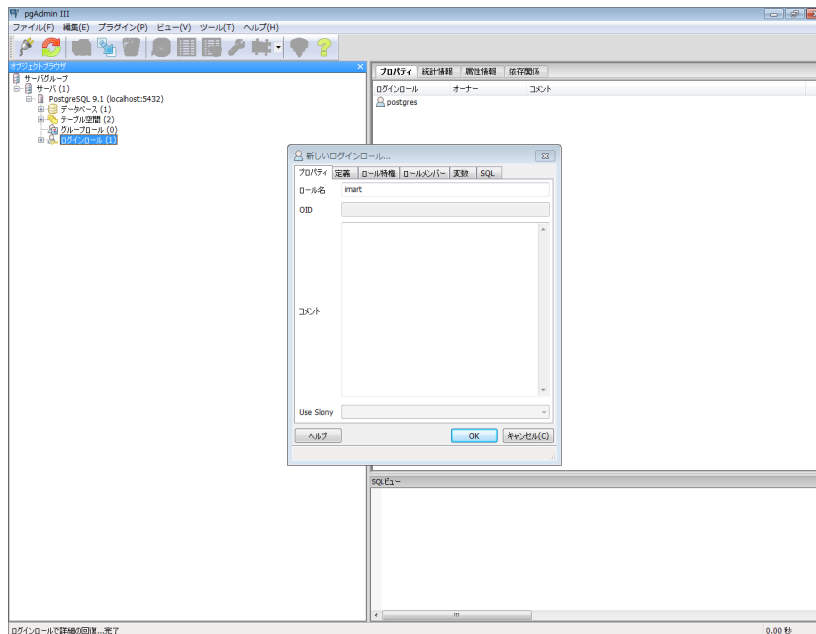
PostgreSQL

PostgreSQL のインストール

- PostgreSQL は次のURLよりダウンロードできます。
環境に適したものをダウンロード後、インストールを行ってください。
<http://www.enterprisedb.com/products-services-training/pgdownload>

データベースとログインロールの作成

- pgAdmin ツールを起動します。
- ログインロールの作成
「オブジェクトブラウザ」から、「ログインロール」を右クリック、「新しいログインロール」をクリックします。
サブウィンドウが表示されます。



次の項目を入力し「OK」をクリックします。

プロパティタブ

「ロール名」(任意)

定義タブ

「パスワード」(任意)



コラム

本書では、例として次の内容を指定します。

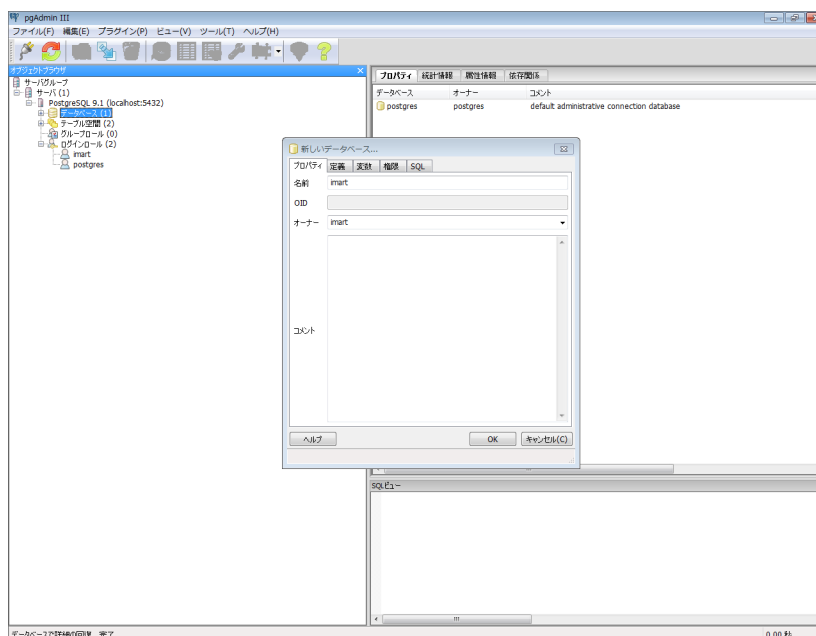
ロール名「imart」

パスワード「imart」

- データベースの作成

「オブジェクトブラウザ」から、「データベース」を右クリック、「新しいデータベース」をクリックします。

サブウィンドウが表示されます。



次の項目を入力、選択し「OK」をクリックします。

プロパティタブ
[名前] (任意)
[オーナー] (上記「2. ログインロールの作成」で作成したログインロール)



コラム

本書では、例として次の内容を指定します。

名前「iap_db」

オーナー「imart」

JDBCドライバ

項目

- JDBCドライバの入手
 - PostgreSQL
 - Oracle Database
 - Microsoft SQL Server
 - IBM DB2

JDBCドライバの入手

ご利用になる データベース が指定するJDBCドライバを入手します。以下の入手先を参考にしてください。



注意

intra-mart Accel Platform が利用するJDBCドライバは **JDBC 4** です。JDBC 4.1 は未検証となります。

PostgreSQL

1. PostgreSQL のJDBCドライバは以下のURLよりダウンロードできます。
PostgreSQL 9.2 の場合は、<postgresql-9.2-1004.jdbc4.jar> ファイルを入手します。
<http://jdbc.postgresql.org/download/postgresql-9.2-1004.jdbc4.jar>
PostgreSQL 9.3 の場合は、<postgresql-9.3-1100.jdbc4.jar> ファイルを入手します。
<http://jdbc.postgresql.org/download/postgresql-9.3-1100.jdbc4.jar>
PostgreSQL 9.4 の場合は、<postgresql-9.4-1200.jdbc4.jar> ファイルを入手します。
<https://jdbc.postgresql.org/download/postgresql-9.4-1201.jdbc4.jar>

Oracle Database

1. Oracle のJDBCドライバは以下のURLよりダウンロードできます。
<http://www.oracle.com/technetwork/indexes/downloads/index.html>
2. <ojdbc6.jar> ファイルを入手します。

Microsoft SQL Server

1. Microsoft SQL Server のJDBCドライバは以下のURLよりダウンロードできます。
<http://www.microsoft.com/ja-jp/download/details.aspx?id=11774>
2. <sqljdbc4.jar> ファイルを入手します。

IBM DB2

1. IBM DB2 に付属の DB2 JDBC Driver Type4 をインストールしてください。
詳しくは、DB2 のドキュメントを参照してください。
2. <db2jcc4.jar> ファイルを入手します。



コラム

弊社ではデータベース接続に利用するJDBCドライバは、Type4 (非XA) にて検証を実施しています。

Web Application Server

Resin (Windows版) のインストール

項目

- インストール
- JDBCドライバの配置

インストール

同梱されているディレクトリの <resin-pro-4.0.43.zip> ファイルを任意のパスに展開します。



注意

Resin を起動するためには「.NET Framework 3.5」が必要です。
Windows Server 2012 環境では、「.NET Framework 4.5」のみインストールされているため、
「.NET Framework 3.5」のセットアップを行ってください。
詳細は、[.NET Framework のセットアップ](#)を参照してください。

JDBCドライバの配置

1. JDBCドライバをダウンロードします。

JDBCドライバの入手先は以下を参照してください。

[JDBCドライバ](#)

2. ダウンロードした、JDBCドライバを以下のディレクトリにコピーします。

```
<%RESIN_HOME%/lib>
```

Resin (Linux版) のインストール

項目

- インストール
- JDBCドライバの配置

インストール

1. 同梱されているディレクトリの <resin-pro-4.0.43.tar.gz> ファイルを任意のパスに配置します。
2. ファイルを展開します。

```
# tar xzvf resin-pro-4.0.xx.tar.gz
```

3. Resin のソースを展開したディレクトリから、以下を実行します。

```
# ./configure --prefix=<%RESIN_HOME%>
# make
# make install
```

 注意

ディストリビューションによってコンパイルオプションを指定する必要があります。
 具体的にどのオプションが必要になるかは、ディストリビューションによって異なります。
 ./configure コマンドの実行結果を確認し、必要なオプションを指定してください。

64bit環境上で Resin のコンパイルを行う際、./configureを呼び出す際に以下のオプションがLinux環境によって必要となります。

```
# ./configure --prefix=<%RESIN_HOME%> --enable-64bit
```

 コラム

make installを実行する際の実行ユーザがroot権限を持っている場合、
 /etc/init.dディレクトリ配下に起動用スクリプトresinが作成されます。

 注意

intra-mart Accel Platform 上の画面より画像を表示した際、マルチバイト文字が文字化けする場合は、「JIS2004 対応フォント」をインストールしてください。

1. 「IPA フォント」をダウンロード後、ZIPファイルを解凍します。
2. 拡張子「.ttf」のフォントファイルを、<%JAVA_HOME%/jre/lib/fonts> ディレクトリにコピーします。
3. <%STORAGE_PATH%/system/fontconfig/jre/lib/fontconfig.properties.linux>ファイルを
 <%JAVA_HOME%/jre/lib/fontconfig.properties>として保存します。

または、

<%JAVA_HOME%/jre/lib/fontconfig.properties.src> を <fontconfig.properties> にリネームし、以下の設定を追加して保存します。

```

(先頭)
#
# Copyright (c) 2003, Oracle and/or its affiliates. All rights reserved.
#
# fontconfig.properties using "IPA font"

##### 追加ここから #####
# The IPA font is an outline font in accordance with JIS X 0213:2004.
# URL -> http://ossipedia.ipa.go.jp/ipafont/
# IPA gothic : ipag.ttf
# IPA P gothic : ipagp.ttf
# IPA UI gothic : ipagui.ttf
# IPA mincho : ipam.ttf
# IPA P mincho : ipamp.ttf
##### 追加ここまで #####

# Version

.
.
.

##### 追加ここから #####
# Font File Names
filename.b&h-lucidasans-medium-r-normal-sans-%d-*-*p-iso8859-1=$JRE_LIB_FONTS/ipagui.ttf
filename.b&h-lucidabright-medium-r-normal-%d-*-*p-iso8859-1=$JRE_LIB_FONTS/ipagui.ttf
filename.b&h-lucidabright-demibold-r-normal-%d-*-*p-iso8859-1=$JRE_LIB_FONTS/ipagui.ttf
filename.b&h-lucidabright-medium-i-normal-%d-*-*p-iso8859-1=$JRE_LIB_FONTS/ipagui.ttf
filename.b&h-lucidabright-demibold-i-normal-%d-*-*p-iso8859-1=$JRE_LIB_FONTS/ipagui.ttf
filename.b&h-lucidasans-bold-r-normal-sans-%d-*-*p-iso8859-1=$JRE_LIB_FONTS/ipagui.ttf
filename.b&h-lucidasans-medium-i-normal-sans-%d-*-*p-iso8859-1=$JRE_LIB_FONTS/ipagui.ttf
filename.b&h-lucidasans-bold-i-normal-sans-%d-*-*p-iso8859-1=$JRE_LIB_FONTS/ipagui.ttf
filename.b&h-lucidatypewriter-medium-r-normal-sans-%d-*-*m-iso8859-1=$JRE_LIB_FONTS/ipagui.ttf
filename.b&h-lucidatypewriter-bold-r-normal-sans-%d-*-*m-iso8859-1=$JRE_LIB_FONTS/ipagui.ttf
filename.b&h-lucidatypewriter-medium-i-normal-sans-%d-*-*m-iso8859-1=$JRE_LIB_FONTS/ipagui.ttf
filename.b&h-lucidatypewriter-bold-i-normal-sans-%d-*-*m-iso8859-1=$JRE_LIB_FONTS/ipagui.ttf
##### 追加ここまで #####
(末尾)

```

4. <%JAVA_HOME%/jre/lib> ディレクトリ内の有効な <fontconfig.OS.Version.properties>、および <fontconfig.OS.Version.bfc> をリネームします。

例)

- ・ fontconfig.RedHat.bfc → fontconfig.RedHat.bfc.back
- ・ fontconfig.Sun.bfc → fontconfig.Sun.bfc.back
- ・ fontconfig.bfc → fontconfig.bfc.back

JDBCドライバの配置

1. JDBCドライバをダウンロードします。

JDBCドライバの入手先は以下を参照してください。

[JDBCドライバ](#)

2. ダウンロードした、JDBCドライバを以下のディレクトリにコピーします。

```
<%RESIN_HOME%/lib>
```

Resinの設定

項目

- スタンドアローン／分散構成共通
- 分散構成
- [TERASOLUNA Server Framework for Java \(5.x\) on Accel Platform](#) を使用する場合の設定

スタンドアローン／分散構成共通

1. <%RESIN_HOME%/conf/resin.properties> ファイルを開きます。
2. 「jvm_args」プロパティに、インストール環境に応じたメモリ値、ヒープの最大サイズを設定します。
本書では、例として次の値を設定します。

```
# Arg passed directly to the JVM
jvm_args : -Xmx1024m -XX:MaxPermSize=256m
```

! 注意

設定する値は「1024m」以上にしてください。
「1024m」より小さい値を設定した場合、正常に動作しない恐れがあります。

i コラム

「jvm_args」プロパティに「-Djava.io.tmpdir」オプションを追加し、Resin が利用する作業ディレクトリを変更できます。
このオプションが指定されていない場合、Resin が利用する作業ディレクトリはJVMのデフォルトの設定が利用されます。
以下の例では、作業ディレクトリを /var/resin-tmp に変更する場合の設定例です。

```
# Arg passed directly to the JVM
jvm_args : -Xmx1024m -XX:MaxPermSize=256m -Djava.io.tmpdir=/var/resin-tmp
```

! 注意

- ・「-Djava.io.tmpdir」オプションにより指定されたディレクトリは事前に作成しておく必要があります。
- ・Resin 実行ユーザが読み込み、書き込みを行うことが出来る権限を設定しておく必要があります。
- ・Linux系の環境では、このオプションが未指定の場合 /tmp が利用されます。
cron等の設定により定期的に/tmp配下の内容が削除される設定が標準で組み込まれている場合があります。
- ・Resin では、作業用ディレクトリに展開した設定ファイル等の変更を検出した場合、
自動的に再起動が行われる場合があります。
その為、Resin が利用する作業ディレクトリを変更しておくことを推奨します。

i コラム

「jvm_args」プロパティに「-Duser.timezone」オプションを追加し、タイムゾーンを変更できます。
タイムゾーンは、タイムゾーンマスタに存在するものを設定してください。
タイムゾーンマスタについては「[設定ファイルリファレンス - タイムゾーンマスタ](#)」を参照してください。
本書では、例としてタイムゾーンを"UTC"として設定します。

```
# Arg passed directly to the JVM
jvm_args : -Xmx1024m -XX:MaxPermSize=256m -Duser.timezone=UTC
```

! 注意

タイムゾーン設定の初期設定は、インストールを行った JDK に依存します。

! 注意

この設定は運用開始前に行ってください。
運用開始後に変更した場合、データベースに登録される日時データに影響があります。
データベースへ登録する日時データは、インストールを行った JDK のタイムゾーンに依存します。
変更前と変更後のタイムゾーンが相違する場合、タイムゾーン変換によって異なる日時データが登録されてしまいます。

i コラム

Resin は、起動後に設定ファイルやプログラムの変更を検出した場合、Resin 自身の再起動を行います。
開発中、運用中等により用途は異なりますが、Resin に変更の検出を行わせたくない場合は、「dependency_check_interval」項目の設定を行います。
下記は、変更の検出を行わない設定例です。

```
dependency_check_interval : -1
```

「dependency_check_interval」は、2s (2秒毎), 5m (5分毎)等の値が設定可能です。

変更が完了したら、Resin を再起動してください。

i コラム

Resin の起動時にポートが重複している旨のエラーが発生する場合、Resin の停止後、ポート番号の設定を変更してください。

```
java.lang.RuntimeException: java.net.BindException: Address already in use: JVM_Bind
    Can't bind to *:8080.
    Check for another server listening to that port.
```

「app.http」プロパティに設定されている「8080」を、「8081」などの使用されていないポート番号に変更してください。

```
# Set HTTP and HTTPS ports.
# Use overrides for individual server control, for example: app-0.http : 8081
app.http      : 8081
```

変更が完了したら、Resin を再起動してください。

分散構成

1. <%RESIN_HOME%/conf/resin.properties> ファイルを開きます。
2. 「app_servers」プロパティに分散構成として構築したResinサーバの情報を設定します。
例としてResinを3台構成で運用する場合、次の値を設定します。

```
# app-tier Triad servers: app-0 app-1 app-2
app_servers    : 192.168.100.1:6800 192.168.100.2:6800 192.168.100.3:6800
```

「app_servers」プロパティに設定した「IPアドレス:ポート番号」は、記述した順に app-0, app-1, app-2 とIDが振られます。



注意

分散構成に含まれるResinの<%RESIN_HOME%/conf/resin.properties>ファイルは、すべて同じ内容にしてください。
「app_servers」プロパティに指定するIPアドレスは、ローカルアドレスのみ指定が可能であり、グローバルアドレスの指定を行うことはできません。
また、分散環境を構築する場合、127.0.0.1等のループバックアドレスが指定された場合正常に動作しません。

TERASOLUNA Server Framework for Java (5.x) on Accel Platform を使用する場合の設定

1. <%RESIN_HOME%/lib>から以下のjarファイルを削除します。
 - hibernate-validator-4.3.0.Final.jar
 - validation-api-1.0.0.GA.jar



コラム

TERASOLUNA Server Framework for Java (5.x)では、Hibernate Validatorのバージョンが5.1.3に上がったので、競合するバージョンのjarを削除します。



コラム

Resin のクラスタリングについては、「[Resinのクラスタリング](#)」を参照してください。



注意

Resin の分散環境を構築する場合には製品版に同梱されている Resin が必要となります。
検証目的などで分散環境を構築されたい場合は、弊社営業までお問い合わせください。

Web Server

intra-mart Accel Platform は、Web Server を利用する事ができます。
ご利用になる Web Server をインストールしてください。インストールの詳細は各 Web Server のドキュメントを参照してください。

Apache HTTP Server

項目

- [mod_proxy/mod_rewriteの利用](#)
- [mod_proxy/mod_rewriteの取得](#)
- [静的ファイルの配置](#)
- [apache設定ファイルの編集](#)
- [アクセスログの編集](#)
- [制限事項](#)

[mod_proxy/mod_rewriteの利用](#)

ここではResinの組み込みモジュールを利用せずApacheのモジュール(mod_proxy/mod_rewrite)を利用して、webサーバとintra-mart Accel Platformの連携を行う場合の設定を行います。

[mod_proxy/mod_rewriteの取得](#)

製品元のマニュアルを参照してください。

[静的ファイルの配置](#)

IM-Juggling から取得した「静的ファイル」をapacheのドキュメントルートに解凍します。
静的ファイルの出力は、[静的ファイルの出力](#)を参照してください。

[apache設定ファイルの編集](#)

1. <%APACHE_HOME%/conf/httpd.conf> ファイルを開きます。
2. Dynamic Shared Object (DSO) Supportエリアに以下の設定を追加またはコメントアウトしてください。

```
LoadModule rewrite_module modules/mod_rewrite.so
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so
```

3. 末尾にAllowEncodedSlashesディレクティブ、mod_proxyおよびmod_rewriteの設定を追加してください。

コンテキストパス	imart
apacheのドキュメントルート	/usr/local/apache/htdocs
静的ファイルの展開フォルダ	/usr/local/apache/htdocs/imart
IPアドレス	192.168.1.1
ポート番号	8080

上記の例である場合の設定は以下の通りです。

```
AllowEncodedSlashes On
RewriteEngine On
RewriteCond %{REQUEST_URI} ^/imart/(.*\.(gif|.*\.(GIF)$ [OR]
RewriteCond %{REQUEST_URI} ^/imart/(.*\.(png|.*\.(PNG)$ [OR]
RewriteCond %{REQUEST_URI} ^/imart/(.*\.(jpg|.*\.(JPG)$ [OR]
RewriteCond %{REQUEST_URI} ^/imart/(.*\.(css|.*\.(CSS)$ [OR]
RewriteCond %{REQUEST_URI} ^/imart/(.*\.(js|.*\.(JS)$ [OR]
RewriteCond %{REQUEST_URI} ^/imart/(.*\.(swf|.*\.(SWF)$ [OR]
RewriteCond %{REQUEST_URI} ^/imart/(.*\.(ico|.*\.(ICO)$ [OR]
RewriteCond %{REQUEST_URI} ^/imart/(.*\.(json|.*\.(JSON)$ [OR]
RewriteCond %{REQUEST_URI} ^/imart/(.*\.(jar|.*\.(JAR)$ [OR]
RewriteCond %{REQUEST_URI} ^/imart/(.*\.(xml|.*\.(XML)$ [OR]
RewriteCond %{REQUEST_URI} ^/imart/(.*\.(yaml|.*\.(YAML)$ [OR]
RewriteCond %{REQUEST_URI} ^/imart/(.*\.(txt|.*\.(TXT)$ [OR]
RewriteCond %{REQUEST_URI} ^/imart/(.*\.(html|.*\.(HTML|.*\.(htm|.*\.(HTM)$
RewriteRule ^/imart/(.*)?$ /imart/$1

ProxyPreserveHost On
ProxyPass /imart/ http://192.168.1.1:8080/imart/ nocanon
ProxyPassReverse /imart/ http://192.168.1.1:8080/imart/ nocanon
```



コラム

上記に指定した拡張子は、intra-mart Accel Platform Advanced版+intra-mart Accel Collaboration+IM-FormaDesigner for Accel Platformで静的ファイルとして扱う必須の拡張子になります。
上記RewriteCondにない静的ファイルを取り扱いたい場合には、RewriteCondの記述を追加してください。



注意

APサーバのコンテキストルートに必ずリクエストに含むようにしてください。
APサーバ側 `http://myapp/imart` であれば、リクエストのURLは `http://myweb/imart` にして下さい。

4. Apacheを再起動してください。

[アクセスログの編集](#)

リバースProxy の場合、Resinが出力するaccess.logのソースIPがすべて127.0.0.1になってしまいます。これを回避するためにaccess.logの出力内容を編集します。

1. <%RESIN_HOME%/conf/resin.xml> ファイルを開きます。
2. <host id="" root-directory=""> ディレクティブ内に以下の設定を追加してResinの再起動を行ってください。

```
<access-log path='log/access.log'>
  <rollover-period>1D</rollover-period>
  <format>%(X-Forwarded-For)i %h %l %u %t "%r" %>s %b "%{Referer}i" "%{User-Agent}i"</format>
</access-log>
```



コラム

access-logについての詳細な設定内容については「[Resin Documentation](#)」を参照してください。

制限事項

制限事項については「[リリースノート](#)」を参照してください。

Internet Information Services (IIS)

項目

- 前提条件
 - Application Request Routing のセットアップ
- 設定
- OAuth認証モジュールを利用する場合の追加設定
- 静的ファイルの配置
- アクセスログの編集
- 動作確認

前提条件

- 「システム要件」に記載の Windows Server を使用していること。
 - 以降の手順では Windows Server 2012 のキャプチャ画像を使用しますが、上記のどのバージョンの Windows Server でも同じ手順でセットアップできます。
- ベースURLの設定
Internet Information Services (IIS) を使用する場合ベースURLの設定が必要です。

i コラム

- ベースURLには <httpもしくはhttps>://<IISのIPアドレス>:<IISのポート番号>/<コンテキストパス> を設定します。
 - 構築するサーバ環境に応じて適切に設定してください。
 - ベースURLについての詳細は、[ベースURL](#)を参照してください。

- Internet Information Services (IIS) を稼働させるには「Application Request Routing」が必要です。
下記「Application Request Routing」のホームページよりインストーラをダウンロードし、以下の手順でセットアップを行ってください。

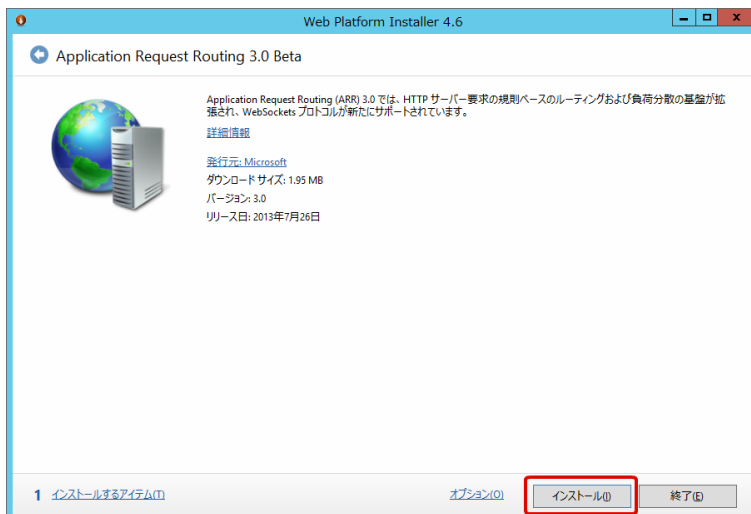
i コラム

URL (2014年8月現在)

- Application Request Routing ホームページ <http://www.iis.net/downloads/microsoft/application-request-routing>
- Application Request Routing 3.0 インストーラ http://www.microsoft.com/web/handlers/webpi.ashx?command=getinstaller&appid=ARRv3_0

Application Request Routing のセットアップ

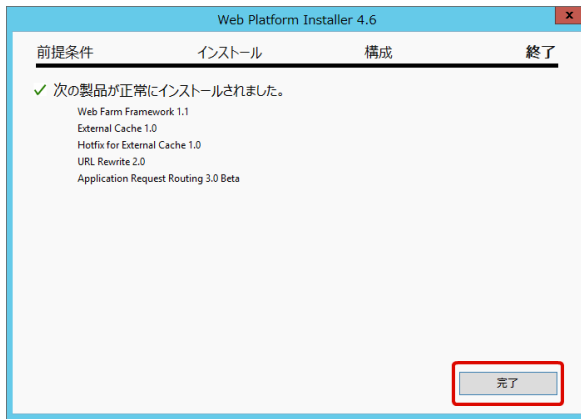
1. ダウンロードした <ARRv3_0.exe> を実行します。
2. 「インストール」をクリックします。



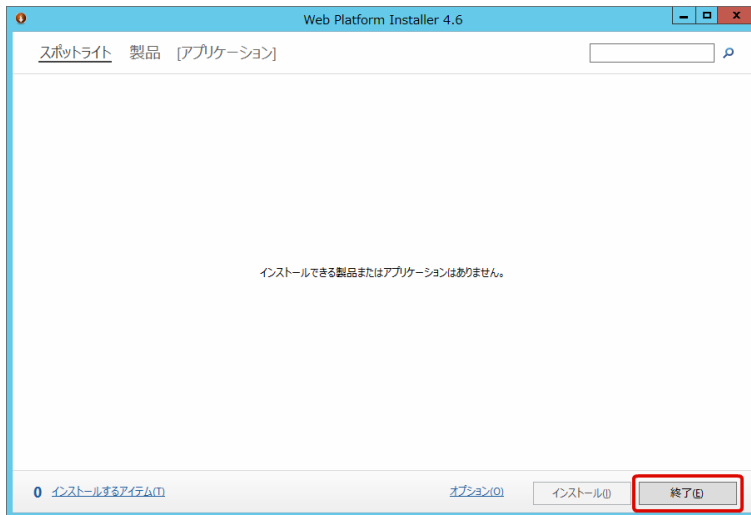
3. 前提条件、ライセンス条項を確認後、「同意する」をクリックします。



4. 「完了」をクリックします。



5. 「終了」をクリックします。



以上で Application Request Routing のセットアップは終了です。

設定

- Resin へ接続するための設定を行います。
以降の手順では下記の環境を想定した手順を記載します。

intra-mart Accel Platform のコンテキストパス imart

IIS のルート ディレクトリ	C:\inetpub\wwwroot
-----------------	--------------------

静的ファイルの展開フォルダ	C:\imart
---------------	----------

Resin の IP アドレス	192.168.1.1
-----------------	-------------

Resin のポート番号	8080
--------------	------

- Internet Information Services (IIS) と Resin を接続するための設定ファイル <web.config> を <C:\inetpub\wwwroot> 直下に作成します。

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <system.web>
    <httpRuntime maxRequestLength="102400" requestPathInvalidCharacters="" />
    <customErrors mode="Off" />
  </system.web>
  <system.webServer>
    <security>
      <requestFiltering>
        <requestLimits maxAllowedContentLength="104857600" />
      </requestFiltering>
    </security>
    <httpErrors errorMode="Detailed" />
    <rewrite>
      <rules>
        <clear />
        <rule name="ReverseProxyInboundStatic" stopProcessing="true">
          <match url="^imart/(.gif|.png|.jpg|.css|.js|.swf|.ico|.json|.jar|.xml|.yaml|.txt|.html|.htm)$" />
          <action type="None" />
        </rule>
        <rule name="ReverseProxyInboundDynamic" stopProcessing="true">
          <match url="^imart/(.*)" />
          <action type="Rewrite" url="http://192.168.1.1:8080/imart/{R:1}" />
        </rule>
      </rules>
    </rewrite>
  </system.webServer>
</configuration>
```

コラム

web.config とは Internet Information Services (IIS) の構成ファイルです。

注意

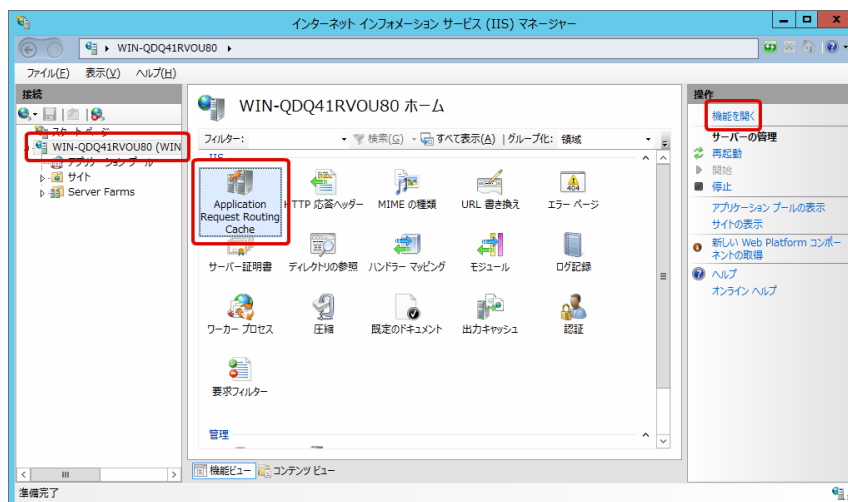
web.config ファイルは IM-Juggling 上では編集できません。

- サーバ環境に合わせて web.config 内の下記の要素等を適宜変更します。
 - maxRequestLength, maxAllowedContentLength 属性
 - アップロードのファイルサイズの制限を変更する場合には、maxRequestLength, maxAllowedContentLength 属性の設定を変更してください。(上記の例では 10

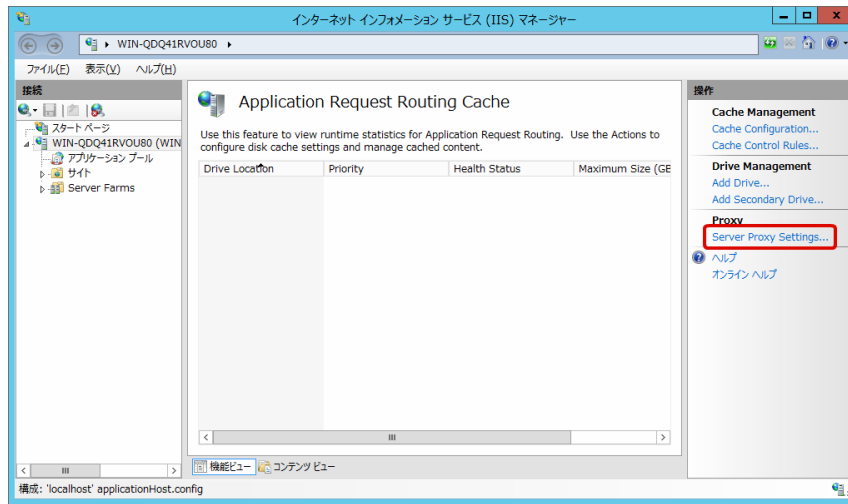
コラム

web.config の詳細は以下のリファレンスを参照してください。
<http://technet.microsoft.com/ja-jp/library/ee431610.aspx>

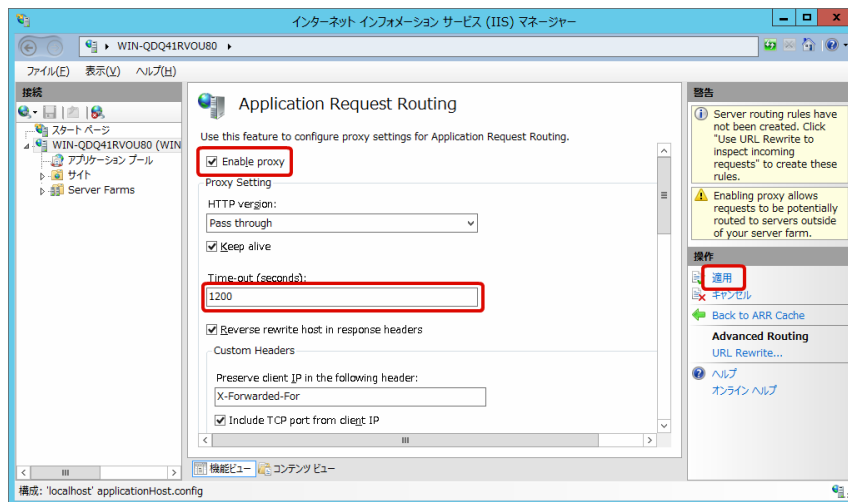
- インターネット インフォメーション サービス (IIS) マネージャー より「IIS サーバ(ここでは WIN-QDQ41RVOU80)」「Application Request Routing Cache」「機能を開く」の順



- 「Server Proxy Settings...」をクリックし、「Application Request Routing」設定画面を表示します。



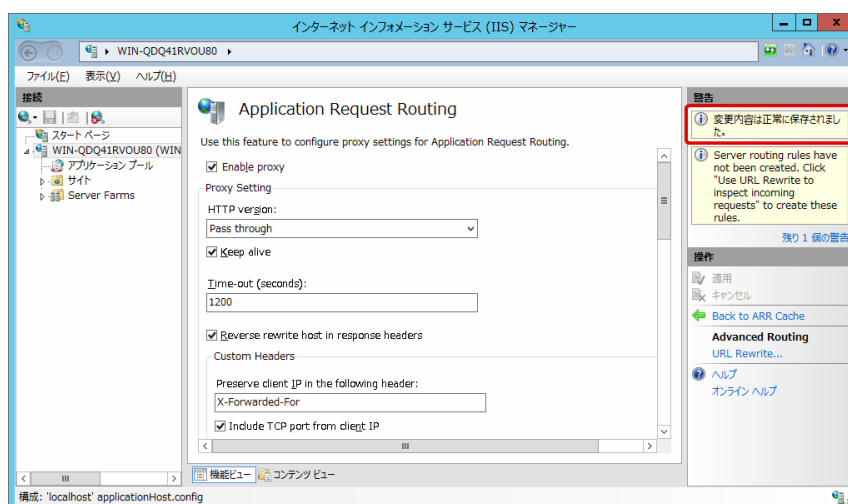
5. 「Enable Proxy」をチェックし、「Time-out (seconds)」にタイムアウト値を秒単位で設定します。
さらに、「適用」をクリックし、設定を反映します。



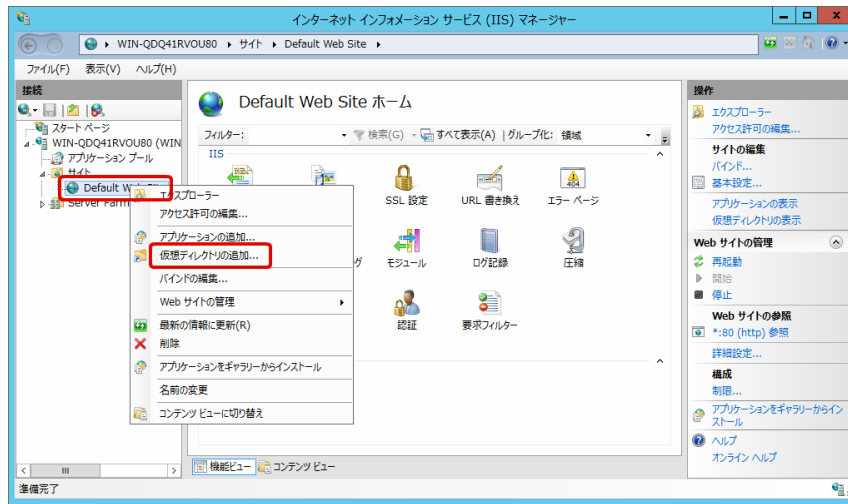
コラム

「Time-out (seconds)」には Application Request Routing が Resin への接続を切断するまでのタイムアウト時間を設定します。
テナント環境セットアップを完了するのに十分に大きい値を設定してください。
ここでは例として 1200 秒を設定しています。

6. 「変更内容は正常に保存されました。」と表示されていることを確認してください。



7. 続いて、静的ファイル用の仮想ディレクトリを作成します。
左ペインの「Default Web Site」のサブメニューより「仮想ディレクトリの追加...」をクリックします。



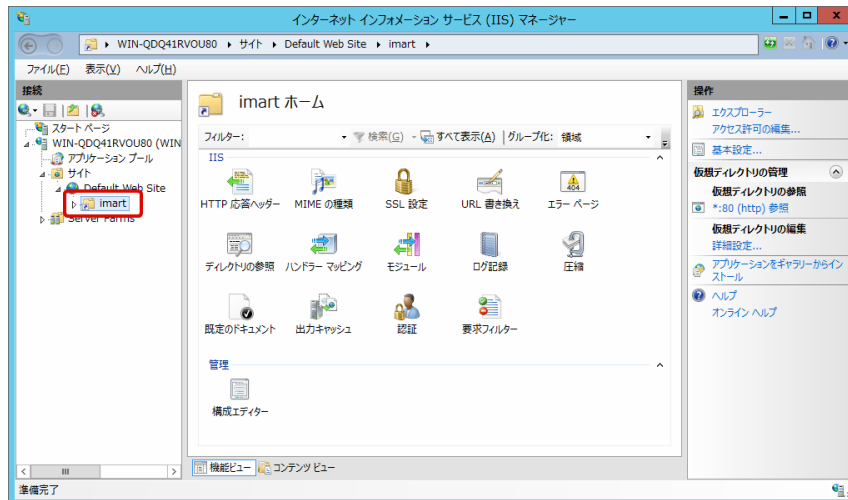
8. 新規ウィンドウ内で「エイリアス(A)」、「物理パス(P)」を次のように設定し、「OK」をクリックします。

エイリアス imart

物理パス C:\imart



9. 「imart」ディレクトリが作成されていることを確認してください。

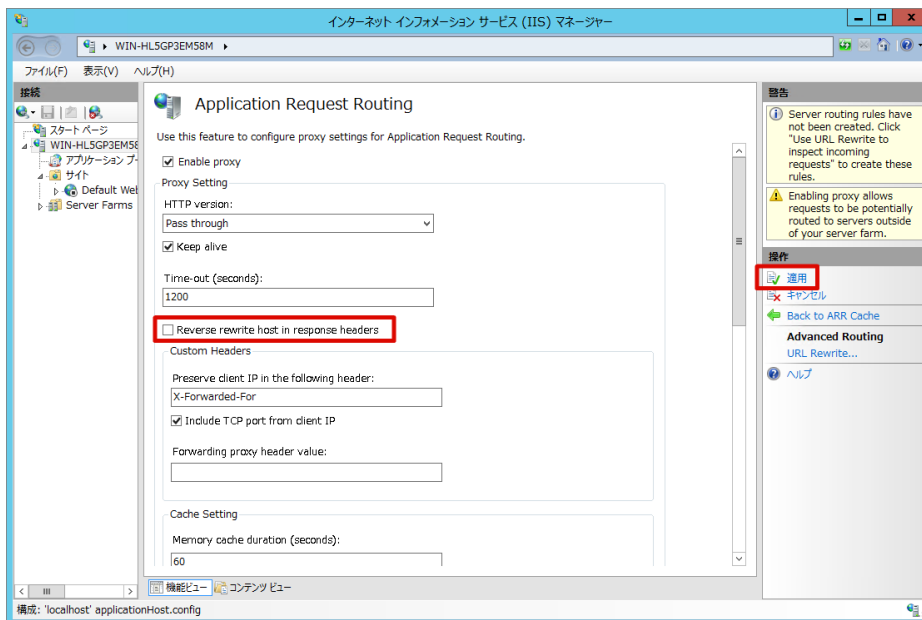


OAuth認証モジュールを利用する場合の追加設定

IIS環境下でOAuth認証モジュールを動かすためには、追加で以下の設定を行う必要があります。
手順では下記の環境を想定して記載します。

intra-mart Accel Platform のコンテキストパス	imart
IIS のルート ディレクトリ	C:\inetpub\wwwroot
静的ファイルの展開フォルダ	C:\imart
Resin の IP アドレス	192.168.1.1
Resin のポート番号	8080
ベースURL	http://192.168.1.1/imart

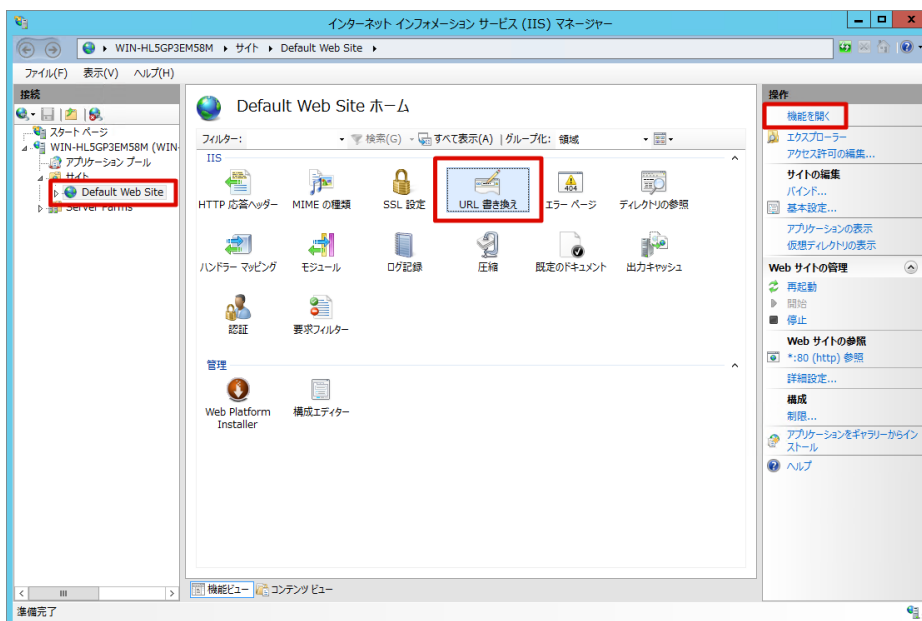
- インターネット インフォメーション サービス (IIS) マネージャー より「Application Request Routing Cache」>「Server Proxy Settings...」の順にクリックし、「Application Request Routing」を表示します。
- 「Reverse rewrite host in response headers」チェックボックスを off にして、「適用」をクリックします。



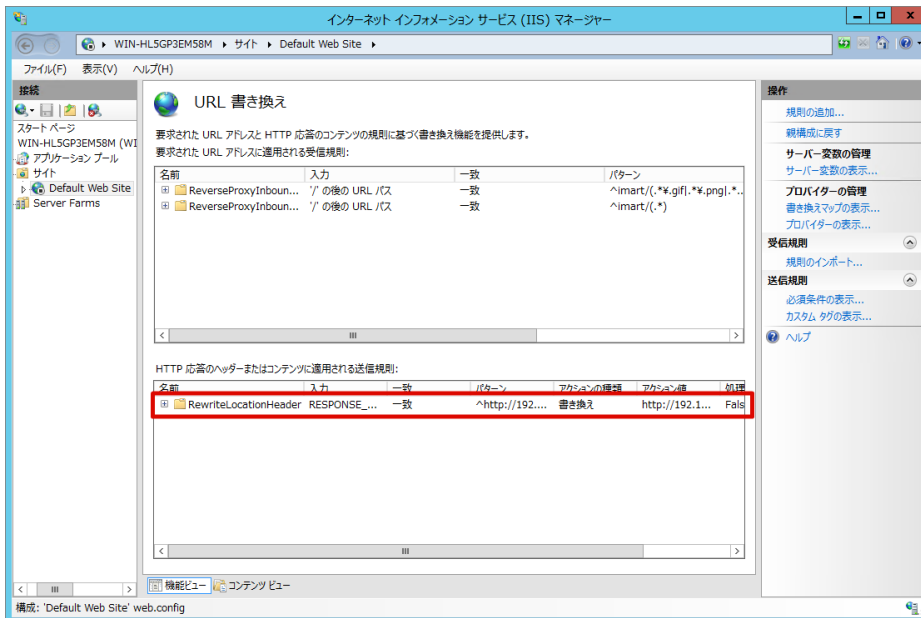
3. <C:/inetpub/wwwroot/web.config>に以下の設定を追加します。

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <system.webServer>
    <rewrite>
      ...
      <outboundRules>
        <rule name="RewriteLocationHeader" preCondition="Redirect Response">
          <match serverVariable="RESPONSE_Location" pattern="^http://192.168.1.1:8080/imart/(.*)" />
          <action type="Rewrite" value="http://192.168.1.1/imart/{R:1}" />
        </rule>
        <preConditions>
          <preCondition name="Redirect Response">
            <add input="{RESPONSE_STATUS}" pattern="3\d\d" />
          </preCondition>
        </preConditions>
      </outboundRules>
    </rewrite>
  </system.webServer>
</configuration>
```

4. インターネット インフォメーション サービス (IIS) マネージャー より「Default Web Site」「URL 書き換え」「機能を開く」の順にクリックします。



5. 「HTTP 応答のヘッダーまたはコンテンツに適用される送信規則」に下図のように追加されていることを確認します。



静的ファイルの配置

IM-Juggling から取得した「静的ファイル」を、仮想ディレクトリに設定した「物理パス」のディレクトリに展開します。
静的ファイルの出力についての詳細は、[静的ファイルの出力](#)を参照してください。

アクセスログの編集

リバース Proxy の場合、Resin が出力する access.log のソース IP がすべて IIS の IP になります。
回避するためには access.log の出力内容を編集します。

1. <%RESIN_HOME%/conf/resin.xml> ファイルを開きます。
2. <host id="" root-directory=""> ディレクティブ内に以下の設定を追加して Resin の再起動を行ってください。

```
<access-log path='log/access.log'>
<rollover-period>1D</rollover-period>
<format>{%[X-Forwarded-For]i %h %l %u %t "%r" %>s %b "%[Referer]i" "%[User-Agent]i"</format>
</access-log>
```

コラム

access.log についての詳細な設定内容については「[Resin Documentation](#)」を参照してください。

動作確認

1. Resin を起動します。
Resin についての詳細は以下を参照してください。
 - [Web Application Server](#)
 - [Web Application Server の起動・停止](#)
2. Internet Information Services (IIS) を再起動します。
3. ブラウザより <http://<IISのIPアドレス>:<IISのポート番号>/imart/login> にアクセスし intra-mart Accel Platform のログイン画面が表示されれば設定は完了です。

Apache Cassandra

Apache Cassandra は、「IMBox」を利用するために必要となります。
詳細は「IMBox Cassandra管理者ガイド」を参照してください。

[IMBox Cassandra管理者ガイド \(日本語\)](#)

[IMBox Cassandra管理者ガイド \(English\)](#)

[IMBox Cassandra管理者ガイド \(中文\)](#)

Apache Solr

Apache Solr は、「IM-ContentsSearch」を利用するために必要となります。
セットアップ方法の詳細は「[Solr管理者ガイド](#)」-「[Solrのセットアップ](#)」を参照してください。

WARファイルの作成

プロジェクトの作成とモジュールの選択

項目

- IM-Juggling とは
- IM-Juggling のインストール
 - インターネットに接続できない環境で IM-Juggling を利用する場合
 - プロキシ設定が必要な環境で IM-Juggling を利用する場合
- プロジェクトの新規作成
- モジュールの選択
- アプリケーションの追加

IM-Juggling とは

IM-Juggling はモジュールの管理、WARファイルを出力できる環境構築ツールです。
このツールを使用し、新しい機能の導入や不具合修正の適用をします。

IM-Juggling のインストール

製品に同梱されているディレクトリより、環境に合わせた IM-Juggling を任意のパスに展開します。

！ 注意

IM-Juggling を利用してプロジェクトの作成や、WARファイルの作成を行っている際にリポジトリ情報の取得等のエラーが発生した場合、下記のディレクトリにあるデータを削除して再度、IM-Juggling を起動して試行してください。

%OSユーザディレクトリ%/juggling/workspace/.repository ディレクトリ

古い情報が残っているためにエラーとなる場合があります。
この古いファイルを削除する事で、最新のデータが再取得され問題を回避します。

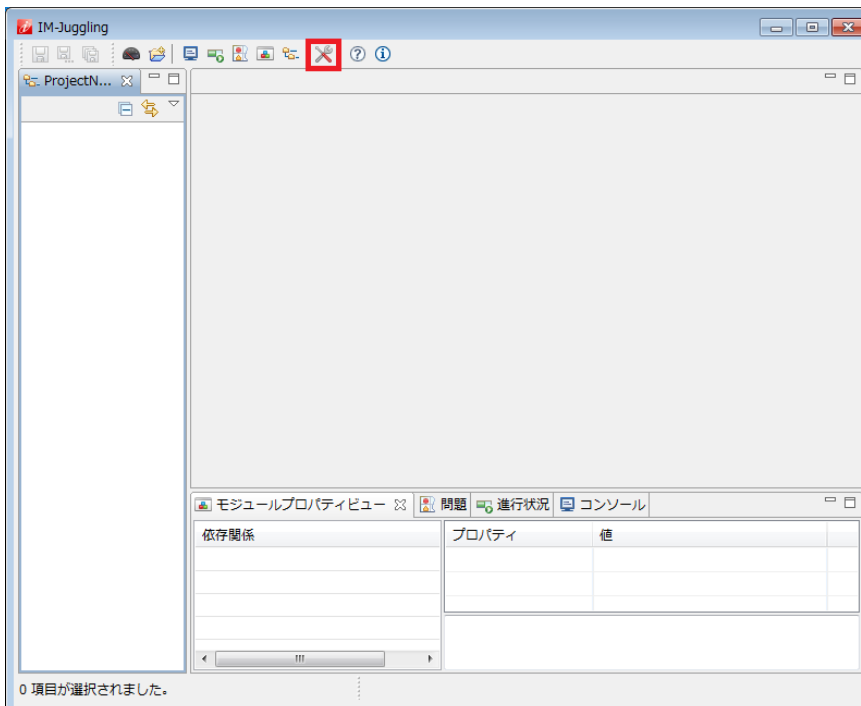
インターネットに接続できない環境で IM-Juggling を利用する場合

製品版には IM-Juggling で利用するリポジトリデータも同梱されています。

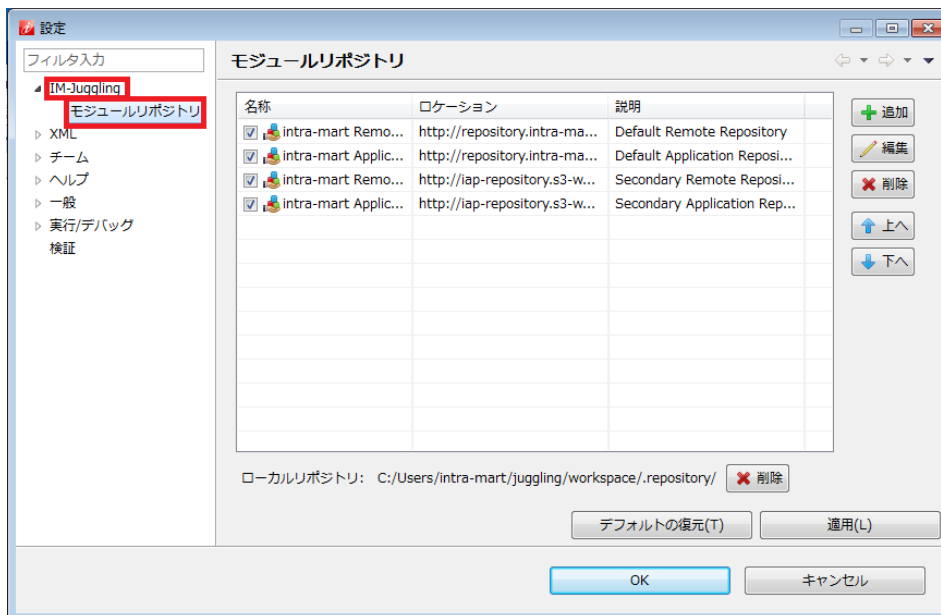
%製品版メディアイメージ(DVD)%/Products/repository ディレクトリ

リポジトリデータをコピーし、IM-Juggling のモジュールリポジトリとして設定することで、インターネットに接続できない環境でも IM-Juggling を利用することができます。
以下の方法で、「platform」「products」のそれぞれのリポジトリデータをモジュールリポジトリとして設定してください。

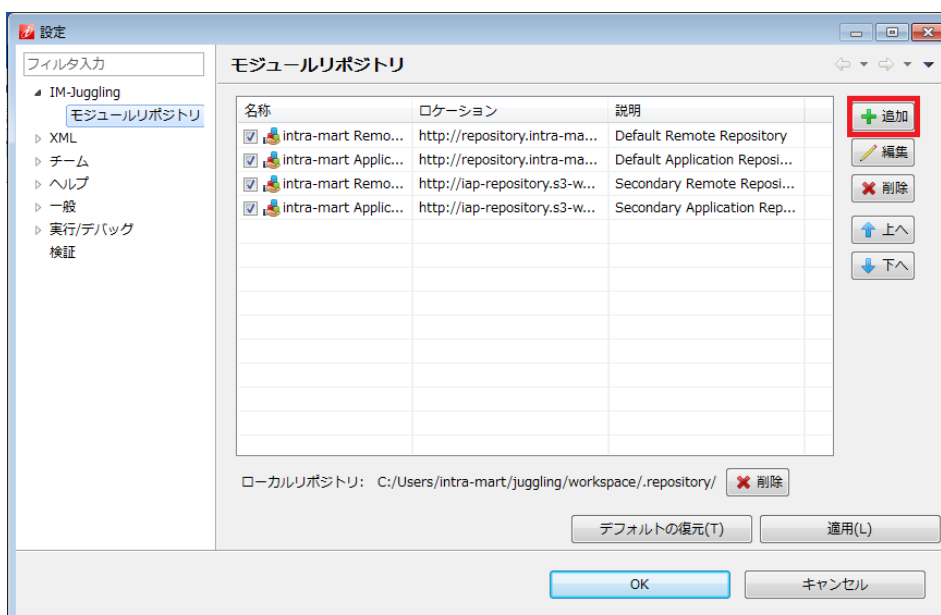
1. IM-Juggling ウィンドウ内-ツールバー右端にある「設定」アイコンをクリックします。



