

Copyright © 2012 NTT DATA INTRAMART CORPORATION

目次

- 改訂情報
- はじめに
 - 対象読者
 - 本書の構成
 - 本書内の説明について

[索引](#)

- Resin
 - resin-web設定
 - ヘルス機能
 - ログ設定
- コアモジュール
 - ネットワーク設定
 - サーバコンテキスト設定
 - ストレージ設定ファイル
 - サービスセレクトア設定
 - initializer 設定
 - IPアドレス取得元設定
- マルチデバイス
 - クライアントタイプマスタ
- 国際化機能
 - 日付と時刻の形式マスタ
 - ロケールマスタ
 - タイムゾーンマスタ
- スクリプト開発モデル
 - source-config.xml
- データベースアクセス機能
 - データソースマッピング設定
- メールモジュール
 - メール設定
- Webモジュール
 - リクエスト制御設定
 - レスポンスヘッダ設定
- IM-Propagation
 - IM-Propagation 設定
 - IM-Propagation 受信側設定
 - IM-Propagation 送信側設定
- テナント管理機能
 - システム管理者用スクリプト開発モデルルーティング設定
 - システム管理者用IM-JavaEE Frameworkルーティング設定
 - システム管理者用サーバレットルーティング設定
 - 認可判断モジュール設定
 - 認可設定画面設定
 - ポリシー部分編集定義設定
 - 認可設定画面 保護リソース設定
 - ルーティングテーブル用 認可リソースマッピング定義設定
 - リソースタイプ拡張設定
 - サブジェクトタイプ拡張設定
 - 暗号化設定
 - サブジェクトリゾルバ (DeclaredSubjectResolver)拡張設定
 - サブジェクトリゾルバ (OnDemandSubjectResolver)拡張設定
 - 認可ポリシーキャッシュ対象設定
 - パスワード履歴管理設定
 - スクリプト開発モデルルーティング設定
 - IM-JavaEE Frameworkルーティング設定
 - サーブレットルーティング設定
 - ショートカットアクセス設定
 - システム期間情報の管理
 - セキュア・トークンフィルタ設定
 - システム管理者用グローバルナビ設定
- 外部メニュー連携
 - 外部メニュー連携接続先設定
- iAP-iWP間SSO連携モジュール(IM-HybridSSO)
 - SSO連携用マッピング設定
- サービス機構モジュール
 - キャッシュ設定
 - キャッシュデフォルト設定
- UI
 - セッション管理設定
 - UI タグコンポーネント設定
 - テーマの適用方法設定
HeadWithFooterThemeBuilder
 - テーマの適用方法設定
HeadWithContainerThemeBuilder
 - テーマの適用方法設定
HeadOnlyThemeBuilder
 - テーマの適用方法設定
BodyOnlyThemeBuilder
 - テーマの適用方法設定
NoThemeBuilder
 - テーマの適用方法設定
FullThemeBuilder
 - ライブラリ群設定
- Webサービス 認証・認可クライアント
 - SOAPClientオブジェクトの設定
- 認証機能
 - 認証設定(一般ユーザ用)
 - 認証確認設定(一般ユーザ用)
 - 認証確認対象ページ設定(一般ユーザ用)
 - 認証セッション設定(一般ユーザ用)
 - 認証外部ページURL許可リスト設定
 - リクエストパラメータ設定
 - セッション情報チェック設定
- 統合Windows認証機能
 - 統合Windows認証設定
 - 統合Windows認証パス設定
- システム管理機能
 - 認証設定(システム管理者用)
 - 認証IPアドレス制限設定(システム管理者用)
 - 認証確認設定(システム管理者用)
 - 認証確認対象ページ設定(システム管理者用)
 - 認証セッション設定(システム管理者用)
- LDAP認証モジュール
 - LDAP認証設定ファイル
- ジョブスケジューラ
 - ジョブスケジューラの設定
- パスワードリマインダ機能
 - パスワードリマインダ設定
- IM-共通マスタ
 - IM-共通マスタ設定
- ViewCreator
 - ViewCreatorの設定
 - ViewCreatorの関数設定
 - ViewCreatorファイルリソースの設定
- IMBox
 - Cassandra設定
 - IMBox設定
 - チャット表示設定
- IM-ContentsSearch

Resin

resin-web設定

項目

- 概要
- リファレンス
 - ログハンドラ設定
 - リソース設定
 - JSP設定
 - タグ再利用設定
 - データベース設定
 - ドライバ設定
 - ドライバタイプ設定
 - ドライバURL設定
 - ユーザ設定
 - パスワード設定
 - 暗号化パスワード設定
 - 最大コネクション数設定
 - プリペアドステートメントキャッシュ設定
 - セッション設定
 - セッションIDの再利用設定
 - セッションタイムアウト設定
 - JSPプリコンパイル設定

概要

Resinに関する設定です。

モジュール Resin

フォーマットファイル
(xsd)

設定場所 WEB-INF/resin-web.xml

```
<web-app xmlns="http://caucho.com/ns/resin" xmlns:resin="urn:java:com
<log-handler name="" class="jp.co.intra_mart.common.platform.log.handler
<!-- im_service(im_asynchronous) -->
<resource jndi-name="jca/work" type="jp.co.intra_mart.system.asynchronous
<jsp>
  <recycle-tags>false</recycle-tags>
</jsp>
<database jndi-name="jdbc/default">
  <driver>
    <type>org.postgresql.Driver</type>
    <url>jdbc:postgresql://localhost:5432/iap_db</url>
    <user>imart</user>
    <password>imart</password>
  </driver>
  <max-connections>20</max-connections>
  <prepared-statement-cache-size>8</prepared-statement-cache-size>
</database>
<session-config>
  <reuse-session-id>false</reuse-session-id>
  <session-timeout>30</session-timeout>
</session-config>
</web-app>
```

リファレンス

ログハンドラ設定

- システム管理者用ホームウィジェット設定
- システム管理者用ユーティリティメニュー設定
- IM-Notice
 - IM-Notice 設定
 - IM-Notice Mobile設定
 - IM-Notice MQ設定
 - IM-Notice MQブローカー設定
- Solrサーバ接続設定
- 検索画面設定
- 検索権限生成設定
- 検索フィールド設定
- テキスト抽出設定
- 検索結果テンプレート設定
- IM-FileExchange
 - IM-FileExchangeの設定
- OAuth認証機能
 - クライアント詳細設定
 - クライアントリソース設定
 - クライアントのアクセス範囲設定
- OAuthクライアント
 - OAuthプロバイダ設定
- ログ
- 互換モジュール
 - 汎用設定ファイル

索引

タグ
log-handler
名

ログハンドラに関する設定です。

【設定項目】

```
<web-app>  
<log-handler name="" class="jp.co.intra_mart.common.platform.log.han  
</web-app>
```

必須項目

x

複数設定

○

設定値・設定する内容

ログハンドラに関する設定をします。

単位・型

なし

省略時のデフォルト値

なし

親タグ

web-app

【属性】

属性名	説明	必須	デフォルト値
name	ログハンドラの名前を指定します。	○	なし
class	ログハンドラのクラス名を指定します。	○	なし



注意

この設定は変更しないでください。

リソース設定

タグ
resource
名

リソースに関する設定です。

【設定項目】

```
<web-app>  
<resource jndi-name="jca/work" type="jp.co.intra_mart.system.asynchro  
</web-app>
```

必須項目

x

複数設定

○

設定値・設定する内容

リソースに関する設定をします。

単位・型

なし

省略時のデフォルト値

なし

親タグ

web-app

改訂情報

変更年月日	変更内容
2012-12-21	初版
2013-04-01	第2版 下記を追加・変更しました 「IPアドレス取得元設定」を追加 「認証IPアドレス制限設定(システム管理者用)」を追加 「IM-FileExchangeの設定」を追加 「認可設定画面設定」を追加 「リソースタイプ拡張設定」を追加 「LDAP認証設定ファイル」を追加 「外部メニュー連携接続先設定」を追加 「ポリシー部分編集定義設定」に「コールバック設定」を追加 「Cassandra設定」に「キースペース」を追加 「IMBox設定」に「Noticeタイムラインのメッセージ表示件数」、「1メッセージの通知先に指定できるユーザ数の上限値」、「1メッセージに添付可能なファイル数の上限値」を追加
2013-07-01	第3版 下記を追加・変更しました 「本書内の説明について」を追加 「UI タグコンポーネント設定」を追加 「テーマの適用方法設定 HeadWithFooterThemeBuilder」を追加 「テーマの適用方法設定 HeadWithContainerThemeBuilder」を追加 「テーマの適用方法設定 HeadOnlyThemeBuilder」を追加 「テーマの適用方法設定 BodyOnlyThemeBuilder」を追加 「テーマの適用方法設定 NoThemeBuilder」を追加 「ViewCreatorの関数設定」を追加 「認可設定画面設定」に「リソースグループバックアップ設定」、「バックアップ先パス設定」、「バックアップ最大件数設定」を追加 「Cassandra設定」に「接続認証設定」「接続ユーザ名」「パスワード」を追加
2013-10-01	第4版 下記を追加・変更しました 「外部メニュー連携接続先設定」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 「システムログ」に「ログメッセージコード」に関する説明を追加 「キャッシュ設定」に「属性」に関する説明を追加、更新 「キャッシュデフォルト設定」に「属性」に関する説明を追加、更新 「resin-web設定」の「プリペアドステートメントキャッシュ設定」に関する説明を更新 「汎用設定ファイル」の「VirtualFile72, NetworkFile72 APIにおけるエンコーディングの設定」に関する説明を追加 「サービスセレクトa設定」を追加 「暗号化設定」の設定ファイルの設置場所を更新 「IMBox設定」に「非表示のグループ一覧のグループ表示件数」に関する説明を追加 「サブレットルーティング設定」に認可を使用する際の注意点を追加 「メール設定」に「SMTPS(SMTP over SSL)設定」に関する説明を追加

属性名	説明	必須	デフォルト値
jndi-name	リソースのJNDI名を指定します。	○	なし
type	リソースのタイプを指定します。	○	なし

 **注意**
この設定は変更しないでください。

JSP設定

タグ名

JSPに関する設定です。

【設定項目】

<pre><web-app> <jsp> ... </jsp> </web-app></pre>	
必須項目	×
複数設定	×
設定値・設定する内容	JSPの設定をします。
単位・型	なし
省略時のデフォルト値	なし
親タグ	web-app

タグ再利用設定

タグ名

JSPに関する設定です。

タグクラスのインスタンスを再利用します。

【設定項目】

<pre><web-app> <jsp> <recycle-tags>false</recycle-tags> </jsp> </web-app></pre>	
必須項目	×
複数設定	×

変更年月日	変更内容
2014-01-01	第5版 下記を追記・変更しました 「ログ」の項目を「ログ仕様書」に移動 「認可判断モジュール設定」の「decision-config」タグ「combinator」属性の説明を修正 「キャッシュ設定」にサイズ計算に関する警告を追加 「キャッシュデフォルト設定」にサイズ計算に関する警告を追加 「IM-ContentsSearch」に「検索画面設定」を追加 「IM-ContentsSearch」に「検索権限生成設定」を追加 「IM-ContentsSearch」に「テキスト抽出設定」を追加 「IM-ContentsSearch」に「検索フィールド設定」を追加 - 属性 path に関するワイルドカードの説明を追加 「システム管理者用スクリプト開発モデルルーティング設定」 「システム管理者用IM-JavaEE Frameworkルーティング設定」 「システム管理者用サーブレットルーティング設定」 「スクリプト開発モデルルーティング設定」 「IM-JavaEE Frameworkルーティング設定」 「サーブレットルーティング設定」 「IMBox設定」に「表示に関する設定」を追加 「IMBox設定」に「利用不可ユーザを表示するか否かを判定する値」を追加 「IMBox設定」に「アンケートに投票をしたユーザー一覧のユーザ表示件数」を追加 「IMBox設定」に「利用不可ユーザのユーザー一覧のユーザ表示件数」を追加 「パスワード履歴管理設定」の「パスワード履歴管理対象外クライアントタイプ」に関する省略時のデフォルト値を更新 「レスポンスヘッダ設定」を追加 「テーマの適用方法設定 HeadWithFooterThemeBuilder」のテーマの適用方法の参照先を変更しました。 「テーマの適用方法設定 HeadWithContainerThemeBuilder」のテーマの適用方法の参照先を変更しました。 「テーマの適用方法設定 HeadOnlyThemeBuilder」のテーマの適用方法の参照先を変更しました。 「テーマの適用方法設定 BodyOnlyThemeBuilder」のテーマの適用方法の参照先を変更しました。 「テーマの適用方法設定 NoThemeBuilder」のテーマの適用方法の参照先を変更しました。 「IM-FileExchangeの設定」の「listener」タグ「show-guest-address」属性を追加 「メール設定」の「smtps」タグ「enable」属性の説明を修正 「メール設定」の「smtps」タグ「starttls」属性の説明を修正 「メール設定」の「encode」タグ「locale」属性の説明を修正 「キャッシュ設定」の「cache」タグ「enable」属性の説明を修正 「キャッシュデフォルト設定」の「cache」タグを「default-cache」に修正 「キャッシュデフォルト設定」の「mbeans」タグを追加

設定値・設定する内容	true タグのインスタンスを再利用します。
	false タグのインスタンスを再利用しません。

単位・型	真偽値 (true/false)
省略時のデフォルト値	なし
親タグ	jsp



注意

この設定は変更しないでください。

データベース設定

タグ名	database
-----	----------

データベース接続に関する設定です。

【設定項目】

```
<web-app>
<database jndi-name="jdbc/default">
...
</database>
</web-app>
```

必須項目	×
複数設定	○
設定値・設定する内容	データベース接続を行うための設定をします。
単位・型	なし
省略時のデフォルト値	なし
親タグ	web-app

【属性】

属性名	説明	必須
jndi-name	このデータベース設定のJNDI名を設定します。	○

ドライバ設定

タグ名	driver
-----	--------

JDBCドライバに関する設定です。

【設定項目】

```
<web-app>
<database jndi-name="jdbc/default">
<driver>
...
</driver>
</database>
</web-app>
```

変更年月日	変更内容
2014-04-01	第6版 下記を追記・変更しました 「外部メニュー連携接続先設定」の「menu-provider」タグに「target-tenant」属性を追加 「外部メニュー連携接続先設定」の「メニュープロバイダの接続先ロギンググループ設定」に関する説明を更新 「統合Windows認証機能」を追加 「IMBox設定」の「tns:comet_flag」タグの説明にアプリケーション通知を追加 「IMBox設定」の「tns:mail」タグと「tns:from」タグに注意事項を追加 「パスワード履歴管理設定」に「group」タグの説明を追加 「パスワード履歴管理設定」の各タグ・属性に関してデフォルト値の説明を更新 「メール設定」の「smtp-server」タグに関する説明を更新 「UI」から「GoogleChromeFrame設定」を削除 「パスワードリマインダ設定」に「urlLimitDateFormat」タグの説明を追加 「パスワードリマインダ設定」に「mailCc」タグの説明を追加 「パスワードリマインダ設定」に「mailBcc」タグの説明を追加 「パスワードリマインダ設定」に「address」タグの説明を追加 「ストレージ設定ファイル」の「パブリックストレージディレクトリ名設定」に関連説明を更新 「システム管理者用グローバルナビ設定」を追加 「システム管理者用ホームウィジェット設定」を追加 「システム管理者用ユーティリティメニュー設定」を追加 「認証設定(システム管理者用)」に「強制ログイン時のセッション再作成の有無」パラメータの説明を追加 「ショートカットアクセス設定」の概要を更新しました。 「パスワード履歴管理設定」の概要を更新しました。 「認証設定(一般ユーザ用)」に「自動ログインについて」と、「強制ログイン時のセッション再作成の有無」パラメータの説明を追加 「リクエストパラメータ設定」に「im_tenant_id」パラメータの説明を追加
2014-05-30	第7版 下記を追記・変更しました 「ヘルス機能」を追加 「initializer 設定」を追加
2014-08-01	第8版 下記を追記・変更しました 「ログ設定」を追加 「検索結果テンプレート設定」を追加 「resin-web設定」の「max-connections」タグのデフォルト値を修正 「テキスト抽出設定」の概要に記載しているサンプルを更新 「検索画面設定」に「sort」タグの説明を追加 「検索画面設定」に「default-order」タグの説明を追加 「LDAP認証設定ファイル」の「connect-timeout」タグの単位・型を修正 「SSO連携用マッピング設定」を追加 「認可設定画面設定」に「Excelインポート設定」タグの説明を追加 「認可設定画面設定」に「Excelエクスポート設定」タグの説明を追加 「リクエストパラメータ設定」で「ログイン・ログアウト時に利用する遷移先URL」設定時の注意を追記

必須項目	○
複数設定	×
設定値・設定する内容	JDBCドライバの設定をします。
単位・型	なし
省略時のデフォルト値	なし
親タグ	database

ドライバタイプ設定

タグ名	type
-----	------

JDBCドライバのタイプに関する設定です。

【設定項目】

```
<web-app>
<database jndi-name="jdbc/default">
  <driver>
    <type>org.postgresql.Driver</type>
    ...
  </driver>
</database>
</web-app>
```

必須項目	○
複数設定	×
設定値・設定する内容	JDBCドライバのタイプを設定をします。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	driver

ドライバURL設定

タグ名	url
-----	-----

JDBCドライバのURLに関する設定です。

【設定項目】

```
<web-app>
<database jndi-name="jdbc/default">
  <driver>
    ...
    <url>jdbc:postgresql://localhost:5432/iap_db</url>
    ...
  </driver>
</database>
</web-app>
```

必須項目	○
複数設定	×

変更年月日	変更内容
2014-12-01	第9版 下記を追記・変更しました 「OAuth認証機能」を追加 「IMBox設定」に「スレッド内の返信メッセージ取得件数」を追加 「スクリプト開発モデル」を追加 「IM-Notice」を追加 「LDAP認証設定ファイル」は、ひな形として利用されることを追記 「セッション管理設定」の注意の内容を修正
2015-04-01	第10版 下記を追記・変更しました 「サービスセクタ設定」リスナ設定のコラムからQueueServiceのIDを削除しました。 「リクエスト制御設定」処理中のリクエストの有効期間単位についてを追記 「メール設定」に「mail-headers」タグの説明を追加 「メール設定」に「header」タグの説明を追加 「IMBox設定」に「Unread Messages (PC版)のメッセージ表示件数」を追加 「IMBox設定」に「Unread Messages (SP版)のメッセージ表示件数」を追加 「ロケールマスタ」のシステムロケール変更に関する説明を修正
2015-08-01	第11版 下記を追記・変更しました 「resin-web設定」に「JSPブリコンパイル設定」を追加。 「source-config.xml」に「source-config.xmlの有効範囲」を追加 「IMBox設定」に「非表示メッセージタイムラインのスレッド表示件数」を追加 「IMBox設定」に「入力チェック設定」を追加 「IMBox設定」に「グループ名の重複チェック設定」を追加 「テーマの適用方法設定 HeadWithFooterThemeBuilder」に属性client-typeとlibraries-pathを追加 「テーマの適用方法設定 HeadWithContainerThemeBuilder」に属性client-typeとlibraries-pathを追加 「テーマの適用方法設定 HeadOnlyThemeBuilder」に属性client-typeとlibraries-pathを追加 「テーマの適用方法設定 BodyOnlyThemeBuilder」に属性client-typeとlibraries-pathを追加 「テーマの適用方法設定 NoThemeBuilder」に属性client-typeとlibraries-pathを追加 「テーマの適用方法設定 FullThemeBuilder」を追加 「ライブラリ群設定」を追加 「認証設定(一般ユーザ用)」の以下に関する警告を修正 ログインページURL ログイン実行ページURL ログアウト実行ページURL 「認証設定(システム管理者用)」の以下に関する警告を修正 ログインページURL ログイン実行ページURL テナント初期設定ページURL 「認証確認設定(一般ユーザ用)」の以下に関する警告を修正 認証確認ページURL 認証確認実行ページURL 「認証確認設定(システム管理者用)」の以下に関する警告を修正 認証確認ページURL 認証確認実行ページURL 「IM-Notice」に「Baidu APIキーの設定」を追加

設定値・設定する内容	JDBCドライバのURLを設定をします。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	driver

ユーザ設定

タグ名	user
-----	------

接続するデータベースのユーザに関する設定です。

【設定項目】

```
<web-app>
<database jndi-name="jdbc/default">
  <driver>
    ...
    <user>imart</user>
    ...
  </driver>
</database>
</web-app>
```

必須項目	○
複数設定	×
設定値・設定する内容	データベースのユーザを設定をします。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	driver

パスワード設定

タグ名	password
-----	----------

接続するデータベースのユーザのパスワードに関する設定です。

【設定項目】

```
<web-app>
<database jndi-name="jdbc/default">
  <driver>
    ...
    <password>imart</password>
    ...
  </driver>
</database>
</web-app>
```

必須項目	○
複数設定	×
設定値・設定する内容	データベースのユーザのパスワードを設定をします。
単位・型	文字列

変更年月日	変更内容
2015-12-01	第12版 下記を追記・変更しました 「source-config.xml」に「ファイル単位での設定方法」を追加 「source-config.xml」に「HTMLコメントを削除する設定」を追加 「JSPプリコンパイル設定」の内容を修正 「パスワード履歴管理設定」の「パスワードの長さチェック」に 属性 min の最小値、および、属性 max の 最大値に関する説明を追記 「LDAP認証設定ファイル」にバージョンによってファイルの扱いが異なることを追記 「OAuthクライアント」に「OAuthプロバイダ設定」を追加 「ジョブスケジューラ」に「ジョブスケジューラサービスの設定」を追記 「ジョブスケジューラ」に「遅延起動の設定」を追記 「ジョブスケジューラ」に「起動判定処理の設定」を追記 「ジョブスケジューラ」に「最大遅延時間(秒)」を追記 「ジョブスケジューラ」に「起動判定チェック間隔(ミリ秒)」を追記 「ライブラリ群の指定」にclient-type に pc を指定した定義を追記 「テーマの適用方法設定 HeadWithFooterThemeBuilder」の属性 libraries-versionのclient-typeにpcを指定した場合の記述を追記 「テーマの適用方法設定 HeadWithContainerThemeBuilder」に属性 libraries-versionのclient-typeにpcを指定した場合の記述を追記 「テーマの適用方法設定 HeadOnlyThemeBuilder」に属性 libraries-versionのclient-typeにpcを指定した場合の記述を追記 「テーマの適用方法設定 BodyOnlyThemeBuilder」に属性 libraries-versionのclient-typeにpcを指定した場合の記述を追記 「テーマの適用方法設定 NoThemeBuilder」に属性 libraries-versionのclient-typeにpcを指定した場合の記述を追記 「テーマの適用方法設定 FullThemeBuilder」に属性 libraries-versionのclient-typeにpcを指定した場合の記述を追記 「resin-web設定」の「ブリベアドステートメントキャッシュ設定」に PostgreSQL JDBC を使用する際の説明を追記 「ストレージ設定ファイル」に「シンボリックリンク設定」を追記 「チャット表示設定」を追加 「ポリシー部分編集定義設定」に、「resource-group-authorizer要素」に指定するクラスの型を追記 「LDAP認証設定ファイル」に「SSL接続(LDAPS)の設定」を追記

省略時のデフォルト値	なし
親タグ	driver

暗号化パスワード設定

タグ名	resin:Password
-----	----------------

接続するデータベースのユーザのパスワードに関する設定です。暗号化されたパスワードを使用する場合に使用します。

このタグを使用する場合は、予めパスワードを暗号化してください。パスワード

```
C:\resin-pro-4.x.xx>resin password-encrypt <パスワード>
password: {RESIN}WFpCuWYyYSFzuOK1Gda0TxQ==
```

Linux環境の場合はresinctlコマンドを使用してください。resinctlコマンドを使用するにはResinがコンパイル、インストール済みである

```
[resin-pro-4.x.xx]# bin/resinctl password-encrypt <パスワード>
password: {RESIN}WFpCuWYyYSFzuOK1Gda0TxQ==
```

このタグを使用するには設定ファイルに以下のネームスペースが記述されて

```
xmlns:resin="urn:java:com.caucho.resin"
```

【設定項目】

```
<web-app xmlns="http://caucho.com/ns/resin" xmlns:resin="urn:java:com
...
<database jndi-name="jdbc/default">
  <driver>
    ...
    <password>
      <resin:Password value="{RESIN}WFpCuWYyYSFzuOK1Gda0TxQ="
    </password>
    ...
  </driver>
</database>
</web-app>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	password

【属性】

属性名	説明	必須	デフォルト値
value	暗号化されたパスワードを設定します。	○	なし

はじめに

項目
■ 対象読者
■ 本書の構成
■ 本書内の説明について

対象読者

次の利用を対象としています。

intra-mart Accel Platform を利用したアプリケーション開発
intra-mart Accel Platform 上でのシステム運用

本書の構成

- 本書は intra-mart Accel Platform で提供されている設定ファイルに関する説明を行っています。
- 機能(モジュール)別の設定ファイルの種類
 - 設定ファイル別の仕様説明
 - 設定項目に対する設定内容ならびに注意点の説明

本書内の説明について

- 「デフォルト値」
該当の設定を省略した場合の値となります。製品出荷時の値ではありません。

最大コネクション数設定

タグ名	max-connections
-----	-----------------

データベースプールで利用可能なコネクションの最大数を設定します。

【設定項目】

```
<web-app>
  <database jndi-name="jdbc/default">
    ...
    <max-connections>20</max-connections>
  </database>
</web-app>
```

必須項目	x
複数設定	x
設定値・設定する内容	コネクションの最大数
単位・数値型	
省略時のデフォルト値	1024
親タグ	database

プリペアドステートメントキャッシュ設定

タグ名	prepared-statement-cache-size Postgresql Driver, Version 9.4-1202 以降の場合 メータに設定
-----	--

PreparedStatementを使用してSQLを発行すると、作成されたステートメントが次回、同じSQLが発行された場合、キャッシュされたステートメントが再利用されます。prepared-statement-cache-sizeは、キャッシュを行うステートメントの個数を設定します。intra-mart Accel Platform では、機能によりSQLの発行や内容が回数で異なるため、8, 16程度の個数が設定されている場合キャッシュにヒットせずこのプリペアドステートメントのキャッシュが適切にヒットするように設定する場合、

【設定項目】

以下のJDBCドライバを使用する場合はprepared-statement-cache-sizeタグ

- Oracle
- Microsoft SQLServer
- Postgresql Driver, Version 9.4-1201 以前

```
<web-app>
  <database jndi-name="jdbc/default">
    ...
    <prepared-statement-cache-size>8</prepared-statement-cache-size>
  </database>
</web-app>
```

Postgresql Driver, Version 9.4-1202 以降を使用する場合はJDBCドライバのprepared-statement-cache-sizeタグの設定は適用されません。

```
<web-app
  <database jndi-name="jdbc/default">
    <driver>
      ...
    <init-param>
      <param-name>preparedStatementCacheQueries</param-name>
      <param-value>8</param-value>
    </init-param>
    </driver>
  </database>
</web-app>
```

必須項目	x
複数設定	x
設定値・設定する内容	プリペアドステートメントのキャッシュ数
単位・型	数値
省略時のデフォルト値	0
親タグ	database Postgresql Driver, Version 9.4-1202 以降の場合は dr

セッション設定

タグ名 session-config

セッションに関する設定です。

【設定項目】

```
<web-app>
  <session-config>
    ...
  </session-config>
</web-app>
```

必須項目	x
複数設定	○
設定値・設定する内容	セッションに関する設定をします。
単位・型	なし
省略時のデフォルト値	なし
親タグ	web-app

セッションIDの再利用設定

タグ名 reuse-session-id

HTTPセッションのIDを再利用するかどうかを設定します。
trueを設定した場合、過去に利用されたセッションIDを再利用します。
再利用されるのはIDのみで、過去に保管されていたセッションの情報は引き
ログイン、ログアウトにおいて、セッションを再作成しますが、その時に異なる

【設定項目】

```
<web-app>
  <session-config>
    <reuse-session-id>false</reuse-session-id>
  </session-config>
</web-app>
```

必須項目	x
複数設定	x
設定値・設定する内容	HTTPセッションのIDを再利用するかどうかを設定します。
単位・型	真偽値(true/false)
省略時のデフォルト値	true
親タグ	session-config

セッションタイムアウト設定

タグ名
session-timeout

HTTPセッションのタイムアウト時間です。指定した時間セッションを維持しま
【設定項目】

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

必須項目	x
複数設定	x
設定値・設定する内容	HTTPセッションがタイムアウトされるまでの時間を設
単位・型	数値
省略時のデフォルト値	30
親タグ	session-config

 注意

セッションタイムアウトは、設定された時間に対して、最大1分間の
例えば、セッションタイムアウトを30分に設定した場合、セッション

JSPプリコンパイル設定

Webアプリケーションに含まれるJSPをサーバ起動時にプリコンパイルを行
通常JSPはページがアクセスされた時にコンパイルされますが、JSPプリコン
JSPプリコンパイル設定を行うには、**com.caucho.jsp.JspPrecompileLi**

```
<web-app>
  <listener>
    <listener-class>com.caucho.jsp.JspPrecompileListener</listener-clas
    <init>
      <extension>jsp</extension>
    </init>
  </listener>
</web-app>
```

<extension> タグにはプリコンパイルを行うファイルの拡張子を指定してく
上記の例ではWebアプリケーションに含まれるすべてのjspファイルをプリコ
さらに詳細にプリコンパイル対象を指定する場合は <fileset> タグを指定し

```
<web-app>
<listener>
<listener-class>com.caucho.jsp.JspPrecompileListener</listener-class>
<init>
<fileset>
<include>/**/*.jsp</include>
<exclude>/a.jsp</exclude>
<exclude>/foo/c.jsp</exclude>
</fileset>
</init>
</listener>
</web-app>
```

<include> タグはプリコンパイルの対象とするファイル名、**<exclude>** タグは上記の例でWebアプリケーション内に含まれるすべてのパスのjspファイルを

ヘルス機能

概要

Resinにはサーバの状態を監視するためのヘルス機能が含まれています。PDFレポートを生成することが可能です。

主に、ヘルス機能は「診断」「メーター」「レポート」の機能から成り立っており、以下にそれぞれの機能の設定・使用方法を記述します。

例えば、ヘルス機能の「診断」部分を設定することにより、ヘルス機能はサーバの稼働状況を監視し、その結果をレポートとして生成します。

以下にそれぞれの機能の設定・使用方法を記述します。

/resin-admin

- 項目
- [/resin-adminの設定](#)

/resin-admin WebコンソールはResinサーバの管理を提供します。ユーザは、このWebコンソールを使用して、サーバの管理を行います。

Webコンソールを有効に利用するためには、スレッドダンプ、プロファイル、ヒ

[/resin-adminの設定](#)

/resin-admin を使用にするためには管理者ユーザとパスワードを作成する。

1. コマンドラインから「generate-password」を実行します。この例ではLinuxの例

```
> resinctl generate-password admin my-password
admin_user : admin
admin_password : {SSHA}HTfP60Ceq0K0IAvN31wQtBxtql9D+8W
```

Windowsの例

```
> resin.exe generate-password admin my-password
admin_user : admin
admin_password : {SSHA}HTfP60Ceq0K0IAvN31wQtBxtql9D+8W
```

2. ユーザとパスワードをresin.propertiesファイルに記述します。

```
# Access to /resin-admin and remote CLI is password restricted.
# Use "resinctl generate-password" and copy/paste here to set the
admin_user : admin
admin_password : {SSHA}HTfP60Ceq0K0IAvN31wQtBxtql9D+8W
```

3. ローカル以外のIPアドレスからアクセスする場合はresin.properties

```
# Permit access to /resin-admin from non-local network ip-address
web_admin_external : true
```

4. Resinインスタンスを再起動し、ブラウザから /resin-admin にアクセス。
<http://localhost:8080/resin-admin/>

コラム

CPUグラフ、NetworkグラフはそれぞれLinuxの /proc/stats と /p

診断

設定

ヘルス設定

項目

- 概要
- [health.xml](#)
- デフォルトの設定
 - [ヘルスシステム](#)
 - [ヘルスチェック](#)
 - [ヘルスアクションとヘルスコンディション](#)

概要

ヘルス機能の設定はhealth.xmlに記述します。設定には以下のものが含まれます。

- **ヘルスシステム** - ヘルスシステム本体の設定
- **ヘルスチェック** - システムのチェックを行う設定
- **ヘルスアクション** - タスクを実行する設定（サーバ再起動、ヒープダンプ）
- **ヘルスコンディション** - ヘルスアクションの実行条件を設定

これらの設定を組み合わせ、システムに異常が発生した場合にどのような

[health.xml](#)

health.xmlは以下のパスに配置されています。

- Linuxのパス
 - デフォルトのインストールパスの場合

```
/etc/resin/health.xml
```

- インストールパスを指定した場合

```
%RESIN_HOME%/conf/health.xml
```

- Windowsのパス

```
%RESIN_HOME%/conf/health.xml
```

デフォルトの設定

ヘルス設定に関する各種設定項目には、デフォルトの値が設定されています。以下に各設定を解説します。

[ヘルスシステム](#)

ヘルスシステム本体の設定はデフォルトで以下のように記述されています。

```
<health:HealthSystem>
  <enabled>true</enabled>
  <startup-delay>15m</startup-delay>
  <period>5m</period>
  <recheck-period>30s</recheck-period>
  <recheck-max>5</recheck-max>
  <check-timeout>5m</check-timeout>
</health:HealthSystem>
```

- **enabled** - ヘルスシステムを有効に設定
- **startup-delay** - Resin起動開始からヘルスシステムが監視を始めるまでの遅延時間
- **period** - ヘルスチェックの監視間隔を5分に設定
- **recheck-period** - 異常を検知した場合に、再確認を行う監視間隔を30秒に設定
- **recheck-max** - 再確認を行う回数を5回に設定
- **check-timeout** - ヘルスチェックのタイムアウトを5分に設定

[ヘルスチェック](#)

ヘルスチェックはデフォルトで以下のように設定されています。

- **<health:ConnectionPoolHealthCheck>** - データベースコネクションの監視
- **<health:CpuHealthCheck>** - CPU使用率が異常に上昇していない
- **<health:HealthSystemHealthCheck>** - ヘルスシステムが正常に稼働している

- <health:HeartbeatHealthCheck> - クラスタのハートビートが正常に
- <health:JvmDeadlockHealthCheck> - デッドロックが発生していな
- <health:MemoryPermGenHealthCheck> - JavaVMのPermGenの
- <health:MemoryTenuredHealthCheck> - JavaVMのTenured領域の
- <health:TransactionHealthCheck> - コミットの失敗はないか

i コラム

<health:CpuHealthCheck> のデフォルト設定はヘルスステータス
環境に応じて必要な場合に *critical-threshold* 属性を設定して

ヘルスアクションとヘルスコンディション

ヘルスアクションとヘルスコンディションの設定内容はデフォルトで以下のよ

```
<health:ActionSequence>
  <health:IfHealthCritical time="2m"/>

  <health:FailSafeRestart timeout="10m"/>
  <health:DumpJmx/>
  <health:DumpThreads/>
  <health:ScoreboardReport/>
  <health:DumpHeap/>
  <health:DumpHeap hprof="true"
    hprof-path="{resin.logDirectory}/heap.hprof"/>
  <health:StartProfiler active-time="2m" wait="true"/>
  <health:Restart/>
</health:ActionSequence>
```

この設定はヘルスシステムがResinの異常を検知した場合、様々な情報を取
以下にその解説を記述します。

1. もしヘルスチェックのどれかひとつでも ヘルスステータス「CRITICAL
2. 情報収集に遅延が発生した場合のために、強制再起動を行うまでの
3. JMXダンプを生成
4. スレッドダンプを生成
5. スコアボードレポートを生成
6. ヒープダンプを生成
7. hprofフォーマットのヒープダンプを生成
8. プロファイラを開始し、2分間分のサーバ情報を生成
9. Resinインスタンスの再起動を実行

i コラム

<health:IfHealthCritical time="2m"/>は、ヘルスシステムが行う!
まず5分間隔で監視を行い、ヘルスステータス「CRITICAL」を検!
もし、2分間「CRITICAL」のままであった場合Resinの再起動が行

i コラム

プロファイルはResinの制限によりPDFレポートに出力されませA
PDFレポートの「CPU Profile」には「A CPU profile was not ger

また、デフォルトで以下のヘルスアクションが設定されています。