2. 「IM-Juggling」-「モジュールリポジトリ」を選択します。

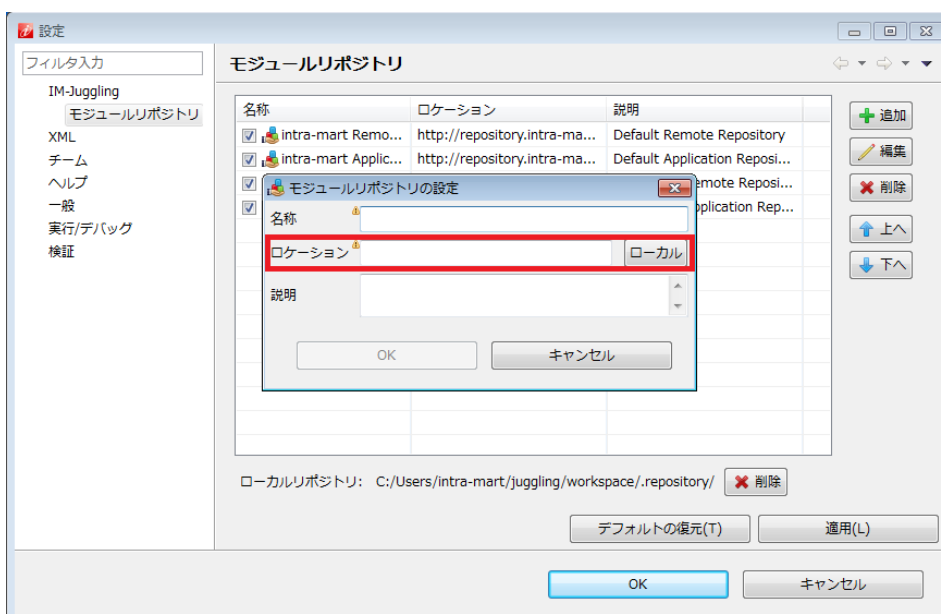


3. 「追加」をクリックします。

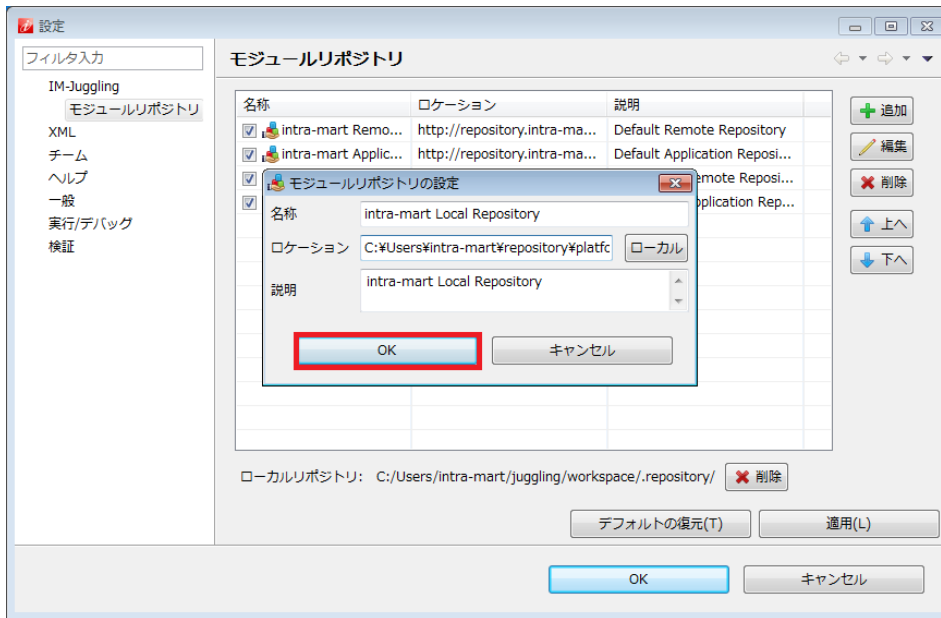


4. モジュールリポジトリの追加画面が表示されます。

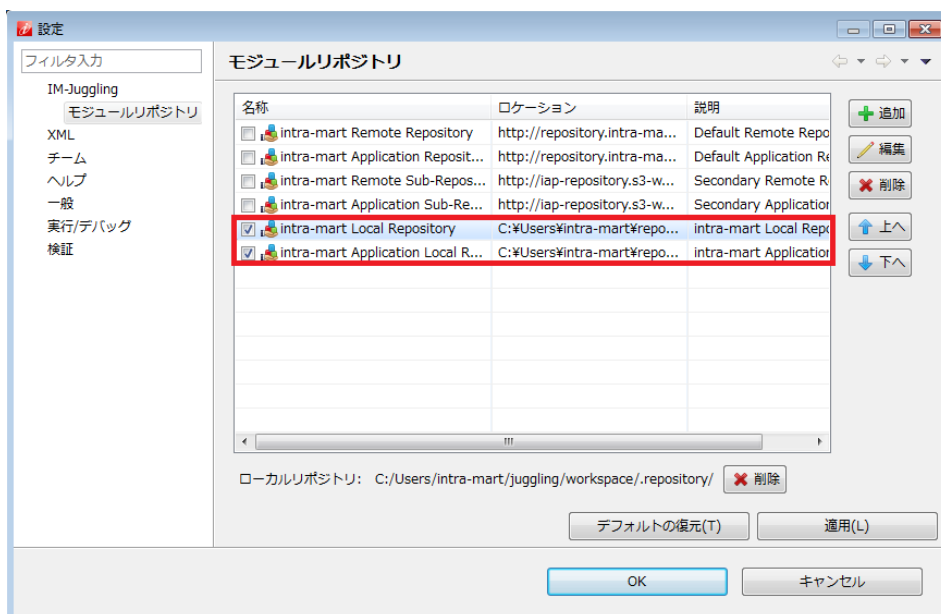
5. 「ロケーション」にコピーしたりリポジトリデータの場所を設定します。



6. 「名称」「説明」を記入し、「OK」をクリックします。

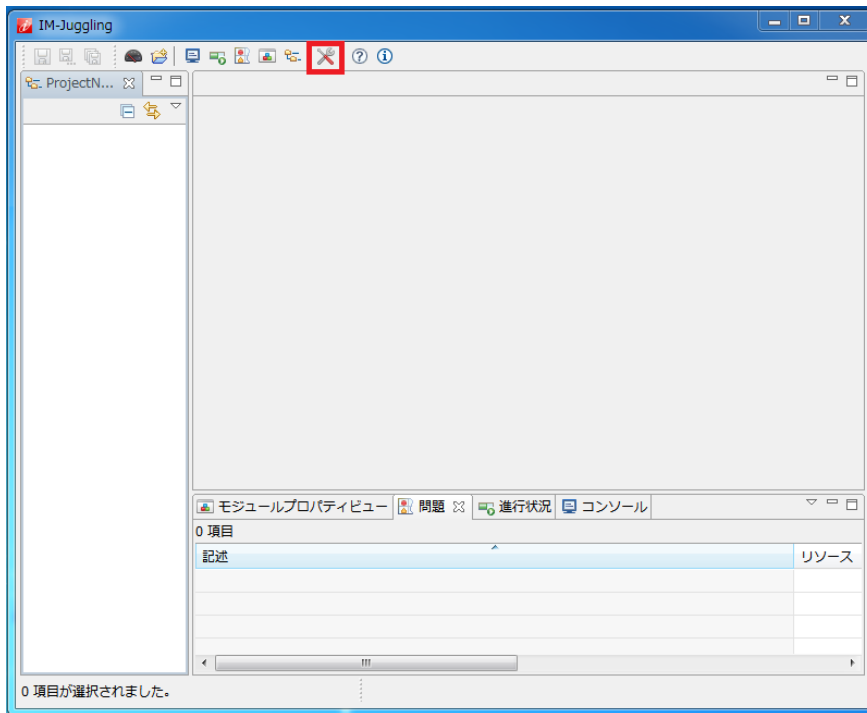


7. 追加したモジュールリポジトリにチェックがついていることを確認し、「OK」をクリックします。

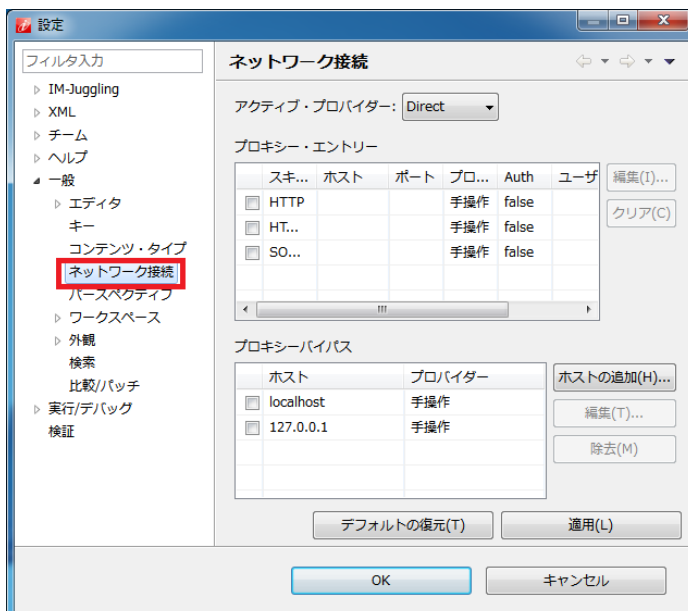


プロキシ設定が必要な環境で IM-Juggling を利用する場合

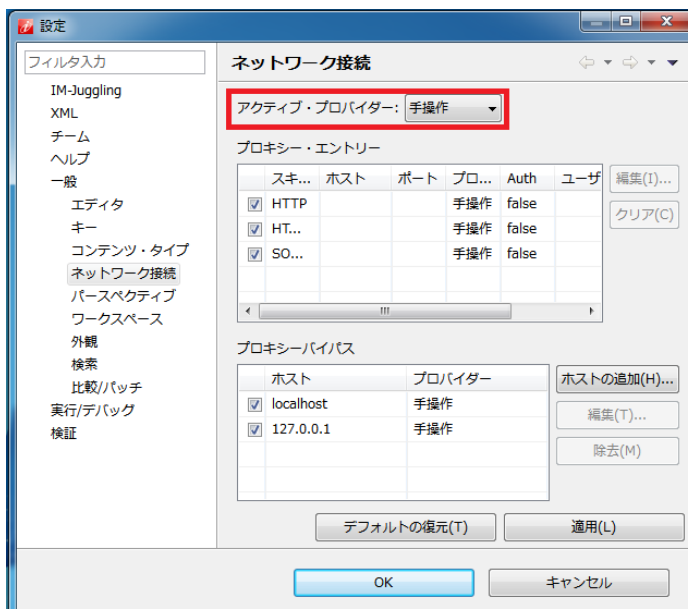
1. IM-Juggling ウィンドウ内-ツールバー右端にある「設定」アイコンをクリックします。



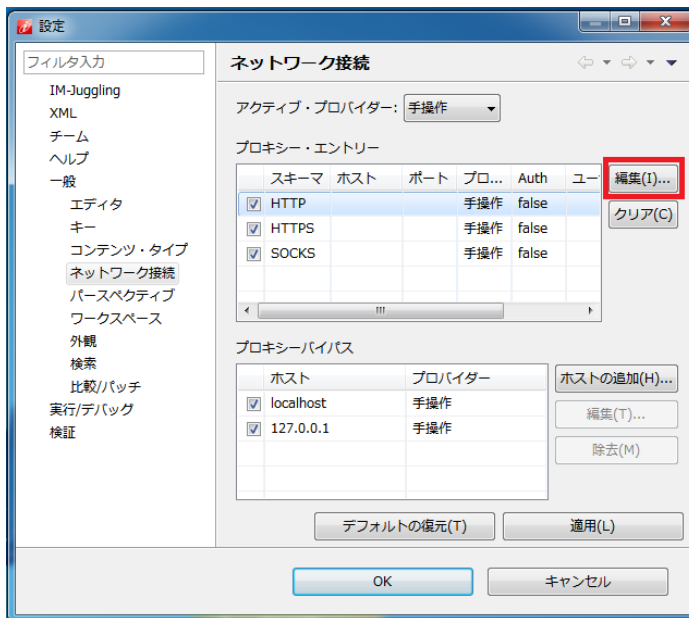
2. 「一般」-「ネットワーク接続」を選択します。



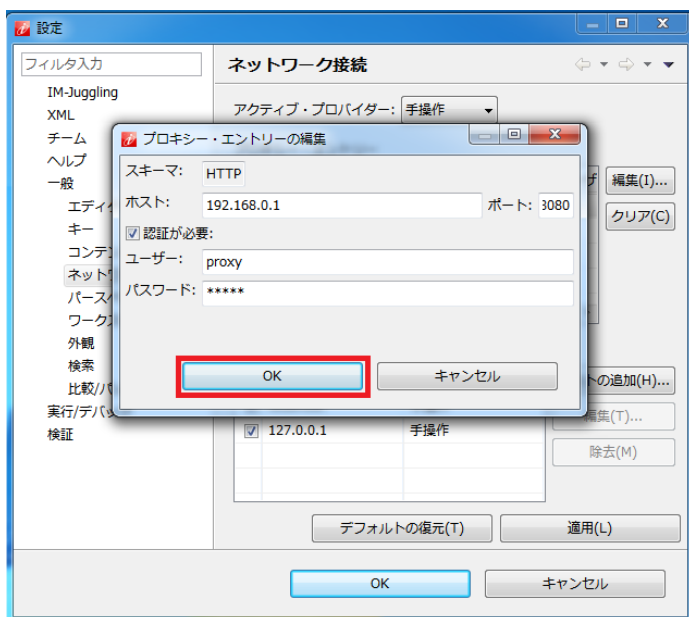
3. アクティブ・プロバイダーに「手操作（マニュアル）」を選択します。



4. プロキシ・エントリーの「HTTP」の「編集」をクリックします。



5. プロキシ・エントリーの編集画面で「ホスト」、「ポート」、「認証情報」を設定して「OK」をクリックします。

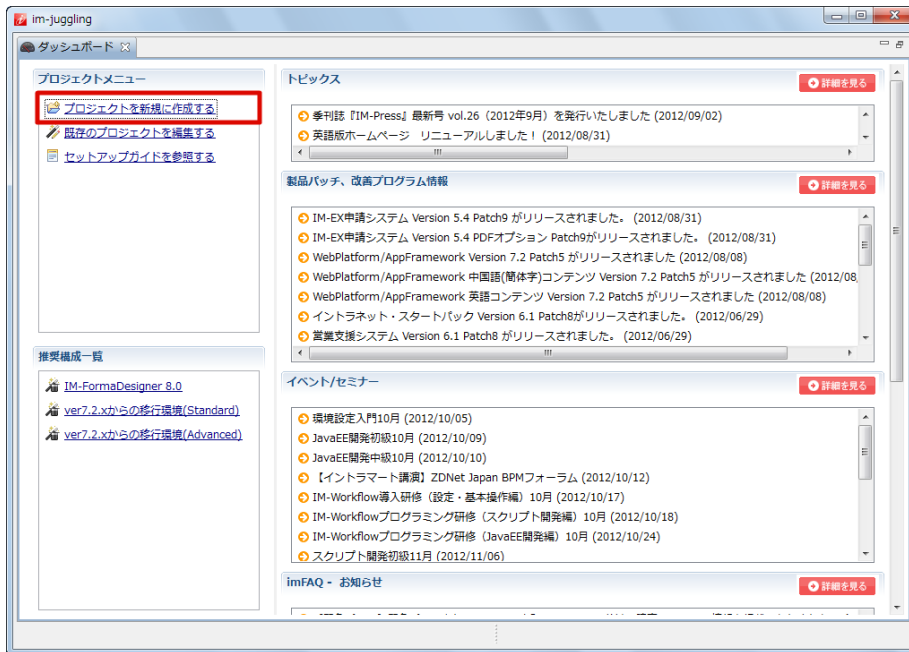


注意

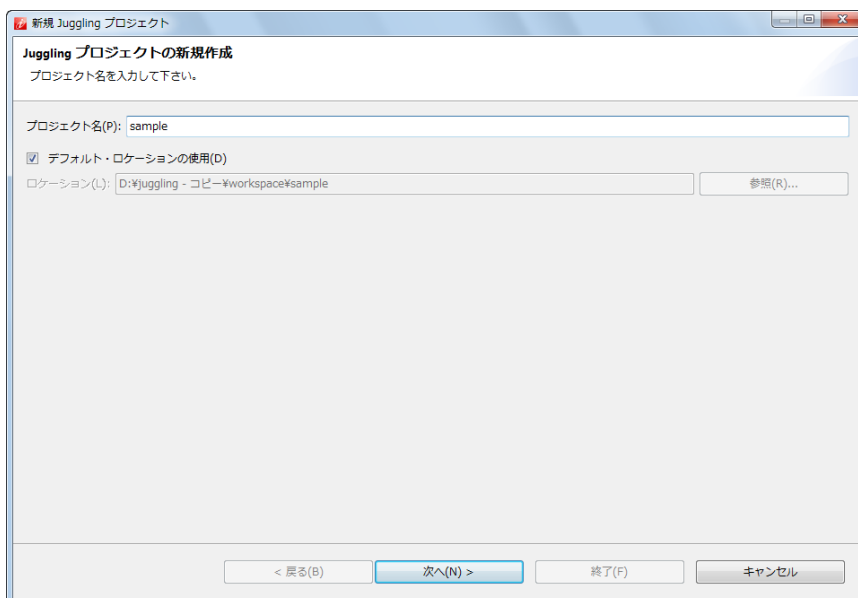
アクティブ・プロバイダーとして「ネイティブ」を選択した場合、Internet Explorer のプロキシサーバの設定情報を引き継ぎます。
(Internet Explorerで「Alt」キーを押してメニューを表示し、「ツール」→「インターネットオプション」→「接続」タブ→「LANの設定」で表示されるプロキシサーバの設定です)

プロジェクトの新規作成

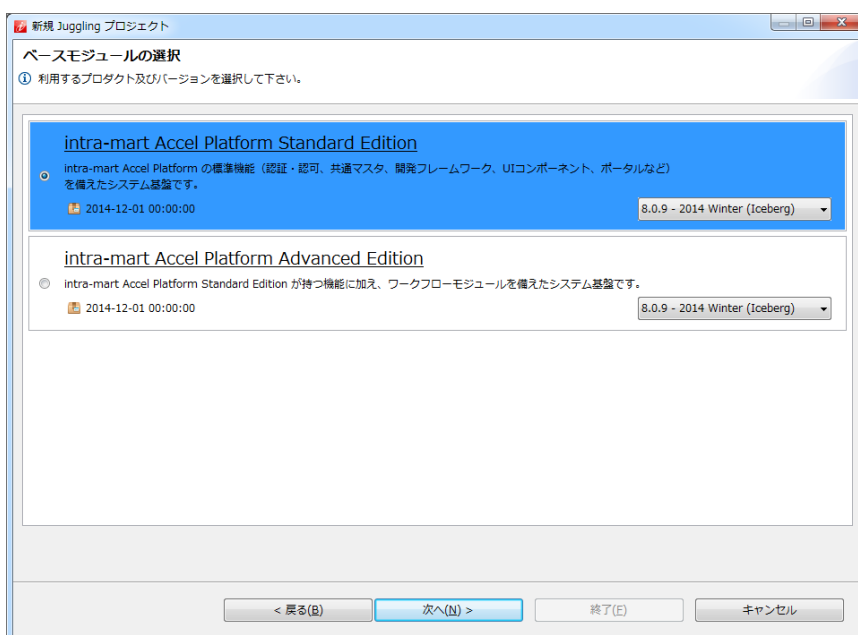
1. 展開したディレクトリ直下にある、**juggling.exe** を起動します。
2. 初回起動時の場合、「使用許諾契約書の同意」が表示されます。
「同意する」を選択し「OK」をクリックします。
3. ダッシュボードが表示されます。
「プロジェクトを新規に作成する」を選択します。



4. ポップアップ表示されたウィザード画面内の「プロジェクト名」に半角英数字の任意のプロジェクト名を入力します。入力後、「次へ(N)」をクリックします。



5. 利用するプロダクト及びバージョンを選択します。選択後、「次へ(N)」をクリックします。



！ 注意

intra-mart Accel Platformには、以下の3つの構成が用意されています。
各構成の選択方法は次の通りです。

スタンダード

＜ intra-mart Accel Platform Standard Edition ＞のツリーから選択します。

アドバンスト

＜ intra-mart Accel Platform Advanced Edition ＞のツリーから選択します。

エンタープライズ

＜ intra-mart Accel Platform Advanced Edition ＞のツリーから選択します。

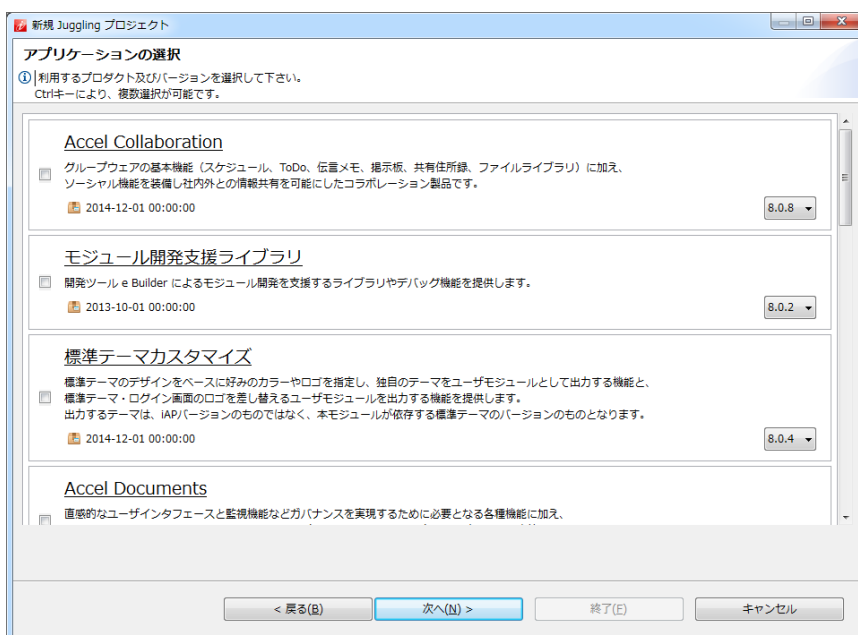
次の「アプリケーションの選択」より、
IM-FormaDesigner for Accel Platform、IM-BIS for Accel Platform を選択します。

ご購入頂いたライセンスは、全ての製品構成とも、「[ライセンスの登録](#)」にて登録を行います。
製品構成の詳細については、「[Introduction](#)」を参照してください。

i コラム

同一プロダクトにおいて複数のバージョンが表示されている場合は最新のものを選択する事を推奨します。

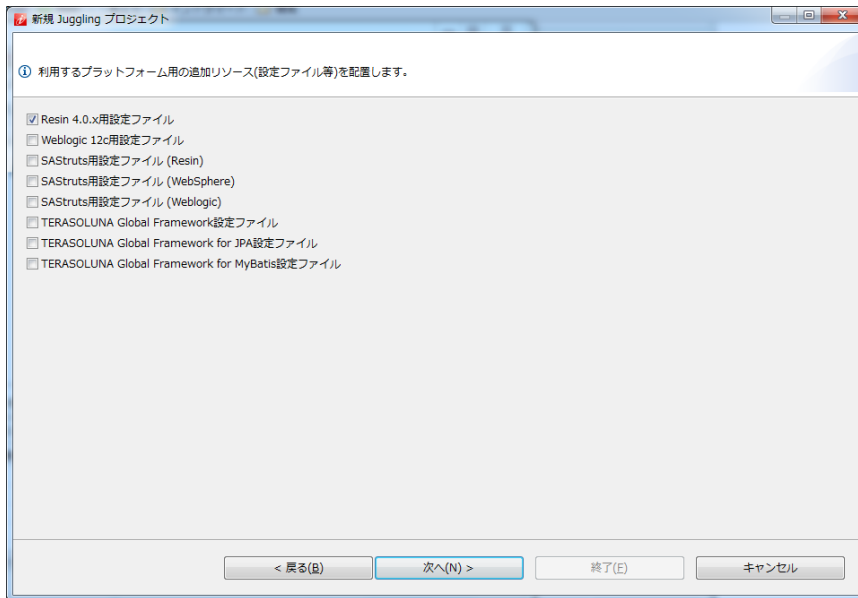
6. 利用するアプリケーションを選択します。
選択後、「次へ(N)」をクリックします。



！ 注意

「移行ツール」、「互換機能」は version7.2 からの移行を行う場合に選択します。
新規で intra-mart Accel Platform を構築する場合は、このプロダクトは選択しないよう注意してください。
version7.2 からの移行を行う場合は、別途公開されている「移行手順書」を参照してください。

7. 利用するプラットフォーム用の追加リソース(設定ファイル等)を選択します。
選択後、「次へ(N)」をクリックします。



注意

WARファイルを Oracle WebLogic Server 12c (12.1.3) ヘッドプロイする場合は、必ず「**Weblogic 12c用設定ファイル**」を選択してください。
この設定ファイルを配置していない場合、WebLogicへデプロイする際にエラーが発生します。

コラム

SAStruts+S2JDBCにてデータベースを利用する場合は、使用するWebアプリケーションサーバに応じた「**SAStruts用設定ファイル**」にチェックをつけてください。
設定変更については、[SAStruts用設定ファイル \(SAStruts+S2JDBCにてデータベースを利用する場合\)](#) で説明します。

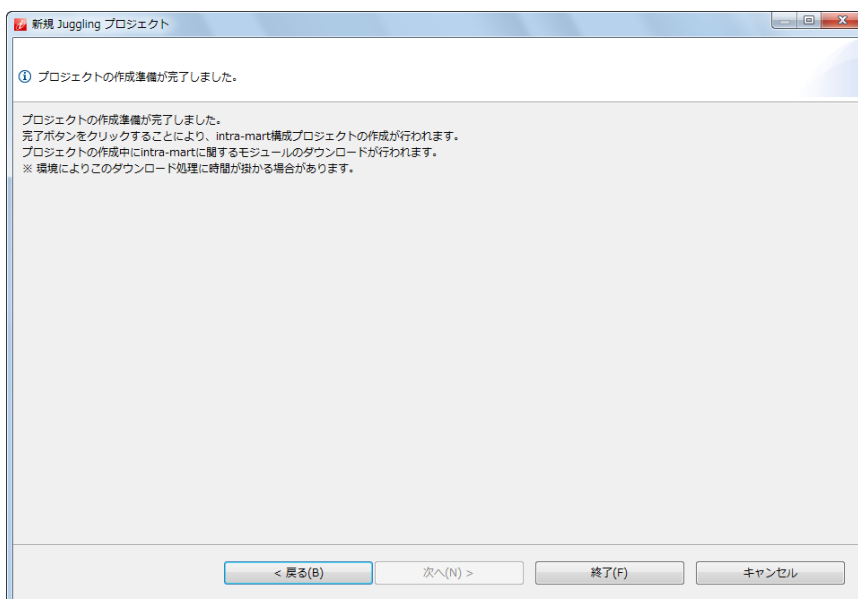
コラム

TERASOLUNA Server Framework for Java (5.x) を利用する場合は、「**TERASOLUNA Server Framework for Java (5.x) 用設定ファイル**」にチェックをつけてください。
設定変更については、[TERASOLUNA Server Framework for Java \(5.x\) 用設定ファイル \(シェアードデータベースを利用する場合\)](#) で説明します。

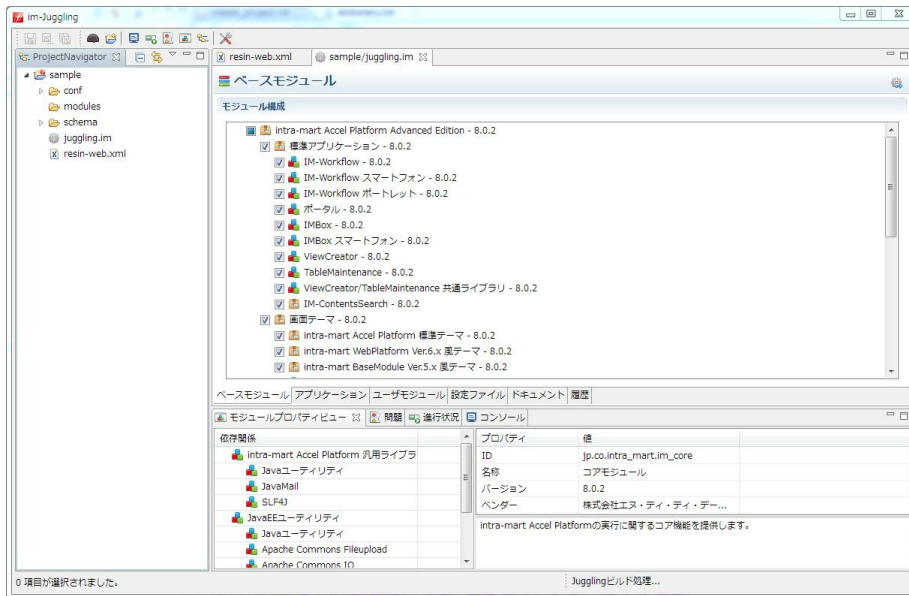
リポジトリ層にJPAを利用する場合は、「**TERASOLUNA Server Framework for Java (5.x) for JPA用設定ファイル**」にチェックをつけてください。
設定変更については、[TERASOLUNA Server Framework for Java \(5.x\) 用設定ファイル \(リポジトリ層にJPAを利用する場合\)](#) で説明します。

リポジトリ層にMyBatisを利用する場合は、「**TERASOLUNA Server Framework for Java (5.x) for MyBatis用設定ファイル**」にチェックをつけてください。
設定変更については、[TERASOLUNA Server Framework for Java \(5.x\) 用設定ファイル \(リポジトリ層にMyBatisを利用する場合\)](#) で説明します。

8. 最後に「終了(F)」をクリックします。



9. プロジェクトの作成処理が行われ、完了すると以下の画面が表示されます。



注意

WARファイルを Oracle WebLogic Server 12c (12.1.3) ヘドブレイスする場合は、下記のモジュールを選択しないでください。これらのライブラリは Oracle WebLogic Server 12c (12.1.3) で利用できないため起動時にエラーが発生します。

- ライブラリ > サードパーティ製ライブラリ > Metro
- ライブラリ > サードパーティ製ライブラリ > OpenPortal WSRP

注意

開発フレームワーク「TERASOLUNA Server Framework for Java (5.x) on Accel Platform」モジュールは Oracle WebLogic Server 12c (12.1.3) および WebSphere Application Server 8.5.5 では利用できません。そのため、WARファイルを Oracle WebLogic Server 12c (12.1.3) または WebSphere Application Server 8.5.5 ヘドブレイスする場合は、開発フレームワーク「TERASOLUNA Server Framework for Java (5.x) on Accel Platform」モジュールを選択しないでください。

モジュールの選択

- プロジェクト内に表示されているモジュールの右クリックメニューで表示されるサブメニューより、一括で関連するモジュールの選択、選択解除を行う事ができます。

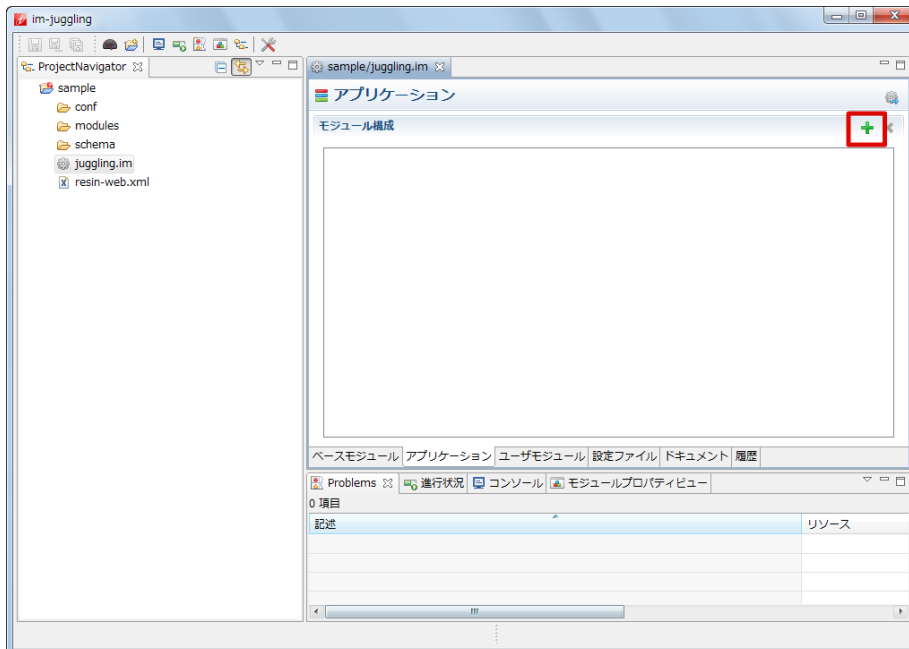
コラム

例として、IMBox を構成に含めない場合、IMBox モジュールを外す必要があります。詳細は、「[IMBox モジュールを外す方法](#)」を参照してください。

アプリケーションの追加

プロジェクト作成後、アプリケーションを追加することができます。

- 「アプリケーション」タブをクリックし、「+」をクリックします。



2. 追加したいアプリケーションをクリックし、「OK」をクリックします。



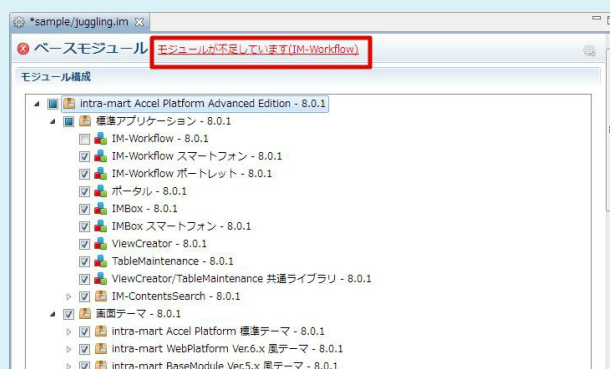
注意

「移行ツール」、「互換機能」は version7.2 からの移行を行う場合に選択します。
新規で intra-mart Accel Platform を構築する場合は、このプロダクトは選択しないよう注意してください。
version7.2 からの移行を行う場合は、別途公開されている「移行手順書」を参照してください。

コラム

画面上部にエラーメッセージが表示される場合

モジュール単位・アプリケーション単位で依存関係を持っています。



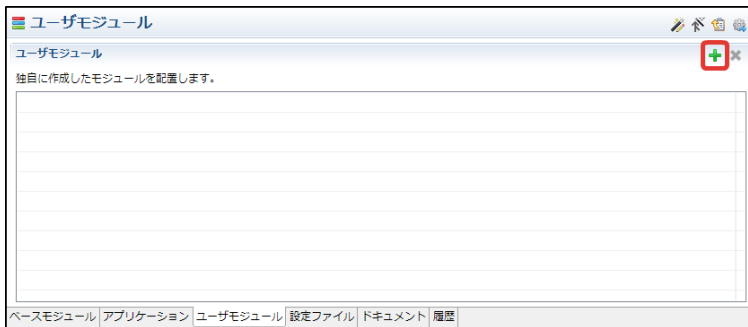
以下の手順で必要な設定を追加することができます。

1. メッセージをクリックします。
2. 「依存関係の解決」画面で「OK」をクリックします。

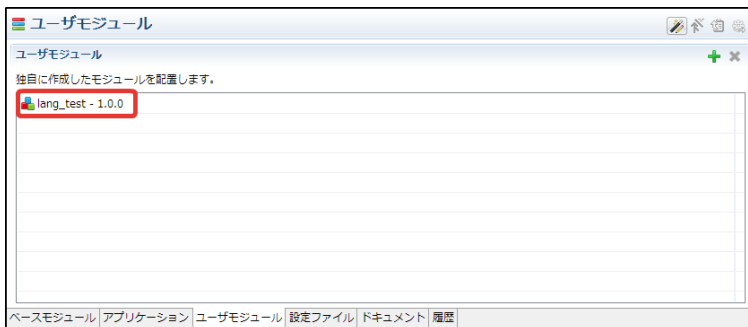
ユーザモジュール

プロジェクト作成後、独自で作成したアプリケーションをユーザモジュールとしてWARファイルに追加することができます。

1. 「ユーザモジュール」タブをクリックし、「+」をクリックします。



2. 追加したいimmファイルを選択し、「OK」をクリックします。



intra-mart Accel Platform の設定ファイル

intra-mart Accel Platform を稼働させるために以下の設定ファイルの編集を行います。

注意

自作したユーザモジュール内に含まれる設定ファイルと、同一の設定ファイルが既に IM-Juggling プロジェクト上に存在する場合、このプロジェクトのWARファイルを出力すると、IM-Juggling プロジェクト上に配置されている設定ファイルが反映されます。IM-Juggling プロジェクトの設定ファイルが反映される動作仕様となります。ユーザモジュールを動作させるための設定は、IM-Juggling プロジェクト上に配置されている設定ファイル側に書き写す必要があります。

DataSource

intra-mart Accel Platform で利用するデータベースの設定を行います。

項目

- 接続先の種類
- DataSourceの設定
 - PostgreSQL
 - Oracle
 - Microsoft SQL Server 2012
 - IBM DB2
- DataSourceマッピングの設定

接続先の種類

intra-mart Accel Platform では、以下の3種類のデータベースに接続することができます。

- システムデータベース

システムのデータを保存するデータベースです。
アプリケーション起動時にあらかじめ接続可能となっている必要があります。
システムデータベースはシステム内部で利用されるため、本番環境等ではアプリケーションの接続先としては推奨しません。

- テナントデータベース

テナント内で利用するデータを保存するデータベースです。

- シェアードデータベース

intra-mart Accel Platform 外のデータを保存するデータベースです。
外部システムと連携したい場合等に利用します。



注意

ViewCreator/TableMaintenance等で利用できる データベース はテナントデータベースとシェアードデータベースが対象です。

システムデータベースとテナントデータベースを同じ接続先として設定することは可能ですが、
システムデータベースとして作成されるテーブルの利用はサポート対象外となります。

DataSourceの設定

1. 「ProjectNavigator」内の <(プロジェクト名)/resin_web.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <web-app>/<database>/<driver> に接続先のデータベース接続情報を設定します。
以下は各データベース毎の接続例です。



コラム

複数テナントの場合

テナント数分、DataSourceを準備してください。

各テナント毎に利用するDataSourceを設定してください。

PostgreSQL

```
<driver>
  <type>org.postgresql.Driver</type>
  <url>jdbc:postgresql://localhost:5432/dbname</url>
  <user>username</user>
  <password>password</password>
</driver>
```

Oracle

```
<driver>
  <type>oracle.jdbc.driver.OracleDriver</type>
  <url>jdbc:oracle:thin:@localhost:1521:orcl</url>
  <user>username</user>
  <password>password</password>
</driver>
```

Microsoft SQL Server 2012

```
<driver>
  <type>com.microsoft.sqlserver.jdbc.SQLServerDriver</type>
  <url>jdbc:sqlserver://localhost:1433;DatabaseName=databasename</url>
  <user>username</user>
  <password>password</password>
  <init-param>
    <param-name>SelectMethod</param-name>
    <param-value>cursor</param-value>
  </init-param>
</driver>
```

IBM DB2

```
<driver>
<type>com.ibm.db2.jcc.DB2Driver</type>
<url>jdbc:db2://localhost:50000/databasename</url>
<user>username</user>
<password>password</password>
</driver>
```

コラム

システムデータベースとテナントデータベースを別々の接続先として設定する場合、JNDI名は違うものを指定してください。

DataSourceマッピングの設定

1. 「ProjectNavigator」内の<(プロジェクト名)/conf/data-source-mapping-config.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. DataSourceとして設定したJNDI名を指定します。
システムデータベースとテナントデータベース毎に設定したDataSourceを指定します。

<data-source-mapping-config>/<system-data-source>/<resource-ref-name> にシステムデータベースのJNDI名を設定します。

```
<system-data-source>
  <resource-ref-name>java:comp/env/jdbc/system</resource-ref-name>
</system-data-source>
```

<data-source-mapping-config>/<tenant-data-source>/<resource-ref-name> にテナントデータベースのJNDI名を設定します。

```
<tenant-data-source>
  <tenant-id>default</tenant-id>
  <resource-ref-name>java:comp/env/jdbc/tenant</resource-ref-name>
</tenant-data-source>
```

注意

<tenant-id> には、初回で作成するテナントIDを設定してください。

Storage

Storage領域として利用するパスを指定します。

1. 「ProjectNavigator」内の「(プロジェクト名)/conf/storage-config.xml」 ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <storage-config>/<storage-info>/<root-path-name> にStorage領域のパスを設定します。

```
<root-path-name>/tmp/storage</root-path-name>
```

- テナント毎に利用するストレージ(パブリックストレージ)については設定ファイルでの指定と画面からの指定で、パスが異なります。
設定ファイルで指定した場合は「<root-path-name>/<public-directory-name>/<storage-directory-name>/テナントID」となります。
後述する画面から指定する場合は「画面から指定したストレージパス/<storage-directory-name>」となります。

コラム

WARファイルによる複数テナントの場合

テナント数分、Storage領域を準備してください。
各テナント毎に利用するStorage領域を設定してください。

コラム

分散環境を構築する場合

Storageのルートディレクトリに指定するパスは全て同じ共有ディレクトリを参照するように設定してください。
サーバ毎に参照先が異なる場合、Storageに配置したファイルが共有できません。

コラム

分散環境のサーバをWindowsサービスに登録する場合

共有ディレクトリのパスはUNC形式で指定してください。
ドライブレターで指定した場合、Windowsサービスから起動を行った際に共有ディレクトリにアクセスできません。

SessionTimeOut

最終操作からタイムアウト時間(分)の設定を行います。

1. 「ProjectNavigator」内の<(プロジェクト名)/resin_web.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <web-app>/<session-config>/<session-timeout> にタイムアウト時間(分)を指定します。

```
<session-config>
  <session-timeout>30</session-timeout>
</session-config>
```

i コラム

複数テナントの場合

各テナント毎にタイムアウト時間(分)を設定してください。

Locale

利用する言語を設定します。

intra-mart Accel Platform は日本語 (ja)、英語 (en)、中国語 (簡体字) (zh_CN) の3言語が利用できます。

1. 「ProjectNavigator」内の「juggling.im」をダブルクリックします。
2. 「設定ファイル」タブより、「国際化機能」-「ロケールマスタ」を選択し、右側にある「出力」をクリックします。
出力されると、「ProjectNavigator」内に <locale-config/im-locale-default.xml> ファイルが表示されます。
3. 「ProjectNavigator」内の <locale-config/im-locale-default.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
4. <locale-config>/<locale> 内を編集します。

```
<locale name="en" default="true">
  <encoding-name>UTF-8</encoding-name>
</locale>
<locale name="ja">
  <encoding-name>UTF-8</encoding-name>
</locale>
<locale name="zh_CN">
  <encoding-name>UTF-8</encoding-name>
</locale>
```

i コラム

例として日本語 (ja) のみで運用する場合の設定内容を説明します。

```
<?xml version="1.0" encoding="UTF-8"?>
<locale-config
  xmlns="http://intra_mart.co.jp/system/i18n/locale/config/locale-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://intra_mart.co.jp/system/i18n/locale/config/locale-config locale-config.xsd">
  <locale name="ja">
    <encoding-name>UTF-8</encoding-name>
  </locale>
</locale-config>
```

! 注意

intra-mart Accel Platform ではこの設定変更を行っても、後述の「テナント環境セットアップ」実行時では、3言語分のデータがインポートされます。
ただし、画面上では選択した言語のみが利用できる仕組みとなります。

! 注意

運用開始後にシステムロケールを **追加する** 場合には制約があります。

運用開始後にシステムロケールを追加した場合、そのまま運用を再開すると、マスターデータ・トランザクションデータともに不整合が発生します。以下のドキュメントを参照し、追加したル分のデータを補完してください。

- [言語追加ガイド](#)
- [国際化支援機能仕様書](#)

なお、運用開始後にシステムロケールを **削除する** 事は推奨していません。

JavaMail

intra-mart Accel Platform のメール送信に関する設定です。

「ProjectNavigator」内の <(プロジェクト名)/conf/javamail-config/javamail-config.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。

「javamail-config.xml」の設定内容については「[設定ファイルリファレンス](#)」を参照してください。

分散・Webサーバ構成時の設定

Network

intra-mart Accel Platform のネットワーク構成を設定します。

1. 「ProjectNavigator」内の <conf/network-agent-config.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <network-agent-config>/<TCP>/<initial-hosts>/<host> に Resin のIPアドレスを指定します。
分散構成の場合は、対象の Resin の数分追加してください。
3. <network-agent-config>/<id> にクラスタリングIDを指定します。
マルチテナント環境を構築する場合はテナント単位でこのクラスタリングIDを指定する必要があります。
分散構成の場合は同一のクラスタリングIDを指定します。
4. <network-agent-config>/<bind-port> にクラスタリングを行う際の通信用ポート番号を指定します。
マルチテナント環境を構築する場合はポート番号が衝突しないようポート番号を分ける必要があります。
5. <network-agent-config>/<port-range> にポートレンジを指定します。
このポートレンジは、<network-agent-config>/<bind-port>により指定されたポート番号が既に利用されていた場合
設定値のレンジだけ代替のポート番号を検索、利用します。

```
<id>prototype</id>
<bind-port>5200</bind-port>
<port-range>2</port-range>
<preferIPv4Stack>true</preferIPv4Stack>

<TCP>
  <initial-hosts>
    <host address="127.248.100.1"/>
    <host address="127.248.100.2"/>
    <host address="127.248.100.3"/>
  </initial-hosts>
</TCP>
```

ベースURL

サーバコンテキスト設定の ベースURL を設定します。

1. 「ProjectNavigator」内の <conf/server-context-config.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <server-context-config>/<base-url> に intra-mart Accel Platform の ベースURL を指定します。

コラム

ベースURLとは？

intra-mart Accel Platform のシステムを外部から参照する際に利用される基底のURLです。
一般的には、<http://example.org/imart> 等となります。

このベースURLは、あくまでも外部から参照される際に利用するURLです。
従って、localhostや、127.0.0.1(ループバックアドレス)等をURLとして指定された場合、外部からの接続時に適切にサーバに接続されなくなります。

注意

ベースURL が設定されていない場合、次の制限が発生します。

- IM-Workflow を利用する場合
メール定義 ならびに IMBox定義 の置換文字列「IM_URL」/「Matter_Detail_URL」の置換が行われません。

モジュール別の設定

LDAP認証設定ファイル(アカウントの認証にLDAP認証を利用する場合)

intra-mart Accel Platform でアカウントの認証にLDAP認証を利用する方法を解説します。

項目

- LDAP認証モジュールの機能
- LDAP認証モジュールの利用
- LDAP認証設定ファイル
 - <ldap-certification-config>
 - <ldap-server>
 - <search-controls>
 - <certification-options>
 - <option>

LDAP認証モジュールの機能

アカウントの認証時に入力されたユーザIDおよびパスワードを利用してLDAPサーバに対して認証を行うモジュールです。

LDAP認証モジュールでは、以下の機能を提供します。

- 設定したLDAPサーバのディレクトリ配下に存在するユーザを検索して認証ができます。(複数 OU 対応)
- 複数のLDAPサーバを設定できます。
 - LDAPサーバがダウンしてる場合に、順次設定されたLDAPサーバに問合せ先を切り替えます。
 - LDAPサーバで認証が失敗した場合、順次設定されたLDAPサーバに問合せ先を切り替えることも可能です。

 注意

LDAPサーバのユーザIDと同じユーザIDのアカウントが intra-mart Accel Platform にも必要となります。

- パスワード以外の情報は intra-mart Accel Platform のアカウント情報を利用します。

 注意

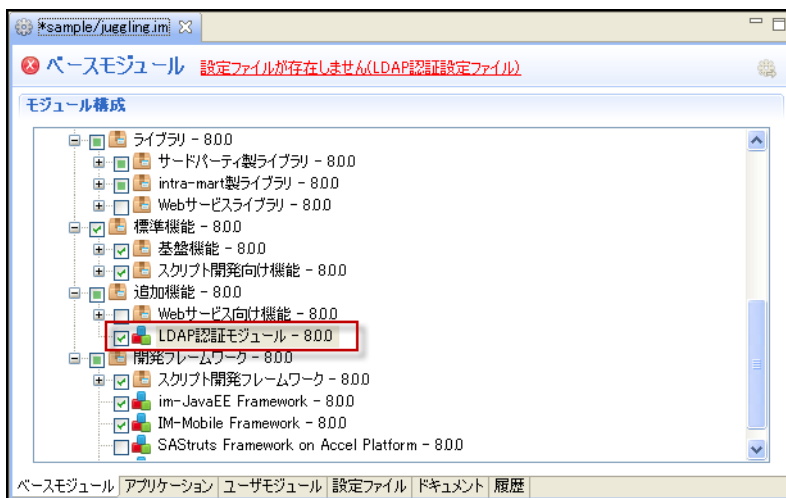
LDAPサーバに Active Directory を利用する場合は、LDAPサーバでユーザの検索を行うための検索用LDAPユーザの設定が必要です。

詳しくは <search-controls> を参照してください。

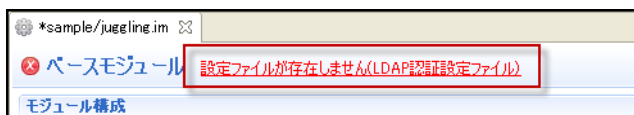
LDAP認証モジュールの利用

LDAP認証モジュールを利用する場合は、以下の手順でモジュールを追加します。

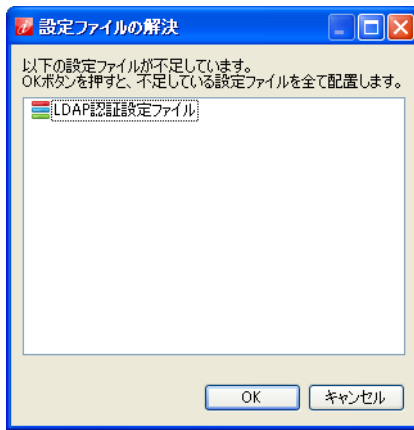
1. IM-Juggling で LDAP認証モジュールを選択してください。



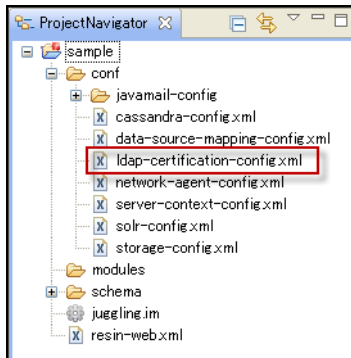
2. LDAP認証モジュールを選択すると、エディタ上部に警告が表示されます。
[設定ファイルが存在しません(LDAP認証設定ファイル)]のリンクをクリックします。



3. 「設定ファイル解決」ダイアログが表示されますので、[OK]をクリックします。
LDAP認証設定ファイル(ldap-certification-config.xml)がプロジェクトのconfに追加されます。



4. 追加されたldap-certification-config.xmlをエディタで編集します。



LDAP認証設定ファイル

LDAP認証に用いるLDAPサーバおよび各種設定を行うファイルです。

```
<?xml version="1.0" encoding="UTF-8"?>
<ldap-certification-config
  xmlns="http://intra-mart.co.jp/system/security/certification/provider/ldap"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://intra-mart.co.jp/system/security/certification/provider/ldap ../schema/ldap-certification-config.xsd">
  <enable>false</enable>
  <load-balancing>false</load-balancing>
  <attempt-on-failed-authentication>true</attempt-on-failed-authentication>
  <log>false</log>
  <ldap-servers>
    <ldap-server>
      <permit-no-password>true</permit-no-password>
      <provider-url>ldap://localhost:389/</provider-url>
      <context-factory>com.sun.jndi.LdapCtxFactory</context-factory>
      <base-dn>dc=ldaps,dc=intra,dc=intra-mart,dc=jp</base-dn>
      <search-filter>sAMAccountName=?</search-filter>
      <search-controls>
        <connect-timeout-property-name>com.sun.jndi.Ldap.connect.timeout</connect-timeout-property-name>
        <connect-timeout>0</connect-timeout>
        <searching-dn>sAMAccountName=admin,cn=User,dc=ldaps,dc=intra,dc=intra-mart,dc=jp</searching-dn>
        <searching-pw>*****</searching-pw>
        <count-limit>0</count-limit>
        <time-limit>0</time-limit>
      </search-controls>
    </ldap-server>
    <ldap-server>
      <permit-no-password>true</permit-no-password>
      <provider-url>ldap://localhost2:389/</provider-url>
      <context-factory>com.sun.jndi.LdapCtxFactory</context-factory>
      <base-dn>ou=People,dc=example,dc=com</base-dn>
      <search-filter>uid=?</search-filter>
      <search-controls>
        <connect-timeout-property-name>com.sun.jndi.Ldap.connect.timeout</connect-timeout-property-name>
        <connect-timeout>0</connect-timeout>
        <searching-dn>uid=admin,ou=People,dc=example,dc=com</searching-dn>
        <searching-pw>*****</searching-pw>
        <count-limit>0</count-limit>
        <time-limit>0</time-limit>
      </search-controls>
    </ldap-server>
  </ldap-servers>
</ldap-certification-config>
```

<ldap-certification-config>

LDAP認証の共通的な設定を行います。

<ldap-certification-config>の子要素

タグ	説明	必須	設定値	デフォルト値
<enable>	LDAP認証を利用するかどうかを設定します。	○	true / false	なし
<load-balancing>	ロードバランサを有効にします。 false : <ldap-servers>に設定されている<ldap-server>順に認証を試みます。 true : <ldap-server>の開始位置をランダムに決定し、決定した位置を基準にし、設定した順に認証を試みます。	×	true / false	false
<attempt-on-failed-authentication>	LDAP 認証失敗(ユーザID またはパスワードが不正)時に次の<ldap-server>に認証を試みます。 false : 次の<ldap-server>に認証を試みません。 true : 次の<ldap-server>に認証を試みます。	×	true / false	false
<log>	ログをコンソールに出力かどうかを設定します。 デバッグ用のオプションです。	×	true / false	false
<ldap-servers>	このタグの配下に<ldap-server>を使用して、認証に利用するLDAPサーバの設定を行います。	○	なし	なし

<ldap-server>

LDAP認証に用いるLDAPサーバの設定を記述します。

<ldap-servers>の子要素です。

<ldap-server>の要素

タグ	説明	必須	設定値	デフォルト値
<ldap-server>	認証に用いるLDAPサーバの設定を記述します。 複数設定した場合、設定順にLDAPサーバに通信を試みます。 通信できない場合は、次のLDAPサーバに通信を試みます。	○	なし	なし

<ldap-server>の子要素

タグ	説明	必須	設定値	デフォルト値
<permit-no-password>	空パスワードによる認証を許可するかどうかを設定します。 false : 空パスワードを認証失敗とみなします。 true : 空パスワードでLDAP に問い合わせます。	×	true/false	false
<provider-url>	プロバイダURL を設定します。【例】ldap://localhost:389/	○	URL	なし
<context-factory>	認証に用いるLDAPサーバの設定を記述します。	×	コンテキストファクトリクラス名	com.sun.jndi.ldap.LdapCtxFactory
<base-dn>	識別情報のベースを設定します。 【例】ou=People,dc=example,dc=com (Open LDAP) 【例】dc=ldaps,dc=intra,dc=intra-mart,dc=jp (Active Directory)	○	なし	なし
<search-filter>	ユーザを特定する為に一意になる属性を設定します。 【例】uid=? (Open LDAP) 【例】sAMAccountName=? (Active Directory) <search-filter> の代わりに <search-filter> も利用できますが、 <search-filter> を利用してください。	○	なし	なし
<search-controls>	LDAPサーバでユーザを検索するためのオプションを設定します。	×	なし	なし
<certification-options>	ユーザ認証において、LDAPサーバに問い合わせる時のオプションを設定します。	×	なし	なし

<search-controls>

ユーザ検索において、LDAPサーバの問い合わせに利用するオプションを設定します。

<ldap-server>の子要素です。

<search-controls>の子要素

タグ	説明	必須	設定値	デフォルト値
<connect-timeout-property-name>	接続タイムアウトを設定するプロパティ名を設定します。 【例】com.sun.jndi.ldap.connect.timeout	×	文字列	なし

タグ	説明	必須	設定値	デフォルト値
<connect-timeout>	接続タイムアウト時間を設定します。 <connect-timeout-property-name> に com.sun.jndi.ldap.connect.timeoutを設定した場合 ミリ秒数を示す0より大きい整数で設定し、LDAPサーバがこの値の期間中に接続を確立できない と、接続の試行は中止されます。 0以下の整数設定した場合または設定しない場合は、TCPなどのネットワークプロトコルのタイム アウト値が使用されます。	×	ミリ秒	なし
<searching-dn> または <serching-dn>	LDAPサーバでユーザを検索するためのユーザ(DN)を設定します。 【例】uid=admin,?ou=People,dc=example,dc=com (Open LDAP) 【例】sAMAccountName=?,dc=ldaps,dc=intra,dc=intra-mart,dc=jp (Active Directory) この設定は、ActiveDirectory の場合には必要になります。 <searching-dn> の代わりに <serching-dn> も利用できますが、 <searching-dn> を利用してください。	×	なし	なし
<searching-pw> または <serching-pw>	LDAPサーバでユーザを検索するためのユーザパスワードを設定します。 この設定は、ActiveDirectory の場合には必要になります。 <searching-pw> の代わりに <serching-pw> も利用できますが、 <searching-pw> を利用してください。	×	なし	なし
<count-limit>	最大検索数を設定します。デフォルト:0(上限無し)	×	最大検索数	0(上限なし)
<time-limit>	時間制限(ミリ秒)を設定します。	×	ミリ秒	0(無制限)
<deref-linf>	間接参照の有無を設定します。	×	true / false	false
<returning-obj>	結果の一部がオブジェクトとして返すかどうかを設定します。	×	true / false	false
<returning-attribute>	結果として返される属性を設定します。 指定あり:指定した属性のみ返します。 指定なし:すべての属性を返します。	×	属性ID(複数指定可能:[.]カンマで区切る)	指定なし(すべての属性)
<options>	ユーザ検索において、LDAPサーバの問い合わせに利用する追加オプションを設定します。 追加オプションを設定するための親タグです。 このタグの配下に<option>を使用して、追加オプションの設定を行います。	×	なし	なし

<certification-options>

ユーザ認証において、LDAPサーバの問い合わせに利用するオプションを設定します。

<ldap-server>の子要素です。

<certification-options>の子要素

タグ	説明	必須	設定値	デフォルト値
<options>	ユーザ認証において、LDAPサーバの問い合わせに利用する追加オプションを設定します。 追加オプションを設定するための親タグです。 このタグの配下に<option>を使用して、追加オプションの設定を行います。	×	なし	なし

<option>

LDAPサーバの問い合わせに利用する追加オプションを設定します。

<options> の子要素です。

```

<options>
  <option name="*****" value="*****"/>
</options>

```

<option>の要素

タグ	説明	必須	設定値	デフォルト値
<option>	ユーザ認証において、LDAPサーバの問い合わせに利用する追加オプションの内容を設定します。	○	なし	なし

<option>の属性

属性名	説明	必須	設定値	デフォルト値
name	オプション名を設定します。	○	文字列	なし
value	オプションの値を設定します。	○	文字列	なし

SAStruts用設定ファイル (SAStruts+S2JDBCにてデータベースを利用する場合)

SAStruts+S2JDBCフレームワークを利用する場合の設定の変更、確認を行います。

- app.diconの編集を行います。
 1. s2jdbc.dicon のコメントアウトをはずして、include を有効にします。

```
<include path="convention.dicon"/>
<include path="aop.dicon"/>
<include path="jee.dicon"/>
<include path="s2jdbc.dicon"/>
```

コラム

warファイルを展開後、s2jdbc.diconにてどのデータベースを利用するかを定義する dialect の設定が必要になります。
 接続したいデータベースに合わせて <property name="dialect"> の定義をコメントの外に出して有効にしてください。
 利用するデータベースに対してどのdialectを設定すれば良いかは、http://s2container.seasar.org/2.4/ja/s2jdbc_setup.html を参照してください。

SAStruts用設定ファイル (SAStruts版ポートレットを利用する場合)

SAStruts版のポートレットを利用する場合、設定ファイルの変更が必要となります。

- struts-config.xmlファイルの編集を行います。
 1. 設定ファイルを「ProjectNavigator」上に追加します。
 - 次の方法があります。
 - ・ IM-Juggling プロジェクトを作成するウィザード中の追加リソースの配置より、「SAStruts用設定ファイル」を追加します。
 - ・ IM-Juggling プロジェクト作成後に「追加リソースの選択」より「SAStruts用設定ファイル」を追加します。
 2. 「ProjectNavigator」内の <struts-config.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
 3. 次の設定を変更します。

変更前 : processorClass="jp.co.intra_mart.framework.extension.seasar.struts.action.IMS2RequestProcessor"

変更後 : processorClass="jp.co.intra_mart.framework.extension.seasar.struts.portlet.action.IMS2RequestProcessor"

TERASOLUNA Server Framework for Java (5.x)

TERASOLUNA Server Framework for Java (5.x) 用設定ファイル (シェアードデータベースを利用する場合)

TERASOLUNA Server Framework for Java (5.x) よりAccel Platformのシェアードデータベースを利用する場合、下記設定ファイルにシェアードデータベースの登録が必要となります。

- classes/META-INF/spring/applicationContext-im_tgfw_common.xmlファイルの編集を行います。
 1. applicationContext-im_tgfw_common.xml の下記コメントアウトをはずして、connectIdパラメータのvalue値にシェアードデータベースでユニークなIDとなる「接続ID」を指す。

```
<bean id="sharedDataSource" class="jp.co.intra_mart.framework.extension.spring.datasource.SharedDataSource">
  <constructor-arg name="connectId" value="＜接続ID＞" />
</bean>
```

コラム

「接続ID」は、テナントにシェアードデータベースを登録する際に指定します。
 テナントにシェアードデータベースを登録する方法については、[システム管理者操作ガイドのシェアードデータベース設定](#) を参照してください。

コラム

複数のシェアードデータベースを利用する場合は、beanタグをそれぞれ定義し、beanのid属性値がそれぞれ一意になるように設定してください。

TERASOLUNA Server Framework for Java (5.x) 用設定ファイル (リポジトリ層にMyBatis3を利用する場合)

TERASOLUNA Server Framework for Java (5.x) のリポジトリ層にMyBatis3を利用する場合、下記設定ファイルの確認および変更が必要となります。

テナントデータベース用の設定

テナントデータベース用の設定は、classes/META-INF/spring/applicationContext-mybatis3.xmlファイルに設定されております。
セットアップ方法の詳細は「 [TERASOLUNA Server Framework for Java \(5.x\) プログラミングガイド](#) 」-「 [MyBatis-Springの設定](#) 」を参照してください。

シェアードデータベースを利用する場合の設定

シェアードデータベースを利用する場合は、applicationContext-im_tgfw_common.xml及びMyBatis-Springの設定を編集する必要があります。
セットアップ方法の詳細は「 [TERASOLUNA Server Framework for Java \(5.x\) プログラミングガイド](#) 」-「 [シェアードデータベースの利用](#) 」を参照してください。

IMBox

IMBox を利用するための IMBox モジュールを説明します。
IMBox で利用する Apache Cassandra の設定方法を解説します。

項目

- IMBox モジュールの機能
- IMBox モジュールの利用
- Cassandraサーバ接続設定ファイル (cassandra-config.xml)
- intra-mart Accel Platform 2014 Spring(Granada)からの変更点