```
<health:Restart>
  <health:IfHealthFatal/>
</health:Restart>
```

これは、ヘルスチェックのどれかひとつでもヘルスステータス「FATAL」を返
デフォルトの設定では下記のヘルスチェックが「FATAL」を返します。

- <health:HealthSystemHealthCheck> - ヘルスシステム自体が異常な
- <health:JvmDeadlockHealthCheck> - JavaVMによってデッドロックが

リファレンス

ヘルスシステム

項目
• <health:HealthSystem> ◦ 属性

<health:HealthSystem>

ヘルスシステムのチェックや再チェックの頻度を設定します。この要素はデフ

属性名	説明
enabled	ヘルスシステムの有効/無効。
startup-delay	サーバ起動時にチェックの開始を遅延させる時間
period	チェックの間隔。
recheck-period	再チェックの間隔。
recheck-max	通常のチェック間隔に戻るまでに行う再チェックの回数。

 コラム

ヘルスチェックの監視間隔を極端に短く設定にすると、環境によ

ヘルスチェック

項目

- ヘルスステータス
- システム チェック
 - <health:ConnectionPoolHealthCheck>
 - <health:CpuHealthCheck>
 - <health:HealthSystemHealthCheck>
 - <health:HeartbeatHealthCheck>
 - <health:JvmDeadlockHealthCheck>
 - <health:LicenseHealthCheck>
 - <health:MemoryPermGenHealthCheck>
 - <health:MemoryTenuredHealthCheck>
 - <health:TransactionHealthCheck>
- ユーザ チェック
 - <health:HttpStatusHealthCheck>
 - <health:ExprHealthCheck>

ヘルスステータス

ヘルスチェックは実行されるたびにヘルスステータスとメッセージを生成しま
下記はヘルスステータスと一般的な意味のリストです。

名前	順序	説明
UNKNOWN	0	ヘルスチェックはまだ実行されていない、または失敗しています。
OK	1	ヘルスチェックは正常な状態であることを報告
WARNING	2	ヘルスチェックは警告の域に達しているか、重要な域に達している
CRITICAL	3	ヘルスチェックは重要な域に達していることを報告
FATAL	4	ヘルスチェックは致命的な状態であることを報告

システム チェック

<health:ConnectionPoolHealthCheck>

Resinのデータベースコネクションプールの状態を監視します。

属性

属性名	説明	型	デフォルト
enabled	チェックの有無	boolean	true

状態

ヘルスステータス	状態
WARNING	resin-web設定 の <max-connections> を超過した場合
CRITICAL	resin-web設定 の <max-overflow-connections> を超過した場合

```
<health:ConnectionPoolHealthCheck/>
```

```
<health:CpuHealthCheck>
```

CPU使用率を監視します。マルチコアマシンでは、各CPUが個別にチェック

属性名	説明
enabled	チェックの有無
warning-threshold	ヘルスステータス「WARNING」のしきい値
critical-threshold	ヘルスステータス「CRITICAL」のしきい値

状態

ヘルスステータス	状態
WARNING	CPUのどれかが <i>warning-threshold</i> を超過した場合。
CRITICAL	CPUのどれかが <i>critical-threshold</i> を超過した場合。

設定例

```
<health:CpuHealthCheck>
  <warning-threshold>95</warning-threshold>
  <critical-threshold>99</critical-threshold>
</health:CpuHealthCheck>
```

<health:HealthSystemHealthCheck>

ヘルスシステム自体を監視します。
独立したスレッドを使用してヘルスチェックがフリーズしていないか、またはB

属性

属性名	説明
enabled	チェックの有無
thread-check-period	独立したスレッドの確認頻度。
freeze-timeout	ヘルスシステムがフリーズしてヘルスチェックが中断するまでの最大時間。

状態

ヘルスステータス	状態
FATAL	ヘルスチェックが <i>freeze-timeout</i> の期間
FATAL	ヘルスチェックが許容される期間内に完了し、出されます。

設定例

```
<health:HealthSystemHealthCheck>
  <thread-check-period>1m</thread-check-period>
  <freeze-timeout>15m</freeze-timeout>
</health:HealthSystemHealthCheck>
```

<health:HeartbeatHealthCheck>

クラスタの他のメンバーからのハートビートを監視します。

属性

属性名	説明	型	デフォルト
enabled	チェックの有無	boolean	true

状態

ヘルスステータス	状態
WARNING	クラスタの既知のメンバーからのハートビートが受
WARNING	クラスタの既知のメンバーからのハートビートが最

設定例

<health:HeartbeatHealthCheck/>

<health:JvmDeadlockHealthCheck>

JavaVMによってデッドロック状態であると認識されたスレッドを監視します。

属性

属性名	説明	型	デフォルト
enabled	チェックの有無	boolean	true

状態

ヘルスステータス	状態
FATAL	デッドロックのスレッドが検出された場合。

設定例

<health:JvmDeadlockHealthCheck/>

<health:LicenseHealthCheck>

Resin-Proのライセンスの期限切れをチェックします。

属性

属性名	説明	型	デフォルト
enabled	チェックの有無	boolean	true
warning-period	ライセンス警告の期間	期間	30日 (30D)

状態

ヘルスステータス	状態
WARNING	ライセンスが <i>warning-period</i> の期間内に期限切れにな

設定例

<health:LicenseHealthCheck>
<warning-period>30D</warning-period>
</health:LicenseHealthCheck>

<health:MemoryPermGenHealthCheck>

JavaVMのPermGenメモリプールの空き領域を監視します。空き領域が低

i

コラム

このヘルスチェックは、JavaVMベンダの実装によっては使用できず、JavaVMにPermGenメモリプールが存在しない場合はヘルスステータスがOKになります。

属性

属性名	説明
enabled	チェックの有無
memory-free-min	クリティカルな状況の空きメモリ
free-warning	警告のしきい値
objectName	明示的にメモリの統計を照会するMBeanを検索します。

状態

ヘルスステータス	状態
----------	----

ヘルスステータス	状態
UNKNOWN	JavaVMにPermGenメモリプールが存在しない場合。
WARNING	空きメモリが、ガベージコレクション実行後に <i>free</i>
CRITICAL	空きメモリが、ガベージコレクション実行後に <i>me</i>

設定例

```
<health:MemoryTenuredHealthCheck>
  <memory-free-min>1m</memory-free-min>
  <free-warning>0.01</free-warning>
</health:MemoryTenuredHealthCheck>
```

<health:MemoryTenuredHealthCheck>

JavaVMのTenuredメモリプールの空き領域を監視します。空き領域が低下

 **コラム**

JavaVMにTenuredメモリプールが存在しない場合はJavaVMで

属性

属性名	説明
enabled	チェックの有無
memory-free-min	クリティカルな状況の空きメモリ
free-warning	警告のしきい値
objectName	明示的にメモリの統計を照会す MBeanを検索します。

状態

ヘルスステータス	状態
UNKNOWN	JavaVMにTenuredメモリプールが存在しない場 合。
WARNING	空きメモリが、ガベージコレクション実行後に <i>free</i>
CRITICAL	空きメモリが、ガベージコレクション実行後に <i>me</i>

設定例

```
<health:MemoryTenuredHealthCheck>
  <memory-free-min>1m</memory-free-min>
  <free-warning>0.01</free-warning>
</health:MemoryTenuredHealthCheck>
```

<health:TransactionHealthCheck>

Resinのトランザクションマネージャのコミット失敗を監視します。

属性

属性名	説明	型	デフォルト
enabled	チェックの有無	boolean	true

状態

ヘルスステータス	状態
WARNING	前回のチェック以降にコミット失敗があった場合。

設定例

```
<health:TransactionHealthCheck/>
```

ユーザ チェック

<health:HttpStatusHealthCheck>

1つ以上のURLを作成し、現在のResinインスタンスにHTTP GETのリクエン

属性

属性名	説明
enabled	チェックの有無
ping-host	チェック対象のホスト (<i>url</i> が URI の場合に
ping-port	チェック対象のポート (<i>url</i> が URI の場合に
url	チェック対象のURLまたはURI
socket-timeout	ソケット接続のタイムアウト時間
regex	HTTPステータスコードの正規表現

状態

ヘルスステータス	状態
CRITICAL	HTTP GET リクエストが接続に失敗した場合。ま

設定例

```
<health:HttpStatusHealthCheck>
  <ping-host>localhost</ping-host>
  <ping-port>8080</ping-port>
  <url>/custom-test-1.jsp</url>
  <url>/custom-test-2.jsp</url>
  <socket-timeout>2s</socket-timeout>
  <regex>^2|^3</regex>
</health:HttpStatusHealthCheck>
```

<health:ExprHealthCheck>

ユーザーが指定したEL式の真偽を評価します。

属性

属性名	説明	型
enabled	チェックの有無	bc
fatal-test	trueと評価された場合にステータスをFATAL するEL式。	El
critical-test	trueと評価された場合にステータスをCRITICALするEL式。	El
warning-test	trueと評価された場合にステータスをWARNINGするEL式。	El

状態

ヘルスステータス	状態
FATAL	<i>fatal-test</i> がtrueと評価された場合。
CRITICAL	<i>critical-test</i> がtrueと評価された場合。
WARNING	<i>warning-test</i> がtrueと評価された場合。

設定例

```
<health:ExprHealthCheck>
  <critical-test>${mbean('java.lang:type=Threading').ThreadCount > 100}
</health:ExprHealthCheck>
```

ヘルスアクション

- 項目
- <health:ActionSequence>
 - <health:CallJmxOperation>
 - <health:DumpHeap>
 - <health:DumpHprofHeap>
 - <health:DumpJmx>
 - <health:DumpThreads>
 - <health:ExecCommand>

- [<health:FailSafeRestart>](#)
- [<health:PdfReport>](#)
- [<health:Restart>](#)
- [<health:ScoreboardReport>](#)
- [<health:SendMail>](#)
- [<health:Snapshot>](#)
- [<health:StartProfiler>](#)

ヘルスアクションは特定の条件に応じてタスクを実行、またはヘルスチェックムによって実行されます。

ヘルスアクションは通常、1つ以上のヘルスコンディションを伴いますが必須があります。コンディションが無いヘルスアクションは、期間ごとに一度実行

[<health:ActionSequence>](#)

子に設定されているヘルスアクションを順に実行します。

属性

なし

設定例

```
<health:ActionSequence>
  <health:DumpThreads/>
  <health:DumpHeap/>
  <health:IfHealthCritical time="5m"/>
</health:ActionSequence>
```

[<health:CallJmxOperation>](#)

パラメータを指定して、JMX MBeanのオペレーションを実行します。

属性

属性名	説明
objectName	JMX MBeanの名前。
operation	メソッドの名前。
operationIndex	複数のメソッドが一致する場合のユニークなインデックス。
param	適切な型に変換されるメソッドのパラメータ。

設定例

```
<health:CallJmxOperation>
  <objectName>java.lang:type=Threading</objectName>
  <operation>resetPeakThreadCount</operation>
  <health:IfNotRecent time='5m'/>
</health:CallJmxOperation>
```

<health:DumpHeap>

メモリヒープダンプを生成します。

ヒープダンプは、Resinの内部ログデータベースとログファイルに記録されます。

属性

属性名	説明
hprof	ユーザが読めるタイプのダンプではなく、HPROFフォーマット。
hprof-path	HPROFファイルの出力パス。(hprof がtrueの場合)
hprof-path-format	タイムスタンプのトークンを使用して、動的に出力パスを生成します。
log	ヒープダンプをResinの内部データベースに格納されます。

設定例

```
<health:DumpHeap>
  <hprof>true</hprof>
  <hprof-path-format>${resin.home}/log/dump-%H:%M:%S.%s.hprof</hprof-path-format>
  <health:OnAbnormalStop/>
</health:DumpHeap>
```

<health:DumpHprofHeap>

以下の設定と同様の意味をもつヘルスアクションです。

```
<health:DumpHeap>
  <hprof>true</hprof>
</health:DumpHeap>
```

<health:DumpJmx>

すべてのJMX属性および値のダンプを生成するヘルスアクションです。

JMXダンプは、Resinの内部ログデータベースとログファイルに記録されます。

属性

属性名	説明
log	JMXダンプをResinの内部データベースに格納され、ログファイルに書き込みます。

設定例

```
<health:DumpJmx/>
```

<health:DumpThreads>

スレッドダンプを生成します。

スレッドダンプは、Resinの内部ログデータベースとログファイルに記録されます。

属性

属性名	説明
only-active	現在アクティブなスレッドのみ出力します。(RUNNAB
log	スレッドダンプをResinの内部データベースに格納され書き込みます。

設定例

```
<health:DumpThreads>
  <only-active>false</only-active>
</health:DumpThreads>
```

<health:ExecCommand>

オペレーティングシステムのシェルコマンドを実行します。

属性

属性名	説明
command	実行するコマンド。 <i>dir</i> が設定されている場合は、
dir	実行するディレクトリのパス。
timeout	コマンド実行のタイムアウト。完了していない場合
env	コマンドに使用する環境変数。

設定例

```
<health:ExecCommand>
  <dir>/tmp</dir>
  <command>remediation.sh</command>
  <timeout>2s</timeout>
  <env>
    <name>resin_home</name>
    <value>${resin.home}</value>
  </env>
  <env>
    <name>password</name>
    <value>foo</value>
  </env>
</health:ExecCommand>
```

<health:FailSafeRestart>

Resinを設定した時間に再起動します。通常、シャットダウン情報を集めるた

属性

属性名	説明	型	デフォルト
timeout	再起動を強制実行するまでの時間。	期間	なし

設定例

```
<health:ActionSequence>
  <health:FailSafeRestart timeout="10m"/>
  <health:DumpThreads/>
  <health:DumpHeap/>
  <health:StartProfiler active-time="5m"/>
  <health:Restart/>

  <health:IfHealthCritical time="5m"/>
</health:ActionSequence>
```

<health:PdfReport>

PHPスクリプトからPDFレポートを生成するヘルスアクションです。

属性

属性名	説明	型	デフォルト
path	PDFを生成する.phpファイルのパス。	String	\$

属性名	説明	型	デフォルト
report	レポートタイプのキー。	String	Summary
period	レポートに出力する過去期間。	期間	7D
log-directory	PDFを出力するディレクトリ。	String	resin.log

設定例

```
<health:PdfReport>
  <path>${resin.home}/doc/admin/pdf-gen.php</path>
  <report>Summary</report>
  <period>7D</period>
  <health:IfCron value="0 0 * * 0"/>
</health:PdfReport>
```

<health:Restart>

Resinを再起動します。

属性

なし

設定例

```
<health:Restart>
```

<health:ScoreboardReport>

関連したスレッドのグループについて簡潔なスレッドアクティビティレポートを生成します。

属性

属性名	説明	型	デフォルト
log	PDFレポートに加えて、サーバログに書き込みます。	boolean	false
type	スコアボードレポートのタイプ。	String	greedy
greedy	falseの場合、スレッドが複数のグループに分類されます。	boolean	false

設定例

```
<health:ScoreboardReport>
  <health:OnAbnormalStop/>
</health:ScoreboardReport>
```

<health:SendMail>

現在のResinのヘルスステータスの概要を記載したメールを送信します。

属性

属性名	説明	型	デフォルト
to	メールの送信先アドレス	String	なし
from	メールの送信元アドレス	String	resin@localhost

設定例

```
<health:SendMail>
  <to>admin@yourdomain.com</to>
  <to>another_admin@yourdomain.com</to>
  <from>resin@yourdomain.com</from>
</health:SendMail>
```

<health:Snapshot>

以下に示す一連のアクションを実行します。

- スレッドダンプ
- ヒープダンプ
- JMXダンプ
- PDFレポート

これらはサーバの問題をデバッグするために必要な全ての情報を想定して生成されます。

属性

属性名	説明	型
log	PDFレポートに加えて、サーバ・ログに書き込みます。	boolean
path	PDFを生成する.phpファイルのパス。	String
report	レポートタイプのキー。	String
period	レポートに出力する過去期間。	期間

設定例

```
<health:Snapshot>
  <health:OnAbnormalStop/>
</health:Snapshot>
```

<health:StartProfiler>

プロファイラ開始します。結果は、Resinの内部ログデータベースとログファイ

属性

属性名	説明	型	デフォルト
active-time	プロファイラを実行する時間	期間	5秒 (5s)
sampling-rate	サンプリングレート	期間	10ミリ秒 (10ms)
depth	スタックトレースの深さ。	int	16

設定例

```
<health:ActionSequence>
  <health:FailSafeRestart timeout="10m"/>
  <health:DumpThreads/>
  <health:DumpHeap/>
  <health:StartProfiler active-time="5m"/>
  <health:Restart/>

  <health:IfHealthCritical time="5m"/>
</health:ActionSequence>
```

ヘルスコンディション

項目

- 基本条件
 - <health:IfCron>
 - <health:IfExpr>
 - <health:IfNotRecent>
 - <health:IfRechecked>
 - <health:IfUptime>
- 組み合わせ条件
 - <health:And>
 - <health:Nand>
 - <health:Nor>
 - <health:Not>
 - <health:Or>
- ヘルスチェック条件
 - <health:IfHealthOk>
 - <health:IfHealthWarning>
 - <health:IfHealthCritical>
 - <health:IfHealthFatal>
 - <health:IfHealthUnknown>
 - <health:IfMessage>
 - <health:IfRecovered>
- ライフサイクル条件
 - <health:OnStart>
 - <health:OnStop>

- `<health:OnAbnormalStop>`
- `<health:OnRestart>`

コンディション(条件)またはプレディケート(叙述)は、一連の基準に基づいてコンディションは期間ごとに評価されます。すべての条件が真と評価されないコンディションが無いアクションは期間毎に一度実行されます。複数のコンラ

基本条件

基本条件は一般的な基準を評価し条件がマッチした場合にtrueを返します。基本条件はヘルスチェックの状態を評価しません。代わりに、その日の時間

`<health:IfCron>`

現在時刻が **cronスタイル** で設定された範囲だった場合にマッチします。これは、定期的なアクションをスケジュールしたり、重要な時間帯に再起動な

属性

属性名 説明		型	デフォルト
enable-at	有効とするcron時間。	cronスタイル	なし
disable-at	無効とするcron時間。	cronスタイル	なし

設定例

```
<health:Restart>
<health:IfCron>
  <enable-at>0 0 * * * </enable-at>
  <disable-at>5 0 * * * </disable-at>
</health:IfCron>
</health:Restart>
```

`<health:IfExpr>`

JSP EL式の評価に基づいてマッチします。式は、システムプロパティ、コンフ

属性

		デフォルト	
属性名	説明	型	
test	JSP EL式。	EL式	なし

設定例

```
<health:Restart>
<health:IfExpr>
  <test>${mbean("java.lang:type=Threading").ThreadCount > 100}</test>
</health:IfExpr>
</health:Restart>
```

<health:IfNotRecent>

前回の実行後からの時間の長さに応じてマッチします。これは頻繁な実行カ

属性

		デフォルト	
属性名	説明	型	
time	再びアクションを実行できるまでの時間。	期間	なし

設定例

```
<health:HttpStatusHealthCheck ee:Named="httpStatusCheck">
  <url>http://localhost:8080/test-ping.jsp</url>
</health:HttpStatusHealthCheck>

<health:DumpHeap>
<health:IfHealthCritical healthCheck="${httpStatusCheck}"/>
<health:IfNotRecent time='5m'/>
</health:DumpHeap>
```

<health:IfRechecked>

再チェックの必要回数が実行されている場合にマッチします。
再チェックはヘルスチェック固有の条件ではないので、HealthSystem のパ

属性

なし

設定例

```
<health:Restart>
<health:IfHealthFatal/>
<health:IfRechecked/>
</health:Restart>
```

<health:IfUptime>

起動してから時間の長さに応じてマッチします。

属性

		デフォルト	
属性名	説明	型	
limit	起動後の時間。	期間	なし

設定例

```
<health:Restart>
<health:IfUptime limit="12h"/>
</health:Restart>
```

組み合わせ条件

基本条件やヘルスチェック条件を組み合わせたり、これらの条件を使用して

<health:And>

全ての子のプレディケートがマッチしている場合にマッチします。

属性

なし

設定例

```
<health:Restart>
<health:And>
  <health:IfHealthCritical health-check="{memoryTenuredHealthCheck}">
  <health:IfHealthCritical health-check="{memoryPermGenHealthCheck}">
</health:And>
</health:Restart>
```

<health:Nand>

全ての子のプレディケートが失敗している場合にマッチします。

属性

なし

設定例

```
<health:Restart>
<health:Nand>
  <health:IfHealthCritical health-check="{memoryTenuredHealthCheck}">
  <health:IfHealthCritical health-check="{memoryPermGenHealthCheck}">
</health:Nand>
</health:Restart>
```

<health:Nor>

全ての子のプレディケートが一致しない場合にマッチします。

属性

なし

設定例

```
<health:Restart>
<health:Nor>
  <health:IfHealthCritical health-check="{memoryTenuredHealthCheck}">
  <health:IfHealthCritical health-check="{memoryPermGenHealthCheck}">
</health:Nor>
</health:Restart>
```

<health:Not>

子のプレディケートがfalseの場合にマッチします。

属性

なし

設定例

```
<health:Restart>
<health:IfHealthCritical health-check="{memoryTenuredHealthCheck}">
<health:Not>
  <health:IfCron>
    <enable-at>0 7 * * * </enable-at>
    <disable-at>0 11 * * * </disable-at>
  </health:IfCron>
</health:Not>
</health:Restart>
```

<health:Or>

子のプレディケートのいずれかがマッチしている場合にマッチします。

属性

なし

設定例

```
<health:Restart>
<health:Or>
  <health:IfHealthCritical health-check="{memoryTenuredHealthCheck}" />
  <health:IfHealthCritical health-check="{memoryPermGenHealthCheck}" />
</health:Or>
</health:Restart>
```

ヘルスチェック条件

ヘルスチェック条件は、ヘルスチェックの結果をあらゆる側面から評価します。パラメータ「health-check」で特定の名前が付けられたヘルスチェックを参照します。

<health:IfHealthOk>

ヘルスステータスが「OK」の場合にマッチします。

属性

属性名	説明
health-check	参照するヘルスチェック。
time	開始状態からの最少時間。

設定例

```
<health:Restart>
<health:Not>
  <health:IfHealthOk health-check="{memoryTenuredHealthCheck}" />
</health:Not>
</health:Restart>
```

<health:IfHealthWarning>

ヘルスステータスが「WARNING」の場合にマッチします。

属性

属性名	説明
health-check	参照するヘルスチェック。
time	開始状態からの最少時間。

設定例

```
<health:Restart>
<health:IfHealthWarning health-check="{memoryTenuredHealthCheck}" />
</health:Restart>
```

<health:IfHealthCritical>

ヘルスステータスが「CRITICAL」の場合にマッチします。

属性

属性名	説明
health-check	参照するヘルスチェック。
time	開始状態からの最少時間。

設定例

```
<health:Restart>
<health:IfHealthCritical health-check="{memoryTenuredHealthCheck}" />
</health:Restart>
```

<health:IfHealthFatal>

ヘルスステータスが「FATAL」の場合にマッチします。

属性

属性名	説明
health-check	参照するヘルスチェック。
time	開始状態からの最少時間。

設定例

```
<health:Restart>
<health:IfHealthFatal health-check="{memoryTenuredHealthCheck}" />
</health:Restart>
```

<health:IfHealthUnknown>

ヘルスステータスが「UNKNOWN」の場合にマッチします。

属性

属性名	説明
health-check	参照するヘルスチェック。
time	開始状態からの最少時間。

設定例

```
<health:Restart>
<health:IfHealthUnknown health-check="{memoryTenuredHealthCheck}" />
</health:Restart>
```

<health:IfMessage>

ヘルスチェック結果のメッセージが正規表現に一致する場合にマッチします。

属性

属性名	説明
health-check	参照するヘルスチェック。
time	開始状態からの最少時間。
regex	ヘルスメッセージにマッチする正規表現。

設定例

```
<health:Restart>
<health:IfHealthCritical />
<health:IfMessage health-check="{httpStatusCheck}" regex="Not Found" />
</health:Restart>
```

<health:IfRecovered>

リカバリされた際にマッチします。リカバリとは「FATAL」、「CRITICAL」、「WARN」の状態から「OK」の状態になることを指します。

属性

属性名	説明
health-check	参照するヘルスチェック。

設定例

```
<health:SendMail>
  <to>admin@yourdomain</to>
  <health:IfRecovered health-check="{cpuHealthCheck}"/>
</health:SendMail>
```

ライフサイクル条件

ライフサイクル条件はResinの状態変化を評価します。Resinのライフサイク

<health:OnStart>

Resinが開始しようとしている場合にマッチします。

属性

なし

設定例

```
<health:SendMail>
  <to>admin@yourdomain.com</to>
  <health:OnStart/>
</health:SendMail>
```

<health:OnStop>

Resinが停止された場合にマッチします。

属性

なし

設定例

```
<health:SendMail>
  <to>admin@yourdomain.com</to>
  <health:OnStop/>
</health:SendMail>
```

<health:OnAbnormalStop>

Resinが正常ではない終了コードを返して停止している場合にマッチします。

属性

なし

設定例

```
<health:PdfReport snapshot='true'>
  <health:OnAbnormalStop/>
</health:PdfReport>
```

<health:OnRestart>

Resinがwatchdogによって再起動された場合にマッチします。これは一般的

属性

なし

設定例

```
<health:SendMail>
  <to>admin@yourdomain.com</to>
  <health:OnRestart/>
</health:SendMail>
```

設定値

期間

単位	意味
ms	ミリ秒
s	秒
m	分
h	時
D	日
W	週
M	月
Y	年

メーター

項目

- 設定
 - health.xml
 - ヘルスマーター名
- JMXメーター
 - <health:JmxMeter>
 - <health:JmxDeltaMeter>
- 統計分析
 - <health:AnomalyAnalyzer>
 - <health:IfHealthEvent>

ヘルスマーターはResinの管理コンソール(/resin-admin)に視覚的にわかり

設定

health.xml

health.xmlにはあらかじめJMXメーターが設定されています。health.xmlの

ヘルスマーター名

ヘルスマーター名はキーをパイプ文字(|)で連結して名前が付けられています。ヘルスマーターの統計はResinクラスタの各メンバー間で共有されているため、名前に含まれるパイプ文字は、ヘルスマーターをドリルダウンに分類して管理

```
<health:JmxDeltaMeter>
  <name>JVM|Compilation|Compilation Time</name>
  <object-name>java.lang.type=Compilation</object-name>
  <attribute>TotalCompilationTime</attribute>
</health:JmxDeltaMeter>
```

この例では **JVM|Compilation|Compilation Time** をベースにヘルスマーター管理コンソールではクラスタメンバーのインデックスと最初の2つのキーを使

```
00|JVM|Compilation|Compilation Time
```

JMXメーター

JMX MBean属性の数値をヘルスマーターにグラフ表示することができます。

<health:JmxMeter>

JMX MBeanの数値属性の現在値をグラフに表示します。

属性

属性名	説明	型
name	管理コンソールに表示するヘルスマーターの名前	String
objectName	JMX MBean の名前	String
attribute	サンプリングするMBeanの属性	String

設定例

```
<health:JmxMeter>
  <name>OS|Memory|Physical Memory Free</name>
  <object-name>java.lang:type=OperatingSystem</object-name>
  <attribute>FreePhysicalMemorySize</attribute>
</health:JmxMeter>
```

<health:JmxDeltaMeter>

JMX MBeanの数値属性の現在値と以前の値の差をグラフに表示します。

属性

属性名	説明	型
name	管理コンソールに表示するヘルスメーターの名前	String
objectName	JMX MBean の名前	String
attribute	サンプリングするMBeanの属性	String

設定例

```
<health:JmxDeltaMeter>
  <name>JVM|Compilation|Compilation Time</name>
  <object-name>java.lang.type=Compilation</object-name>
  <attribute>TotalCompilationTime</attribute>
</health:JmxDeltaMeter>
```

統計分析

<health:AnomalyAnalyzer>

AnomalyAnalyzerは平均値からの偏差をチェックし現在の値を監視します。突然スレッドがブロックされた場合など異常な変化を検出することができま

属性

属性名	説明
meter	分析するヘルスメーターの名前 (<health:名前>)
health-event	<health:IfHealthEvent>の条件に一致さ
min-samples	平均値を計算するために必要なサンプル
sigma-threshold	標準偏差から異常とみなされるサンプル

設定例

```
<health:JmxMeter>
  <name>JVM|Thread|JVM Blocked Count</name>
  <objectName>resin:type=JvmThreads</objectName>
  <attribute>BlockedCount</attribute>
</health:JmxMeter>

<health:AnomalyAnalyzer>
  <meter>JVM|Thread|JVM Blocked Count</meter>
  <health-event>caucho.thread.anomaly.jvm-blocked</health-event>
</health:AnomalyAnalyzer>

<health:DumpThreads>
  <health:IfHealthEvent regexp="caucho.thread"/>
  <health:IfNotRecent time="15m"/>
</health:DumpThreads>
```

<health:IfHealthEvent>

一致するヘルスイベントに応じてアクションを起こします。これは、通常、<he

属性

属性名	説明	型
regexp	<health-event> が一致する正規表現。	java.util.regex.f

設定例

```

<health:JmxMeter>
  <name>JVM|Thread|JVM Blocked Count</name>
  <objectName>resin:type=JvmThreads</objectName>
  <attribute>BlockedCount</attribute>
</health:JmxMeter>

<health:AnomalyAnalyzer>
  <meter>JVM|Thread|JVM Blocked Count</meter>
  <health-event>caucho.thread.anomaly.jvm-blocked</health-event>
</health:AnomalyAnalyzer>

<health:DumpThreads>
  <health:IfHealthEvent regexp="caucho.thread"/>
  <health:IfNotRecent time="15m"/>
</health:DumpThreads>

```

レポート

- 項目
- レポートの取得
 - 自動的なPDFレポートの生成
 - レポートの概要
 - ヒープダンプ
 - クラスローダーのヒープダンプ
 - スレッドダンプ
 - CPUプロファイル
 - ロギング
 - JMXダンプ

Resin にはシステムの状態をPDFレポートに出力する機能が含まれています。

レポートの取得

レポートを取得する簡単な方法は、コマンド「pdf-report」を使用することです。

Linuxの例

```
> resinctl pdf-report
```

Windowsの例

```
> resin.exe pdf-report
```

自動的なPDFレポートの生成

PDFレポートは毎週生成されるようにhealth.xmlに設定することができます。

```

<health:PdfReport>
  <path>${resin.root}/doc/admin/pdf-gen.php</path>
  <report>Summary</report>
  <period>7D</period>
  <snapshot/>
  <mail-to>${email}</mail-to>
  <mail-from>${email_from}</mail-from>
  <!-- <profile-time>60s</profile-time> -->

  <health:IfCron value="0 0 * * 0"/>
</health:PdfReport>

```

上の例では、スナップショットを作成して、PDFレポートを生成し指定したメー

送信先のメールアドレスはresin.propertiesに記述します。

```
# Set the email address to receive weekly and restart PDF reports
email : admin@example.com
```

次の例では、watchdogによる再起動時にPDFファイルを生成します。

```

<health:PdfReport>
  <path>${resin.root}/doc/admin/pdf-gen.php</path>
  <title>Restart</title>
  <watchdog/>
  <mailto>user@example.com</mailto>

  <health:OnRestart/>
</health:PdfReport>

```

レポートの概要

以下の内容がレポートに出力されます。

- **サーバ環境の概要** - JavaVM、OS、Resinのバージョンなど
- **ヘルスメーターのグラフ** - レポートが出力されるまでの統計グラフ
- **ヒープダンプ** - ヒープメモリ使用量の上位
- **CPUプロファイル** - CPUプロファイルの最も活動的な位置
- **スレッドダンプ** - スナップショット取得時のResin内の全てのスレッド
- **ログレポート** - java.util.loggingから出力された最新の「warning」レベル
- **JMXダンプ** - スナップショット取得時の全てのJMX Mbeanデータ

レポートはいくつかの状況で使用するよう設計されています。

- **Resinのバグレポート** - Resin自体のサポートのために、スナップショット
- **再起動やクラッシュの分析** - JavaVMが何らかの理由でクラッシュした
- **CPUとパフォーマンス** - JavaVMが予想外にCPUを使用した場合、レスポンス
- **スレッドのロック** - スレッドダンプおよびCPUは、ロックされたコードを突

ヒープダンプ

ヒープダンプは、システムのメモリ割り当ての概要を示します。異常なメモリ使用

ヒープダンプはオブジェクト自身のサイズとその子孫のサイズで「self+desc

以下の例は、一般的なアイドル状態のResinのヒープダンプです。

Class Name	self+desc self count		
byte[]	22.23M	22.23M	14741
com.caucho.db.block.Block		20.28M	134.9K 2410
char[]	13.89M	13.89M	122606
com.caucho.util.LruCache		7.52M	30.4K 317
java.lang.String	7.15M	1.97M	61426
..			

最初にあるbyte[]とBlockは、主にResinの内部プロキシキャッシュと分散されたbyte[]バッファを持っているためです。

同様に、char[]バッファを含んでいるため、java.lang.Stringの「self+desc」は

クラスローダーのヒープダンプ

ヒープダンプには、クラスローダーのメモリ使用量を個別に表示する項目がある。クラスに基づいたメモリーリークをチェックするためにこのレポートを使用する。

スレッドダンプ

CPUに関する問題やスタックスレッドの場合、スレッドダンプを確認すれば各スレッドが異常な個所で連なりロック状態になっているかをレポート

スレッドダンプのレポートは、同様のスタックトレースを有しているスレッドを

ブロックされたスレッドとロックを保持しているスレッドがグループ化されてい

以下の例は正常なブロック状態を示しています。JDKのSSL実装は1つのスレッド。この場合「http://*:8444-17」という名前のスレッドがSocksSocketImplを用いて

```

http://*:8444-17
java.net.PlainSocketImpl.socketAccept
-- locked java.net.SocksSocketImpl@1199747469
java.net.PlainSocketImpl.accept
java.net.ServerSocket.implAccept
...
com.caucho.env.thread.ResinThread.runTasks
com.caucho.env.thread.ResinThread.run

http://*:8444-1
waiting on java.net.SocksSocketImpl@4782b18d owned by [126] http://*
http://*:8444-10
waiting on java.net.SocksSocketImpl@4782b18d owned by [126] http://*
http://*:8444-11
waiting on java.net.SocksSocketImpl@4782b18d owned by [126] http://*
java.net.PlainSocketImpl.accept
java.net.ServerSocket.implAccept
com.sun.net.ssl.internal.ssl.SSLServerSocketImpl.accept
...
com.caucho.env.thread.ResinThread.runTasks
com.caucho.env.thread.ResinThread.run

...

```

CPUプロファイル

Resinの制限により、CPUプロファイルをPDFレポートに出力することはできず、PDFレポートの「CPU Profile」には「A CPU profile was not generated」が表示されます。

ロギング

最新の警告ログが出力されます。

Log(Warning)

2011-09-21 11:06:07 warning WarningService: Resin restarting due to configuration change

JMXダンプ

JMXダンプはシステム内のすべてのJMX MBeanとその値を出力します。

JMX Dump

```

JMImplementation:type=MBeanServerDelegate
ImplementationName      Resin-JMX
ImplementationVendor    Caucho Technology
ImplementationVersion    Resin-4.0.s110921
MBeanServerId           Resin-JMX
SpecificationName        Java Management Extensions
SpecificationVendor      Sun Microsystems
SpecificationVersion     1.4

```

```

com.sun.management:type=HotSpotDiagnostic
...

```

watchdog

項目

- [概要](#)
- [コマンドライン](#)
 - [console](#)
 - [start](#)
 - [stop](#)
 - [status](#)

信頼性とセキュリティを高めるために、独立したResin watchdogプロセスに合、Resinのインスタンスを再起動します。

概要

watchdogは独立したサービスとして密かに実行されるため、ほとんどの場合、ユーザーはwatchdogの存在に気づきません。ただし、すべてのResin JavaVMを監視します。

ユーザがwatchdogに注意を払う必要がある場合は、環境に深刻な問題が

Resinがクラッシュしたり予期せずプロセスが終了した場合、watchdogは自

必要があります。

コマンドラインから起動、停止、再起動を行ってください。

コマンドライン

console

「console」コマンドは開発時のために使用します。コンソールウィンドウに新

Linuxの例

```
> resinctl console -server app-0
```

Windowsの例

```
> resin.exe console -server app-0
```

start

「start」コマンドは指定されたIDの新しいResinインスタンスを起動します。リ

Linuxの例

```
> resinctl start -server app-0
Resin/4.0.xx started -server 'app-0' with watchdog at 127.0.0.1:6600
```

Windowsの例

```
> resin.exe start -server app-0
Resin/4.0.xx started -server 'app-0' with watchdog at 127.0.0.1:6600
```

stop

「stop」コマンドは指定されたIDのResinインスタンスを停止します。管理す

Linuxの例

```
> resinctl stop -server app-0
Resin/4.0.xx stopped for watchdog at 127.0.0.1:6600
```

Windowsの例

```
> resin.exe stop -server app-0
Resin/4.0.xx stopped for watchdog at 127.0.0.1:6600
```

status

「status」コマンドはwatchdogサービスによって管理されている現在のResi

Linuxの例

```
> resinctl status
Resin/4.0.xx status for watchdog at 127.0.0.1:6600

watchdog:
  watchdog-pid: 25088

server 'app-0' : ACTIVE
  password: missing
  watchdog-user: imart
  user: imart
  root: /home/imart/resin-pro-4.0.xx
  conf: /home/imart/resin-pro-4.0.xx/conf/resin.xml
..
```

Windowsの例