IMBox モジュールの機能

IMBox を利用するためのモジュールです。

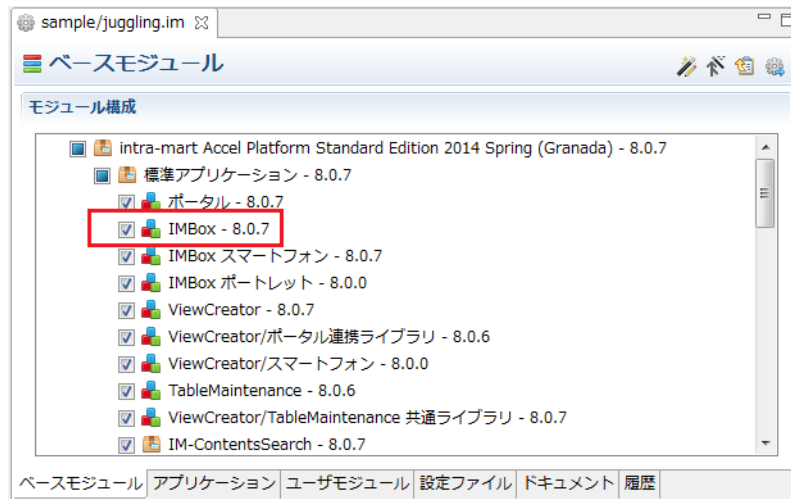
IMBox モジュールでは、以下の機能を提供します。

- IMBox を利用することができます。

IMBox モジュールの利用

IMBox モジュールを利用する場合は、IM-Juggling で IMBox モジュールを選択してください。

IM-Juggling のプロジェクト作成時の初期設定では、IMBox モジュールは選択済みになっています。



i コラム

IMBox をスマートフォンで利用する場合は、IMBox スマートフォンモジュールが必要になります。
IMBox をポートレットで利用する場合は、IMBox ポートレットモジュールが必要になります。

Cassandraサーバ接続設定ファイル (cassandra-config.xml)

設定内容

Cassandraサーバ接続に関する設定情報を保持するファイルです。

! 注意

cassandra-configへの設定値は、intra-mart Accel Platform 2014 Spring(Granada)から使用用途が変更されました。
詳細は [intra-mart Accel Platform 2014 Spring\(Granada\)からの変更点](#) を参照してください。

! 注意

cassandra-config.xmlには、Cassandraサーバの構築時に設定した<%CASSANDRA_HOME%/conf/cassandra.yaml>ファイルの接続情報を指定してください。
Apache Cassandraの設定に関する詳細は「IMBox Cassandra管理者ガイド」を参照してください。

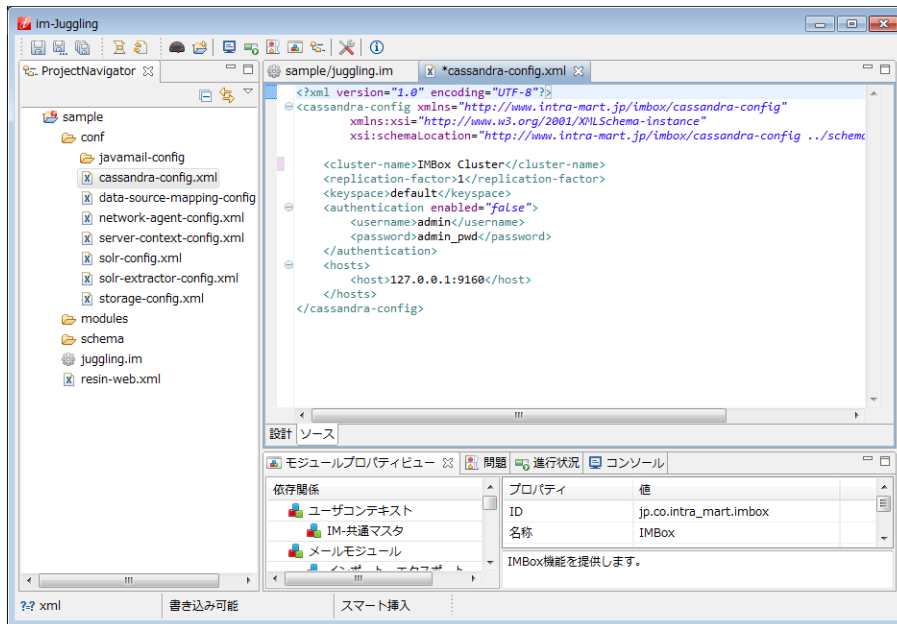
- [IMBox Cassandra管理者ガイド \(日本語\)](#)
- [IMBox Cassandra管理者ガイド \(English\)](#)
- [IMBox Cassandra管理者ガイド \(中文\)](#)

設定方法

以下は標準で用意されているcassandra-config.xmlの一部です。

```
<cluster-name>Test Cluster</cluster-name>
<replication-factor>1</replication-factor>
<keyspace>default</keyspace>
<authentication enabled="false">
  <username>admin</username>
  <password>admin_pwd</password>
</authentication>
<hosts>
  <host>127.0.0.1:9160</host>
</hosts>
```

- IM-Juggling で設定する場合



設定項目

Cassandraサーバ接続設定ファイルの項目を説明します。

cassandra-config.xmlの要素

タグ	説明	必須	設定値	デフォルト値	複数記
<cluster-name>	Cassandraサーバのクラスタ名を設定します。 初期値に設定してある“Test Cluster”は <%CASSANDRA_HOME%/conf/cassandra.yaml>ファイルの初期値と なっています。	○	クラスタ名	Test Cluster	×
<replication-factor>	分散構成とするクラスタ内部のデータのレプリカ数を設定します。 レプリカ数の設定は、「 Cassandra管理者ガイド - Cassandra のクラスタ構築 」を参照してください。	○	数値	1	×
<keyspace>	Cassandraサーバのキースペースを設定します。 新規に intra-mart Accel Platform を構築する場合は、初期値の“default” のみまで問題ありません。 複数の intra-mart Accel Platform から同じ Cassandraクラスタを利用する 場合は、intra-mart Accel Platform のテナント毎に異なるキースペースを指 定してください。	○	キースペース	default	×
<authentication>	Cassandraへの接続に対する認証情報を設定します。 認証情報の設定はenabled属性の値に“true”を設定した場合のみ有効とな ります。 認証情報を使用する場合、Cassandra自体への接続認証の設定を行う必 要があります。 接続認証に関しては、「 Cassandra管理者ガイド - Cassandra への接続認 証設定 」を参照してください	○	なし	なし	×
<username>	接続ユーザ名を設定します。 認証情報を使用する場合のみ設定が適用されます。	認証設定を使用する場合のみ○	接続ユーザ名	admin	×
<password>	接続パスワードを設定します。 認証情報を使用する場合のみ設定が適用されます。	認証設定を使用する場合のみ○	接続パスワード	admin_pwd	×
<hosts>	Cassandraが稼働しているサーバの設定を行います。	○	なし	なし	×
<username>	Cassandraが稼働しているサーバのIPアドレスとポート番号の設定を行いま す。 分散構成で複数のCassandraが稼働している場合、すべてのCassandra サーバのIPアドレスとポート番号を設定してください。	○	IPアドレス:ポート 番号	127.0.0.1:9160	○



注意

Cassandraサーバが複数存在する場合、クラスタ名で同一クラスタのCassandraであるか判断され、クラスタが組まれてしまう場合があります。
新規に intra-mart Accel Platform を構築する場合、初期値を変更することを推奨します。

intra-mart Accel Platform 2014 Spring(Granada)からの変更点

cassandra-configの設定値は以下のように使用用途が変更されました。

- intra-mart Accel Platform 2013 Winter(Felicia)までの場合
 - Cassandraへの接続の設定値
- intra-mart Accel Platform 2014 Spring(Granada)以降の場合
 - テナント環境セットアップのCassandra接続情報の初期値



コラム

intra-mart Accel Platform 2014 Spring(Granada)からは、テナント管理 - Cassandra接続情報 で登録したCassandra接続情報が接続時に使用されます。

Cassandra接続情報画面の初期値と設定値

Cassandra接続情報の項目名	cassandra-configの設定値	備考
クラスタ名	cluster-name	
キースペース	keyspace	
接続先	host	hostが複数設定されている場合、すべて反映されます。
レプリケーションファクタ	replication-factor	
認証情報設定	authentication	
認証ユーザ名	username	authenticationのenabled属性がtrueの場合のみ表示されます。
認証パスワード	password	authenticationのenabled属性がtrueの場合のみ表示されます。

- Cassandra接続情報画面

テナント 設定

Step 1 Step 2 Step 3 Step 4 Step 5 Step 6 Step 7 Step 8 Step 9

Step 7 - Cassandra接続情報

クラスタ名 *	IMBox Cluster
キースペース *	default
接続先 *	127.0.0.1:9160
レプリケーションファクタ *	1
認証情報設定	☒ 設定する 認証情報が必要なCassandraへの接続時のみ、設定してください。認証情報を設定する場合は書き込み権限の確認を行うため、事前にキースペースを作成しておく必要があります。
認証ユーザ名 *	admin
認証パスワード *	*****

テスト接続

次へ

IM-ContentsSearch

IM-ContentsSearch for Accel Platform で利用する Apache Solr の設定を行います。

項目

- IM-ContentsSearch の機能
- IM-ContentsSearch モジュールの利用
- IM-ContentsSearch 検索対象追加モジュールの利用
- Solrサーバ接続設定 (solr-config.xml)
 - intra-mart Accel Platform 2014 Spring(Granada)からの変更点
- テキスト抽出設定 (solr-extractor-config.xml)

IM-ContentsSearch の機能

intra-mart Accel Platform で全文検索機能を実現するためのモジュールです。

全文検索機能を利用するためには「Apache Solr」サーバを構築する必要があります。

検索対象を追加するには検索対象追加モジュールを追加する必要があります。

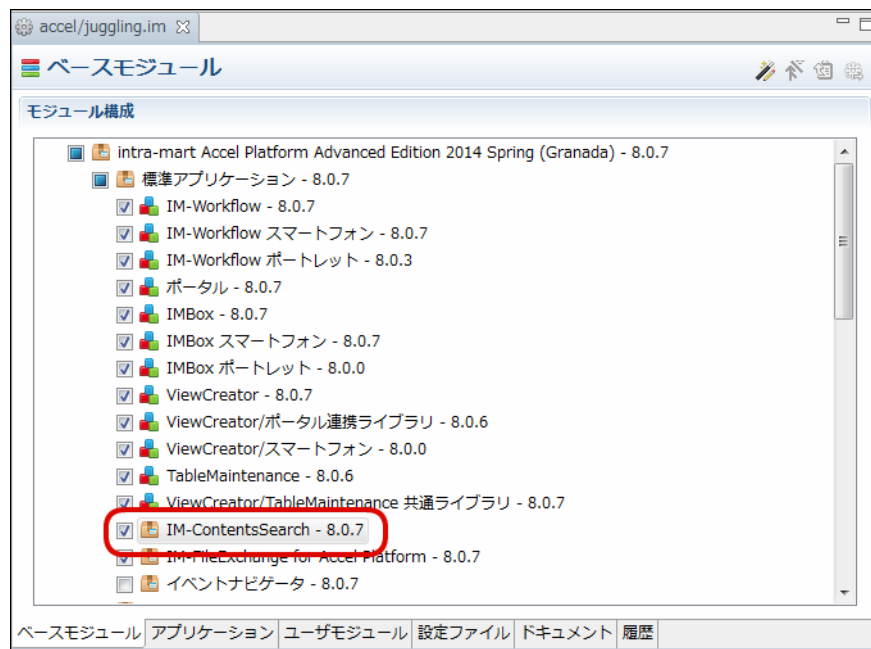
i コラム

- Apache Solr のセットアップについては、「[Solr管理者ガイド](#)」を参照してください。
- 追加した検索対象のコンテンツを全文検索機能で検索できるようにするためには、検索対象追加モジュールの機能に含まれているジョブをジョブスケジューラ機能で実行する必要があります。
IM-ContentsSearch のジョブ・ジョブネットについては「[ジョブ・ジョブネットリファレンス](#)」を参照してください。

IM-ContentsSearch モジュールの利用

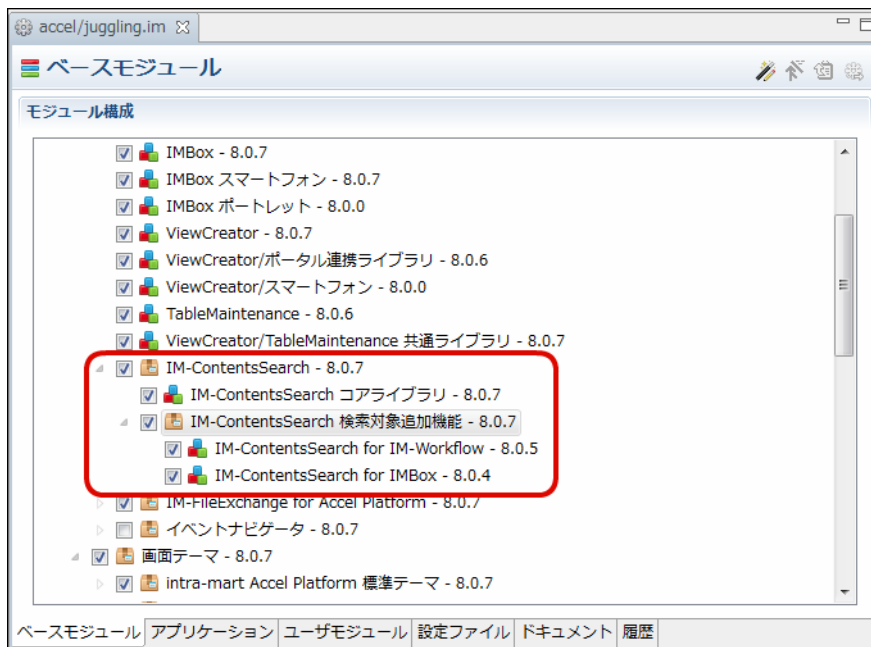
IM-ContentsSearch モジュールを利用する場合は、IM-Juggling で IM-ContentsSearch モジュールを選択してください。

IM-Juggling のプロジェクト作成時の初期設定では、IM-ContentsSearch モジュールは選択済みになっています。



IM-ContentsSearch 検索対象追加モジュールの利用

検索対象追加モジュールを変更する場合、IM-ContentsSearch > IM-ContentsSearch 検索対象追加機能 のモジュールのツリー配下のモジュールを変更してください。



Solrサーバ接続設定 (solr-config.xml)

設定内容

Solrサーバへの接続設定情報を保持するファイルです。

注意

solr-config.xmlの設定値は、intra-mart Accel Platform 2014 Spring(Granada)から使用用途が変更されました。
詳細は [intra-mart Accel Platform 2014 Spring\(Granada\)からの変更点](#) を参照してください。

設定方法

以下は標準で用意されているsolr-config.xmlの一部です。

```
<group name="default">
  <searcher>
    <method>POST</method>
    <distribution-policy>FIRST</distribution-policy>
    <servers>
      <url>http://localhost:8983/solr/default</url>
    </servers>
  </searcher>
  <indexer>
    <distribution-policy>IDHASH</distribution-policy>
    <servers>
      <url>http://localhost:8983/solr/default</url>
    </servers>
  </indexer>
  <extractor ref="im_default" />
</group>
```

設定項目

solr-config.xmlの要素

タグ	説明	必須	設定値	デフォルト値	複数設定
<group>	Solrサーバグループ名を設定します。 <group>のnameにSolrサーバグループ名を設定します。 初期値の "default" のままで問題ありません。	○	Solrサーバグループの設定	なし	○
<searcher>	IM-ContentsSearchが検索時に接続するSolrサーバを設定します。	○	検索用Solrサーバの設定	なし	×
<method>	IM-ContentsSearchが検索時にSolrサーバへ送るリクエストのメソッドを設定します。 通常はPOSTから変更する必要はありません。	×	検索時に使用するリクエストメソッドの設定	POST	×
<distribution-policy>	IM-ContentsSearchが検索リクエストを送るSolrサーバを選択するポリシーを設定します。 検索用と索引作成用で利用可能なポリシーが異なります。	×	検索用Solrサーバの選択ポリシー設定	FIRST	×
<servers>	IM-ContentsSearchが検索時に接続するSolrサーバ一覧を設定します。 Solrサーバ設定は配下のurlタグで行います。	○	検索用Solrサーバ一覧の設定	なし	×

タグ	説明	必須	設定値	デフォルト値	複数設定						
<url>	IM-ContentsSearchが検索時に接続するSolrサーバのURLを設定します。 Solrサーバを構築しているホストのアドレス、および、ポート番号を指定してください。 URL内の「default」は Apache Solr core名であり、Solrサーバグループ名とは別の設定値であるため、変更しないでください。 参考：各Web Application Serverのデフォルトのポート番号 <table><tr><td>Resin</td><td>8080</td></tr><tr><td>Tomcat</td><td>8080</td></tr><tr><td>Jetty</td><td>8983</td></tr></table>	Resin	8080	Tomcat	8080	Jetty	8983	○	検索用SolrサーバのURL の設定	なし	○
Resin	8080										
Tomcat	8080										
Jetty	8983										
<indexer>	IM-ContentsSearchが索引作成時に接続するSolrサーバを設定します。	○	索引作成用Solrサーバの 設定	なし	○						
<distribution-policy>	索引作成用サーバを選択するポリシーを設定します。 検索用と索引作成用で利用可能なポリシーが異なります。	×	索引作成用Solrサーバの 選択ポリシー設定	IDHASH	×						
<servers>	IM-ContentsSearchが索引作成時に接続するSolrサーバ一覧を設定します。 Solrサーバ設定は配下のurlタグで行います。	○	索引作成用Solrサーバ一 覧の設定	なし	×						
<url>	IM-ContentsSearchが索引作成時に接続するSolrサーバのURLを設定しま す。 defaultは Apache Solr core名であり、Solrサーバグループ名とは別の設定値 であるため、変更しないでください。	○	索引作成用Solrサーバの URLの設定	なし	○						
<extractor>	IM-ContentsSearchが索引作成時にテキスト抽出するファイルの設定グルー プを指定します。 ref属性にsolr-extractor-config.xmlに定義されたテキスト抽出設定グループ IDを指定します。	○	テキスト抽出設定グルー プの指定	なし	×						

intra-mart Accel Platform 2014 Spring(Granada)からの変更点

intra-mart Accel Platform 2014 Spring(Granada)以降のバージョンをご利用の場合、solr-configの設定値は以下のように使用用途が変更されました。

- 初回テナント環境セットアップ時、または、Solr接続設定が1件も登録されていない場合のテナント環境セットアップ時に登録されるSolr接続設定
- Solr接続設定新規作成画面の初期値



コラム

<group>のnameが"default"である設定のみ使用されます。

テキスト抽出設定 (solr-extractor-config.xml)

設定内容

以下の内容に関する設定情報を保持するファイルです。

- 共通パラメータの設定
- テキスト抽出対象ファイルの設定
- テキスト抽出方法の設定

設定方法

以下は標準で用意されているsolr-extractor-config.xmlの一部です。

```
<extractor name="im_default">
  <default>
    <min-size>0</min-size>
    <max-size>100M</max-size>
    <properties>
      <!-- temporary directory sample for Linux -->
      <property name="tmpDir" type="string">/tmp</property>
      <!-- temporary directory sample for Windows -->
      <!--
      <property name="tmpDir" type="string">C:\temp</property>
      -->
    </properties>
  </default>

  <mapping>
    <!-- Entry for text files -->
    <entry type="text/plain" extension="txt" max-size="1M" class="jp.co.nttdata.intra_mart.contentssearch.text.extraction.extractor.PlainTextExtractor" />
    ...
  </mapping>
</extractor>
```

設定項目

solr-extractor-config.xmlに設定する項目を記載します。

一部の要素のみ記載しているため、詳細は [設定ファイルリファレンス テキスト抽出設定](#) を参照してください。

solr-extractor-config.xmlの要素(一部)

タグ	説明	必須	設定値	デフォルト値	複数設
<extractor>	一つのテキスト抽出設定グループを表します。 name属性にはテキスト抽出設定グループの名称を指定します。	○	テキスト抽出設定グループ	なし	○
<min-size>	テキスト抽出対象ファイルサイズ最小値の既定値を設定します。	×	テキスト抽出対象ファイル サイズ最小値の既定値	0 (制限なし)	×
<min-size>	テキスト抽出対象ファイルサイズ最大値の既定値を設定します。	×	テキスト抽出対象ファイル サイズ最大値の既定値	0 (制限なし)	×
<property>	プロパティの既定値を設定します。	×	プロパティ既定値の設定	なし	○



注意

一時ディスク領域 (propertyタグのname属性"tmpDir") は必ず設定してください。
指定するディレクトリは、intra-mart Accel Platform の実行ユーザからアクセス可能である必要があります。

IM-Workflow

IM-Workflow システム設定

IM-Workflow のシステム共通設定について設定します。

i コラム

IM-Workflow システム設定は IM-Workflow に含まれています。

1. 「ProjectNavigator」内の <conf/im-workflow-system-config.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. 設定の内容を運用に応じて変更します。

i コラム

詳細については「[IM-Workflow 仕様書](#)」の「4.1.1.1 システム設定」を参照してください。

i コラム

IM-Workflow デザイン設定は IM-Workflow に含まれています。

1. 「ProjectNavigator」内の <conf/im-workflow-designer-config.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. 設定の内容を運用に応じて変更します。

i コラム

詳細については「[IM-Workflow 仕様書](#)」の「4.1.1.2 デザイン設定」を参照してください。

! 注意

<im-workflow-designer-config>/<icon> のアイコンパス定義を標準から変更する場合、または、ノードアイコンの差し替えを行う場合は、ユーザモジュールによるアイコン配置・差し替えが必要となります。

定義したアイコンパスが示す場所（システムストレージ配下）に、差し替え後のアイコンがデプロイされるようユーザモジュールを作成し、warにユーザモジュールを含めてください。

ユーザモジュールの作成方法については「[e Builder for Accel Platform / ユーザ操作ガイド](#)」を参照してください。

なお、ユーザモジュールではmodule.xmlを編集し、IM-Workflow（モジュールID:jp.co.intra_mart.im_workflow）モジュールを「依存するモジュール」として必ず指定してください。指定がない場合、アイコンの差し替えが正しく行われない可能性があります。

i コラム

モジュール・プロジェクトの作成例を示します。

モジュール・プロジェクト作成例の前提

プロジェクト名（アーティファクトID）	workflow_user_icon_replace
グループID	mypackage

- module.xml の例

IM-Workflow（モジュールID:jp.co.intra_mart.im_workflow）モジュールを「依存するモジュール」として指定します。

```
<module xmlns="urn:intramart:jackling:module" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:conf="urn:intramart:jackling:toolkit:configurations"
  xsi:schemaLocation="urn:intramart:jackling:module module.xsd"
  conf:schemaLocation="urn:intramart:jackling:toolkit:configurations configurations.xsd">

  <id>mypackage.workflow_user_icon_replace</id>
  <version>1.0.0</version>
  <type>module</type>

  <name>${module.name}</name>
  <vendor>${module.vendor}</vendor>
  <description>${module.description}</description>

  <!-- 変更を不可とする場合やサードパーティモジュールの場合のみ以下を指定する -->
  <tags>
    <tag>immutable</tag>
    <tag>3rd-party</tag>
  </tags>

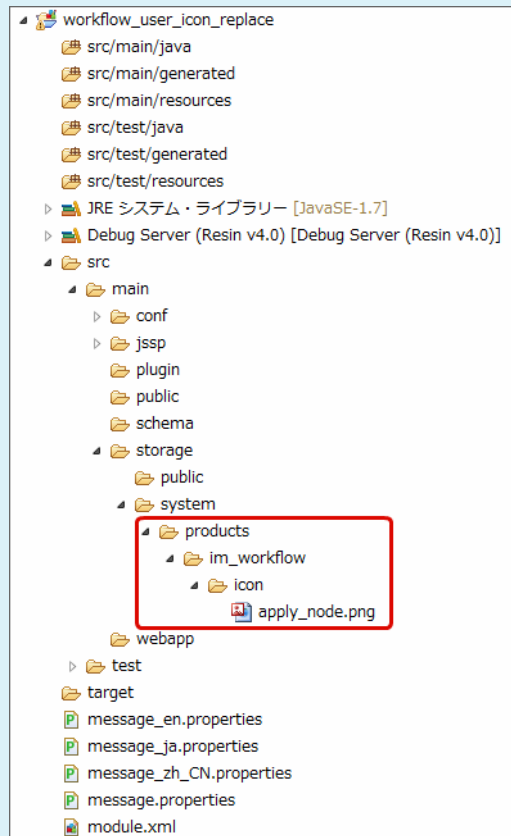
  <dependencies>
    <dependency>
      <module-id>jp.co.intra_mart.im_workflow</module-id>
      <verified-version min="8.0.7">8.0.7</verified-version>
    </dependency>
  </dependencies>

</module>
```

- モジュール・プロジェクトのリソース配置例1

<im-workflow-designer-config>/<icon> のアイコンパス定義は標準のままとし、申請ノードアイコンを差し替える場合

差し替えるアイコンのみを配置します。



■ モジュール・プロジェクトのリソース配置例2

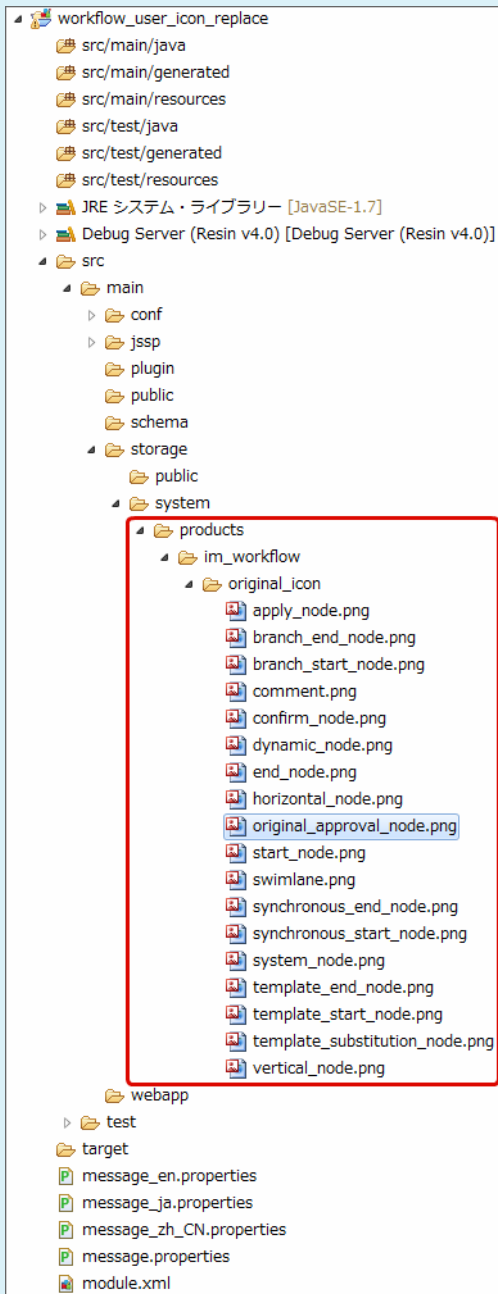
<im-workflow-designer-config>/<icon> のアイコンパス定義の変更を以下の通りに指定した場合

※記載のないパラメータについては、標準の値から変更なしの状態とする。

アイコンパス定義の設定

node-icon-dir	products/im_workflow/original_icon/
approval-node-icon	original_approval_node.png

ノードアイコン保存ディレクトリを変更しているため、変更後のディレクトリにすべてのノードアイコンを配置します。
また、承認ノードについては、変更後のアイコンファイル名と合致する承認ノードアイコンファイルを配置します。



IM-Workflow 用設定ファイル (シーケンスオブジェクト採番機能を利用する場合)

シーケンスオブジェクト採番機能を利用する場合、プラグイン設定の変更が必要となります。

IM-Workflow の WorkflowNumberingManager を利用して連続した番号を取得する際にデータベースのシーケンスオブジェクトを利用して採番処理を行う機能です。



注意

シーケンスオブジェクト採番機能は、2013 Winter(8.0.6) から利用することが可能となります。

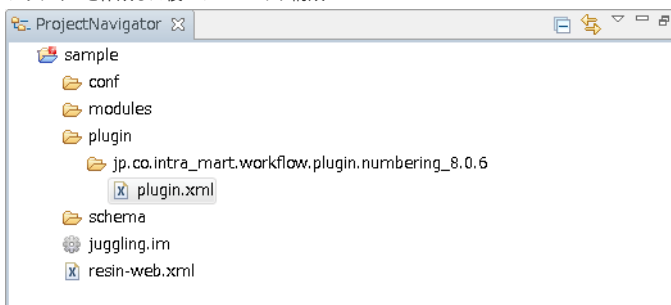
シーケンスオブジェクト採番機能の利用

シーケンスオブジェクト採番機能を利用する場合は、以下の手順でプラグインを作成します。

1. IM-Juggling でプロジェクトを選択し、フォルダを作成します。
フォルダは「plugin/jp.co.intra_mart.workflow.plugin.numbering_8.0.6」にしてください。
2. 作成されたフォルダに「plugin.xml」を作成します。
「plugin.xml」に以下の内容をコピーしてください。

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin>
  <extension point="jp.co.intra_mart.workflow.plugin.numbering" >
    <numbering
      name="simpleNumberCounter"
      id="jp.co.intra_mart.workflow.plugin.numbering.simpleNumberCounter"
      version="8.0.6"
      rank="0"
      enable="false">
      <extend>
        <java class="jp.co.intra_mart.system.workflow.plugin.numbering.SimpleNumberCounterEvent" />
      </extend>
    </numbering>
    <numbering
      name="databaseSequence"
      id="jp.co.intra_mart.workflow.plugin.numbering.databaseSequence"
      version="8.0.6"
      rank="0"
      enable="true">
      <extend>
        <java class="jp.co.intra_mart.system.workflow.plugin.numbering.DatabaseSequenceEvent" />
      </extend>
    </numbering>
  </extension>
</plugin>
```

- プラグインを作成した後のプロジェクト構成



iPadからアクセス時にクライアントタイプをPCとして扱う場合

intra-mart Accel Platform ではアクセスする端末によりクライアントタイプが割り当てられます。

intra-mart Accel Platform で標準で提供しているクライアントタイプは以下の通りです。

クライアントタイプID	提供バージョン	提供モジュール	備考
pc	2012 Autumn	マルチデバイス	PC端末からの利用を想定しています。(標準)
sp	2012 Autumn	IM-Mobile Framework	スマートフォン、タブレットなどのモバイル端末から想定しています。

intra-mart Accel Platform では標準設定では iPad がスマートフォン端末として扱われます。

この章では、iPad をPC端末として扱う(スマートフォン端末として扱わない)設定方法を紹介します。

iPadプラグインの無効化

クライアントタイプの判別を行う設定方法としてpluginを採用しています。

pluginの設定仕様については、「[PluginManagerのJavaDoc](#)」を参照してください。

以下ではiPadのクライアントタイプを sp と判別している設定を無効化します。

1. IM-Juggling でプロジェクトを選択し、フォルダを作成します。
フォルダは「plugin/ignore_sp_for_ipad」にしてください。
2. 作成されたフォルダに「plugin.xml」を作成します。
「plugin.xml」に以下の内容をコピーしてください。

```
<?xml version="1.0" encoding="utf-8"?>
<plugin>
  <extension point="jp.co.intra_mart.foundation.multi_device.client_type.matcher">
    <client-type-config id="ipad" version="8.0.999" enable="false" />
  </extension>
</plugin>
```



コラム

pluginのバージョン機能を利用して、ipad に対しての設定を優先的に無効化します。

自動ログイン機能を利用する場合

intra-mart Accel Platform は、ログイン実行ページURLに対してリクエストパラメータを付与することにより、自動的にログインすることが可能です。

例えば、以下のURLでアクセスすると、ユーザ「ueda」でログインし「/imbox/mybox」に遷移します。

```
http://<HOST>:<PORT>/<CONTEXT_PATH>/certification?im_user=ueda&im_password=(パスワード)&im_url=/imbox/mybox
```

自動ログインの詳細は「[設定ファイルリファレンス - 認証機能 - 認証設定\(一般ユーザ用\) - 自動ログインについて](#)」を参照してください。

この自動ログイン機能は、セキュリティの観点からデフォルトでは無効化されています。

この章では、自動ログイン機能を有効にする方法を紹介します。

自動ログイン機能の有効化

- 「conf/token-filtering-target-config/im_certification.xml」ファイルの編集を行います。
 1. 設定ファイルを「ProjectNavigator」上に追加します。
 1. <(プロジェクト名)/juggling.im> ファイルをダブルクリックします。
 2. 「設定ファイル」タブをクリックします。
 3. 「認証機能」の「セキュア・トークンフィルタ設定(認証機能用)」を選択し、右側にある「出力」をクリックします。
 2. 「ProjectNavigator」内の <conf/token-filtering-target-config/im_certification.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
 3. 以下の設定をコメントアウトします。
 - 変更前

```
<p:entry url="/certification" />
```

- 変更後

```
<!-- <p:entry url="/certification" /> -->
```

外部メニュー連携

intra-mart Accel Platform の外部メニュー連携を行うための接続先情報を設定します。



注意

外部メニュー連携は、2013 Spring(Climbing) から利用することが可能となります。

項目

- 外部メニュー連携とは
- メニュークライアント環境構築手順
- メニュープロバイダから取得できる情報
- Webサービスモジュール

外部メニュー連携とは

外部メニュー連携とは、シングルサインオン(以下 SSO と記述します)を利用してシステムを構築している場合に、複数の [メニュープロバイダ](#) のメニュー情報を intra-mart Accel Platform 上メニューとして表示し、利用することを可能とするための機能です。

取得したメニュー情報は、テーマのグローバルナビ、ツリーメニューおよびサイトマップに表示されます。

intra-mart Accel Platform のログインユーザのユーザコードと、メニュープロバイダのユーザコードは一致している必要があります。

メニュープロバイダとは

メニュー情報を取得するための接続先の Webアプリケーションを、メニュープロバイダと呼びます。

外部メニュー連携機能では、メニュープロバイダから SOAP Webサービスを利用してメニュー情報を取得します。

メニュープロバイダには、あらかじめ Webサービスのモジュールをインストールする必要があります。

Webサービスモジュールに関しては、「[Webサービスモジュール](#)」を参照してください。

メニュークライアントとは

メニュープロバイダからメニュー情報を取得し、1つのメニュー情報に統合して表示する側の Webアプリケーションを、メニュークライアントと呼びます。

ユーザは、メニュークライアントとして稼働する intra-mart Accel Platform にアクセスすることで、複数の Webアプリケーションのサービスを意識することなく利用することが可能となります。

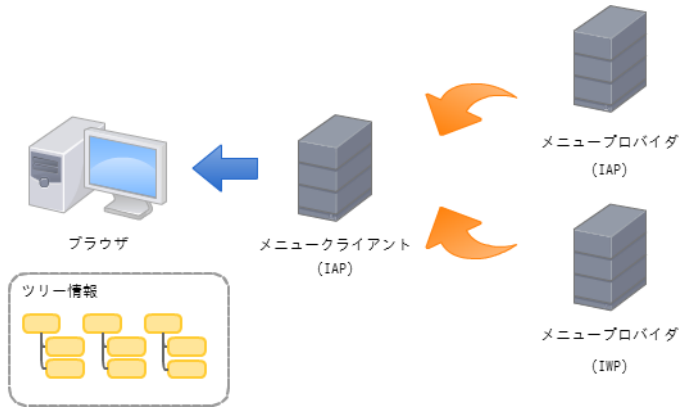


図 外部メニュー連携

！ 注意

外部メニュー連携機能を利用する場合は、連携する各アプリケーションサーバで、SSO による認証を行うことを推奨します。

SSOを利用せずに、外部メニュー連携機能を利用した場合、以下の様な制限事項が発生します。

- メニュープロバイダから取得したメニューにアクセスしても、自動的にログインされません。
該当のページに権限設定がある場合、エラーページが表示されます。
エラーページからログイン画面へ遷移してログインすることも可能ですが、ブラウザの設定や環境によっては正常にログインできない場合があります。
また、メニュープロバイダが intra-mart Accel Platform の場合、グローバルナビを表示しない制御を行っていますが、ログインしていない状態でアクセスした場合は制御が行
メニュークライアントのグローバルナビとメニュープロバイダのグローバルナビが2重に表示されます。
- メニュークライアントとメニュープロバイダに別々にログインすることになるため、同じユーザであることが保証されません。
外部メニュー連携で取得するメニュー情報は、メニュークライアントにログインしているユーザにひもづく情報です。
メニュープロバイダに別のユーザでログインしていても画面上では判断できないため、誤ったオペレーションを行う可能性があります。
- メニュークライアントからログアウトしても、メニュープロバイダ側は自動ログアウトされません。
メニュープロバイダに直接アクセスすることで、ログイン状態のままアクセスが可能です。

IM-HybridSSO

2014 Summer(Honoka) から、intra-mart WebPlatform を対象とした簡易SSO機能として IM-HybridSSO が利用可能です。

IM-HybridSSO については、「[IAP-IWP間SSO連携 \(IM-HybridSSO\)](#)」を参照してください。

IM-HybridSSO では、intra-mart WebPlatform v7.2 patch7 で利用するための個別パッチモジュールが提供されています。

個別パッチモジュールには IM-HybridSSO の機能以外に、patch8 で提供される予定の外部メニュー連携に関する不具合の修正が含まれます。

不具合の内容については、個別パッチモジュールに添付する readme.txt を参照してください。

メニュークライアント環境構築手順

外部メニュー連携モジュールの選択

IM-Juggling で、「追加機能」-「外部メニュー連携」モジュールを選択します。

メニュープロバイダの設定

「設定ファイル」タブから、「外部メニュー連携」-「外部メニュー連携接続先設定」を出力します。

「外部メニュー連携接続先設定 (menu-provider-config.xml)」に、以下のようにメニュープロバイダの設定を記述します。

```

<?xml version="1.0" encoding="UTF-8"?>
...
<menu-provider id="sample">
  <end-point> http://<HOST>:<PORT>/<CONTEXT_PATH>/services/MenuService </end-point>
  <login-group> <接続先のログイングループID> </login-group>
  <user> <接続先ユーザのユーザコード> </user>
  <password> <接続先ユーザパスワード> </password>
</menu-provider>
...
  
```

メニュープロバイダ情報について

設定する情報はあらかじめメニュープロバイダのシステム管理者から提供されている必要があります。

以下の情報をメニュープロバイダのシステム管理者に問い合わせてください。

接続先エンドポイント	メニュープロバイダの接続先URLです。 以下のフォーマットとなります。 <code>http://<HOST>:<PORT>/<CONTEXT_PATH>/services/MenuService</code>
------------	---

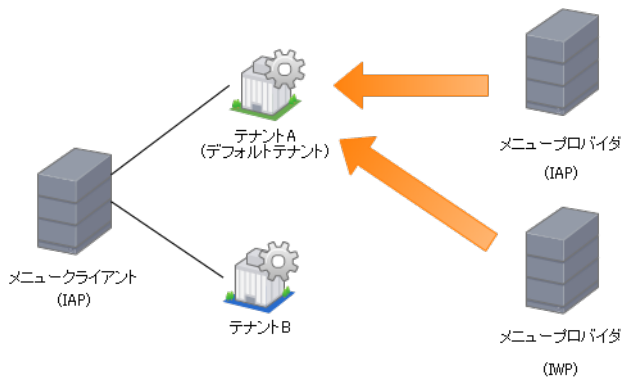
接続先ログイングループID	メニュープロバイダが intra-mart WebPlatform の場合のログイングループIDです。 メニュープロバイダが intra-mart Accel Platform の場合はテナントIDです。
---------------	---

接続先ユーザコード	Webサービスの権限を付与された Webサービス接続用ユーザのユーザコードです。
-----------	--

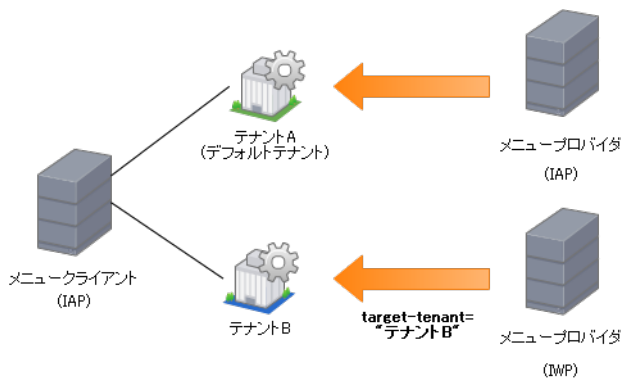
接続先ユーザパスワード Webサービスの権限を付与された Webサービス接続用ユーザのパスワードです。

バーチャルテナントによる複数テナント運用時のメニュープロバイダの設定について

バーチャルテナントによる複数テナントを運用する場合、標準ではメニュープロバイダの設定はデフォルトテナントに対して有効になります。



デフォルトテナント以外のテナントにメニュープロバイダを設定したい場合、「target-tenant」オプションを設定することで任意のテナントへメニュープロバイダを設定することができます。



メニュープロバイダの設定順序について

メニュープロバイダは複数設定することが可能です。

標準では、グローバルナビ、ツリーメニューおよびサイトマップには、設定された順に取得したメニュー情報が表示されます。

グローバルナビの場合は、「**editable**」オプションを設定することで、以下の様な動作となります。

- メニュー設定画面で表示対象を選択し、任意の位置に表示するように設定可能です。
- メニュー設定画面で選択しなかったメニュー情報は表示されません。

設定ファイルの詳細な設定内容については「[設定ファイルリファレンス](#)」-「[外部メニュー連携接続先設定](#)」を参照してください。

メニュー設定画面での表示順序の設定については「[テナント管理者操作ガイド](#)」-「[外部サイトのメニューリンクを登録する](#)」を参照してください。

注意

「**editable**」オプションは、2013 Autumn(Eden) から利用することが可能となります。

メニュープロバイダから取得できる情報

メニュープロバイダからは、以下の情報が取得できます。

- [一般ユーザが、メニューを表示する場合のメニュー情報](#)
- [メニュー管理者が、メニューを設定する場合のメニュー情報](#)

一般ユーザが、メニューを表示する場合のメニュー情報

一般ユーザが、グローバルナビやサイトマップにメニューを表示する場合に取得できるメニュー情報です。

メニュープロバイダからは、メニュークライアントにログインしているユーザが表示権限を持っているメニュー情報を取得します。
intra-mart Accel Platform のログインユーザのユーザコードと、メニュープロバイダのユーザコードは一致している必要があります。

取得できる主なメニュー情報は、以下の情報です。

- メニューの構成
- メニューの表示名
- メニュー情報の URL

「メニューの構成」は、メニュープロバイダで設定されているメニューのフォルダとアイテムの表示順、階層の情報です。

「メニューの表示名」は、メニュープロバイダのユーザロケールの表示名になります。

メニュープロバイダとメニュークライアントのユーザロケールが一致しない場合、表示名のロケールは統一されません。

「メニュー情報の URL」は、メニュープロバイダにアクセスする場合の絶対 URL となります。

メニュープロバイダが、プロキシサーバ等を利用してアクセスする必要がある場合は、メニュープロバイダ側で適切なベース URL を設定する必要があります。

次項「[Webサービスモジュール](#)」で提供される、各 WebサービスモジュールごとのベースURLの設定方法を確認し、設定を行ってください。

メニュー管理者が、メニューを設定する場合のメニュー情報

メニュー管理者が、メニュー設定画面で表示対象と表示先を設定する場合に、設定元として取得できるメニュー情報です。

メニュープロバイダからは、メニュークライアントのメニュー設定画面を操作しているユーザが、管理権限を持っているメニュー情報を取得します。
intra-mart Accel Platform のログインユーザのユーザコードと、メニュープロバイダのユーザコードは一致している必要があります。

取得できる主なメニュー情報は、以下の情報です。

- メニューの構成
- メニューの表示名
- メニュー情報の ID

「メニューの構成」は、メニュープロバイダで設定されているメニューのフォルダとアイテムの表示順、階層の情報です。

「メニューの表示名」は、メニュープロバイダのユーザロケルの表示名になります。

ここで取得した表示名は、メニュー設定画面でのみ利用されます。

グローバルナビなどに表示する場合は、「[一般ユーザが、メニューを表示する場合のメニュー情報](#)」で取得した表示名が利用されます。

「メニュー情報の ID」は、グローバルナビなどに表示する場合の対象のメニュー ID となります。

メニュー設定画面で表示先を設定した場合、「プロバイダの ID」と「メニュー情報の ID」が保存されます。



注意

外部メニュー連携サービス for intra-mart WebPlatform v7.2 で提供されるモジュールでは、一般ユーザは管理権限を持たないため、常に全てのメニュー情報を取得します。

Webサービスモジュール

イントラマートより提供される Webサービスは、以下のモジュールです。

- 外部メニュー連携サービス for intra-mart Accel Platform
- 外部メニュー連携サービス for intra-mart WebPlatform v7.2

外部メニュー連携サービス for intra-mart Accel Platform

外部メニュー連携サービス for intra-mart Accel Platform は、オプションモジュールです。

IM-Juggling で、「追加機能」-「Webサービス向け機能」-「外部メニュー連携 Webサービス」モジュールを選択して War を作成します。

プロキシサーバ等を利用している場合、メニュー情報生成のため、ベースURLの設定が必要です。

ベースURL設定は intra-mart Accel Platform 共通の設定を利用しています。「[ベースURL](#)」の章を参照してください。

Webサービスモジュールを有効にするためには、認可の設定を行う必要があります。

Webサービス接続用のユーザを新規に作成し、認可設定により以下のリソースに実行許可を与えてください。（既存のユーザに権限を与えることも可能です。）

以下の2つのリソースに同じ権限を設定してください。

- リソースの種類：Webサービス
- リソースグループ／リソース名：
 1. 外部メニュー連携／外部メニュー情報取得
 2. 外部メニュー連携／外部メニュー情報取得（管理）

メニュークライアントのシステム管理者に以下の情報を提供します。

Webサービス接続用エンドポイント 以下のフォーマットとなります。

`http://<HOST>:<PORT>/<CONTEXT_PATH>/services/MenuService`

Webサービス接続用ユーザコード 権限を与えられたユーザのユーザコードです。

Webサービス接続用ユーザパスワード 権限を与えられたユーザのパスワードです。

外部メニュー連携サービス for intra-mart WebPlatform v7.2

外部メニュー連携サービス for WebPlatform v7.2 は、patch6 で提供されました。

また、メニュークライアントで外部メニューの表示順序を設定するための機能は、patch7 で提供されました。

外部メニュー連携を利用するためには、最新のパッチを利用するようにしてください。

プロキシサーバ等を利用している場合、メニュー情報生成のため、ベースURLの設定が必要です。

このモジュールでは、以下の設定を利用しています。設定がない場合は、追加するようにしてください。

- 設定ファイル：<%ServerManager のインストールパス% /conf/parameter.xml>
- 設定内容：

```
<param>
  <param-name>im.web_server.url</param-name>
  <param-value>http://<アクセス可能なホスト名>:<PORT></param-value>
</param>
```

Webサービスモジュールを有効にするためには、ログイングループ管理者により、Webサービスアクセス設定を行う必要があります。

Webサービス接続用のロールとそのロールを付与したユーザを新規に作成し、Webサービスアクセス設定により以下のWebサービスに実行許可を与えてください。（既存のユーザに権限を付与することも可能です。）

以下の2つのオペレーションに同じ権限を設定してください。

- Webサービス名：MenuService
- オペレーション名：
 - getAvailableMenuTree
 - getManagedMenuTree (patch7 以降で提供されます。)

メニュークライアントのシステム管理者に以下の情報を提供します。

Webサービス接続用エンドポイント 以下のフォーマットとなります。

`http://<HOST>:<PORT>/<CONTEXT_PATH>/services/MenuService`

Webサービス接続先ログイングループID メニュー情報を取得するログイングループのIDです。

Webサービス接続用ユーザコード 権限を与えられたユーザのユーザコードです。

Webサービス接続用ユーザパスワード 権限を与えられたユーザのパスワードです。

！ 注意

intra-mart WebPlatform v7.2 patch6 では、外部メニューの表示順序を設定することができません。

intra-mart WebPlatform v7.2 patch6 を利用する場合は、メニュークライアントの設定ファイルに「`editable`」オプションを設定しないようにしてください。

i コラム

intra-mart Accel Platform のメニューから intra-mart WebPlatform のリンクをクリックした場合、intra-mart WebPlatform へのログインは自動的には行われません。自動的にログインが行われるようにするためには、別途 SSO 製品の対応が必要です。

2014 Summer(Honoka) から、簡易的にSSOを実現するための機能(IM-HybridSSO)が追加されました。

IM-HybridSSO を利用する場合は、「[iAP-iWP間SSO連携\(IM-HybridSSO\)](#)」を参照してください。

IM-SecureSignOn for Accel Platform (以下 IM-SSO と記述します) は、intra-mart Accel Platform に対応した SSO 製品です。

intra-mart WebPlatform の patch6 以降では、IM-SSO を利用した場合に、外部メニュー連携のメニューから自動的にログインが行われるようにするための機能が提供されています。機能を有効にするためには、IM-SSO 導入後、以下の設定を行ってください。

<%ApplicationRuntime のインストールパス% /doc/imart/WEB-INF/web.xml> を修正します。

(HTTPContextHandlingFilter の設定の直後に正しく設定を追加してください。設定箇所が間違っている場合、正常に動作しません。)

```
<filter>
  <filter-name>HTTPContextHandlingFilter</filter-name>
  <filter-class>jp.co.intra_mart.common.aid.jsdk.java.servlet.filter.HTTPContextHandlingFilter</filter-class>
</filter>
<!-- ↓↓↓ 以下を追加 -->
<filter>
  <filter-name>AutoLoginFilter</filter-name>
  <filter-class>jp.co.intra_mart.foundation.security.filter.AutoLoginFilter</filter-class>
</filter>
<!-- ↑↑↑ ここまで -->

:

<filter-mapping>
  <filter-name>HTTPContextHandlingFilter</filter-name>
  <servlet-name>MenuServlet</servlet-name>
</filter-mapping>
<!-- ↓↓↓ 以下を追加 -->
<filter-mapping>
  <filter-name>AutoLoginFilter</filter-name>
  <servlet-name>MenuServlet</servlet-name>
</filter-mapping>
<!-- ↑↑↑ ここまで -->
```

以上の設定により、メニューURL(～.menu)へのアクセス時に、VANADIS® Login Server で設定された Cookie 情報を参照して自動ログインを行います。

Cookie 情報が設定されている場合、intra-mart WebPlatform で一般ユーザのログイン・ログアウトは実行できません。

必ず VANADIS® Login Server におけるログイン・ログアウトを実行してください。

iAP-iWP間SSO連携(IM-HybridSSO)

intra-mart Accel Platform (以下 iAP と記述します) と iWP / iAF v7.2 (以下 iWP と記述します) の間で、シングルサインオン (以下 SSO と記述します) 連携するための機能 (IM-HybridSSO) について説明します。

！ 注意

IM-HybridSSO は、2014 Summer(Honoka) から利用することが可能となります。

！ 注意

IM-HybridSSO の iWP用のモジュールは、patch8 で提供されました。

iWP の patch7 で IM-HybridSSO を利用するためには、別途個別パッチモジュールをインストールする必要があります。

個別パッチモジュールは、以下の URL からダウンロード可能です。

http://www.intra-mart.jp/download/product/iap/archive/iwp727_id26113_20140731.zip

該当のファイルを解凍し、展開された readme.txt を参照してインストールおよび設定を行ってください。

なお、上記個別パッチモジュールを適用し、その後 patch8 の適用を行う場合、patch8 の適用によって一部設定ファイルが初期化されるためSSO連携が行えなくなります。
必ず patch8 の readme.txt を参照した上でパッチの適用を行ってください。



注意

IM-HybridSSO を利用する上での制限事項については、「[リリースノート](#)」-「[制限事項](#)」-「[HybridSSO・外部メニュー連携](#)」を参照してください。

項目

- 概要
 - IM-HybridSSO とは
 - IM-HybridSSO の構成
 - IM-HybridSSO の機能
- IM-HybridSSO の設定
 - iWP の設定
 - iAP の設定
 - 同一ホストで運用する場合の設定

概要

IM-HybridSSO とは

IM-HybridSSO とは、iAP と iWP の間で、SSO 連携するための機能です。
iWP の認証情報の管理と、外部メニュー連携を利用して取得したメニューからのアクセス時に iWP へ自動的にログインすることにより実現されます。
これにより、iAP と iWP のサービスをシームレスに利用することが可能となります。

外部メニュー連携については、「[外部メニュー連携とは](#)」を参照してください。

IM-HybridSSO を利用するためには、外部メニュー連携の設定が必要です。
「[IM-HybridSSO の設定](#)」では、IM-HybridSSO と外部メニュー連携の設定を併せて行いますので、手順に沿って設定を行ってください。

IM-HybridSSO の構成

IM-HybridSSO では、接続元と接続先を以下のように定義します。

SSO認証プロバイダ	接続元の iAP のログイン対象のテナント 外部メニュー連携機能のメニュークライアントとしての役割も担います。
SSOサービスプロバイダ	接続先の iWP のログイングループ 外部メニュー連携機能のメニュープロバイダとしての役割も担います。

IM-HybridSSO の基本構成は以下の通りです。

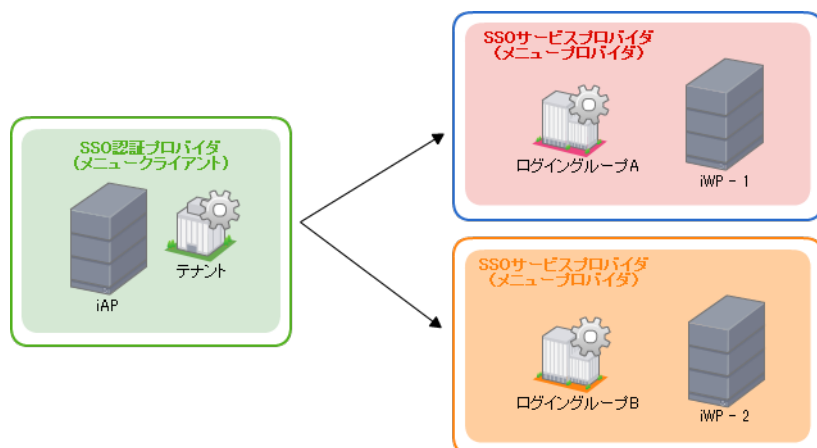


図 IM-HybridSSO の構成

設定はテナントごとに連携先のSSOサービスプロバイダを定義することで行います。

IM-HybridSSO の機能

IM-HybridSSO は、以下の機能により実現されます。

- 認証情報の管理
iAP にログインすることで、iWP にログイン確認を行い、ログイン可能な場合は、ブラウザの **Cookie** に認証情報を保存します。
- SSO自動ログイン
外部メニュー連携で取得した外部メニューからアクセスすることで、**Cookie** の認証情報を利用して iWP に自動的にログインし、該当のページを表示します。
- ログアウト連携
iAP からログアウトすることで、連携している全ての iWP からログアウトし、**Cookie** から認証情報を削除します。

IM-HybridSSO では、iAP にログインしたユーザのユーザコードを利用して iWP にログインします。
iWP に同一のユーザコードのユーザが存在しない場合は、SSO連携できません。

iAP と iWP のアカウント情報を同期するためには、マスター情報の同期機能の BackwardSync が利用可能です。
BackwardSync については「[BackwardSync\(version 7.2へのマスタ同期\)仕様書](#)」、および、「[BackwardSync\(version 7.2へのマスタ同期\)操作ガイド](#)」を参照してください。

IM-HybridSSO の設定

iWP の設定

SSO自動ログインの設定

IM-HybridSSO によるSSO自動ログインを有効にするため、以下の設定を行います。

 注意

修正内容の位置に正しく設定を追加してください。
設定箇所が間違っている場合、正常に動作しません。

1. メニューサーブレットにフィルタを追加します。

■ 設定ファイル

<%ApplicationRuntime のインストールパス%/doc/imart/WEB-INF/web.xml>

■ 修正内容

```

:

<filter>
  <filter-name>HTTPContextHandlingFilter</filter-name>
  <filter-class>jp.co.intra_mart.common.aid.jsdk.java.servlet.filter.HTTPContextHandlingFilter</filter-class>
</filter>
<!-- ↓ ↓ ↓ 以下を追加 -->
<filter>
  <filter-name>AutoLoginFilter</filter-name>
  <filter-class>jp.co.intra_mart.foundation.security.filter.AutoLoginFilter</filter-class>
</filter>
<!-- ↑ ↑ ↑ ここまで -->

:

<filter-mapping>
  <filter-name>HTTPContextHandlingFilter</filter-name>
  <servlet-name>MenuServlet</servlet-name>
</filter-mapping>
<!-- ↓ ↓ ↓ 以下を追加 -->
<filter-mapping>
  <filter-name>AutoLoginFilter</filter-name>
  <servlet-name>MenuServlet</servlet-name>
</filter-mapping>
<!-- ↑ ↑ ↑ ここまで -->

```

2. SSO自動ログイン用のリクエスト解析モジュールを追加します。

■ 設定ファイル

<%ServerManager のインストールパス%/conf/access-security.xml>

■ 修正内容

```

<user-security>

:

<initial-request-analyzer>
  <request-analyzer-class>jp.co.intra_mart.foundation.security.certification.ShortCutInitialRequestAnalyzer</request-analyzer-class>
</initial-request-analyzer>
<!-- ↓ ↓ ↓ 以下を追加 -->
<initial-request-analyzer>
  <request-analyzer-class>jp.co.intra_mart.system.hybrid_sso_provider.HybridSSORequestAnalyzer</request-analyzer-class>
</initial-request-analyzer>
<!-- ↑ ↑ ↑ ここまで -->

```

SSOエラーページの設定

以下のエラーページを IM-HybridSSO 用に差し替えます。

- セッションタイムアウトエラーページ
- アクティブセッション無効エラーページ
- ログインブロックエラーページ

差し替えには以下のファイルをそれぞれ修正します。

1. セッションタイムアウトエラーページ

■ 設定ファイル

<%ServerManager のインストールパス%/conf/access-security.xml>

■ 修正内容

```

:

<security-config>
<error-page-provider>
<page-provider-class>jp.co.intra_mart.foundation.security.certification.StandardErrorPageProvider</page-provider-class>

:

<init-param>
<!-- このパスを修正してください -->
<param-name>session-timeout-page</param-name>
<param-value>system/security/error/im_hybrid_sso/session_timeout_page.jsp</param-value>
</init-param>

:

<init-param>
<!-- スマートフォンを利用している場合は以下も修正してください -->
<param-name>session-timeout-page.sp</param-name>
<param-value>system/security/error/im_hybrid_sso/session_timeout_page_smartphone.jsp</param-value>
</init-param>

```

2. アクティブセッション無効エラーページ

- 設定ファイル

```
<%ServerManager のインストールパス% /conf/active-session-config.xml>
```

- 修正内容

```

:

<access-security-page-provider provider-class="jp.co.intra_mart.system.security.AccessSecurityPageProviderImpl">
<access-security-page page-id="active-session-invalidated">
<location request-header="maskat_layoutid">system/security/error/active_session_invalidated_page_maskat.jsp</location>
<location client-type="mobile">system/security/error/active_session_invalidated_page_mobile.jsp</location>

<!-- このパスを修正してください -->
<location>system/security/error/im_hybrid_sso/active_session_invalidated_page.jsp</location>

<!-- スマートフォンを利用している場合は以下も修正してください -->
<location client-type="sp">system/security/error/im_hybrid_sso/active_session_invalidated_page_smartphone.jsp</location>
</access-security-page>
</access-security-page-provider>

```

3. ログインブロックエラーページ

- 設定ファイル

```
<%ApplicationRuntime のインストールパス% /doc/imart/WEB-INF/web.xml>
```

- 修正内容

```

:

<filter>
<filter-name>LoginBlockFilter</filter-name>
<filter-class>jp.co.intra_mart.foundation.security.filter.LoginBlockFilter</filter-class>
<init-param>
<param-name>pagePath</param-name>
<!-- このパスを修正してください -->
<param-value>/system/security/error/im_hybrid_sso/login_block_page.jsp</param-value>
</init-param>
<init-param>
<param-name>mobilePagePath</param-name>
<param-value>/system/security/error/login_block_page_mobile.jsp</param-value>
</init-param>
<init-param>
<param-name>pagePath.sp</param-name>
<!-- スマートフォンを利用している場合は以下も修正してください -->
<param-value>/system/security/error/im_hybrid_sso/login_block_page_smartphone.jsp</param-value>
</init-param>
</filter>

```

SSOサービスプロバイダの設定

SSOサービスプロバイダのIDと接続先のSSO連携するログイングループの設定を行います。

- 設定ファイル

```
<%ServerManager のインストールパス% /conf/hybrid-sso-provider-config.xml>
```

- 設定例

```
<hybrid-ssso-provider-config
  xmlns="http://www.intra-mart.jp/hybrid_sso_provider/hybrid-ssso-provider-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/hybrid_sso_provider/hybrid-ssso-provider-config"
  provider-id="sample">

  <mappings>
    <mapping login-group="default" encrypt-key="default-enc-key" />
    <mapping login-group="other" encrypt-key="other-enc-key" />
  </mappings>

</hybrid-ssso-provider-config>
```

SSOサービスプロバイダ設定ファイルには、以下の設定を行います。

- プロバイダIDの設定

<hybrid-ssso-provider-config> の `provider-id` 属性にプロバイダIDを設定します。
プロバイダIDは、同一のSSOを構成するドメイン上の iWP に対して、ユニークなIDとなるように設定してください。

- ログイングループの設定

<mappings> 内にSSO連携に利用するログイングループの数だけ <mapping> タグを記述します。
`login-group` 属性には、ログイングループIDを設定します。
`encrypt-key` 属性には、ブラウザの Cookie に設定される認証情報を暗号化するためのキーを設定します。

注意

認証情報の暗号化方式は、Blowfish を利用しています。
そのため暗号化のキーには、Blowfish でキーとして設定可能な文字列長の文字列を設定する必要があります。

また、環境によりキーの文字列長が制限されている場合があります。
Windows 用の Oracle JDK の暗号化プロバイダの実装では、キーの文字列長はデフォルトでは128ビット(半角16文字)に制限されています。(日本で利用する場合)

`encrypt-key` 属性には、利用している環境で設定可能な文字列長の文字列を設定してください。

Oracle JDK の暗号化プロバイダについては、以下を参照してください。

<http://docs.oracle.com/javase/7/technotes/guides/security/SunProviders.html> (日本語) <http://docs.oracle.com/javase/7/docs/technotes/guides/security/SunProviders.html> (English)

ベースURLの設定

Webサーバを利用するなどして、ブラウザからのリクエストURLと iWP が受け付けるリクエストURLが異なる場合は、ベースURLの設定が必要です。

ベースURLを設定するために、以下の設定が必要です。

設定がない場合は、追加するようにしてください。

- 設定ファイル

<%ServerManager のインストールパス% /conf/parameter.xml>

- 修正内容

```
:

<param>
<param-name>im.web_server.url</param-name>
<param-value><http または https>://<アクセス可能なホスト名>:<PORT></param-value>
</param>
```

以下の構成の場合は、<param-value> に `http://www.intra-mart.jp` を設定します。

WebサーバリクエストURL	<code>http://www.intra-mart.jp/imart/～</code>
iWP が受け付けるリクエストURL	<code>http://iap.intra-mart.jp:8080/imart/～</code>

外部メニュー連携用モジュールでは、この設定を利用してメニュー情報の URL を生成しています。

この設定がない場合は、iWP が受け付けるリクエストURL を基にメニュー情報の URL が生成されるため、Webサーバ経由のアクセスができなくなります。

Webサービス実行ユーザ設定

Webサービスを実行する特定のユーザを定義します。

- Webサービス実行ユーザを定義します。
Webサービス実行用に新しくユーザを作成し、特定のロールを付与してください。
新しくユーザが作成できない場合は、既存のユーザにロールを付与して、実行用ユーザとして利用することも可能です。
- 「ログイングループ管理画面」の「Webサービスアクセスメニュー」より、以下のWebサービスにWebサービス実行ユーザの権限(1.で設定したロール)を付与します。
IM-HybridSSO 用と外部メニュー連携用のWebサービスがありますが、それぞれのオペレーションに同じ権限を設定してください。

Webサービス	オペレーション	説明
AdmissionService	publishKey	IM-HybridSSO のログイン確認と認証情報の作成。
MenuService	getAvailableMenuTree	表示用の外部メニューツリーの取得。 グローバルナビやサイトマップに表示されるメニューツリーを取得し

Webサービス	オペレーション	説明
MenuService	getManagedMenuTree	管理用の外部メニューツリーの取得。 メニュー設定画面でメニューの表示位置を設定するためのメニュー 得します。

各画面の操作手順については、iWP の「[グループ管理者操作ガイド](#)」を参照してください。

- ロールの作成について・・・「1.5.2 ロールの設定」
- ユーザの作成とロールの付与について・・・「1.5.3 ユーザの登録と削除(アカウント情報とプロフィール情報)」
- Webサービス実行ユーザの権限について・・・「1.14 Webサービスアクセス設定」

また、iWP の Webサービスについては詳しく知りたい場合は、iWP の「[Web サービス プログラミングガイド](#)」を参照してください。

SSOサービスプロバイダ情報の提供

SSO 連携する iAP のシステム管理者に以下の情報を提供します。

iAP のシステム管理者は、この情報を参照して、iAP の設定を行います。

iAP の設定については、「[iAP の設定](#)」を参照してください。

プロバイダID	iWP の設定ファイル「 <code>hybrid-ssso-provider-config.xml</code> 」に設定したプロバイダID
認証情報用エンドポイント	以下のフォーマットとなります。 <code>http://<HOST>:<PORT>/<CONTEXT_PATH>/services/AdmissionService</code>
外部メニュー連携用エンドポイント	以下のフォーマットとなります。 <code>http://<HOST>:<PORT>/<CONTEXT_PATH>/services/MenuService</code>
Webサービス接続先 ログイングループID	SSOサービスプロバイダのログイングループIDです。
Webサービス接続用 ユーザコード	権限を与えられたユーザのユーザコードです。
Webサービス接続用 ユーザパスワード	権限を与えられたユーザのパスワードです。
ログアウトURL	以下のフォーマットとなります。 <code>http://<HOST>:<PORT>/<CONTEXT_PATH>/user.logout</code>

iAP の設定

IM-HybridSSO モジュールの選択

IM-Juggling で、「追加機能」-「認証拡張機能」-「iAP-iWP間SSO連携モジュール(IM-HybridSSO)」を選択します。

「iAP-iWP間SSO連携モジュール(IM-HybridSSO)」を利用するには、「外部メニュー連携」モジュールが必要です。
「外部メニュー連携」は、「iAP-iWP間SSO連携モジュール(IM-HybridSSO)」に対して「依存関係も含めて選択」を行うことで自動的に選択されます。

SSOサービスプロバイダの設定

利用するSSOサービスプロバイダを設定ファイルに記述します。

「設定ファイル」タブから、「iAP-iWP間SSO連携モジュール(IM-HybridSSO)」-「SSO連携用マッピング設定(hybrid-ssso-mapping-config)」を出力します。

「SSO連携用マッピング設定(hybrid-ssso-mapping-config)」に、SSOサービスプロバイダの設定を記述します。
連携するSSOサービスプロバイダの数だけ `<mapping>` タグにSSOサービスプロバイダの情報を記述してください。
プロバイダIDには、「[SSOサービスプロバイダの設定](#)」で iWP に設定した値を記述してください。

IM-HybridSSO を構成するサーバは、全て同一のドメイン上に構築する必要があります。
以下の構成の場合、設定ファイルのサイトドメインに設定する値は「intra-mart.jp」となります。

- SSO認証プロバイダ: iap.intra-mart.jp
- SSOサービスプロバイダ1: iwp1.intra-mart.jp
- SSOサービスプロバイダ2: iwp2.intra-mart.jp

設定ファイルの詳細な設定内容については「[設定ファイルリファレンス](#)」-「[SSO連携用マッピング設定](#)」を参照してください。

メニュープロバイダの設定

利用する外部メニュー連携用のメニュープロバイダを設定ファイルに記述します。

IM-HybridSSO を利用する場合、メニュープロバイダの接続先情報には、「[SSOサービスプロバイダの設定](#)」で設定した情報と同じ iWP の接続先情報を設定します。
SSOサービスプロバイダと同じ接続先(URLのホスト)、ログイングループ、Webサービス実行ユーザを設定してください。

設定方法については、「[メニュープロバイダの設定](#)」、および、「[設定ファイルリファレンス](#)」-「[外部メニュー連携接続先設定](#)」を参照してください。

同一ホストで運用する場合の設定

iAP と iWP を同一ホストで運用する場合、一部の `Cookie` 名が重複するため、SSO を利用するためには、`Cookie` 名を変更する必要があります。
以下の設定を変更してください。

1. セッション `Cookie` 名の変更

セッションの Cookie 名は、アプリケーションサーバにより変更が可能です。
Resin の場合の設定方法は、以下の通りです。

- 設定ファイル

```
<%RESIN_HOME%/conf/resin.xml>
```

- 修正内容

```
<!--
- Resin 4.0 configuration file.
-->
<resin xmlns="http://caucho.com/ns/resin"
  xmlns:resin="urn:java:com.caucho.resin">

  :

  <cluster id="app">

    :

    <session-cookie>IAP_SESSIONID</session-cookie>

    :
  </cluster>
</resin>
```

Resin 以外の設定方法は、アプリケーションサーバのドキュメントを参照してください。

2. セッション情報チェック用 Cookie 名の変更

セッション情報の有効性をチェックするための Cookie 名を変更します。

セッション情報チェック用 Cookie 名を変更するためには、「[設定ファイルリファレンス](#)」-「[セッション情報チェック設定](#)」-「[セッション情報 Cookie 設定名](#)」を参照してください。
Cookie 名は、デフォルトでは「jp.co.intra_mart.session.cookie」が設定されていますので、この値を任意の値に変更してください。

- 設定例

```
:

<!-- セッション情報管理で使用する設定 -->
<category name="im_session">
  <!-- セッション管理情報を保存するクッキー名 -->
  <param>
    <param-name>cookie_name</param-name>
    <param-value>jp.co.intra_mart.session.cookie.iap</param-value>
  </param>
</category>
```

キャッシュ設定

認可ポリシー設定キャッシュ

intra-mart Accel Platform の認可ポリシー設定のキャッシュについて設定します。

コラム

認可ポリシー設定キャッシュ設定は [テナント管理機能](#) に含まれています。

1. 「ProjectNavigator」内の <conf/im-ehcache-config/authz-policy.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <im-ehcache-config>/<cache> の内容を運用規模に応じて変更します。

```
<cache
  name="IM_TENANT-AUTHZPOLICY"
  enable="true"
  max-bytes-memory="160M"
  time-to-live-seconds="3600"
  time-to-idle-seconds="1800"
/>
```

コラム

詳細については「[認可仕様書 認可のキャッシュ設定](#)」を参照してください。

認可リソースグループ設定キャッシュ

intra-mart Accel Platform の認可設定のリソースグループに関するキャッシュについて設定します。

コラム

認可リソースグループ設定キャッシュ設定は [テナント管理機能](#) に含まれています。

1. 「ProjectNavigator」内の <conf/im-ehcache-config/authz-resourcetype-service.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <im-ehcache-config>/<cache> の内容を運用規模に応じて変更します。

```
<cache
  name="IM_TENANT-AUTHZ_RESOURCECETYPE_GENERALSERVICE"
  enable="true"
  max-bytes-memory="128k"
  time-to-live-seconds="86400"
  time-to-idle-seconds="86400"
/>
```

コラム

詳細については「[認可仕様書 認可のキャッシュ設定](#)」を参照してください。

認可リソース閉塞情報キャッシュ

intra-mart Accel Platform の認可リソースの閉塞情報に関するキャッシュについて設定します。

コラム

認可リソース閉塞情報キャッシュ設定は テナント管理機能 に含まれています。

1. 「ProjectNavigator」内の <conf/im-ehcache-config/authz-resource-block.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <im-ehcache-config>/<cache> の内容を運用規模に応じて変更します。

```
<cache
  name="IM_TENANT-AUTHZ_RESOURCE_BLOCK"
  enable="true"
  max-bytes-memory="320k"
  time-to-live-seconds="86400"
  time-to-idle-seconds="86400"
/>
```

コラム

詳細については「[認可仕様書 認可のキャッシュ設定](#)」を参照してください。

メニュー側ルーティング設定キャッシュ

intra-mart Accel Platform のメニュー設定のルーティングに関するキャッシュについて設定します。

コラム

メニュー側ルーティング設定キャッシュ設定は テナント管理機能 に含まれています。

1. 「ProjectNavigator」内の <conf/im-ehcache-config/authz-mapped-entry-url.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <im-ehcache-config>/<cache> の内容を運用規模に応じて変更します。

```
<cache
  name="IM_TENANT-AUTHZ_MAPPED_ENTRY_URL"
  enable="true"
  max-bytes-memory="128k"
  time-to-live-seconds="86400"
  time-to-idle-seconds="86400"
/>
```

コラム

詳細については「[認可仕様書 認可のキャッシュ設定](#)」を参照してください。

認可IPv4サブジェクト設定キャッシュ

intra-mart Accel Platform の認可設定のIPv4サブジェクトに関するキャッシュについて設定します。

コラム

認可IPv4サブジェクト設定キャッシュ設定は テナント管理機能 に含まれています。

1. 「ProjectNavigator」内の <conf/im-ehcache-config/authz-subjecttype-ipv4.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <im-ehcache-config>/<cache> の内容を運用規模に応じて変更します。

```
<cache
  name="IM_TENANT-AUTHZ-SUBJECTTYPE-IPV4"
  enable="true"
  max-bytes-memory="64k"
  time-to-live-seconds="3600"
  time-to-idle-seconds="1800"
/>
```

コラム

「max-bytes-memory」及び、「max-bytes-disk」属性が設定されている場合、Cacheにオブジェクトを登録する際に、そのオブジェクトのサイズの計算処理が行われます。この際、登録するオブジェクトが、別のオブジェクトの参照を大量に持つ場合、計算処理に時間がかかりパフォーマンスの低下の原因となる可能性があります。登録するオブジェクトが1000以上の参照を持つ場合、下記のようなメッセージがログに出力されます。

The configured limit of 1,000 object references was reached while attempting to calculate the size of the object graph.
Severe performance degradation could occur if the sizing operation continues.
This can be avoided by setting the CacheManger or Cache <sizeOfPolicy> elements maxDepthExceededBehavior to "abort" or adding stop points with @IgnoreSizeOf annotation.
If performance degradation is NOT an issue at the configured limit, raise the limit value using the CacheManager or Cache <sizeOfPolicy> elements maxDepth attribute.
For more information, see the Ehcache configuration documentation.

このログが出力される場合は、キャッシュに格納するオブジェクトの構成を見直すか、「max-bytes-memory」または、「max-bytes-disk」の代わりに、「max-elements-on-memory」または「max-elements-on-disk」の利用を検討して下さい。

キャッシュするオブジェクトの単位

認可IPv4サブジェクト設定は、テナント単位でキャッシュされます。
そのため、キャッシュするオブジェクト数は1つです。

キャッシュサイズの計算式

キャッシュが行われる対象データのおおまかなサイズは以下の計算式で求めることができます。

$$\text{キャッシュサイズ} = ((1) + (2)) \times (3)$$

- (1) ... IPv4 に関するサブジェクトID のバイト数 (64byte)
- (2) ... アドレスパターン文字列 (平均 100byte)
- (3) ... 登録されている IPv4 に関するサブジェクトの数

intra-mart Accel Platform のデフォルト値は以下の計算をもとに設定されています。

$$(64 + 100) \times 400 = 65600$$

$$\Rightarrow 64KB$$

グローバルナビキャッシュ

intra-mart Accel Platform のグローバルナビキャッシュについて設定します。
グローバルナビキャッシュは intra-mart Accel Platform 標準テーマを使用した場合に利用されます。

i コラム

グローバルナビキャッシュ設定は テナント管理機能 に含まれています。

- 「ProjectNavigator」内の <conf/im-ehcache-config/menu-dropdown.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
- <im-ehcache-config>/<cache> の内容を運用規模に応じて変更します。

```
<cache
  name="IM_TENANT-MENU_DROPDOWN"
  enable="true"
  max-bytes-memory="62M"
  time-to-live-seconds="259200"
  time-to-idle-seconds="259200"
/>
```

i コラム

「max-bytes-memory」及び、「max-bytes-disk」属性が設定されている場合、Cacheにオブジェクトを登録する際に、そのオブジェクトのサイズの計算処理が行われます。
この際、登録するオブジェクトが、別のオブジェクトの参照を大量に持つ場合、計算処理に時間がかかりパフォーマンスの低下の原因となる可能性があります。
登録するオブジェクトが1000以上の参照を持つ場合、下記のようなメッセージがログに出力されます。

```
The configured limit of 1,000 object references was reached while attempting to calculate the size of the object graph.
Severe performance degradation could occur if the sizing operation continues.
This can be avoided by setting the CacheManger or Cache <sizeOfPolicy> elements maxDepthExceededBehavior to "abort" or adding stop points with @IgnoreSizeOf annotation.
If performance degradation is NOT an issue at the configured limit, raise the limit value using the CacheManager or Cache <sizeOfPolicy> elements maxDepth attribute.
For more information, see the Ehcache configuration documentation.
```

このログが出力される場合は、キャッシュに格納するオブジェクトの構成を見直すか、「max-bytes-memory」または、「max-bytes-disk」の代わりに、「max-elements-on-memory」または「max-elements-on-disk」の利用を検討して下さい。

キャッシュするオブジェクトの単位

グローバルナビキャッシュは、ユーザ単位でキャッシュされます。
そのため、キャッシュするオブジェクト数は、利用ユーザ数によって見積もります。

キャッシュサイズの計算式

キャッシュが行われる対象データのおおまかなサイズは以下の計算式で求めることができます。

$$\text{キャッシュサイズ} = ((1) \times (2) + (3) \times (4) + (5)) \times (6)$$

- (1) ... 各メニューアイテムに設定されている内容サイズ(平均600byte)
 - (1) = (1a) + (1b) + (1c) + (1d) + (1e) + (1f) + ((1g) × (1h))
 - (1a) ... メニューアイテム毎に出力されるHTMLのバイト数
 - (1b) ... メニューアイテムIDのバイト数
 - (1c) ... URLのバイト数
 - (1d) ... メニューアイテム表示名
 - (1e) ... メニューアイテム表示オプション (ポップアップ表示、iframe表示など)
 - (1f) ... 画像情報
 - (1g) ... 引数(キー+値)のバイト数
 - (1h) ... 引数の数
- (2) ... 表示されるメニューアイテムの数

- (3) ... 各メニューフォルダに設定されている内容サイズ(平均150byte)
- (3) = (3a) + (3b) + (3c)
- (3a) ... メニューフォルダ毎に出力されるHTMLのバイト数
- (3b) ... メニューフォルダ表示名
- (3c) ... 画像情報
- (4) ... 表示されるメニューフォルダの数
- (5) ... グローバルナビが常に出力するHTML(700byte)
- (6) ... 利用ユーザ数

intra-mart Accel Platform のデフォルト値は以下の計算をもとに設定されています。

$$(600 \times 50 + 150 \times 10 + 700) \times 2000 = 64400000$$

$$\Rightarrow 62MB$$

ヘルプドロップダウンキャッシュ

intra-mart Accel Platform のヘルプドロップダウンのキャッシュについて設定します。

コラム

ヘルプドロップダウンキャッシュ設定は テナント管理機能 に含まれています。

1. 「ProjectNavigator」内の <conf/im-ehcache-config/help-dropdown.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <im-ehcache-config>/<cache> の内容を運用規模に応じて変更します。

```
<cache
  name="IM_TENANT-MENU_SITE_HELP_DROPDOWN"
  enable="true"
  max-bytes-memory="3700K"
  time-to-live-seconds="259200"
  time-to-idle-seconds="259200"
/>
```

コラム

「max-bytes-memory」及び、「max-bytes-disk」属性が設定されている場合、Cacheにオブジェクトを登録する際に、そのオブジェクトのサイズの計算処理が行われます。この際、登録するオブジェクトが、別のオブジェクトの参照を大量に持つ場合、計算処理に時間がかかりパフォーマンスの低下の原因となる可能性があります。登録するオブジェクトが1000以上の参照を持つ場合、下記のようなメッセージがログに出力されます。

The configured limit of 1,000 object references was reached while attempting to calculate the size of the object graph.
Severe performance degradation could occur if the sizing operation continues.
This can be avoided by setting the CacheManger or Cache <sizeOfPolicy> elements maxDepthExceededBehavior to "abort" or adding stop points with @IgnoreSizeOf annotation
If performance degradation is NOT an issue at the configured limit, raise the limit value using the CacheManager or Cache <sizeOfPolicy> elements maxDepth attribute.
For more information, see the Ehcache configuration documentation.

このログが出力される場合は、キャッシュに格納するオブジェクトの構成を見直すか、「max-bytes-memory」または、「max-bytes-disk」の代わりに、「max-elements-on-memory」または「max-elements-on-disk」の利用を検討して下さい。

キャッシュするオブジェクトの単位

ヘルプドロップダウンは、ユーザ単位でキャッシュされます。
そのため、キャッシュするオブジェクト数は、利用ユーザ数によって見積もります。

キャッシュサイズの計算式

キャッシュが行われる対象データのおおまかなサイズは以下の計算式で求めることができます。

$$\text{キャッシュサイズ} = ((1) \times (2)) \times (3)$$

- (1) ... 各メニューアイテムに設定されている内容サイズ(平均630byte)
- (1) = (1a) + (1b) + (1c)
- (1a) ... メニューアイテムIDのバイト数
- (1b) ... URLのバイト数
- (1c) ... メニューアイテム表示名
- (2) ... 表示されるメニューアイテムの数
- (3) ... 利用ユーザ数

intra-mart Accel Platform のデフォルト値は以下の計算をもとに設定されています。

$$(630 \times 3) \times 2000 = 3780000$$

$$\Rightarrow 3,700KB$$

個人設定メニューキャッシュ

intra-mart Accel Platform の個人設定メニューのキャッシュについて設定します。

コラム

個人設定メニューキャッシュ設定は テナント管理機能 に含まれています。

1. 「ProjectNavigator」内の <conf/im-ehcache-config/menu-personal-settings-item.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <im-ehcache-config>/<cache> の内容を運用規模に応じて変更します。

```
<cache
  name="IM_TENANT-MENU_PERSONAL_SETTINGS_ITEM"
  enable="true"
  max-bytes-memory="11M"
  time-to-live-seconds="259200"
  time-to-idle-seconds="259200"
/>
```

i コラム

「max-bytes-memory」及び、「max-bytes-disk」属性が設定されている場合、Cacheにオブジェクトを登録する際に、そのオブジェクトのサイズの計算処理が行われます。この際、登録するオブジェクトが、別のオブジェクトの参照を大量に持つ場合、計算処理に時間がかかりパフォーマンスの低下の原因となる可能性があります。登録するオブジェクトが1000以上の参照を持つ場合、下記のようなメッセージがログに出力されます。

The configured limit of 1,000 object references was reached while attempting to calculate the size of the object graph.
Severe performance degradation could occur if the sizing operation continues.
This can be avoided by setting the CacheManger or Cache <sizeOfPolicy> elements maxDepthExceededBehavior to "abort" or adding stop points with @IgnoreSizeOf annotation
If performance degradation is NOT an issue at the configured limit, raise the limit value using the CacheManager or Cache <sizeOfPolicy> elements maxDepth attribute.
For more information, see the Ehcache configuration documentation.

このログが出力される場合は、キャッシュに格納するオブジェクトの構成を見直すか、「max-bytes-memory」または、「max-bytes-disk」の代わりに、「max-elements-on-memory」または「max-elements-on-disk」の利用を検討して下さい。

キャッシュするオブジェクトの単位

個人設定メニューは、ユーザ単位でキャッシュされます。
そのため、キャッシュするオブジェクト数は、利用ユーザ数によって見積もります。

キャッシュサイズの計算式

キャッシュが行われる対象データのおおまかなサイズは以下の計算式で求めることができます。

キャッシュサイズ = (1) × (2)

- (1) ... 個人設定メニューのサイズ(平均5500byte)
- (2) ... 利用ユーザ数

intra-mart Accel Platform のデフォルト値は以下の計算をもとに設定されています。

5500 × 2000 = 11000000
⇒ 11MB

ポータルへの認可サブジェクト情報キャッシュ

intra-mart Accel Platform のポータルの認可サブジェクト情報のキャッシュについて設定します。

i コラム

ポータルの認可サブジェクト情報キャッシュ設定は ポータル機能 に含まれています。

1. 「ProjectNavigator」内の <conf/im-ehcache-config/im_portal-subjecttype.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. <im-ehcache-config>/<cache> の内容を運用規模に応じて変更します。

```
<cache
  name="IM_PORTAL-SUBJECTTYPE"
  enable="true"
  max-elements-on-memory="200"
  time-to-live-seconds="3600"
  time-to-idle-seconds="1800"
/>
```

i コラム

「max-bytes-memory」及び、「max-bytes-disk」属性が設定されている場合、Cacheにオブジェクトを登録する際に、そのオブジェクトのサイズの計算処理が行われます。この際、登録するオブジェクトが、別のオブジェクトの参照を大量に持つ場合、計算処理に時間がかかりパフォーマンスの低下の原因となる可能性があります。登録するオブジェクトが1000以上の参照を持つ場合、下記のようなメッセージがログに出力されます。

The configured limit of 1,000 object references was reached while attempting to calculate the size of the object graph.
Severe performance degradation could occur if the sizing operation continues.
This can be avoided by setting the CacheManger or Cache <sizeOfPolicy> elements maxDepthExceededBehavior to "abort" or adding stop points with @IgnoreSizeOf annotation
If performance degradation is NOT an issue at the configured limit, raise the limit value using the CacheManager or Cache <sizeOfPolicy> elements maxDepth attribute.
For more information, see the Ehcache configuration documentation.

このログが出力される場合は、キャッシュに格納するオブジェクトの構成を見直すか、「max-bytes-memory」または、「max-bytes-disk」の代わりに、「max-elements-on-memory」または「max-elements-on-disk」の利用を検討して下さい。

キャッシュするオブジェクトの単位

ポータルへの認可サブジェクト情報は、HTTPセッション単位でキャッシュされます。
そのため、キャッシュするオブジェクト数は、同時アクセスの最大HTTPセッション数によって見積もります。

キャッシュサイズの計算式

キャッシュが行われる対象データのおおまかなサイズは以下の計算式で求めることができます。

キャッシュサイズ(オブジェクトの数) = (1)

(1) ... HTTPセッション数

intra-mart Accel Platform のデフォルト値は以下の計算をもとに設定されています。

同時アクセスの最大HTTPセッション数を200と仮定しています。

機能制御の設定

ファイルのアップロードを制限する方法

intra-mart Accel Platform を稼働させるために以下の設定ファイルの編集を行います。

リクエスト制御設定



コラム

リクエスト制御設定は Webモジュール機能 に含まれています。

1. 「ProjectNavigator」内の <(プロジェクト名)/conf/request-control-config.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
2. 「request-control-config.xml」の設定内容については「[設定ファイルリファレンス](#)」を参照してください。

SAStrutsを利用している場合

SAStruts でファイルのアップロードを行うには、「struts-config.xml」の設定が必要です。

- struts-config.xmlファイルの編集を行います。
1. 設定ファイルを「ProjectNavigator」上に追加します。
 - 次の方法があります。
 - ・ IM-Juggling プロジェクトを作成するウィザード中の追加リソースの配置より、「SAStruts用設定ファイル」を追加します。
 - ・ IM-Juggling プロジェクト作成後に「追加リソースの選択」より「SAStruts用設定ファイル」を追加します。
 2. 「ProjectNavigator」内の <(プロジェクト名)/struts-config.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。
 3. 「ソース」タブをクリックし、Controller Configurationが定義されている行に移動します。::

```
<!-- ===== Controller Configuration -->

<controller
  maxFileSize="1024K"
  bufferSize="1024"
  processorClass="org.seasar.struts.action.S2RequestProcessor"
  multipartClass="org.seasar.struts.upload.S2MultipartRequestHandler"/>
```

4. 「maxFileSize」属性に定義されている値を変更します。



コラム

「ProjectNavigator」内のツリー上に設定ファイルがない場合

1. <(プロジェクト名)/juggling.im> ファイルをダブルクリックします。
2. 「設定ファイル」タブをクリックします。
3. 対象の設定ファイルを選択し、右側にある「出力」をクリックします。
4. 「ProjectNavigator」内のツリー上に表示されたファイルをダブルクリックして編集を行います。



コラム

「WARファイルによる複数テナント」環境の場合、[WARファイルによる複数テナント](#) を参考にしてください。



コラム

各アプリケーションで選択したモジュール内の設定ファイルは、各アプリケーションのドキュメントを参照してください。

WARファイルの出力

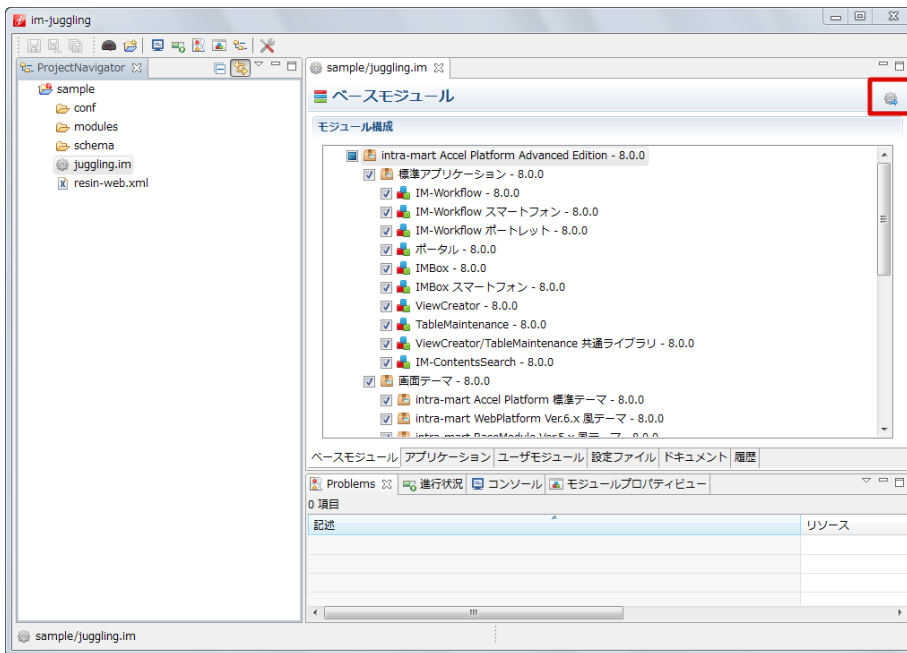
i コラム

WARファイルとは？

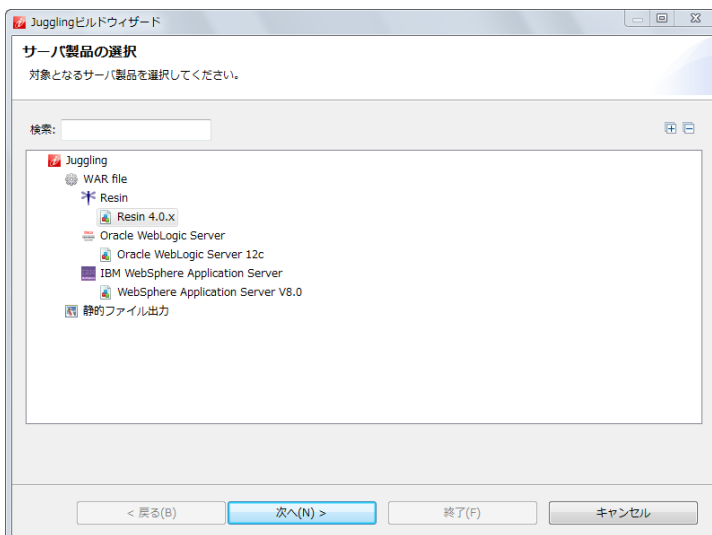
JavaEEで定義されている Web Application Archive ファイルです。
 intra-mart Accel Platform の動作に必要な各種ファイルが格納され、
 intra-mart Accel Platform 用のWARファイルの作成は IM-Juggling にて行います。
 IM-Juggling にて作成されたWARファイルを Resin などの Web Application Server に
 配置(デプロイ)して、Web Application Server を起動することにより、
 WARファイルに格納された intra-mart Accel Platform 自体や intra-mart Accel Platform に対応した
 各種アプリケーションが起動され、利用することができます。

このことから intra-mart Accel Platform の製品自体の最小単位になるため、
 intra-mart Accel Platform のライセンスはWARファイル単位でカウントすることになります。

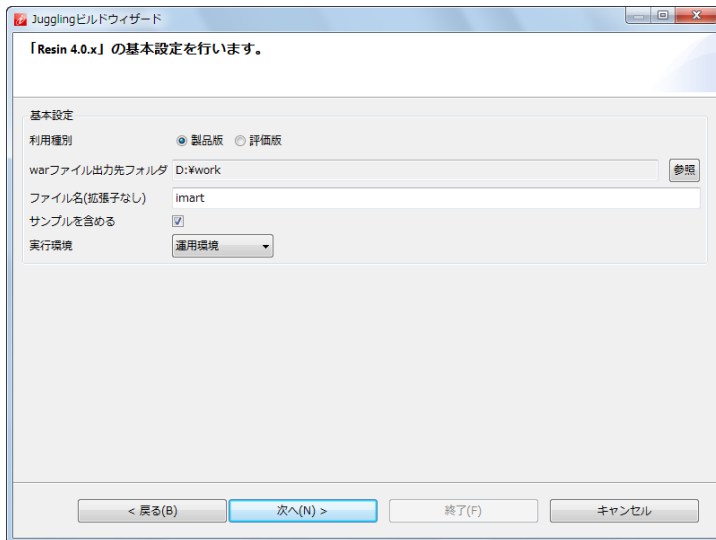
1. <juggling.im> ファイルを開き、右上にある「ビルドウィザード」アイコンをクリックします。



2. 「サーバ製品を選択」画面で対象となるサーバを選択し、「次へ」をクリックします。



3. 「基本設定」画面で設定を行い、「次へ」をクリックします。



利用種別

- 製品版
ライセンスを購入されたお客様はこちらを選択してください。
- 評価版
評価版となっております。
正規に使用する場合はライセンスを購入してください。

サンプルデータを含める

チェックを入れた場合、サンプルデータを投入します。

実行環境

- intra-mart Accel Platform を稼働する環境に応じて、実行環境を選択します。
詳細は、「[WARファイル作成時の実行環境の変更](#)」を参照してください。



注意

「WARファイル出力先フォルダ」には、実行するユーザの権限があるフォルダを指定してください。

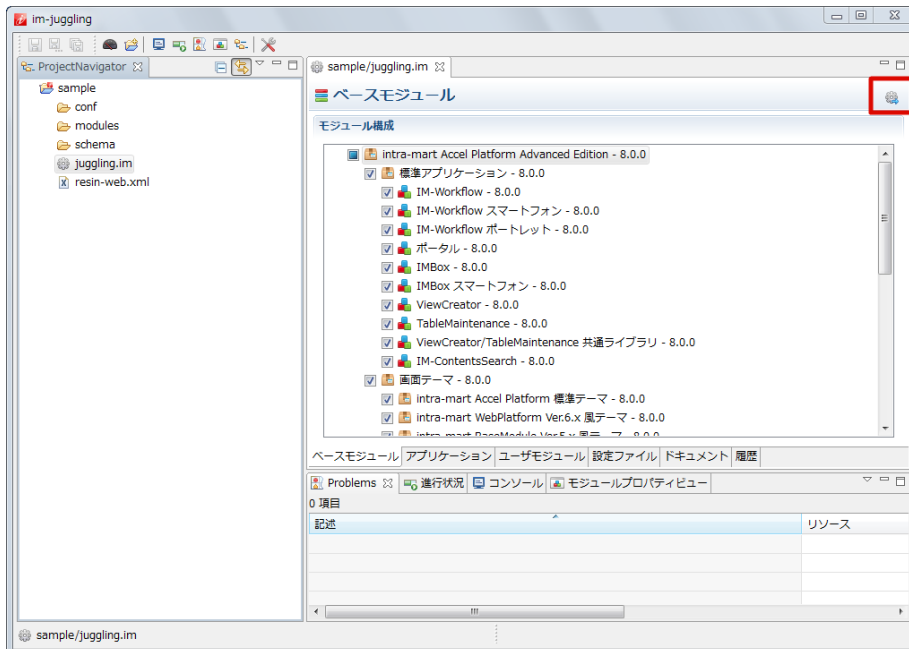
4. 「ライセンスのレビュー」画面で内容を確認し、同意頂ける場合は「使用条件の条項に同意します。」を選択し、「次へ」をクリックします。
5. 「設定項目の確認」画面で内容を確認し、「終了」をクリックします。
Jugglingビルドウィザード画面が立ち上がり、ダウンロードが開始されます。
ダウンロードに時間がかかりますので、完了するまで操作を行わないでください。



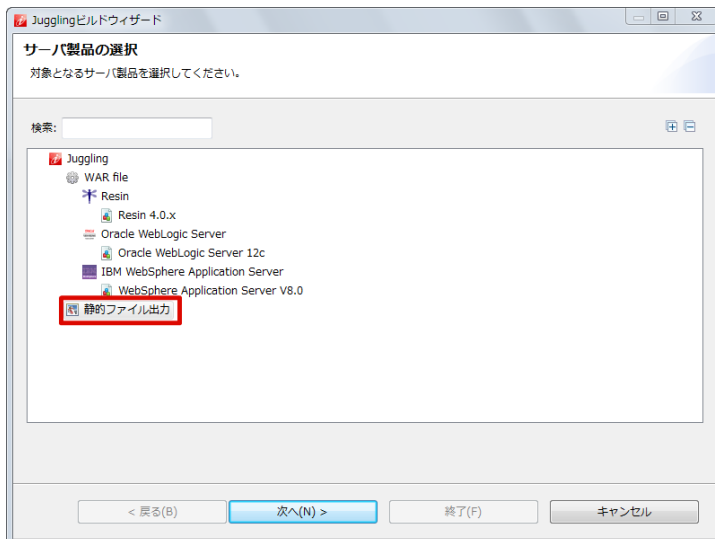
6. WARファイルが指定の場所に出力されると、ビルドウィザード画面が閉じます。
指定したディレクトリにWARファイルが出力されます。

静的ファイルの出力

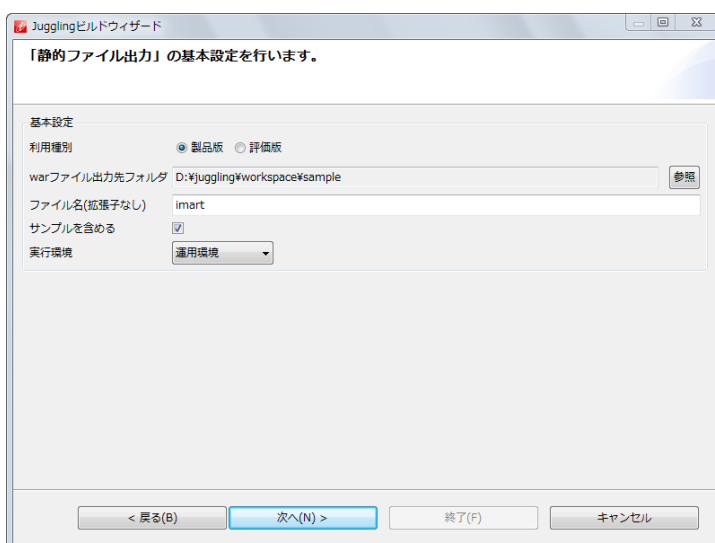
1. <juggling.im> ファイルを開き、右上にある「ビルドウィザード」アイコンをクリックします。



2. 「サーバ製品の選択」画面で「静的ファイル」を選択し、「次へ」をクリックします。



3. 「基本設定」画面で設定を行い、「次へ」をクリックします。



利用種別

- 製品版
ライセンスを購入されたお客様はこちらを選択してください。
- 評価版
評価版となっております。
正規に使用する場合はライセンスを購入してください。

サンプルデータを含める

チェックを入れた場合、サンプルデータを投入します。

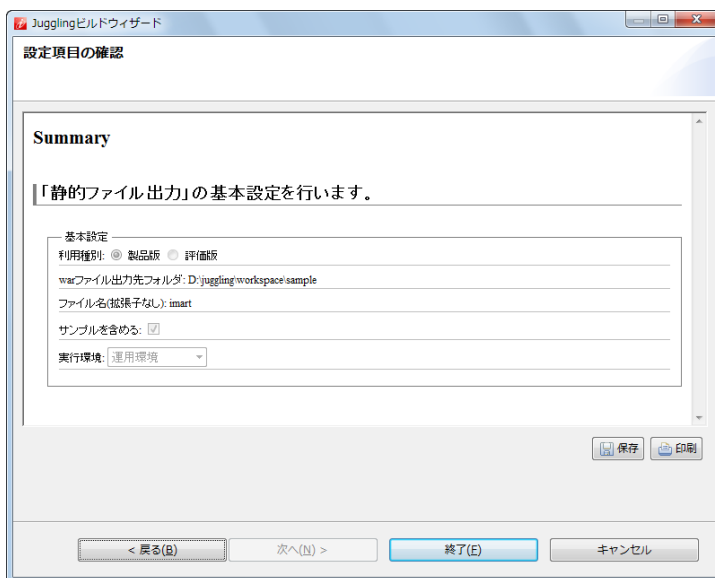
実行環境

- 単体テスト環境
単体テスト環境を表します。
JUnitなどを利用する場合を想定しています。
パフォーマンスに関しては考慮しません。
- 結合テスト環境
結合テスト環境を表します。
IDEを使用したテスト環境などを想定しています。
一般的に最も利用される環境を想定しています。
必要最低限の動作環境です。
- 統合テスト環境
統合テスト環境を表します。
本来の実行環境 (APサーバ) を使用した統合テスト環境を想定しています。
カットオーバー直後等でもこの環境を利用することを想定しています。
- 運用環境
運用環境を表します。
パフォーマンスを考慮した動作環境です。

 注意

「warファイル出力先フォルダ」には、実行するユーザの権限があるフォルダを指定してください。

4. 「ライセンスのレビュー」画面で内容を確認し、同意頂ける場合は「使用条件の条項に同意します。」を選択し、「終了」をクリックします。
5. 「設定項目の確認」画面で内容を確認し、「終了」をクリックします。
Jugglingビルドウィザード画面が立ち上がり、ダウンロードが開始されます。
ダウンロードに時間がかかりますので、完了するまで操作を行わないでください。



6. zipファイルが指定の場所に出力されると、ビルドウィザード画面が閉じます。
7. 出力されたzipファイルを、Web Server の仮想ディレクトリで設定したディレクトリにzipファイルを展開して配置します。

Web Application Server の起動・停止

Windows

コンソール起動・停止

項目

- Resin の起動
 - コンソールから起動する
 - 複数のIPアドレスが設定されている場合
- Resin の停止
 - コンソールから停止する

Resin の起動

 注意

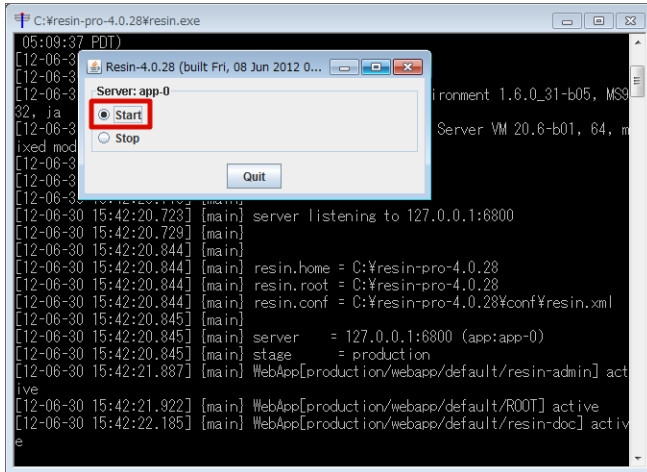
Apache Cassandra をご利用の場合は、Resin の起動前に、Apache Cassandra を起動する必要があります。

 注意

Resin を起動するためには「.NET Framework 3.5」が必要です。
Windows Server 2012 環境では、「.NET Framework 4.5」のみインストールされているため、

「.NET Framework 3.5」のセットアップを行ってください。
詳細は、[.NET Framework のセットアップ](#)を参照してください。

1. <%RESIN_HOME%\resin.exe> ファイルをダブルクリックします。
起動停止画面とコンソールが表示されます。



2. コンソールに以下のメッセージが表示されたら起動は完了です。

```
[12-07-02 18:30:00.000] {main} http listening to *:8080
[12-07-02 18:30:00.000] {main}
[12-07-02 18:30:00.000] {main} Resin[id=app-0] started in 55480ms
```

コンソールから起動する

resin.exeはコンソールから実行することが可能です。

フォアグラウンドで起動する場合、引数に「console」を指定します。

```
> cd <%RESIN_HOME%>
> resin.exe console
```

バックグラウンドで起動する場合、引数に「start」を指定します。

```
> cd <%RESIN_HOME%>
> resin.exe start
```

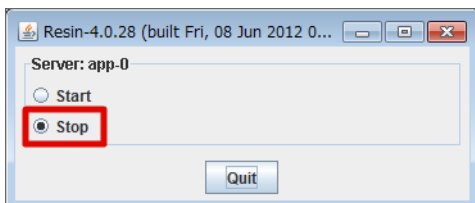
複数のIPアドレスが設定されている場合

起動する環境に複数のIPアドレスが設定されている場合、resin.exeをダブルクリックではなく、コンソールより「jgroups.bind_addr」プロパティを指定して起動する必要があります。

```
> cd <%RESIN_HOME%>
> resin.exe console -Djgroups.bind_addr=192.168.1.1
```

Resin の停止

1. 起動時に立ち上がった起動停止画面にて「Stop」をクリックします。



2. コンソールに以下のメッセージが表示されたら停止した状態となります。

```
{resin-shutdown} Shutdown Resin reason: OK
```

コンソールから停止する

コンソールからバックグラウンドで起動している場合、停止するには引数に「stop」を指定します。

これは、引数「console」により起動した場合、またはresin.exeをダブルクリックで起動した場合には利用できません。

```
> cd <%RESIN_HOME%>
> resin.exe stop
```


項目

- Windowsサービスへの登録
 - 分散構成の場合
 - 複数のIPアドレスが設定されている場合
- Windowsサービスの削除

Windowsサービスへの登録

- Resin をインストールした直下にある「setup.exe」を起動します。

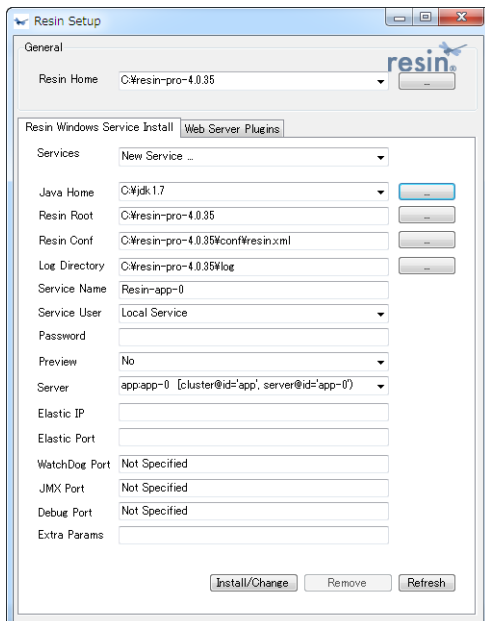
```
%RESIN_HOME%/setup.exe
```



注意

setup.exe を実行する場合は、.NET Framework 3.5 が必要です。

- 起動すると次の画面が表示されます。



- 「Resin Windows Service Install」タブ内の項目を編集します。

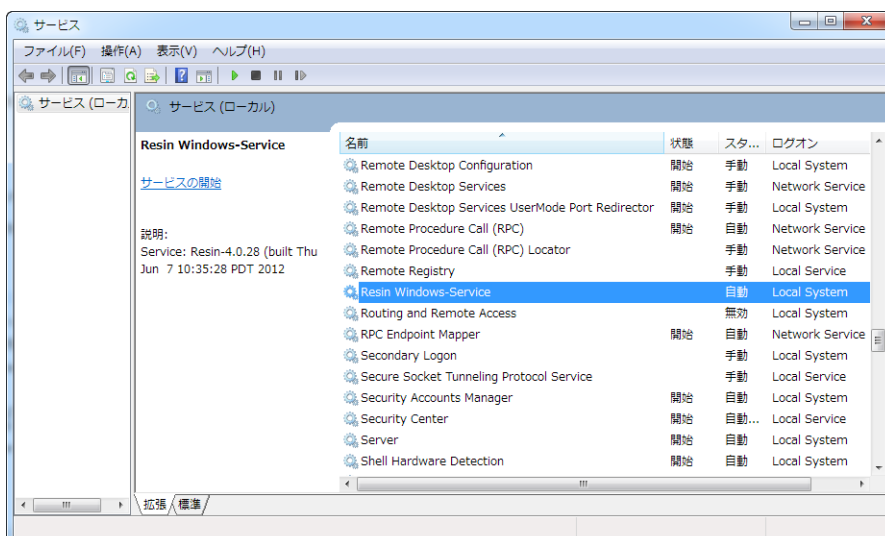
i コラム

本書ではサービス名を変更します。

変更前: 「Resin-app-0」

変更後: 「Resin Winwdows-Service」

- 下部にある「Install/Change」ボタンをクリックします。
- 次の画面が表示されたらWindowsサービスへの登録は完了です。必要に応じて、スタートアップの種類を「自動」等に変更してください。



分散構成の場合

- 分散構成により複数の Resin を使用している場合、それぞれの Resin に対してWindowsサービスへの登録を行う必要があります。「Server」項目に、各サーバに応じて自身のIDを指定してください。

- 「Service User」項目に、共有ディレクトリに対するアクセス権を持つユーザを指定してください。
Storageの設定については、「[Storage](#)」を参照してください。

コラム

同一のOS上に複数の Resin を起動する場合、watchdogのポートが重複しないように設定する必要があります。
「WatchDog Port」項目に、ポート番号を指定してください。
ポート番号を指定しない場合、「6600」が適用されます。

watchdogについては、「[設定ファイルリファレンス - watchdog](#)」を参照してください。

複数のIPアドレスが設定されている場合

起動する環境に複数のIPアドレスが設定されている場合、「Extra Params」項目に、「jgroups.bind_addr」プロパティを指定する必要があります。

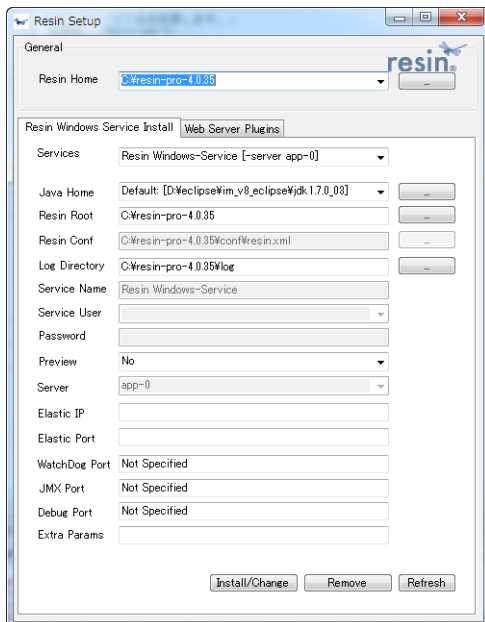
```
-Djgroups.bind_addr=192.168.1.1
```

Windowsサービスの削除

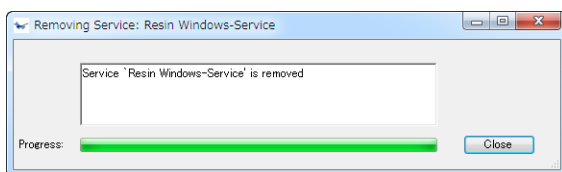
- Resin をインストールした直下にある「setup.exe」を起動します。

```
%RESIN_HOME%/setup.exe
```

- 起動すると次の画面が表示されます。



- 下部にある「Remove」ボタンをクリックします。
- 次の画面が表示されたらWindowsサービスへの削除は完了です。



Linux

コンソール起動・停止

項目

- Resin の起動
 - 複数のIPアドレスが設定されている場合
- Resin の停止

Resin の起動



注意

Apache Cassandra をご利用の場合は、Resin の起動前に、Apache Cassandra を起動してください。

1. 以下のコマンドを実行します。

```
# <%RESIN_HOME%/bin/resinctl -server app-0 start
```



注意

分散構成の場合、各 Resin がインストールされている環境より起動してください。

「app-0」の部分は <%RESIN_HOME%/conf/resin.properties> ファイルの「app_servers」プロパティに設定したIPアドレスに対応するIDを指定してください。

2. コンソールに以下のメッセージが表示されたら起動は完了です。

```
[12-07-02 18:30:00.000] {main} http listening to *:8080
[12-07-02 18:30:00.000] {main}
[12-07-02 18:30:00.000] {main} Resin[id=app-0] started in 55480ms
```

複数のIPアドレスが設定されている場合

起動する環境に複数のIPアドレスが設定されている場合、「jgroups.bind_addr」システムプロパティを指定して起動する必要があります。

```
# <%RESIN_HOME%/bin/resinctl -server app-0 start -Djgroups.bind_addr=192.168.1.1
```

Resin の停止

1. 以下のコマンドを実行します。

```
# %RESIN_HOME%/bin/resinctl -server app-0 stop
```

2. コンソールに以下のメッセージが表示されたら停止した状態となります。

```
{resin-shutdown} Shutdown Resin reason: OK
```

Linuxデーモンへの登録、削除

項目

- Linuxデーモンへの登録
 - 複数のIPアドレスが設定されている場合
- Linuxデーモンからの削除
- 実行ユーザの変更

Linuxデーモンへの登録



注意

intra-mart Accel Platform (Linux編)のインストールと設定 で説明している方法でインストールされている必要があります。
また、インストール時にroot権限を持つユーザにて実行されている必要があります。

- chkconfig コマンドを実行して、起動時に自動起動するようにしてください。

```
# chkconfig --add resin
```

- chkconfig コマンドを実行して、以下のように表示されれば成功です。

```
# chkconfig --list resin
resin    0:off 1:off 2:off 3:on  4:on  5:on  6:off
```

複数のIPアドレスが設定されている場合

起動する環境に複数のIPアドレスが設定されている場合、/etc/init.dディレクトリに配置されているresinスクリプトを編集し、システムプロパティ「jgroups.bind_addr」を指定する必要があります
25行目前後に、環境変数が指定されている部分が存在します。そこに、環境変数「ARGS」を追加し、システムプロパティを追加します。

```
JAVA_HOME="/usr/lib/jvm/java-7-sun"
RESIN_HOME="/usr/local/resin"
CONSOLE="/var/log/resin/console.log"
# システムプロパティ追加
ARGS="-Djgroups.bind_addr=192.168.1.1"
```

Linuxデーモンからの削除

- chkconfig コマンドを実行して、削除してください。

```
# chkconfig --del resin
```

- chkconfig コマンドを実行して、以下のように表示されれば成功です。

```
# chkconfig --list resin
サービス resin は chkconfig をサポートしますが実行レベルで参照されていません (run 'chkconfig --add resin')
```

- 必要に応じて、/etc/init.d/resin を削除してください。

実行ユーザの変更

/etc/init.dディレクトリに配置されているresinスクリプトを編集することにより、Linuxデーモンとして起動するResinの実行ユーザを変更することが可能です。
25行目前後に、環境変数が指定されている部分が存在します。そこに、環境変数「USER」を追加し、Resinの実行ユーザを指定します。
実行ユーザは事前に作成されている必要があります。

```
JAVA_HOME="/usr/lib/jvm/java-7-sun"
RESIN_HOME="/usr/local/resin"
CONSOLE="/var/log/resin/console.log"
# 実行ユーザ追加
USER=imart
```

WARファイルのデプロイ

- IM-Juggling で作成したWARファイルを任意のディレクトリに配置します。
- 以下のコマンドを実行してください。
Windowsの場合 (コマンド・プロンプトを起動)