```
> resin.exe status
Resin/4.0.xx status for watchdog at 127.0.0.1:6600

watchdog:
  watchdog-pid: 192

server 'app-0' : ACTIVE
  password: missing
  watchdog-user: imart
  user: imart
  root: /C:/resin-pro-4.0.xx
  conf: /C:/resin-pro-4.0.xx/conf/resin.xml
..
```

ログ設定

項目

- [java.util.logging](#)
 - 概要
 - ログ名
 - ログレベル
 - `<log-handler>`
 - ログハンドラ タイムスタンプ
 - ログハンドラ アーカイブ
 - ログハンドラ ELフォーマット
 - `Logger`: アプリケーションのロギング
 - カスタムログハンドラとライブラリログハンドラ
 - カスタムログフォーマット
 - Resinビルトイン ログハンドラ
 - `BamLogHandler`
 - `EventLogHandler`
 - `JmsLogHandler`
 - `MailLogHandler`
 - `SyslogHandler`
 - ログローテーションとアーカイブ
 - サイズによるロールオーバー
 - 時間によるロールオーバー
 - アーカイブファイル
 - ロールオーバーの無効
 - 圧縮
 - 標準出力のリダイレクト
 - `stdout`ログ
 - 属性
 - `stderr`ログ
 - 属性
 - `<access-log>`
 - フォーマットパターン
 - 属性
- ログのパス

Resinは JDK logging インタフェース の出力先を指定し、Webアプリケーション

[java.util.logging](#)

概要

Resinは すべての内部ログにJDK 標準の `java.util.logging` を使用し、ログ
ログ設定は、ログハンドラとログレベルの2つの部分があります。

ログハンドラはResinのログをどこに出力するかを指示します。Resinには、つ
まれています。

ファイルベースログハンドラの例

```
<log-handler name="com.foo" level="all"
  path="${resin.root}/log/foo.log"
  timestamp="[!%y-%m-%d %H:%M:%S.%s] {%{thread}}"/>
```

`<logger>` は名前付きローガーのログレベルを設定します。`<logger>` は一般

ロガーおよびログハンドラ名は階層的であるため、「com.foo」<logger> は「

「fine」レベルのログ出力設定の例

```
<logger name="com.foo" level="fine"/>
<logger name="com.foo.bar" level="finest"/>
```

ログ名

JDK logging API は階層型の命名方式を使用しています。通常、名前はJava

す。

例えば、<logger name="example.hogwarts" ...> は “example.hogwarts.

Resin自身のロギングは、Resinのクラス名に基づいています。以下はResin

名前	意味
all	すべてをデバッグします。
com.caucho.ejb	EJBの処理。
com.caucho.jsp	JSPのデバッグ。
com.caucho.java	Java コンパイル。
com.caucho.server.port	TCPポートのデバッグおよびスレッド。
com.caucho.server.http	HTTP関連のデバッグ。
com.caucho.server.webapp	Webアプリケーション関連のデバッグ。
com.caucho.server.cache	キャッシュ関連のデバッグ。
com.caucho.sql	データベースプール。
com.caucho.transaction	トランザクションの処理。

ログレベル

ロガーレベルは、与えられたデバッグ粒度のログを有効にします。「severe」

報を表示します。

ログレベルは JDK java.util.logging.Level に定義される値と一致していま

名前	API	用途
off		ログをオフにする。
severe	log.severe("...")	Webアプリケーションの重大なエラーや障害の発生を妨げる障害。
warning	log.warning("...")	ブラウザにレスポンスコードを返す際に警告メッセージを送信する。
info	log.info("...")	Webアプリケーションの動作に関する一般的なメッセージ。
config	log.config("...")	設定に関する詳細なログ。
fine	log.fine("...")	ソースコードに精通している開発者が使用する詳細なログ。
finer	log.finer("...")	開発者が使用する詳細なログ。
finest	log.finest("...")	詳細なトレースを行うデバッグログ。
all		全てのメッセージがログに出力される。

<log-handler>

JDK java.util.logging.* API のためのログハンドラを設定します。

java.util.logging には 2 つの手順があります。ログハンドラのセットの設定と

ラをアタッチします。

カスタムハンドラを構成する <log-handler> には最も共通的な設定である「

ログハンドラ タイムスタンプ

logタグのタイムスタンプは、日付と時刻の値に置換される「%」のコードを含

コード	意味
%a	曜日（省略）
%A	曜日（詳細）

コード	意味
%b	月(省略)
%B	月(詳細)
%c	Javaロケールの 日付
%d	月(数字2桁)
%H	24時間の時(数字2桁)
%I	12時間の時(数字2桁)
%j	年の日数
%m	月(数字2桁)
%M	分
%p	午前/午後
%S	秒
%s	ミリ秒
%W	年の週数(数字3桁)
%w	週の日数(数字1桁)
%y	年(数字2桁)
%Y	年(数字4桁)
%Z	タイムゾーン(名称)
%z	タイムゾーン(+/-0800)
%{thread}	現在のスレッドの名前
%{level}	現在のログレベル
%{env}	現在のクラス ローダ環境

典型的なlogタグのタイムスタンプの例

```
<log-handler name=" path='stderr:' timestamp="[%H:%M:%S.%s] {%{thr
```

```
[22:50:11.648] WebApp[/doc] starting
[22:50:11.698] http listening to *:8080
[22:50:11.828] hmx listening to *:6800
```

ログハンドラ アーカイブ

以下の例はロールオーバー ファイルに書き込む標準的なログハンドラです。

```
<log-handler name="" level="all"
  timestamp="[%Y/%m/%d %H:%M:%S.%s] {%{thread}} "/>

<logger name="com.caucho" level="info"/>
```

デフォルトのアーカイブ形式は以下のようになっています。

- rollover-period が 1日(1D)以上の場合。

```
path + ".%Y%m%d"
```

- rollover-period が 1日(1D)より小さい場合。

```
path + ".%Y%m%d.%H"
```

たとえば、標準エラー出力を使用するすべてのログに設定する例は以下のよう

```
<log-handler name=" level='all' path='stderr:' timestamp='[%H:%M:%S.%f
timestamp='[%H:%M:%S.%s]'
rollover-period='1h' rollover-count='1'/>
```

役立つテクニックとして、問題を追跡するために全てのデバッグログの出力が

ログハンドラ ELフォーマット

Resinの <log-handler> のformat属性には、各ログメッセージのフォーマット
フォーマット文字列の例

```
<log-handler name=" level='all' path='stderr:' timestamp='[%H:%M:%S.%f
format=" ${log.level} ${log.name} ${log.message}"/>
```

ログ EL 変数

変数	説明
<code>\${log.level}</code>	ログレベル。
<code>\${log.name}</code>	ロガー名。
<code>\${log.shortName}</code>	ロガー名を短くした はなく "Foo")
<code>\${log.message}</code>	ログメッセージ。
<code>\${log.millis}</code>	イベントが発生した
<code>\${log.sourceClassName}</code>	ロギングを要求しな ない場合があります。
<code>\${log.sourceMethodName}</code>	ロギングを要求しな ない場合があります。
<code>\${log.threadID}</code>	ロギングの要求が 別子を取得。
<code>\${log.thrown}</code>	ロギングの要求に java.lang.Throwable
<code>\${thread}</code>	現在のスレッドの名
<code>\${request}</code>	サーブレットリクエ
<code>\${session}</code>	HTTPセッション。
<code>\${cookie['JSESSIONID']}</code>	リクエスト クッキー

また、フォーマット文字列で環境EL変数を使用することもできます。

```
<web-app>
<log name=" level='all' path='log/debug.log' timestamp='[%H:%M:%S.%f
format=" ${app.contextPath} ${log.message}"/>
...
</web-app>
```

```
[14:55:10.189] [/foo] `null` returning JNDI java:
model for EnvironmentClassLoader[web-app:http://localhost:8080/foo
[14:55:10.189] [/foo] JNDI lookup `java:comp/env/caucho/auth`
exception javax.naming.NameNotFoundException: java:comp/env/ca
[14:55:10.199] [/foo] Application[http://localhost:8080/foo] starting
```

Logger: アプリケーションのロギング

アプリケーションでロギングを行うために、JDKのロギング機能を利用するこ
ギングは少ないロギングと同じくらい開発者の混乱を招きます。

ログ名は実装しているクラスの完全なクラス名にすべきです。別の名前を付
ログレベルは、アプリケーション全体で一貫しているべきです。Resinでは、以
finerのロギングの例

```
import java.util.logging.Logger;
import java.util.logging.Level;

public class Foo {
    private static final Logger log
        = Logger.getLogger(Foo.class.getName());

    ...

    void doFoo(String bar)
    {
        // check for log level if your logging call does anything more
        // than pass parameters
        if (log.isLoggable(Level.FINER))
            log.finer(this + "doFoo(" + bar + ")");

        ...

        log.info(...);

        try {
            ...
        } catch (ExpectedException ex) {
            log.log(Level.FINEST, "expected exception", ex);
        }
        ...
    }
}
```

カスタムログハンドラとライブラリログハンドラ

カスタムハンドラとライブラリのログハンドラは、CanDI XMLの構文を使って

JDK FileHandler の例

```
<web-app xmlns="http://caucho.com/ns/resin"
  xmlns:jdk-logging="urn:java.util.logging">

  <log-handler name="com.foo" level="info">
    <jdk-logging:FileHandler>
      <new>
        <value>/tmp/test.out</value>
      </new>
    </jdk-logging:FileHandler>
  </log-handler>

</web-app>
```

```
package com.foo.demo;

import java.util.logging.*;

public class MyHandler extends Handler
{
    @Override
    public void publish(LogRecord record)
    {
        System.out.println(getFormatter().format(record));
    }

    @Override
    public void flush();
    {
    }

    @Override
    public void close();
    {
    }
}
```

カスタムログフォーマット

ログハンドラと同様に、ログメッセージの書式をカスタマイズできます。フォーサイトの情報をより適切に収集するために、ログメッセージの書式を変更してカスタムフォーマットの設定例

```
<log-handler name="com.foo" level="warning" path="WEB-INF/log.log">
  <formatter><mypkg:MyFormatter/></formatter>
</log-handler>
```

```
package com.mycom.mypkg;

import java.util.logging.*;

public class MyFormatter extends Formatter
{
    @Override
    public String format(LogRecord record)
    {
        return "[" + record.getLevel() + "] " + record.getMessage();
    }
}
```

Resinビルトイン ログハンドラ

ResinはJMS、HMTTPおよび syslog サービスなどに一般的なログパターン また、カスタム ハンドラを作成することも簡単です。

BamLogHandler

BAMハンドラは、BAMエージェントにログメッセージを発行します。エージェント JID (Jabber id) が必要です。

BAM ハンドラ設定の例

```
<logger name="com.foo">
  <resin:BamLogHandler level="warning">
    <to>test@localhost</to>
  </resin:BamLogHandler>
</logger>
```

EventLogHandler

イベントハンドラは、CanDIイベントシステムへの LogEvent を発行します。このクラス名は com.caucho.log.EventLogHandler です。

イベントハンドラ設定の例

```
<logger name="com.foo">
  <resin:EventLogHandler level="warning"/>
</logger>
```

JmsLogHandler

JMS ハンドラは JMS キューにログ メッセージを発行します。

JMS ハンドラ設定の例

```
<web-app xmlns="http://caucho.com/ns/resin"
  xmlns:ee="urn:java:ee"
  xmlns:resin="urn:java:com.caucho.resin">

  <resin:MemoryQueue ee:Named="myQueue"/>

  <logger name="com.foo">
    <resin:JmsLogHandler level="warning">
      <target>${myQueue}</target>
    </resin:JmsLogHandler>
  </logger>

</web-app>
```

MailLogHandler

メールハンドラは email アドレスにログメッセージを送信します。メールの数

MailLogHandler 属性

属性名	説明	デフォルト
to	メールアドレス	なし (必須)
delay-time	最初のメールを送信する前に待機する時間	1分

属性名	説明	デフォルト
mail-interval-min	メールメッセージの最小間隔	1h(間)
properties	JavaMailのプロパティ	なし

メールハンドラ設定の例

```
<logger name="">
  <resin:MailLogHandler level="warning">
    <to>admin@foo.com</to>
    <properties>
      mail.smtp.host=127.0.0.1
      mail.smtp.port=25
    </properties>
  </resin:MailLogHandler>
</logger>
```

SyslogHandler

UNIXシステムでは、SyslogHandlerを使用してsyslogにメッセージを記録す

SyslogHandler 設定の例

```
<logger name="">
  <resin:SyslogLogHandler level="warning">
    <facility>daemon</facility>
    <severity>notice</severity>
  </resin:SyslogLogHandler>
</logger>
```

facility に使用可能な値は、user, mail, daemon, auth, lpr, news, uucp, cr
severityには、emerg, alert, crit, err, warning, notice, info, debugが使用可
「man 3 syslog」と「man syslog.conf」を参照してください。

ログローテーションとアーカイブ

ログローテーションは毎週または毎日ごとにログファイルをアーカイブします
ResinのログローテーションはJDK logging、HTTPアクセスログ、標準入出

サイズによるロールオーバー

ファイルサイズが一定量に達したときにロールオーバーが行われます。Resi
「rollover-size」は最大サイズを指定するために使用し、バイト (50000)、キ

時間によるロールオーバー

最後にロールオーバーされてから一定の期間が経過したときにロールオー
ければ、時間によるロールオーバーを実行しません。サイズによるロールオー
「rollover-period」は期間を指定するために使用し、日 (15D)、週 (2W)、月

アーカイブファイル

ロールオーバーが行われると、ログファイルの名前が変更され (アーカイブ)
「archive-format」は、アーカイブファイルの名前を指定します。これは、通常
同じです。
デフォルトの動作は「rollover-period」の値に依存します。「rollover-period」
パスに「.%Y%m%d」が追加されたものです。
「rollover-period」が一日未満の場合、アーカイブファイル名はオリジナルの

ロール オーバーのを無効

ロールオーバーを無効にするには、「rollover-size」に起こりえないような大

```
<stdout-log path="log/stdout.log" rollover-size="1024mb"/>
```

圧縮

ログのロールオーバー ファイルは gzip や zip で圧縮することができます。ア

```
<log name="" level="warning" path='log/error.log'
  archive-format="%Y-%m-%d.error.log.gz"
  rollover-period="1D"/>

<access-log path="log/access.log"
  archive-format="access-%Y%m%d.log.gz"
  rollover-period="1D"/>
```

標準出力のリダイレクト

stdoutログ

System.out の送信先を設定します。

stdoutログの設定は、親の設定より優先されます。例えば、<web-app>の子設定より優先されます。


注意

「path」はコマンドライン「-stdout」で指定されたパスと同じであっ

属性

属性名	説明
archive-format	ロールオーバーされたアーカイブファイル名のフォーマット。
path	出力先のパス
path-format	パスを決定するためのフォーマットを指定します。構文
rollover-count	ロールオーバーファイルの最大数。
rollover-period	ロールオーバーを行う頻度。日 (15D)、週 (2W)、月 (
rollover-size	ロールオーバーを行うファイルの最大サイズ。バイト (10mb)。
timestamp	行の先頭に使用する タイムスタンプ のフォーマット。

デフォルトのアーカイブフォーマットは以下のようになっています。

- rollover-period が 1日 (1D) 以上の場合。

```
path + ".%Y%m%d"
```

- rollover-period が 1日 (1D) より小さい場合。

```
path + ".%Y%m%d.%H"
```

次の例は <host> に System.out を構成します。<web-app>の出力ログ設

```
<host id='foo.com'>
  <stdout-log path='/var/log/foo/stdout.log'
    rollover-period='1W'/>
  ...
</host>
```

stderrログ

System.err の送信先を設定します。

stderrログの設定は、親の設定より優先されます。例えば、<web-app>の子設定より優先されます。


注意

「path」はコマンドライン「-stderr」で指定されたパスと同じであっ

属性名	説明
archive-format	ロールオーバーされたアーカイブファイル名のフォーマット。
path	出力先のパス
path-format	パスを決定するためのフォーマットを指定します。構文は <code>path</code> のフォーマットと同じです。
rollover-count	ロールオーバーファイルの最大数。
rollover-period	ロールオーバーを行う頻度。日 (15D)、週 (2W)、月 (1M)。
rollover-size	ロールオーバーを行うファイルの最大サイズ。バイト (1024) (10mb)。
timestamp	行の先頭に使用する タイムスタンプ のフォーマット。

デフォルトのアーカイブフォーマットは以下のようになっています。

- rollover-period が 1日 (1D) 以上の場合。

```
path + ".%Y%m%d"
```

- rollover-period が 1日 (1D) より小さい場合。

```
path + ".%Y%m%d.%H"
```

次の例は `<host>` に `System.out` を構成します。`<web-app>`の出力ログ設定

```
<host id='foo.com'>
  <stderr-log path='/var/log/foo/stderr.log'
    rollover-period='1W'/>
  ...
</host>
```

<access-log>

`<access-log>` は、アクセス ログ ファイルを構成します。

`<web-app>` の子に定義されている場合、`<host>`の定義をオーバーライドし

デフォルトのアーカイブフォーマットは以下のようになっています。

- rollover-period が 1日 (1D) 以上の場合。

```
path + ".%Y%m%d"
```

- rollover-period が 1日 (1D) より小さい場合。

```
path + ".%Y%m%d.%H"
```

アクセス ログのフォーマット変数は Apache 変数に従ってください。

フォーマットパターン

パターン	説明
%b	返却されるコンテンツの長さ。
%D	リクエストの処理が完了するまでにかかった時間。(マイクロ秒)
%h	リモートIPアドレス。
%{xxx}i	リクエストヘッダ。
%{xxx}o	レスポンスヘッダ。
%{xxx}c	Cookieの値。
%n	リクエストの属性値。
%r	リクエストURL。
%s	ステータスコード。
%S	セッションID。
%{xxx}t	日時のフォーマット。
%T	リクエストの処理が完了するまでにかかった時間。(秒)

パターン	説明
%u	リモートユーザ。
%U	リクエストURI。
%v	バーチャルホストのサーバ名。

デフォルトのフォーマットは以下のようになっています。

```
"%h %l %u %t \"%r\" %>s %b \"%[Referer]\" \"%[User-Agent]\""
```

属性

属性名	説明
path	出力先のパス
path-format	パスを決定するためのフォーマットを指定します
archive-format	ロールオーバーされたアーカイブファイル名のフ
auto-flush	要求のたびにメモリバッファをフラッシュする場
auto-flush-time	メモリ バッファをフラッシュする時間間隔。
exclude	リクエストのURIが一致する場合、アクセス ログ
format	アクセス ログのフォーマット。
hostname-dns-lookup	IPアドレスの代わりにDNS名を記録。(パフォー
rollover-period	ロールオーバーを行う頻度。日(15D)、週(2W)
rollover-size	ロールオーバーを行うログファイルの最大サイ
rollover-count	ロールオーバーファイルの最大数。

<host> に <access-log> を設定する例。

```
<cluster id="app-tier">

  <host id="">
    <access-log path="log/access.log">
      <rollover-period>2W</rollover-period>
    </access-log>
  </host>
</cluster>
```

ログのパス

「path」はメッセージの送信先を決定するために使用します。一般的には「ア」または「コンソール」に表示するために標準出力、標準エラー出力に移行する。

パス	説明
ファイル	ファイルに出力します。
システム	システムのパス
stdout:	標準出力に出力します。
stderr:	標準エラー出力に出力します。

標準出力にログメッセージを出力する例

```
<log name="" level="all" path="stdout:"/>
```

コアモジュール

ネットワーク設定

項目
概要
リファレンス
クラスタリングID

- ポート番号
- ポートレンジ
- プロトコルバージョン設定
- ユニキャスト設定
- マルチキャスト設定
- 初期起動ホスト設定
- ホストアドレス設定
- マルチキャストアドレス
- マルチキャストポート番号
- 初期起動ホスト数

概要

intra-mart Accel Platform を分散環境で利用する場合のクラスタリングをい

モジュール	コアモジュール
フォーマットファイル(xsd)	WEB-INF/schema/network-agent-config.xsd
設定場所	WEB-INF/conf/network-agent-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<network-agent-config
  xmlns="http://intra-mart.co.jp/system/service/client/config/network-ager
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://intra-mart.co.jp/system/service/client/config
  <id>clusterId</id>
  <bind-port>5200</bind-port>
  <port-range>2</port-range>
  <preferIPv4Stack>true</preferIPv4Stack>

  <TCP>
    <initial-hosts>
      <host address="127.0.0.1"/>
    </initial-hosts>
  </TCP>
</network-agent-config>
```

リファレンス

クラスタリングID

タ	id
グ	
名	

クラスタリングを構成するグループの一意のIDです。
Webアプリケーション毎に一意となるIDを設定してください。

【設定項目】

```
<network-agent-config>
  <id>clusterId</id>
</network-agent-config>
```

必須項目	○
複数設定	×
設定値・設定内容	Webアプリケーション毎に一意のIDを設定します。
単位・型	文字列 (xxxxxxx)
省略時のデフォルト値	なし
親タグ	network-agent-config



注意

Webアプリケーション間で同じIDが設定されている場合、予期しなく、Webアプリケーション毎に一意となるIDを指定してください

ポート番号

タグ bind-port
名

Web Application Server 間で通信を行う際に利用するポート番号を指定し

【設定項目】

```
<network-agent-config>  
<bind-port>5200</bind-port>  
</network-agent-config>
```

必須項目 ☐

複数設定 x

設定値・設定する
内容 利用するポート番号を設定します。

単位・型 数値 (1024 - 65535)

省略時のデフォルト値 なし

親タグ network-agent-config

ポートレンジ

タグ port-range
名

bind-portで指定されたポート番号が既に使用されている場合に代替えとし
例えば、bind-portに5200、port-rangeに2が設定されており、5200番ポート；

【設定項目】

```
<network-agent-config>  
<port-range>2</port-range>  
</network-agent-config>
```

必須項目 ☐

複数設定 x

設定値・設定する
内容 利用するポート番号のレンジを設定します。

単位・型 数値 (0 -)

省略時のデフォルト値 なし

親タグ network-agent-config

プロトコルバージョン設定

タグ preferIPv4Stack
名

通信を行う際のプロトコルバージョンを指定します。

trueを設定するとIPv4 ソケットを使用して通信を行います。

【設定項目】

```
<network-agent-config>
  <preferIPv4Stack>true</preferIPv4Stack>
</network-agent-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	false IPv6 ソケットを使用して通信を行います。 true IPv4 ソケットを使用して通信を行います。
単位・型	真偽値 (true/false)
省略時のデフォルト値	なし
親タグ	network-agent-config

 注意

intra-mart Accel Platform では、IPv6をサポートしていないため、

ユニキャスト設定

タグ名

この設定を行うとユニキャストでの通信を行います。

【設定項目】

```
<network-agent-config>
  <TCP>
  ...
</TCP>
</network-agent-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	ユニキャストでの通信を行う際に必要になる設定をします。
単位・型	なし
省略時のデフォルト値	なし
親タグ	network-agent-config

 注意

この設定を行った場合、マルチキャストの設定(UDP)は行えません。

マルチキャスト設定

タグ名

この設定を行うとマルチキャストでの通信を行います。

【設定項目】

```
<network-agent-config>
  <UDP>
  ...
</UDP>
</network-agent-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	マルチキャストでの通信を行う際に必要になる設定をし
単位・型	なし
省略時のデフォルト値	なし
親タグ	network-agent-config

 **注意**
この設定を行った場合、ユニキャストの設定(TCP)は行えません。

初期起動ホスト設定

タグ名

initial-hosts

intra-mart Accel Platform を初期起動時のサーバのホストアドレスを設定し初期起動ホストはクラスタリングを行う際に通信を行う起点になります。後から起動したサーバはこの初期起動ホストに指定されているサーバと通信

【設定項目】

```
<network-agent-config>
<TCP>
<initial-hosts>
...
</initial-hosts>
</TCP>
</network-agent-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	初期起動時するサーバのホストアドレスを設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	TCP

ホストアドレス設定

タグ名

host

Web Application Server のサーバアドレスを指定します。

【設定項目】

```
<network-agent-config>
<TCP>
<initial-hosts>
<host address="127.0.0.1"/>
</initial-hosts>
</TCP>
</network-agent-config>
```

必須項目

目

複数設定

設定 値・設定する内容
サーバアドレスを設定します。

単位・型 文字列 (xxx.xxx.xxx.xxx)

省略時のデフォルト値 なし

親タグ initial-hosts

【属性】

属性名	説明	必須	デフォルト値
address	サーバアドレスを指定します。	○	なし

マルチキャストアドレス

タグ名 mcast-address

通信時に使用するマルチキャストアドレスを指定します。

【設定項目】

```
<network-agent-config>
<UDP>
  <mcast-address>228.10.10.10</mcast-address>
</UDP>
</network-agent-config>
```

必須項目 ○

複数設定 ×

設定値・設定する内容 マルチキャストアドレスを設定します。

単位・型 文字列 (xxx.xxx.xxx.xxx)

省略時のデフォルト値 なし

親タグ UDP

マルチキャストポート番号

タグ名 mcast-port

通信時に使用するマルチキャストアドレスのポート番号を指定します。

【設定項目】

```
<network-agent-config>
<UDP>
  <mcast-port>45588</mcast-port>
</UDP>
</network-agent-config>
```

必須項目 ○

複数設定 ×

設定値・設定する内容 マルチキャストアドレスのポート番号を設定します。

単位・型	数値 (1024 - 65535)
省略時のデフォルト値	なし
親タグ	UDP

初期起動ホスト数

タグ名	initial-members
-----	-----------------

intra-mart Accel Platform を初期起動時のサーバ台数を指定します。

【設定項目】

```
<network-agent-config>
  <UDP>
    <initial-members>1</initial-members>
  </UDP>
</network-agent-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	初期起動時のサーバ台数を設定します。
単位・型	数値 (0 -)
省略時のデフォルト値	なし
親タグ	UDP

サーバコンテキスト設定

- 項目
- 概要
 - リファレンス
 - ホームディレクトリ
 - 一時ファイルディレクトリ
 - ログファイルディレクトリ
 - サーバ文字コード
 - ベースURL

概要

intra-mart Accel Platform が起動するサーバ情報を設定します。

モジュール	コアモジュール
フォーマットファイル(xsd)	WEB-INF/schema/server-context-config.xsd
設定場所	WEB-INF/conf/server-context-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<server-context-config
  xmlns="http://intra-mart.co.jp/system/platform/config/server-context-cor
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://intra-mart.co.jp/system/platform/config/serv
  <home-directory>WEB-INF</home-directory>
  <work-directory>WEB-INF/work</work-directory>
  <log-directory>WEB-INF/log</log-directory>
  <server-charset>UTF-8</server-charset>
  <!--
  <base-url>http://127.0.0.1:8080/imart</base-url>
  -->
</server-context-config>
```

リファレンス

ホームディレクトリ

タグ名 home-directory

アプリケーションのホームディレクトリを指定します。
ホームディレクトリは設定ファイルやソースのパスを解決する際の基底ディレ

【設定項目】

```
<server-context-config>
  <home-directory>WEB-INF</home-directory>
</server-context-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	アプリケーションの基底ディレクトリを設定します。
単位・型	文字列 (xxx/xxx)
省略時のデフォルト値	なし
親タグ	server-context-config

 注意

この設定を変更した場合、intra-mart Accel Platform が起動し

一時ファイルディレクトリ

タグ名 work-directory

intra-mart Accel Platform で利用する一時ファイルディレクトリを設定します

【設定項目】

```
<server-context-config>
  <work-directory>WEB-INF/work</work-directory>
</server-context-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	一時ファイルディレクトリを設定します。
単位・型	文字列 (xxx/xxx)

省略時のデフォルト値	なし
親タグ	server-context-config

ログファイルディレクトリ

タグ名	log-directory
-----	---------------

intra-mart Accel Platform が出力するログファイルの出力先を指定します。

【設定項目】

```
<server-context-config>
  <log-directory>WEB-INF/log</log-directory>
</server-context-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ログファイルディレクトリを設定します。
単位・型	文字列 (xxx/xxx)
省略時のデフォルト値	なし
親タグ	server-context-config

サーバ文字コード

タグ名	server-charset
-----	----------------

intra-mart Accel Platform の文字コードを設定します。
この設定は、サーバで文字列を扱う時に Unicode との文字コード変換をす

【設定項目】

```
<server-context-config>
  <server-charset>UTF-8</server-charset>
</server-context-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	サーバ文字コードを設定します。
単位・型	文字列 (xxx)
省略時のデフォルト値	なし
親タグ	server-context-config

 注意

この設定は変更しないでください。

ベースURL

タ base-url
グ
名

intra-mart Accel Platform のベースURLを設定します。
この設定は、サーバで特定のURLを生成する際に利用されます。
Web Server と Web Application Server でURLが異なる場合等に、クライ

【設定項目】

```
<server-context-config>  
<base-url>http://127.0.0.1:8080/imart</base-url>  
</server-context-config>
```

必須項目	×
複数設定	×
設定値・設定する内 容	クライアント（ブラウザ等）からアクセスを行うコン 容
単位・型	文字列
省略時のデフォルト 値	リクエストURLから自動的に解決されます。
親タグ	server-context-config

ストレージ設定ファイル

項目

- 概要
- リファレンス
 - 文字コード設定
 - ストレージAPI実装クラス設定
 - 使用禁止文字群
 - 使用禁止文字設定
 - ストレージ設定
 - ストレージルートパス設定
 - システムストレージディレクトリ名設定
 - パブリックストレージディレクトリ名設定
 - グループストレージディレクトリ名設定
 - 設定ファイル配置用ディレクトリ名設定
 - 互換モジュール用ディレクトリ名設定
 - ストレージディレクトリ名設定
 - テンポラリストレージディレクトリ名設定
 - ファイル同期キー設定
 - シンボリックリンク設定

概要

ストレージの利用に関する設定ファイルです。

モジュール	コアモジュール
フォーマット ファイル(xsd)	WEB-INF/schema/storage-config.xsd
設定場所	WEB-INF/conf/storage-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<storage-config xmlns="http://jp.co/intra-mart/foundation/service/client/file"
  <charset>UTF-8</charset>
  <class-name>jp.co.intra_mart.system.service.client.file.StorageLocalFile
  <unable-characters>
    <unable-character>\</unable-character>
    <unable-character>.</unable-character>
    <unable-character>*</unable-character>
    <unable-character>?</unable-character>
    <unable-character>&quot;</unable-character>
    <unable-character>&lt;</unable-character>
    <unable-character>&gt;</unable-character>
    <unable-character>|</unable-character>
  </unable-characters>
  <storage-info>
    <root-path-name>/tmp/storage</root-path-name>
    <system-directory-name>system</system-directory-name>
    <public-directory-name>public</public-directory-name>
    <group-directory-name>groups</group-directory-name>
    <configuration-directory-name>conf</configuration-directory-name>
    <compatible-directory-name>public</compatible-directory-name>
    <storage-directory-name>storage</storage-directory-name>
    <temporary-directory-name>temp</temporary-directory-name>
  </storage-info>
  <synchronize-info>
    <synchronize-key>im_system_storage_synchronize_key</synchroni
  </synchronize-info>
</storage-config>
```

リファレンス

文字コード設定

タ charset
グ
名

テキストファイルの読み込み、書き込み時に利用する文字コードを指定します。

【設定項目】

```
<storage-config>
  <charset>UTF-8</charset>
  ...
</storage-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	テキストファイルの読み込み、書き込み時に利用する文字コード
単位・型	文字列
省略時のデフォルト値	なし
親タグ	storage-config

 注意

intra-mart Accel Platform では、利用可能な文字コードをUTF-8に設定する必要があります。

ストレージAPI実装クラス設定

タ class-name
グ
名

Storage APIにおける実装クラスの指定を行います。

【設定項目】

```
<storage-config>
...
<class-name>jp.co.intra_mart.system.service.client.file.StorageLocalFile
...
</storage-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ストレージに対する読み込み、書き込み等指定されるクラスは、jp.co.intra_mart標準で提供されている実装はローカルです。 代替となる実装は提供されていません
単位・型	文字列
省略時のデフォルト値	なし
親タグ	storage-config

 注意

この設定はストレージに対する操作を行うための実装を定義した

使用禁止文字群

タグ名	unable-charactors
-----	-------------------

ファイル名、ディレクトリ名等における使用禁止文字群です。

【設定項目】

```
<storage-config>
...
<unable-charactors>
<unable-character>\</unable-character>
<unable-character>:</unable-character>
<unable-character>*</unable-character>
<unable-character>?</unable-character>
<unable-character>&quot;</unable-character>
<unable-character>&lt;</unable-character>
<unable-character>&gt;</unable-character>
<unable-character>|</unable-character>
</unable-charactors>
...
</storage-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ファイル名、ディレクトリ名等で使用禁止となる文字群を
単位・型	なし
省略時のデフォルト値	なし
親タグ	storage-config

使用禁止文字設定

タグ名	unable-character
-----	------------------

ファイル名、ディレクトリ名等における使用禁止文字の設定を行います。

```
<storage-config>
...
<unable-charactors>
<unable-character>\</unable-character>
<unable-character>:</unable-character>
<unable-character>*</unable-character>
<unable-character>?</unable-character>
<unable-character>&quot;</unable-character>
<unable-character>&lt;</unable-character>
<unable-character>&gt;</unable-character>
<unable-character>|</unable-character>
</unable-charactors>
...
</storage-config>
```

必須項目	×
複数設定	○
設定値・設定 する内容	ファイル名、ディレクトリ名等で使用禁止となる文字を設定 標準で禁止している文字は：* ? " < > が設定されてい
単位・型	文字列
省略時のデ フォルト値	なし
親タグ	unable-charactors

 注意

この設定はWindows, Linux等で利用されるファイルシステムで利

ストレージ設定

タグ

storage-info

名

ストレージのルートパス等を指定します。

【設定項目】

```
<storage-config>
...
<storage-info>
...
</storage-info>
...
</storage-config>
```

必須項目	○
複数設定	×
設定値・ 設定する 内容	ストレージのパス等の情報を設定します。
単位・型	なし
省略時の デフォルト 値	なし
親タグ	storage-config

ストレージルートパス設定

タグ

root-path-name

名

ストレージルートとして利用するファイルシステム上のパスを指定します。
この設定では、システムプロパティの利用が可能です。
\${システムプロパティのキー} 形式で記述を行うことによりシステムプロパティ
例えば、Resinを利用した場合は、Resinがインストール(展開)された位置を
その為 \${resin.home}/storage と記述した場合はResinがインストール(展
この値は通常、ストレージとして利用するNFS等の共有ディスクのパスを指し
/tmp等、OS起動時に削除される可能性のあるパス等の指定は行わないで

【設定項目】

```
<storage-config>
...
<storage-info>
  <root-path-name>/tmp/storage</root-path-name>
...
</storage-info>
...
</storage-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ストレージルートとして利用するファイルシステム上のパス
単位・型	文字列
省略時のデフォルト値	なし
親タグ	storage-info

システムストレージディレクトリ名設定

タグ名	system-directory-name
-----	-----------------------

システムストレージとして利用されるディレクトリ名を設定します。
指定されたディレクトリ名は、ストレージルートパス設定が行われたパス直下

【設定項目】

```
<storage-config>
...
<storage-info>
  <system-directory-name>system</system-directory-name>
...
</storage-info>
</storage-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	システムストレージとして利用されるディレクトリ名を設定
単位・型	文字列
省略時のデフォルト値	なし
親タグ	storage-info

 注意

この設定は、他のストレージディレクトリ名設定 (public-directory-name) と
この設定は、intra-mart Accel Platform において利用される内容

パブリックストレージディレクトリ名設定

タグ名	public-directory-name
-----	-----------------------

パブリックストレージとして利用されるディレクトリ名を設定します。
指定されたディレクトリ名は、ストレージルートパス設定が行われたパス直下

intra-mart Accel Platform 2013 Winter以前
指定したディレクトリがパブリックストレージのルートディレクトリになります。

intra-mart Accel Platform 2014 Spring以降
指定したディレクトリの直下に作成されるテナントIDをディレクトリ名とした
この設定はテナント作成時にストレージルートパスが指定されていない場合
テナント作成時にストレージルートパスが指定されている場合は、そちらが

【設定項目】

```
<storage-config>
...
<storage-info>
...
<public-directory-name>public</public-directory-name>
...
</storage-info>
</storage-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	パブリックストレージとして利用されるディレクトリ名を設
単位・型	文字列
省略時のデフォルト値	なし
親タグ	storage-info

 注意

この設定は、他のストレージディレクトリ名設定 (system-director
この設定は、intra-mart Accel Platform において利用される内容

グループストレージディレクトリ名設定

タグ group-directory-name
名

グループストレージとして利用されるディレクトリ名を設定します。
指定されたディレクトリ名は、ストレージルートパス設定が行われたパス直下

【設定項目】

```
<storage-config>
...
<storage-info>
...
<group-directory-name>groups</group-directory-name>
...
</storage-info>
</storage-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	グループストレージとして利用されるディレクトリ名を設
単位・型	文字列
省略時のデフォルト値	なし

親タグ storage-info

注意

この設定は、他のストレージディレクトリ名設定（system-directorグループストレージは過去の互換性および、将来的な利用の為にこの設定は、intra-mart Accel Platform において利用される内容

設定ファイル配置用ディレクトリ名設定

タグ configuration-directory-name
名

設定ファイル配置用として利用されるディレクトリ名を設定します。
指定されたディレクトリ名は、ストレージルートパス設定が行われたパス直下

【設定項目】

```
<storage-config>
...
<storage-info>
...
  <configuration-directory-name>conf</configuration-directory-name>
...
</storage-info>
</storage-config>
```

必須項目	☐
複数設定	x
設定値・設定内容	設定ファイル配置用として利用されるディレクトリ名を設定する内容
単位・型	文字列
省略時のデフォルト値	なし
親タグ	storage-info

注意

この設定は、他のストレージディレクトリ名設定（system-director設定ファイル配置用ディレクトリ名設定は、将来的な利用の為にこの設定は、intra-mart Accel Platform において利用される内容

互換モジュール用ディレクトリ名設定

タグ compatible-directory-name
名

互換モジュール用として利用されるディレクトリ名を設定します。
指定されたディレクトリ名は、ストレージルートパス設定が行われたパス直下

【設定項目】

```
<storage-config>
...
<storage-info>
...
  <compatible-directory-name>public</compatible-directory-name>
...
</storage-info>
</storage-config>
```

必須項目	☐
複数設定	x

設定値・設定する内容	互換モジュール用として利用されるディレクトリ名を設定
単位・型	文字列
省略時のデフォルト値	なし
親タグ	storage-info

注意

互換モジュールを利用した場合のストレージディレクトリはパブリック互換モジュール用ディレクトリ名設定は、将来的な利用の為此の設定は、intra-mart Accel Platform において利用される内容

ストレージディレクトリ名設定

タグ storage-directory-name
名

パブリックストレージ、システムストレージ、テンポラリストレージそれぞれの指定されたディレクトリ名は、パブリックストレージ、システムストレージ、テン

【設定項目】

```
<storage-config>
...
<storage-info>
...
<storage-directory-name>storage</storage-directory-name>
...
</storage-info>
</storage-config>
```

必須項目 ○

複数設定 ×

設定値・設定する内容 ストレージディレクトリとして利用されるディレクトリ名を

単位・型 文字列

省略時のデフォルト値 なし

親タグ storage-info

注意

この設定は、intra-mart Accel Platform において利用される内容

テンポラリストレージディレクトリ名設定

タグ temporary-directory-name
名

テンポラリストレージとして利用されるディレクトリ名を設定します。指定されたディレクトリ名は、ストレージルートパス設定が行われたパス直下このディレクトリは、セッションストレージ等で利用される一時領域として使用

【設定項目】

```
<storage-config>
...
<storage-info>
...
<temporary-directory-name>temp</temporary-directory-name>
...
</storage-info>
</storage-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	テンポラリストレージとして利用されるディレクトリ名を記述する内容
単位・型	文字列
省略時のデフォルト値	なし
親タグ	storage-info

 **注意**

この設定は、他のストレージディレクトリ名設定 (system-director) と併用する必要があります。
この設定は、intra-mart Accel Platform において利用される内容です。

ファイル同期キー設定

タグ名

synchronize-key

分散環境において、ストレージを操作する際の同期化用ロックキーの設定でこの項目は将来的な利用の為に予約項目です。

【設定項目】

```
<storage-config>
<synchronize-info>
  <synchronize-key>im_system_storage_synchronize_key</synchronize-key>
</synchronize-info>
</storage-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	同期化用のロックに利用するキー値を設定します。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	synchronize-info

シンボリックリンク設定

タグ名

permit-symlink

ストレージのルートパスの配下外に実体をもつシンボリックリンクやジャンクストレージのルートパスとは、ストレージ設定で設定される各ストレージのルートパス設定は、intra-mart Accel Platform 2015 Winter(Lydia) 以降で利用できます。

【設定項目】

```
<storage-config>
...
<permit-symlink>false</permit-symlink>
...
</storage-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	true ストレージのルートパス配下外に実体がある場合 false ストレージのルートパス配下外に実体がない場合

単位・型	真偽値 (true/false)
省略時のデフォルト値	false
親タグ	storage-config

**注意**

シンボリックリンクやジャンクションを利用した場合、ストレージの

サービスセクタ設定

項目

- [概要](#)
- [リファレンス](#)
 - [サービス設定](#)
 - [リスナの設定](#)
 - [サービスの起動するサーバの指定](#)
 - [サービスの起動させないサーバの指定](#)
 - [サーバアドレス](#)

概要

intra-mart Accel Platform の各サービスが起動するサーバの選出方法を記

モジュール	コアモジュール
フォーマットファイル (xsd)	WEB-INF/schema/service-selector-config.xsd
設定場所	WEB-INF/conf/service-selector-config/service

```
<?xml version="1.0" encoding="UTF-8"?>
<service-selector-config
  xmlns="http://intra-mart.co.jp/system/service/selector/config/service-selector"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://intra-mart.co.jp/system/service/selector/config/service-selector http://intra-mart.co.jp/system/service/selector/config/service-selector.xsd">
  <services>
    <service service-id="server.service.controller"/>
  </services>
</service-selector-config>
```

リファレンス

サービス設定

タグ services
属性
名

各サービスの選出方法を設定します。

【設定項目】

```
<server-selector-config>
  <services>
    <service service-id="XXXXXX" />
  </services>
</server-selector-config>
```

必須項目 ○

複数設定 x

設定値・設定する内容	service 各サービスの選出方法の設定
単位・型	なし
省略時のデフォルト値	なし
親タグ	service-selector-config

リスナの設定

タグ名	service
-----	---------

サービスが起動するサーバの選出方法を設定します。

【設定項目】

```
<services>
<service service-id="XXXXX" />
</services>
```

必須項目	☐
複数設定	☐
設定値・設定する内容	サービスが起動するサーバの選出方法を設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	services

【属性】

属性名	説明
service-id	選出方法を設定するサービスのID。
select-num	サービスが起動するサーバの台数を指定します。(繰り返します。)

 コラム

intra-mart Accel Platform で提供されている各サービスのID

- ServerManager : 「server.service.controller」
- TaskService : 「server.service.task.management」
- JobSchedulerService : 「server.service.job_schedule

サービスの起動するサーバの指定

タグ名	assign-hosts
-----	--------------

サービスが起動するサーバを指定する場合に設定します。
単一のサーバで起動するサービスの場合、指定されているサーバを優先的に指定されているサーバが起動していない場合は、動的にサービスが起動する。

複数のサーバで起動するサービスの場合、指定されているサーバでサービスが起動していない場合は、動的にサービスが起動する。

【設定項目】

```
<service service-id="XXXXX">
<assign-hosts>
<host>10.0.0.1</host>
</assign-hosts>
</service>
```

必須項目	×
複数設定	×
設定値・設定する内容	host サービスが起動するサーバを指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	service

サービスの起動させないサーバの指定

タグ名

サービスを起動させないサーバを指定する場合に設定します。
この設定に指定されているサーバのみ起動している場合はサービスは起動

【設定項目】

```
<service service-id="XXXXX">
  <invalid-hosts>
    <host>10.0.0.1</host>
  </invalid-hosts>
</service>
```

必須項目	×
複数設定	×
設定値・設定する内容	host サービスを起動させないサーバのアドレスを指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	service

サーバアドレス

タグ名

サーバアドレスを設定します。
サービスを起動するサーバ、または、サービスを起動しないサーバのアドレス

【設定項目】

```
<assign-hosts>
  <host>10.0.0.1</host>
</assign-hosts>
```

必須項目

複数設定

設定値・設定する内容

単位・型

文字列 (xxx.xxx.xxx.xxx)

省略時のデフォルト値	なし
親タグ	assign-hosts, invalid-hosts

サーバアドレスには以下のワイルドカードを指定することができます。

- アスタリスク（*）

同じネットワークセグメントの全てのサーバを指定する場合に使用します。例えば以下のような指定が可能です。

* : 全てのサーバ
10.* : 「10」のセグメント内のサーバ
10.0.* : 「10.0」のセグメント内のサーバ
10.0.0.* : 「10.0.0」のセグメント内のサーバ



- 注意**
- アスタリスクは「10.0.0.1*」のようにドット(.)単位で区切る必要があります。
 - アスタリスクは末尾のみ指定可能です。「10.0.*.0」のような指定はできません。

- レンジ（[XX-XX]）

サーバアドレスの範囲指定を行う場合に使用します。例えば以下のような指定が可能です。

10.0.[1-100].1 : 「10.0.1」から「10.0.100」のセグメントのアドレスが「10.0.0.1」のサーバ
10.0.0.[1-100] : 「10.0.0.1」から「10.0.0.100」までのアドレスのサーバ



- コラム**
- レンジはセグメント毎に指定してください。
 - レンジは「10.0.[1-10].[1.-100]」のように複数指定することができます。

initializer 設定

項目

- 概要
- リファレンス
 - サーバサイドJavaScript グローバル関数、または拡張APIの設定
 - サーバサイドJavaScript グローバル関数:Javaクラス
 - サーバサイドJavaScript グローバル関数:JavaScript関数
 - 拡張API:Javaクラス
 - 拡張API:JavaScript関数
 - IMARTタグの設定
 - 拡張IMARTタグ:Javaクラス
 - 拡張IMARTタグ:JavaScript関数
 - ApplicationInitializer設定
 - 起動時初期化:Javaクラス
 - 起動時初期化js ファイルのパス

概要

intra-mart Accel Platform 起動時に、実行する初期化処理を設定します。拡張API、グローバル関数、拡張imartタグの設定を含みます。また、初期化クラス、および、初期化スクリプトの設定を含みます。既存のファイルは編集しないでください。

モジュール	コアモジュール
フォーマットファイル名(xsd)	WEB-INF/schema/initializer-config.xsd
設定場所	WEB-INF/conf/products/initializer/initializer-XXX.xml



- 注意**
- ファイル名の initializer-XXX.xml のXXXはショートモジュールID。ショートモジュールIDとはモジュールIDを"."で分割した末尾になります。例として、モジュールIDが「org.example.foo」場合は「initializer-foo.xml」になります。

```
<?xml version="1.0" encoding="UTF-8"?>
<initializer-config
  xmlns="http://intra-mart.co.jp/system/secure/product/initializer/config/initi
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://intra-mart.co.jp/system/secure/product/initi

<java-script-api>
  <global-function-class>sample.common_libs.global_fnction.GlobalFn
  <global-function-script>sample/common_libs/global_fnction#global_f
  <api-class>sample.common_libs.api.InstanceApi</api-class>
  <api-script>sample/common_libs/api#staticApi</api-script>
</java-script-api>

<jssp-tag>
  <tag-class>sample.common_libs.imart_tag.ImSampleTag#sample3</t
  <tag-script>sample/common_libs/imart_tag#imSampleTag</tag-script>
</jssp-tag>

<initializer>
  <class-name>sample.common_libs.Init</class-name>
  <script-name>sample/common_libs/init</script-name>
</initializer>
</initializer-config>
```

リファレンス

サーバサイドJavaScript グローバル関数、または拡張APIの設定

タグ名

サーバサイドJavaScript グローバル関数、拡張APIの設定を行います。

【設定項目】

```
<initializer-config
  xmlns="http://intra-mart.co.jp/system/secure/product/initializer/config/initi
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://intra-mart.co.jp/system/secure/product/initi

<java-script-api>
  <global-function-class>sample.common_libs.global_fnction.GlobalFn
  <global-function-script>sample/common_libs/global_fnction#global_f
  <api-class>sample.common_libs.api.InstanceApi</api-class>
  <api-script>sample/common_libs/api#staticApi</api-script>
</java-script-api>

</initializer-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	global-function-class グローバル関数を定義関数
	global-function-script グローバル関数を定義関数
	api-class オブジェクトを定義
	api-script オブジェクトを定義ト
単位・型	なし
省略時のデフォルト値	なし
親タグ	initializer-config

サーバサイドJavaScript グローバル関数: Javaクラス

タグ global-function-class
名

指定したJavaクラスで サーバサイドJavaScript グローバル関数の登録を行います。

【設定項目】

```
<java-script-api>  
<global-function-class>sample.common_libs.global_fnction.GlobalFnc  
</java-script-api>
```

必須項目	×
複数設定	○
設定値・設定内容	グローバル関数を実装したJavaクラスのパス#実行関数
単位・型	文字列 (xx.xxx.xx#xxxx)
省略時のデフォルト値	なし
親タグ	java-script-api

サーバサイドJavaScript グローバル関数:JavaScript関数

タグ global-function-script
名

指定したJavaScript関数で サーバサイドJavaScript グローバル関数の登録を行います。

【設定項目】

```
<java-script-api>  
<global-function-script>sample/common_libs/global_fnction#global_fn  
</java-script-api>
```

必須項目	×
複数設定	○
設定値・設定内容	グローバル関数を定義する js ファイルのパス#実行関数
単位・型	文字列 (xx/xxx/xx#xxxx)
省略時のデフォルト値	なし
親タグ	java-script-api

拡張API:Javaクラス

タグ api-class
名

指定したJavaクラスで拡張APIの登録を行います。

【設定項目】

```
<java-script-api>  
<api-class>sample.common_libs.api.InstanceApi</api-class>  
</java-script-api>
```

必須項目	×
複数設定	○
設定値・設定内容	オブジェクトを定義した Javaクラスパス
単位・型	文字列 (xx.xxx.xx)

省略時のデフォルト値	なし
親タグ	java-script-api

拡張API:JavaScript関数

タグ名	api-script
-----	------------

指定したJavaScript関数で拡張APIの登録を行います。

【設定項目】

```
<java-script-api>
  <api-script>sample/common_libs/api#staticApi</api-script>
</java-script-api>
```

必須項目	×
複数設定	○
設定値・設定する内容	オブジェクトを定義した js ファイルパス#オブジェクト
単位・型	文字列 (xx/xxx/xx#xxxx)
省略時のデフォルト値	なし
親タグ	java-script-api

IMARTタグの設定

タグ名	jssp-tag
-----	----------

IMARTタグの設定を行います。

【設定項目】

```
<initializer-config
  xmlns="http://intra-mart.co.jp/system/secure/product/initializer/config/initi
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://intra-mart.co.jp/system/secure/product/initi

  <jssp-tag>
    <tag-class>sample.common_libs.imart_tag.ImSampleTag</tag-class>
    <tag-script>sample/common_libs/imart_tag#imSampleTag</tag-script>
  </jssp-tag>

</initializer-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	tag-classIMARTタグを定義したJavaクラス
	tag-scriptIMARTタグを定義js ファイルのバ行関数
単位・型	なし
省略時のデフォルト値	なし
親タグ	initializer-config

拡張IMARTタグ:Javaクラス

タグ名

指定したJavaクラスでIMARTタグの登録を行います。

【設定項目】

```
<jssp-tag>
  <tag-class>sample.common_libs.imart_tag.ImSampleTag</tag-class>
</jssp-tag>
```

必須項目	x
複数設定	☐
設定値・設定する内容	IMARTタグを定義したJavaクラスのパス
単位・型	文字列 (xx.xxx.xx)
省略時のデフォルト値	なし
親タグ	jssp-tag

拡張IMARTタグ:JavaScript関数

タグ名

指定したJavaScript関数でIMARTタグの登録を行います。

【設定項目】

```
<jssp-tag>
  <tag-script>sample/common_libs/imart_tag#imSampleTag</tag-script>
</jssp-tag>
```

必須項目	x
複数設定	☐
設定値・設定する内容	IMARTタグを定義js ファイルのパス#実行関数
単位・型	文字列 (xx.xxx.xx#xxxx)
省略時のデフォルト値	なし
親タグ	jssp-tag

ApplicationInitializer設定

タグ名

初期化クラス、および、初期化スクリプトの設定を行います。

【設定項目】

```
<initializer-config
  xmlns="http://intra-mart.co.jp/system/secure/product/initializer/config/ini
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://intra-mart.co.jp/system/secure/product/ini

<initializer>
  <class-name>sample.common_libs.Init</class-name>
  <script-name>sample/common_libs/init</script-name>
</initializer>

</initializer-config>
```

必須項目	×	
複数設定	×	
設定値・設定する内容	class-name	初期化Javaクラスのパス
	script-name	初期化js ファイルのパス
単位・型	なし	
省略時のデフォルト値	なし	
親タグ	initializer-config	

起動時初期化Javaクラス

タグ
class-name

指定したJavaクラスが起動時に初期化対象となる設定を行います。

【設定項目】

```
<initializer>
  <class-name MBean="false">sample.common_libs.imart_tag.ImSamp
</initializer>
```

必須項目	×	
複数設定	○	
設定値・設定する内容	初期化Javaクラスのパス	
単位・型	文字列 (xx.xxx.xx)	
省略時のデフォルト値	なし	
親タグ	initializer	

【属性】

属性名	説明	必須
MBean	MBeanServerに登録するかどうかを指定します。	×

起動時初期化js ファイルのパス

タ script-name
グ
名

指定したJavaクラスが起動時に初期化対象となる設定を行います。

【設定項目】

```
<initializer>
<script-name>sample/common_libs/init</script-name>
</initializer>
```

必須項目	x
複数設定	○
設定値・設定する内容	初期化js ファイルのパス
単位・型	文字列 (xx/xxx/xx)
省略時のデフォルト値	なし
親タグ	initializer

IPアドレス取得元設定

項目
● 概要
● リファレンス
○ リモートIPアドレス設定
○ リモートIPアドレス取得元HTTPヘッダ設定

概要

リモートホストのIPアドレスを取得する元となるHTTPヘッダを設定します。

リモートホストのIPアドレスを使用して認可でアクセス制御を行ったり、シス：

デフォルト(未設定)の状態では javax.servlet.ServletRequest#getRemot
しかし、ロードバランサやプロキシサーバを使用している環境では、 javax.s
この場合、代替となるHTTPヘッダ(X-Forwarded-For など)を設定すること

ここで設定された順にヘッダ値を取得していき、IPアドレスに変換できたもの
最後まで取得できなかった場合は、 javax.servlet.ServletRequest#getRei
javax.servlet.ServletRequest#getRemoteAddr() の動作について詳しくは

モジュール	テナント管理機能
フォーマットファイル	WEB-INF/schema/ip-address-detector-config.xsd ル(xsd)
設定場所	WEB-INF/conf/ip-address-detector-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<ip-address-detector-config xmlns="http://www.intra-mart.jp/http/utility/i
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/http/utility/ip-address-det
<remote-address>
  <http-header delimiter="," position="first">X-Forwarded-For</http-hea
  <http-header>X-LB-Remote-Addr</http-header>
</remote-address>
</ip-address-detector-config>
```

リファレンス

リモートIPアドレス設定

タグ remote-address
グ
名

アクセス元のリモートホストのIPアドレスを取得する元の場所を定義します。

【設定項目】

```
<ip-address-detector-config>
<remote-address>
  .....
</remote-address>
</ip-address-detector-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	ip-address-detector-config

リモートIPアドレス取得元HTTPヘッダ設定

タグ http-header
グ
名

アクセス元のリモートホストのIPアドレスを取得するHTTPヘッダ名を定義し

ここで設定された順にヘッダ値を取得していき、IPアドレスに変換できたもの
最後まで取得できなかった場合は、javax.servlet.ServletRequest#getRei

【設定項目】

```
<ip-address-detector-config>
<remote-address>
  <http-header delimiter="," position="first">X-Forwarded-For</http-hea
  <http-header>X-LB-Remote-Addr</http-header>
</remote-address>
</ip-address-detector-config>
```

必須項目	×
複数設定	○

設定 値・設 定す る内 容	HTTPヘッダ名
単位・ 型	文字列
省略 時の デフォ ルト 値	なし
親タ グ	remote-address

【属性】

属性名	説明
delimiter	ヘッダに設定されている値を区切る文字列です。 省略した場合は区切らず、取得できたそのままの値を返す。
position	delimiter で区切られた複数の値から評価に使用する delimiter を省略した場合は使用されません。 first 最初の値を評価します。 last 最後の値を評価します。 数値 指定された位置の値を評価します。範囲外の 値はエラーを返す。 正の数を指定した場合は、最初から数えて何 番目の値を評価します。 負の数を指定した場合は、最後から数えて何 番目の値を評価します。

！ 注意

X-Forwarded-For ヘッダは非標準のため、ロードバランサやプロキシサーバーの実装に応じて、delimiter の値が異なる場合があります。

マルチデバイス

クライアントタイプマスタ

- 項目
- 概要
 - リファレンス
 - クライアントタイプの設定

概要

クライアントタイプを使用するための設定を行います。

モジュール	マルチデバイス
フォーマットファイル(xsd)	WEB-INF/schema/client-type-config.xsd
設定場所	WEB-INF/conf/client-type-config/{任意のファイル名}.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<client-type-config
  xmlns="http://intra-mart.co.jp/system/multi_device/client_type/config/client_type"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://intra-mart.co.jp/system/multi_device/client_type/config/client_type http://intra-mart.co.jp/system/multi_device/client_type/config/client_type.xsd">
  <client-type-info class="jp.co.intra_mart.system.multi_device.client_type">
  </client-type-info>
</client-type-config>
```

リファレンス

クライアントタイプの設定

タ client-type-info
グ
名

intra-mart Accel Platform で使用できるクライアントタイプを定義します。

【設定項目】

```
<client-type-config>  
<client-type-info class="jp.co.intra_mart.system.multi_device.client_typ  
</client-type-config>
```

必須項目	☐
複数設定	☐
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	client-type-config

【属性】

属性名	説明
class	Javaインタフェース jp.co.intra_mart.foundation. ラスのFQDNを指定します。
default	このクライアントタイプをシステム・デフォルトのグ

!

注意

default="true" のクライアントタイプ設定が複数ある場合、設定フ
default="true" のクライアントタイプが見つからない場合、設定フ

国際化機能

日付と時刻の形式マスタ

- 項目
- 概要
 - リファレンス
 - 「日付と時刻の形式」の設定
 - 「日付と時刻の形式」識別子の設定
 - 「日付と時刻の形式」ロケールの設定
 - 「日付と時刻の形式」フォーマットの設定
 - 「日付と時刻の形式」フォーマット識別子の設定
 - 「日付と時刻の形式」フォーマットパターンの設定

intra-mart Accel Platform で利用する日付と時刻の形式のマスタ情報を設
ここで設定した内容は、以下の画面から選択して利用することができます。

- テナント管理 > テナント情報 > 日付と時刻の形式
- 個人設定 > 日付と時刻の形式

モジュール	国際化機能
フォーマットファイル	WEB-INF/schema/date-time-format-config.xsd (xsd)
設定場所	WEB-INF/conf/date-time-format-config/{任意の

```
<?xml version="1.0" encoding="UTF-8"?>
<date-time-format-config
  xmlns="http://www.intra-mart.jp/system/i18n/datetime/date-time-format-c
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/system/i18n/datetime/date
  <format-set default="true">
    <format-set-id>IM_DATETIME_FORMAT_SET_EN_BASE</format-s
    <locale>en</locale>
    <format type="date">
      <format-id>IM_DATETIME_FORMAT_DATE_STANDARD</format-i
      <pattern default="true">MMM d, yyyy</pattern>
      <pattern>MMM dd, yyyy</pattern>
      <pattern>d/M/yyyy</pattern>
      <pattern>d/MM/yyyy</pattern>
      <pattern>dd/MM/yyyy</pattern>
      <pattern>d-MMM-yyyy</pattern>
      <pattern>dd-MMM-yyyy</pattern>
      <pattern>d MMM, yyyy</pattern>
      <pattern>dd MMM, yyyy</pattern>
      <pattern>d MMM yyyy</pattern>
      <pattern>dd MMM yyyy</pattern>
      <pattern>yyyy-MM-dd</pattern>
    </format>
    <format type="date">
      <format-id>IM_DATETIME_FORMAT_DATE_SIMPLE</format-id>
      <pattern default="true">MMM d</pattern>
      <pattern>MMM dd</pattern>
      <pattern>d/M</pattern>
      <pattern>d/MM</pattern>
      <pattern>dd/MM</pattern>
      <pattern>d-MMM</pattern>
      <pattern>dd-MMM</pattern>
      <pattern>d MMM</pattern>
      <pattern>dd MMM</pattern>
      <pattern>MM-dd</pattern>
    </format>
    <format type="date">
      <format-id>IM_DATETIME_FORMAT_DATE_INPUT</format-id>
      <pattern default="true">yyyy/MM/dd</pattern>
    </format>
    <format type="time">
      <format-id>IM_DATETIME_FORMAT_TIME_STANDARD</format-i
      <pattern default="true">h:mm a</pattern>
      <pattern>hh:mm a</pattern>
      <pattern>H:mm</pattern>
      <pattern>HH:mm</pattern>
    </format>
    <format type="time">
      <format-id>IM_DATETIME_FORMAT_TIME_TIMESTAMP</format-i
      <pattern default="true">h:mm:ss a</pattern>
      <pattern>hh:mm:ss a</pattern>
      <pattern>H:mm:ss</pattern>
      <pattern>HH:mm:ss</pattern>
    </format>
    <format type="time">
      <format-id>IM_DATETIME_FORMAT_TIME_INPUT</format-id>
      <pattern default="true">HH:mm</pattern>
    </format>
  </format-set>
  <format-set>
    <format-set-id>IM_DATETIME_FORMAT_SET_JA_BASE</format-s
    <locale>ja</locale>
    <format type="date">
      <format-id>IM_DATETIME_FORMAT_DATE_STANDARD</format-i
      <pattern>yyyy'年'M'月'd'日'</pattern>
      <pattern>yyyy'年'MM'月'dd'日'</pattern>
```

```

<pattern>yyyy/M/d</pattern>
<pattern default="true">yyyy/MM/dd</pattern>
<pattern>yyyy-MM-dd</pattern>
</format>
<format type="date">
  <format-id>IM_DATETIME_FORMAT_DATE_SIMPLE</format-id>
  <pattern>M'月'd'日'</pattern>
  <pattern>MM'月'dd'日'</pattern>
  <pattern>M/d</pattern>
  <pattern default="true">MM/dd</pattern>
  <pattern>MM-dd</pattern>
</format>
<format type="date">
  <format-id>IM_DATETIME_FORMAT_DATE_INPUT</format-id>
  <pattern default="true">yyyy/MM/dd</pattern>
</format>
<format type="time">
  <format-id>IM_DATETIME_FORMAT_TIME_STANDARD</format-id>
  <pattern>ah:mm</pattern>
  <pattern>ahh:mm</pattern>
  <pattern default="true">H:mm</pattern>
  <pattern>HH:mm</pattern>
</format>
<format type="time">
  <format-id>IM_DATETIME_FORMAT_TIME_TIMESTAMP</format-id>
  <pattern>ah:mm:ss</pattern>
  <pattern>ahh:mm:ss</pattern>
  <pattern default="true">H:mm:ss</pattern>
  <pattern>HH:mm:ss</pattern>
</format>
<format type="time">
  <format-id>IM_DATETIME_FORMAT_TIME_INPUT</format-id>
  <pattern default="true">HH:mm</pattern>
</format>
</format-set>
<format-set>
  <format-set-id>IM_DATETIME_FORMAT_SET_ZHCN_BASE</format-set-id>
  <locale>zh_CN</locale>
  <format type="date">
    <format-id>IM_DATETIME_FORMAT_DATE_STANDARD</format-id>
    <pattern>yyyy'年'M'月'd'日'</pattern>
    <pattern>yyyy'年'MM'月'dd'日'</pattern>
    <pattern>yyyy/M/d</pattern>
    <pattern>yyyy/MM/dd</pattern>
    <pattern default="true">yyyy-M-d</pattern>
    <pattern>yyyy-MM-dd</pattern>
    <pattern>d MMM yyyy</pattern>
    <pattern>dd MMM yyyy</pattern>
  </format>
  <format type="date">
    <format-id>IM_DATETIME_FORMAT_DATE_SIMPLE</format-id>
    <pattern>M'月'd'日'</pattern>
    <pattern>MM'月'dd'日'</pattern>
    <pattern>M/d</pattern>
    <pattern>MM/dd</pattern>
    <pattern default="true">M-d</pattern>
    <pattern>MM-dd</pattern>
    <pattern>d MMM</pattern>
    <pattern>dd MMM</pattern>
  </format>
  <format type="date">
    <format-id>IM_DATETIME_FORMAT_DATE_INPUT</format-id>
    <pattern default="true">yyyy/MM/dd</pattern>
  </format>
  <format type="time">
    <format-id>IM_DATETIME_FORMAT_TIME_STANDARD</format-id>
    <pattern>ah:mm</pattern>
    <pattern>ahh:mm</pattern>
    <pattern default="true">H:mm</pattern>
    <pattern>HH:mm</pattern>
  </format>
  <format type="time">
    <format-id>IM_DATETIME_FORMAT_TIME_TIMESTAMP</format-id>
    <pattern>ah:mm:ss</pattern>
    <pattern>ahh:mm:ss</pattern>
    <pattern default="true">H:mm:ss</pattern>
    <pattern>HH:mm:ss</pattern>
  </format>
  <format type="time">
    <format-id>IM_DATETIME_FORMAT_TIME_INPUT</format-id>
    <pattern default="true">HH:mm</pattern>
  </format>
</format-set>
</date-time-format-config>

```

リファレンス

「日付と時刻の形式」の設定

タグ
format-set
名

intra-mart Accel Platform で利用する「日付と時刻の形式」を定義します。
詳細な設定は配下のタグで行います。
このタグでは、システム・デフォルトの設定を行います。

【設定項目】

```
<date-time-format-config>  
  <format-set default="true">  
    .....  
  </format-set>  
</date-time-format-config>
```

必須項目	☐
複数設定	☐
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	date-time-format-config

【属性】

属性名	説明
default	true の場合、システム・デフォルトの日付と時刻の形式になります。

「日付と時刻の形式」識別子の設定

タグ
format-set-id
名

「日付と時刻の形式」を識別するための識別子を設定します。

【設定項目】

```
<date-time-format-config>  
  <format-set>  
    <format-set-id>IM_DATETIME_FORMAT_SET_EN_BASE</format-set-id>  
    .....  
  </format-set>  
</date-time-format-config>
```

必須項目	☐
------	-----------------------

複数設定	×
設定値・「日付と時刻の形式」の識別子 設定する内容	
単位・型	文字列
省略時のデフォルト値	なし
親タグ	format-set

「日付と時刻の形式」ロケールの設定

タグ名	locale
-----	--------

「日付と時刻の形式」のロケールを設定します。
設定するロケールは、ロケールマスタファイルに定義されている必要があります。

【設定項目】

```
<date-time-format-config>
  <format-set>
    .....
    <locale>en</locale>
    .....
  </format-set>
</date-time-format-config>
```

必須項目	○
複数設定	×
設定値・ロケールID 設定する内容	
単位・型	文字列
省略時のデフォルト値	なし
親タグ	format-set

「日付と時刻の形式」フォーマットの設定

タグ名	format
-----	--------

フォーマットの種類(「日付」「時刻」)を設定します。

【設定項目】

```
<date-time-format-config>
  <format-set>
    .....
    <format type="date">
      .....
    </format>
  </format-set>
</date-time-format-config>
```

必須 ☐

項目

複数 ☐

設定

設定 なし

値・

設定

する

内容

単 なし

位・

型

省略 なし

時の

デ

フォ

ルト

値

親タ format-set

グ

【属性】

属性名	説明	必須
type	date 日付用のフォーマットとして扱われます。	☐
	time 時刻用のフォーマットとして扱われます。	

「日付と時刻の形式」フォーマット識別子の設定

タ format-id

グ

名

フォーマットを識別するための識別子を設定します。

【設定項目】

```
<date-time-format-config>
  <format-set>
    .....
    <format type="date">
      <format-id>IM_DATETIME_FORMAT_DATE_STANDARD</format-i
      .....
    </format>
  </format-set>
</date-time-format-config>
```

必 ☐

須

項

目

複数設定	x
設定値・設定する内容	なし
単位・型	文字列
省略時のデフォルト値	なし
親タグ	format

「日付と時刻の形式」フォーマットパターンの設定

タグ名	pattern
-----	---------

ユーザに提供するフォーマットパターン文字列を設定します。

【設定項目】

```
<date-time-format-config>
<format-set>
  ....
  <format type="date">
    ....
    <pattern default="true">MMM d, yyyy</pattern>
    <pattern>MMM dd, yyyy</pattern>
    ....
  </format>
</format-set>
</date-time-format-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	フォーマットパターン文字列
単位・型	文字列
省略時のデフォルト値	なし
親タグ	format

【属性】

属性名	説明	必須
default	true の場合、初期状態におけるデフォルト値となります。	×

ロケールマスタ

- 項目
- 概要
 - リファレンス
 - ロケールの設定
 - 文字エンコーディングの設定

概要

intra-mart Accel Platform で利用するロケールの設定を行います。

モジュール	国際化機能
フォーマットファイル(xsd)	WEB-INF/schema/locale-config.xsd
設定場所	WEB-INF/conf/locale-config/{任意のファイル名}.xml

```
<?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 http://intra_mart.co.jp/system/i18n/locale/config/locale-config.xsd">
  <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>
</locale-config>
```

リファレンス

ロケールの設定

タグ名
locale

intra-mart Accel Platform で利用するロケールを定義します。

【設定項目】

```
<locale-config>
  <locale name="en">
    . . . . .
  </locale>
</locale-config>
```

必須 ☐

項目

複数 ☐

設定

設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	locale-config

【属性】

属性名	説明	必須
name	ロケール ID を設定します。	○
default	true の場合、システム・デフォルトのロケールになります。	×

 **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、そのまま運用を行う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。

 **注意**

default="true" のロケールが複数ある場合、設定ファイルの先頭: default="true" のロケールが見つからない場合、設定ファイルで

文字エンコーディングの設定

タグ名	encoding-name
-----	---------------

intra-mart Accel Platform で利用する文字エンコーディングを定義します。

【設定項目】

```
<locale-config>
<locale name="en">
  <encoding-name>UTF-8</encoding-name>
</locale>
</locale-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	文字エンコーディングを設定します。
単位・型	文字列

省略時のデフォルト値	なし
親タグ	locale

タイムゾーンマスタ

- 項目
- 概要
 - リファレンス
 - タイムゾーンの設定

概要

intra-mart Accel Platform で利用するタイムゾーンの設定を行います。

モジュール	国際化機能
フォーマットファイル(xsd)	WEB-INF/schema/time-zone-config.xsd
設定場所	WEB-INF/conf/time-zone-config/{任意のファイル名}

```
<time-zone-config
xmlns="http://www.intra_mart.co.jp/system/i18n/timezone/config/time-zone"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.intra_mart.co.jp/system/i18n/timezone/config/time-zone http://www.intra_mart.co.jp/system/i18n/timezone/config/time-zone.xsd">
  <time-zone-id>Asia/Tokyo</time-zone-id>
  <time-zone-id>Asia/Shanghai</time-zone-id>
  <time-zone-id>UTC</time-zone-id>
  <time-zone-id>America/New_York</time-zone-id>
</time-zone-config>
```

リファレンス

タイムゾーンの設定

タグ名	time-zone-id
-----	--------------

intra-mart Accel Platform で利用するタイムゾーンをタイムゾーン ID によって指定します。

【設定項目】

```
<time-zone-config>
  <time-zone-id>Asia/Tokyo</time-zone-id>
</time-zone-config>
```

必須項目	☐
複数設定	☐
設定値・設定する内容	タイムゾーン ID
単位・データ型	文字列

省略時のデフォルト値	なし
親タグ	time-zone-config

 注意

設定可能なタイムゾーン ID は、JDK でサポートされるタイムゾーン

スクリプト開発モデル

source-config.xml

- 項目
- 概要
 - リファレンス
 - 文字エンコーディングの設定
 - JavaScriptの設定
 - JavaScriptコンパイラの設定
 - 最適化レベルの設定
 - Viewの設定
 - Viewコンパイラの設定
 - XMLエスケープの設定
 - JavaScriptエスケープの設定
 - ID属性を自動的に付加する設定
 - HTMLコメントを削除する設定
 - source-config.xmlの有効範囲
 - ファイル単位での設定方法

概要

source-config.xmlは、スクリプト開発モデルのプログラムソースおよび実行この設定ファイルは、ディレクトリに対して有効であり、そのディレクトリのサフ下のすべてのプログラムソースに対して設定内容を適用する事ができます。