```
> cd <%RESIN_HOME%>
> resin deploy 任意のディレクトリパス/WARファイル名
```

Linuxの場合

```
# <%RESIN_HOME%>/bin/resinctl deploy 任意のディレクトリパス/warファイル名
```



コラム

マルチテナントをご利用の場合は、必要なWARファイルに対してそれぞれ実行してください。

注意

WARファイルのデプロイには、Resin が起動している必要があります。

テナント環境セットアップ

- Webブラウザを利用して、テナント環境セットアップを行います。
Webブラウザより 以下のURLへアクセスします。

システム管理者ログイン画面: `http://<HOST>:<PORT>/<CONTEXT_PATH>/system/login`

※ Web Application Server をローカル環境にセットアップ、かつ、本書の例に合わせてセットアップした場合、次のアドレスで接続できます。

システム管理者ログイン画面: <http://localhost:8080/imart/system/login>

コラム

コンテキストパスはデプロイ時に設定したものを指定します。

Resin の場合、デプロイ時のWARファイル名となります。

[WARファイルのデプロイ](#) を参照してください。

Resin 以外の Web Application Server については、デプロイ時に指定した「コンテキスト・ルート」となります。

- intra-mart Accel Platform の テナント環境セットアップ 手順は次の通りです。

テナント設定

Step 1 Step 2 Step 3 Step 4 Step 5 Step 6 Step 7 Step 8 Step 9

Step 1 - はじめに

① ようこそ、intra-mart Accel Platformへ。

ここでは、Accel Platformを運用するために必要な設定を行います。
この画面は、タブキーまたは画面下部の[次へ]ボタンにより次のページへ遷移します。
必要な項目に関する詳細はセットアップガイドを御覧ください。

http://www.intra-mart.jp/download/product/iap/setup/iap_setup_guide/texts/tenant/index.html

次へ

各ウィザードの詳細については下記を参照してください。

システム管理者情報

1. システム全体の管理者 を設定します。

テナント設定

Step 1 Step 2 Step 3 Step 4 Step 5 Step 6 Step 7 Step 8 Step 9

Step 2 - システム管理者情報

ユーザコード *

パスワード

パスワード(確認)

ロケール *

日本語 ▼

次へ

項目	必須/任意	説明
ユーザコード	必須	システム管理者のユーザコードを入力します。 (例:system)
パスワード	任意	システム管理者のパスワードを入力します。
パスワード(確認)	任意	システム管理者のパスワードを入力します。
ロケール	必須	システム管理者のロケールを選択します。 初期表示されているロケールは、アクセスしているブラウザのロケール設定です。



コラム

“任意”項目は、必要に応じて入力してください。必要がなければ入力の必要はありません。

テナント情報

1. テナントの基本的な情報 を設定します。

テナント設定

Step 1 Step 2 Step 3 Step 4 Step 5 Step 6 Step 7 Step 8 Step 9

Step 3 - テナント情報

テナントID *	
デフォルトロケール *	日本語 ▼
タイムゾーン *	(GMT+09:00) 日本 / 東京 ▼
アカウントライセンス数 *	☒ 無制限

次へ

項目	必須/任意	説明
テナントID	必須	テナントのIDを入力します。 「 DataSourceマッピングの設定 」の <tenant-id> で設定した値を入力します。 DataSourceマッピングの設定を行っていない場合は、「 default 」を入力します。
デフォルトロケール	必須	テナントのデフォルトロケールを選択します。 初期表示されているロケールは、アクセスしているブラウザのロケール設定です。 このロケールは運用中に変更することは推奨していません。 運用に応じたロケールを設定してください。
タイムゾーン	必須	テナントのタイムゾーンを選択します。
アカウントライセンス数	必須	テナントのアカウントライセンス数を入力します。 テナント管理者を登録するため「1」以上を入力してください。



注意

TRY版利用などにおいてサンプルデータセットアップを行う場合は、「アカウントライセンス数」は **無制限** を選択する事を推奨します。

テナント環境情報

1. テナントの環境情報を設定します。

コラム

「Resinデータソース設定」モジュールを適用している場合、リソース参照名にセレクトボックスが表示されます。モジュールが適用されていない場合はテキスト入力になります。

テナント設定

Step 1 Step 2 Step 3 Step 4 Step 5 Step 6 Step 7 Step 8 Step 9

Step 4 - テナント環境情報

リソース参照名

ストレージパス

ベースURL

グローバルナビ最大表示数

次へ

項目	必須/任意	説明
リソース参照名	任意	リソース参照名を選択します。
ストレージパス	任意	ストレージパスを入力します。
ベースURL	任意	ベースURLを入力します。
グローバルナビ最大表示数	任意	グローバルナビの最大表示件数を入力します。

コラム

ストレージパスに storage-config.xml の <storage-directory-name>を付加したパスがパブリックストレージパス %PUBLIC_STORAGE_PATH% になります。
 (例)
 ストレージパスに /var/imart と入力し
 設定ファイル storage-config.xml の <storage-directory-name> に storage
 と設定した場合、
 %PUBLIC_STORAGE_PATH% は /var/imart/storage となります。

コラム

未指定の場合は、それぞれの設定ファイルの内容が有効になります。

テナント管理者情報

1. テナントの管理者 を設定します。

テナント設定

Step 1 Step 2 Step 3 Step 4 Step 5 Step 6 Step 7 Step 8 Step 9

Step 5 - テナント管理者情報

ユーザコード *	
パスワード	
パスワード(確認)	

次へ

項目	必須/任意	説明
ユーザコード	必須	テナント管理者のユーザコードを入力します。 (例:tenant)
パスワード	任意	テナント管理者のパスワードを入力します。
パスワード(確認)	任意	テナント管理者のパスワードを再入力します。

コラム

“任意”項目は、必要に応じて入力してください。必要がなければ入力の必要はありません。

コラム

テナント環境セットアップ中にエラーが発生してセットアップが中断してしまった場合、再度セットアップを実施してもテナント管理者は登録されません。このように正常にテナント管理者を登録できなかった場合、テナント管理画面から改めてテナント管理者を登録することができます。詳しくは、「システム管理者操作ガイド」-「テナント管理」の「テナント管理者を新規に作成する」の項を参照してください。

LDAP連携・設定

注意

IM-Jugglingにおいて、**LDAP認証モジュール** を選択した場合のみ、この画面が表示されます。

1. LDAP連携・設定 情報を設定します。

テナント設定

Step 1 Step 2 Step 3 Step 4 Step 5 **Step 6** Step 7 Step 8 Step 9

Step 6 - LDAP連携・設定

設定内容 *

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ldap-certification-config xmlns="http://intra-mart.co.jp/system/security/certification/provider/ldap">
  <enable>false</enable>
  <load-balancing>false</load-balancing>
  <attempt-on-failed-authentication>true</attempt-on-failed-authentication>
  <log>false</log>
  <ldap-servers>
    <ldap-server>
      <permit-no-password>true</permit-no-password>
      <provider-url>ldap://localhost:389/</provider-url>
      <context-factory>com.sun.jndi.ldap.LdapCtxFactory</context-factory>
      <base-dn>dc=ldaps,dc=intra,dc=intra-mart,dc=jp</base-dn>
      <search-filter>sAMAccountName=?</search-filter>
      <search-controls>
        <connect-timeout-property-name>com.sun.jndi.ldap.connect.timeout</connect-timeout-property-name>
        <connect-timeout>0</connect-timeout>
        <searching-dn>sAMAccountName=admin,cn=User,dc=ldaps,dc=intra,dc=intra-mart,dc=jp</searching-dn>
        <searching-pw>*****</searching-pw>
        <count-limit>0</count-limit>
        <time-limit>0</time-limit>
      </search-controls>
    </ldap-server>
  </ldap-servers>
</ldap-certification-config>
```

次へ

コラム

初期表示は IM-Juggling のプロジェクト/conf/ldap-certification-config.xmlの内容が表示されます。ただし、<enable>タグの値はfalseとなっています。

注意

<enable>タグの内容がtrueである場合のみLDAP認証が有効となります。LDAP認証を有効とする場合、認証先であるLDAPの設定を正しく行ってください。

項目	必須/任意	説明
設定内容	必須	LDAP認証の有効/無効、および、認証先であるLDAPの情報を入力します。入力内容については、「 LDAP認証設定ファイル 」の説明を参照してください。

ログインセッション管理

 注意

IM-Jugglingにおいて、**ログインセッション管理モジュール** を選択した場合のみ、この画面が表示されます。

1. 一般ユーザのログイン時に二重ログインを検出した場合の動作を設定します。



ログインセッション管理の設定内容は以下のとおりです。

- 標準の認証エラーページを表示する

標準の認証エラーページを表示します。

- 二重ログインの検出を表示する

二重ログインを検出したことを一般ユーザに通知します。

一般ユーザは通知された画面からログインを再試行することができます。

- ログインユーザによるセッションの無効化を許可する

二重ログインを検出したことを一般ユーザに通知します。

一般ユーザは通知された画面からログイン中のセッションを強制的に無効化しログインをすることができます。

Apache Cassandra接続情報

 注意

IM-Jugglingにおいて、**IMBoxモジュール** を選択した場合のみ、この画面が表示されます。

1. Apache Cassandra接続情報を入力します。


 注意

Apache Cassandra接続情報はIMBox利用時のみ表示されます。

テナント環境セットアップを実行する前にApache Cassandraの設定、起動が行われている必要があります。
Apache Cassandraの設定に関する詳細は「IMBox Cassandra管理者ガイド」を参照してください。

- [IMBox Cassandra管理者ガイド \(日本語\)](#)
- [IMBox Cassandra管理者ガイド \(English\)](#)
- [IMBox Cassandra管理者ガイド \(中文\)](#)

項目	必須/任意	説明
クラスタ名	必須	Cassandraサーバのクラスタ名を入力します。 (例: IMBox Cluster)
キースペース	必須	Cassandraサーバのキースペースを入力します。 (例: default)
接続先	必須	Cassandraが稼働しているサーバのIPアドレスとポート番号を入力します。 接続先は「IPアドレス」または、「IPアドレス:ポート番号」の形式で入力します。(ポート番号を省略、9160を利用します。) 9160以外のポート番号を指定した場合、新規ノードの検出機能にてエラーが発生します。 分散構成で複数のCassandraが稼働している場合、すべての接続先を1行ずつ入力してください。 (例: 127.0.0.1:9160)
レプリケーションファクタ	必須	クラスタ内部のデータのレプリカ数を入力します。 レプリケーションファクタは、キースペース作成時のみ使用されます。 (例: 1)
認証情報設定	任意	Cassandraへの接続における認証の利用を選択します。 認証情報を設定する場合には書き込み権限の確認を行うため、事前にキースペースを作成してあります。
認証ユーザ名	認証設定利用時のみ必須	Cassandraへの認証接続における接続ユーザ名を入力します。 認証設定利用時のみ表示されます。 (例: admin)
認証パスワード	認証設定利用時のみ必須	Cassandraへの認証接続におけるパスワードを入力します。 認証設定利用時のみ表示されます。 (例: admin)
テスト接続		入力した内容でCassandraが接続可能であるかのテストが行えます。 テナント作成時には、必ず行うことを推奨します。

コラム

“任意”項目は、必要に応じて入力してください。必要がなければ入力の必要はありません。

コラム

Cassandra接続情報の登録時の初期値は、Cassandraサーバ接続設定 (cassandra-config.xml) の設定値となります。
cassandra-config.xmlに関する詳細は、[Cassandraサーバ接続設定 \(cassandra-config.xml\)](#) を参照してください。

Apache Solr接続情報

注意

- IM-Jugglingにおいて、**IM-ContentsSearchモジュール** を選択した場合のみ、この画面が表示されます。
- IM-ContentsSearch for Accel Platformを利用する場合は、Apache Solrのセットアップが必要となります。
Apache Solrのセットアップに関する詳細は「[Apache Solr](#)」の「セットアップ」を参照してください。

1. Apache Solr接続情報を入力します。

テナント設定

Step 1

Step 2

Step 3

Step 4

Step 5

Step 6

Step 7

Step 8

Step 9

Step 8 - Solr接続情報

このタイミングで設定せずに、テナント環境セットアップ実行後にテナント管理画面で設定することも可能です。

Solr接続情報を設定する

グループID

標準接続先

default

default

次へ

項目	必須/任意	説明
Solr接続情報を設定する	任意	テナント環境セットアップ時にApache Solr接続情報を設定するかどうかをこのボタンで切り替えることが「Solr接続情報」ボタンを未選択状態にすることで、Apache Solr接続情報を設定せずにテナント環境セットアップを行うことができます。
グループID	固定("default")	標準接続先のグループIDは"default"から変更することはできません。
標準接続先	必須(「Solr接続情報」ボタンが有効になっている場合のみ)	接続先を選択します。 「Solr接続情報」ボタンが選択状態になっている場合のみ選択することができます。 他のテナントの標準接続先に設定されている接続先は表示されません。

Solr接続情報の設定方法

テナント環境セットアップ時に、システムデータベースへ登録するSolr接続情報を設定する方法は以下の通りです。

- 設定ファイルから変更する方法

初回のテナント環境セットアップ前にSolrサーバ接続設定ファイル(solr-config.xml)を設定しておくことで、solr-config.xmlの設定(<group name="default">)で定義した設定値がSolr接続情報としてシステムデータベースへ登録されます。

Solrサーバ接続設定ファイル(solr-config.xml)は初回のテナント環境セットアップ時、または、Solr接続設定が1件も登録されていない場合に標準接続先の項目として登録されます。

solr-config.xmlに関する詳細は、「[Solrサーバ接続設定\(solr-config.xml\)](#)」を参照してください。

- 画面から変更する方法

別のテナントを新規作成するときは「Solr接続設定」画面で登録したSolr接続情報が標準接続先の選択項目として表示されます。

システム管理者メニューのテナント管理画面で、作成したテナントにApache Solr接続情報を設定することができます。

「Solr接続設定」画面に関する詳細は「[システム管理者操作ガイド](#)」-「[Solr接続設定](#)」を参照してください。

登録

1. 各ステップの設定が完了したら、「登録」をクリックします。



注意

セットアップでエラーが発生した場合、「[セットアップで困ったら・・・](#)」を参照してください。

テナント設定

Step 1 Step 2 Step 3 Step 4 Step 5 Step 6 Step 7 Step 8

Step 9

Step 9 - 登録

テナント環境セットアップを行います。
デプロイされたモジュールの数やサーバの性能によっては処理に時間がかかる場合があります。

登録

セットアップが完了すると、下記の画面が表示されます。

セットアップ結果

テナント (default) を作成し、作業テナントに設定しました。

処理結果	モジュールID	インポート種別	インポート対象名	エラーメッセージ
✓	im_javamail	EXTENSION	jp.co.intra_mart.system.mail.template.migration.MailTemplateMigrator	-
✓	im_tenant	DDL	products/import/basicim_tenant/im_tenant_common/im_tenant_common-ddl.sql	-
✓	im_tenant	DDL	products/import/basicim_tenant/im_calendar/im_calendar-ddl.sql	-
✓	im_tenant	DDL	products/import/basicim_tenant/im_admin/im_admin-ddl.sql	-
✓	im_tenant	DDL	products/import/basicim_tenant/im_auth/im_auth-ddl.sql	-
✓	im_tenant	DDL	products/import/basicim_tenant/im_menu/im_menu-ddl.sql	-
✓	im_tenant	DDL	products/import/basicim_tenant/im_password_history/im_password_history-ddl.sql	-
✓	im_tenant	DML	products/import/basicim_tenant/im_calendar/im_calendar-role.xml	-
✓	im_tenant	DML	products/import/basicim_tenant/im_calendar/im_calendar-role_en.xml	-
✓	im_tenant	DML	products/import/basicim_tenant/im_calendar/im_calendar-role_ja.xml	-
✓	im_tenant	DML	products/import/basicim_tenant/im_calendar/im_calendar-role_zh_CN.xml	-
✓	im_tenant	DML	products/import/basicim_tenant/im_admin/im_admin-role.xml	-
✓	im_tenant	DML	products/import/basicim_tenant/im_admin/im_admin-role_en.xml	-
✓	im_tenant	DML	products/import/basicim_tenant/im_admin/im_admin-role_ja.xml	-
✓	im_tenant	DML	products/import/basicim_tenant/im_admin/im_admin-role_zh_CN.xml	-



注意

「プロジェクトの作成とモジュールの選択」で選択したモジュールによりウィザードのステップ数が異なります。



コラム

旧アップデート版における **テナント環境セットアップ** については、
[intra-mart Accel Platform 2013 Winter までのテナント環境セットアップ](#) を参照してください。

ライセンスの登録

intra-mart Accel Platform を購入されている場合、ライセンスの登録を行ってください。

intra-mart Accel Platform で利用するアプリケーション製品を購入されている場合も、同様にライセンスの登録を行ってください。

1. システム管理者のメニュー画面を表示します。
メニューから「ライセンス管理」をクリックします。
2. 「ライセンスキー登録」からライセンスの登録を行ってください。



コラム

詳細は「[システム管理者操作ガイド](#)」-「[システム環境構築](#)」-「[ライセンス管理](#)」を参照してください。

アップデート・パッチの適用・モジュール構成の変更

- IM-Juggling を利用して最新モジュールの適用や、モジュール構成を変更する事ができます。

アップデート

- アップデートとは、一定の期間内でリリースする機能追加、不具合修正を含めた形で提供するリリース形態です。この他に弊社から提供している、Resin の更新、システム要件 (サポート環境・検証済み環境) の改変も含まれます。

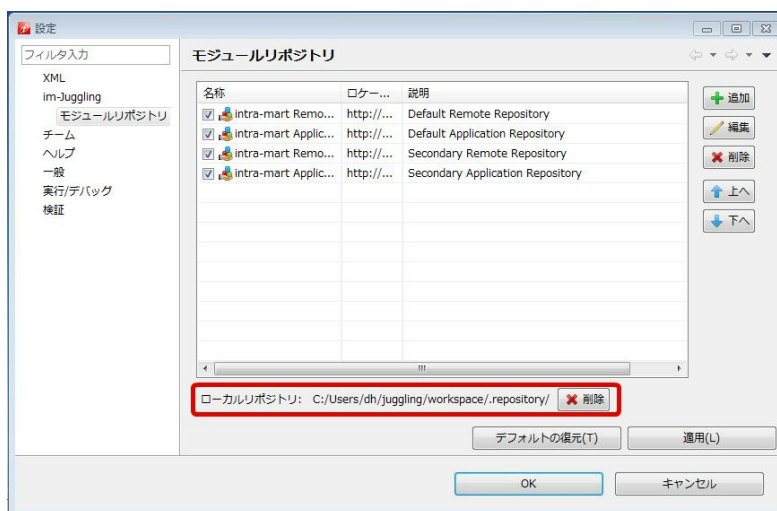
モジュールのアップデート

項目

- 古いリポジトリ情報の削除
- アップデート対象の選択とWARファイルの作成
- 設定ファイルの編集 (アップデートによる設定項目のメンテナンス)
- WARファイルの出力
- WARファイルのアンデプロイ
- Web Application Server の更新
- WARファイルの再デプロイ
- 静的ファイルの出力と再配置
- テナント環境セットアップ

古いリポジトリ情報の削除

- 前回使用した古いリポジトリ情報を削除します。
 - IM-Juggling を起動し、ウィンドウ内-ツールバー右端にある「設定」-「IM-Juggling」-「モジュールリポジトリ」を開きます。
 - 下部にある「ローカルリポジトリ」の削除をクリックしてください。



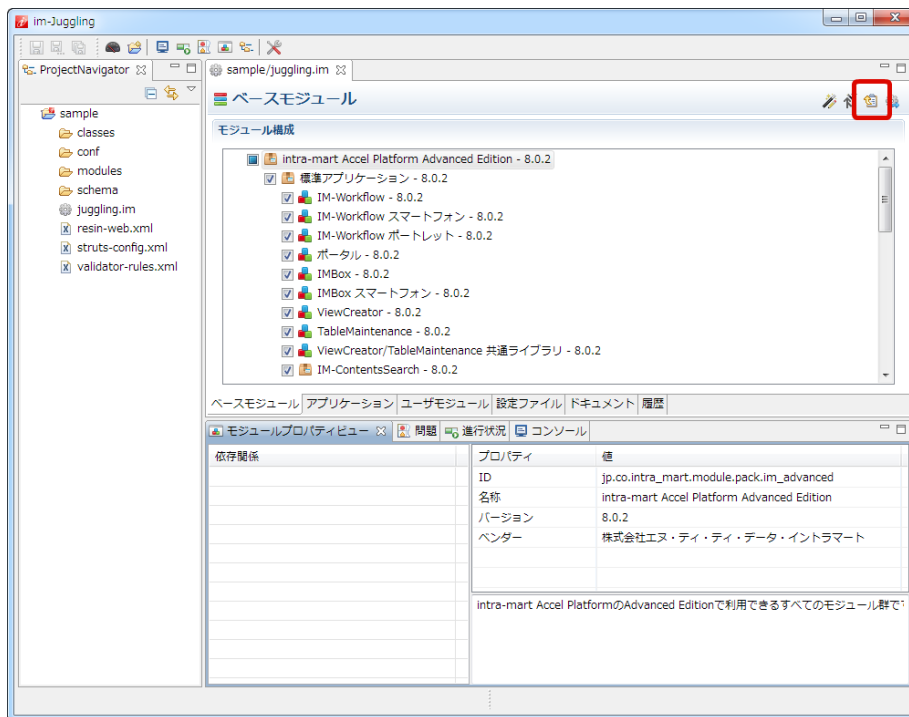
注意

ファイルの削除処理が行われている間は IM-Juggling の操作が行えない状態となります。

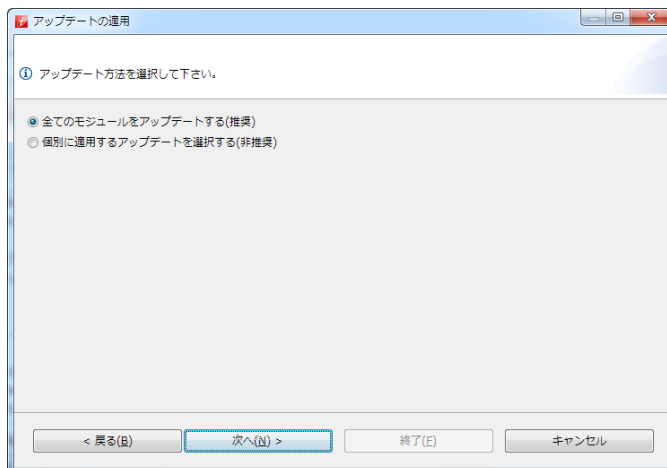
- 削除が完了したらOKをクリックします。

アップデート対象の選択とWARファイルの作成

- 前述の「古いリポジトリ情報の削除」を実施している事が前提です。
 - 画面右上の「アップデート」アイコンをクリックします。



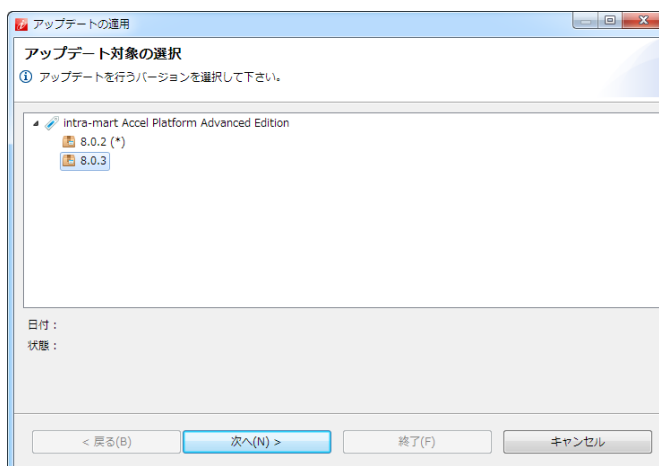
2. アップデート方法を選択します。
通常は「全てのモジュールをアップデートする(推奨)」を選択してください。
選択後、「次へ(N)」をクリックします。



注意

「個別に適用するアップデートを選択する(非推奨)」を選択した場合は、今後のアップデートが正常に行えない可能性があります。

3. ベースモジュールのアップデート対象を選択します。
選択後、「次へ(N)」をクリックします。



注意

既存モジュールのバージョンがすべて同一の場合(上図では「8.0.2」を指します)、同一バージョン(「8.0.2(*)」)に対して適用できるモジュールは存在しません。

同一バージョン(「8.0.2(*)」)を選択しても、一覧には表示されません。

！ 注意

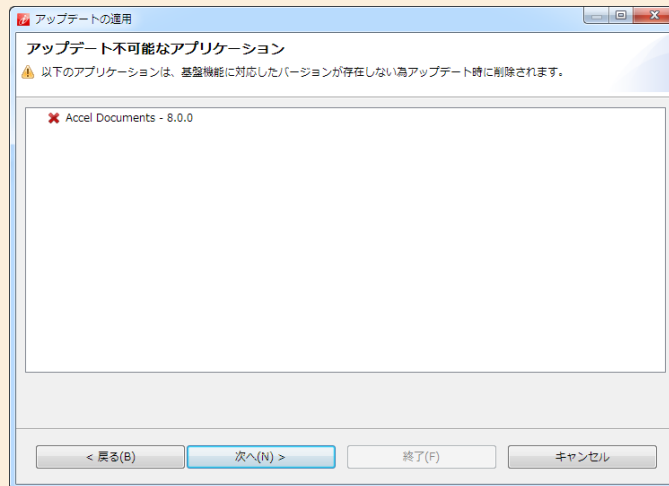
ベースモジュールのアップデート内容によっては、既存のアプリケーションが利用できない場合があります。

例: 既存のプロジェクトにおいて次の構成だった場合

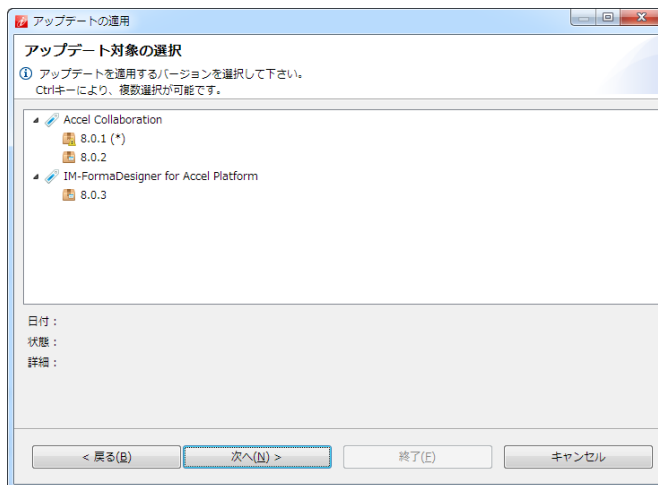
ベースモジュール: 8.0.2

アプリケーション (Accel Documents) : 8.0.0

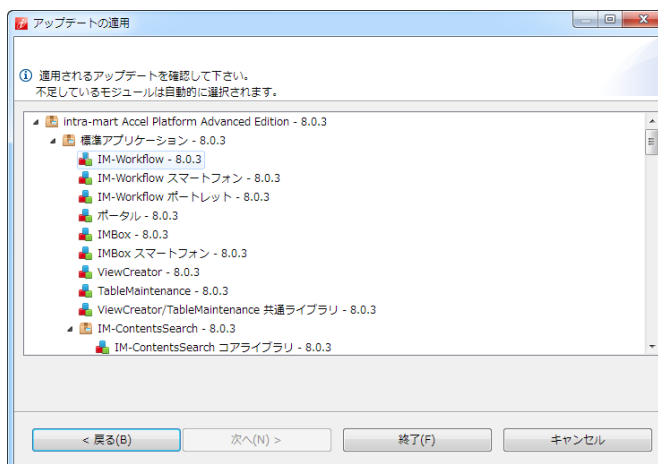
ベースモジュールのアップデートで「8.0.3」を選択すると、
アプリケーション (Accel Documents) は、ベースモジュール「8.0.3」には対応していないため、
下記の警告画面が表示されます。



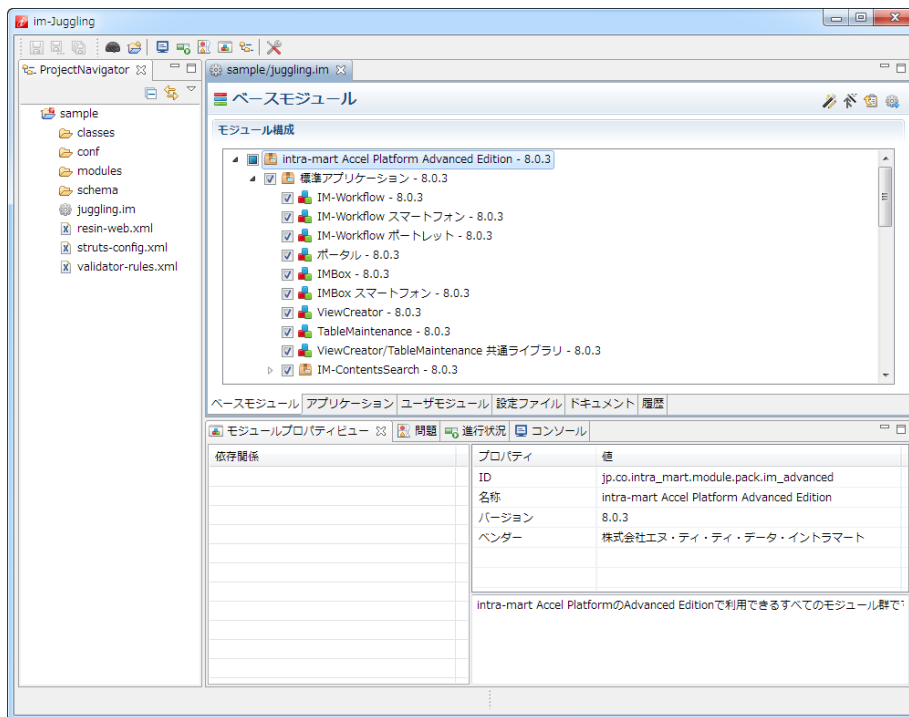
4. アプリケーションのアップデート対象を選択します。
ベースモジュールのアップデート対象で選択したバージョンに基づくアプリケーションとそのバージョンが選択できます。
選択後、「次へ(N)」をクリックします。



5. アップデート内容を確認し「終了(E)」をクリックします。



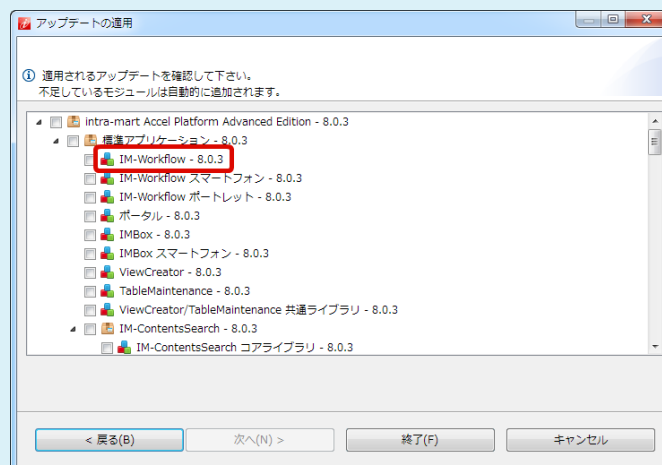
アップデート情報を取得し、既存プロジェクトのモジュール情報が変更されます。



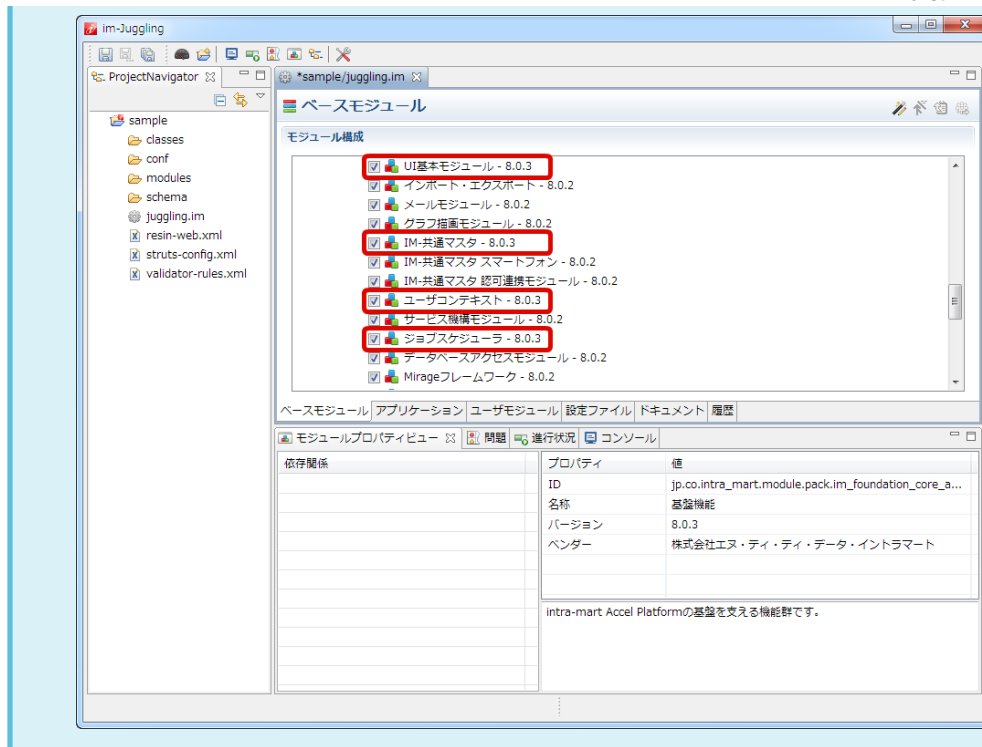
コラム

「個別に適用するアップデートを選択する(非推奨)」を選択した場合、既存プロジェクトのモジュール情報は、選択したモジュールと、そのモジュールが依存するモジュールのバージョンが更新されます。

例: 「IM-Workflow」モジュールのみを選択します。



アップデート適用後のモジュール構成では、「IM-Workflow」と「IM-Workflow」が依存する「UI基本モジュール」、「IM-共通マスタ」、「ユーザコンテキスト」、「ジョブスケジューラ」などのモジュールも自動的に更新されます。



アップデート対象がプロジェクトに反映されます。

設定ファイルの編集(アップデートによる設定項目のメンテナンス)

- アップデート版を適用前に環境を構築している場合、個別作業が必要となります。
詳細は、[設定ファイルの編集\(アップデートによる設定項目のメンテナンス\)](#)を参照してください。

コラム

※ intra-mart Accel Platform 以外のアプリケーションについては、各セットアップガイドを参照してください。

WARファイルの出力

- 前述の IM-Juggling 上の設定が完了したら [WARファイルの出力](#) を行います。

WARファイルのアンデプロイ

- 現在稼働している環境のアンデプロイを行います。
Resin の場合は、[WARファイルのアンデプロイ](#)を参照してください。
Resin 以外の Web Application Server については、各製品ベンダから提供されているドキュメントを参照してください。

Web Application Server の更新

- アップデートではシステム要件の改変により、Web Application Server の更新が必要となる場合があります。
Resin の場合は次の通りです。

コラム

最新版の Resin は製品メディアイメージ/Products/tools/iAP/Resinディレクトリもしくは、
次のサイトよりダウンロードして利用する事ができます。
<http://www.intra-mart.jp/download/product/iap/index.html>
各種ダウンロード(zipファイル)
※ダウンロードには保守契約時に配布されたIDとパスワードが必要となります。

- 既存の Resin 環境を停止します。

コラム

必要に応じて、Windowsサービスの解除、Linuxデーモンの解除が必要です。
詳細は、[分散構成の場合](#)や [複数のIPアドレスが設定されている場合](#)を参照してください。

コラム

既存の Resin 環境は以後利用する事はないため、必要に応じて退避等を行ってください。

- アップデート版で提供される最新版の Resin をインストールします。
詳細は、[Web Application Server \(Resin \)](#)を参照してください。

コラム

必要に応じて、Windowsサービスの登録、Linuxデーモンの登録が必要です。
詳細は、[Windowsサービスへの登録](#) や [Linuxデーモンへの登録](#) を参照してください。

3. 既存の Resin 環境で設定していた内容を新しい Resin 環境へ反映します。
例: %RESIN_HOME%/conf/resin.propertiesファイル内のJVMの設定等



注意

Resin 以外の Web Application Server においてシステム要件の変更が発生した場合、各製品の更新方法については、各製品ベンダから提供されているドキュメントを参照してください。データベース製品についても同様です。

WARファイルの再デプロイ

1. アップデート版を反映したWARファイルにてデプロイを行います。
Resin は、[WARファイルのデプロイ](#)を参照してください。
WebSphere Application Server 8.5.5 は、[セットアップガイド for WebSphere](#) を参照してください。
Oracle WebLogic Server 12c (12.1.3) は、[セットアップガイド for WebLogic](#) を参照してください。

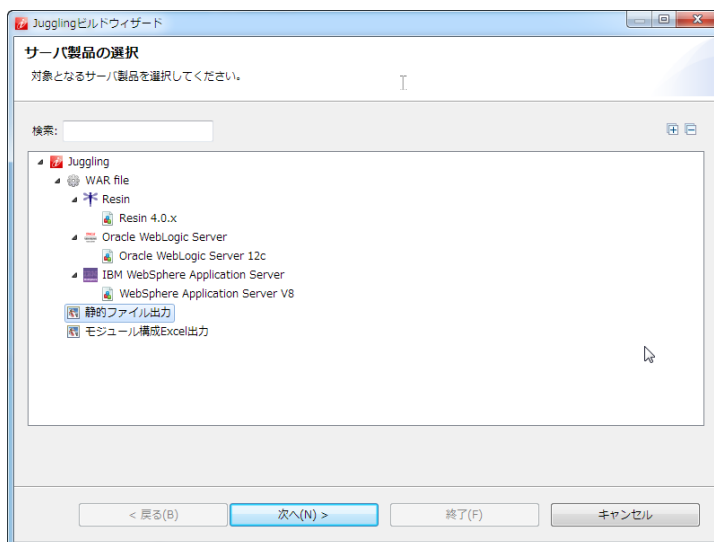


注意

再デプロイを行ったが、追加したモジュール（資材）がデプロイ先に反映されないといった事象が発生した場合は、[「Resin でWARファイルの再デプロイが正常にできない場合」](#)を参照してください。

静的ファイルの出力と再配置

1. Web Server を利用している場合、IM-Juggling より静的ファイルを出力します。



2. 出力した静的ファイルは、Web Server でエイリアスとして設定しているディレクトリ配下に配置します。
※既存のファイルとは差し替えてください。

テナント環境セットアップ

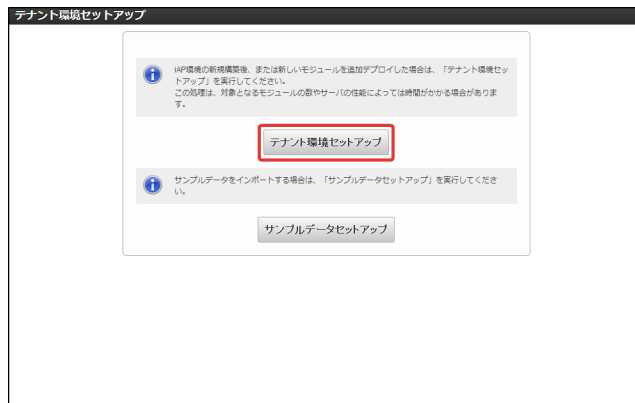
- Web Application Server を起動後、テナント環境セットアップを実施します。
システム管理者のメニュー画面を表示します。
メニューから「テナント環境セットアップ」をクリックします。



下図のように、「テナント環境は最新です。セットアップが必要なモジュールはありません。」という旨のメッセージが表示されていれば、テナント環境セットアップは正常に完了しています。



下図のように、「テナント環境セットアップ」ボタンが表示されている場合、テナント環境セットアップは未完了です。「テナント環境セットアップ」ボタンをクリックしセットアップを再度実施します。



注意

Web Server を経由してテナント環境セットアップを行う場合、Web Server のタイムアウト設置値を変更する、もしくは、Web Application Server 経由でセットアップを実施する事を推奨します。

コラム

サンプルをセットアップするボタンは処理結果に関わらず、常に表示される仕様となります。このため、サンプルデータ投入中にエラーが発生した場合、再度セットアップを実施するとデータベース で一意制約違反が発生します。この場合は、「[アンインストール](#)」を行い、改めてセットアップからやり直す事を推奨します。

- IMBoxをご利用のお客様は、既知の不具合が改修された Apache Cassandra へのバージョンアップを実施することを強く推奨しています。
 - 「[Cassandra管理者ガイド](#)」-「[Cassandraのバージョンアップ](#)」

パッチ

- パッチとは、アップデートとアップデートの間に発生した緊急度が高い修正等をモジュール単位で提供するリリース形態です。このパッチで提供される内容(不具合を修正した内容)は、今後リリース予定のアップデート版に含まれます。

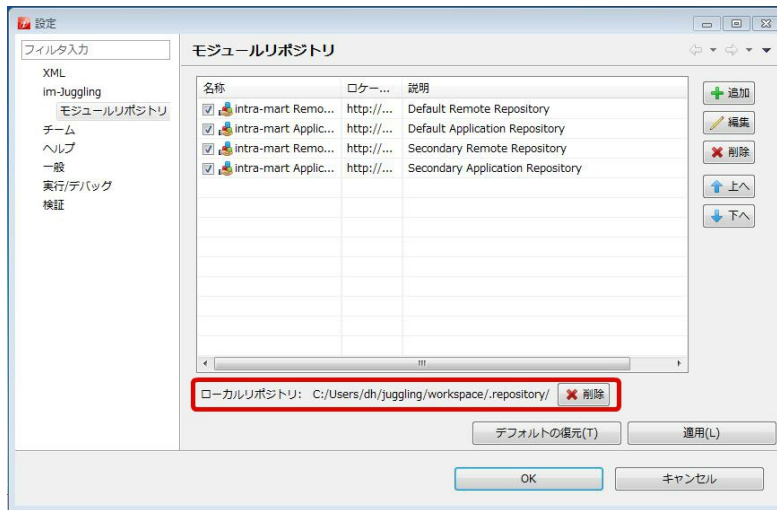
モジュールのパッチ適用

項目

- 古いリポトリ情報の削除
- パッチを適用するモジュールの選択とWARファイルの作成
- WARファイルのアンデプロイ
- WARファイルの再デプロイ
- 静的ファイルの出力と再配置
- テナント環境セットアップ

古いリポトリ情報の削除

- 前回使用した古いリポトリ情報を削除します。
 - IM-Juggling を起動し、ウィンドウ内-ツールバー右端にある「設定」-「IM-Juggling」-「モジュールリポトリ」を開きます。
 - 下部にある「ローカルリポトリ」の削除をクリックしてください。



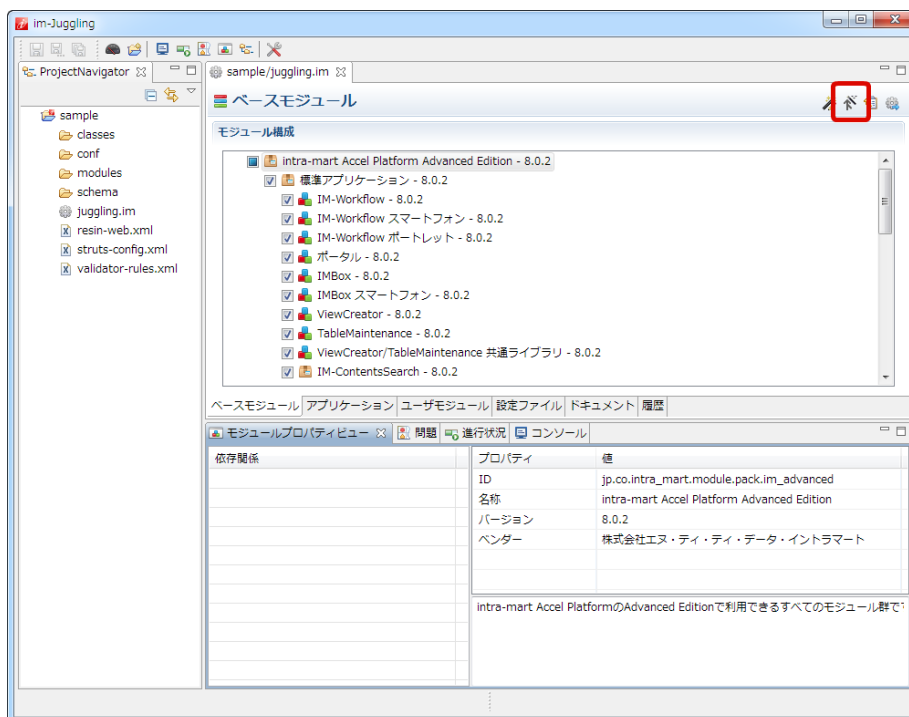
注意

ファイルの削除処理が行われている間は IM-Juggling の操作が行えない状態となります。

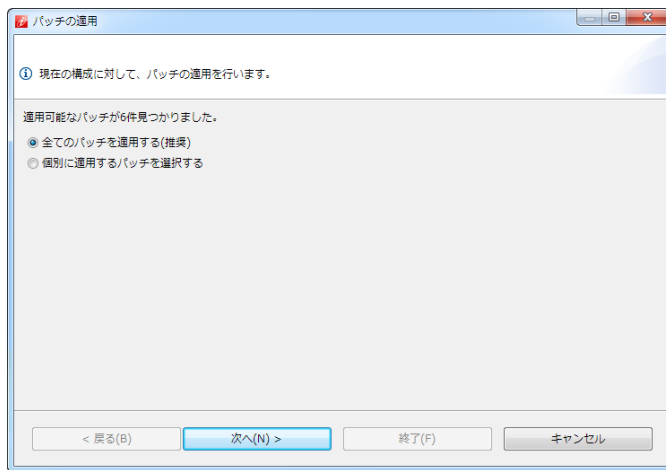
3. 削除が完了したらOKをクリックします。

パッチを適用するモジュールの選択とWARファイルの作成

- 前述の「古いリポジトリ情報の削除」を実施している事が前提です。
1. 画面右上の「パッチ」アイコンをクリックします。



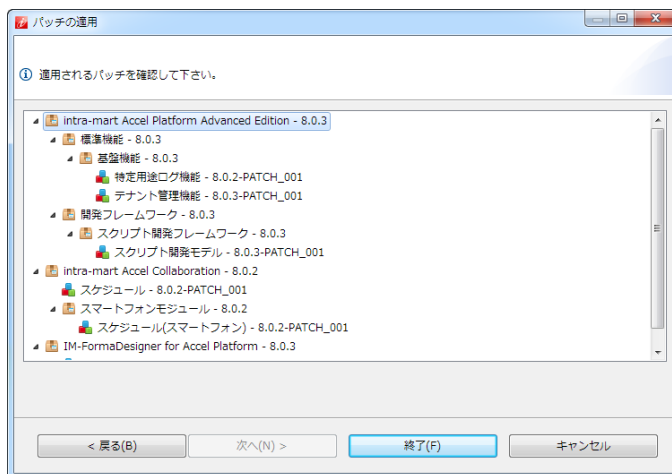
2. パッチ適用方法を選択します。
「全てのパッチを適用する(推奨)」
「個別に適用するパッチを選択する」
のどちらかを選択してください。選択後、「次へ(N)」をクリックします。



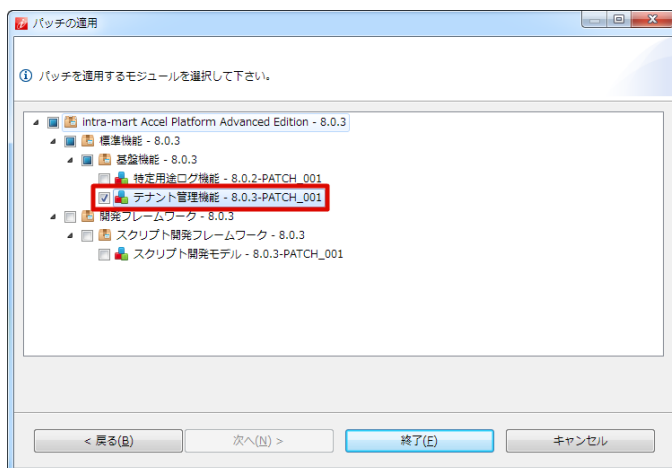
コラム

パッチとして提供される内容(不具合を修正した内容)は、今後リリース予定のアップデート版に含まれます。

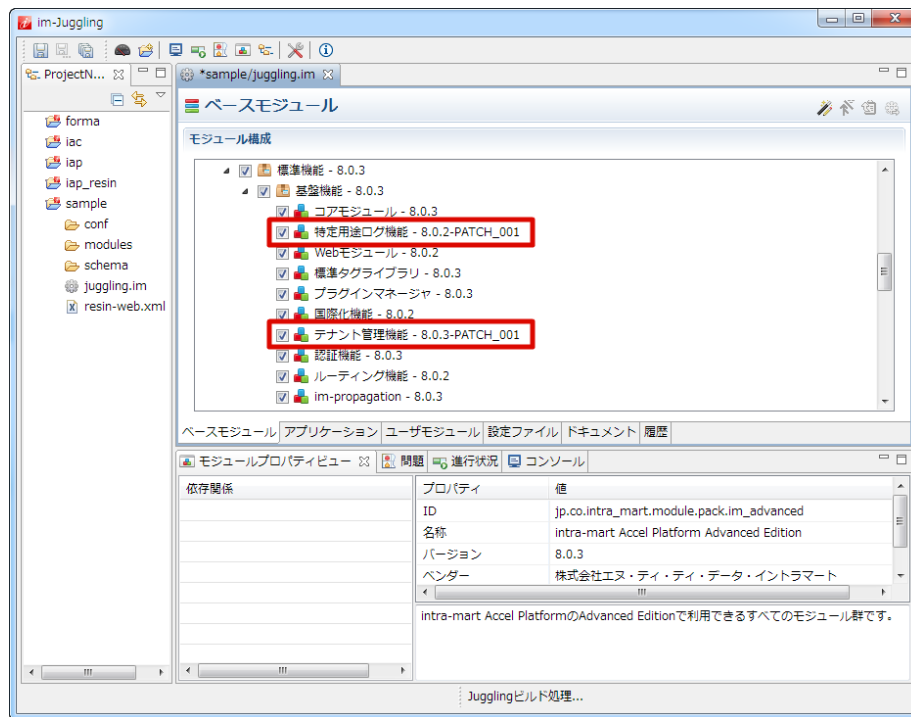
3. 前画面で「全てのパッチを適用する(推奨)」を選択していた場合、適用するパッチ内容が表示されます。適用するパッチ内容を確認し「終了 (E)」をクリックします。



4. 前画面で「個別に適用するパッチを選択する」を選択していた場合、パッチを適用するモジュールを選択します(チェックボックスにチェックをいれます)。選択後、「終了 (E)」をクリックします。



パッチ情報を取得し、既存プロジェクトのモジュール情報が変更されます。



5. パッチがプロジェクトに反映されたら、**WARファイルの出力**を行います。

WARファイルのアンデプロイ

- 現在稼働している環境のアンデプロイを行います。
Resin の場合は、[WARファイルのアンデプロイ](#)を参照してください。
Resin 以外の Web Application Server については、各製品ベンダから提供されているドキュメントを参照してください。

WARファイルの再デプロイ

- パッチを反映したWARファイルにてデプロイを行います。
Resin は、[WARファイルのデプロイ](#)を参照してください。
WebSphere Application Server 8.5.5 は、[セットアップガイド for WebSphere](#)を参照してください。
Oracle WebLogic Server 12c (12.1.3) は、[セットアップガイド for WebLogic](#)を参照してください。

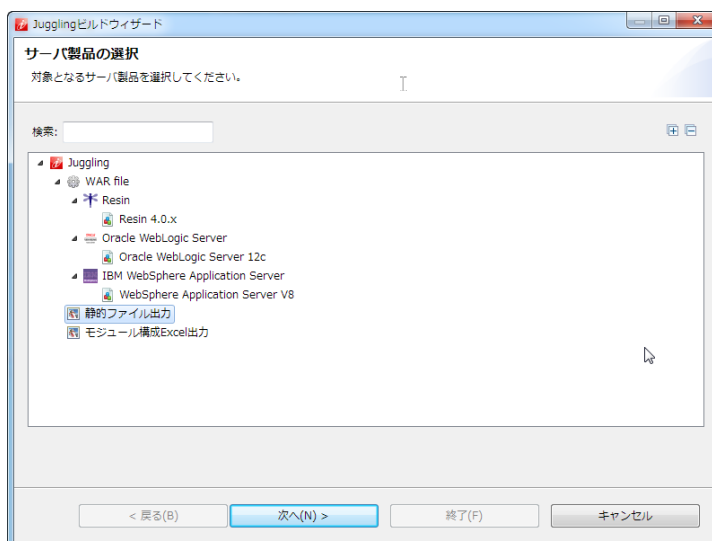


注意

再デプロイを行ったが、追加したモジュール（資材）がデプロイ先に反映されないといった事象が発生した場合は、[「Resin でWARファイルの再デプロイが正常にできない場合」](#)を参照してください。

静的ファイルの出力と再配置

- Web Server を利用している場合、IM-Juggling より静的ファイルを出力します。

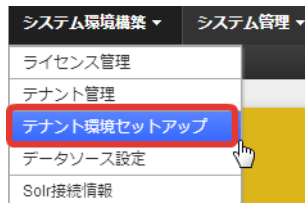


- 出力した静的ファイルは、Web Server でエイリアスとして設定しているディレクトリ配下に配置します。
※既存のファイルとは差し替えてください。

テナント環境セットアップ

- Web Application Server を起動後、テナント環境セットアップを実施します。

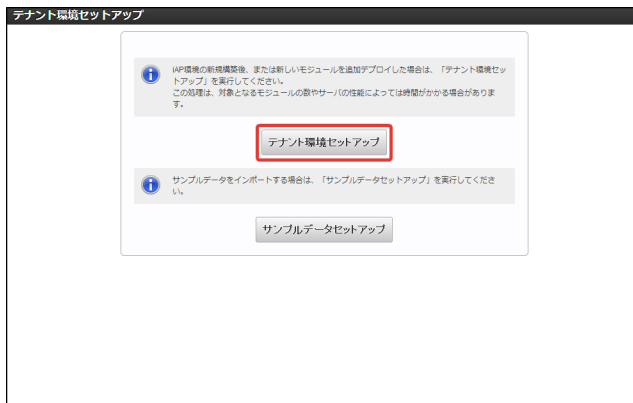
システム管理者のメニュー画面を表示します。
メニューから「テナント環境セットアップ」をクリックします。



下図のように、「テナント環境は最新です。セットアップが必要なモジュールはありません。」という旨のメッセージが表示されていれば、テナント環境セットアップは正常に完了しています。



下図のように、「テナント環境セットアップ」ボタンが表示されている場合、テナント環境セットアップは未完了です。「テナント環境セットアップ」ボタンをクリックしセットアップを再度実施します。



！ 注意

Web Server を経由してテナント環境セットアップを行う場合、Web Server のタイムアウト設置値を変更する、もしくは、Web Application Server 経由でセットアップを実施する事を推奨します。

i コラム

サンプルをセットアップするボタンは処理結果に関わらず、常に表示される仕様となります。このため、サンプルデータ投入中にエラーが発生した場合、再度セットアップを実施するとデータベース で一意制約違反が発生します。この場合は、「[アンインストール](#)」を行い、改めてセットアップからやり直す事を推奨します。

モジュール構成の変更

- IM-Juggling を利用し、モジュールやアプリケーションの追加を行うことができます。

モジュール構成の変更

項目

- ミドルウェアのセットアップ
- 古いリポジトリ情報の削除
- モジュール構成の変更
- アプリケーションの追加
- 設定ファイルの編集
- WARファイルの出力
- WARファイルのアンデプロイ
- WARファイルの再デプロイ
- 静的ファイルの出力と再配置
- テナント環境セットアップ

ミドルウェアのセットアップ

- モジュールやアプリケーションを追加する場合、新たにミドルウェアのセットアップが必要になる場合があります。例えば、IMBoxを追加する場合、Apache Cassandra のインストールおよび設定が必要です。詳細は、[ミドルウェアのセットアップ](#)を参照してください。

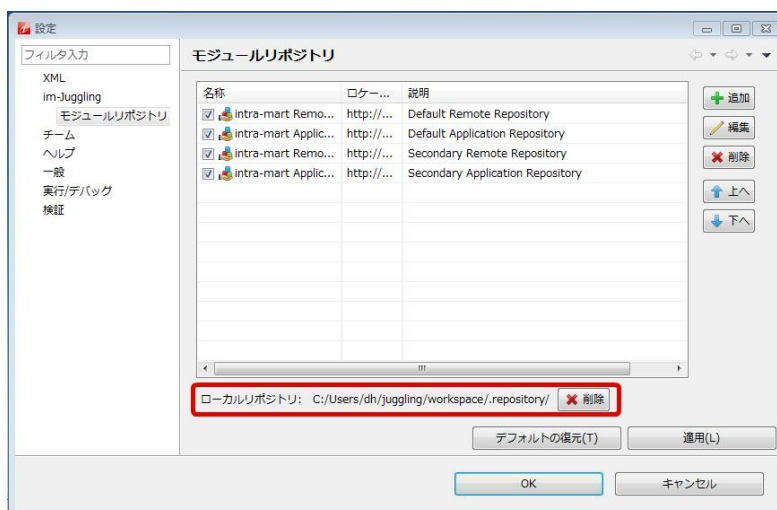


コラム

※ intra-mart Accel Platform 以外のアプリケーションについては、各セットアップガイドを参照してください。

古いリポジトリ情報の削除

- 前回使用した古いリポジトリ情報を削除します。
 1. IM-Juggling を起動し、ウィンドウ内-ツールバー右端にある「設定」-「IM-Juggling」-「モジュールリポジトリ」を開きます。
 2. 下部にある「ローカルリポジトリ」の削除をクリックしてください。



注意

ファイルの削除処理が行われている間は IM-Juggling の操作が行えない状態となります。

3. 削除が完了したらOKをクリックします。

モジュール構成の変更

1. モジュール構成の変更を行います。
IM-Juggling プロジェクト内に表示されているモジュールの右クリックメニューで表示されるサブメニューより、一括で関連するモジュールの選択、選択解除を行う事ができます。



注意

intra-mart Accel Platform の Edition の変更は IM-Juggling のプロジェクトの再作成が必要になります。

アプリケーションの追加

1. アプリケーションの追加を行います。
詳細は、[アプリケーションの追加](#)を参照してください。

設定ファイルの編集

- モジュールやアプリケーションを追加する場合、新たに設定ファイルの編集が必要になる場合があります。すでに出力して編集している設定ファイルは上書きされないため、再度編集する必要はありません。詳細は、[intra-mart Accel Platform の設定ファイル](#)を参照してください。また、新たに設定ファイルが必要となった場合は「[設定ファイルリファレンス](#)」を参照のうえ編集してください。



コラム

※ intra-mart Accel Platform 以外のアプリケーションについては、各セットアップガイドを参照してください。



コラム

※ 不要となったモジュールが出力したファイルが存在する場合は削除してください。

WARファイルの出力

1. 前述の IM-Juggling 上の設定が完了したら [WARファイルの出力](#) を行います。

WARファイルのアンデプロイ

1. 現在稼働している環境のアンデプロイを行います。
Resin の場合は、[WARファイルのアンデプロイ](#) を参照してください。
Resin 以外の Web Application Server については、各製品ベンダから提供されているドキュメントを参照してください。

WARファイルの再デプロイ

1. アップデート版を反映したWARファイルにてデプロイを行います。
Resin は、[WARファイルのデプロイ](#) を参照してください。
WebSphere Application Server 8.5.5 は、[セットアップガイド for WebSphere](#) を参照してください。
Oracle WebLogic Server 12c (12.1.3) は、[セットアップガイド for WebLogic](#) を参照してください。

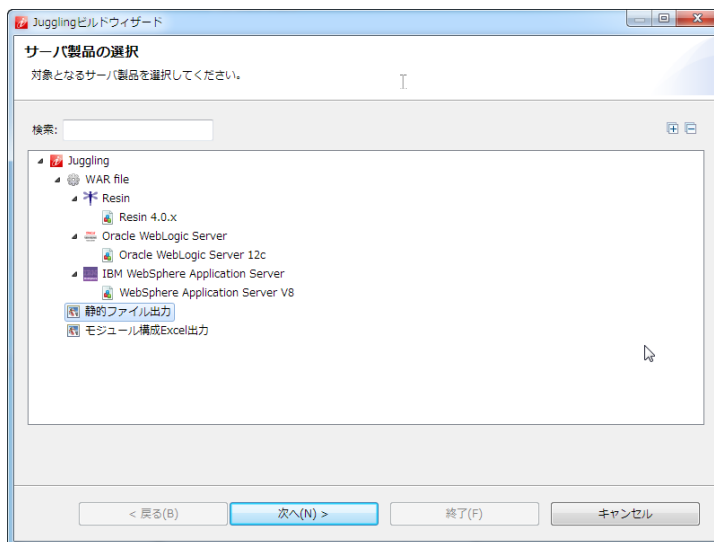


注意

再デプロイを行ったが、追加したモジュール（資材）がデプロイ先に反映されないといった事象が発生した場合は、
「[Resin でWARファイルの再デプロイが正常にできない場合](#)」を参照してください。