モジュール	スクリプト開発モデル
フォーマットファイル (xsd)	なし
設定場所	スクリプト開発モデルのプログラムが配置されて

```
<resource-file>
  <charset>UTF-8</charset>
  <javascript>
    <compiler enable="true" />
    <!-- enable:true = Auto compiler to Java class -->
    <!-- enable:false = Interpreter -->

    <optimize level="0" />
    <!-- level:0 to 9 = Optimize level of Compile -->
  </javascript>
  <view>
    <compiler enable="true" />
    <!-- enable:true = Auto compiler -->
    <!-- enable:false = Interpreter -->
  </view>
</resource-file>
```

リファレンス

タグ名

プログラムソースの文字エンコーディングの指定です。intra-mart Accel Pla

【設定項目】

```
<resource-file>
  <charset>UTF-8</charset>
</resource-file>
```

必須項目	x
複数設定	x
設定値・設定する内容	文字エンコーディング名
単位・型	なし
省略時のデフォルト値	上位ディレクトリに配置された source-config.xml の設定
親タグ	resource-file

注意

intra-mart Accel Platform および関連するアプリケーション製品
この設定には必ず「UTF-8」を指定してください。

JavaScriptの設定

タグ名

スクリプト開発モデルのJavaScriptに関する設定を行います。

【設定項目】

```
<resource-file>
  <javascript>
  </javascript>
</resource-file>
```

必須項目	x
複数設定	x
設定値・設定する内容	JavaScriptの各種設定を指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	resource-file

JavaScriptコンパイラの設定

タグ名

JavaScriptコンパイラに関する設定を行います。

【設定項目】

```
<resource-file>
  <javascript>
    <compiler enable="true" />
  </javascript>
</resource-file>
```

必須項目 x

複数設定 x

設定値・設定する内容 JavaScriptコンパイラの設定を指定します。

単位・型 なし

省略時のデフォルト値 なし

親タグ javascript

【属性】

属性名	説明
enable	JavaScriptコンパイラの有効・無効を指定します。 <i>true</i> を設定した場合、JavaScriptをJavaメモリ上にキャッシュし、以後のプログラム上しますが、ソースの変更が一切反映されなければいけません。この設定は、運用環境で有効にしてください。 <i>false</i> を設定した場合、JavaScriptをインラインで実行されます。開発をスムーズに進めることができません。この設定は、開発環境で有効にしてください。 IM-JugglingのビルドウィザードからWARファイル生成時に「WARファイル生成テスト環境」を選択していると、source-compatibility-modeがインタプリタモードで実行されます。

最適化レベルの設定

タグ名 optimize

JavaScriptコンパイラの最適化に関する設定を行います。

【設定項目】

```
<resource-file>
  <javascript>
    <optimize level="0" />
  </javascript>
</resource-file>
```

必須項目 x

複数設定 x

設定値・設定する内容 JavaScriptコンパイラの最適化レベルを指定します。

単位・型 なし

省略時のデフォルト値 なし

親タグ javascript

【属性】

属性名	説明
level	JavaScriptファイルのプログラム解析における大きくなります。0を設定した場合は、最適化はプログラムを最適化して実行する事は、パフォーマンスが複雑な機能な為、プログラムの内容最適化レベルを適用する場合は、その適用範囲

 **注意**

intra-mart Accel Platform および関連するアプリケーション製品
この設定には必ず「0」を指定してください。

Viewの設定

タグ名

スクリプト開発モデルのHTMLに関する設定を行います。

【設定項目】

```
<resource-file>
  <view>
</view>
</resource-file>
```

必須項目	×
複数設定	×
設定値・設定する内容	HTMLの各種設定を指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	resource-file

Viewコンパイラの設定

タグ名

Viewコンパイラに関する設定を行います。

【設定項目】

```
<resource-file>
  <view>
    <compiler enable="true" />
  </view>
</resource-file>
```

必須項目	×
複数設定	×

設定値・ Viewコンパイラの設定を指定します。
設定する
内容

単位・型 なし

省略時の
デフォル
ト値

親タグ view

【属性】

属性名	説明
enable	Viewコンパイラの有効・無効を指定します。 <i>true</i> を設定した場合、HTMLをコンパイル上にキャッシュされ、以後のプログラム実行が向上しますが、ソースの変更が一切反映されなければいけません。この設定は、運用環境で有効にしてください。 <i>false</i> を設定した場合、HTMLをインタプリタで実行します。開発をスムーズに進めることができません。この設定は、開発環境で有効にしてください。 IM-JugglingのビルドウィザードからWARファイル生成時に「開発環境」を選択していると、source-codeがインタプリタモードで実行されます。

XMLエスケープの設定

タグ
escapeXml
タグ
名

XMLエスケープに関する設定を行います。
<imart> タグの属性に指定された文字列に対してXMLエスケープを行う設定を行います。
詳しい説明はAPIリストの [エスケープ処理に関して](#) を参照してください。



コラム

全ての<imart>タグがエスケープ機能に対応しているわけでは
ありません。ドキュメントを参照してください。

【設定項目】

```
<resource-file>
  <view>
    <escapeXml enable="true" />
  </view>
</resource-file>
```

必須項目 ×

複数設定 ×

設定値・ XMLエスケープの各種設定を指定します。
設定する
内容

単位・型 なし

省略時の
デフォルト
値

親タグ view

【属性】

属性名	説明
enable	XMLエスケープの有効・無効を指定します。

属性名	説明
exclusion	XMLエスケープ処理の対象外とする文字列を指
delimiter4exclusion	XMLエスケープ処理の対象外とする文字列のシ

JavaScriptエスケープの設定

タグ名	escapeJs
-----	----------

JavaScriptエスケープに関する設定を行います。
<imart> タグの属性に指定された文字列に対してJavaScriptエスケープを行います。
詳しい説明はAPIリストの [エスケープ処理に関して](#) を参照してください。

i

コラム

全ての<imart>タグがエスケープ機能に対応しているわけではあ
ドキュメントを参照してください。

【設定項目】

```
<resource-file>
  <view>
    <escapeJs enable="true" />
  </view>
</resource-file>
```

必須項目	x
複数設定	x
設定値・設定内容	JavaScriptエスケープの各種設定を指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	view

【属性】

属性名	説明
enable	JavaScriptエスケープの有効・無効を指定しま
exclusion	JavaScriptエスケープ処理の対象外とする文字
delimiter4exclusion	JavaScriptエスケープ処理の対象外とする文字

ID属性を自動的に付加する設定

タグ名	replicateNameTold
-----	-------------------

<imart type="hidden"> タグ使用時に、name属性に指定された値をid属性
例えば、name属性に「foo」と値が指定された場合、プログラム実行時に以

```
<INPUT type="hidden" name="foo" id="foo">
```

【設定項目】

```
<resource-file>
  <view>
    <replicateNameTold enable="true" />
  </view>
</resource-file>
```

必須項目	x
------	---

複数設定	x
------	---

設定値・設定する内容	ID属性を自動的に付加する設定を指定します。
------------	------------------------

単位・型	なし
------	----

省略時のデフォルト値	なし
------------	----

親タグ	view
-----	------

【属性】

属性名	説明
enable	• <i>true</i> を設定した場合、name属性に設定された • <i>false</i> を設定した場合、id属性を付加しません。



注意

intra-mart Accel Platform および関連するアプリケーション製品
replicateNameTold は対応しているプログラムソースにのみ使用

HTMLコメントを削除する設定

タグ名	removeHTMLComments
-----	--------------------

プログラム実行時に「`<!-- -->`」で囲まれているHTMLコメント部分を削除し

【設定項目】

```
<resource-file>
  <view>
    <removeHTMLComments enable="true" />
  </view>
</resource-file>
```

必須項目	x
------	---

複数設定	x
------	---

設定値・設定する内容	HTMLコメントを削除する設定を指定します。
------------	------------------------

単位・型	なし
------	----

省略時のデフォルト値	なし
------------	----

親タグ	view
-----	------

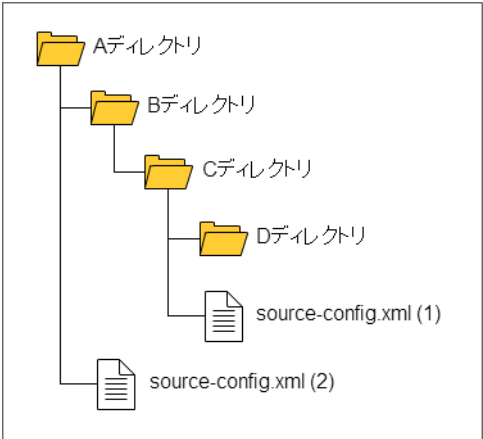
【属性】

属性名	説明
enable	• <i>true</i> を設定した場合、HTMLコメントを削除します • <i>false</i> を設定した場合、HTMLコメントを削除しま

source-config.xmlの有効範囲

下図のようにsource-config.xmlを配置した場合、各プログラムが参照する記

- Aディレクトリ直下のプログラム:(2)の設定内容が有効になります。
- Bディレクトリ直下のプログラム:(2)の設定内容が有効になります。
- Cディレクトリ直下のプログラム:(1)の設定内容が有効になります。
- Dディレクトリ直下のプログラム:(1)の設定内容が有効になります。



ファイル単位での設定方法

スクリプト開発モデルのプログラムは、HTMLファイルとJSファイルのペア単
「対象ファイルラベル名.properties」ファイルを作成し、以下のように設定す
ファイルラベル名.properties」ファイルの設定内容が有効になります。

```
charset=プログラムの文字エンコーディング名
javascript.compile.enable=JavaScriptコンパイラの設定
javascript.optimize.level=JavaScriptコンパイラの最適化レベル
view.compile.enable=Viewコンパイラの設定
view.escapeXml.enable=XMLエスケープの有効・無効
view.escapeXml.exclusion=XMLエスケープ処理の対象外とする文字列
view.escapeXml.delimiter4exclusion=「view.escapeXml.exclusion」に指定
view.escapeJs.enable=JavaScriptエスケープの有効・無効
view.escapeJs.exclusion=JavaScriptエスケープ処理の対象外とする文字列
view.escapeJs.delimiter4exclusion=「view.escapeJs.exclusion」に指定し
view.replicateNameToId.enable=ID属性を自動的に付加する設定
view.removeHTMLComments.enable=HTMLコメントを削除する設定
```

例えば、ファイルが文字コード「UTF-8」で作成されたsample.html とsampl
作成し、以下の内容を記述します。

```
charset=UTF-8
javascript.compile.enable=true
javascript.optimize.level=0
view.compile.enable=false
```

データベースアクセス機能

データソースマッピング設定

項目
● 概要
● リファレンス
○ システムデータベース設定
○ シェアードデータベース設定
○ テナントデータベース設定
○ リソース参照名
○ データベース種別
○ シェアードデータベース接続ID
○ テナントデータベース テナントID

概要

intra-mart Accel Platform で利用するデータベースと Web Application S

モジュール	データベースアクセスモジュール
フォーマットファイル(xsd)	WEB-INF/schema/data-source-mapping-config.xml
設定場所	WEB-INF/conf/data-source-mapping-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<data-source-mapping-config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://intra-mart.co.jp/system/database/config/data-source-mapping"
  xsi:schemaLocation="http://intra-mart.co.jp/system/database/config/data-source-mapping http://intra-mart.co.jp/system/database/config/data-source-mapping.xsd">
  <system-data-source>
    <resource-ref-name>java:comp/env/jdbc/default</resource-ref-name>
    <database-name>oracle</database-name>
  </system-data-source>
  <shared-data-source>
    <connect-id>default</connect-id>
    <resource-ref-name>java:comp/env/jdbc/default</resource-ref-name>
  </shared-data-source>
  <tenant-data-source>
    <tenant-id>default</tenant-id>
    <resource-ref-name>java:comp/env/jdbc/default</resource-ref-name>
  </tenant-data-source>
</data-source-mapping-config>
```

リファレンス

システムデータベース設定

タグ名
system-data-source

システムデータベースとして利用するデータソースを設定します。

【設定項目】

```
<data-source-mapping-config>
  <system-data-source>
    <resource-ref-name>XXXXXXXXXXXX</resource-ref-name>
    <database-name>XXXXXXXXXXXX</database-name>
  </system-data-source>
</data-source-mapping-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	resource-ref-name システムデータベースとし database-name 接続データベースの種類
単位・型	なし
省略時のデフォルト値	なし
親タグ	data-source-mapping-config

シェアードデータベース設定

タグ名
shared-data-source

シェアードデータベースとして利用するデータソースを設定します。

【設定項目】

```
<data-source-mapping-config>
<shared-data-source>
  <connect-id>XXXXXXXXXXXXX</connect-id>
  <resource-ref-name>XXXXXXXXXXXXX</resource-ref-name>
  <database-name>XXXXXXXXXXXXX</database-name>
</shared-data-source>
</data-source-mapping-config>
```

必須項目	x						
複数設定	○						
設定値・設定する内容	<table> <tr> <td>connect-id</td><td>シェアードデータベースの</td></tr> <tr> <td>resource-ref-name</td><td>シェアードデータベースと</td></tr> <tr> <td>database-name</td><td>接続データベースの種類</td></tr> </table>	connect-id	シェアードデータベースの	resource-ref-name	シェアードデータベースと	database-name	接続データベースの種類
connect-id	シェアードデータベースの						
resource-ref-name	シェアードデータベースと						
database-name	接続データベースの種類						
単位・型	なし						
省略時のデフォルト値	なし						
親タグ	data-source-mapping-config						

テナントデータベース設定

タグ
tenant-data-source

テナントデータベースとして利用するデータソースを設定します。

【設定項目】

```
<data-source-mapping-config>
<tenant-data-source>
  <tenant-id>XXXXXXXXXXXXX</tenant-id>
  <resource-ref-name>XXXXXXXXXXXXX</resource-ref-name>
  <database-name>XXXXXXXXXXXXX</database-name>
</tenant-data-source>
</data-source-mapping-config>
```

必須項目	x						
複数設定	○						
設定値・設定する内容	<table> <tr> <td>tenant-id</td><td>このデータベース設定を利用するテナントデータベースのID</td></tr> <tr> <td>resource-ref-name</td><td>テナントデータベースとして利用するデータソースの種類</td></tr> <tr> <td>database-name</td><td>接続データベースの種類</td></tr> </table>	tenant-id	このデータベース設定を利用するテナントデータベースのID	resource-ref-name	テナントデータベースとして利用するデータソースの種類	database-name	接続データベースの種類
tenant-id	このデータベース設定を利用するテナントデータベースのID						
resource-ref-name	テナントデータベースとして利用するデータソースの種類						
database-name	接続データベースの種類						
単位・型	なし						
省略時のデフォルト値	なし						
親タグ	data-source-mapping-config						

リソース参照名

タグ
resource-ref-name

Web Application Server に設定されているリソース参照名を指定する項目

【設定項目】

```
<data-source-mapping-config>
<system-data-source>
  <resource-ref-name>XXXXXXXXXXXXX</resource-ref-name>
</system-data-source>
</data-source-mapping-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	intra-mart Accel Platform で利用するデータソースの内容
単位・型	文字列 (java.comp/env/xxxxx)
省略時のデフォルト値	なし
親タグ	system-data-source, shared-data-source, tenant-d

データベース種別

タグ名 database-name

接続先データベースの種別を指定する項目です。
この項目が指定されていない場合は、データベースの種別は接続先データ

【設定項目】

```
<data-source-mapping-config>
  <system-data-source>
    <database-name>XXXXXXXXXXXXXX</database-name>
  </system-data-source>
</data-source-mapping-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	接続先データベースの種類
単位・型	文字列 (oracle, sql server, db2, postgresql)
省略時のデフォルト値	接続先データベースのデータベース種別にあわせて
親タグ	system-data-source, shared-data-source, tenant-c



注意
指定されたデータベースの種別と接続先データベースの種別が

シェアードデータベース接続ID

タグ名 connect-id

シェアードデータベースの接続IDを指定する項目です。
接続IDには各シェアードデータベース毎の一意のIDを設定してください。

【設定項目】

```
<data-source-mapping-config>
  <shared-data-source>
    <connect-id>XXXXXXXXXXXXXX</connect-id>
  </shared-data-source>
</data-source-mapping-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	シェアードデータベースの接続ID

単位・型	文字列 (xxxxxxxxxxxxxx)
省略時のデフォルト値	なし
親タグ	shared-data-source

テナントデータベース テナントID

タグ名	tenant-id
-----	-----------

このデータベース設定を利用するテナントのテナントIDを指定する項目です

【設定項目】

```
<data-source-mapping-config>
<tenant-data-source>
  <tenant-id>xxxxxxxxxxxxxx</tenant-id>
</tenant-data-source>
</data-source-mapping-config>
```

必須項目	☐
複数設定	×
設定値・設定する内容	テナントID
単位・型	文字列 (xxxxxxxxxxxxxx)
省略時のデフォルト値	なし
親タグ	tenant-data-source

メールモジュール

メール設定

項目
● 概要
● リファレンス
○ SMTPの設定
○ SMTPサーバ設定
○ SMTPS(SMTP over SSL)設定
○ SMTP Authentication設定
○ SMTP Authenticationユーザ設定
○ SMTP Authenticationパスワード設定
○ X-Mailerヘッダ設定
○ デバッグ設定
○ コネクションタイムアウト設定
○ タイムアウト設定
○ コンテンツタイプ群
○ コンテンツタイプ設定
○ リスナ群設定
○ リスナ設定
○ メールヘッダ群設定
○ メールヘッダ設定
○ メールエンコード設定
○ 文字コード設定

- [MIMEエンコード設定](#)
- [ボディエンコード設定](#)

概要

メール送信に関する設定です。
デフォルトの設定としてjavamail-config.xmlが用意されています。
多言語対応として、javamail-config_{ロケールID}.xmlが存在します。

モジュール	メールモジュール
フォーマットファイル(xsd)	WEB-INF/schema/javamail-config.xsd
設定場所	WEB-INF/conf/javamail-config/javamail-config.xr WEB-INF/conf/javamail-config/{任意のファイル名}

```
<?xml version="1.0" encoding="UTF-8"?>
<javamail-config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://intra-mart.co.jp/system/mail/javamail/config/javamail-config"
  xsi:schemaLocation="http://intra-mart.co.jp/system/mail/javamail/config"

  <smtp>
    <smtp-server id="default" host="localhost" port="25">
      <smtps enable="false" starttls="false"/>
      <auth enable="false">
        <user/>
        <password/>
      </auth>
      <x-mailer>intra-mart MailSender ver 8.0</x-mailer>
      <debug>false</debug>
      <connection-timeout>-1</connection-timeout>
      <timeout>-1</timeout>
    </smtp-server>
  </smtp>

  <content-type>
    <file extension=".txt" content-type="text/plain"/>
    <file extension=".gif" content-type="image/gif"/>
    <file extension=".jpg" content-type="image/jpeg"/>
    <file extension=".png" content-type="image/png"/>
  </content-type>

  <listener>
    <listener-class>jp.co.intra_mart.foundation.mail.javamail.listener.impl.
  </listener>

  <encode>
    <charset>UTF-8</charset>
    <mime-encoding>B</mime-encoding>
    <content-transfer-encoding>7bit</content-transfer-encoding>
  </encode>

</javamail-config>
```

リファレンス

SMTPの設定

タグ	smtp-server
グループ	
名前	

SMTPの設定を行います。
メールの送信時に使用するSMTPの設定を行います。

【設定項目】

```
<javamail-config>
<smtp>
  <smtp-server id="default" host="localhost" port="25">
    ....
  </smtp-server>
</smtp>
</javamail-config>
```

必須項目	x
複数設定	x
設定値・設定する内容	SMTPの各種設定を指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	javamail-config

SMTPサーバ設定

タグ名	smtp-server
-----	-------------

SMTPサーバの設定を行います。

複数のSMTPサーバの定義を行うことが可能です。
APIからのメール送信時にSMTPサーバが指定されていない場合、テナントテナントIDと同じIDのSMTPサーバの設定が存在しない場合に利用される。

【設定項目】

```
<javamail-config>
<smtp>
  <smtp-server id="default" host="localhost" port="25">
    ....
  </smtp-server>
</smtp>
</javamail-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	SMTPサーバの設定を指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	smtp

【属性】

属性名	説明
id	このSMTPサーバの設定を特定する一意のID APIからのメール送信時にこのIDを指定する時にIDを指定しなかった場合、テナントIDまたは、標準設定 (default) が有効になります。
host	SMTPサーバのホスト名またはIPアドレスを

属性名	説明
port	SMTPサーバのポート名を指定します。

SMTPS(SMTP over SSL)設定

タグ名

SMTPS(SMTP over SSL)の設定を行います。
SMTPSを利用するにはJavaランタイムがSMTPサーバのSSL証明書を信頼するSSL証明書が信頼されていない場合、javax.net.ssl.SSLHandshakeExceptionがJavaランタイムがSMTPサーバのSSL証明書を信頼済みであることを確認し

【設定項目】

```
<javamail-config>
<smtp>
  <smtp-server id="default" host="localhost" port="25">
    <smtps enable="true" starttls="false"/>
    ....
  </smtp-server>
</smtp>
</javamail-config>
```

必須項目	x
複数設定	x
設定値・設定内容	SMTPS(SMTP over SSL)の設定を指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	smtp-server

【属性】

属性名	説明
enable	SMTPサーバとの通信をSSLによる暗号化通信場合はfalseを設定してください。
starttls	TLS/STARTTLSによる暗号化通信を行います、falseを設定してください。 この設定を有効にする場合はenable属性も有効

SMTP Authentication設定

タグ名

SMTP Authenticationを利用する場合に指定する要素です。

【設定項目】

```
<javamail-config>
<smtp>
  <smtp-server id="default" host="localhost" port="25">
    <auth enable="false">
      ....
    </auth>
  </smtp-server>
</smtp>
</javamail-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	SMTP Authenticationを利用する際の設定を指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	smtp-server

【属性】

属性名	説明
enabled	SMTP Authenticationの利用設定を行います。場合はfalseを設定してください。

SMTP Authenticationユーザ設定

タグ名

user

SMTP Authenticationを利用する際のユーザを設定します。

【設定項目】

```
<javamail-config>
<smtp>
  <smtp-server id="default" host="localhost" port="25">
    <auth enable="true">
      <user>USER</user>
    ...
  </auth>
</smtp-server>
</smtp>
</javamail-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	SMTP Authenticationを利用する際のユーザを設定します。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	auth

SMTP Authenticationパスワード設定

タグ名

password

SMTP Authenticationを利用する際のパスワードを設定します。

【設定項目】

```
<javamail-config>
<smtp>
  <smtp-server id="default" host="localhost" port="25">
    <auth enable="true">
      ...
      <password>PASSWORD</password>
    </auth>
  </smtp-server>
</smtp>
</javamail-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	SMTP Authenticationを利用する際のパスワードを設定する内容
単位・型	文字列
省略時のデフォルト値	なし
親タグ	auth

X-Mailerヘッダ設定

タグ x-mailer

X-Mailerメールヘッダに指定する値の設定を行います。

【設定項目】

```
<javamail-config>
<smtp>
  <smtp-server id="default" host="localhost" port="25">
    ...
    <x-mailer>intra-mart MailSender ver 8.0</x-mailer>
    ...
  </smtp-server>
</smtp>
</javamail-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	メール送信時にX-Mailerヘッダとして送信する値を設定する内容
単位・型	文字列
省略時のデフォルト値	なし
親タグ	smtp-server

デバッグ設定

タグ debug

JavaMailにデバッグオプションを設定します。

【設定項目】

```
<javamail-config>
<smtp>
  <smtp-server id="default" host="localhost" port="25">
    ...
    <debug>false</debug>
    ...
  </smtp-server>
</smtp>
</javamail-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	false デバッグ設定を行いません。 true デバッグ設定を行います。
単位・型	真偽値 (true/false)

省略時のデフォルト値	なし
親タグ	smtp-server

コネクションタイムアウト設定

タグ名	connection-timeout
-----	--------------------

SMTPサーバ接続時のコネクションタイムアウト値を設定します。

【設定項目】

```
<javamail-config>
<smtp>
  <smtp-server id="default" host="localhost" port="25">
    ...
    <connection-timeout>-1</connection-timeout>
    ...
  </smtp-server>
</smtp>
</javamail-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	SMTPサーバ接続時のコネクションタイムアウト値を設定します。 -1が設定された場合はコネクションタイムアウトは発生し、設定可能な数値の単位はミリ秒です。
単位・型	数値(1以上 または -1)
省略時のデフォルト値	なし
親タグ	smtp-server

タイムアウト設定

タグ名	timeout
-----	---------

メール送信時におけるタイムアウト値を設定します。

【設定項目】

```
<javamail-config>
<smtp>
  <smtp-server id="default" host="localhost" port="25">
    ...
    <timeout>-1</timeout>
    ...
  </smtp-server>
</smtp>
</javamail-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	メール送信時におけるタイムアウト値を設定します。 -1が設定された場合はタイムアウトは発生しません。 設定可能な数値の単位はミリ秒です。
単位・型	数値(1以上 または -1)
省略時のデフォルト値	なし
親タグ	smtp-server

タグ名

メール送信時に添付するファイルのコンテンツタイプ群の設定を行います。

【設定項目】

```
<javamail-config>
<content-type>
...
</content-type>
</javamail-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	メール送信時に添付するファイルのコンテンツタイプ群
単位・型	なし
省略時のデフォルト値	なし
親タグ	javamail-config

コンテンツタイプ設定

タグ名

メール送信時に添付するファイルのコンテンツタイプの設定を行います。
コンテンツタイプの設定は添付ファイルの拡張子に対しての設定となります。

【設定項目】

```
<javamail-config>
<content-type>
  <file extension=".txt" content-type="text/plain"/>
  <file extension=".gif" content-type="image/gif"/>
  <file extension=".jpg" content-type="image/jpeg"/>
  <file extension=".png" content-type="image/png"/>
</content-type>
</javamail-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	メール送信時に添付するファイルのコンテンツタイプを
単位・型	なし
省略時のデフォルト値	なし
親タグ	content-type

【属性】

属性名	説明
extension	拡張子を指定します。必ず、で始まる形式で記述してくだ
content-type	コンテンツタイプ属性です。拡張子に対応したコンテンツ

リスナ群設定

タグ
listener

メール送信時に動作するリスナ群を指定します。

【設定項目】

```
<javamail-config>
<listener>
  <listener-class></listener-class>
</listener>
</javamail-config>
```

必須項目	x
複数設定	x
設定値・設定する内容	メール送信時に動作するリスナ群を指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	javamail-config

リスナ設定

タグ
listener-class

メール送信時に動作するリスナを指定します。
指定するリスナは、jp.co.intra_mart.foundation.mail.javamail.listener.Mai
デフォルトでは、半角カナを全角カナに変換する為のリスナ (jp.co.intra_ma

【設定項目】

```
<javamail-config>
<listener>
  <listener-class></listener-class>
</listener>
</javamail-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	メール送信時に動作するリスナを指定します。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	listener

メールヘッダ群設定

タグ
mail-headers

メールヘッダ群を指定します。

【設定項目】

```
<javamail-config>
<mail-headers>
...
</mail-headers>
</javamail-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	メール送信時に付与されるメールヘッダ群を指定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	javamail-config

メールヘッダ設定

タグ名

header

メール送信時に付与する任意のメールヘッダを指定します。
同じヘッダが複数指定された場合は、最下行に指定されたものがヘッダとしAPI (jp.co.intra_mart.foundation.mail.javamail.StandardMail#addHead

【設定項目】

```
<javamail-config>
<mail-headers>
<header name="Error-To" value="xxxx@intra-mart.jp" />
</mail-headers>
</javamail-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	メール送信時に付与する任意のメールヘッダを指定します。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	mail-headers

メールエンコード設定

タグ名

encode

ロケールに対応したメールエンコードに関する設定を行います。
locale属性が設定されていない要素は、ロケールに対応したメールエンコー

【設定項目】

```
<javamail-config>
<encode>
<charset>UTF-8</charset>
<mime-encoding>B</mime-encoding>
<content-transfer-encoding>7bit</content-transfer-encoding>
</encode>
</javamail-config>
```

必須項目	×
複数設定	○

設定値・設定内容
ローケルに対応したメールエンコードに関する設定を行う内容

親タグ javamail-config

【属性】

		デフォルト	
属性名	説明	必須	値
locale	ローケルIDを指定します。	×	なし

文字コード設定

タグ charset
名

文字コードの指定を行います。

【設定項目】

```
<javamail-config>
<encode>
  <charset>UTF-8</charset>
  ...
</encode>
</javamail-config>
```

必須項目 ○

複数設定 ×

設定値・設定内容
文字コードの指定を行います。

単位・型 文字列

省略時のデフォルト値 なし

親タグ encode

MIMEエンコード設定

タグ mime-encoding
名

MIMEエンコードの指定を行います。
指定可能な値は B または Q です。

【設定項目】

```
<javamail-config>
<encode>
  ...
  <mime-encoding>B</mime-encoding>
  ...
</encode>
</javamail-config>
```

必須項目 ○

複数設定 ×

設定値・ MIMEエンコードの指定を行います。
設定する
内容

単位・型 文字列

省略時のデフォルト値

親タグ encode

ボディエンコード設定

タグ content-transfer-encoding
名

ボディのエンコードの指定を行います。
7bit または 8bit 、 Quoted Printable等の値を指定します。

【設定項目】

```
<javamail-config>
<encode>
...
<content-transfer-encoding>7bit</content-transfer-encoding>
</encode>
</javamail-config>
```

必須項目 ☐

複数設定 ☒

設定値・ ボディのエンコードの指定を行います。
設定する
内容

単位・型 文字列

省略時のデフォルト値

親タグ encode

Webモジュール

リクエスト制御設定

項目

- [概要](#)
- [リファレンス](#)
 - [サーバの同時処理の制限](#)
 - [リクエストの監視](#)
 - [リクエストの処理制限](#)
 - [リクエストクエリの制限](#)
 - [リクエストクエリの長さ制限](#)
 - [リクエスト処理の直列化](#)
 - [直列処理を行うリクエストクエリの制限](#)
 - [直列処理を行うリクエストクエリの長さ制限](#)

概要

リクエスト制御の設定です。
リクエストの同時処理制限やクエリの長さによるページ処理を制御する機能

モジュール Webモジュール

フォーマットファイル WEB-INF/schema/request-control-config.xsd
イル(xsd)

```
<?xml version="1.0" encoding="UTF-8"?>
<request-control-config
  xmlns="http://intra-mart.co.jp/system/servlet/filter/xml/request_control/r
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://intra-mart.co.jp/system/servlet/filter/xml/req
<thread max="32">
  <inspection keep-alive="3000"/>
</thread>
<accept queue="100">
  <query>
    <length max="0"/>
  </query>
</accept>
<synchronized queue="8">
  <query>
    <length min="1048576"/>
  </query>
</synchronized>
</request-control-config>
```

リファレンス

サーバの同時処理の制限

タグ名

サーバの同時処理制限を指定します。
制限数を超えるリクエストを受け付けた場合、制限数を超えたリクエストに
maxに 0 を指定するとリクエストの同時処理数を制限しません。

【設定項目】

```
<request-control-config>
  <thread max="32">
    ...
  </thread>
</request-control-config>
```

必須項目 ○

複数設定 x

設定値・設定する内容
サーバの同時処理制限を設定します。

単位・型 なし

省略時のデフォルト値
なし

親タグ request-control-config

【属性】

属性名	説明	必須	デフォルト値
max	サーバの最大同時処理数を設定します。	○	なし

リクエストの監視

タグ名

処理中のリクエストの有効期間を指定します。
処理を開始してから指定された時間が経過しても終了していないリクエスト

```
<request-control-config>
  <thread max="32">
    <inspection keep-alive="3000"/>
  </thread>
</request-control-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	処理中のリクエストの有効期間を設定します。
単位・型	数値[ミリ秒]
省略時のデフォルト値	なし
親タグ	thread

【属性】

属性名	説明	必須
keep-alive	処理中のリクエストの有効期間を指定します。	○

リクエストの処理制限

タグ名

accept

サーバが受け付けたリクエストの処理制限を指定します。
同時処理数を超えた場合の処理待ちを行うリクエストの最大数やリクエスト

【設定項目】

```
<request-control-config>
  <accept queue="100">
    ...
  </accept>
</request-control-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	リクエストの処理制限を設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	request-control-config

【属性】

属性名	説明	必須
queue	処理待ちを行うリクエストの最大数を設定します。	○

リクエストクエリの制限

タグ名

リクエストクエリの制限を指定します。
リクエストクエリに対する制限を行います。

【設定項目】

```
<request-control-config>
  <accept queue="100">
    <query>
      ...
    </query>
  </accept>
</request-control-config>
```

必須項目	☐
複数設定	☒
設定値・設定する内容	リクエストクエリの制限を設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	accept

リクエストクエリの長さ制限

タグ名

リクエストクエリの長さ制限を指定します。
リクエストクエリの長さがこの設定値よりも大きい場合は、HTTPレスポンス: maxに 0 を指定するとリクエストクエリの長さによる制限しません。

【設定項目】

```
<request-control-config>
  <accept>
    <query>
      <length max="0"/>
      ...
    </length>
  </query>
</accept>
</request-control-config>
```

必須項目	☐
複数設定	☒
設定値・設定する内容	リクエストクエリの長さ制限を設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	query

【属性】

属性名	説明
max	この設定値に指定された長さよりリクエストクエリの長処理を行います。

タグ
名前

クエリサイズの大きいリクエストの処理に関する制限を指定します。
サイズの大きいリクエストを処理する場合に、リソース不足によるエラーが
この設定を行うとサイズの大きいリクエストは受付順に直列処理されます。

【設定項目】

```
<request-control-config>
  <synchronized queue="8">
    ...
  </synchronized>
</request-control-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	リクエストの処理制限を設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	request-control-config

【属性】

属性名	説明	必須
queue	処理待ちを行うリクエストの最大数を設定します。	○

直列処理を行うリクエストクエリの制限

タグ
名前

直列処理を行うリクエストのクエリの制限を指定します。
この設定値を基準に並列処理するリクエストと直列処理するリクエストを分

【設定項目】

```
<request-control-config>
  <synchronized queue="8">
    <query>
      ...
    </query>
  </synchronized>
</request-control-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	直列処理を行うリクエストのクエリの制限を設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	synchronized

直列処理を行うリクエストクエリの長さ制限

タグ
長さ
名前

直列処理を行うリクエストのクエリの長さを指定します。
minに指定されたサイズよりリクエストクエリのサイズが小さいリクエストは3

【設定項目】

```
<request-control-config>
  <synchronized>
    <query>
      <length min="1048576"/>
    </query>
  </synchronized>
</request-control-config>
```

必須項目	○
複数設定	×
設定値・設定内容	直列処理を行うリクエストのクエリの長さを設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	query

【属性】

属性名	説明	必須
min	直列処理を行うリクエストのクエリの長さを設定します。	○

レスポンスヘッダ設定

- 項目
- 概要
 - リファレンス
 - 静的なヘッダーの指定
 - 動的なヘッダーの指定

概要

レスポンスヘッダの設定です。
レスポンスに任意のヘッダを追加する機能を提供します。

モジュール	Webモジュール
フォーマットファイル	WEB-INF/schema/response-header-config.xsd
設定場所	WEB-INF/conf/response-header-config/default.xr

```
<?xml version="1.0" encoding="UTF-8"?>
<response-header-config
  xmlns="http://intra-mart.co.jp/system/servlet/filter/xml/response_header/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://intra-mart.co.jp/system/servlet/filter/xml/res
  <!-- Avoid IE Content Sniffing XSS Attacks -->
  <!--
  <static-response-header name="X-Content-Type-Options" value="nosn
  -->
</response-header-config>
```

静的なヘッダーの指定

タグ static-response-header
名

静的なレスポンスヘッダを指定します。

【設定項目】

```
<response-header-config>

  <static-response-header name="X-Content-Type-Options" value="nosniff">

  </static-response-header>

</response-header-config>
```

必須項目	x
複数設定	○
設定値・設定する内容	静的なレスポンスヘッダを設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	response-header-config

【属性】

属性名	説明
name	ヘッダー名です。
value	ヘッダー値です。
type	ヘッダー値の型です。text, int, date のいずれかです。
format	type 属性に date を指定した際、value を解読する際の value は long を指定します。
useAddHeader	ヘッダーを追加するかどうかのフラグです。デフォルトは false です。

動的なヘッダーの指定

タグ dynamic-response-header
名

静的なレスポンスヘッダを指定します。

【設定項目】

```
<response-header-config>

  <dynamic-response-header name="X-Dynamic-Sample" class="jp.co.intra.mart.accel.sample">

  </dynamic-response-header>

</response-header-config>
```

必須項目	x
複数設定	○
設定値・設定する内容	動的なレスポンスヘッダを設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	response-header-config

属性名	説明
name	ヘッダー名です。
class	ヘッダー値を返すJavaのクラス名です。jp.co.intra.mart.HEADERクラスを指定してください。
type	ヘッダー値の型です。text, int, date のいずれかです。
useAddHeader	ヘッダーを追加するかどうかのフラグです。デフォルトはtrueです。