静的ファイルの出力と再配置

1. Web Server を利用している場合、IM-Juggling より静的ファイルを出力します。



2. 出力した静的ファイルは、Web Server でエイリアスとして設定しているディレクトリ配下に配置します。
※既存のファイルとは差し替えてください。

テナント環境セットアップ

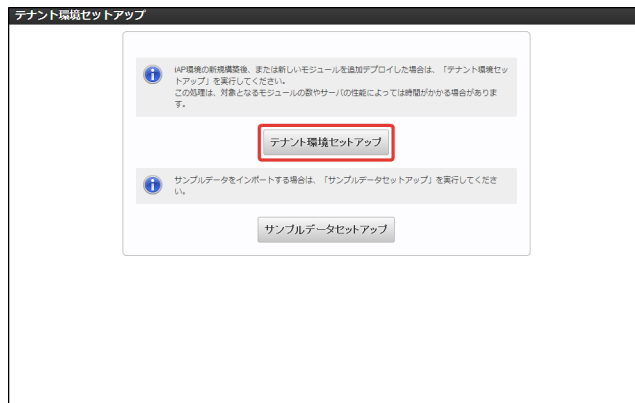
- Web Application Server を起動後、テナント環境セットアップを実施します。
システム管理者のメニュー画面を表示します。
メニューから「テナント環境セットアップ」をクリックします。



下図のように、「テナント環境は最新です。セットアップが必要なモジュールはありません。」という旨のメッセージが表示されていれば、テナント環境セットアップの実施は必要ありません。そのまま利用することができます。



下図のように、「テナント環境セットアップ」ボタンが表示されている場合、テナント環境セットアップが必要です。「テナント環境セットアップ」ボタンをクリックし、セットアップを再度実施します。



注意

Web Server を経由してテナント環境セットアップを行う場合、Web Server のタイムアウト設置値を変更する、もしくは、Web Application Server 経由でセットアップを実施する事を推奨します。

コラム

サンプルをセットアップするボタンは処理結果に関わらず、常に表示される仕様となります。このため、サンプルデータ投入中にエラーが発生した場合、再度セットアップを実施するとデータベース で一意制約違反が発生します。この場合は、「[アンインストール](#)」を行い、改めてセットアップからやり直す事を推奨します。

コラム

アップデート、パッチの適用およびモジュール構成の変更を行う前にバックアップを行う事を推奨します。詳細については、「[バックアップ・リストア\(復元\)](#)」を参照してください。

コラム

外部に接続できない環境で IM-Juggling を利用する場合、事前に次のサイトより最新版の IM-Juggling 、リボジトリデータをダウンロードして利用する事ができます。
<http://www.intra-mart.jp/download/product/iap/index.html>
 各種ダウンロード(zipファイル)
 ※ダウンロードには保守契約時に配布されたIDとパスワードが必要となります。

付録

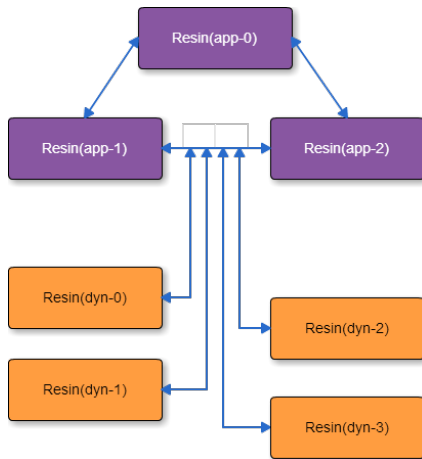
Resinのクラスタリング

項目

- 概要
- トライアドサーバ
- ダイナミックサーバ
- クラスタへのデプロイ

概要

Resinのクラスタリングは静的な3台のサーバ(トライアドサーバ)と、負荷に応じて自由に手動で追加・削除が可能なサーバ(ダイナミックサーバ)により構成されます。クラスタ構成にすることで負荷分散、デプロイとアンデプロイの伝播、セッションフェールオーバーが可能です。



トライアドサーバ

トライアドサーバは常に起動している静的なサーバです。クラスタの中心部でハブの役割となり互いに活性監視を行います。また、3重の冗長性により1台がメンテナンスを行っている場合でも残り2台が処理を継続することにより信頼性を保つことができます。

トライアドサーバを設定するには **resin.properties** ファイルを編集してください。トライアドサーバとなる3台のIPアドレスを「**app_servers**」に記述します。同様の設定を3台すべてに記述します。

```
# app-tier Triad servers: app-0 app-1 app-2
app_servers : 192.168.100.100 192.168.100.101 192.168.100.102
```

i コラム

app_servers にIPアドレスのみを記述した場合、ポート番号は resin.xml ファイルの <server-multi port="6800"/> が使用されます。任意のポート番号を使用する場合は「[IPアドレス]:[ポート番号]」と記述してください。

```
# app-tier Triad servers: app-0 app-1 app-2
app_servers : 192.168.100.100:6801 192.168.100.101:6802 192.168.100.102:6803
```

IPアドレスを記述した順に自動的にIDが割り振られます。IDはプレフィックスが「**app-**」となり0から始まる数字が割り当てられます。この記述例では下記のようにIDが割り当てられます。

```
192.168.100.100 app-0
192.168.100.101 app-1
192.168.100.102 app-2
```

各サーバ上からコマンドを実行してトライアドサーバを起動してください。コマンドに含まれるIDは各サーバに応じて自身のIDを指定してください。下記はIPアドレスが192.168.100.100のサーバから実行するコマンドの例です。192.168.100.101のサーバではIDを「app-1」に、192.168.100.102のサーバでは「app-2」に変更してコマンドを実行してください。

- Windowsの場合

```
> resin.exe -server app-0 console
```

- Linuxの場合

```
> resinctl -server app-0 console
```

以上でトライアドサーバを使用する手順は終了です。

i コラム

トライアドサーバは3重の冗長性を意味する名称です。1台または2台でも役割を行います。1台に障害が発生したりメンテナンスの場合でも、トライアドサーバは常に起動していなければならないため、最大で3台が起動可能な仕組みになっています。トライアドサーバが起動していないシステムではダイナミックサーバがクラスタに参加することができません。想定するシステム負荷からダイナミックサーバも必要なく2台で十分な場合は、2台のみのクラスタリング構成も可能です。



Windowsサービスへの登録については、「[Windowsサービスへの登録](#)」を参照してください。

ダイナミックサーバ

ダイナミックサーバを使用することでシステムの利用状況に応じて、自由に手動でサーバを追加・削除することができます。新規にダイナミックサーバを追加すると自動的にクラスタリングの調整監視の対象となります。

また、トライアドサーバからWebアプリケーションの配信とセッションのフェールオーバーが自動的に行われます。

ダイナミックサーバを設定するには **resin.properties** ファイルを編集してください。下記の3つの設定が必要です。

- **app_servers**
トライアドサーバとなる3台のIPアドレスを記述します。
- **elastic_cloud_enable**
コメントアウトされている属性を有効化し値を「true」とします。
- **home_cluster**
ダイナミックサーバを追加するクラスタのIDを指定します。**resin.xml** ファイルがデフォルト設定の場合は「app」としてください。

下記はダイナミックサーバを使用する場合の記述例です。

この設定はトライアドサーバとダイナミックサーバの両方に同じ設定を行ってください。

```
# app-tier Triad servers: app-0 app-1 app-2
app_servers : 192.168.100.100 192.168.100.101 192.168.100.102

...

# Allow elastic nodes to join the cluster (enable for cloud mode)
elastic_cloud_enable : true

...

# The cluster that elastic nodes should join - each will contact a Triad server
# Use a separate resin.properties file for each cluster
home_cluster : app
```

設定が記述された **resin.properties** ファイルは、サーバを容易に複製するためにコピーを保存しておくことをお勧めします。

ダイナミックサーバを実行します。サーバ上からコマンドを実行してください。

全てのトライアドサーバは予め起動してください。

- Windowsの場合

```
> resin.exe --elastic-server --cluster app console
```

- Linuxの場合

```
> resinctl --elastic-server --cluster app console
```

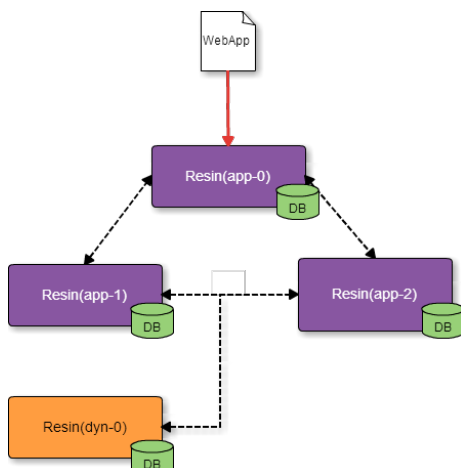
以上でダイナミックサーバを使用する手順は終了です。

クラスタへのデプロイ

クラスタへのデプロイは、単一のサーバへデプロイを実行すると全てのサーバへWebアプリケーションがレプリケートされデプロイが行われます。

ユーザはトライアドサーバへのみデプロイを行います。トライアドサーバへデプロイのコマンドを実行するとwarファイルはResinの内部データベースに登録されます。

登録されたwarファイルはその他のトライアドサーバとダイナミックサーバにレプリケートされ、各サーバでデプロイが自動的に行われます。



- デプロイの実行
トライアドサーバのどれか1台にwarファイルを転送し、サーバ上からコマンドを実行してください。

- Windowsの場合

```
> resin.exe deploy c:\imart.war
```

- Linuxの場合

```
> resinctl deploy ./imart.war
```

- アンデプロイの実行

アンデプロイも同様にサーバ上からコマンドを実行してください。トライアドサーバのどれか1台にのみ実行すれば、全てのサーバでアンデプロイが行われます。

- Windowsの場合

```
> resin.exe undeploy imart
```

- Linuxの場合

```
> resinctl undeploy imart
```

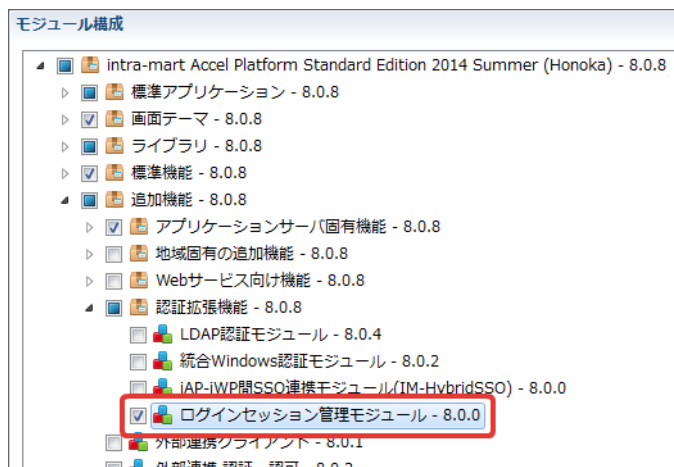
以上でデプロイの手順は終了です。

コラム

デプロイ、アンデプロイのコマンドはトライアドサーバにのみ可能です。
app-0,app-1,app-2のどれか1台に実行してください。
ダイナミックサーバに対してコマンドを実行することはできません。

二重ログイン防止機能

二重ログイン防止機能を利用する場合、IM-Juggling よりログインセッション管理モジュールをチェックしWARを作成してください。



統合Windows認証

項目

- 概要
- 前提条件
- セットアップ
 - 統合Windows認証機能の設定
 - テナント解決プラグインの設定
 - 統合Windows認証環境でWebサービスを利用する
 - 統合Windows認証環境で外部ソフトウェア連携機能を利用する
 - Internet Explorer の設定
- 統合Windows認証機能を無効化するには
- 統合Windows認証機能をリクエストに応じて無効化するには
 - Apache を使用して無効化する場合
 - Internet Information Services (IIS) を使用して無効化する場合
- 統合Windows認証機能の認証失敗時に通常のログイン機能を利用するには

概要

統合Windows認証機能は、ドメインコントローラ上の認証済みユーザと同じユーザコードをもつユーザで intra-mart Accel Platform アクセス時に、統合Windows認証済みユーザ情報を取得インを行うことができる機能です。

これにより、シングルサインオンを実現することができます。



注意

統合Windows認証は、Internet Information Services (IIS) が必須です。



注意

ログイン画面からのログインはサポートしません。詳しくは以下の制限を参照してください。

[SSO \(SingleSignOn\) 環境での注意点があります。](#)

前提条件

- アプリケーションサーバには Resin を使用してください。
- Resin を Windows Server 上で動作させる必要があります。
- Resin を実行する Windows Server がドメインに参加している必要があります。
- ブラウザは「[リリースノート](#)」に記した Internet Explorer を使用してください。
- 統合Windows認証モジュールが必要になります。
 - IM-Juggling 上で「追加機能」-「統合Windows認証モジュール」を選択し intra-mart Accel Platform を構築する必要があります。
 - 統合Windows認証モジュールは、8.0.0-PATCH_001 以降のバージョンを使用してください。



注意

シングルサインオンを実現するためには、ドメインコントローラ上のユーザコードと intra-mart Accel Platform 上のユーザコードが一致している必要があります。



注意

統合Windows認証機能には、ドメインコントローラ、統合Windows認証に対応したブラウザが必要になります。



注意

スマートフォンでの統合Windows認証機能の利用はサポートしていません。



注意

統合Windows認証は intra-mart Accel Platform の機能として Resin 上で実行されます。そのため、IIS の Windows 認証を無効化してください。
その他認証は必要に応じて設定してください。

セットアップ

統合Windows認証機能の設定

IM-Juggling で次の設定を行いwarファイルを作成してください。

- 機能の有効化の設定
 1. <(プロジェクト名)/juggling.im>の「ベースモジュール」タブで統合Windows認証モジュール(im-sso-windows)をチェックします。
「設定ファイル」タブの統合Windows認証モジュールを選択しim-sso-windows-config.xmlを出力します。
 2. <(プロジェクト名)/conf> 配下に出力されたim-sso-windows-config.xmlを開き<im-sso-windows-config>/<authentication>/<@enable> を true に設定します。
- テナント解決の設定 (intra-mart Accel Platform 2014 Spring(Granada) 以降のみ)

認証対象となるテナントを解決する方法として plugin を採用しています。
plugin の設定仕様については、「[PluginManagerのAPIドキュメント](#)」を参照してください。

1. 「plugin/jp.co.intra_mart.foundation.admin.tenant.context.tenant.resolver.windows_8.0.1」フォルダを作成し、そのフォルダ内に以下のようなplugin.xmlを配置してくださ

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin>
  <extension point="jp.co.intra_mart.foundation.admin.tenant.context.tenant.resolvers">
    <tenant-id-resolvers
      id="jp.co.intra_mart.foundation.admin.tenant.context.tenant.resolver.windows"
      name="Windows Tenant Id Resolver"
      version="8.0.1"
      rank="90">
      <!-- ドメインをテナントIDとするリゾルバー参加しているドメインをテナントIDとして解決します。 -->
      <tenant-id-resolver class="jp.co.intra_mart.foundation.admin.tenant.context.DomainBasedWindowsTenantIdResolver"/>
      <!-- 設定された固定値をテナントIDとするリゾルバーtenant_idパラメータに設定した値をテナントIDとして解決します。 -->
      <!-- tenant-id-resolver class="jp.co.intra_mart.foundation.admin.tenant.context.FixedConfigBasedWindowsTenantIdResolver">
      <init-param>
        <param-name>tenant_id</param-name>
        <param-value>default</param-value>
      </init-param>
      </tenant-id-resolver-->
    </tenant-id-resolvers>
  </extension>
</plugin>
```

2. <plugin>/<extension>/<tenant-id-resolvers>/<tenant-id-resolver>/<@class> に実装クラスを指定し、必要に応じて実装クラスにパラメータを設定します。
 - 使用可能な実装クラス、および設定するパラメータについては [テナント解決プラグインの設定](#) を参照してください。

以上で統合Windows認証機能の設定は終了です。

テナント解決プラグインの設定

テナント解決を行う仕組みとして、以下の二つの実装クラスを用意しています。

実装クラス	概要
jp.co.intra_mart.foundation.admin.tenant.context.DomainBasedWindowsTenantIdResolver	ユーザが参加しているドメインをテナントIDとして解決します。
jp.co.intra_mart.foundation.admin.tenant.context.FixedConfigBasedWindowsTenantIdResolver	設定された固定値をテナントIDとして解決します。

- jp.co.intra_mart.foundation.admin.tenant.context.DomainBasedWindowsTenantIdResolver
ユーザが参加しているドメインをテナントIDとして解決します。
本実装クラスに必要なパラメータはありません。
- jp.co.intra_mart.foundation.admin.tenant.context.FixedConfigBasedWindowsTenantIdResolver

tenant_id パラメータを受け取り、その値をテナントIDとして解決します。
下記の設定例では、「default」がテナントIDとして解決されます。

```
<tenant-id-resolver class="jp.co.intra_mart.foundation.admin.tenant.context.FixedConfigBasedWindowsTenantIdResolver">
  <init-param>
    <param-name>tenant_id</param-name>
    <param-value>default</param-value>
  </init-param>
</tenant-id-resolver>
```

統合Windows認証環境でWebサービスを利用する

統合Windows認証環境でWebサービスを利用するためには、以下の設定が必要です。

IM-Juggling で次の設定を行いwarファイルを作成してください。

- 機能の有効化の設定
 1. <(プロジェクト名)/juggling.im>の「ベースモジュール」タブでWebサービス 認証・認可(im_ws_auth)、Webサービス認証・認可クライアント(im_ws_auth_client)、Apache / をチェックします。
「設定ファイル」タブのwebサービス 認証・認可を選択しim-sso-windows-path-config/im-sso-windows-path-config_ws_auth.xmlを出力します。
 2. <(プロジェクト名)/conf/im-sso-windows-path-config> 配下に出力されたim-sso-windows-path-config_ws_auth.xmlを開き以下のようにコメントを外して、設定を有効化

```
<no-authentication>
  <path regex="true">/services/. * </path>
  <path regex="true">/axis2-admin/. * </path>
</no-authentication>
```

コラム

ベースモジュール選択時にWebサービスライブラリをチェックし依存関係を解消することでモジュールを探すことなく設定ファイルを出力可能になります。

統合Windows認証環境で外部ソフトウェア連携機能を利用する

統合Windows認証環境で外部ソフトウェア連携機能を利用するためには、以下の設定が必要です。

IM-Juggling で次の設定を行いwarファイルを作成してください。

- 機能の有効化の設定
 1. <(プロジェクト名)/juggling.im>の「ベースモジュール」タブで 外部連携認証 認可(imaca_provider)、外部連携クライアント(imaca_client)をチェックします。

「設定ファイル」タブの外部連携 認証・認可を選択しconf/im-sso-windows-path-config/im-sso-windows-path-config_imaca_provider.xmlを出力します。

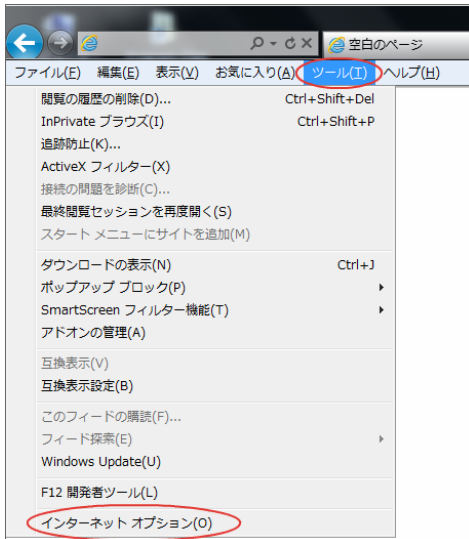
2. <(プロジェクト名)/conf/im-sso-windows-path-config> 配下に入力されたim-sso-windows-path-config_imaca_provider.xmlを開き以下のようにコメントを外して、設定をする。

```
<no-authentication>
<path>/HTTPActionEventListener</path>
</no-authentication>
```

Internet Explorer の設定

Internet Explorer の設定を変更することにより、Windows にログインしているユーザで自動的に intra-mart Accel Platform にログインすることができます。

1. インターネット オプションを表示します。



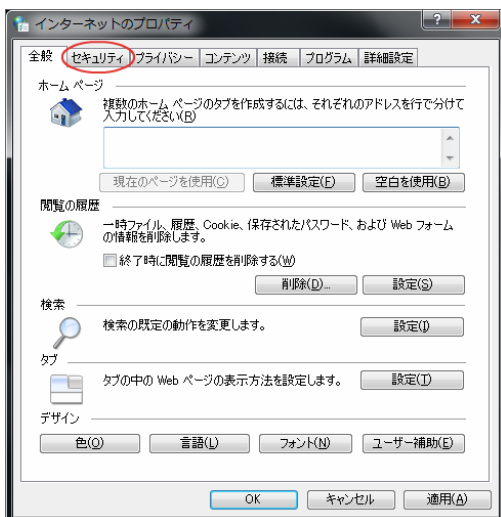
コラム

ツールメニューが表示されていない場合、Alt キーを押すことで表示できます。

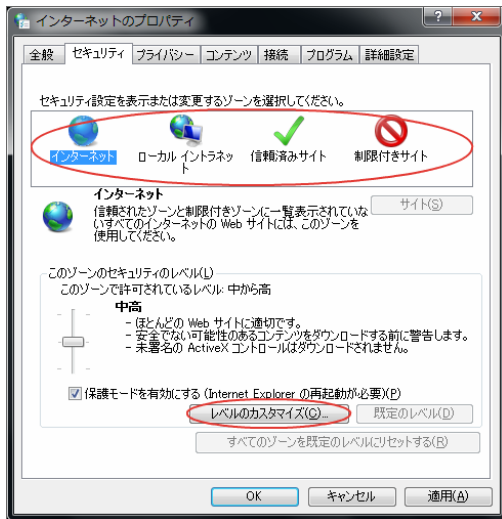
コラム

インターネット オプションはコントロールパネルからも表示することができます。

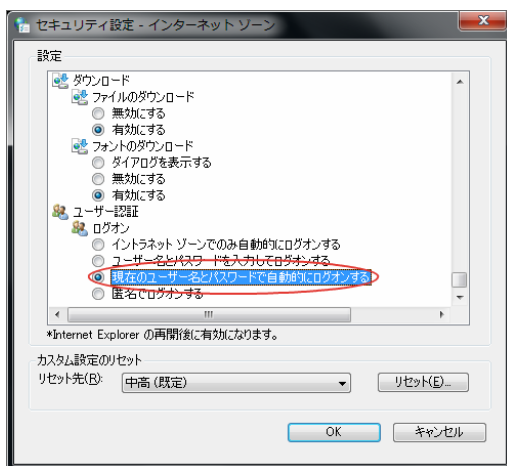
2. セキュリティ設定画面を表示します。



3. intra-mart Accel Platform に該当するゾーンのレベルのカスタマイズ画面を表示します。



4. ユーザ認証の設定を変更します。
「現在のユーザ名とパスワードで自動的にログオンする」をチェックします。



コラム

上記の設定を行うことで、Internet Explorer 利用時に自動的にユーザ認証を行うことができます。

統合Windows認証機能を無効化するには

統合Windows認証機能を無効化したい場合、以下の手順を実施するか、または単に統合Windows認証モジュールを含めずに war を作成し、再デプロイを行ってください。

IM-Juggling で次の設定を行いwarファイルを作成してください。

- 機能の無効化の設定
 1. <(プロジェクト名)/conf> 配下に出力されているim-sso-windows-config.xmlを開き<im-sso-windows-config>/<authentication>/<@enable> を false に設定します。

```
<?xml version="1.0" encoding="UTF-8"?>
<im-sso-windows-config
  xmlns="http://www.intra-mart.jp/sso-windows/config/im-sso-windows-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/sso-windows/config/im-sso-windows-config ../schema/im-sso-windows-config.xsd">
  <authentication enable="false">
    <parameter>
      <param-name>allow-guest-login</param-name>
      <param-value>false</param-value>
    </parameter>
  </authentication>
</im-sso-windows-config>
```

コラム

デフォルトではfalseが指定されています。

2. plugin配下にフォルダを作成(例:「jp.co.intra_mart.foundation.security.certification.sso.user.provider.windows_8.0.0.disable」)し、そのフォルダ内に以下のようなplugin してください。
version 属性にはすでに差し込まれているプラグイン ID の version 属性より大きい値を設定してください。

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin>
  <extension point="jp.co.intra_mart.foundation.security.certification.sso.user.providers">
    <sso-user-providers
      id="jp.co.intra_mart.foundation.security.certification.sso.user.provider.windows"
      name="Windows SSO User Provider"
      version="8.0.1"
      rank="90"
      enable="false">
      <sso-user-provider class="jp.co.intra_mart.foundation.security.certification.sso.WindowsSSOUserProvider"/>
    </sso-user-providers>
  </extension>
</plugin>
```

3. plugin配下にフォルダを作成(例:「jp.co.intra_mart.foundation.admin.tenant.context.tenant.resolver.windows_8.0.1.disable」)し、そのフォルダ内に以下のようなplugin.x
てください。
version 属性にはすでに差し込まれているプラグイン ID の version 属性より大きい値を設定してください。

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin>
  <extension point="jp.co.intra_mart.foundation.admin.tenant.context.tenant.resolvers">
    <tenant-id-resolvers
      id="jp.co.intra_mart.foundation.admin.tenant.context.tenant.resolver.windows"
      name="Windows Tenant Id Resolver"
      version="8.0.2"
      rank="90"
      enable="false">
      <!-- ドメインをテナントIDとするリゾルバー参加しているドメインをテナントIDとして解決します。 -->
      <tenant-id-resolver class="jp.co.intra_mart.foundation.admin.tenant.context.DomainBasedWindowsTenantIdResolver"/>

      <!-- 設定された固定値をテナントIDとするリゾルバー
            tenant_idパラメータに設定した値をテナントIDとして解決します。 -->
      <!--tenant-id-resolver class="jp.co.intra_mart.foundation.admin.tenant.context.FixedConfigBasedWindowsTenantIdResolver">
        <init-param>
          <param-name>tenant_id</param-name>
          <param-value>default</param-value>
        </init-param>
      </tenant-id-resolver-->
    </tenant-id-resolvers>
  </extension>
</plugin>
```

4. plugin配下にフォルダを作成(例:「jp.co.intra_mart.foundation.admin.tenant.context.tenant.validator.windows_8.0.1.disable」)し、そのフォルダ内に以下のようなplugin.:
てください。
version 属性にはすでに差し込まれているプラグイン ID の version 属性より大きい値を設定してください。

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin>
  <extension point="jp.co.intra_mart.foundation.admin.tenant.context.tenant.validators">
    <tenant-id-validators
      id="jp.co.intra_mart.foundation.admin.tenant.context.tenant.validator.windows"
      name="Windows TenantIdValidator"
      version="8.0.2"
      rank="100"
      enable="false">
      <tenant-id-validator class="jp.co.intra_mart.system.sso_windows.context.WindowsSSOTenantIdValidator">
        <!-- テナントID解決必須チェック -->
        <init-param>
          <param-name>required_tenant_id</param-name>
          <param-value>true</param-value>
        </init-param>
        <!-- テナントID存在チェック -->
        <init-param>
          <param-name>valid_tenant_id</param-name>
          <param-value>true</param-value>
        </init-param>
      </tenant-id-validator>
    </tenant-id-validators>
  </extension>
</plugin>
```



コラム

プラグイン配下に作成するフォルダ名は任意です。
他のフォルダ名と重複することの無いようユニークな名称を設定してください。

統合Windows認証機能をリクエストに応じて無効化するには

intra-mart Accel Platform 2014 Summer(Honoka) 以降では、以下のリクエストヘッダーを設定することで、統合Windows認証機能、テナントバリデーション機能を無効化することができます

リクエストヘッダー	値	動作
x-jp-co-intra-mart-sso-windows-disable-authentication	true	該当のリクエストにおいて統合Windows認証を行いません。
x-jp-co-intra-mart-sso-windows-disable-tenant-validation	true	該当のリクエストにおいてテナントバリデーションを行いません。

**注意**

値には“true”を指定してください。その他の値を指定した場合の動作は保証しません。

Apache を使用して無効化する場合

Apache を利用して上記ヘッダーを設定するには、以下の手順を行ってください。

1. <%APACHE_HOME%/conf/httpd.conf> ファイルを開きます。
2. Dynamic Shared Object (DSO) Supportエリアに以下の設定を追加またはコメントアウトを外してください。

```
LoadModule headers_module modules/mod_headers.so
```

3. 下記設定を追加してください。

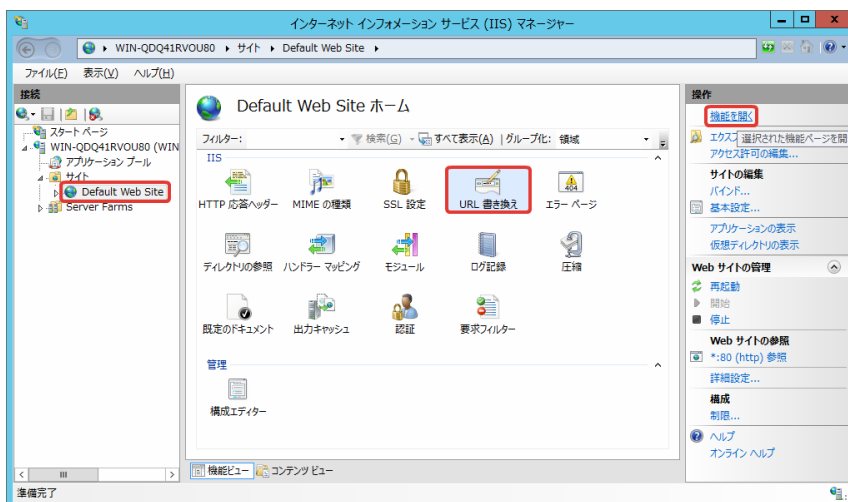
```
RequestHeader append x-jp-co-intra-mart-sso-windows-disable-authentication "true"
RequestHeader append x-jp-co-intra-mart-sso-windows-disable-tenant-validation "true"
```

4. Apache を再起動してください。

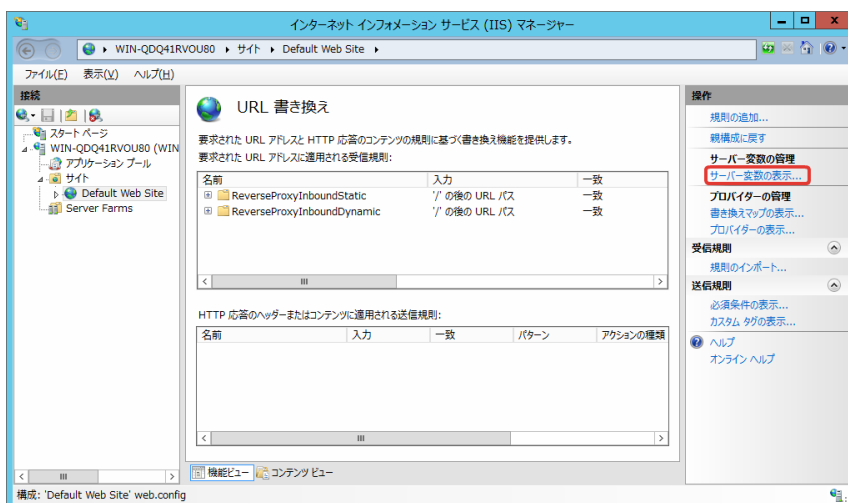
Internet Information Services (IIS) を使用して無効化する場合

IIS を利用して上記ヘッダーを設定するには、以下の手順を行ってください。

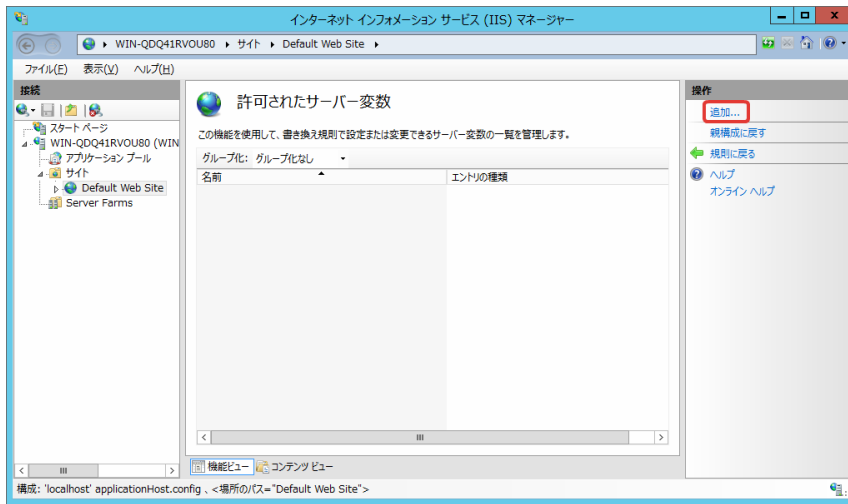
1. 統合Windows認証機能を無効化したいサイト、「URL 書き換え」、「機能を開く」の順にクリックします。



2. 「サーバ変数の表示」をクリックします。



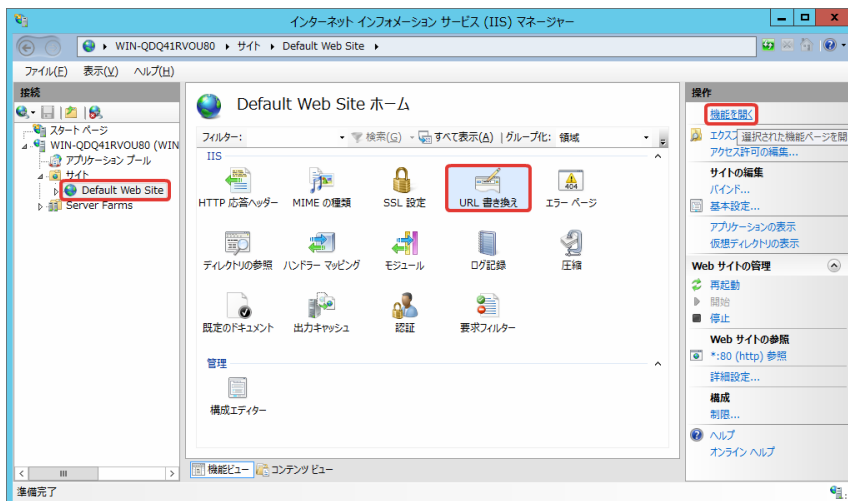
3. 「追加」をクリックします。



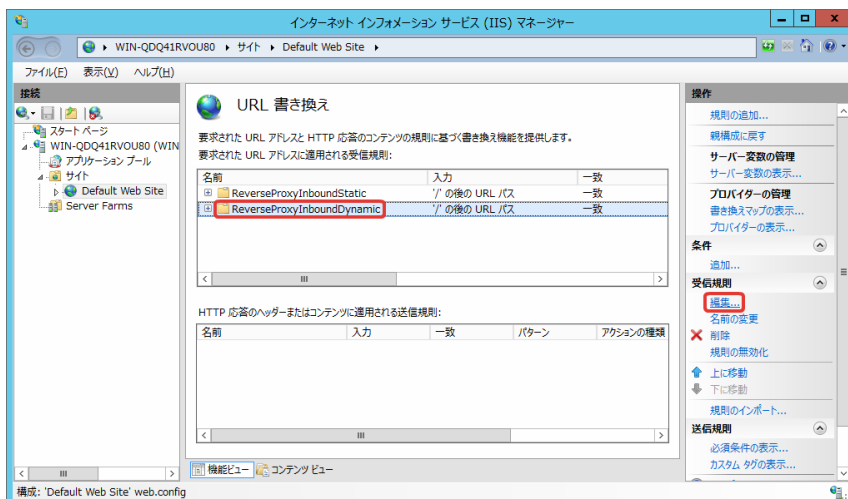
4. 「サーバー変数名」に「HTTP_X_JP_CO_INTRA_MART_SSO_WINDOWS_DISABLE_AUTHENTICATION」を入力し、「OK」をクリックします。



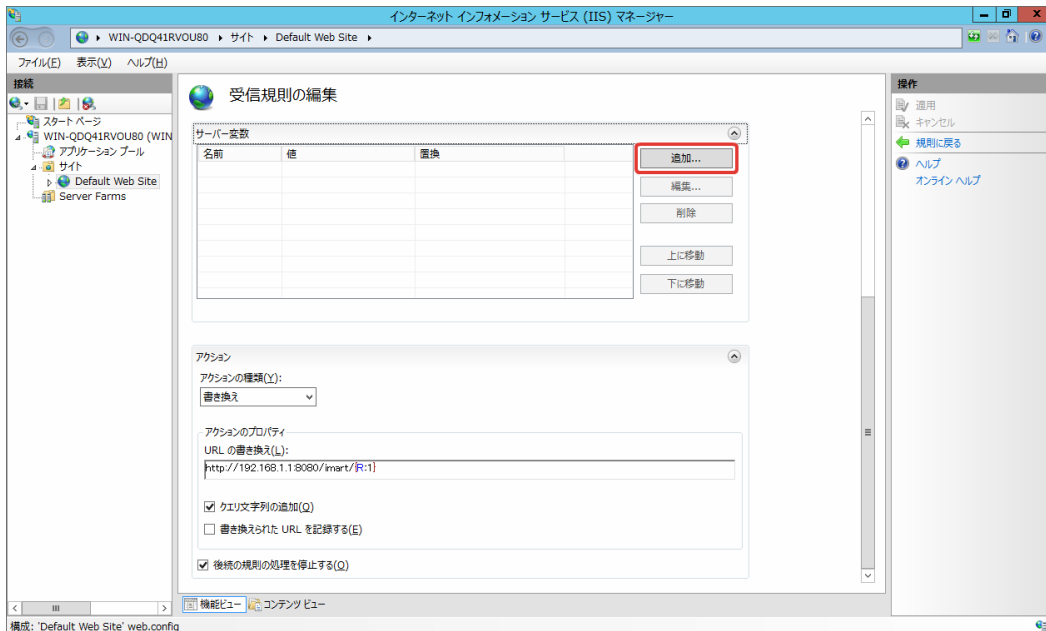
5. 同様に、サーバー変数「HTTP_X_JP_CO_INTRA_MART_SSO_WINDOWS_DISABLE_TENANT_VALIDATION」を追加します。
6. 統合Windows認証機能を無効化したいサイト、「URL 書き換え」、「機能を開く」の順にクリックします。



7. Resin への振り分けルール（ここでは ReverseProxyInboundDynamic）を選択し、編集をクリックします。



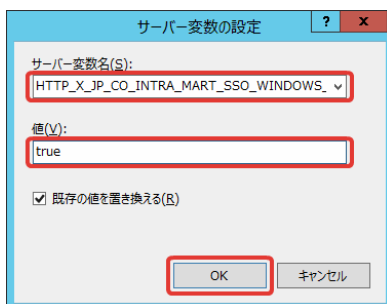
8. 「サーバー変数」の「追加」をクリックします。



9. 以下の内容を入力し、「OK」をクリックします。

サーバー変数名 HTTP_X_JP_CO_INTRA_MART_SSO_WINDOWS_DISABLE_AUTHENTICATION

値 true

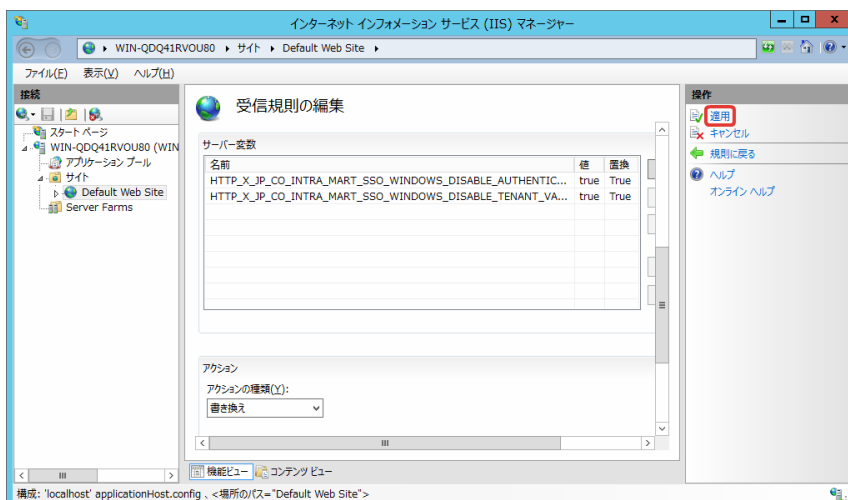


10. 同様に、以下のサーバー変数を追加します。

サーバー変数名 HTTP_X_JP_CO_INTRA_MART_SSO_WINDOWS_DISABLE_TENANT_VALIDATION

値 true

11. 「適用」をクリックします。



12. Internet Information Services (IIS) を再起動します。

統合Windows認証機能の認証失敗時に通常のログイン機能を利用するには

intra-mart Accel Platform 2014 Winter(Iceberg) 以降では、<(プロジェクト名)/conf> 配下に出力されている im-sso-windows-config.xml の allow-fallback-login パラメータに true を設定
統合Windows認証機能による認証に失敗したユーザは通常のログイン機能を使用することができます。

```
<?xml version="1.0" encoding="UTF-8"?>
<im-sso-windows-config
  xmlns="http://www.intra-mart.jp/sso-windows/config/im-sso-windows-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/sso-windows/config/im-sso-windows-config ../schema/im-sso-windows-config.xsd">
  <authentication enable="true">
    <parameter>
      <param-name>allow-guest-login</param-name>
      <param-value>false</param-value>
    </parameter>
    <parameter>
      <param-name>allow-fallback-login</param-name>
      <param-value>true</param-value>
    </parameter>
  </authentication>
</im-sso-windows-config>
```

⚠ 注意

本機能における、統合Windows認証 機能の認証失敗の条件は以下のような場合です。

- ユーザ認証に使用されるユーザ名、パスワードがドメインコントローラー上のものと一致しない場合。
- 認証ダイアログによる認証要求をキャンセルした場合。
- ゲストログインが無効の場合に、ドメインコントローラー上のゲストユーザで認証をした場合。

⚠ 注意

本機能有効時に 統合Windows認証 に失敗したユーザに対しては、以降セッションが破棄されるまで 統合Windows認証 は行われません。

IM-Notice

項目

- 概要
- 通知機能全般
 - モジュールの選択
 - IM-Notice 設定ファイルの編集
 - ベースURLの設定
- デスクトップ通知機能
 - スタンドアローン構成の場合
 - IM-Notice MQ設定ファイルの編集
 - 分散構成やWeb Serverを利用している構成の場合
 - IM-Notice MQ設定ファイルの編集
 - ブローカーサービス実行ファイルの取得
 - IM-Notice MQブローカー設定ファイルの編集
 - ブローカーサービスの実行
- モバイル通知機能(iOS版)
 - P12証明書ファイルの取得
 - Amazon SNSの設定
 - IM-Notice Mobile設定ファイルの編集
- モバイル通知機能(Android版)
 - GCMAを使用する場合
 - GCMAの設定
 - IM-Notice Mobile設定ファイルの編集
 - GCMA+Amazon SNSを使用する場合
 - GCMA の設定
 - Amazon SNS の設定
 - IM-Notice Mobile設定ファイルの編集

概要

IM-Noticeは、intra-mart Accel Platform 上で動作しているさまざまなアプリケーションからの通知をデスクトップやスマートフォンに配信するための通知機能です。
Amazon Simple Notification Service (以下、Amazon SNS) や Google Cloud Messaging for Android (以下、GCMA) のサービスを使用し、スマートフォンアプリへ通知を配信することが可

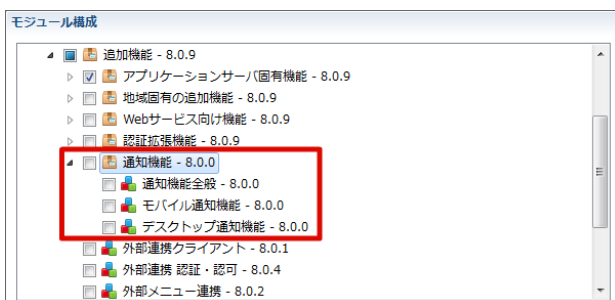
通知機能全般

通知機能全般を使用するための設定を行います。

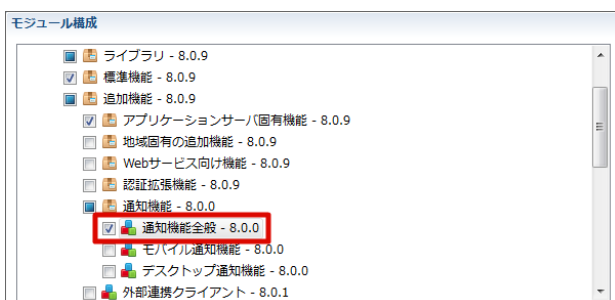
モジュールの選択

IM-Juggling でモジュールを選択します。

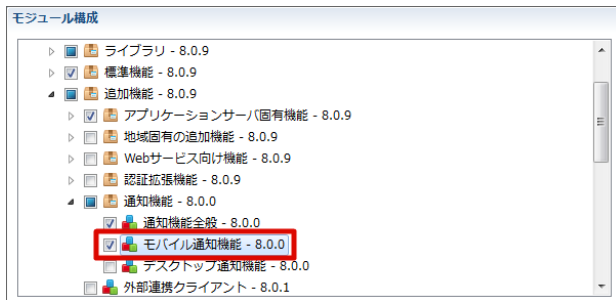
1. <(プロジェクト名)/juggling.im>の「ベースモジュール」タブから「通知機能」を開いてください。



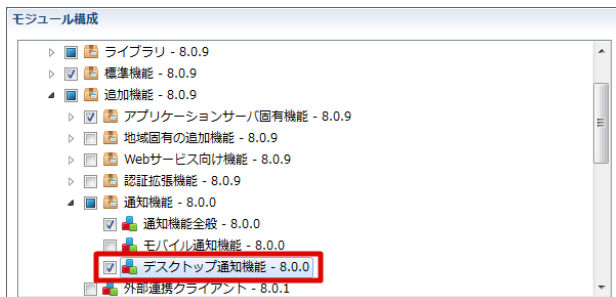
2. 通知機能全般をチェックします。



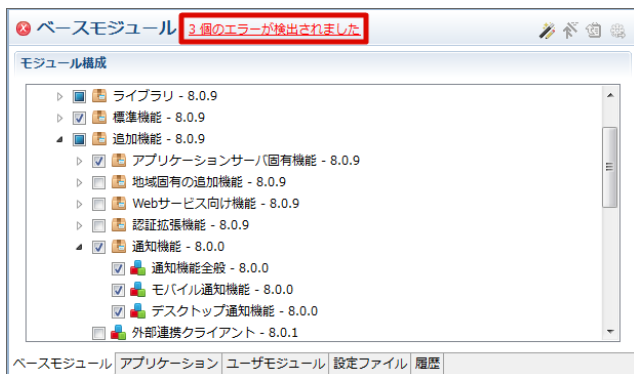
3. モバイルへの通知を行う場合、モバイル通知機能をチェックしてください。



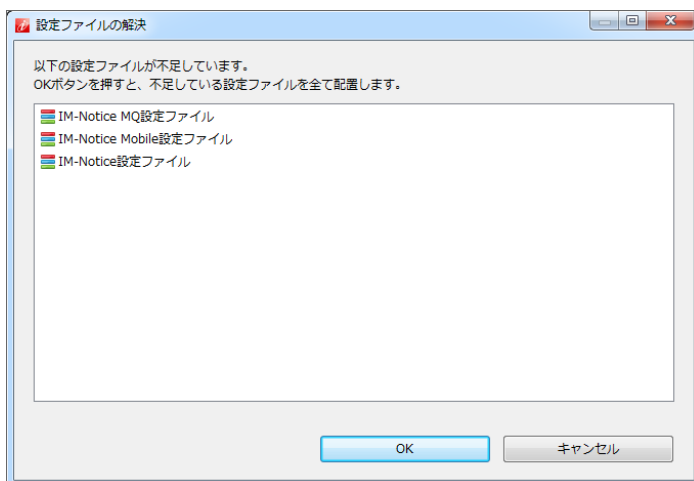
4. デスクトップへの通知を行う場合、デスクトップ通知機能をチェックしてください。



5. メッセージをクリックし、依存関係を解決してください。



6. 設定ファイルの解決を行うと、<(プロジェクト名)/conf> 配下に設定ファイルが出力されます。



IM-Notice 設定ファイルの編集

IM-Juggling の <(プロジェクト名)/conf> 配下に出力されたim-notice-config.xmlを開き、以下の設定を行ってください。

1. <base-url>

- ベースURLを設定してください。
この設定はショートカットURLの作成に使用されます。
未設定の場合、server-context-config.xml で指定したベースURLが使用されます。

```
<base-url>http://example.org/imart</base-url>
```

2. <short-cut-duration-minutes>

- ショートカットURLの有効期間を設定してください。単位は分です。
「0」を設定した場合、ショートカットURLは作成されません。
デフォルトでは43200分=30日が設定されています。

```
<short-cut-duration-minutes>43200</short-cut-duration-minutes>
```

ベースURLの設定

IM-Noticeへのメッセージ配信処理をジョブ経由で行う場合は、ベースURLの設定が必要です。「[ベースURL](#)」または、「[テナント環境情報](#)」から設定を行ってください。



注意

ベースURLを指定しない場合、以下のエラーが発生します。

jp.co.intra_mart.system.notice.exception.NoticeRuntimeException: [E.NOTICE.CORE.00016] ベースURLを解決できませんでした。

デスクトップ通知機能

デスクトップ通知機能を使用するための設定を行います。

スタンドアローン構成の場合

スタンドアローン構成の場合は、Web Application Server とデスクトップ通知のクライアントが直接通信を行います。

IM-Notice MQ設定ファイル (im-notice-mq-config) を設定する必要があります。

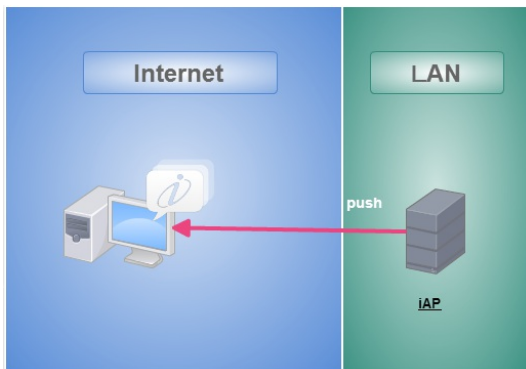


図 スタンドアローンのサーバ構成例 (デスクトップ通知)

IM-Notice MQ設定ファイルの編集

IM-Juggling の <(プロジェクト名)/conf> 配下に出力されたim-notice-mq-config.xmlを開き、以下の設定を行ってください。

1. <address>

- デスクトップ通知で使用するポート番号を設定してください。
デフォルトでは40608ポートを使用する設定になっています。
クライアントからサーバまでのネットワーク機器などに対して、ここで指定したポートを開放してください。

```
<address bind="true">tcp://*:40608</address>
```

2. <ping>

- ping機能の設定をします。
有効にするには、enable属性に「true」を設定してください。
- interval-seconds属性で、クライアントにpingを流す間隔を設定します。単位は秒です。
- time-to-live-seconds属性で、クライアントがサーバに再接続を行うまでの時間を設定します。単位は秒です。
設定した秒数だけサーバから応答がなければ、クライアントはサーバに再接続を行います。

```
<ping enable="true" interval-seconds="270" time-to-live-seconds="300" />
```



コラム

環境により、一定時間データが流れないソケットは切断されてしまう場合があります。
そのような環境では、数分間隔でpingを流し続けることで自動切断を避けることが出来ます。
(Microsoft AzureやAmazon Elastic Compute Cloudなど)

3. <endpoint>

- クライアントの接続先を設定します。
<address>で設定したポート番号を指定してください。

```
<endpoints>
  <endpoint>tcp://example.org:40608</endpoint>
</endpoints>
```

分散構成やWeb Serverを利用している構成の場合

分散構成やWeb Serverを利用している構成の場合は、ブローカーのサービスを実行するサーバの起動、および設定ファイルの編集が必要です。

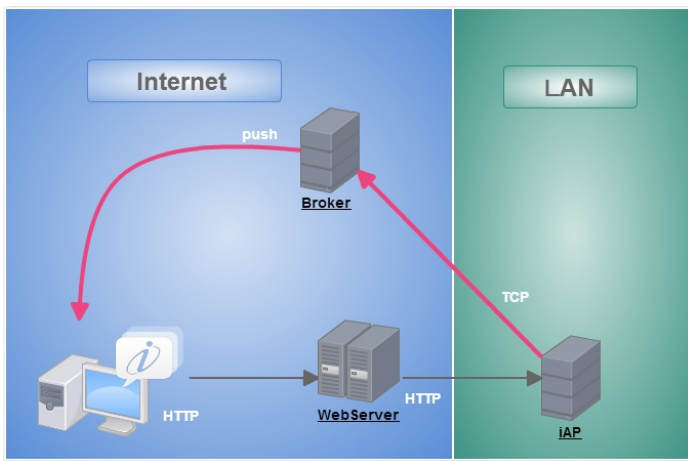


図 WebServerなどを含む分散構成のサーバ構成例 (デスクトップ通知)

コラム

ブローカーとは、ネットワーク中継を行うためのプロキシです。

コラム

ブローカーサービスを実行するサーバのシステム要件は「リリースノート」-「システム要件」を参照してください。

IM-Notice MQ設定ファイルの編集

IM-Juggling で出力されたim-notice-mq-config.xmlを編集してください。

1. <address>
bind属性にfalseを指定してください。

```
<address bind="false">tcp://{ブローカーを実行しているサーバのホストまたはIPアドレス}:{ブローカーのバックエンドポート番号}</address>
<ping enable="true" interval-seconds="270" time-to-live-seconds="300" />
<endpoints>
  <endpoint>tcp://{ブローカーを実行しているサーバのホストまたはIPアドレス}:{ブローカーのフロントエンドポート番号}</endpoint>
</endpoints>
```

コラム

IM-Notice MQ設定ファイルのその他の項目については [IM-Notice MQ設定ファイルの編集](#) を参照してください。

ブローカーサービス実行ファイルの取得

ブローカーサービスを実行する実行ファイル(im_notice_mq_broker)を取得します。
弊社サイトのダウンロードライブラリ、または製品メディアイメージから取得してください。

ダウンロードライブラリ



IM-Notice MQブローカー設定ファイルの編集

取得したブローカーサービス実行ファイルを展開したconfディレクトリ内のim-notice-mq-broker-config.xmlを編集します。

1. <frontend>
 - デスクトップ通知アプリケーションとの通信を行うポート番号を設定してください。
デフォルトでは40608ポートを使用する設定になっています。

クライアントからブローカーまでのネットワーク機器などに対して、ここで指定したポートを開放してください。

```
<frontend bind="true">tcp://*:40608</frontend>
```

2. <backend>

- Web Application Server との通信を行うポート番号を設定してください。
デフォルトでは40609ポートを使用する設定になっています。
ブローカーからサーバまでのネットワーク機器などに対して、ここで指定したポートを開放してください。

```
<backend bind="true">tcp://*:40609</backend>
```

コラム

IM-Notice MQブローカー設定ファイルのその他の項目は IM-Notice ブローカー設定ファイルと同様です。
IM-Notice ブローカー設定ファイルについては [IM-Notice MQ設定ファイルの編集](#) を参照してください。

ブローカーサービスの実行

取得したブローカーサービス実行ファイルを展開したディレクトリにある im_notice_mq_broker-8.0.0.jar をjavaコマンドで実行してください。

フォアグラウンドで動作させる場合 (例)

```
java -jar im_notice_mq_broker-8.0.0.jar
```

バックグラウンドで動作させる場合 (例)

```
nohup java -jar im_notice_mq_broker-8.0.0.jar > stdout.log 2>stderr.log &
```

モバイル通知機能 (iOS版)

モバイル通知機能を使用するための設定を行います。iOSへの通知を行うには、Amazon SNSを使用します。

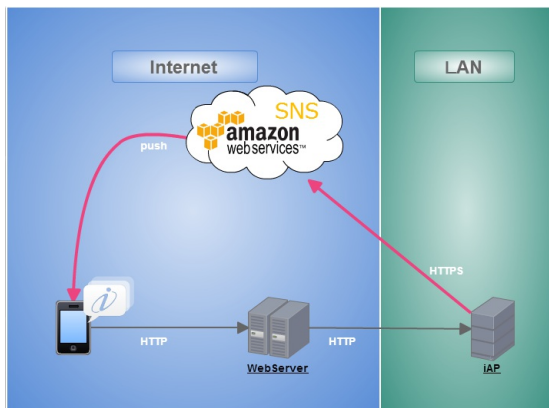


図 Amazon SNSを利用したサーバ構成例

P12証明書ファイルの取得

P12証明書ファイル (IM-Notice Push Notifications Production.p12) を取得します。
弊社サイトのダウンロードライブラリから取得してください。

ダウンロード ライブラリ

intra-mart Intra-mart Accel Series ダウンロードライブラリ

IM-Notice		
デスクトップ通知 2014-12-01 現在の最新バージョン: 8.0.0		
デスクトップ通知	[日本語]	
デスクトップ通知	[英語]	
デスクトップ通知	[中国語]	
IM-Noticeブローカー		(191 KB)
IM-Notice用証明書	(有効期限 2015/10/31)	(4 KB)
ソース		
製品公開ソース		(13 MB)
移行ソース		(41 KB)
サンプルソース		(96 KB)
互換モジュールソース		(5 MB)
IM-共通マスタ / IM-Workflow 資料		
intra-mart.jp/download/-/im_notice_mq_broker.zip	マスタ インポート・エクスポートXMLスキーマ	(13 KB)