IM-Propagation

IM-Propagation 設定

- 項目
- 概要
 - リファレンス
 - ロック時待機動作の設定
 - デッドロック検知時間の設定

概要

IM-Propagation 基本機能の設定です。

モジュール	IM-Propagation
フォーマットファイル(xsd)	WEB-INF/schema/propagation-config.xsd
設定場所	WEB-INF/conf/propagation-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<propagation-config xmlns="http://www.intra-mart.jp/propagation/config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/propagation/config propagation-config.xsd">
  <each-thread-sleep>100</each-thread-sleep>
  <max-thread-sleep>1000</max-thread-sleep>
</propagation-config>
```

リファレンス

ロック時待機動作の設定

タグ名	each-thread-sleep
-----	-------------------

データの受信側で IM-Propagation を使用して次のデータ送信を行う場合、

【設定項目】

```
<propagation-config>
  <each-thread-sleep>100</each-thread-sleep>
</propagation-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	待機する時間

単位・ 型	数値[ミリ秒] (0 - 1000)
省略時 のデ フォ ルト値	100
親タグ	propagation-config

 **注意**

通常はデフォルト値 (設定なし) のまま使用して、IM-Propagation

数値を小さく設定すると、ロック確認の間隔が短くなり処理が滞り、サーバー負荷が増加します。

数値を大きく設定すると、ロック確認の間隔が長くなりサーバー負荷が増加します。

デッドロック検知時間の設定

タグ名	max-thread-sleep
-----	------------------

次のデータ受信先がロック状態の場合に最大で待機する時間を設定します。この時間を超えた場合、デッドロックが発生したとみなされ、IM-Propagation

【設定項目】

```
<propagation-config>
<max-thread-sleep>1000</max-thread-sleep>
</propagation-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	デッドロック判定までの時間
単位・ 型	数値[ミリ秒] (0 - 60000)
省略時のデフォルト値	1000
親タグ	propagation-config

 **注意**

通常はデフォルト値 (設定なし) のまま使用して、IM-Propagation

数値を小さく設定すると、デッドロック検知が顕著になりデータの取得が遅くなります。

数値を大きく設定すると、デッドロック検知までの時間が長くなり、サーバー負荷が増加します。

IM-Propagation 受信側設定

項目
● 概要
● リファレンス
○ 受信データ定義の設定
○ 受信側データ変換クラスの設定
○ データ変換クラスパラメータの設定
○ データ変換クラスのパラメータ値の設定
○ 受信側データ処理クラスの設定
○ データ処理クラスパラメータの設定
○ データ処理クラスのパラメータ値の設定

IM-Propagation を利用してデータを伝搬するための受信側を設定します。
受信するデータを格納するクラスと、データ変換クラスを紐付けるための設

IM-Propagation を利用してデータを他モジュールから伝搬するためには、ま

データを伝搬する際に他モジュールとの依存関係を切り離す必要があるた
データの受信側は IM-Propagation 用のモデルから自モジュールで定義さ
そこで、データをコピーする作業を行うデータ変換クラスの設定を行います。

また、受信したデータをもとに処理を行うデータ処理クラスの設定も同時に

モジュール	IM-Propagation
フォーマットファイル (xsd)	WEB-INF/schema/propagation-receivers-conf
設定場所	WEB-INF/conf/propagation-receivers-config/{

 **注意**

ファイル名は、他のモジュールが提供しているものと重複しないよ

```
<?xml version="1.0" encoding="UTF-8"?>
<propagation-receivers-config xmlns="http://www.intra-mart.jp/propaga
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.intra-mart.jp/propagation/receivers-co
<receiver source="jp.co.intra_mart.foundation.sample.model.SampleMc
<decoder class="jp.co.intra_mart.system.sample.propagation.Sampleh
  <params>
    <param key="key1">value1</param>
    <param key="key2">value2</param>
  </params>
</decoder>
<procedure class="jp.co.intra_mart.system.sample.propagation.Sampl
  <params>
    <param key="key1">value1</param>
    <param key="key2">value2</param>
  </params>
</procedure>
</receiver>
</propagation-receivers-config>
```

リファレンス

受信データ定義の設定

タ receiver
グ
名

IM-Propagation を使用してデータを受信する側が受信するデータの型と如

【設定項目】

<propagation-receivers-config>
<receiver source="jp.co.intra_Mc
.....
</receiver>
</propagation-receivers-config>

必須項 x
目

複数設 O
定

設定値・なし
設定す
る内容

単位・型 なし

省略時のデフォルト値	なし
親タグ	propagation-receivers-config

【属性】

属性名	説明
source	データ送信側が IM-Propagation に渡すクラスの完全なクラス名です。
operationType	データ送信側が IM-Propagation に渡すデータの処理タイプ。

i

コラム

source 属性には IM-Propagation 用のデータが格納されたクラス名を指定します。これにより、データの送信側を意識せず、常に共通的なデータを共有できます。

!

注意

データ送信側設定の sender タグに設定された source と operationType が存在しないクラスや operationType が指定されている場合は、エラーが発生します。

受信側データ変換クラスの設定

タグ名	decoder
-----	---------

IM-Propagation 用のデータを受信側のデータに変換するための、データ変換クラスの設定。

【設定項目】

```
<propagation-receivers-config>
  <receiver source="jp.co.intra_mart.foundation.sample.model.SampleModel">
    <decoder class="jp.co.intra_mart.system.sample.propagation.SamplePropagationDecoder">
    </decoder>
  </receiver>
</propagation-receivers-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし

親タ receiver
グ

【属性】

		デフォルト値	
属性名	説明	必須	ルト値
class	データ変換クラスの完全修飾クラス名です。	☐	なし

 注意

class 属性に指定したクラスが見つからない場合や適切なインタフェースを実装していない場合、データを受信することができませんのでご注意ください。

データ変換クラスパラメータの設定

タ params
グ
名

データ変換クラスに受け渡す任意のパラメータ情報を設定します。

【設定項目】

```
<propagation-receivers-config>  
<receiver source="jp.co.intra_mart.foundation.sample.model.SampleModel">  
  <decoder class="jp.co.intra_mart.system.sample.propagation.SamplePropagationDecoder">  
    <params>  
      .....  
    </params>  
  </decoder>  
</receiver>  
</propagation-receivers-config>
```

必須項目

複数設定

設定値・設定する内容

単位・型

省略のデフォルト値

親タ decoder
グ

データ変換クラスのパラメータ値の設定

タ param
グ
名

【設定項目】

```
<propagation-receivers-config>
<receiver source="jp.co.intra_mart.foundation.sample.model.SampleModel">
  <decoder class="jp.co.intra_mart.system.sample.propagation.SampleModelDecoder">
    <params>
      <param key="key1">value1</param>
      <param key="key2">value2</param>
    </params>
  </decoder>
</receiver>
</propagation-receivers-config>
```

必須項目	x
複数設定	○
設定値・設定する内容	任意のパラメータ値
単位・型	文字列
省略時のデフォルト値	空文字
親タグ	params

【属性】

属性名	説明	必須	デフォルト値
key	パラメータ値を取得するためのキー。	○	なし

受信側データ処理クラスの設定

タグ名	procedure
-----	-----------

データ変換クラスによって変換されたデータを処理するための、データ処理クラス

【設定項目】

```
<propagation-receivers-config>
<receiver source="jp.co.intra_mart.foundation.sample.model.SampleModel">
  <procedure class="jp.co.intra_mart.system.sample.propagation.SampleModelProcedure">
  </procedure>
</receiver>
</propagation-receivers-config>
```

必須項目	○
複数設定	x

設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	receiver

【属性】

属性名	説明	デフォルト値	必須
class	データ処理クラスの完全修飾クラス名です。	○	なし

 注意

class 属性に指定したクラスが見つからない場合や適切なインタ
この場合、データを受信することができませんのでご注意ください

データ処理クラスパラメータの設定

タグ名	params
-----	--------

データ処理クラスに受け渡す任意のパラメータ情報を設定します。

【設定項目】

```
<propagation-receivers-config>
<receiver source="jp.co.intra_mart.foundation.sample.model.SampleMc
<procedure class="jp.co.intra_mart.system.sample.propagation.Sampl
<params>
.....
</params>
</procedure>
</receiver>
</propagation-receivers-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	なし

省略	なし
時の	
デ	
フォ	
ルト	
値	
親タ	procedure
グ	

データ処理クラスのパラメータ値の設定

タ	param
グ	
名	

データ処理クラスに受け渡す任意のパラメータ情報を設定します。

【設定項目】

```
<propagation-receivers-config>
<receiver source="jp.co.intra_mart.foundation.sample.model.SampleModel">
  <procedure class="jp.co.intra_mart.system.sample.propagation.SamplePropagation">
    <params>
      <param key="key1">value1</param>
      <param key="key2">value2</param>
    </params>
  </procedure>
</receiver>
</propagation-receivers-config>
```

必須	×
項目	
複数	○
設定	
設定	任意のパラメータ値
値・設	
定する	
内容	
単位・	文字列
型	
省略	空文字
時の	
デフォ	
ルト値	
親タグ	params

【属性】

属性名	説明	必須	デフォルト値
key	パラメータ値を取得するためのキー。	○	なし

IM-Propagation 送信側設定

- 項目
- 概要
 - リファレンス
 - 送信データ定義の設定
 - 送信側データ変換クラスの設定
 - データ変換クラスパラメータの設定
 - データ変換クラスのパラメータ値の設定
 - 必須呼出データ処理クラスの設定

概要

IM-Propagation を利用してデータを伝搬するための送信側を設定します。

送信するデータを格納するクラスと、データ変換クラスを紐付けるための設定

IM-Propagation を利用してデータを他モジュールへ伝搬するためには、送

データを伝搬する際に他モジュールとの依存関係を切り離す必要があるた
データの送信側は自モジュールで定義されている独自のモデルから、IM-Pr
そこで、データをコピーする作業を行うデータ変換クラスの設定を行います。

モジュール	IM-Propagation
フォーマットファイル (xsd)	WEB-INF/schema/propagation-senders-config (xsd)
設定場所	WEB-INF/conf/propagation-senders-config/{付

 注意

ファイル名は、他のモジュールが提供しているものと重複しないよ

```
<?xml version="1.0" encoding="UTF-8"?>
<propagation-senders-config xmlns="http://www.intra-mart.jp/propagati
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.intra-mart.jp/propagation/senders-con
<sender source="jp.co.intra_mart.foundation.sample.model.SampleMoc
<encoder class="jp.co.intra_mart.system.sample.propagation.SampleM
<params>
  <param key="key1">value1</param>
  <param key="key2">value2</param>
</params>
</encoder>
<required-procedure class="jp.co.intra_mart.system.sample.propagat
</sender>
</propagation-senders-config>
```

リファレンス

送信データ定義の設定

タグ名	sender
-----	--------

IM-Propagation を使用してデータを送信する側が送信するデータの型と如

【設定項目】

```
<propagation-senders-config>
  <sender source="jp.co.intra_mart.foundation.sample.model.SampleMoc
    .....
  </sender>
</propagation-senders-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	propagation-senders-config

【属性】

属性名	説明
source	データ送信側が IM-Propagation に渡すクラスの完全修飾クラス名です。
operationType	データ送信側が IM-Propagation に渡すデータの処理タイプです。

 コラム

operationType 属性には IM-Propagation によりいくつかの候補の中から、送信側で完結する伝搬処理の場合は、独自の operationType を指定する必要があります。

 注意

source 属性に定義されていないクラスが IM-Propagation に渡されるデータの定義は必ず設定ファイルに記述してください。

送信側データ変換クラスの設定

タ
グ
名

送信側のデータを IM-Propagation 用のデータに変換するための、データ変換クラスの設定

【設定項目】

```
<propagation-senders-config>
  <sender source="jp.co.intra_mart.foundation.sample.model.SampleModel" >
    <encoder class="jp.co.intra_mart.system.sample.propagation.SamplePropagationEncoder" />
  </sender>
</propagation-senders-config>
```

必須項目	☐
複数設定	☒
設定値・設定内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	sender

【属性】

属性名	説明	デフォルト値
class	データ変換クラスの完全修飾クラス名です。	☐ なし



注意

class 属性に指定したクラスが見つからない場合や適切なインタ

データ変換クラスパラメータの設定

タ params
グ
名

データ変換クラスに受け渡す任意のパラメータ情報を設定します。

【設定項目】

```
<propagation-senders-config>
  <sender source="jp.co.intra_mart.foundation.sample.model.SampleMoc
    <encoder class="jp.co.intra_mart.system.sample.propagation.Sample
      <params>
        .....
      </params>
    </encoder>
  </sender>
</propagation-senders-config>
```

必 x
須
項
目

複 x
数
設
定

設 なし
定
値・
設
定す
る内
容

単 なし
位・
型

省 なし
略
時
のデ
フォ
ルト
値

親タ encoder
グ

データ変換クラスのパラメータ値の設定

タ param
グ
名

データ変換クラスに受け渡す任意のパラメータ情報を設定します。

【設定項目】

```
<propagation-senders-config>
<sender source="jp.co.intra_mart.foundation.sample.model.SampleMoc
<encoder class="jp.co.intra_mart.system.sample.propagation.Samplel
  <params>
    <param key="key1">value1</param>
    <param key="key2">value2</param>
  </params>
</encoder>
</sender>
</propagation-senders-config>
```

必須項目	x
複数設定	○
設定値・設定する内容	任意のパラメータ値
単位・型	文字列
省略時のデフォルト値	空文字
親タグ	params

【属性】		必須	デフォルト値
属性名	説明		
key	パラメータ値を取得するためのキー。	○	なし

必須呼出データ処理クラスの設定

タグ名	required-procedure
-----	--------------------

IM-Propagation でデータを必ず受け取って欲しい受信側のデータ処理クラ

【設定項目】

```
<propagation-senders-config>
<sender source="jp.co.intra_mart.foundation.sample.model.SampleMoc
  <required-procedure class="jp.co.intra_mart.system.sample.propagat
</sender>
</propagation-senders-config>
```

必須項目	x
複数設定	○
設定値・設定する内容	なし

単
位・
型

省
略
時
の
デ
フォ
ルト
値

親タ
グ sender

【属性】

属性名	説明
-----	----

class	データ受信側のデータ処理クラスの完全修飾クラス名です。
--------------	-----------------------------



注意

class 属性に指定したクラスに対してデータが送信されなかった

テナント管理機能

システム管理者用スクリプト開発モデルルーティ

項目

- [概要](#)
- [リファレンス](#)
 - [ファイルマッピング設定](#)
 - [フォルダマッピング設定](#)

概要

システム管理者用のURLに対して、スクリプト開発モデルの処理のマッピン

モジュール	テナント管理機能
-------	----------

フォーマットファイル	WEB-INF/schema/admin-routing-jssp-config.xsd
------------	--

設定場所	WEB-INF/conf/admin-routing-jssp-config/{任意}
------	---

```
<?xml version="1.0" encoding="UTF-8"?>
<admin-routing-jssp-config
  xmlns="http://www.intra-mart.jp/router/admin/admin-routing-jssp-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/router/admin/admin-routing-jssp-config http://www.intra-mart.jp/router/admin/admin-routing-jssp-config.xsd">
  <file-mapping path="/system/application/main" page="system/application/main"></file-mapping>
  <file-mapping path="/system/application/list" page="system/application/list"></file-mapping>
  <file-mapping path="/system/application/process/create" page="system/application/process/create"></file-mapping>
  <file-mapping path="/system/application/process/update" page="system/application/process/update"></file-mapping>
  <file-mapping path="/system/application/process/delete" page="system/application/process/delete"></file-mapping>
</admin-routing-jssp-config>
```

リファレンス

[ファイルマッピング設定](#)

タグ
file-mapping
名

URLとスクリプト開発モデルのプログラムのマッピングを行います。

【設定項目】

```
<admin-routing-jssp-config>
  <file-mapping path="/system/application/main" page="system/applicati
  <file-mapping path="/system/application/list" page="system/application/
</admin-routing-jssp-config>
```

必須項目	x
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	admin-routing-jssp-config

【属性】

属性名	説明
path	マッピングを行うURLを指定します。 値の末尾に ワイルドカード 「*」 を指定すること 値に {<識別子>} を記述することでURLの途中 のプログラム中で使用可能です。 例: /sample/view/{dataId}
page	マッピングを行うスクリプト開発モデルのプログラ
action	page属性に指定されたプログラムの実行前に呼 from属性を指定した場合はfrom属性に指定さ from属性が未指定の場合はpage属性に指定さ ます。
from	action属性で指定した関数を呼び出すプログラ

 コラム

page, from 属性で指定するスクリプト開発モデルのプログラムは
スクリプト開発モデルプログラムのソースディレクトリからの相対
デフォルトではWEB-INF/jssp/srcからの相対パス形式になります

フォルダマッピング設定

タグ
folder-mapping
名

URLのパターンとスクリプト開発モデルのプログラムのマッピングを行います。
特定のスクリプト開発モデルのプログラムのディレクトリ配下すべてのプロ

【設定項目】

```
<admin-routing-jssp-config>
  <folder-mapping path-prefix="/system/application" folder="system/appli
</admin-routing-jssp-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	admin-routing-jssp-config

【属性】

属性名	説明
path-prefix	マッピングを行うURLを指定します。
folder	マッピングを行うスクリプト開発モデルのプログラムのデ

 **注意**

この設定を使用するとルーティングに時間が掛かる、スクリプト開
れません。
使用するスクリプト開発モデルのプログラムに対してfile-mappin

システム管理者用IM-JavaEE Frameworkルー...

項目
● 概要
● リファレンス
○ サービスマッピング設定
○ アプリケーションマッピング設定

概要

システム管理者用のURLに対して、IM-JavaEE Frameworkの処理のマッピ

モジュール	テナント管理機能
フォーマットファイル (xsd)	WEB-INF/schema/admin-routing-service-conf
設定場所	WEB-INF/conf/admin-routing-service-config/{

```
<?xml version="1.0" encoding="UTF-8"?>
<admin-routing-service-config
  xmlns="http://www.intra-mart.jp/router/admin-routing-service-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/router/admin-routing-serv
  <service-mapping path="/system/application/main" application="applic
</admin-routing-service-config>
```

リファレンス

サービスマッピング設定

タグ名	service-mapping
-----	-----------------

【設定項目】

```
<admin-routing-service-config>
  <service-mapping path="/system/application/main" application="application" />
</admin-routing-service-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	admin-routing-service-config

【属性】

属性名	説明
path	マッピングを行うURLを指定します。 値の末尾に ワイルドカード 「*」 を指定することが可能。 値に {識別子} を記述することでURLの途中の HttpServletRequest#getAttribute() で取得可能で 例: /sample/view/{dataId}
application	マッピングを行うIM-JavaEE Frameworkのアプリケーション名を指定します。
service	マッピングを行うIM-JavaEE Frameworkのサービス名を指定します。

アプリケーションマッピング設定

タグ名	application-mapping; admin-routing-service-config
-----	---

URLのパターンとIM-JavaEE Frameworkのアプリケーションのマッピングを特定する。特定のIM-JavaEE Frameworkのアプリケーションを持つすべてのサービス

【設定項目】

```
<admin-routing-service-config>
  <application-mapping path-prefix="/system/application" application="application" />
</admin-routing-service-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	admin-routing-service-config

【属性】

属性名	説明
path-prefix	マッピングを行うURLを指定します。
application	マッピングを行うIM-JavaEE Frameworkのプログラム指定します。

 注意

この設定を使用するとルーティングに時間が掛かる、IM-JavaEE
れません。
使用するIM-JavaEE Frameworkのサービスに対してservice-m

システム管理者用サーブレットルーティング設定

項目
- 概要 - リファレンス - サーブレットマッピング設定

概要

システム管理者用のURLに対して、フォワード先のサーブレットパスへのマッ

モジュール	テナント管理機能
フォーマットファイル (xsd)	WEB-INF/schema/admin-routing-servlet-confi
設定場所	WEB-INF/conf/admin-routing-servlet-config/{f

```
<?xml version="1.0" encoding="UTF-8"?>
<admin-routing-servlet-config
  xmlns="http://www.intra-mart.jp/router/admin-routing-servlet-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/router/admin-routing-serv
  <servlet-mapping path="/system/application/main" servlet="application/
</admin-routing-servlet-config>
```

リファレンス

サーブレットマッピング設定

タ グ 名	servlet-mapping
-------------	-----------------

URLとサーブレットパスのマッピングを行います。

【設定項目】

```
<admin-routing-servlet-config>
  <servlet-mapping path="/user/application/main" servlet="application/m
</admin-routing-servlet-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし

省略時のデフォルト値	なし
親タグ	admin-routing-servlet-config

【属性】

属性名	説明
path	マッピングを行うURLを指定します。 値の末尾にワイルドカード「*」を指定することが可能で、値に「{識別子}」を記述することでURLの途中の「{識別子}」をHttpServletRequest#getAttribute()で取得可能です。 例: /sample/view/{dataId}
servlet	マッピングを行うサーブレットパスを指定します。

認可判断モジュール設定

- 項目
- 概要
 - リファレンス
 - 認可判断に使用する認可判断モジュールの構成
 - 使用する実装クラスの設定

概要

認可判断に使用する認可判断モジュールの構成と、結果を取りまとめる方法について、この設定による動作の詳細については認可仕様書も合わせて参照してください。

モジュール	テナント管理機能
フォーマットファイル(xsd)	WEB-INF/schema/authz-decision-config.xsd
設定場所	WEB-INF/conf/authz-decision-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<authz-decision-config
  xmlns="http://www.intra-mart.jp/authz/authz-decision-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/authz/authz-decision-config http://www.intra-mart.jp/authz/authz-decision-config.xsd">
  <decision-config name="default" combinator="permit-overrides">
    <module class="jp.co.intra_mart.foundation.authz.services.decision.impl.DefaultDecisionModule">
    </module>
    <module class="jp.co.intra_mart.foundation.authz.services.decision.impl.DefaultDecisionModule">
    </module>
    <module class="jp.co.intra_mart.foundation.authz.services.decision.impl.DefaultDecisionModule">
    </module>
  </decision-config>
</authz-decision-config>
```

リファレンス

認可判断に使用する認可判断モジュールの構成

タグ名	decision-config
-----	-----------------

子要素に認可判断に使用するモジュールを <module> タグとして列挙します。認可判断に使用するモジュールの列挙と、複数のモジュールの下した判断の優先順位は、<decision-config> タグの combinator 属性で指定します。

【設定項目】

```
<authz-decision-config
  xmlns="http://www.intra-mart.jp/authz/authz-decision-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/authz/authz-decision-config http://www.intra-mart.jp/authz/authz-decision-config.xsd">
  <decision-config name="default" combinator="permit-overrides">
    ....
  </decision-config>
</authz-decision-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	authz-decision-config

【属性】

属性名	説明
name	この設定の名前です。将来の拡張のために：
combinator	子要素に指定する情報 <module> タグに指定する情報 (NotApplicable)。その結果をどのように取り扱うか。設定できる値は以下の3種類です。 - permit-overrides - 上から順に評価して最初に Permit - deny-overrides - 上から順に評価して最初に Deny - first-applicable - 上から順に評価して最初に Permit

使用する実装クラスの設定

タグ名	module
-----	--------

使用する認可判断モジュールのクラス名を設定します。認可判断モジュール

 注意

認可判断機能は<module>タグの記述された順番にモジュールが

【設定項目】

```
<authz-decision-config
  xmlns="http://www.intra-mart.jp/authz/authz-decision-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/authz/authz-decision-conf
  <decision-config name="default" combinator="permit-overrides">
    <module class="jp.co.intra_mart.foundation.authz.services.decision.i
  </decision-config>
</authz-decision-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	なし

単位・型	なし
省略時のデフォルト値	なし
親タグ	decision-config

【属性】

属性名	説明
class	認可判断モジュールとして使用するクラスの完全修飾名を指定します。

認可設定画面設定

項目
- 概要 - リファレンス - 認可ポリシー更新処理設定 - キャッシュ更新エラー設定 (認可ポリシー) - 認可リソースグループ更新処理設定 - キャッシュ更新エラー設定 (認可リソースグループ) - リソースグループバックアップ設定 - バックアップ先パス設定 - バックアップ最大件数設定 - Excelインポート設定 - Excelインポート処理結果配置パス設定 - Excelインポートコミット間隔設定 - 未設定認可ポリシー検証設定 - リソースグループ・リソース存在検証設定 - サブジェクトグループ存在検証設定 - Excelインポート処理の処理結果ファイル数設定 - Excelエクスポート設定 - テンプレートファイル設定 - Excelエクスポートファイル配置パス設定 - Excelエクスポートファイル名設定 - Excelエクスポートファイル拡張子設定 - Excelエクスポート保存ファイル数設定 - 保護パスワード設定

概要

認可設定画面の設定を行います。

モジュール	テナント管理機能
フォーマットファイル(xsd)	WEB-INF/schema/authz-editor-config.xsd
設定場所	WEB-INF/conf/authz-editor-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<authz-editor-config
  xmlns="http://www.intra-mart.jp/authz/authz-editor-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/authz/authz-editor-config
  <policy-update>
    <cache-update-error rollback="true" />
  </policy-update>
  <resource-group-update>
    <cache-update-error rollback="true" />
  </resource-group-update>
  <resource-group-backup>
    <path>im_authz/backup</path>
    <max-size>10</max-size>
  </resource-group-backup>
  <xlsx-import>
    <path>im_authz/xlsx_import</path>
    <commit-count>0</commit-count>
    <validate-inherits>false</validate-inherits>
    <validate-resource-exists>true</validate-resource-exists>
    <validate-subject-exists>false</validate-subject-exists>
    <max-size>10</max-size>
  </xlsx-import>
  <xlsx-export>
    <template-file>products/im_authz_xlsx_import_export/authz_template
    <path>im_authz/xlsx_export</path>
    <file-name>yyyyMMdd_HHmms</file-name>
    <file-ext>.xlsx</file-ext>
    <max-size>10</max-size>
    <xlsx-password>intramart</xlsx-password>
  </xlsx-export>
</authz-editor-config>
```

リファレンス

認可ポリシー更新処理設定

タグ policy-update
属性
名前

認可ポリシー更新時の処理の設定を行います。

【設定項目】

```
<authz-editor-config>
  <policy-update>
    ...
  </policy-update>
</authz-editor-config>
```

必須項目	x
複数設定	x
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	authz-editor-config

キャッシュ更新エラー設定 (認可ポリシー)

タグ名
cache-update-error

認可ポリシーのキャッシュ更新によるエラー時の挙動について設定します。

【設定項目】

```
<authz-editor-config>
  <policy-update>
    <cache-update-error rollback="true" />
  </policy-update>
</authz-editor-config>
```

必須項目	x
複数設定	x
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	policy-update

【属性】

属性名	説明
rollback	キャッシュ更新失敗し認可ポリシーの更新をロールバック設定します。

認可リソースグループ更新処理設定

タグ名
resource-group-update

認可リソースグループ更新時の処理の設定を行います。

【設定項目】

```
<authz-editor-config>
  <resource-group-update>
    ...
  </resource-group-update>
</authz-editor-config>
```

必須項目	x
複数設定	x
設定値・設定する内容	なし
単位・型	なし

省略時のデフォルト値	なし
親タグ	authz-editor-config

キャッシュ更新エラー設定(認可リソースグループ)

タグ名	cache-update-error
-----	--------------------

認可リソースグループのキャッシュ更新によるエラー時の挙動について設定

【設定項目】

```
<authz-editor-config>
  <resource-group-update>
    <cache-update-error rollback="true" />
  </resource-group-update>
</authz-editor-config>
```

必須項目	x
複数設定	x
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	resource-group-update

【属性】

属性名	説明
rollback	キャッシュ更新失敗し認可リソースグループの更新をロールバックを設定します。

リソースグループバックアップ設定

タグ名	resource-group-backup
-----	-----------------------

リソースグループと、紐づくリソースを削除した場合に、関係する認可設定を

【設定項目】

```
<authz-editor-config>
  <resource-group-backup>
    ...
  </resource-group-backup>
</authz-editor-config>
```

必須項目	x
複数設定	x

設定	なし
値・設定する内容	
単位・型	なし
省略時のデフォルト値	なし
親タグ	authz-editor-config

バックアップ先パス設定

タグ名	path
-----	------

バックアップした認可設定のファイルを保存する場所について設定します。
パブリックストレージのルートからの相対パスです。

【設定項目】

```
<authz-editor-config>
  <resource-group-backup>
    <path>im_authz/backup</path>
  </resource-group-backup>
</authz-editor-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	パブリックストレージのルートからの相対パスを指定します
単位・型	文字列
省略時のデフォルト値	im_authz/backup
親タグ	resource-group-backup

バックアップ最大件数設定

タグ名	max-size
-----	----------

バックアップとして認可設定のファイルを保存する最大件数について設定します。
0を指定した場合、保存件数は無制限になります。この場合、パブリックスト

【設定項目】

```
<authz-editor-config>
  <resource-group-backup>
    <max-size>30</max-size>
  </resource-group-backup>
</authz-editor-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	バックアップを残す最大件数を指定します。
単位・型	数値

省略時の デフォルト 値	30
親タグ	resource-group-backup

Excelインポート設定

タグ 名	xlsx-import
---------	-------------

認可設定画面上からExcelファイルを用いて認可設定のインポートを行う際

【設定項目】

```
<authz-editor-config>
  <xlsx-import>
  ...
</xlsx-import>
</authz-editor-config>
```

必須 項目	x
複数 設定	x
設定 値・設 定す る内 容	なし
単位・ 型	なし
省略 時の デフォ ルト値	なし
親タグ	authz-editor-config

Excelインポート処理結果配置パス設定

タグ 名	path
---------	------

Excelインポート処理の処理結果が保存されているJSON形式のファイルをi
ファイルはパブリックストレージに配置されるため、パブリックストレージのル
認可設定画面にてExcelインポートを行う際にアップロードを行ったExcelフ

【設定項目】

```
<authz-editor-config>
  <xlsx-import>
    <path>im_authz/xlsx_import</path>
  </xlsx-import>
</authz-editor-config>
```

必須項目	x
複数設定	x
設定値・設定する内容	Excelインポート処理結果ファイルを格納する場 す。
単位・型	文字列
省略時のデフォルト値	im_authz/xlsx_import
親タグ	xlsx-import

タグ

commit-count

名

インポート処理で、認可ポリシー変更（追加・更新・削除を指します）を行った例えば、10 を指定した場合は 10 件の認可ポリシーを変更する度にデータ0 以下を指定した場合、インポート処理が完了するまでコミットが行われませんインポート処理にてコミットを行う順序については、[「IM-Authz（認可）」インポ](#)

【設定項目】

```
<authz-editor-config>
  <xlsx-import>
    <commit-count>0</commit-count>
  </xlsx-import>
</authz-editor-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	コミットを行うまでのポリシー変更件数
単位・型	数値
省略時のデフォルト値	0
親タグ	xlsx-import

未設定認可ポリシー検証設定

タグ

validate-inherits

名

インポートファイルにて明示的に「許可」／「禁止」を設定されていない認可検証は、インポートファイルの編集者のミスなどにより意図しない認可ポリシー検証を行う場合、「許可」／「禁止」以外が設定されている認可ポリシーのセ

- セルの内容が「未設定（継承した結果、許可）」であり、セルの上位リソ
- セルの内容が「未設定（継承した結果、禁止）」であり、セルの上位リソ

未設定認可ポリシー検証設定によるインポート時の挙動の違いについての

【設定項目】

```
<authz-editor-config>
  <xlsx-import>
    <validate-inherits>false</validate-inherits>
  </xlsx-import>
</authz-editor-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	未設定である認可ポリシーセルの検証を行うかどうか
単位・型	真偽値 (true/false)
省略時のデフォルト値	false
親タグ	xlsx-import

リソースグループ・リソース存在検証設定

タグ

validate-resource-exists

名

インポートするExcel (xlsx) ファイルに定義されたリソースグループ・リソース検証を行う場合、存在しないリソースグループ・リソースが定義されていた場

【設定項目】

```
<authz-editor-config>
<xlsx-import>
  <validate-resource-exists>true</validate-resource-exists>
</xlsx-import>
</authz-editor-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	リソースグループ・リソース存在検証を行うかどうかを記述する内容
単位・型	真偽値 (true/false)
省略時のデフォルト値	true
親タグ	xlsx-import

サブジェクトグループ存在検証設定

タグ名 validate-subject-exists

インポートするExcel (xlsx) ファイルに定義されたサブジェクトグループに関存在しないサブジェクトグループが定義されていた場合、エラー扱いとして

【設定項目】

```
<authz-editor-config>
<xlsx-import>
  <validate-subject-exists>false</validate-subject-exists>
</xlsx-import>
</authz-editor-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	サブジェクトグループを行うかどうかを設定します。
単位・型	真偽値 (true/false)
省略時のデフォルト値	false
親タグ	xlsx-import

Excelインポート処理の処理結果ファイル数設定

タグ名 max-size

Excelインポート処理の処理結果ファイルを保存するファイル数を設定します
Excelインポート処理の処理結果ファイルの数が設定値を超えた場合、古い認可設定画面のインポート結果一覧に表示される結果数はこの設定で指定した0以下を指定した場合、保存件数は無制限になります。この場合、パブリック

【設定項目】

```
<authz-editor-config>
<xlsx-import>
  <max-size>10</max-size>
</xlsx-import>
</authz-editor-config>
```

必須項目	x
複数設定	x
設定値・設定する内容	Excelインポート処理の処理結果のファイル数
単位・型	数値
省略時のデフォルト値	10
親タグ	xlsx-import

Excelエクスポート設定

タグ名	xlsx-export
-----	-------------

認可設定画面上からExcelファイルを用いて認可設定のエクスポートを行う

【設定項目】

```
<authz-editor-config>
<xlsx-export>
...
</xlsx-export>
</authz-editor-config>
```

必須項目	x
複数設定	x
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	authz-editor-config

テンプレートファイル設定

タグ名	template-file
-----	---------------

エクスポートするExcel (xlsx) ファイルを作成するための、テンプレートファイル

【設定項目】

```
<authz-editor-config>
<xlsx-export>
  <template-file>products/im_authz_xlsx_import_export/authz_template
</xlsx-export>
</authz-editor-config>
```

必須項目	x
複数設定	x
設定値・設定する内容	エクスポートするExcel (xlsx) ファイルを付します。

単位・型	文字列
省略時のデフォルト値	products/im_authz_xlsx_import_expor
親タグ	xlsx-export

Excelエクスポートファイル配置パス設定

タグ名

Excelエクスポートを行い出力されるファイルを配置する場所を設定します。
ファイルはパブリックストレージに配置されるため、パブリックストレージのル

【設定項目】

```
<authz-editor-config>
  <xlsx-export>
    <path>im_authz/xlsx_export</path>
  </xlsx-export>
</authz-editor-config>
```

必須項目	x
複数設定	x
設定値・設定する内容	Excelエクスポートされたファイルを格納する場 す。
単位・型	文字列
省略時のデフォルト値	im_auhtz/xlsx
親タグ	xlsx-export

Excelエクスポートファイル名設定

タグ名

Excelエクスポートされたファイルのファイル名を設定します。
ファイル名には、日付と時刻のフォーマットパターンを利用することが可能で
JavaDocを参照してください。

【設定項目】

```
<authz-editor-config>
  <xlsx-export>
    <file-name>yyyyMMdd_HHmms</file-name>
  </xlsx-export>
</authz-editor-config>
```

必須項目	x
複数設定	x
設定値・設定する内容	エクスポートするExcelファイルのファイル名を設定します 内容
単位・型	文字列
省略時のデフォルト値	yyyyMMdd_HHmms
親タグ	xlsx-export

Excelエクスポートファイル拡張子設定

タグ名

Excelエクスポートされたファイルの拡張子を設定します。
設定値に置換文字列が利用可能です。利用可能な置換文字列は以下の通

- {^userCd^} : 出力操作を行ったユーザコード
- {^partId^} : 出力した part-id

【設定項目】

```
<authz-editor-config>
<xlsx-export>
  <file-ext>.xlsx</file-ext>
</xlsx-export>
</authz-editor-config>
```

必須項目	x
複数設定	x
設定値・設定する内容	エクスポートするExcelファイルの拡張子を設定します。
単位・型	文字列
省略時のデフォルト値	.xlsx
親タグ	xlsx-export

Excelエクスポート保存ファイル数設定

タグ名

Excelエクスポートを行い出力されるファイルを保存するファイル数を設定し、Excelエクスポートを行い出力されるファイルの数が設定値を超えた場合、承認設定画面のエクスポート済みファイル一覧に表示される結果数はこの値以下を指定した場合、保存件数は無制限になります。この場合、パブリック

【設定項目】

```
<authz-editor-config>
<xlsx-export>
  <max-size>10</max-size>
</xlsx-export>
</authz-editor-config>
```

必須項目	x
複数設定	x
設定値・設定する内容	Excelエクスポートを行い出力されるファイルの数が
単位・型	数値
省略時のデフォルト値	10
親タグ	xlsx-export

保護パスワード設定

タグ名

エクスポートするExcel (xlsx) ファイルの変更不可部分を保護するパスワードを指定しない場合、保護パスワードはかからず行・列の追加・更新・削除が可

【設定項目】

```
<authz-editor-config>
<xlsx-export>
  <xlsx-password>intramart</xlsx-password>
</xlsx-export>
</authz-editor-config>
```

必須項目	x
複数設定	x
設定値・設定する内容	保護するパスワードの値を設定します。
単位・型	文字列
省略時のデフォルト値	intramart
親タグ	xlsx-export

ポリシー部分編集定義設定

項目
- 概要 - リファレンス - 認可設定画面の部品化設定 - 設定ID - 設定のキャプション - 対象リソースグループ - リソースグループ - 対象サブジェクトタイプ - サブジェクトタイプ - コールバック設定 - リソースグループ権限設定可否判断クラス

概要

認可設定画面を部品として使用する際に必要になる設定を記述します。
この設定を使用する前に、この設定に記載するリソースグループセットが予め

モジュール	テナント管理機能
フォーマットファイル (xsd)	WEB-INF/schema/authz-partial-policy-edit-cc (xsd)
設定場所	WEB-INF/conf/authz-partial-policy-edit-config

```
<?xml version="1.0" encoding="UTF-8"?>
<authz-partial-policy-edit-config xmlns="http://www.intra-mart.jp/authz/">
  <part-config>
    <part-id>im_authz_impl_router</part-id>
    <caption-cd>CAP.Z.IWP.ROUTER.AUTHZ.PARTCONFIG.TITLE</caption-cd>
    <resource-groups>
      <resource-group-id>http-services</resource-group-id>
    </resource-groups>
    <subject-types>
      <subject-type-id>im_authz_meta_subject</subject-type-id>
      <subject-type-id>imm_user</subject-type-id>
      <subject-type-id>imm_company_post</subject-type-id>
      <subject-type-id>imm_department</subject-type-id>
      <subject-type-id>imm_public_grp</subject-type-id>
      <subject-type-id>imm_public_grp_role</subject-type-id>
      <subject-type-id>b_m_role</subject-type-id>
    </subject-types>
    <callbacks>
      <resource-group-authorizer>jp.co.intra_mart.system.authz.ResourceGroupAuthorizer</resource-group-authorizer>
    </callbacks>
  </part-config>
</authz-partial-policy-edit-config>
```

リファレンス

認可設定画面の部品化設定

タグ名

認可設定画面を部品として使用するための設定項目のひとつとまりです。

【設定項目】

```
<authz-partial-policy-edit-config xmlns="http://www.intra-mart.jp/authz/;
<part-config>
...
</part-config>
</authz-partial-policy-edit-config>
```

必須項目	x
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	authz-partial-policy-edit-config

設定ID

タグ名

設定のIDを指定します。この値は部品化された認可設定画面を使用するブ
先頭にモジュールのIDを使用するなどし、他の設定と重複しないようIDを付

【設定項目】

```
<authz-partial-policy-edit-config xmlns="http://www.intra-mart.jp/authz/;
<part-config>
  <part-id>im_authz_impl_router</part-id>
</part-config>
</authz-partial-policy-edit-config>
```

必須項目	○
複数設定	x
設定値・設定する内容	設定のIDを指定します。他の設定と重複しないよう
単位・型	文字列 (半角英数、ハイフン、アンダースコア)
省略時のデフォルト値	なし
親タグ	part-config

設定のキャプション

タグ名

この設定の表示名を取得するためのメッセージコードを設定します。このコー

【設定項目】

```
<authz-partial-policy-edit-config xmlns="http://www.intra-mart.jp/authz/;
<part-config>
  <caption-cd>CAP.Z.IWP.ROUTER.AUTHZ.PARTCONFIG.TITLE</ca
</part-config>
</authz-partial-policy-edit-config>
```

必須項目	x
複数設定	x
設定値・設定する内容	キャプションのメッセージコード
単位・型	文字列
省略時のデフォルト値	なし
親タグ	part-config

対象リソースグループ

タグ名	resource-groups
-----	-----------------

この設定で使用するリソースグループを指定します。

認可設定画面ではここで指定されたリソースグループ配下を表示するため、場合、それらのグループ配下がマージして表示されます。複数のリソースグループ

【設定項目】

```
<authz-partial-policy-edit-config xmlns="http://www.intra-mart.jp/authz/;
<part-config>
  <resource-groups>
    ...
  </resource-groups>
</part-config>
</authz-partial-policy-edit-config>
```

必須項目	○
複数設定	x
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	part-config

リソースグループ

タグ resource-group-id
グループ名

リソースグループIDを指定します。リソースグループIDは動的に作成されるIDです。

【設定項目】

```
<authz-partial-policy-edit-config xmlns="http://www.intra-mart.jp/authz/policy">
  <part-config>
    <resource-groups>
      <resource-group-id>http-services</resource-group-id>
    </resource-groups>
  </part-config>
</authz-partial-policy-edit-config>
```

必須項目	☐
複数設定	☐
設定値・設定する内容	使用するリソースグループID
単位・型	文字列
省略時のデフォルト値	なし
親タグ	resource-groups

対象サブジェクトタイプ

タグ subject-types
グループ名

この設定で使用するサブジェクトタイプを指定します。

ここで指定していないサブジェクトタイプは、この設定を使用して開いた認可トタイプのIDでなければなりません。

【設定項目】

```
<?xml version="1.0" encoding="UTF-8"?>
<authz-partial-policy-edit-config xmlns="http://www.intra-mart.jp/authz/policy">
  <part-config>
    <subject-types>
      ...
    </subject-types>
  </part-config>
</authz-partial-policy-edit-config>
```

必須項目	☐
複数設定	☒
設定値・設定する内容	なし
単位・型	なし

省略	なし
時の	
デ	
フォ	
ルト	
値	
親タ	part-config
グ	

サブジェクトタイプ

タ	subject-type-id
グ	
名	

使用するサブジェクトタイプのIDを設定します。

【設定項目】

```
<authz-partial-policy-edit-config xmlns="http://www.intra-mart.jp/authz/">
  <part-config>
    <subject-types>
      <subject-type-id>b_m_role</subject-type-id>
    </subject-types>
  </part-config>
</authz-partial-policy-edit-config>
```

必須項目	☐
複数設定	☐
設定値・ 設定する 内容	この設定で使用するサブジェクトタイプID
単位・型	文字列
省略時の デフォルト 値	なし
親タグ	subject-types

コールバック設定

タ	callbacks
グ	
名	

この設定によって認可設定画面を部品として使用した際に利用されるコール

バック設定画面を部品として使用する場合は呼び出し元が認可設定画面に

【設定項目】

```
<authz-partial-policy-edit-config xmlns="http://www.intra-mart.jp/authz/">
  <part-config>
    <callbacks>
      ...
    </callbacks>
  </part-config>
</authz-partial-policy-edit-config>
```

必須	x
項目	
複数	x
設定	
設定	なし
値・ 設定 する 内容	

単位・型	なし
省略時のデフォルト値	なし
親タグ	part-config

リソースグループ権限設定可否判断クラス

タグ名	resource-group-authorizer
-----	---------------------------

この設定によって認可設定画面を部品として使用した際に利用されるリソース `jp.co.intra_mart.foundation.authz.partial.AuthzPartialResourceGroupA`

認可設定画面を部品として使用する場合は呼び出し元が認可設定画面に指定された場合に、認証済みユーザの権限を超えて権限設定ができないよ

この設定は省略可能です。省略した場合はデフォルトの判断クラスが使用

【設定項目】

```
<authz-partial-policy-edit-config xmlns="http://www.intra-mart.jp/authz/...>
  <part-config>
    <callbacks>
      <resource-group-authorizer>jp.co.intra_mart.system.authz.ResourceGroupAuthorizer</resource-group-authorizer>
    </callbacks>
  </part-config>
</authz-partial-policy-edit-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	jp.co.intra_mart.foundation.authz.partial.AuthzPartialResourceGroupAの名称。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	callbacks

認可設定画面 保護リソース設定

- 項目
- 概要
 - リファレンス
 - 保護するリソースとアクションの指定

概要

この設定は、認可設定画面上での変更を保護するリソースの一覧です。

操作中のユーザが、この設定に記載されているリソース・アクションについて、無権限でこれらの権限すべてを失わせる操作を行った場合、エラーとして

これは認可設定の変更中に自身で操作の続行ができなくなるような変更を

モジュール	テナント管理
フォーマットファイル(xsd)	WEB-INF/schema/authz-protected-resource-configuration.xsd
設定場所	WEB-INF/conf/authz-protected-resource-configuration.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<authz-protected-resource-config
  xmlns="http://www.intra-mart.jp/authz/ui/protected-resources"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/authz/ui/protected-resourc

  <protected-resource uri="service://authz/settings/basic" action="execut
  <protected-resource uri="service://authz/settings/parts" action="execut
  <protected-resource uri="service://authz/settings/procedure" action="e:

</authz-protected-resource-config>
```

リファレンス

保護するリソースとアクションの指定

タグ名
protected-resource

保護するリソースとアクションを指定します。

【設定項目】

```
<authz-protected-resource-config
  xmlns="http://www.intra-mart.jp/authz/ui/protected-resources"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/authz/ui/protected-resourc

  <protected-resource uri="service://authz/settings/basic" action="execut

</authz-protected-resource-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	authz-protected-resource-config

【属性】

属性名	説明
uri	保護する対象のリソースのURIを指定します。データが必要があります。
action	uriに対して保護するアクションを指定します。

 注意

この設定は通常変更の必要はありません。

ルーティングテーブル用 認可リソースマッパー設定

- 項目
- 概要
 - リファレンス
 - リソースマッパー設定

ルーティングテーブルの認可設定で使用するマッパーの登録を行う設定です。

ルータにはそれぞれのパスに対してマッピングされるプログラムの実体と、認可リソースの紐づけに関して、パスに対して特定のリソースを指定するかどうか登録されたものから選択できるようになっています。

モジュール	テナント管理機能
フォーマットファイル (xsd)	WEB-INF/schema/authz-resource-mappers.xsd
設定場所	WEB-INF/conf/authz-resource-mappers/{任意のファイル名}.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<authz-resource-mappers xmlns="http://www.intra-mart.jp/router/authz-resource-mappers">
  <mapper name="dev-auto-register" class="jp.co.intra_mart.system.router.authz.user.AuthzResourceMapper">
    <mapper name="welcome-all" class="jp.co.intra_mart.system.router.authz.user.AuthzResourceMapper">
    </mapper>
  </authz-resource-mappers>
```

リファレンス

リソースマッパー設定

タグ名

マッパーの名前と実際にマッピングを行う実装クラスを設定します。

マッパーの実装は `jp.co.intra_mart.system.router.authz.user.AuthzResourceMapper` の実装クラスを指定します。

【設定項目】

```
<authz-resource-mappers xmlns="http://www.intra-mart.jp/router/authz-resource-mappers">
  <mapper name="welcome-all" class="jp.co.intra_mart.system.router.authz.user.AuthzResourceMapper">
  </authz-resource-mappers>
```

必須項目

複数設定

設定値・設定する内容

単位・型

省略時のデフォルト値

親タグauthz-resource-mappers

【属性】

属性名	説明
name	マッパーの名前です。ルーティングテーブルの<authz-resource-mappers>要素の<mapper>要素のname属性に指定した値です。
class	この名前に割り当てるマッパーの実装クラスの完全なクラス名を指定します。

リソースタイプ拡張設定

項目
概要

- リファレンス
 - リソースタイプ設定

概要

認可機構にリソースタイプを追加するための設定です。
リソースタイプと、リソースタイプの取り扱うリソースモデルを定義します。
リソースタイプの詳細については認可仕様書を参照してください。

モジュール	テナント管理機能
フォーマットファイル (xsd)	WEB-INF/schema/authz-resource-type-config
設定場所	WEB-INF/conf/authz-resource-type-config/{任

```
<?xml version="1.0" encoding="UTF-8"?>
<authz-resource-type-config xmlns="http://www.intra-mart.jp/authz/authz-resource-type-config"
  <resource-type
    type-class="jp.co.intra_mart.system.router.authz.resourcetype.GeneralResourceType"
    model-class="jp.co.intra_mart.system.router.authz.resourcetype.GeneralResourceModel"
    cache-class="jp.co.intra_mart.system.router.authz.resourcetype.GeneralResourceCache"
  </resource-type>
</authz-resource-type-config>
```

リファレンス

リソースタイプ設定

タグ名

resource-type

リソースタイプの実装クラスと、この実装クラスが扱うリソースモデルクラスを指定します。

【設定項目】

```
<authz-resource-type-config xmlns="http://www.intra-mart.jp/authz/authz-resource-type-config"
  <resource-type
    type-class="jp.co.intra_mart.system.router.authz.resourcetype.GeneralResourceType"
    model-class="jp.co.intra_mart.system.router.authz.resourcetype.GeneralResourceModel"
    cache-class="jp.co.intra_mart.system.router.authz.resourcetype.GeneralResourceCache"
  </resource-type>
</authz-resource-type-config>
```

必須項目	☐
複数設定	☐
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	authz-resource-type-config

【属性】

属性名	説明
type-class	リソースタイプの実装クラスの完全修飾クラス名を指定します。
model-class	このリソースタイプで扱うリソースモデルクラスの完全修飾クラス名を指定します。

属性名	説明
cache-class	このリソースタイプで扱うキャッシュコントローラクラスを定義します。

サブジェクトタイプ拡張設定

項目
- 概要 - リファレンス - サブジェクトタイプ設定

概要

認可機構にサブジェクトタイプを追加するための設定です。
サブジェクトタイプと、サブジェクトタイプの取り扱うサブジェクトモデルを定義します。
サブジェクトタイプの詳細については認可仕様書を参照してください。

モジュール	テナント管理機能
フォーマットファイル (xsd)	WEB-INF/schema/authz-subject-type-config.xsd
設定場所	WEB-INF/conf/authz-subject-type-config/{任意}

```
<?xml version="1.0" encoding="UTF-8"?>
<authz-subject-type-config xmlns="http://www.intra-mart.jp/authz/authz-subject-type" >
  <subject-type type-class="jp.co.intra_mart.foundation.authz.subjecttype.RoleInfo" model-class="jp.co.intra_mart.foundation.admin.role.model.RoleInfo" />
</authz-subject-type-config>
```

リファレンス

サブジェクトタイプ設定

タグ名	subject-type
-----	--------------

サブジェクトタイプの実装クラスと、この実装クラスが扱うサブジェクトモデルを定義します。
【設定項目】

```
<authz-subject-type-config xmlns="http://www.intra-mart.jp/authz/authz-subject-type" >
  <subject-type type-class="jp.co.intra_mart.foundation.authz.subjecttype.RoleInfo" model-class="jp.co.intra_mart.foundation.admin.role.model.RoleInfo" />
</authz-subject-type-config>
```

必須項目	☐
複数設定	☐
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	authz-subject-type-config

属性名	説明
type-class	サブジェクトタイプの実装クラスの完全修飾クラス名を
model-class	このサブジェクトタイプで扱うサブジェクトモデルクラス 定します。

暗号化設定

- 項目
- 概要
 - リファレンス
 - 暗号化設定
 - 暗号化クラス設定

概要

Java API CryptionUtil を利用して、暗号／復号を行うための暗号化方式の

モジュール	テナント管理機能
フォーマットファイル(xsd)	WEB-INF/schema/cryption-config.xsd
設定場所	intra-mart Accel Platform 2013 Autumn 以降 WEB-INF/conf/cryption-config/cryption-config intra-mart Accel Platform 2013 Summer 以前 WEB-INF/conf/cryption-config.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<cryption-config xmlns="http://intra-mart.co.jp/system/security/cryption":

  <!-- 該当するカテゴリがない場合使われます -->
  <cryption category="default" default="true">
    <cryption-class>jp.co.intra_mart.foundation.security.cryption.standar
  </cryption>

  <!-- アカウントパスワード用 -->
  <cryption category="account">
    <cryption-class>jp.co.intra_mart.foundation.security.cryption.standar
  </cryption>

  <!-- システム管理者パスワード用 -->
  <cryption category="administrator">
    <cryption-class>jp.co.intra_mart.foundation.security.cryption.standar
  </cryption>

</cryption-config>
```

リファレンス

暗号化設定

タグ
名

暗号化方式の定義を行います。

【設定項目】

```
<cryption-config>
  <cryption category="default" default="true">
    . . . . .
  </cryption>
</cryption-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	crypton-config

【属性】

属性名	説明
category	暗号化設定にプログラムから利用するためのカテゴリを記します。
default	true の場合、システム・デフォルトの暗号化設定になります。

!

注意

default="true" の暗号化設定が複数ある場合、設定ファイルの先頭の default="true" の暗号化設定が見つからない場合、設定ファイルの最後の default="true" の暗号化設定が適用されます。

!

注意

カテゴリは一意な値となるように指定してください。

暗号化クラス設定

タグ名	crypton-class
-----	---------------

暗号化方式の実装クラスの設定を行います。

【設定項目】

```
<crypton-config>
  <crypton category="default" default="true">
    <crypton-class>jp.co.intra_mart.foundation.security.cryption.standard
  </crypton>
</crypton-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	Javaインタフェース jp.co.intra_mart.foundation.security.cryption.standard
単位・型	文字列
省略時のデフォルト値	なし
親タグ	crypton

項目

- 概要
- リファレンス
 - DeclaredSubjectResolverの追加設定

概要

認可のサブジェクト解決系を追加する設定です。ここではサブジェクトリゾルバ、DeclaredSubjectResolverについての詳細は認可仕

モジュール	テナント管理機能
フォーマットファイル (xsd)	WEB-INF/schema/declared-subject-resolvers.
設定場所	WEB-INF/conf/declared-subject-resolvers/{任

```
<?xml version="1.0" encoding="UTF-8"?>
<declared-subject-resolvers xmlns="http://www.intra-mart.jp/authz/decl
<class-name>jp.co.intra_mart.foundation.authz.subjectresolver.im_mas
</declared-subject-resolvers>
```

リファレンス

DeclaredSubjectResolverの追加設定

タ class-name

グ

名

認可機構に追加するDeclaredSubjectResolverを定義します。

【設定項目】

```
<declared-subject-resolvers xmlns="http://www.intra-mart.jp/authz/decl
<class-name>jp.co.intra_mart.foundation.authz.subjectresolver.im_mas
</declared-subject-resolvers>
```

必須項目	×
複数設定	○
設定値・設定する内容	認可機構に追加するDeclaredSubjectResolverのデ
単位・型	文字列
省略時のデフォルト値	なし
親タグ	declared-subject-resolvers

サブジェクトリゾルバ(OnDemandSubjectResol

項目

- 概要
- リファレンス
 - OnDemandSubjectResolverの追加設定

概要

認可のサブジェクト解決系を追加する設定です。ここではサブジェクトリゾル

モジュール	テナント管理機能
フォーマットファイル (xsd)	WEB-INF/schema/ondemand-subject-resolve
設定場所	WEB-INF/conf/ondemand-subject-resolvers/{

```
<?xml version="1.0" encoding="UTF-8"?>
<ondemand-subject-resolvers xmlns="http://www.intra-mart.jp/authz/on
<class-name>sample.TimeDependResolver</class-name>
</ondemand-subject-resolvers>
```

リファレンス

OnDemandSubjectResolverの追加設定

タ class-name
グ
名

認可機構に追加するOnDemandSubjectResolverを定義します。

【設定項目】

```
<ondemand-subject-resolvers xmlns="http://www.intra-mart.jp/authz/on
<class-name>sample.TimeDependResolver</class-name>
</ondemand-subject-resolvers>
```

必須項目	×
複数設定	○
設定値・設定する内容	認可機構に追加するOnDemandSubjectResolver
単位・型	文字列
省略時のデフォルト値	なし
親タグ	ondemand-subject-resolvers

認可ポリシーキャッシュ対象設定

- 項目
- 概要
 - リファレンス
 - キャッシュ対象設定
 - キャッシュ対象リソースタイプ設定

概要

認可ポリシー取得処理でのキャッシュ設定を行います。

モジュール	テナント管理機能
フォーマットファイル (xsd)	WEB-INF/schema/policy-cache-config.xsd
設定場所	WEB-INF/conf/policy-cache-config/{任意のファイ

```
<?xml version="1.0" encoding="UTF-8"?>
<policy-cache-config
  xmlns="http://www.intra-mart.jp/authz/policy-cache-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/authz/policy-cache-config
  <cache-target>
    <resource-type>service</resource-type>
  </cache-target>
</policy-cache-config>
```

リファレンス

キャッシュ対象設定

タグ名

キャッシュ対象を設定します。

【設定項目】

```
<policy-cache-config>
  <cache-target>
    ...
  </cache-target>
</policy-cache-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	policy-cache-config

キャッシュ対象リソースタイプ設定

タグ名

認可ポリシーのキャッシュの対象となるリソースタイプを設定します。

【設定項目】

```
<policy-cache-config>
  <cache-target>
    <resource-type>service</resource-type>
  </cache-target>
</policy-cache-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	キャッシュ対象とするリソースタイプを記述します。
単位・型	文字列

省略時のデフォルト値	なし
親タグ	cache-target

パスワード履歴管理設定

項目

- 概要
- リファレンス
 - デフォルト設定
 - テナントごとの設定
 - 初回ログイン時パスワード変更
 - パスワード有効日数
 - 管理するパスワード履歴数
 - パスワード履歴管理対象外クライアントタイプ
 - パスワード変更画面パス
 - パスワード文字列チェック
 - パスワード暗号化アルゴリズムクラス
 - パスワード変更後ログイン画面表示
 - パスワードの長さチェック
 - パスワード利用可能文字(英文字)
 - パスワード利用可能文字(数字)
 - パスワード利用可能文字(その他)
 - パスワード履歴内のパスワードの利用禁止
 - ユーザコードパスワードの利用禁止

概要

パスワードの履歴管理機能の設定です。
パスワード履歴管理機能は、パスワードの履歴管理を行い、パスワードの定
また、パスワードの様々な制約を設定することが可能です。

モジュール	テナント管理機能モジュール
フォーマットファ イル(xsd)	WEB-INF/schema/password-history-config.xsd
設定場所	WEB-INF/conf/password-history.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<password-history>
  <group-default accessor-class="jp.co.intra_mart.foundation.security.pa
    <change-password-first-login>true</change-password-first-login>
    <password-expire-limit>0</password-expire-limit>
    <password-history-count>0</password-history-count>
    <deny-client-types>sp</deny-client-types>
    <password-expire-page>/user/password/expire</password-expire-p
    <check-password enable="true">
      <check-password-length enable="false" min="0" max="50"/>
      <allow-latin-letters required="false">ABCDEFGHIJKLMNOPQRSTUVWXYZ
      <allow-number required="false">0123456789</allow-number>
      <allow-extra-char required="false">!"#$%&apos;()+,-./
      <deny-old-password>false</deny-old-password>
      <deny-userid>false</deny-userid>
    </check-password>
    <password-cryption-class>jp.co.intra_mart.foundation.security.cryptic
    <return-initial-page>false</return-initial-page>
  </group-default>

  <group name="secondary" accessor-class="jp.co.intra_mart.foundation.
    <change-password-first-login>true</change-password-first-login>
    <password-expire-limit>0</password-expire-limit>
    <password-history-count>0</password-history-count>
    <deny-client-types>sp</deny-client-types>
    <password-expire-page>/user/password/expire</password-expire-p
    <check-password enable="true">
      <check-password-length enable="false" min="0" max="50"/>
      <allow-latin-letters required="false">ABCDEFGHIJKLMNOPQRSTUVWXYZ
      <allow-number required="false">0123456789</allow-number>
      <allow-extra-char required="false">!"#$%&apos;()+,-./
      <deny-old-password>false</deny-old-password>
      <deny-userid>false</deny-userid>
    </check-password>
    <password-cryption-class>jp.co.intra_mart.foundation.security.cryptic
    <return-initial-page>false</return-initial-page>
  </group>

  <group name="thirdly" accessor-class="jp.co.intra_mart.foundation.seci
    <change-password-first-login>true</change-password-first-login>
    <password-expire-limit>0</password-expire-limit>
    <password-history-count>0</password-history-count>
    <deny-client-types>sp</deny-client-types>
    <password-expire-page>/user/password/expire</password-expire-p
    <check-password enable="true">
      <check-password-length enable="false" min="0" max="50"/>
      <allow-latin-letters required="false">ABCDEFGHIJKLMNOPQRSTUVWXYZ
      <allow-number required="false">0123456789</allow-number>
      <allow-extra-char required="false">!"#$%&apos;()+,-./
      <deny-old-password>false</deny-old-password>
      <deny-userid>false</deny-userid>
    </check-password>
    <password-cryption-class>jp.co.intra_mart.foundation.security.cryptic
    <return-initial-page>false</return-initial-page>
  </group>
</password-history>
```

i コラム

groupタグは複数設定が行えます。groupタグを複数定義すること
詳細はリファレンス内の「テナントごとの設定」を参照してください。

i コラム

password-history.xmlはpassword-history-config.xsdを参照し
xmlファイル修正時は上記設定例を参照の上、実施してください。

リファレンス

デフォルト設定

タ group-default
グ
名

パスワード履歴管理機能の標準設定を行います。

【設定項目】

```
<password-history>
  <group-default accessor-class="StandardPasswordHistoryAccessor">
    ...
  </group-default>
</password-history>
```

必須項目	○
複数設定	×
設定値・設定する内容	パスワード履歴管理機能の標準設定を設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	password-history

【属性】	
属性名	説明
accessor-class	パスワード履歴管理モジュールの実装クラスを設定します。 jp.co.intra_mart.foundation.security.password.PasswordHistoryAccessor を実装している必要があります。

テナントごとの設定

タグ名	group
-----	-------

テナントごとに、パスワード履歴管理機能の設定を行います。

【設定項目】

```
<password-history>
  <group name="secondary" accessor-class="StandardPasswordHistoryAccessor">
    ...
  </group>
</password-history>
```

必須項目	×
複数設定	○
設定値・設定する内容	テナントごとの、パスワード履歴管理機能を設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	password-history

【属性】	
属性名	説明
name	テナントID
accessor-class	パスワード履歴管理モジュールの実装クラス jp.co.intra_mart.foundation.security.password.PasswordHistoryAccessor を実装している必要があります。

初回ログイン時パスワード変更

タグ名	change-password-first-login
-----	-----------------------------

初回ログイン時にパスワード変更を要求するかどうかを指定します。
ユーザ属性の初回ログイン設定の値をチェックして、初回ログイン判定を行います。

```
<password-history>
  <group-default accessor-class="jp.co.intra_mart.foundation.security.pa
    <change-password-first-login>true</change-password-first-login>
  </group-default>
</password-history>
```

必須項目	×				
複数設定	×				
設定値・設定 する内容	<table> <tr> <td>false</td><td>初期ログイン時にパスワード変更を要求しま</td></tr> <tr> <td>true</td><td>初期ログイン時にパスワード変更を要求しま</td></tr> </table>	false	初期ログイン時にパスワード変更を要求しま	true	初期ログイン時にパスワード変更を要求しま
false	初期ログイン時にパスワード変更を要求しま				
true	初期ログイン時にパスワード変更を要求しま				
単位・型	真偽値 (true/false)				
省略時のデ フォルト値	親タグが group-default の場合、true 親タグが group の場合、group-default タグ内の設定値				
親タグ	group-default, group				

パスワード有効日数

タグ名 password-expire-limit

パスワードの有効日数を設定します。
0を指定した場合、有効期間は無制限になります。

【設定項目】

```
<password-history>
  <group-default accessor-class="jp.co.intra_mart.foundation.security.pa
    <password-expire-limit>0</password-expire-limit>
  </group-default>
</password-history>
```

必須項目	×
複数設定	×
設定値・設 定する内容	パスワードの有効日数を設定します。
単位・型	数値 (0 -)
省略時のデ フォルト値	親タグが group-default の場合、0 親タグが group の場合、group-default タグ内の設定値
親タグ	group-default, group

管理するパスワード履歴数

タグ名 password-history-count

パスワードの履歴を管理する世代。
0を指定した場合、履歴管理を行いません。

【設定項目】

```
<password-history>
  <group-default accessor-class="jp.co.intra_mart.foundation.security.pa
    <password-history-count>0</password-history-count>
  </group-default>
</password-history>
```

必須項目	×
複数設定	×
設定値・設 定する内容	管理するパスワード履歴数を設定します。

単位・型	数値 (0 -)
省略時のデフォルト値	親タグが group-default の場合、0 親タグが group の場合、group-default タグ内の設定値
親タグ	group-default, group

パスワード履歴管理対象外クライアントタイプ

タグ名	deny-client-types
-----	-------------------

パスワード履歴管理対象とならないクライアントタイプを設定します。
複数指定する場合は、カンマ区切りで指定してください。

【設定項目】

```
<password-history>  
  <group-default accessor-class="jp.co.intra_mart.foundation.security.pa  
    <deny-client-types>sp</deny-client-types>  
  </group-default>  
</password-history>
```

必須項目	×
複数設定	×
設定値・設定する内容	パスワード履歴管理対象とならないクライアントタイプ
単位・型	文字列 (xxxxxx,xxxxxxxx)
省略時のデフォルト値	親タグが group-default の場合、なし 親タグが group の場合、group-default タグ内の設定値
親タグ	group-default, group

パスワード変更画面パス

タグ名	password-expire-page
-----	----------------------

パスワード変更画面のパスを設定します。

【設定項目】

```
<password-history>  
  <group-default accessor-class="jp.co.intra_mart.foundation.security.pa  
    <password-expire-page>/user/password/expire</password-expire-p  
  </group-default>  
</password-history>
```

必須項目	×
複数設定	×
設定値・設定する内容	パスワード変更画面のパスを設定します。
単位・型	文字列 (xxx/xxx/xxx/xxxxx)
省略時のデフォルト値	親タグが group-default の場合、/user/password/expire 親タグが group の場合、group-default タグ内の設定値
親タグ	group-default, group

【属性】

属性名	説明
client-type	クライアントタイプを設定します。 クライアントタイプ別にパスワード変更画面を設 てください。 クライアントタイプが設定されていない場合はラ 変更画面として扱われます。

タグ check-password
グループ名

パスワード文字列に対する有効チェックの内容を設定します。

【設定項目】

```
<password-history>  
  <group-default accessor-class="jp.co.intra_mart.foundation.security.pa  
    <check-password enable="true">  
    ...  
  </check-password>  
</group-default>  
</password-history>
```

必須項目	x
複数設定	x
設定値・設定する内容	パスワード文字列に対する有効チェックの内容を設定し
単位・型	なし
省略時のデフォルト値	親タグが group-default の場合、なし 親タグが group の場合、group-default タグ内の設定値
親タグ	group-default, group

【属性】

属性名	説明
enable	パスワード文字列の有効チェックを行うかどうかを設定します。

パスワード暗号化アルゴリズムクラス

タグ password-cryption-class
名

パスワードの履歴を保管する際に用いられる暗号化アルゴリズムクラス。

【設定項目】

```
<password-history>  
  <group-default accessor-class="jp.co.intra_mart.foundation.security.pa  
    <password-cryption-class>jp.co.intra_mart.foundation.security.crypti  
  </group-default>  
</password-history>
```

必須項目	x
複数設定	x
設定値・設定する内容	jp.co.intra_mart.foundation.security.cryption.
単位・型	文字列 (xxxxxxxxxxxxxx)
省略時のデフォルト値	親タグが group-default の場合、jp.co.intra_m 親タグが group の場合、group-default タグ内
親タグ	group-default, group

パスワード変更後ログイン画面表示

タグ return-initial-page
グループ名

パスワード変更要求画面よりパスワードを変更した後に、ログイン画面に戻

【設定項目】

```
<password-history>
  <group-default accessor-class="jp.co.intra_mart.foundation.security.pa
  <return-initial-page>false</return-initial-page>
</group-default>
</password-history>
```

必須項目	×
複数設定	×
設定値・設定 する内容	false パスワード変更後にユーザのホーム画面を遷移する true パスワード変更後にログイン画面を表示する
単位・型	真偽値 (true/false)
省略時のデ フォルト値	親タグが group-default の場合、false 親タグが group の場合、group-default タグ内の設定値
親タグ	group-default, group

パスワードの長さチェック

タグ check-password-length
名

パスワードの長さ制約に関する設定です。

【設定項目】

```
<password-history>
  <group-default>
    <check-password enable="true">
      <check-password-length enable="false" min="0" max="50"/>
    </check-password>
  </group-default>
</password-history>
```

必須項目	×
複数設定	×
設定値・設定する 内容	パスワードの長さ制約を設定します。
単位・型	なし
省略時のデフォルト 値	親タグが group-default/check-password の場合、false 親タグが group/check-password の場合、group-default/check-password タグ内の設定値
親タグ	check-password

【属性】

属性名	説明
enable	パスワードの長さ制約を行うかどうかを設定します。
min	パスワードの最小の長さを設定します。最小値に設定した値未満の長さのパスワードは利用できません。
max	パスワードの最大の長さを設定します。最大値に設定した値以上の長さのパスワードは利用できません。

パスワード利用可能文字(英文字)

タグ allow-latin-letters
名

パスワードに利用できる英文字を設定します。

【設定項目】

```
<password-history>
<group-default>
<check-password enable="true">
  <allow-latin-letters required="false">ABCDEFGHIJKLMNOPQRSTUVWXYZ
</check-password>
</group-default>
</password-history>
```

必須項目	×
複数設定	×
設定値・設定する内容	パスワードに利用できる英文字を設定します。
単位・型	文字列 (xxxxxxxxxxxxxx)
省略時のデフォルト値	親タグが group-default/check-password の場合 親タグが group/check-password の場合、group-
親タグ	check-password

【属性】

属性名	説明
required	パスワード利用可能文字(英文字)の制約を行う true の場合は、少なくとも1文字は設定した文

パスワード利用可能文字(数字)

タグ allow-number
名

パスワードに利用できる数字を設定します。

【設定項目】

```
<password-history>
<group-default>
<check-password enable="true">
  <allow-number required="false">0123456789</allow-number>
</check-password>
</group-default>
</password-history>
```

必須項目	×
複数設定	×
設定値・設定する内容	パスワードに利用できる数字を設定します。
単位・型	文字列 (xxxxxxxxxxxxxx)
省略時のデフォルト値	親タグが group-default/check-password の場合 親タグが group/check-password の場合、group-
親タグ	check-password

【属性】

属性名	説明
required	パスワード利用可能文字(数字)の制約を行う true の場合は、少なくとも1文字は設定した文

パスワード利用可能文字(その他)

タグ allow-extra-char
名

パスワードに利用できるその他の文字を設定します。

【設定項目】

```

<password-history>
  <group-default>
    <check-password enable="true">
      <allow-extra-char required="false">!"&quot;#$$%&amp;&apos;()+,-./
    </check-password>
  </group-default>
</password-history>

```

必須項目	×
複数設定	×
設定値・設定する 内容	パスワードに利用できるその他の文字を設定します
単位・型	文字列 (xxxxxxxxxxxxxx)
省略時のデフォルト 値	親タグが group-default/check-password の場合 親タグが group/check-password の場合、group-
親タグ	check-password

【属性】

属性名	説明
required	パスワード利用可能文字(その他)の制約を行 true の場合は、少なくとも1文字は設定した文

パスワード履歴内のパスワードの利用禁止

タグ名 deny-old-password

パスワード履歴に存在するパスワードの利用を禁止します。

【設定項目】

```

<password-history>
  <group-default>
    <check-password enable="true">
      <deny-old-password>false</deny-old-password>
    </check-password>
  </group-default>
</password-history>

```

必須項目	×
複数設定	×
設定値・設定する 内容	false パスワード履歴に存在するパスワードの true パスワード履歴に存在するパスワードの
単位・型	真偽値 (true/false)
省略時のデフォルト 値	親タグが group-default/check-password の場合 親タグが group/check-password の場合、group-
親タグ	check-password

ユーザコードパスワードの利用禁止

タグ名 deny-userid

ユーザコードと同じパスワードの利用を禁止します。

【設定項目】

```
<password-history>
<group-default>
<check-password enable="true">
  <deny-userid>false</deny-userid>
</check-password>
</group-default>
</password-history>
```

必須項目	×
複数設定	×
設定値・設定する内容	false ユーザコードと同じパスワードの利用を認 true ユーザコードと同じパスワードの利用を認
単位・型	真偽値 (true/false)
省略時のデフォルト値	親タグが group-default/check-password の場合 親タグが group/check-password の場合、group-
親タグ	check-password