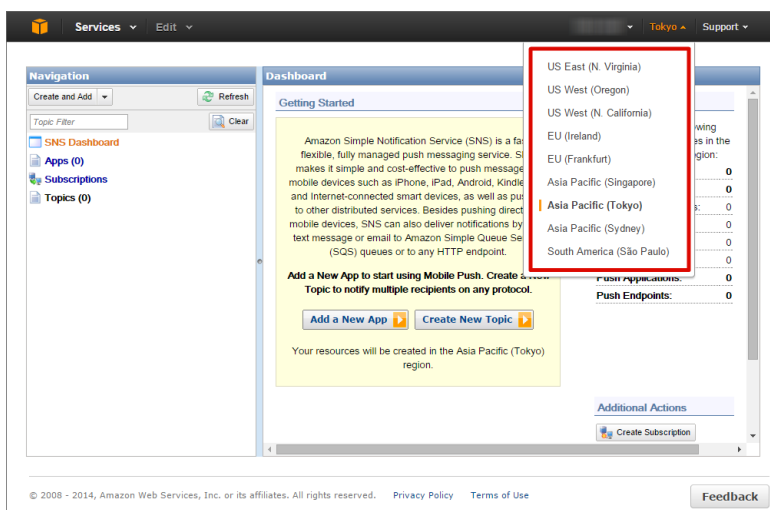
Amazon SNSの設定

Amazon SNSの設定を行います。

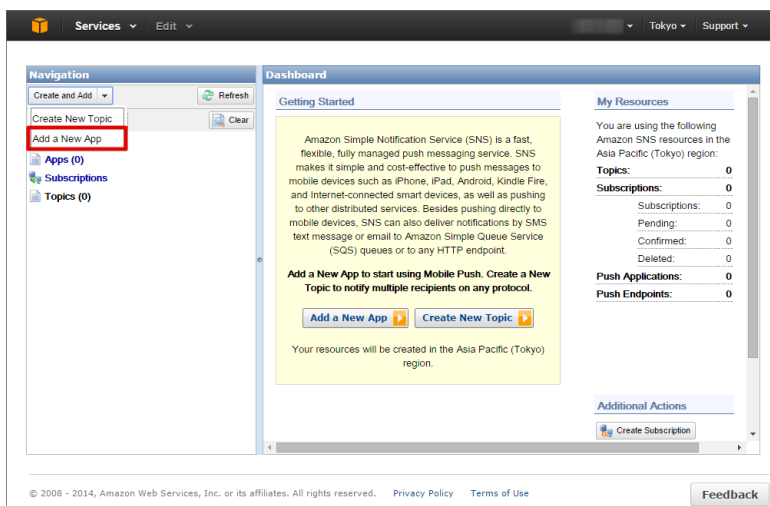
1. Amazon SNS にサインインしてください。



2. リージョンを選択してください。



3. 「Create and Add」から「Add a New App」を選択してください。



4. フォームに以下の内容を入力し、「Add New App」をクリックしてください。

項目	説明
Application Name	任意の名前を入力します。
Push Platform	「Apple Push Notification Service(APNS)」を選択します。
Choose P12 File	「IM-Notice Push Notifications Production.p12」を添付します。
Enter Password	「intra-mart」と入力し、「Load Credentials from File」をクリックします。

5. 「Apps」にアプリケーションが追加されました。

6. 「Application ARN」を確認することができます。
設定ファイルを編集する際に必要な文字列ですので、控えておいてください。

IM-Notice Mobile設定ファイルの編集

IM-Juggling の <(プロジェクト名)/conf> 配下に出力されたim-notice-mobile-config.xmlを開き、以下の設定を行ってください。

- <proxy>
 1. Amazon SNSへ接続するためにプロキシを使用する場合、設定を行ってください。

```
<proxy enable="true">
  <host></host>
  <port xsi:nil="true"></port> <!-- ポート番号を指定する場合は xsi:nil="true" を削除してください。 -->
  <username></username>
  <password></password>
  <workstation></workstation>
  <domain></domain>
</proxy>
```

- <asns>

1. <asns>のname属性に任意の名前を設定してください。

```
<asns name="asns-apns">
```

2. <endpoint>

- Amazon SNSのリージョンに対応するEndpointを設定してください。

```
<endpoint>sns.ap-northeast-1.amazonaws.com</endpoint>
```

コラム

Endpointは [AWS documentation - Regions and Endpoints](#) で確認できます。
リージョンが「Asia Pacific (Tokyo)」の場合、Endpoint は「sns.ap-northeast-1.amazonaws.com」です。

3. <platform-application-arn>

- [Amazon SNSの設定](#) で確認したApplication ARNを設定してください。

```
<platform-application-arn>arn:aws:sns:ap-northeast-1:XXXXXXXXXXXX:app/APNS/IM-Notice_APNS</platform-application-arn>
```

4. <access-key>

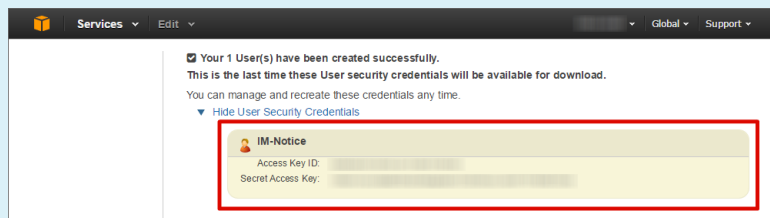
- アクセスキーを設定してください。

5. <secret-key>

- シークレットキーを設定してください。

コラム

アクセスキーおよびシークレットキーは [AWS Identity and Access Management \(IAM\)](#) で作成することができます。



- <push type="ios" />

1. engine属性に、<asns>のname属性に設定した名前を指定してください。

```
<push type="ios" engine="asns-apns" />
```

モバイル通知機能 (Android版)

モバイル通知機能を使用するための設定を行います。Androidへの通知を行うには、以下の2種類の構成が利用できます。

- GCMA を使用する場合の設定方法は、[GCMAを使用する場合](#) を参照してください。

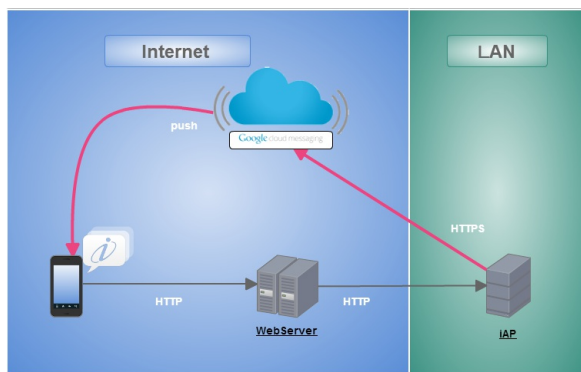


図 GCMA を使用する場合のサーバ構成例

- GCMA+Amazon SNSを使用する場合の設定方法は、[GCMA+Amazon SNSを使用する場合](#) を参照してください。

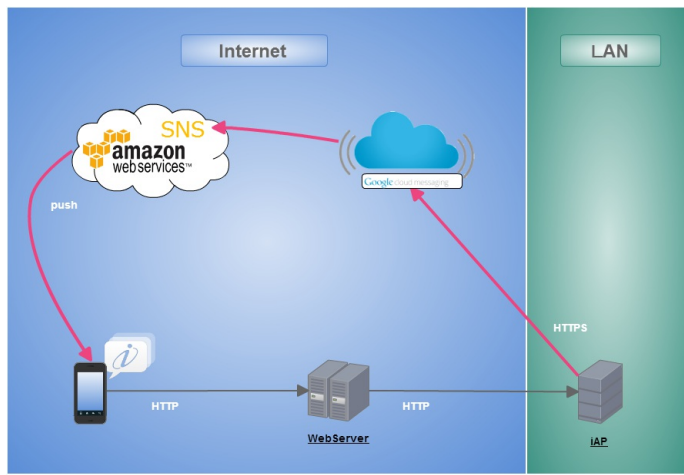


図 GCMA+Amazon SNS を使用する際のサーバ構成例

i コラム

Amazon SNSを組み合わせて使用することで、Push通知対象となっているデバイスのトークンリストや、その状態（有効あるいは無効）をAmazon SNSで一覧することができます。また、AndroidとiOSを併用する場合、Amazon SNSで一元管理することができます。

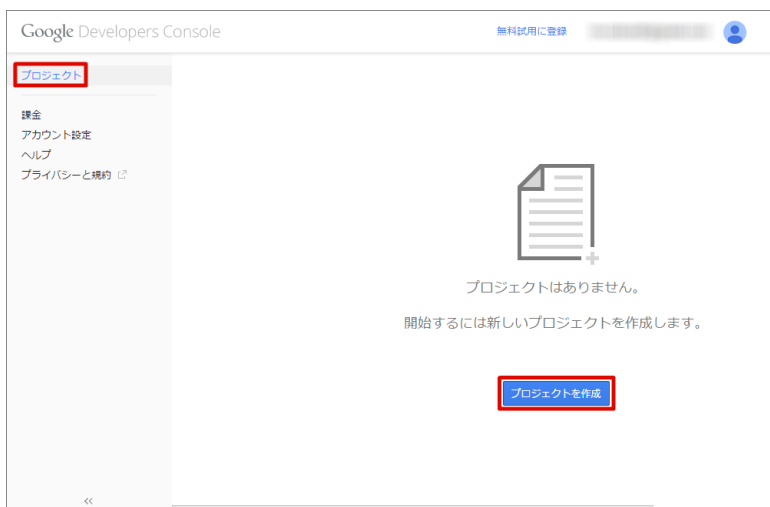
GCMAを使用する場合

GCMAを使用しAndroidへの通知を行う場合、以下の設定を行ってください。

GCMAの設定

GCMAの設定を行います。

1. [Google Play Developer Console](#) にログインしてください。
2. プロジェクトを作成します。

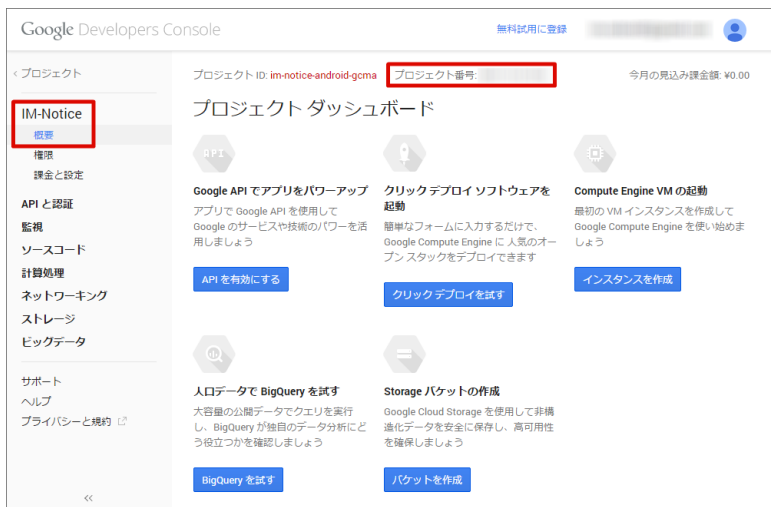


3. フォームに以下の内容を入力し、「作成」をクリックしてください。

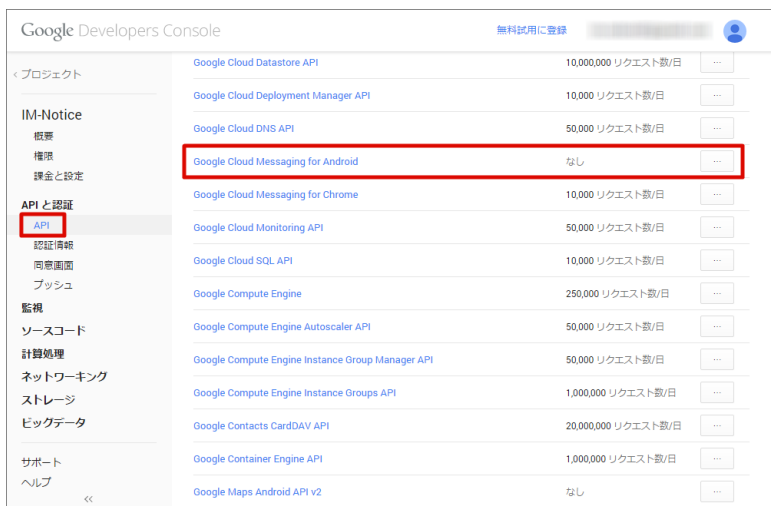
項目	説明
プロジェクト名	任意の名前を入力してください。
プロジェクトID	任意のIDを入力してください。



4. プロジェクトの概要から「プロジェクト番号」を確認することができます。
設定ファイルを編集する際に必要な文字列ですので、控えておいてください。



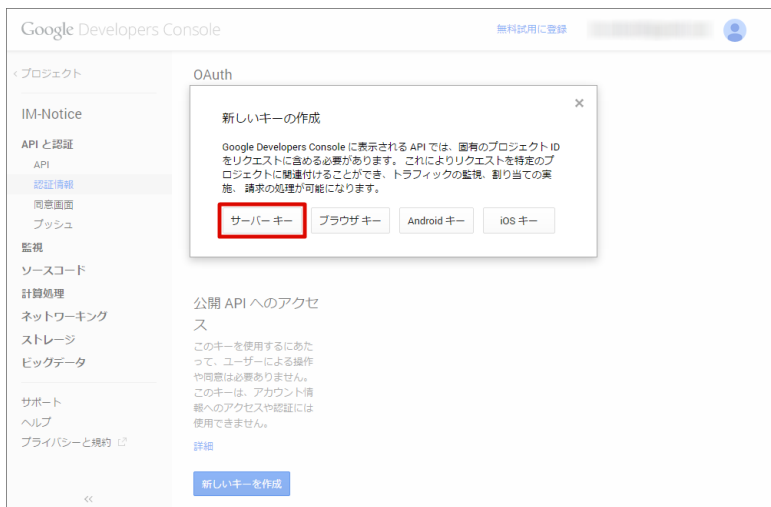
5. 「Google Cloud Messaging for Android」を有効にします。



6. サーバーアプリケーションのキーを作成します。
「新しいキーを作成」をクリックします。



「サーバーキー」を選択し、キーを作成してください。



7. 「APIキー」を確認することができます。
設定ファイルを編集する際に必要な文字列ですので、控えておいてください。



IM-Notice Mobile設定ファイルの編集

IM-Juggling の <(プロジェクト名)/conf> 配下に出力されたim-notice-mobile-config.xmlを開き、以下の設定を行ってください。

- <proxy>
 1. Google Cloud Messaging for Androidへ接続するためにプロキシを使用する場合、設定を行ってください。

```
<proxy enable="true">
  <host></host>
  <port xsi:nil="true"></port> <!-- ポート番号を指定する場合は xsi:nil="true" を削除してください。 -->
  <username></username>
  <password></password>
  <workstation></workstation>
  <domain></domain>
</proxy>
```

- <gcma>

1. <gcma>のname属性に任意の名前を設定してください。

```
<gcma name="gcma">
```

2. <api-key>

- APIキーを設定してください。

3. <project-number>

- プロジェクト番号を設定してください。



コラム

APIキーおよびプロジェクト番号の確認方法は、[GCMAの設定](#)を参照してください。

- <push type="android" />

1. engine属性に、<gcma>のname属性に設定した名前を指定してください。

```
<push type="android" engine="gcma" />
```

- <max-push-subject-length>

1. Push通知受信時に表示するタイトルの最大文字数を設定してください。

```
<max-push-subject-length>30</max-push-subject-length>
```

- <max-push-body-length>

1. Push通知受信時に表示する本文の最長文字数を設定してください。

```
<max-push-body-length>70</max-push-body-length>
```

GCMA+Amazon SNSを使用する場合

GCMAとAmazon SNSを使用しAndroidへの通知を行う場合、以下の設定を行ってください。

GCMA の設定

GCMAの設定を行います。



コラム

GCMAの設定方法は、[GCMAの設定](#)を参照してください。

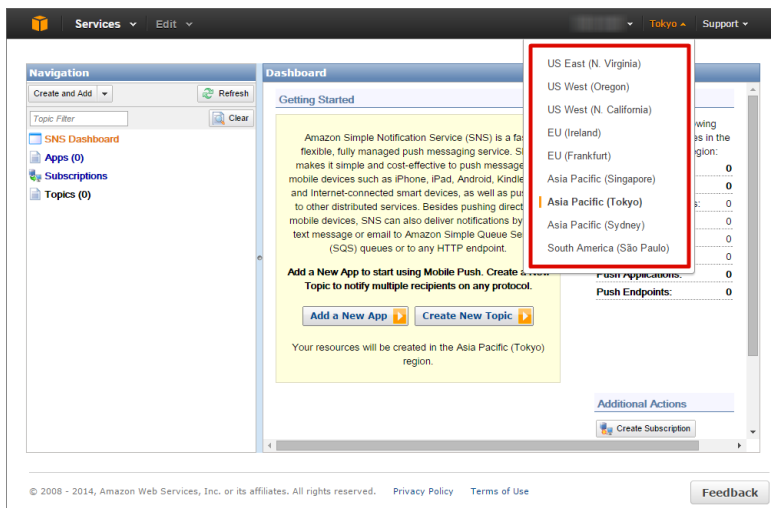
Amazon SNS の設定

Amazon SNSの設定を行います。

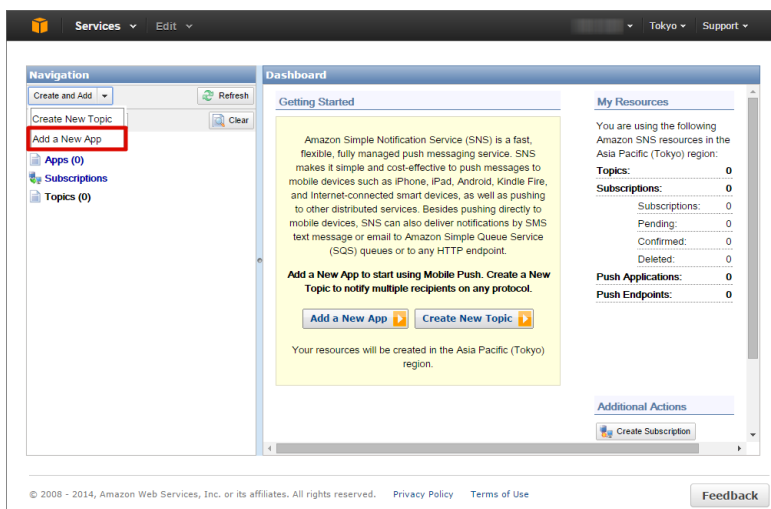
1. [Amazon SNS](#) にサインインしてください。



2. リージョンを選択してください。

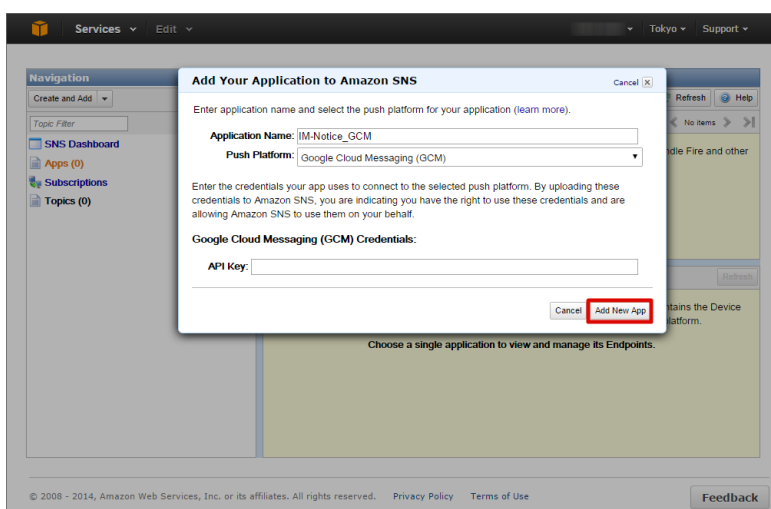


3. 「Create and Add」から「Add a New App」を選択してください。

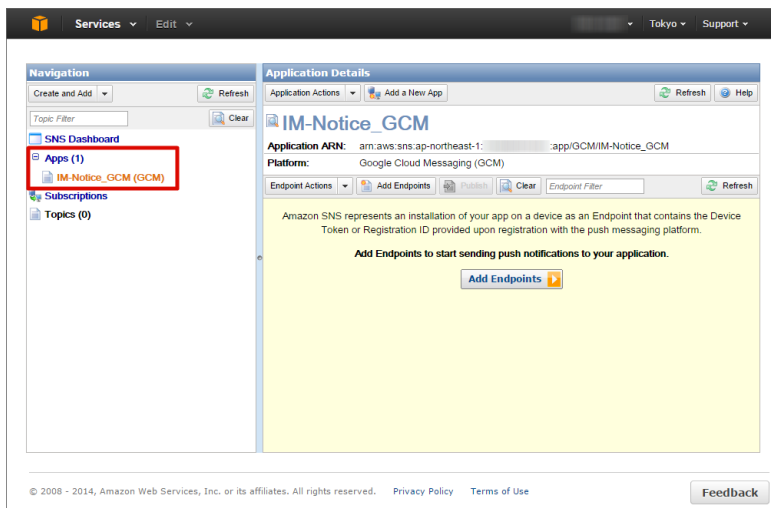


4. フォームに以下の内容を入力し、「Add New App」をクリックしてください。

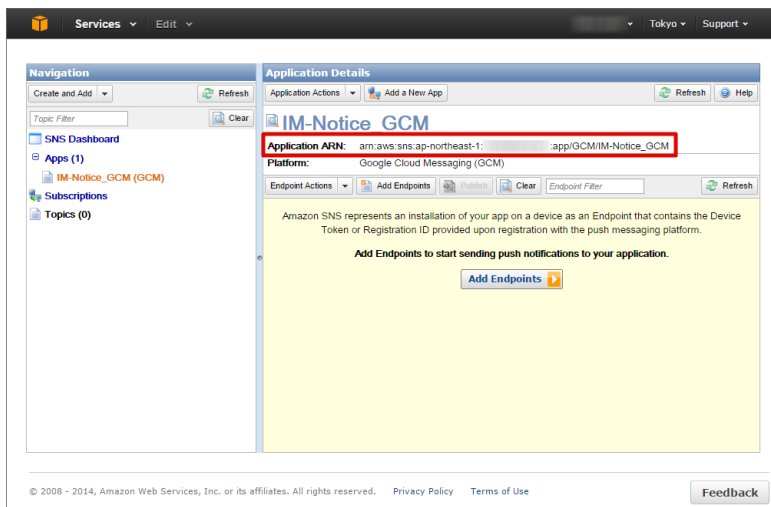
項目	説明
Application Name	任意の名前を入力します。
Push Platform	「Google Cloud Messaging(GCM)」を選択します。
API Key	GCMA の設定 で作成したAPIキーを入力します。



5. 「Apps」にアプリケーションが追加されました。



6. 「Application ARN」を確認することができます。
設定ファイルを編集する際に必要な文字列ですので、控えておいてください。



IM-Notice Mobile設定ファイルの編集

IM-Juggling の <(プロジェクト名)/conf> 配下に出力されたim-notice-mobile-config.xmlを開き、以下の設定を行ってください。

- <proxy>
 1. Google Cloud Messaging for AndroidやAmazon SNSへ接続するためにプロキシを使用する場合、設定を行ってください。

```
<proxy enable="true">
  <host></host>
  <port xsi:nil="true"></port> <!-- ポート番号を指定する場合は xsi:nil="true" を削除してください。 -->
  <username></username>
  <password></password>
  <workstation></workstation>
  <domain></domain>
</proxy>
```

- <asns>
 1. <asns>のname属性に任意の名前を設定してください。

```
<asns name="asns-gcm">
```

2. <endpoint>
 - Amazon SNSのリージョンに対応するEndpointを設定してください。

```
<endpoint>sns.ap-northeast-1.amazonaws.com</endpoint>
```



コラム

Endpointは [AWS documentation - Regions and Endpoints](#) で確認できます。
リージョンが「Asia Pacific (Tokyo)」の場合、Endpointは「sns.ap-northeast-1.amazonaws.com」です。

3. <platform-application-arn>
 - Application ARNを設定してください。



コラム

Application ARNの確認方法は、[Amazon SNS の設定](#) を参照してください。

4. <access-key>
 - アクセスキーを設定してください。
5. <secret-key>
 - シークレットキーを設定してください。

コラム

アクセスキーおよびシークレットキーは [AWS Identity and Access Management \(IAM\)](#) で作成することができます。



6. <gcma-project-number>
 - プロジェクト番号を設定してください。

コラム

プロジェクト番号の確認方法は、[GCMAの設定](#) を参照してください。

- <push type="android" />
 1. engine属性に、<asns>のname属性に設定した名前を指定してください。

```
<push type="android" engine="asns-gcm" />
```

- <max-push-subject-length>
 1. Push通知受信時に表示するタイトルの最大文字数を設定してください。

```
<max-push-subject-length>30</max-push-subject-length>
```

- <max-push-body-length>
 1. Push通知受信時に表示する本文の最長文字数を設定してください。

```
<max-push-body-length>70</max-push-body-length>
```

WARファイルによる複数テナント

intra-mart Accel Platform では、複数のwarファイルをResin上にデプロイすることによりマルチテナント環境を実現します。

マルチテナント環境は、それぞれのテナント環境とリソースを共有しません。

その為、intra-mart Accel Platform の設定はテナント単位でそれぞれ分ける必要があります。

設定を分ける必要がある設定は以下の通りです。

1. クラスタリングID、クラスタリング用ポート番号、及びポートレンジ
[Network](#) ネットワーク設定を参照して下さい。
 クラスタリングID及び、クラスタリング用ポート番号、ポートレンジがテナント間で重複しないよう設定する必要があります。
2. DataSource
[DataSource](#) を参照して下さい。
 テナント毎にデータベース接続先をそれぞれ設定する必要があります。
3. Storage領域
[Storage](#) を参照して下さい。
 テナント毎に利用するStorage領域をそれぞれ設定する必要があります。
4. Apache Cassandra の接続先
[IMBox](#) を参照して下さい。
 テナント毎に接続する Apache Cassandra の接続先、またはkeyspaceをそれぞれ設定する必要があります。
5. Apache Solr の接続先
[IM-ContentsSearch](#) を参照して下さい。
 テナント毎に接続する Apache Solr を設定する必要があります。
6. warファイルの名称
 Resinでは、warファイルの名称を元にコンテキストパスが決定されます。
 マルチテナント環境の場合、warファイルの名称を分ける必要があります。

注意

WARファイルによる複数テナント 環境を構築する場合、単体の場合と比べメモリを消費しますのでご注意ください。
 メモリ使用量の設定に関しては [Resinの設定](#) を参照して下さい。

概要

この章では、バーチャルテナント機能利用時における、操作対象のテナントがどのように解決されるかを説明します。

intra-mart Accel Platform では、アカウントコンテキストのテナントIDプロパティが、操作対象のテナントを意味しています。
アカウントコンテキストのテナントIDプロパティは、テナント解決機能を利用して解決されます。

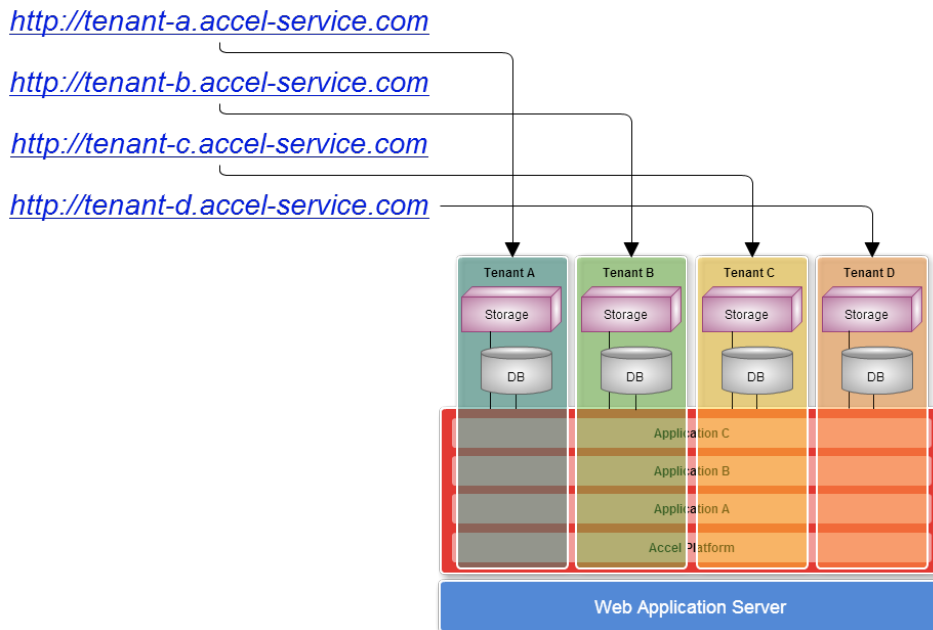
以降では、テナント解決機能の仕様について説明します。

なお、intra-mart Accel Platform 上に作成するプログラムは、操作対象のテナントを意識する必要はありません。
(操作対象のテナントは、このページで説明するテナント解決の仕組みによって自動的に解決されます。)

リクエスト情報を利用したテナント自動解決機能について

intra-mart Accel Platform では、リクエスト情報を利用して操作対象となるテナントを決定することが可能です。
この機能を利用することにより、利用者の認証状態を問わず特定のテナントが操作対象になり、ログイン時のテナントIDの指定が不要となります。

例えば、URLのサブドメインを利用してテナントの自動解決を行うことで、以下のように、URLごとに操作対象となるテナントを決定することが可能です。



i コラム

標準では、リクエスト情報を利用したテナント自動解決機能は利用しません。

リクエスト情報を利用したテナント自動解決機能は、「IM-SecureSingOn for Accel Platform」アプリケーションや、機能を提供するモジュールを利用することで有効になります。

i コラム

リクエスト情報を利用したテナント自動解決機能を実装する方法については、「[リクエスト情報を利用したテナント自動解決機能を提供する](#)」を参照してください。

テナント解決のパターン

Webアクセス時にテナントを解決するパターンは、以下の通りです。

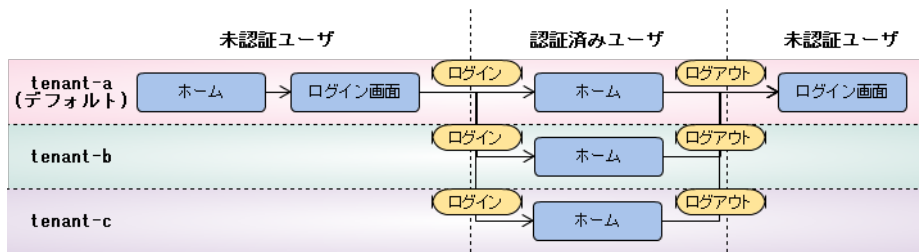
- 一般ユーザ
 - ログイン時にテナントを指定する場合
 - リクエスト情報を利用したテナント自動解決機能を利用する場合
- システム管理者

一般ユーザ

ログイン時にテナントを指定する場合

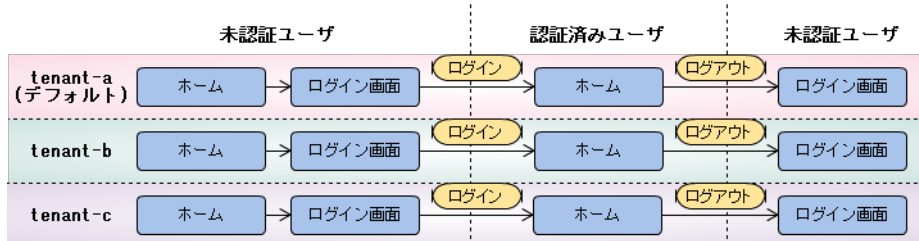
テナントが複数存在する場合は、一般ユーザのログイン画面でテナントを指定してログインを行います。

ログインに成功すると、ログイン画面で指定したテナントが操作対象のテナントとなります。
ログアウトを行うことで、未認証状態となり、デフォルトテナントが操作対象に切り替わります。
未認証状態では、デフォルトテナントが操作対象のテナントとなります。



リクエスト情報を利用したテナント自動解決機能を利用する場合

リクエスト情報を利用したテナント自動解決機能を利用することで、リクエストの任意の情報からテナントの解決を行うことが可能です。
この機能を利用している状態では、ログイン／ログアウトにより操作対象テナントが切り替わることは無く、ログイン画面でテナントを指定してログインを行うことはできません。
未認証状態では、自動解決されたテナントが操作対象のテナントとなります。



システム管理者

システム管理者の場合、操作対象のテナントはデフォルトテナントとなります。
システム管理者は、テナントの切り替え機能を利用して操作対象のテナントを切り替えることが可能です。詳細は「システム管理者操作ガイド」の「[テナントの切り替え](#)」を参照してください。

リクエスト情報を利用したテナント自動解決機能を提供する

概要

リクエスト情報を利用したテナント自動解決機能が有効な場合、操作対象のテナントは、以下の流れで解決されます。

1. 提供されているテナント自動解決機能にて、テナントIDの解決を行います。
2. 提供されているテナント検証機能にて、テナントIDの検証を行います。
3. テナントIDをアカウントコンテキストのテナントIDプロパティに設定します。

以下では、[テナント自動解決機能](#)、[テナント検証機能](#) とその提供方法について説明します。

テナント自動解決機能

リクエスト情報から、テナントIDを解決する機能です。

`RequestBasedTenantIdResolver` インタフェースを実装したクラスを作成し、プラグイン設定ファイルを設定することで、リクエスト情報を利用したテナント自動解決機能を利用することが可能

テナント解決機能は、複数設定して提供することが可能です。
複数のテナント解決機能が提供されている場合は、プラグインの仕様に従い、優先度の高いテナント解決機能から順番に実行されます。
テナントの解決が行われた (`getTenantId(HttpServletRequest)` メソッドで `null` 以外を返した) 時点でテナント解決処理が終了します。後続のテナント解決機能は実行されません。
テナントが解決できない (`getTenantId(HttpServletRequest)` メソッドで `null` を返した) 場合は、後続のテナント解決機能にてテナントの解決を行います。
全てのテナント解決機能がテナントを解決できなかった (全て `null` を返した) 場合、次項の [テナント検証機能](#) の検証で問題がなかった場合は、デフォルトテナントと解決されます。

テナント自動解決機能の作成

以下のインタフェースを実装したクラスを作成します。

- `jp.co.intra_mart.foundation.admin.tenant.context.RequestBasedTenantIdResolver`

以下はCookieを利用してテナントを解決するサンプル実装です。

```

package sample.context;

import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServletRequest;

import jp.co.intra_mart.foundation.admin.tenant.context.RequestBasedTenantIdResolver;

public class CookieTenantIdResolver implements RequestBasedTenantIdResolver {

    private static final String COOKIE_KEY = "X-TENANT-ID";

    @Override
    public String getTenantId(final HttpServletRequest request) {
        return getCookieValue(request, COOKIE_KEY);
    }

    private Cookie getCookie(final HttpServletRequest request, final String name) {
        final Cookie[] cookies = request.getCookies();

        if (cookies == null) {
            return null;
        }

        for (final Cookie cookie : cookies) {
            if (name.equals(cookie.getName())) {
                return cookie;
            }
        }

        return null;
    }

    private String getCookieValue(final HttpServletRequest request, final String name) {
        final Cookie cookie = getCookie(request, name);

        if (cookie == null) {
            return null;
        }

        return cookie.getValue();
    }
}

```



コラム

RequestBasedTenantIdResolver インタフェースの詳細は「[RequestBasedTenantIdResolverインタフェースのAPIリスト](#)」を参照してください。

テナント自動解決機能の設定

テナント自動解決機能はプラグインファイルで設定を行います。従って、プラグインの機能を利用した設定が可能です。

リクエスト情報を利用したテナント自動解決機能を設定するための拡張ポイントは `jp.co.intra_mart.foundation.admin.tenant.context.tenant.resolvers` です。

作成したプラグインファイルを `WEB-INF/plugin/sample_tenant_resolver/plugin.xml` に配置します。(plugin直下のフォルダ名は、他のプラグインと重複しない名前でも任意に変更することが可能ですが、上記のCookieを利用したテナント自動解決機能の実装を動作させるプラグイン設定のサンプルは以下の通りです。

```

<?xml version="1.0" encoding="UTF-8"?>
<plugin>
  <extension point="jp.co.intra_mart.foundation.admin.tenant.context.tenant.resolvers">
    <tenant-id-resolvers
      id="sample.context.cookie_resolver"
      version="1.0.0"
      rank="1">

      <!-- tenant-id-resolver タグは複数指定可能です。 -->
      <!-- RequestBasedTenantIdResolverインタフェースを実装したクラスの完全修飾クラス名をclass属性に指定してください。 -->
      <tenant-id-resolver class="sample.context.CookieTenantIdResolver" />

    </tenant-id-resolvers>
  </extension>
</plugin>

```



コラム

プラグインの設定仕様については、「[PluginManagerのAPIリスト](#)」を参照してください。

上記の資材をデプロイすると、Cookieのキー `X-TENANT-ID` の値をテナントIDとして扱うようになります。

テナント検証機能

テナント自動解決機能により解決したテナントIDが妥当な値であるかを判別する機能です。

テナント自動解決機能により解決したテナントIDは検証機能にてエラーとならない場合、アカウントコンテキストのテナントIDプロパティに設定されます。例えば、存在しないテナントIDがアカウントのテナントIDプロパティに設定されてしまうと intra-mart Accel Platform が提供する機能が正常に動作しなくなります。このような状態を回避するために、テナント検証機能にて妥当性を検証します。

テナント検証機能は、複数設定して提供することが可能です。

複数のテナント検証機能が提供されている場合は、プラグインの仕様に従い、優先度の高いテナント検証機能から順番に実行されます。

テナントの検証にてエラー（`validate(String, Resource)` メソッドで `InvalidTenantIdException` ）が発生した時点でテナント検証処理が終了します。後続のテナント検証機能は実行されません。

テナント検証機能の作成

以下のインタフェースを実装したクラスを作成します。

- `jp.co.intra_mart.foundation.admin.tenant.context.TenantIdValidator`

以下にサンプル実装を例示します。

```
package jp.co.intra_mart.foundation.admin.tenant;

import jp.co.intra_mart.foundation.admin.tenant.context.TenantIdValidator;
import jp.co.intra_mart.foundation.context.core.Resource;
import jp.co.intra_mart.foundation.admin.tenant.InvalidTenantIdException;

public class SampleTenantIdValidator implements TenantIdValidator {

    @Override
    public void validate(final String tenantId, final Resource resource) throws InvalidTenantIdException {

        if (tenantId != null && !tenantId.startsWith("TENANT_")) {
            // テナントID書式チェック
            // テナントIDが "TENANT_" で始まっていることをチェックします。
            throw new InvalidTenantIdException("テナントIDの書式が正しくありません。");
        }

    }

}
```

コラム

`TenantIdValidator` インタフェースの詳細は「[TenantIdValidatorインタフェースのAPIリスト](#)」を参照してください。

テナント検証機能の設定

テナント検証機能も、テナント自動解決機能と同様にプラグイン設定ファイルで設定を行います。

テナント検証機能を設定するための拡張ポイントは `jp.co.intra_mart.foundation.admin.tenant.context.tenant.validators` です。

以下にサンプルのプラグイン設定ファイルを例示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin>
  <extension point="jp.co.intra_mart.foundation.admin.tenant.context.tenant.validators">
    <tenant-id-validators>
      id="sample.context.sample_tenant_id_validator"
      version="1.0.0"
      rank="1">

      <!-- tenant-id-validator タグは複数指定可能です。 -->
      <!-- TenantIdValidatorインタフェースを実装したクラスの完全修飾クラス名をclass属性に指定してください。 -->
      <tenant-id-validator class="sample.context.SampleTenantIdValidator" />

    </tenant-id-validators>
  </extension>
</plugin>
```

StandardTenantIdValidator

標準で提供されている `TenantIdValidator` の実装クラスです。

テナント自動解決機能によるテナント解決チェック、テナントの存在チェックなどを行います。

完全クラス修飾子

`jp.co.intra_mart.system.admin.context.StandardTenantIdValidator`

プラグインの初期化パラメータ

`StandardTenantIdValidator` は、`PropertyInitParamable` インタフェースを実装しているため、プラグイン設定ファイルにて初期化パラメータを受け渡すことが可能です。

初期化パラメータは `<init-param>` タグで指定します。

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin>
  <extension point="jp.co.intra_mart.foundation.admin.tenant.context.tenant.validators">
    <tenant-id-validators>
      id="jp.co.intra_mart.tenant_id.validator.standard"
      version="8.0.7"
      rank="100">
        <tenant-id-validator class="jp.co.intra_mart.system.admin.context.StandardTenantIdValidator">

          <init-param>
            <param-name>required_tenant_id</param-name>
            <param-value>true</param-value>
          </init-param>
          <init-param>
            <param-name>valid_tenant_id</param-name>
            <param-value>true</param-value>
          </init-param>

        </tenant-id-validator>
      </tenant-id-validators>
    </extension>
  </plugin>
```

param-name	説明
required_tenant_id	テナントIDが解決されているかを確認します。 <code>true</code> / <code>false</code> で指定し、 <code>true</code> の場合は確認を行います。 テナントIDが解決されていない場合は、例外 <code>InvalidTenantIdException</code> が発生し、システムにアクセスできません。 省略した場合は、確認を行いません。
valid_tenant_id	解決されたテナントIDが存在するテナントのテナントIDであるかを確認します。 <code>true</code> / <code>false</code> で指定し、 <code>true</code> の場合は確認を行います。 解決されたテナントIDが存在するテナントのテナントIDではない場合は、例外 <code>InvalidTenantIdException</code> が発生し、システムにアクセスできません。 省略した場合は、確認を行いません。



コラム

PropertyInitParamable インタフェースの詳細は「[PropertyInitParamableインタフェースのAPIリスト](#)」を参照してください。

ポート一覧

- intra-mart Accel Platform で利用するポート番号について説明します。

項目

- Resin
 - HTTP通信ポート
 - HTTPS通信ポート
 - クラスタポート(トライアドサーバ)
 - クラスタポート(ダイナミックサーバ)
 - watchdogポート
- JGroup
- Apache Cassandra
 - クライアント接続ポート
 - ノード間通信ポート
 - ノード間通信ポート (ノード間通信にSSLを利用した時)
 - JMX接続ポート
- Apache Solr
- IM-Notice

Resin

- Resin は以下の機能のポートを使用します。
 - HTTP(HTTPS)ポート
 - クラスタポート



注意

クラスタポートはトライアドサーバ、ダイナミックサーバで設定方法が異なります。

- トライアドサーバ
resin.propertiesファイルを編集
- ダイナミックサーバ
コマンドライン引数を指定

- watchdogポート

HTTP通信ポート

- ResinがHTTP通信を行うためのポートの設定です。

- 設定項目

デフォルトのポート番号	8080
デフォルトの有効/無効	有効
設定ファイル	resin.properties
プロパティ	app.http

- 設定例

全てのサーバで同じポートを使用する場合は「**app.http**」をひとつ記述してください。

```
app.http      : 8080
```

サーバ毎に使用するポートを指定する場合は「**[サーバID].http**」を記述します。
下記の記述例では「**app-0**」のResinは「8081」を使用しますが、それ以外のResinは「8080」を使用します。

```
app.http      : 8080
app-0.http    : 8081
```

HTTPS通信ポート

- ResinがHTTPS通信を行うためのポートの設定です。

- 設定項目

デフォルトのポート番号	8443
デフォルトの有効/無効	無効
設定ファイル	resin.properties
プロパティ	app.https

- 設定例

全てのサーバで同じポートを使用する場合は「**app.https**」をひとつ記述してください。

```
app.https     : 8443
```

サーバ毎に使用するポートを指定する場合は「**[サーバID].https**」を記述します。
下記の記述例では「**app-0**」のResinは「8444」を使用しますが、それ以外のResinは「8443」を使用します。

```
app.https     : 8443
app-0.https   : 8444
```

クラスタポート(トライアドサーバ)

- Resinがクラスタリングを行うためのポートの設定です。

- 設定項目

デフォルトのポート番号	6800
デフォルトの有効/無効	有効
設定ファイル	resin.properties
プロパティ	app_servers

- 設定例

デフォルトのポート番号を指定する場合は「**app_servers**」にIPアドレスのみを記述してください。

```
app_servers   : 192.168.100.100 192.168.100.101 192.168.100.102
```

任意のポート番号を使用する場合は「**[IPアドレス]:[ポート番号]**」と記述してください。

```
app_servers   : 192.168.100.100:6801 192.168.100.101:6802 192.168.100.102:6803
```

クラスタポート(ダイナミックサーバ)

- Resinがクラスタリングを行うためのポートの設定です。

- 設定項目

デフォルトのポート番号	6830
コマンドライン引数	--elastic-server-port

- 設定例

ポート番号を指定しない起動コマンドではデフォルトのポート番号を使用します。

```
> resin.exe --elastic-server --cluster app console
```

任意のポート番号を使用する場合は「-elastic-server-port」を起動コマンドに指定してください。

```
> resin.exe --elastic-server --cluster app --elastic-server-port 6831 console
```

watchdogポート

- 設定項目

デフォルトのポート番号	6600
-------------	------

デフォルトの有効/無効	有効
-------------	----

設定ファイル	resin.xml
--------	-----------

タグ	<watchdog-port>
----	-----------------

- 設定例

```
<watchdog-port>6700</watchdog-port>
```

JGroup

- JGroups はサーバ間通信として利用します。

デフォルトのポート番号	5200-5202
-------------	-----------

Apache Cassandra

- Apache Cassandra は以下の機能のポートを使用します。
 - クライアント接続ポート
 - ノード間通信ポート
 - ノード間通信ポート (ノード間通信にSSLを利用した時)
 - JMX接続ポート



コラム

詳細は、「[Cassandra管理者ガイド](#)」を参照してください。

クライアント接続ポート

デフォルトのポート番号	9160
-------------	------

ノード間通信ポート

デフォルトのポート番号	7000
-------------	------

ノード間通信ポート (ノード間通信にSSLを利用した時)

デフォルトのポート番号	7001
-------------	------

JMX接続ポート

デフォルトのポート番号	7199
-------------	------

Apache Solr

- 「[Apache Solr](#)」にて構築したアプリケーションサーバによって異なります。

デフォルトのポート番号	アプリケーションサーバ「Jetty」	8983
-------------	--------------------	------

	アプリケーションサーバ「Resin」	8080
--	--------------------	------

	アプリケーションサーバ「Tomcat」	8080
--	---------------------	------

IM-Notice

- IM-Noticeが通知を行うためのポートの設定です。

デフォルトのポート番号	40608
-------------	-------

ブローカー利用時	40608, 40609
----------	--------------

IM-Juggling の応用

IM-Juggling を利用せず、CUIでWARファイルを作成する方法

intra-mart Accel Platform は IM-Juggling を利用してWARファイルを作成しますが、このWARファイルの作成をCUIで行うことが可能です。

IM-Juggling をインストールしたディレクトリのpluginsディレクトリ配下に下記形式のディレクトリが存在します。

```
jp.co.intra_mart.flashcat.ant_1.0.0.XXXXXXXXXXX/ant
```

※XXXXXXXXXXXX はタイムスタンプが入ります。
※自動更新適用後の場合、複数ディレクトリが存在することがあります。
複数存在する場合は最新のタイムスタンプのディレクトリをご利用ください。

ディレクトリ内の構造は下記の通りです。

```
jp.co.intra_mart.flashcat.ant_1.0.0.XXXXXXXXXXX/ant
├── build.xml // WAR作成用のantスクリプト
└── lib // WAR作成に必要なライブラリ群
    ├── commons-compress-1.3.jar
    ├── im_commons-8.0.3.jar
    ├── im_core_base-8.0.3.jar
    ├── im_j2ee_assist-8.0.2.jar
    ├── im_j2ee_assist_base-8.0.2.jar
    ├── im_j2ee_assist_impl-8.0.2.jar
    ├── im_jackling_core-0.0.2.jar
    ├── im_jackling_toolkit-0.0.2.jar
    ├── im_jdk_assist-8.0.2.jar
    ├── im_juggling_ant-1.0.0.jar
    ├── im_modules-8.0.5.jar
    ├── javassist-3.7.ga.jar
    ├── ognl-2.7.3.jar
    ├── slf4j-api-1.6.6.jar
    └── xalan-2.7.0.jar
```

このantディレクトリ直下に、projectという名称のディレクトリを作成し、その中に IM-Juggling プロジェクトのファイル群を配置します。その後、コマンドラインから 以下のantコマンドを実行する WARファイルが作成されます。

```
ant clean make
```

また、コマンドラインから 以下のantコマンドを実行することによりWARファイルとあわせて静的ファイルも作成することができます。

```
ant clean make static
```



注意

CUIでWARファイルを作成する際、使用許諾の同意確認が表示されません。
このままWARファイルを利用された場合、使用許諾は強制的に同意されたものとなります。予めご了承ください。

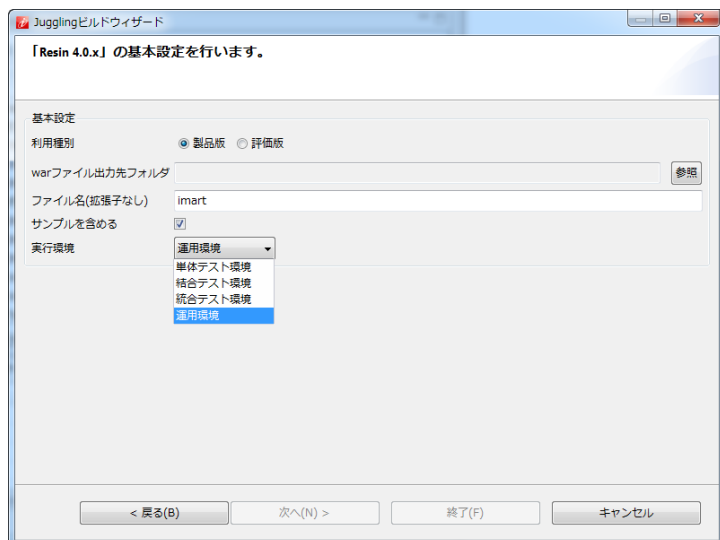
WARファイル作成時の実行環境の変更

IM-Juggling でWARファイルを作成する際、以下の画面で「実行環境」項目を選択する事ができます。

項目

- 概要
- 環境種別毎における各モジュールの動作仕様
 - アクセスコンテキスト情報をログに出力します。
 - Debug#browse 画面にコールスタック情報を表示します。
 - すべてのスクリプト開発モデルのプログラムをインタプリタモードで実行します。
 - 同一パスに対して異なる認可設定が存在している場合に警告ログを出力します。
 - デバッグ情報を付与したキークラスを生成します。
 - CertificationServletUtil#validateCertificationConfirm()がtrueを返却します。
 - 検索実行エラー発生時に、開発者向けの詳細メッセージを含めて出力します。
 - 設定ファイル読み込み時に、Instance.PROTOTYPE を渡します。
 - Routerを提供するためのFactoryを提供します。
 - script type="text/javascript" タグのラッパータグファイルを最小化します。
 - link rel="stylesheet" タグのラッパータグファイルを最小化します。

概要



「実行環境」項目の主な違いは次の通りです。

種別	説明
単体テスト環境 (UT)	JUnitなどを利用する場合を想定しています。パフォーマンスに関しては考慮しません。
結合テスト環境 (CT)	IDEを使用したテスト環境などを想定しています。一般的に最も利用される環境を想定しています。必要最低限の動作環境です。
統合テスト環境 (IT)	本来の実行環境 (APサーバ) を使用した統合テスト環境を想定しています。カットオーバー直後等でもこの環境を利用することを想定しています。
運用環境 (PRODUCT)	パフォーマンスを考慮した動作環境です。

注意

デバッグ情報の出力処理やキャッシュで生成される情報と違い、レスポンス面やエラー画面の表示などで大きな違いが発生します。本番の商用環境や運用環境では必ず、実行環境を「運用環境」に選択して、WARファイルを作成することを推奨します。

コラム

「実行環境」項目の設定情報は、WEB-INF/classes/im_env.txtに以下のような文字列で定義されています。

```
単体テスト環境:ut
結合テスト環境:si
総合テスト環境:pt
運用環境:product
```

WARファイルをデプロイした後に一時的に実行環境を変更したい場合は、上記ファイルの実行環境に関する文字列に変更して再起動することで、WARファイル作成時と違う実行環境で動作させることもできますが、非推奨となります。

環境種別毎における各モジュールの動作仕様

環境種別毎のモジュールの動作仕様については次の通りです。

環境種別	対象モジュール	概要
------	---------	----

環境種別	対象モジュール	概要
単体テスト環境 (UT)	コアモジュール	「 アクセスコンテキスト情報をログに出力します。 」
	スクリプト開発モデル	「 Debug#browse 画面にコールスタック情報を表示します。」
	スクリプト開発モデル	「 すべてのスクリプト開発モデルのプログラムをインタプリタモードで実行します。 」
	テナント管理機能	「 同一パスに対して異なる認可設定が存在している場合に警告ログを出力します。 」
	テナント管理機能	「 デバッグ情報を付与したキークラスを生成します。 」
	認証機能	「 CertificationServletUtil#validateCertificationConfirm()がtrueを返却します。 」
	IM Contents Serach	「 検索実行エラー発生時に、開発者向けの詳細メッセージを含めて出力します。 」
結合テスト環境 (CT)	IM Contents Serach	「 設定ファイル読み込み時に、Instance.PROTOTYPE を渡します。 」
	コアモジュール	「 アクセスコンテキスト情報をログに出力します。 」
	スクリプト開発モデル	「 Debug#browse 画面にコールスタック情報を表示します。」
	スクリプト開発モデル	「 すべてのスクリプト開発モデルのプログラムをインタプリタモードで実行します。 」
総合テスト環境 (IT)	ルーティング機能	「 Routerを提供するためのFactoryを提供します。 」
	コアモジュール	「 アクセスコンテキスト情報をログに出力します。 」
運用テスト環境 (PRODUCT)	ルーティング機能	「 Routerを提供するためのFactoryを提供します。 」
	UI基本モジュール	「 script type="text/javascript" タグのラッパータグファイルを最小化します。 」
	UI基本モジュール	「 link rel="stylesheet" タグのラッパータグファイルを最小化します。 」

アクセスコンテキスト情報をログに出力します。

キャッシュされたアクセスコンテキストの取得、および、アクセスコンテキストをキャッシュに格納する際、アクセスコンテキスト情報を ライフサイクル共有情報に格納し、ログを出力します。
アクセスコンテキスト情報とは、以下の2点です。

- ・アクセスコンテキスト種別
- ・有効判定の条件 (実装ごとに出力される内容は異なります)
ログ出力するには、jp.co.intra_mart.system.context.impl.cache.SessionContextCachePolicy のログレベルを TRACE にする必要があります。

<http://localhost:8080/imart/sample/tutorial/edit> にアクセスした際のログ

```
[TRACE] j.c.i.s.c.i.c.SessionContextCachePolicy - [T.IWP.NO_CODE] Use Cach for interface jp.co.intra_mart.foundation.context.model.ClientContext (Condition = Use Cache: false)
[TRACE] j.c.i.s.c.i.c.SessionContextCachePolicy - [T.IWP.NO_CODE] Use Cach for interface jp.co.intra_mart.foundation.context.model.AccountContext (Condition = Use Cache: Thu J
[TRACE] j.c.i.s.c.i.c.SessionContextCachePolicy - [T.IWP.NO_CODE] Use Cach for interface jp.co.intra_mart.foundation.user_context.model.UserContext (Condition = Use Cache: Thu
[TRACE] j.c.i.s.c.i.c.SessionContextCachePolicy - [T.IWP.NO_CODE] Use Cach for interface jp.co.intra_mart.foundation.authz.context.AuthzSubjectContext (Condition = Use Cache: TI
[TRACE] j.c.i.s.c.i.c.SessionContextCachePolicy - [T.IWP.NO_CODE] Use Cach for interface jp.co.intra_mart.foundation.context.model.ClientContext (Condition = Use Cache: false)
[TRACE] j.c.i.s.c.i.c.SessionContextCachePolicy - [T.IWP.NO_CODE] Use Cach for interface jp.co.intra_mart.foundation.context.model.AccountContext (Condition = Use Cache: Thu J
[TRACE] j.c.i.s.c.i.c.SessionContextCachePolicy - [T.IWP.NO_CODE] Use Cach for interface jp.co.intra_mart.foundation.user_context.model.UserContext (Condition = Use Cache: Thu
[TRACE] j.c.i.s.c.i.c.SessionContextCachePolicy - [T.IWP.NO_CODE] Use Cach for interface jp.co.intra_mart.foundation.authz.context.AuthzSubjectContext (Condition = Use Cache: TI
```

Debug#browse 画面にコールスタック情報を表示します。

スクリプト開発モデルの Debug#browse 画面にコールスタック情報を表示します。
また、Debug#printStackTrace メソッドを利用すると、コンソール上にコールスタック情報を出力します。
このコールスタックの取得機能は、im-jssp-config.xml の trace 設定が有効の場合のみ利用可能です。

すべてのスクリプト開発モデルのプログラムをインタプリタモードで実行します。

すべてのスクリプト開発モデルの HTML と JavaScript をインタプリタモードで実行します。
インタプリタモードではソースコードの変更がサーバーを再起動することなく反映されます。
「単体テスト環境」と「結合テスト環境」以外を選択した場合のモードは [source-config.xml](#) の設定値によって決まります。

同一パスに対して異なる認可設定が存在している場合に警告ログを出力します。

同一パスに対して異なる認可設定がされているかどうかを確認し、異なる認可設定がされているパスが存在した場合、警告ログを出力します。
具体的には、以下の警告ログが出力されます。

```
[W_IWP_ROUTER_AUTHZ_00003] 認可設定が取得できませんでした。path = {0}
[W_IWP_ROUTER_AUTHZ_00004] 同一パスに対して異なる認可URIが設定されています。path = {0} uri1 = {1}, uri2 = {2}
[W_IWP_ROUTER_AUTHZ_00005] 同一パスに対して異なる認可アクションが設定されています。path = {0} action1 = {1}, action2 = {2}
```

デバッグ情報を付与したキークラスを生成します。

認可設定をキャッシュする際のキークラスに対し、デバッグ情報を付与したキークラスを生成します。
その他の場合は必要最低限の情報のみ保持するキークラスが生成されます。

CertificationServletUtil#validateCertificationConfirm()がtrueを返却します。

CertificationServletUtil#validateCertificationConfirm(HttpServletRequest, HttpServletResponse)は、常にtrueを返却します。
これは、認証確認時、遷移先情報(リクエストパラメータ名:im_page_key)は、常に設定されているものとして動作することを意味します。
これにより、認証確認画面に直接アクセスすることを可能とし、認証確認画面自体の動作確認が可能となります。

検索実行エラー発生時に、開発者向けの詳細メッセージを含めて出力します。

検索実行エラー発生時に出力するメッセージに、開発者向けの詳細メッセージを含めて出力します。

設定ファイル読み込み時に、Instance.PROTOTYPE を渡します。

ContentsSearch系の設定ファイル読み込み時に、ConfigurationLoader#load or #loadAll に渡す Instance(enum) を判別し渡します。
環境種別が、**単体テスト(UT)** の場合 : Instance.PROTOTYPE を渡します。
環境種別が、**単体テスト(UT)以外** の場合 : Instance.SINGLETON を渡します。

Routerを提供するためのFactoryを提供します。

実行時間などをログ出力するRouterでラップします。
ログ出力するには、jp.co.intra_mart.system.router.DebugRouter のログレベルを DEBUG にする必要があります。

```
[DEBUG] j.c.i.s.r.DebugRouter - [D.IWP.NO_CODE] dispatch: 148 [msec]
[DEBUG] j.c.i.s.r.DebugRouter - [D.IWP.NO_CODE] dispatch: 7 [msec]
[DEBUG] j.c.i.s.r.DebugRouter - [D.IWP.NO_CODE] no route: /imart/ui/libs/jquery-validation-1.9.0/jquery.validate.js 0 [msec]
[DEBUG] j.c.i.s.r.DebugRouter - [D.IWP.NO_CODE] no route: /imart/ui/theme/im_theme_dropdown_blue/css/images/ui-icons_004276_256x240.png 0 [msec]
```

script type="text/javascript" タグのラッパータグファイルを最小化します。

src に指定された文字列の最後の .js を .min.js に置換して href の文字列として使用します。
「運用環境」以外を選択した場合、src に指定された文字列をそのまま src の文字列として使用します。
※ファイル内のコメントなどの情報を省いた、最少ファイル化を行っています。

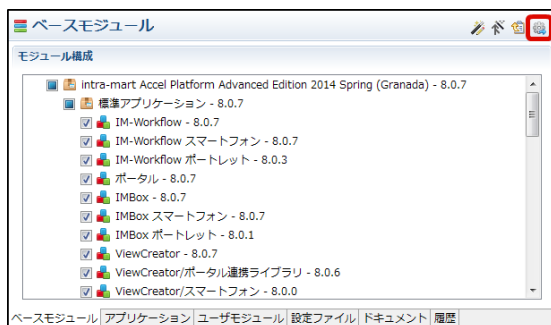
link rel="stylesheet" タグのラッパータグファイルを最小化します。

href に指定された文字列の最後の .css を .min.css に置換して href の文字列として使用します。
「運用環境」以外を選択した場合、href に指定された文字列をそのまま href の文字列として使用します。
※ファイル内のコメントなどの情報を省いた、最少ファイル化を行っています。

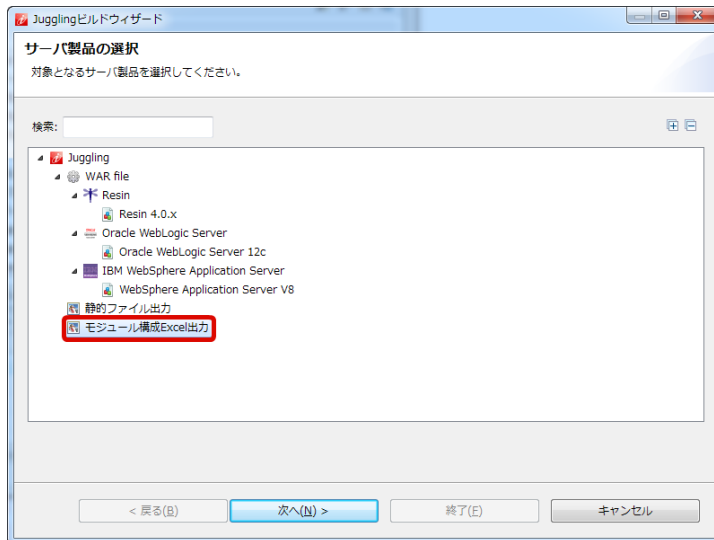
WARファイルに含まれるモジュール情報・ショートモジュールIDの一覧を確認する方法

ユーザアプリケーションとして作成したプログラムをユーザモジュール (immファイル)として作成する場合、
モジュールを識別する、ショートモジュールIDが重複しないようにする必要があります。
ここでは、既存のモジュール内で定義されているショートモジュールIDの一覧を確認する方法を説明します。

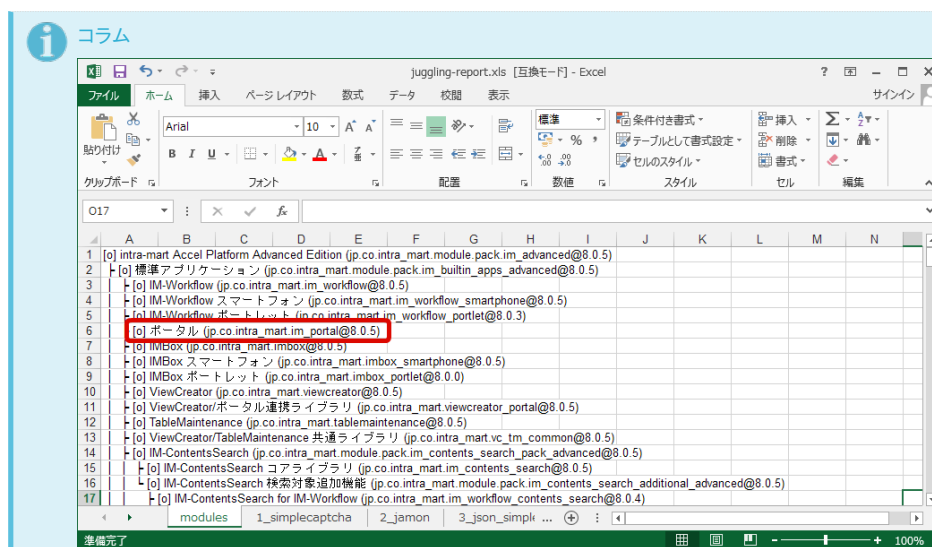
- モジュール構成Excel出力手順
 1. IM-Juggling の「juggling.im」を開き、「ベースモジュール」タブを表示します。



2. 画面右上のビルドウィザードを起動し、「モジュール構成Excel出力」を選択しウィザードに沿ってファイルを出力します。



3. 出力されたExcelファイルを開くと、次のようにモジュール情報を確認する事ができます。



ポータルモジュールの場合、次の内容を確認する事ができます。

・ポータル (jp.co.intra_mart.im_portal@8.0.5)
 モジュール名:ポータル
 モジュールID:jp.co.intra_mart.im_portal
 ショートモジュールID:im_portal
 モジュールバージョン:8.0.5

intra-mart Accel Platform のチューニング

- intra-mart Accel Platform を最適な運用環境にチューニングするための負荷試験のポイント、ノウハウについて説明します。

負荷試験実施の際の注意点

- intra-mart Accel Platform で行う負荷試験についての観点、注意点を以下に記載します。

項目

- 負荷試験の目的・性能の目標値の設定
- 負荷試験を実施する観点
- 多重アクセスのシナリオ
- スレッド数等の各設定値
- GCのチューニング
- キャッシュの設定
- Resin のNative機能のコンパイル
- GZip圧縮の検討
- 静的ファイルはAPサーバではなくWEBサーバに配置
- データベース 側のチューニング
- PreparedStatementCache
- VisualVM等を利用したプロファイリング
- 負荷試験の繰り返し実施

負荷試験の目的・性能の目標値の設定

負荷試験の目的や性能の目標値(どのような環境でどのくらいの性能を出したいのか)をはじめに設定します。本番環境と同等の検証環境で負荷試験を行う場合や、本番環境の何分の1かでの結果から本番環境での性能が類推比較できる環境であれば問題ありませんが、例えば、ノートPC構築して負荷試験を行った場合、その負荷試験の結果を確認しても本番環境での性能値の参考にはなりません。必ず負荷試験の目的や性能の目標値を明確にしてから、負荷試験環境やその実施内容を検討する必要があります。

負荷試験を実施する観点

どのような観点で何を調査したいのかを明確にする必要があります。観点を例としては以下があります。

- サーバサイドでのレスポンス
- クライアントサイドでのレスポンス
- サーバサイドでのメモリ消費状況
- 自作アプリケーションのパフォーマンス
- データベースにかかる負荷状況

また、一度の負荷検証で抽出した観点をすべて確認するという事は、データを取得するだけでも時間を要します。特にJMeterなどのオープンソース系の負荷試験ツールで負荷試験と、同時にサーバリソース状況を取得するのは困難となります。データ取得後のデータ解析には工数が掛かりますので、予めご注意ください。

多重アクセスのシナリオ

負荷試験のシナリオとしてよく見受けられるケースが、全リクエストを同時に実行させるようなシナリオを作成・実施するケースです。このような負荷は、実際の運用ではまず発生しません。上記シナリオで取得した数値をもとにサイジング等を実施したとしても、実際の運用ではオーバースペックのサーバ構成となったり、反対にレスポンスが出ないという問題が発生したりします。シナリオを作成する場合は、多重アクセスの観点以外にも実際の運用に近い負荷を再現できるように、シンクタイム(思考時間遅延)等をシナリオに組み込んで作成・実施を行う検討が必要となります。

スレッド数等の各設定値

Resin やintra-martのスレッド数等の各設定値を極端に大きく設定すれば良いという事はありません。想定される利用者数等を考慮した上で値を設定することが重要となります。設定値は1回設定したら完了ということではなく、幾度も設定値を変更しながら負荷試験を実施し環境に応じた最適値を見つける必要があります。

GCのチューニング

G1GCを使用する、ConcurrentGCを使用する等、どの領域に何を置きたいのかを考えると、適切な値を見つけることは非常に難しい作業となります。特に、WebSphereならGCポリシーを変更するだけで大幅にメモリの使い方が変わります。VisualVM等を利用して、プロファイリングやGCログでの精査が必要となります。

キャッシュの設定

「[intra-mart Accel Platform の設定ファイル](#)」-「[キャッシュ設定](#)」、および、「[認可仕様書](#)」を基に適切な値を算出して設定してください。メモリを考慮しない場合、キャッシュ容量ベースのLRUキャッシュより、数ベースのキャッシュのほうが効果的です。

Resin のnative機能のコンパイル

Resin はFile/IO、Socket/IOの最適化の為に一部nativeな機能を利用しています。Linux環境の場合、「./configure --prefix=pwd' && make && sudo make install」でのインストールを行ってください。resin.properties にある「sendfile」や「tcp_cork」の設定は native 向けの設定です。

GZip圧縮の検討

WEBサーバを構築して運用を行う場合、GZip圧縮の検討を行ないます。なお負荷試験でGZip圧縮を行う場合、負荷をかけるクライアント側が送信するリクエストのヘッダに「Accept-Encoding:gzip, deflate」を入れてください。

静的ファイルはAPサーバではなくWEBサーバに配置

Resin で静的ファイルの処理をさせることは可能ですが、処理を行わせると静的ファイル処理分のスレッドを消費し、本来必要となる処理が回せなくなる可能性もあります。小規模サイトで無い限り、静的ファイルはWEBサーバに配置する事を推奨します。WEBサーバは、Apache HTTP Server を推奨します。

データベース 側のチューニング

データベース のチューニングは非常に重要となります。必ず負荷試験を兼ねてチューニングを行う必要があります。チューニングを実施しない場合(例. PostgreSQL を入れただけで shared_buffers がデフォルト設定のままである等)、月末等の処理集中時にパフォーマンス問題を引き起こす事例かまた、**Oracle Database Appliance(ODA)** や **SQL Server SSD Appliance** などのアプライアンスサーバを導入する案もあります。

PreparedStatementCache

テナント環境セットアップ完了後であれば、適切な値を設定してください。Resin のprepared-statement-cacheはLRUで管理していますので、ある程度大きく設定を行います。

VisualVM等を利用してのプロファイリング

GCのチューニング以外でも、CPUの使用時間、メソッドの呼び出し回数等について、VisualVM等を利用してのプロファイリングを行って確認を行います。

負荷試験の繰り返し実施

1回負荷試験を行っただけでは、レスポンスが期待する結果になる事は極めて低いと考えられます。結果を基にレスポンス等を改善する方法を検討し、設定と負荷試験の再試行を繰り返す必要があります。CPUの処理速度に依存したり、メモリやサイズだけではなくネットワークトラフィック等の影響も発生する事があります。さらにディスクIOの影響などあらゆる要因があるため、さまざまな視点で検証・分析を実施することが重要となります。



コラム

以上に関しては、弊社のコンサルティングサービスをご利用頂く事も可能です。詳細は担当営業までお問い合わせください。

負荷試験を実施する際の各種設定

- intra-mart Accel Platform 上で負荷試験を実施する際に、設定の変更が必要となる項目について説明します。

項目

- Resin のJVMオプション設定
- 任意のユーザでログインを行うための設定
- jsspRpc、非推奨であるformタグを利用した業務画面の試験を行うための設定
- 多重アクセスを想定した試験を行うための設定

Resin のJVMオプション設定

Javaのヒープサイズ等のJVMのオプション設定を変更してください。

設定例(MaxHeapSize・MaxPermSizeを設定及びオプション設定)
jvm_args : -Xmx2048m -XX:MaxPermSize=512m



注意

Linux系OSで、JVMオプションでtempディレクトリの指定を行わない場合、「/tmp」が利用され、cron等の設定により、定期的に「/tmp」配下の内容が削除される設定が標準化されている場合があります。Resin では、作業用ディレクトリに展開した設定ファイル等の変更を検出した際に自動的に再起動が行われる場合があり、回避するためには下記のような設定を行います。

```
jvm_args : -Xmx2048m -XX:MaxPermSize=512m -
Djava.io.tmpdir=/var/resin-tmp
```

※-Djava.io.tmpdirに指定したディレクトリは事前に作成し、権限設定をお願いします。

任意のユーザでログインを行うための設定

セキュア・トークン設定を無効化した上で、「`http://ホスト名:ポート番号/コンテキストパス/certification`」にパラメータを指定してアクセスすることで任意のユーザで認証することが可能。設定方法はWEB-INF/conf/token-filtering-target-config/im_certification.xml の「/certification」をコメントアウトします。詳細は以下の通りです。

WEB-INF/conf/token-filtering-target-config/im_certification.xml

```
<p:entry url="/certification">
↓
<!-- <p:entry url="/certification" /> -->
```

これで以下のURLでログイン可能です。

```
http://ホスト名:ポート番号/コンテキストパス/certification?im_user=(ユーザ名)&im_password=(パスワード)
```

また、「&im_url」属性を付加することにより自動ログイン後に任意のアプリケーションの画面に遷移させることも可能です。

```
http://ホスト名:ポート番号/コンテキストパス/certification?im_user=(ユーザ名)&im_password=(パスワード)&im_url=(ログイン後に表示したいパス)
```



注意

デフォルトで無効化されているとおり、この設定を変更することでURL回数やPOSTリクエストでログイン可能でかつ、任意のアプリを実行させることが可能になりますので、ご利用の場合はセキュリティ観点で問題が無いかを十分に確認の上、ご利用ください。

jsspRpc、非推奨であるformタグを利用した業務画面の試験を行うための設定

jsspRpcは、通常セッションを利用したトークンチェックが都度行われるため、ログインを行うごとにトークンチェックに必要なパラメータの値が変更します。このため、負荷試験のシナリオのリクエストのパラメータを引き渡してもエラーとなってしまいます。回避するにはWEB-INF/web.xmlを編集し、SecureJSPServlet、および、jsspRpcServletのサーブレットパラメータ"security"の値を"false"に変更します。

WEB-INF/web.xml

```

<servlet>
  <servlet-name>SecureJSSPServlet</servlet-name>
  <servlet-class>jp.co.intra_mart.system.servlet.jsp.SecureJSSPServlet</servlet-class>
  <init-param>
    <param-name>security</param-name>
    <param-value>>false</param-value>
  </init-param>
</servlet>
<servlet>
  <servlet-name>jsspRpcServlet</servlet-name>
  <servlet-class>jp.co.intra_mart.system.servlet.jsp.jsspRpcServlet</servlet-class>
  <init-param>
    <param-name>security</param-name>
    <param-value>>false</param-value>
  </init-param>
</servlet>

```

多重アクセスを想定した試験を行うための設定

多重アクセスを想定した試験を行うために、以下の値を設定します。

- リクエスト制御
intra-mart Accel Platform にはリクエストに対して同時に実行可能な数を制限する機能があります。複数のユーザが同時アクセスを行うような試験を行う場合は、この制限に該当してしまう可能性があるため、制限数を調整します。サーバの同時処理の制限を解除するには リクエスト制御設定のthreadタグのmax属性に"0"を指定します。
WEB-INF/conf/request-control-config.xml

```

<request-control-config>
  <thread max="0">
    ...
  </thread>
</request-control-config>

```

リクエスト制御設定については、「[設定ファイルリファレンス](#)」-「[サーバの同時処理の制限](#)」を参照してください。

- データソースへの最大コネクション数
アプリケーションサーバにより、データベースに同時に接続できる数が制限がかかる可能性があります。アプリケーションサーバのデータソース設定にて、同時に接続可能な数が一致することを確認してください。
アプリケーションサーバに Resin を利用する場合は、resin-web.xmlを編集することにより変更可能です。
設定方法・詳細は、「[設定ファイルリファレンス](#)」-「[最大コネクション数設定](#)」を参照してください。
- Resin のスレッド数
Resin のスレッド数を増やす場合は、以下のファイルの該当箇所を変更してください。
<%RESIN_HOME%/conf/resin.properties

```

# Throttle the number of active threads for a port
port_thread_max : 256
accept_thread_max : 32
accept_thread_min : 4
# keepalive_max : 512

```

！ 注意

以上の設定は、負荷試験を行う際に必要となる設定項目となります。本番の運用環境では、必ず環境に応じた適切な設定を行うようにします。

バックアップ・リストア（復元）

！ 注意

お客様の運用環境によって、バックアップ・リストア（復元）の対象、および、作業方法が異なります。

ここで説明する内容は、基本的な内容とその注意点です。実際の作業について、事前に作業の洗い出しと作業手順について十分にご検討頂き、お客様の責任において作業を実施しをお願いします。

作業中に発生したデータ破損及び、データ不整合による運用障害については、弊社では保証致しかねます。予めご了承ください。

- 弊社製品のアップデートやパッチの適用のためにバックアップを取得する場合、各設定ファイルに変更が加わるため、バックアップ対象データのフルバックアップを行ってください。
- 正常なバックアップ・リストア（復元）を行うためには、関連するシステムの静止点が一致している必要があります。そのため、コールドバックアップ（関連するシステムの停止）を推奨します。

i コラム

intra-mart Accel PlatformのStorage領域と、データベースの静止点が異なる場合、整合性が失われ、申請したワークフローが消失するなどの恐れがあります。

項目

- バックアップ対象
 - IM-Juggling関連データ
 - データベース
 - ストレージ
 - Apache Cassandra（IMBoxをご利用の場合）
 - Apache Solr（IM-ContentsSearch for Accel Platformをご利用の場合）
 - ミドルウェア製品
- リストア（復元）手順

バックアップ対象

！ 注意

バーチャルテナントによる各テナントごとのバックアップを取得する場合には、各テナントで使用している、データベーススキーマ、パブリックストレージ、Apache Cassandra、Apache Solrのデータを取得する必要があります。

IM-Juggling関連データ

必要に応じて、下記のデータをバックアップしてください。

- WARファイル
 - IM-Jugglingからエクスポート（出力）したWARファイル
- 静的コンテンツ
 - IM-Jugglingからエクスポート（出力）、またはWeb Serverの %WEB_PATH% ディレクトリ配下に配置した静的コンテンツ
- IM-Juggling上のプロジェクトファイルや設定ファイル
- リポジトリデータ（製品版メディアイメージをご利用の場合）
- 独自に作成したアプリケーションのファイル（プログラムソース、設定ファイル、および、ユーザモジュールのファイル）

データベース

データベースデータ（intra-mart Accel Platformが接続している表領域）をエクスポートします。

詳細については、データベースのマニュアルを参照してください。

対象となるデータベースは以下の通りです。

- システムデータベース（システムのデータを保存するデータベース）
- テナントデータベース（テナント内で利用するデータを保存する、テナントごとのデータベース）
- シェアードデータベース（intra-mart Accel Platform外のデータを保存するデータベース）（利用している場合のみ）

ストレージ

- パブリックストレージ（%PUBLIC_STORAGE_PATH% ディレクトリ 配下のファイル・ディレクトリ）

テナント単位で利用されるストレージです。アップロードされたファイルや利用者間で共有したいファイルを保存する領域です。

- システムストレージ（%STORAGE_PATH%/system ディレクトリ配下のファイル・ディレクトリ）

intra-mart AccelPlatform のシステム内部で利用しているシステム領域です。
システム環境のデータ、各テナントの設定情報等が格納されています。

Apache Cassandra（IMBoxをご利用の場合）

IMBoxの投稿内容の情報です。

詳細は「[Cassandra管理者ガイド](#)」-「[スナップショット](#)」を参照してください。

Apache Solr（IM-ContentsSearch for Accel Platformをご利用の場合）

IM-ContentsSearch for Accel Platformの全文検索用インデックス情報です。

詳細は「[Solr管理者ガイド](#)」-「[Solrのバックアップ](#)」を参照してください。

ミドルウェア製品

上記のファイルやデータを除き、必要に応じてミドルウェア製品をインストールしたサーバのイメージバックアップやインストールディレクトリのバックアップを行ってください。

- データベース
詳細はご利用のデータベースのマニュアルを参照してください。
- Resin
%RESIN_HOME% ディレクトリ配下をバックアップしてください。
- Web Server
詳細はご利用のWeb Serverのマニュアルを参照してください。
- Apache Cassandra
Apache Cassandraをセットアップしたディレクトリ配下をバックアップしてください。
- Apache Solr
Apache Solrを動作しているWeb Application Server(Jetty, Resin, Tomcat)をセットアップしたディレクトリ配下をバックアップしてください。

リストア(復元)手順

！ 注意

リストア(復元)先は、新規のディレクトリ先を対象としてください。
障害発生後の環境(ディレクトリ)へのリストア(復元)は、データ不整合等の事象が発生する可能性があります。

1. どの範囲でリストア(復元)を行うのかを確認します。
 - ミドルウェア製品からリストア(復元)を行う場合
新しいディレクトリにミドルウェア製品を新規に構築します。
ミドルウェア製品によっては以前の環境を完全に削除する必要があります。詳細は各製品のマニュアルを参照してください。
 - intra-mart(プログラム資材、運用データ)のみリストア(復元)を行う場合
バックアップ前と同じミドルウェア環境を利用される場合、既存データが残っているため、各ミドルウェア製品の管理者とリストア先の検討を実施してください(新規スキーマにリ
元)する等)。
バックアップ前と同じディレクトリ先へリストア(復元)する場合は、既存データが削除されている必要があります。
 - アンデプロイの手順は「[WARファイルのアンデプロイ](#)」を参照してください。
 - 運用データについては「[アンインストール](#)」を参照してください。
2. 取得していた各バックアップファイルを再配置します。
 - データベース
バックアップしたエクスポートファイルを、新規で作成したスキーマにインポートします。詳細については、データベースのマニュアルを参照してください。

！ 注意

バーチャルテナントによる複数テナントを構築している場合、システムデータベース内にある各テナントのデータが必要となります。

- Web Server
仮想ディレクトリ(エイリアス)を設定したディレクトリに、バックアップした静的コンテンツを配置します。
 - Storageデータ
バックアップしたStorageデータを配置します。
 - Apache Cassandra
詳細は「[Cassandra管理者ガイド](#)」-「[スナップショットデータによる復旧\(リストア\)](#)」を参照してください。
 - Apache Solr
「[Solr管理者ガイド](#)」-「[Solrのバックアップ](#)」にて、バックアップしたデータをSolrの環境に合わせて配置します。
3. Web Application Serverを起動して、WARファイルのデプロイを行ってください。

アンインストール

- この章では intra-mart Accel Platform に関わる全てのファイル、データのアンインストールを行います。
ミドルウェア製品等のアンインストールについては、同製品のマニュアルをご確認ください。

！ 注意

バーチャルテナントによる複数テナント で作成した任意のテナントを削除する場合は、
「[システム管理者ガイド](#)」-「[テナント情報を更新する](#)」を参照してください。

項目

- [WARファイルのアンデプロイ](#)
- [アプリケーションの削除](#)
- [Storage領域の削除](#)
- [ミドルウェア製品の削除](#)

WARファイルのアンデプロイ

1. 以下のコマンドを実行してください。
Windowsの場合 (コマンド・プロンプトを起動)

```
> cd <%RESIN_HOME%>
> resin undeploy アプリケーション名 (<%RESIN_HOME%>/webapps配下に展開されたWARファイルと同名のディレクトリ名)
```

Linuxの場合

```
# <%RESIN_HOME%>/bin/resinctl undeploy アプリケーション名 (<%RESIN_HOME%>/webapps配下に展開されたwarファイルと同名のディレクトリ名)
```

2. アンデプロイ後、以下のファイルを手動で削除してください。

```
<%RESIN_HOME%>/webapps> 配下のWARファイルと同名のディレクトリ
```



コラム

マルチテナントをご利用の場合は、アンデプロイ対象のWARファイルに対してそれぞれ実行してください。



注意

WARファイルのアンデプロイには、Resin が起動している必要があります。

アプリケーションの削除

1. Resin 上に展開されている以下のファイル・ディレクトリを削除します。

```
<%RESIN_HOME%>/resin-data> 配下のファイル・ディレクトリ
<%RESIN_HOME%>/webapps> 配下のディレクトリ
```



注意

Resinが停止している必要があります。
停止方法については [Web Application Server の起動・停止](#) を参照してください。

Storage領域の削除

1. <%STORAGE_PATH%> 配下のファイル・ディレクトリを削除します。
2. <%PUBLIC_STORAGE_PATH%> 配下のファイル・ディレクトリを削除します。

ミドルウェア製品の削除

- Resin
<%RESIN_HOME%> ディレクトリを削除します。
- データベース
ご利用の データベース のマニュアルを参照してください。
- Web Server
ご利用の Web Server のマニュアルを参照してください。



注意

静的ファイルを配置している <%WEB_PATH%> ディレクトリ配下のファイル・ディレクトリを削除します。

- Apache Cassandra
Apache Cassandra をセットアップしたディレクトリを削除します。
- Apache Solr
Apache Solr を動作している Web Application Server (Jetty , Resin , Tomcat) をセットアップしたディレクトリを削除します。

サンプルデータの投入

サンプルデータをご利用される場合は、以下の手順を実行してください。



注意

テナント環境セットアップの直後に、サンプル・セットアップを実行してください。
サンプル・セットアップ前にユーザやロールなどのマスタデータが変更されていると、
サンプル・セットアップの実行に失敗する可能性があります。

1. システム管理者のメニュー画面を表示します。
メニューから「テナント環境セットアップ」をクリックします。



2. 「サンプルデータセットアップ」をクリックします。



コラム

テナント環境・サンプルデータセットアップの途中で失敗した場合

接続先のデータベースのデータを削除し、再度テナント環境セットアップを行ってください。

データの削除については [アンインストール](#) を参照してください。

データの削除後、Resin を起動し、再度テナント環境セットアップを行ってください。

Apache Cassandra をご利用されている場合は、合わせてデータの削除および起動を行ってください。

セットアップのログについては [テナント環境セットアップ・サンプルデータセットアップに失敗した場合](#) を参照してください。

セットアップで困ったら・・・

- セットアップ中のトラブルシューティングについて次を参考にしてください。

Web Server

.NET Framework のセットアップ

- intra-mart Accel Platform を稼働させる Windows Server では、必要に応じたバージョンの「.NET Framework」をセットアップを行う必要があります。
 - ・ Resin : 「.NET Framework 3.5(.NET 2.0 および3.0を含む)」が必要
 - ・ Internet Information Services (IIS) : 「.NET Framework 4(4.5を含む)」が必要

セットアップ手順

下記のセットアップ手順は、Windows Server 2012 に「.NET Framework 3.5(.NET 2.0 および3.0を含む)」をセットアップする例です。

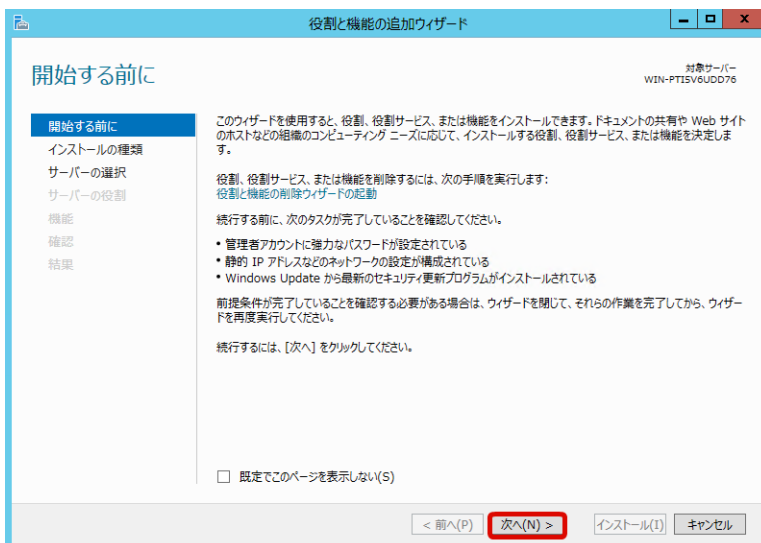
1. Windows Server に付属する「サーバーマネージャー」を開きます。



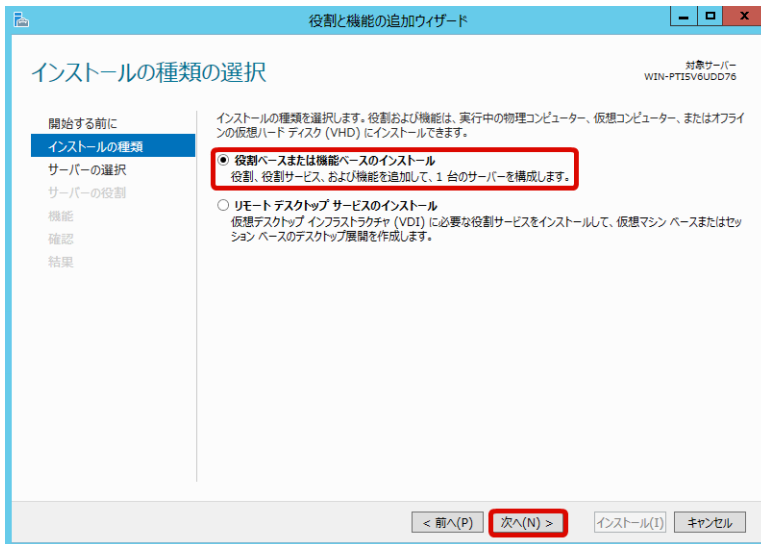
2. 右ペインより「役割と機能の追加」をクリックします。



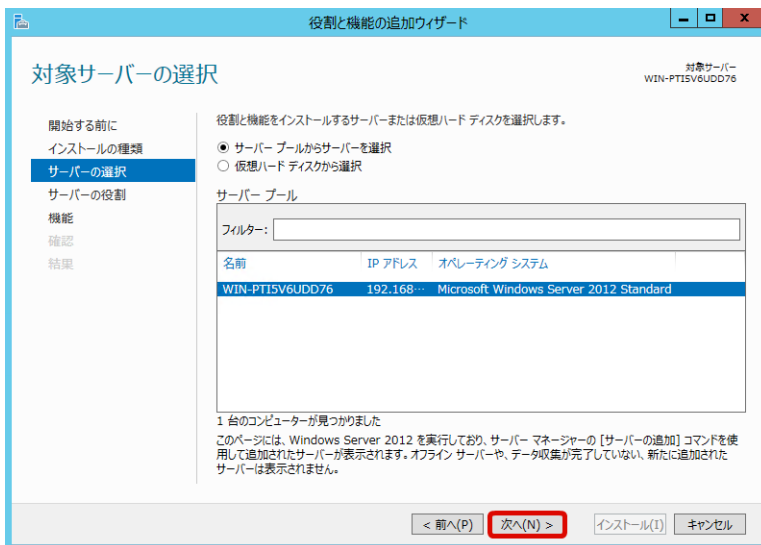
3. 別ウィンドウで表示された「役割と機能の追加ウィザード」に従い、.NET Framework のセットアップを進めます。
→「次へ (N)>」をクリックします。



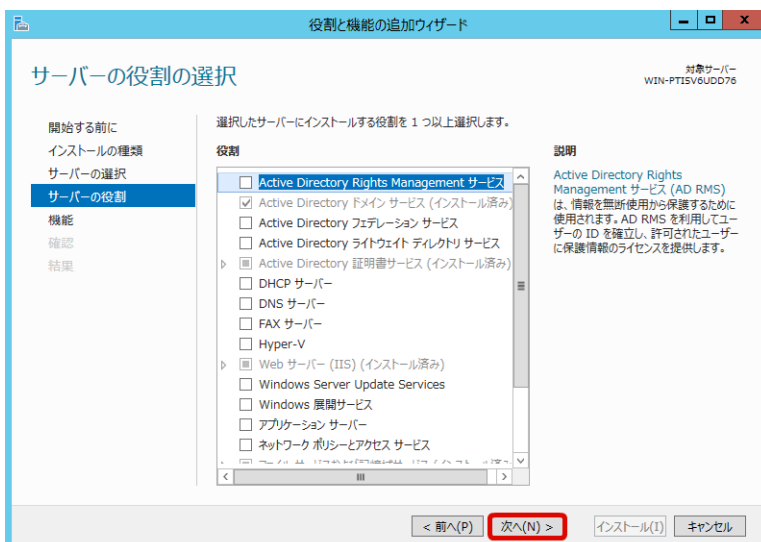
4. 「インストールの種類」より、「役割ベースまたは機能ベースのインストール」を選択します。
→「次へ (N)>」をクリックします。



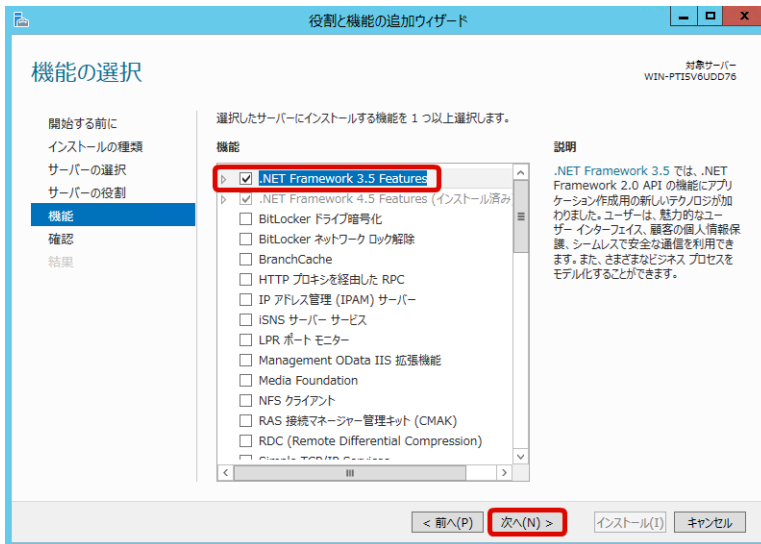
5. 対象サーバを選択します。
→「次へ (N)」をクリックします。



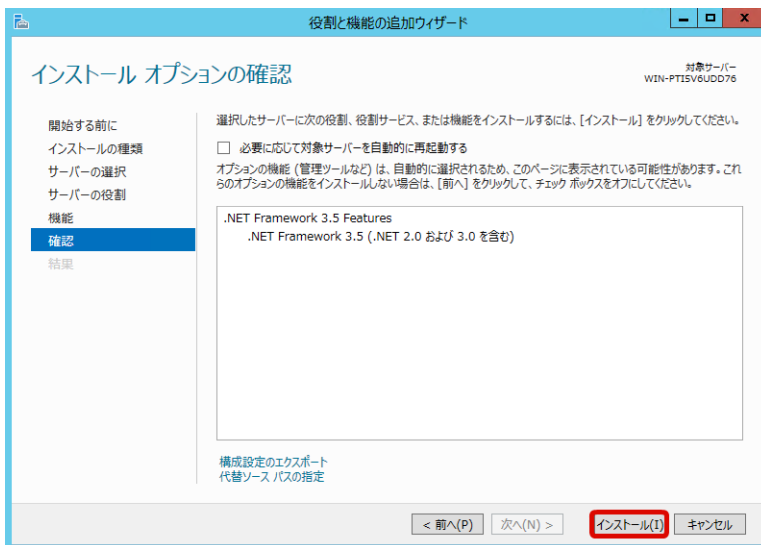
6. 「サーバの役割」の選択は必要ありません。
→「次へ (N)」をクリックします。



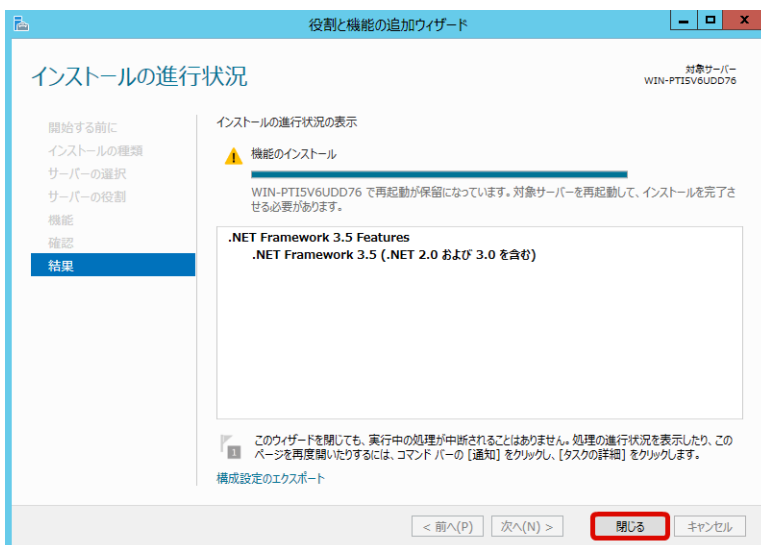
7. 「機能の選択」より、「.NET Framework 3.5 Features」を選択します。
→「次へ (N)」をクリックします。



8. インストール オプションの確認を行います。
→「インストール (I)」をクリックします。



9. インストールが完了したら「閉じる」をクリックします。
必要に応じてOSの再起動を行ってください。



IM-Juggling（モジュールの取得、WARの作成）

IM-Juggling を利用中にエラーが発生してしまう場合

- IM-Juggling を利用中に発生するエラーの原因と対処方法を説明します。

IM-Juggling が最新版になっている必要があります

- 最新のモジュールを取得する場合などにエラーが発生してしまう場合、IM-Juggling ツールが最新状態になっていない可能性があります。
- ・オンライン状態の場合
IM-Juggling の起動時に最新状態への自動更新を行うメッセージが表示されます。
このメッセージにしたがい、IM-Juggling を最新版に更新してください。
- ・オフライン状態の場合
IM-Juggling の自動更新が行えないため、インターネットが接続可能な環境より、最新版の IM-Juggling をダウンロードしてください。
最新版の IM-Juggling は、製品最新情報ダウンロードページ(<http://www.intra-mart.jp/download/product/iap/index.html>)より、ダウンロードが可能です。

ローカル上のファイルが古い可能性があります。

IM-Juggling を利用してプロジェクトの作成や、WARファイルの作成を行っている際にリポジトリ情報の取得等のエラーが発生した場合、古い情報が残っているためにエラーとなる場合があります。
下記のディレクトリにあるデータを削除して再度、IM-Juggling を起動して試行してください。

%OSユーザディレクトリ%/juggling/workspace/.repository ディレクトリ

この古いファイルを削除する事で、最新のデータが再取得され問題を回避します。

WARファイルのデプロイ

Linux環境でWARファイルのデプロイ中にファイル入出力エラーが発生する場合

- Linux環境でwarファイルのデプロイ中にIOException, FileNotFoundException 等が発生した場合の原因と対処方法を説明します。

Web Application Server のエラーログを確認します

- Web Application Server のプロセスが利用できるファイル数の上限がOSにより制限されている場合、エラーログにIOException, FileNotFoundException 等が出力されデプロイに失敗があります。
- Web Application Server 別のエラー内容参照元

Web Application Server	エラー内容の参照元
Resin	起動時のコンソール情報
WebSphere Application Server 8.5.5	%WEBSPHERE_ROOT%/profiles/AppSrv01/logs/server1/SystemErr.log
Oracle WebLogic Server 12c (12.1.3)	%WEBLOGIC_ROOT%/%%USER_PROJECT%/domains/base_domain/servers/AdminServer/logs/AdminServer.log

ファイルディスクリプタの上限を変更します

/etc/system/limits.conf または/etc/security/limits.confで設定されるOSのファイルディスクリプタ数を環境に合わせた以下の数を追加することにより、回避することができますので、適切な値に変更してください。

```
* soft nfile 32768
* hard nfile 32768
root soft nfile 32768
root hard nfile 32768
```

※ ユーザ及び値はサンプルです、環境に合わせて適切な値を設定してください。



コラム

ファイルディスクリプタの現在の設定値は、

```
ulimit -n
```

で確認できます。

Resin でWARファイルのデプロイ中にエラーが発生する場合

- warファイルのデプロイ時に「タイムアウト」のメッセージが表示された場合の原因と対処方法を説明します。

Resin 起動時のコンソール情報の確認と対処方法

Resin のデプロイ時に次のメッセージが出力されているかを確認します。

```
java.lang.IllegalStateException: future timeout
```

原因と対応方法

このメッセージが確認できた場合、デプロイするwarファイルのサイズが大きくなり「%RESIN_HOME%/conf/resin.xml」に指定されている「web-app-deploy dependency-check-interval」より時間がかかる場合(デフォルト2秒)に発生します。

「%RESIN_HOME%/conf/resin.xml」の「dependency-check-interval」の値を大きくします。
 値の単位にはs(秒)のほかに Y(年)/M(月)/W(週)/D(日)/h(時)/m(分)/ms(ミリ秒)を設定できます。

```
<dependency-check-interval>300s</dependency-check-interval>
```

※設定値は環境などによって異なる場合があります。

もしくは、「%RESIN_HOME%/conf/resin.xml」の「web-app-deploy」に「redeploy-mode="manual"」を設定します。

```
<host-default>
<!-- creates the webapps directory for .war expansion -->
<web-app-deploy path="webapps" redeploy-mode="manual"
  expand-preserve-files="WEB-INF/work/*"
  multiversion-routing="{webapp_multiversion_routing}"
  path-suffix="{elastic_webapp?resin.id:}" />
</host-default>
```



コラム

「dependency-check-interval」の値に単位を設定しない場合、s(秒)が適用されます。



コラム

「redeploy-mode="manual"」を設定する事でwarファイルに対する更新チェックが行われなため、上記メッセージが表示されなくなります。

Resin でWARファイルの再デプロイが正常にできない場合

- [WARファイルの再デプロイ](#) (アップデート適用)、[WARファイルの再デプロイ](#) (パッチ適用)において、再デプロイを行ったが、追加したモジュール(資材)がデプロイ先に反映されないといった事象が発生した場合、次の手順を行ってください。

原因

- 稼働するサーバOS上でセキュリティソフトが常駐している場合などに、
%RESIN_HOME%/resin-data ディレクトリ内のファイルが整合性が取れない状態となる可能性があります。

対応方法

- 「resin-data」ディレクトリ配下のファイルを削除します。

アンデプロイ（アンデプロイコマンド及び、<%RESIN_HOME%/webapps/再デプロイ対象のWARファイルのディレクトリ配下>の削除後、
次のディレクトリ配下のファイル群を削除します。

```
<%RESIN_HOME%/resin-data/*>
```

**注意**

Resin は停止した状態で行ってください。

削除後、再デプロイを行ってください。

**注意**

WARファイルによる複数テナント の場合、再デプロイ対象外のテナントも全てアンデプロイ、ファイルの削除を行ってください。

**注意**

分散構成として複数の Resin 環境を構築している場合、全ての Resin 環境において、同様の対応を行ってください。

Web Application Server 起動時

Web Application Server 起動後にログインができない(データベースに接続できない)場合

- 接続するデータベース設定に誤りがある可能性があります。この場合の原因の確認方法を説明します。

Web Application Server 起動時のコンソール情報の確認

Web Application Server 起動時に次のメッセージが出力されているかを確認します。

```
[WARN] c.c.s.w.WebApp - [] java.lang.RuntimeException: resource: iwp/platform/schema/exists_table.sql is not found.  
java.lang.RuntimeException: resource: iwp/platform/schema/exists_table.sql is not found.
```

このメッセージが確認できた場合、データベース設定に誤りがあります。

より詳しい原因を確認するために下記に説明する「データベースログの設定」を変更します。

データベースログの設定

Resin の場合、<%RESIN_HOME%/webapps> 配下のwarファイルと同名のディレクトリ /WEB-INF/conf/log/im_logger_database.xml ファイルをエディタで開き、L.43行目を次のように変更します。

変更前 <level value="off" />

変更後 <level value="trace" />

編集後、Web Application Server を再起動します。

この設定変更によりデータベース関連のトレース情報が有効となり、JDBCドライバからのログ情報がコンソール上で確認できます。



コラム

Resin 以外の Web Application Server の場合、
各 Web Application Server の管理コンソールにある、データソース設定画面より、接続確認を行います。



コラム

この場合における主な原因としては次が考えられます。

- 接続先のデータベースのIPアドレス、ポート番号、データベースユーザ、パスワードの間違い
- データベースユーザの権限不足（権限不足により接続ができない。テーブル作成ができない。）

デブロイ直後の Web Application Server 起動時にエラーが発生する場合

Web Application Server の起動ログの一番最初に出力されるエラー内容を確認します。

- デプロイ直後の Web Application Server の起動時にエラーが発生した場合、根本となるエラーに起因して別のエラーが発生する事があります。このような場合は、Web Application Server の起動ログの一番最初に出力されるエラー内容を確認してください。
- Web Application Server 別のエラー内容参照元

Web Application Server	エラー内容の参照元
Resin	起動時のコンソール情報
WebSphere Application Server 8.5.5	%WEBSPHERE_ROOT%/profiles/AppSrv01/logs/server1/startServer.log
Oracle WebLogic Server 12c (12.1.3)	%WEBLOGIC_ROOT%/%%USER_PROJECT%/domains/base_domain/servers/AdminServer/logs/AdminServer.log

初回アクセス時に「[E.IWP.ADMIN.CONTEXT.10004] Tenant ID cannot be resolved.」が発生します。

現象

- 以下のスタックトレースが出力されます。

```
jp.co.intra_mart.foundation.admin.tenant.InvalidTenantIdException: [E.IWP.ADMIN.CONTEXT.10004] Tenant ID cannot be resolved.
    at jp.co.intra_mart.system.admin.context.StandardTenantIdValidator.validate(StandardTenantIdValidator.java:189)
    at jp.co.intra_mart.system.admin.tenant.TenantIdProvider.validate(TenantIdProvider.java:107)
    at jp.co.intra_mart.system.admin.tenant.TenantIdProvider.getTenantId(TenantIdProvider.java:75)
    at jp.co.intra_mart.system.admin.context.TenantInfoAccountContextDecorator.decorate(TenantInfoAccountContextDecorator.java:23)
    at jp.co.intra_mart.foundation.context.core.ContextBuilderSupport.decorate(ContextBuilderSupport.java:131)
    at jp.co.intra_mart.foundation.context.core.ContextBuilderSupport.build(ContextBuilderSupport.java:54)
    at jp.co.intra_mart.system.context.core.cache.CachingContextBuilderSupport.build(CachingContextBuilderSupport.java:40)
    at jp.co.intra_mart.system.context.impl.command.LifecycleBeginOperation.buildContext(LifecycleBeginOperation.java:96)
    at jp.co.intra_mart.system.context.impl.command.LifecycleBeginOperation.execute(LifecycleBeginOperation.java:64)
    at jp.co.intra_mart.system.context.impl.LifecycleImpl.begin(LifecycleImpl.java:82)
    at jp.co.intra_mart.system.context.web.impl.ContextFilter.doContextFilter(ContextFilter.java:114)
    at jp.co.intra_mart.system.context.web.impl.PreContextFilterChain.doFilter(PreContextFilterChain.java:47)
    at jp.co.intra_mart.system.context.web.impl.ContextFilter.doFilter(ContextFilter.java:78)
    以下省略
```

原因

- IM-SSO または 統合Windows認証モジュールを利用している場合に発生します。
IM-SSO または 統合Windows認証モジュールのテナントID自動解決機能が、テナントIDを自動解決できなかったためです。

回避方法

- IM-SSO、統合Windows認証をモジュールを利用しない場合
IM-SSO、統合Windows認証をモジュールを構成から削除し、WARファイルを再作成後、再デプロイします。
もしくは、以下の URL に従って、機能を無効化します。

IM-SSO	「IM-SecureSignOnを無効化するには」
統合Windows認証	「統合Windows認証機能を無効化するには」

- IM-SSO を利用する場合
IM-SSO の設定が間違っている可能性があります。以下の URL を参照し、設定を確認してください。
[「認証を行うテナントIDの解決方法について」](#)
- 統合Windows認証モジュールを利用する場合
統合Windows認証モジュールの設定が間違っている可能性があります。以下の URL を参照し、設定を確認してください。
[「テナント解決プラグインの設定」](#)

テナント環境セットアップ

テナント環境セットアップ・サンプルデータセットアップに失敗した場合

- セットアップに失敗した原因の確認方法を説明します。

セットアップ実行結果ログの確認

テナント環境セットアップ、サンプルデータセットアップの実行結果ログを確認します。

セットアップ実行結果ログの出力先は以下のとおりです。

- intra-mart Accel Platform 2013 Winter 以前
 - テナント環境セットアップの場合
<STORAGE_PATH>/system/storage/import_result/basic/import-result-detail-data_YYYY-MM-dd_HH-mm-ss.xml
 - サンプルデータセットアップの場合
<STORAGE_PATH>/system/storage/import_result/sample/import-result-detail-data_YYYY-MM-dd_HH-mm-ss.xml
- intra-mart Accel Platform 2014 Spring 以降
 - テナント環境セットアップの場合
<STORAGE_PATH>/system/storage/import_result/basic/<テナントID>/import-result-detail-data_YYYY-MM-dd_HH-mm-ss.xml
 - サンプルデータセットアップの場合
<STORAGE_PATH>/system/storage/import_result/sample/<テナントID>/import-result-detail-data_YYYY-MM-dd_HH-mm-ss.xml

セットアップに失敗した場合、該当の処理では次のようなログが出力されます。（失敗した処理部分のみを抜粋。）

処理結果 ([import-result-detail-data@success](#)) が false となり、エラーメッセージが出力されます。

```

:
<import-result-detail-data success="false">
  <module-id>im_workflow</module-id>
  <execute-id>36b27536-5a95-46d2-9752-489b261b20a1</execute-id>
  <import-type>DML</import-type>
  <importer-id>jp.co.intra_mart.import.StandardAuthzPolicyXmlImporter</importer-id>
  <target-name>products/import/basic/im_workflow/im_workflow-authz-policy.xml</target-name>
  <message>[E.IWP.IMPORTEXPORT.IMPORTER.10001] データのインポートに失敗しました。実行クラス = jp.co.intra_mart.system.authz.services.admin.batch.imex.policy.PolicyXmlImport
</import-result-detail-data>
:

```

より詳しい情報を確認したい場合は、下記の「インポート処理結果ログ」を確認します。

インポート処理結果ログの確認

インポート処理結果ログを確認します。

インポート処理結果ログの出力先は以下の通りです。

<STORAGE_PATH>/system/storage/log/import-export/<%処理実行ID%>.log

<%処理実行ID%> には、セットアップ実行結果ログにおける処理実行ID (execute-id) が適用されます。

セットアップで失敗した場合は、失敗したインポート処理の実行IDによってインポート処理結果ログを特定し、内容を参照することで、例外発生の詳細情報を確認することができます。

i コラム

セットアップ実行結果ログの詳細については「[テナント環境セットアップ 仕様書](#)」を参照してください。

i コラム

例外情報の追跡方法については「[ログ 仕様書 - インポートエラーの原因をログから追跡する](#)」もあわせて参照してください。

トランザクション・タイムアウト設定の確認

Webアプリケーションサーバによってはインポート処理中にトランザクション・タイムアウトが発生し、テナント環境セットアップが失敗する場合があります。
トランザクション・タイムアウトが発生した場合タイムアウト時間を長くするよう設定を変更してください。設定方法については下記のガイドを参照してください。

- [WebSphereのタイムアウト設定](#)
- [WebLogicのタイムアウト設定](#)

テナント環境セットアップ中にタイムアウトが発生した場合

- タイムアウトが発生した場合の原因と対応方法を説明します。

現象・条件

テナント環境セットアップを実施中、Web Application Server 側のログ更新が停止しているのにも関わらず、ブラウザが処理中のステータスのままとなっている（例えば1時間以上経過しているにも関わらず状況が変わらないなど）。

i コラム

Web Server、ブラウザのタイムアウトに限らず、仮にブラウザを閉じてしまっても、テナント環境セットアップ処理はそのまま継続され、Web Application Server 側のログ更新が継続されます。
このような場合は、Web Application Server 側のログ更新が完了するまで待機します。

原因

Web Server、ブラウザのタイムアウトにより結果画面に遷移していない可能性があります。

i コラム

Web Application Server や データベース 側でエラーが発生する場合、Web Application Server や データベース 側のタイムアウトが原因である可能性があります。

Web Application Server については下記を参照してください。
「[Oracle WebLogic Server 12c \(12.1.3\) でテナント環境セットアップに失敗した場合](#)」、
「[WebSphere Application Server 8.5.5 - トランザクション・タイムアウトの設定](#)」
データベース については、製品元のドキュメントを参照してください。

対応方法

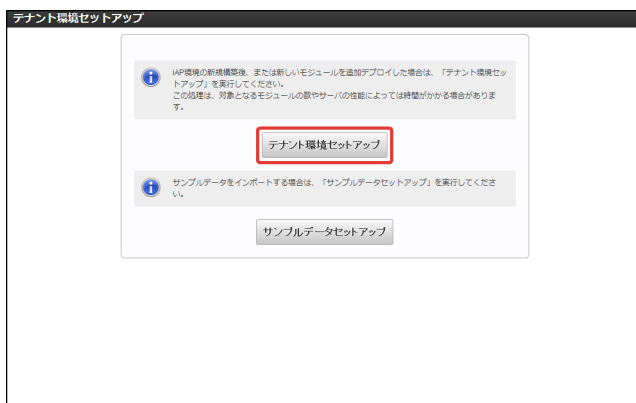
- テナント環境セットアップが正常に完了しているかを確認します。
システム管理者のメニュー画面を表示します。
メニューから「テナント環境セットアップ」をクリックします。



下図のように、「テナント環境は最新です。セットアップが必要なモジュールはありません。」という旨のメッセージが表示されていれば、テナント環境セットアップは正常に完了しています。



下図のように、「テナント環境セットアップ」ボタンが表示されている場合、テナント環境セットアップは未完了です。「テナント環境セットアップ」ボタンをクリックしセットアップを再度実施します。



！ 注意

再度セットアップを実施する前に、現象が再発しないよう Web Server 等のタイムアウト設置値を変更する、もしくは、Web Application Server 経由でセットアップを実施する事を推奨します。

i コラム

サンプルをセットアップするボタンは処理結果に関わらず、常に表示される仕様となります。このため、サンプルデータ投入中にエラーが発生した場合、再度セットアップを実施するとデータベースで一意制約違反が発生します。この場合は、「[アンインストール](#)」を行い、改めてセットアップからやり直す事を推奨します。

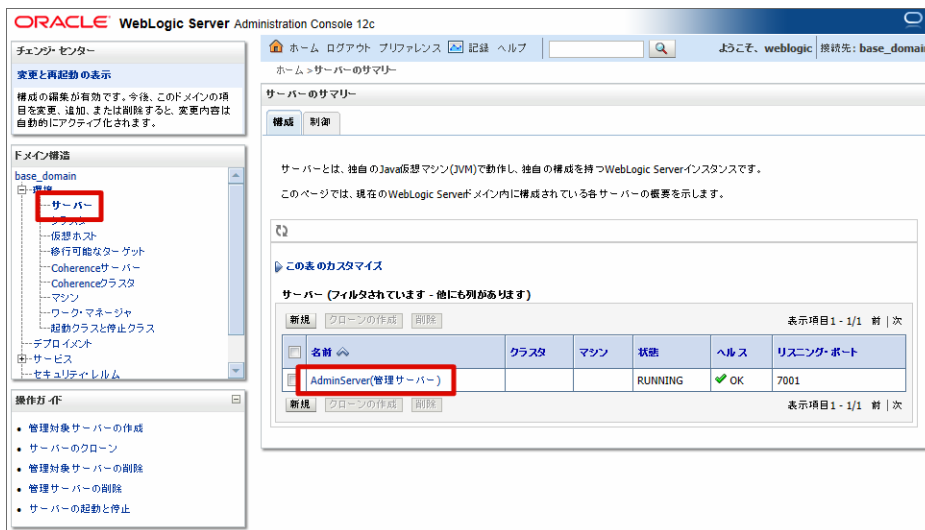
Oracle WebLogic Server 12c (12.1.3) でテナント環境セットアップに失敗した場合

- Oracle WebLogic Server 12c (12.1.3) 環境でテナント環境セットアップに失敗した場合の原因と対処方法を説明します。

原因と対応方法

データベースの性能等により、テナント環境セットアップに時間がかかりスレッドをスタック状態とみなされセットアップが失敗する場合があります。
下記の手順で設定を変更してください。

1. 左メニューの [環境]-[サーバ] を選択します。
右画面の使用するサーバを選択します。



2. [チューニング] タブを選択し、[スタック・スレッド最大時間] にデフォルト値より長い時間を入力します。
入力後、[保存へ] ボタンをクリックします。



3. Oracle WebLogic Server 12c (12.1.3) を再起動します。

テナント環境セットアップ後に、テナント管理者でログインができない場合

- テナント環境セットアップ時に作成したテナント管理者でログインができない、またアカウントの設定で「ライセンス」を有効としても更新時にデータベースエラーとなる場合の原因と対処方法を説明します。

発生条件

- この現象は データベース に、Microsoft SQL Server 2008 R2、Microsoft SQL Server 2012 を利用している場合に確認されています。これ以外の データベース を利用されている場合においても、下記に挙げる原因によっては該当する事が考えられます。

原因と対応方法

アカウントの設定で「ライセンス」を有効としても、更新時にエラーとなった場合、Resin のコンソール上に次のメッセージが出力されているかを確認します。

```
[ERROR] j.c.i.s.j.i.UserLicenseObject - [] [E.IWP.LICENSE.USER.00003] データベースエラーが発生したため、ライセンスの登録に失敗しました。  
jp.co.intra_mart.system.secure.license.exception.LicenseException: [E.IWP.LICENSE.USER.00003] データベースエラーが発生したため、ライセンスの登録に失敗しました。
```

このメッセージが確認できた場合、次の原因が考えられます。

- JDBCドライバのバージョンが古い

[JDBCドライバ](#)を参照してください。

この章で記載しているリンク先のJDBCドライバである必要があります。



注意

Microsoft SQL Server 2008 R2、Microsoft SQL Server 2012 以外の データベース の場合でも使用しているJDBCドライバが古いと同様に正常に動作しない可能性が考えられます。

- Resin の照合順序の設定

[データベースサーバ](#)を参照してください。

この章で記載している照合順序の設定が必要となります。

ベースURLを設定した環境でシステム管理者ログイン画面表示時にエラーが発生してしまう場合

- ベースURLを設定した場合に発生するエラーの原因と対処方法を説明します。

適切でないベースURLが設定されていることが考えられます。

- ベースURLにローカルホストなどの不適なURLをしていることが考えられます。

適切なURLを設定して再度環境セットアップを行ってください。



コラム

ベースURLとは？

intra-mart Accel Platform のシステムを外部から参照する際に利用される基底のURLです。
一般的には、<http://example.org/imart> 等となります。

このベースURLは、あくまでも外部から参照される際に利用するURLです。

従って、localhostや、127.0.0.1 (ループバックアドレス)等をURLとして指定された場合、外部からの接続時に適切にサーバに接続されなくなります。

その他

SAStruts版ポートレットが404エラーでアクセスできない場合

事象内容

- 作成・登録したSAStruts版ポートレットが404でアクセスできない事象が発生します。
登録したSAStruts版ポートレットにアクセス可能なURLを直接要求すると表示されるようになりますが、Web Application Server を再起動すると事象が再発します。

原因と対応方法

- 「[SAStruts用設定ファイル \(SAStruts版ポートレットを利用する場合\)](#)」を参照してください。
※上記対応を行った後、再度WARを作成し、再デプロイを実施する必要があります。

Apache Solr で利用するJavaのバージョンによって、起動パラメータを設定する必要があります。

- 詳細と対応内容は「[Solr管理者ガイド](#)」 - 「[Java SE Development Kit 7u40 以上を利用する場合](#)」を参照してください。