スクリプト開発モデルルーティング設定

項目
● 概要
● リファレンス
○ デフォルト認可設定
○ ファイルマッピング設定
○ フォルダマッピング設定
○ 認可設定
○ 認可パラメータ

概要

URLに対して、スクリプト開発モデルの処理のマッピングを行うための設定

モジュール	テナント管理機能
フォーマットファイル(xsd)	WEB-INF/schema/routing-jssp-config.xsd
設定場所	WEB-INF/conf/routing-jssp-config/{任意のファイル}

```
<?xml version="1.0" encoding="UTF-8"?>
<routing-jssp-config
  xmlns="http://www.intra-mart.jp/router/routing-jssp-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/router/routing-jssp-config
  <authz-default uri="service://application/default" action="execute" />
  <file-mapping path="/user/application/main" page="user/application/view/
  <file-mapping path="/user/application/list" page="user/application/view/
  <authz mapper="sample-mapper">
    <param key="foo" value="FOO" />
    <param key="bar" value="BAR" />
  </authz>
  </file-mapping>
  <file-mapping path="/user/application/process/create" page="user/appli
  <authz uri="service://application/update" action="execute" />
  </file-mapping>
  <file-mapping path="/user/application/process/update" page="user/appli
  <authz uri="service://application/update" action="execute" />
  </file-mapping>
  <file-mapping path="/user/application/process/delete" page="user/appli
  <authz uri="service://application/update" action="execute" />
  </file-mapping>
</routing-jssp-config>
```

リファレンス

タグ名
authz-default

同じ設定ファイル上でのデフォルトの認可設定を行います。

【設定項目】

```
<routing-jssp-config>
<authz-default uri="service://application/default" action="execute" />
</routing-jssp-config>
```

必須項目	x
複数設定	x
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	routing-jssp-config

【属性】

属性名	説明
uri	認可リソースURIを指定します。
action	認可アクションを指定します。
mapper	認可リソースマッパーを指定します。 使用可能な値はauthz-resource-mapper設定で設定済みです。

 注意

この設定は、以下のいずれかの設定を行う必要があります。
uri属性とaction属性
mapper属性

 注意

authz-defaultタグを省略した場合、file-mappingタグ、folder-ma

ファイルマッピング設定

タグ名
file-mapping

URLとスクリプト開発モデルのプログラムのマッピングを行います。

【設定項目】

```
<routing-jssp-config>
  <authz-default uri="/service:/application/default" action="execute" />
  <file-mapping path="/user/application/main" page="user/application/view/...
  <file-mapping path="/user/application/list" page="user/application/view/...
  ...
  </file-mapping>
</routing-jssp-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	routing-jssp-config

【属性】

属性名	説明
path	マッピングを行うURLを指定します。 値の末尾に ワイルドカード 「*」 を指定すること 値に {<識別子>} を記述することでURLの途中 プログラム中で使用可能です。 例: /sample/view/{dataId}
page	マッピングを行うスクリプト開発モデルのプログラ
action	page属性に指定されたプログラムの実行前に from属性を指定した場合はfrom属性に指定さ from属性が未指定の場合はpage属性に指定さ す。
from	action属性で指定した関数を呼び出すプログラ
client-type	マッピングが有効となるクライアントタイプを指定 省略時は、すべてのクライアントタイプにおいてマ

 コラム

page, from 属性で指定するスクリプト開発モデルのプログラムは
スクリプト開発モデルプログラムのソースディレクトリからの相対
デフォルトではWEB-INF/jssp/srcからの相対パス形式になります

フォルダマッピング設定

タグ名	folder-mapping
-----	----------------

URLのパターンとスクリプト開発モデルのプログラムのマッピングを行います。
特定のスクリプト開発モデルのプログラムのディレクトリ配下すべてのプログラ

【設定項目】

```
<routing-jssp-config>
  <folder-mapping path-prefix="/user/application" folder="user/application/...
  ...
  </folder-mapping>
</routing-jssp-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	routing-jssp-config

【属性】

属性名	説明
path-prefix	マッピングを行うURLを指定します。
folder	マッピングを行うスクリプト開発モデルのプログラムのデシます。
client-type	マッピングが有効となるクライアントタイプを指定します。 省略時は、すべてのクライアントタイプにおいて有効なマます。

 注意

この設定を使用するとルーティングに時間が掛かる、スクリプト開
れません。
使用するスクリプト開発モデルのプログラムに対してfile-mappin

認可設定

タグ名	authz
-----	-------

認可設定

【設定項目】

```
<routing-jssp-config>
<file-mapping path="/user/application/list" page="user/application/view/
<authz mapper="sample-mapper">
...
</authz>
</file-mapping>
<file-mapping path="/user/application/process/create" page="user/appli
<authz uri="service://application/update" action="execute" />
</file-mapping>
</routing-jssp-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	なし

省略時のデフォルト値	なし
親タグ	file-mapping
子タグ	folder-mapping

【属性】

属性名	説明
uri	認可リソースURIを指定します。
action	認可アクションを指定します。
mapper	認可リソースマッパーを指定します。 使用可能な値はルーティング用 認可リソースマッパー記述の値です。

 **コラム**

authz-defaultタグが指定されていて、かつ、file-mappingタグまたは

 **注意**

この設定は、以下のいずれかの設定を行う必要があります。

- uri属性とaction属性
- mapper属性

認可パラメータ

タグ名	param
-----	-------

認可リソースマッパーに対するパラメータを設定します。
親タグにmapper属性が指定されていない場合は無視されます。

【設定項目】

```
<routing-jssp-config>
  <file-mapping path="/user/application/list" page="user/application/view/
  <authz mapper="sample-mapper">
    <param key="foo" value="FOO" />
    <param key="bar" value="BAR" />
  </authz>
  </file-mapping>
</routing-jssp-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし

省略	なし
時の	
デ	
フォ	
ルト	
値	
親タ	authz-default
グ	authz

【属性】

属性名	説明	必須	デフォルト値
key	パラメータのキーを指定します。	○	なし
value	パラメータの値を指定します。	○	なし

IM-JavaEE Frameworkルーティング設定

項目
- 概要 - リファレンス - デフォルト認可設定 - サービスマッピング設定 - アプリケーションマッピング設定 - 認可設定 - 認可パラメータ

概要

URLに対して、IM-JavaEE Frameworkの処理のマッピングを行うための設

モジュール	テナント管理機能
フォーマットファイル	WEB-INF/schema/routing-service-config.xsd (xsd)
設定場所	WEB-INF/conf/routing-service-config/{任意のフ

```
<?xml version="1.0" encoding="UTF-8"?>
<routing-service-config
  xmlns="http://www.intra-mart.jp/router/routing-service-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/router/routing-service-co
  <authz-default uri="service://application/default" action="execute" />
  <service-mapping path="/user/application/main" application="applicac
  <service-mapping path="/user/application/list" application="application"
  <authz mapper="sample-mapper">
    <param key="foo" value="FOO" />
    <param key="bar" value="BAR" />
  </authz>
</service-mapping>
<service-mapping path="/user/application/process/create" application=
  <authz uri="service://application/update" action="execute" />
</service-mapping>
<service-mapping path="/user/application/process/update" application=
  <authz uri="service://application/update" action="execute" />
</service-mapping>
<service-mapping path="/user/application/process/delete" application=
  <authz uri="service://application/update" action="execute" />
</service-mapping>
</routing-service-config>
```

リファレンス

デフォルト認可設定

タグ
authz-default
名

同じ設定ファイル上でのデフォルトの認可設定を行います。

【設定項目】

```
<routing-service-config>
  <authz-default uri="service://application/default" action="execute" />
</routing-service-config>
```

必須項目	x
複数設定	x
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	routing-service-config

【属性】

属性名	説明
uri	認可リソースURIを指定します。
action	認可アクションを指定します。
mapper	認可リソースマッパーを指定します。 使用可能な値はルーティング用 認可リソースマッパー記述の値です。

注意

この設定は、以下のいずれかの設定を行う必要があります。

- uri属性とaction属性
- mapper属性

注意

authz-defaultタグを省略した場合、service-mappingタグ、applic

サービスマッピング設定

タグ
service-mapping
名

URLとIM-JavaEE Frameworkのプログラムのマッピングを行います。

【設定項目】

```
<routing-service-config>
  <authz-default uri="service://application/default" action="execute" />
  <service-mapping path="/user/application/main" application="application">
  <service-mapping path="/user/application/list" application="application">
  ...
  </service-mapping>
  <service-mapping path="/user/application/process/create" application="application">
  ...
  </service-mapping>
</routing-service-config>
```

必須項目	x
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	routing-service-config

【属性】

属性名	説明
path	マッピングを行うURLを指定します。 値の末尾に ワイルドカード 「*」 を指定することが可能。 値に {<識別子>} を記述することでURLの途中の HttpServletRequest#getAttribute()で取得可能で 例: /sample/view/{dataId}
application	マッピングを行うIM-JavaEE Frameworkのアプリケーション
service	マッピングを行うIM-JavaEE Frameworkのサービス
client-type	マッピングが有効となるクライアントタイプを指定し、省略時は、すべてのクライアントタイプにおいて有効

アプリケーションマッピング設定

タグ名	application-mapping
-----	---------------------

URLのパターンとIM-JavaEE Frameworkのアプリケーションのマッピングを特定のIM-JavaEE Frameworkのアプリケーションが持つすべてのサービス

【設定項目】

```
<routing-service-config>
<application-mapping path-prefix="/user/application" application="appl
...
</application-mapping>
</routing-service-config>
```

必須項目	x
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	routing-service-config

【属性】

属性名	説明
path-prefix	マッピングを行うURLを指定します。
application	マッピングを行うIM-JavaEE Frameworkのプログラム指定します。
client-type	マッピングが有効となるクライアントタイプを指定します。省略時は、すべてのクライアントタイプにおいて有効な

 注意

この設定を使用するとルーティングに時間が掛かる、IM-JavaEE
れません。
使用するIM-JavaEE Frameworkのサービスに対してservice-m

認可設定

タ authz
グ
名

認可設定

【設定項目】

```
<routing-service-config>  
<service-mapping path="/user/application/list" application="application"  
<authz mapper="sample-mapper">  
...  
</authz>  
</service-mapping>  
<service-mapping path="/user/application/process/create" application=  
<authz uri="service://application/update" action="execute" />  
</service-mapping>  
</routing-service-config>
```

必須項目	x
複数設定	x
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	service-mapping application-mapping

【属性】

属性名	説明
uri	認可リソースURIを指定します。
action	認可アクションを指定します。
mapper	認可リソースマッパーを指定します。 使用可能な値はルーティング用 認可リソースマッパー記 の値です。

 コラム

authz-defaultタグが指定されていて、かつ、service-mappingタグ

 注意

この設定は、以下のいずれかの設定を行う必要があります。

- uri属性とaction属性
- mapper属性

認可パラメータ

タグ
パラメータ
名

認可リソースマップパーに対するパラメータを設定します。
親タグにmapper属性が指定されていない場合は無視されます。

【設定項目】

```
<routing-service-config>
  <service-mapping path="/user/application/list" application="application">
    <authz mapper="sample-mapper">
      <param key="foo" value="FOO" />
      <param key="bar" value="BAR" />
    </authz>
  </service-mapping>
</routing-service-config>
```

必須項目	x
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	authz-default authz

【属性】

属性名	説明	必須	デフォルト値
key	パラメータのキーを指定します。	○	なし
value	パラメータの値を指定します。	○	なし

サブレットルーティング設定

項目
- 概要 - リファレンス - デフォルト認可設定 - サブレットマッピング設定 - 認可設定 - 認可パラメータ

概要

URLに対して、フォワード先のサブレットパスへのマッピングを行うための設定

モジュール	テナント管理機能
フォーマットファイル (xsd)	WEB-INF/schema/routing-servlet-config.xsd
設定場所	WEB-INF/conf/routing-servlet-config/{任意のフ

```
<?xml version="1.0" encoding="UTF-8"?>
<routing-servlet-config
  xmlns="http://www.intra-mart.jp/router/routing-servlet-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/router/routing-servlet-con
<authz-default uri="service://application/default" action="execute" />
<servlet-mapping path="/user/application/main" servlet="application/m
<servlet-mapping path="/user/application/list" servlet="application/list"
<authz mapper="sample-mapper">
  <param key="foo" value="FOO" />
  <param key="bar" value="BAR" />
</authz>
</servlet-mapping>
<servlet-mapping path="/user/application/process/create" servlet="app
<authz uri="service://application/update" action="execute" />
</servlet-mapping>
<servlet-mapping path="/user/application/process/update" servlet="app
<authz uri="service://application/update" action="execute" />
</servlet-mapping>
<servlet-mapping path="/user/application/process/delete" servlet="app
<authz uri="service://application/update" action="execute" />
</servlet-mapping>
</routing-servlet-config>
```

リファレンス

デフォルト認可設定

タグ authz-default
グループ
名前

同じ設定ファイル上でのデフォルトの認可設定を行います。

【設定項目】

```
<routing-servlet-config>
  <authz-default uri="service://application/default" action="execute" />
</routing-servlet-config>
```

必須項目	x
複数設定	x
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	routing-servlet-config

【属性】

属性名	説明
uri	認可リソースURIを指定します。
action	認可アクションを指定します。

属性名	説明
mapper	認可リソースマッパーを指定します。 使用可能な値はルーティング用 認可リソースマッパー記 の値です。

 注意

この設定は、以下のいずれかの設定を行う必要があります。
uri属性とaction属性
mapper属性

 注意

authz-defaultタグを省略した場合、servlet-mappingタグにauthz

サブレットマッピング設定

タグ名	servlet-mapping
-----	-----------------

URLとサブレットパスのマッピングを行います。

【設定項目】

```
<routing-servlet-config>  
<authz-default uri="service://application/default" action="execute" />  
<servlet-mapping path="/user/application/main" servlet="application/main" />  
<servlet-mapping path="/user/application/list" servlet="application/list" />  
...  
</servlet-mapping>  
</routing-servlet-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	routing-servlet-config

【属性】

属性名	説明
path	マッピングを行うURLを指定します。 値の末尾に ワイルドカード 「*」 を指定することが可 値に {識別子} を記述することでURLの途中の HttpServletRequest.getAttribute()で取得可能で 例: /sample/view/{dataId}
servlet	マッピングを行うサブレットパスを指定します。
client-type	マッピングが有効となるクライアントタイプを指定し 省略時は、すべてのクライアントタイプにおいて有効

 注意

認可によるアクセス制御について
サブレットに対して認可によるアクセス制御を行う場合、直接
具体的には、以下の3つで同じパスを指定したマッピング設定

- web.xml に定義されたサーブレットに対するURL (servlet-name)
- <routing-servlet-config> タグ内 <servlet-mapping> タグ内
- <routing-servlet-config> タグ内 <servlet-mapping> タグ内

このマッピング設定がない場合、認可を経由せずにサーブレットにアクセス可能。

例えば web.xml で以下のようにサーブレットが定義されている。

```
<servlet-mapping>
  <servlet-name>SampleServlet</servlet-name>
  <url-pattern>/sample_servlet</url-pattern>
</servlet-mapping>
```

この場合 routing-servlet-config には、以下のような <servlet-mapping> が定義されている。

```
<servlet-mapping path="/sample_servlet" servlet="sample_servlet">
  <authz uri="service://sample/servlet" action="execute" />
</servlet-mapping>
```

ワイルドカード指定について

path属性、servlet属性には、* (アスタリスク) 等のワイルドカードが使用可能。
web.xml で定義されたサーブレットに対するURLがワイルドカードで指定されている場合は、そのURLに一致するリソースに対してのみ認可が適用される。

認可設定

タグ名
authz

認可設定

【設定項目】

```
<routing-servlet-config>
  <servlet-mapping path="/user/application/list" servlet="application/list">
    <authz mapper="sample-mapper">
      ...
    </authz>
  </servlet-mapping>
  <servlet-mapping path="/user/application/process/create" servlet="application/process/create">
    <authz uri="service://application/update" action="execute" />
  </servlet-mapping>
</routing-servlet-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	servlet-mapping

【属性】

属性名	説明
uri	認可リソースURIを指定します。
action	認可アクションを指定します。

属性名	説明
mapper	認可リソースマッパーを指定します。 使用可能な値はルーティング用 認可リソースマッパー記 の値です。

 コラム

authz-defaultタグが指定されていて、かつ、servlet-mappingタグ

 注意

この設定は、以下のいずれかの設定を行う必要があります。
uri属性とaction属性
mapper属性

認可パラメータ

タグ名

認可リソースマッパーに対するパラメータを設定します。
親タグにmapper属性が指定されていない場合は無視されます。

【設定項目】

```
<routing-servlet-config>  
<servlet-mapping path="/user/application/list" servlet="application/list" ,  
<authz mapper="sample-mapper">  
  <param key="foo" value="FOO" />  
  <param key="bar" value="BAR" />  
</authz>  
</servlet-mapping>  
</routing-servlet-config>
```

必須項目	x
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	authz-default authz

【属性】

属性名	説明	必須	デフォルト値
key	パラメータのキーを指定します。	○	なし
value	パラメータの値を指定します。	○	なし

ショートカットアクセス設定

- 項目
- 概要
 - リファレンス
 - ショートカット操作モジュール
 - ショートカット操作実装クラス
 - エラーページパス
 - メインページパス
 - ショートカット情報検証設定

概要

ショートカットアクセス機能の設定です。
ショートカットアクセス機能はショートカットアクセス用のURLからアクセスす

モジュール	テナント管理機能モジュール
フォーマット ファイル (xsd)	なし
設定場所	WEB-INF/conf/short-cut-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<short-cut-config>
  <short-cut-accessor>
    <short-cut-accessor-class>jp.co.intra_mart.foundation.security.shortcut.ShortcutAccessImpl</short-cut-accessor-class>
    <error-page>/user/shortcut/error</error-page>
    <main-page>/home</main-page>
    <deny-client-types>sp</deny-client-types>
    <validator code="RegExpUser" class="jp.co.intra_mart.foundation.security.validator.RegExpValidator"></validator>
    <validator code="RoleUser" class="jp.co.intra_mart.foundation.security.validator.RoleValidator"></validator>
    <validator code="Script" class="jp.co.intra_mart.foundation.security.validator.ScriptValidator"></validator>
  </short-cut-accessor>
</short-cut-config>
```

 コラム

short-cut-config.xmlのXMLスキーマ定義ファイル(xsdファイル)に
xmlファイル修正時は上記設定例を参照の上、実施してください。

リファレンス

ショートカット操作モジュール

タ グ 名	short-cut-accessor
-------------	--------------------

ショートカット操作モジュールに関する設定を行います。

【設定項目】

```
<short-cut-config>
  <short-cut-accessor>
    ...
  </short-cut-accessor>
</short-cut-config>
```

必須項目	○
複数設定	×
設定値・設 定する内容	ショートカット操作モジュールに関する設定を行います。
単位・型	なし
省略時のデ フォルト値	なし

親タグ	short-cut-config
-----	------------------

ショートカット操作実装クラス

タグ名	short-cut-accessor-class
-----	--------------------------

ショートカット情報を操作するモジュール(ショートカットアクセッサ)の実装クラス

【設定項目】

```
<short-cut-config>
  <short-cut-accessor>
    <short-cut-accessor-class>jp.co.intra_mart.foundation.security.shortcut.S
  </short-cut-accessor>
</short-cut-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	jp.co.intra_mart.foundation.security.shortcut.S
単位・型	文字列 (xxxxxxxxxxxxxx)
省略時のデフォルト値	なし
親タグ	short-cut-accessor

エラーページパス

タグ名	error-page
-----	------------

ショートカット情報が存在しない場合に表示するページパスを指定します。

【設定項目】

```
<short-cut-config>
  <short-cut-accessor>
    <error-page>/user/shortcut/error</error-page>
  </short-cut-accessor>
</short-cut-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ショートカット情報が存在しない場合に表示するページパス
単位・型	文字列 (xxxxxxxxxxxxxx)
省略時のデフォルト値	なし
親タグ	short-cut-accessor

メインページパス

タグ名	main-page
-----	-----------

ショートカットページを表示するためのメインページパスを設定します。

【設定項目】

```
<short-cut-config>
<short-cut-accessor>
  <main-page>/home</main-page>
</short-cut-accessor>
</short-cut-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ショートカットページを表示するためのメインページバ
単位・型	文字列 (xxxxxxxxxxxxxx)
省略時のデフォルト値	なし
親タグ	short-cut-accessor

ショートカット情報検証設定

タグ名

検証コードと検証プログラムの紐付けを行い、ショートカット情報の検証プ

【設定項目】

```
<short-cut-config>
<short-cut-accessor>
  <validator code="RegExpUser" class="jp.co.intra_mart.foundation.sec
</short-cut-accessor>
</short-cut-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	ショートカット情報の検証プログラムを設定します。
単位・型	なし
省略時のデフォルト値	なし
親タグ	short-cut-accessor

【属性】

属性名	説明
code	検証コードを指定します。
class	検証プログラムのクラスを指定します。 jp.co.intra_mart.foundation.security.shortcut.Sho クラスを指定します。

システム期間情報の管理

- 項目
- 概要
 - リファレンス
 - システム最小日の設定
 - システム最大日の設定

概要

システム期間情報を管理するための設定を行います。

モジュール	テナント管理機能
フォーマットファイル(xsd)	WEB-INF/schema/system-valid-date-config.xsd
設定場所	WEB-INF/conf/system-valid-date-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<system-valid-date-config
  xmlns="http://www.intra_mart.co.jp/foundation/admin/validddate/config/sy
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra_mart.co.jp/foundation/admin/valid
  <min-date value="1900-01-01" />
  <max-date value="3000-01-01" />
</system-valid-date-config>
```

リファレンス

システム最小日の設定

タグ名

システムで取り扱う最小日を設定します。

【設定項目】

```
<system-valid-date-config>
  <min-date value="1900-01-01" />
</system-valid-date-config>
```

必須項目

複数設定

設定値・設定する内容

単位・型

省略時のデフォルト値

親タグ system-valid-date-config

【属性】

属性名	説明	必須	デフォルト値
value	システム最小日の日付 (YYYY-MM-DD) です。	○	なし

システム最大日の設定

タグ名

システムで取り扱う最大日を設定します。

【設定項目】

```
<system-valid-date-config>
  <max-date value="3000-01-01" />
</system-valid-date-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	system-valid-date-config

【属性】

属性名	説明	デフォルト値	必須
value	システム最大日の日付 (YYYY-MM-DD) です。	○	なし

セキュア・トークンフィルタ設定

- 項目
- 概要
 - リファレンス
 - セキュア・トークン対象URL設定

概要

セキュア・トークンによるリクエスト正当性チェックを実施するための設定を行います。

ここで設定されたURLにアクセスされた場合、特定のリクエストパラメータの内部的には、正当性のチェックには `SecureTokenManager` を利用して行い、リクエストパラメータの付与は `<imSecureToken>` タグを利用します。

詳しくは、APIドキュメントの [スクリプト開発モデル・タグライブラリ imSecureToken](#) を参照してください。

正当だと判断された場合は、対象となるURLの内容が表示されます。正当で

 コラム

intra-mart Accel Platform 2014 Summer(Honoka) 以降、システムプロパティ `jp.co.intra_mart.foundation.secure_token` を利用します。運用中は利用しないでください。）

モジュール	テナント管理機能
フォーマットファイル	WEB-INF/schema/token-filtering-target-config(xsd)
設定場所	WEB-INF/conf/token-filtering-target-config/{tenantId}

```
<?xml version="1.0" encoding="UTF-8"?>
<token-filtering-target-config
  xmlns="http://www.intra-mart.jp/secure-token/target-url-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/secure-token/target-url-cc
  <entry url="/application/process/create" />
  <entry url="/application/process/update" />
  <entry url="/application/process/delete" />
</token-filtering-target-config>
```

リファレンス

セキュア・トークン対象URL設定

タ entry
グ
名

リクエスト時にセキュア・トークンによるチェックを行うパスを指定します。

【設定項目】

```
<token-filtering-target-config>
  <entry url="/application/process/create" />
  <entry url="/application/process/update" />
  <entry url="/application/process/delete" />
</token-filtering-target-config>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	token-filtering-target-config

【属性】

属性名	説明
url	チェック対象となるURLを指定します。 指定された文字列に完全一致するURLへアクセス時 クが実施されます。

システム管理者用グローバルナビ設定

- 項目
- 概要
 - リファレンス
 - メニュー設定

概要

システム管理者用の画面に表示するグローバルナビの項目を設定します。

モジュール	テナント管理機能
フォーマットファイル (xsd)	WEB-INF/schema/admin-global-navi-config.xsd
設定場所	WEB-INF/conf/admin-global-navi-config/{任意のファイル名}

```
<?xml version="1.0" encoding="UTF-8"?>
<admin-global-navi-config
  xmlns="http://www.example.org/im-admin-home"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/admin-global-navi-config
  <menu label="%CAP.Z.IWP.SYSTEM.HOME.CONSTRUCTION.TITLE"
    <children>
      <menu href="system/settings/license" label="%CAP.Z.IWP.SYST
      <menu href="system/settings/tenant" label="%CAP.Z.IWP.SYSTE
      <menu href="system/setup" label="%CAP.Z.IWP.SYSTEM.HOME.C
    </children>
  </menu>
  <menu label="%CAP.Z.IWP.SYSTEM.HOME.ADMINISTRATION.TITLE"
    <children>
      <menu href="master/maintenance/call/user" label="%CAP.Z.IWP.
      <menu href="system/service/module" label="%CAP.Z.IWP.SYSTI
      <menu href="system/service/status" label="%CAP.Z.IWP.SYSTEI
      <menu href="system/service/lock" label="%CAP.Z.IWP.SYSTEM.
      <menu href="system/storage/view" label="%CAP.Z.IWP.SYSTEM
      <menu href="system/database" label="%CAP.Z.IWP.SYSTEM.HC
      <menu href="portal/portlet/app/list" label="%CAP.Z.IWP.SYSTEM
      <menu href="portal/setting" label="%CAP.Z.IWP.SYSTEM.HOME
      <menu href="system/async/admin/task_queue_list" label="%CAP
      <menu href="system/settings/shared_database" label="%CAP.Z.I
    </children>
  </menu>
</admin-global-navi-config>
```

リファレンス

メニュー設定

タ menu
グ
名

メニューツリーの設定を行います。

【設定項目】

```
<menu label="%CAP.Z.IWP.SYSTEM.HOME.CONSTRUCTION.TITLE" i
  <children>
    <menu href="system/settings/license" label="%CAP.Z.IWP.SYSTEM
    <menu href="system/settings/tenant" label="%CAP.Z.IWP.SYSTEM
    <menu href="system/setup" label="%CAP.Z.IWP.SYSTEM.HOME.C
  </children>
</menu>
```

必須項 x
目

複数設 ○
定

設定 なし
値・設
定する
内容

単位・ なし
型

省略時 なし
のデ
フォルト
値

親タグ admin-global-navi-config

属性名	説明
label	メニューに表示するラベルを指定します。
href	メニューをクリックした際に遷移するURLを指
target	メニューをクリックした際のtargetを指定しま
id	このメニューのIDを指定します。複数ファイル 場合、そのメニューはマージされて表示され;
sort	メニューの要素のルートに同じIDが指定さ
children	メニューの子要素を指定します。子要素には 無効になります。

 コラム

label属性に%から始まる値を指定すると、MessageManager に

システム管理者用ホームウィジェット設定

項目

- 概要
- リファレンス
 - ホームウィジェット設定
 - パラメータ設定

概要

システム管理者のホーム画面に表示するウィジェットを設定します。

モジュール	テナント管理機能
フォーマットファイル (xsd)	WEB-INF/schema/admin-home-widget-config.
設定場所	WEB-INF/conf/admin-home-widget-config/{任:

```
<?xml version="1.0" encoding="UTF-8"?>
<admin-home-widget-config
  xmlns="http://www.intra-mart.jp/im-admin-home-widget-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/im-admin-home-widget-c
<widget url="system/home/widgets/status/status" captionUrl="system/l
<widget url="system/home/widgets/module/module" id="module" row=
<widget url="system/home/widgets/license/license" id="license" row="
<widget url="system/home/widgets/system-info/system-info" id="system
<widget url="system/home/widgets/bookmark/bookmark" id="bookmar
  <parameter name="bookmark" value="[
    {&quot;href&quot;:: &quot;http://www.intra-mart.jp/&quot;, &quot;ne
    {&quot;href&quot;:: &quot;http://www.intra-mart.jp/imart/partner.por
    {&quot;href&quot;:: &quot;http://www.intra-mart.jp/support/intramar
    {&quot;href&quot;:: &quot;http://imfaq.intra-mart.jp/imfaq/&quot;, &quot;
    {&quot;href&quot;:: &quot;http://www.intra-mart.jp/download/produ
    {&quot;href&quot;:: &quot;http://demo.intra-mart.jp/&quot;, &quot;n
    {&quot;href&quot;:: &quot;http://www.intra-mart.jp/apilist/&quot;, &quot;c
  ]"/>
</widget>
<widget url="system/home/widgets/node/node" captionUrl="system/ho
<widget url="system/home/widgets/tenant/tenant" id="tenant" row="2"
<widget url="system/home/widgets/memory/memory" id="memory" row
</admin-home-widget-config>
```

リファレンス

ホームウィジェット設定

タ widget
グ
名

ホーム画面のウィジェット設定を行います。

【設定項目】

```
<widget url="system/home/widgets/status/status" captionUrl="system/hom
```

必須項 x
目

複数設 ○
定

設定 なし
値・設
定する
内容

単位・ なし
型

省略時 なし
のデ
フォルト
値

親タグ admin-home-widget-config

【属性】

属性名	説明
id	このウィジェットのidを指定します。
url	このウィジェットの内容を返すURLを指定します。
captionUrl	このウィジェットのキャプションを返すURLを指定します。
row	このウィジェットが表示される行を指定します。
col	このウィジェットが表示される列を指定します。
sizeX	このウィジェットの横幅を指定します。
sizeY	このウィジェットの縦幅を指定します。
color	このウィジェットの背景色を指定します。
flip	このウィジェットにキャプションを表示するかどうかを指定し す。

パラメータ設定

タ parameter
グ
名

ウィジェットのパラメータ設定を行います。

【設定項目】

```
<parameter name="bookmark" value="[  
{&quot;href&quot;: &quot;http://www.intra-mart.jp/&quot;; &quot;name&  
}"/>
```

必 x
須
項
目

複 ○
数
設
定

設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	widget

【属性】

属性名	説明	デフォルト値	必須
name	パラメータの名前です。	○	なし
value	パラメータの値です。	○	なし

システム管理者用ユーティリティメニュー設定

- 項目
- 概要
 - リファレンス
 - ユーティリティメニュー設定

概要

システム管理者のユーティリティメニューの項目を設定します。

モジュール	テナント管理機能
フォーマットファイル(xsd)	WEB-INF/schema/admin-utility-config.xsd
設定場所	WEB-INF/conf/admin-utility-config/{任意のファイル名}

```
<?xml version="1.0" encoding="UTF-8"?>
<admin-utility-config xmlns="http://www.example.org/im-admin-utility-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.example.org/im-admin-utility-config http://www.example.org/im-admin-utility-config.xsd">
  <menu href="javascript:void(0);" label="%CAP.Z.IWP.SYSTEM.HOME">
    <children>
      <menu href="system/administrator/password" label="%CAP.Z.IWP.SYSTEM.ADMIN.PASSWORD">
      <menu href="system/administrator/locale" label="%CAP.Z.IWP.SYSTEM.ADMIN.LOCALE">
    </children>
  </menu>
  <menu href="logout" label="%CAP.Z.IWP.SYSTEM.COMMON.LOGOUT">
</admin-utility-config>
```

リファレンス

タ
グ
名

システム管理者のユーティリティメニュー設定を行います。

【設定項目】

```
<menu href="javascript:void(0);" label="%CAP.Z.IWP.SYSTEM.HOME.A
<children>
  <menu href="system/administrator/password" label="%CAP.Z.IWP.
  <menu href="system/administrator/locale" label="%CAP.Z.IWP.SYS
</children>
</menu>
```

必須項目	×
複数設定	○
設定値・設定する内容	なし
単位・型	なし
省略時のデフォルト値	なし
親タグ	admin-utility-config

【属性】

属性名	説明
label	メニューに表示するラベルを指定します。
href	メニューをクリックした際に遷移するURLを指定します。
target	メニューをクリックした際のtargetを指定します。
id	このメニューのIDを指定します。複数ファイルの場合、そのメニューはマージされて表示されます。
sort	メニューの要素のルートに同じIDが指定された場合、そのメニューはマージされて表示されます。
children	メニューの子要素を指定します。子要素には無効になります。

 コラム

label属性に%から始まる値を指定すると、MessageManager に

IM-Notice

IM-Notice 設定

項目
- 概要 - リファレンス - ベースURLの設定 - ショートカットURLの有効期間の設定

概要

通知機能全般に関する設定です。

モジュール	通知機能全般
-------	--------

フォーマット	WEB-INF/schema/im-notice-config.xsd
ファイル(xsd)	
設定場所	WEB-INF/conf/im-notice-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<im-notice-config
  xmlns="http://www.intra-mart.jp/notice/config/im-notice-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/notice/config/im-notice-c

  <base-url></base-url>
  <short-cut-duration-minutes>43200</short-cut-duration-minutes>
</im-notice-config>
```

リファレンス

ベースURLの設定

タグ名
base-url

intra-mart Accel Platform のベースURLを設定します。
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

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は「セットアップガイド」の「ベースURL」または、「テナント環境情報」から設

 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.NoticeRuntimeExce

【設定項目】

```
<base-url>http://example.org/imart</base-url>
```

必須項目	×
複数設定	×
設定値・設定する内容	クライアント（ブラウザ等）からアクセスを行うコン
単位・型	URI
省略時のデフォルト値	server-context-config.xmlで設定したベースUR
親タグ	im-notice-config

ショートカットURLの有効期間の設定

タグ名
short-cut-duration-minutes

ショートカットURLの有効期間を設定します。
クライアントがショートカットURLを使用しログインを行った後、有効期間が満

【設定項目】

```
<short-cut-duration-minutes>43200</short-cut-duration-minutes>
```

必須項目
○
複数設定
×

設定値・設定	ショートカットURLの有効期間を設定してください。
する内容	「0」を設定した場合、ショートカットURLは作成されませ
単位・型	数値 (分)(0-)
省略時のデフォルト値	なし
親タグ	im-notice-config

IM-Notice Mobile設定

項目
- 概要 - リファレンス - プロキシの設定 - ホストの設定 - ポート番号の設定 - ユーザ名の設定 - パスワードの設定 - ワークステーションの設定 - ドメインの設定 - Amazon SNSの設定 - エンドポイントの設定 - Amazon リソースネームの設定 - アクセスキーの設定 - シークレットキーの設定 - Google Cloud Messaging for Androidのプロジェクト番号の設定 - Baidu APIキーの設定 - Google Cloud Messaging for Androidの設定 - URLの設定 - APIキーの設定 - プロジェクト番号の設定 - 通知方法の設定 - 通知タイトルの最長文字数の設定 - 通知内容の最長文字数の設定

概要

モバイル通知機能に関する設定です。

モジュール	モバイル通知機能
フォーマットファイル(xsd)	WEB-INF/schema/im-notice-mobile-config.xsd
設定場所	WEB-INF/conf/im-notice-mobile-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<im-notice-mobile-config
  xmlns="http://www.intra-mart.jp/notice/mobile/config/im-notice-mobile-c
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/notice/mobile/config/im-n

  <proxy enable="false">
    <host></host>
    <port xsi:nil="true"></port>
    <username></username>
    <password></password>
    <workstation></workstation>
    <domain></domain>
  </proxy>

  <asns name="asns-apns">
    <endpoint>sns.ap-northeast-1.amazonaws.com</endpoint>
    <platform-application-arn></platform-application-arn>
    <access-key></access-key>
    <secret-key></secret-key>
  </asns>

  <asns name="asns-gcm">
    <endpoint>sns.ap-northeast-1.amazonaws.com</endpoint>
    <platform-application-arn></platform-application-arn>
    <access-key></access-key>
    <secret-key></secret-key>
    <gcm-project-number></gcm-project-number>
  </asns>

  <asns name="asns-baidu">
    <endpoint>sns.ap-northeast-1.amazonaws.com</endpoint>
    <platform-application-arn></platform-application-arn>
    <access-key></access-key>
    <secret-key></secret-key>
    <baidu-api-key></baidu-api-key>
  </asns>

  <gcm name="gcm">
    <url>https://android.googleapis.com/gcm/send</url>
    <api-key></api-key>
    <project-number></project-number>
  </gcm>

  <push type="android" engine="gcm" />
  <push type="ios" engine="asns-apns" />

  <max-push-subject-length>30</max-push-subject-length>
  <max-push-body-length>70</max-push-body-length>
</im-notice-mobile-config>
```

リファレンス

プロキシの設定

タ
グ
名

Amazon SNSやGoogle Cloud Messaging for Androidへ接続するための

【設定項目】

```
<im-notice-mobile-config>
  <proxy enable="false">
    . . . . .
  </proxy>
</im-notice-mobile-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	プロキシを使用する際に必要な設定をしてください。
単位・型	なし

省略時のデフォルト値	なし
親タグ	im-notice-mobile-config

【属性】

属性名	説明	必須
enable	プロキシを使用する場合、trueを設定してください。 プロキシを使用しない場合、falseを設定してください。	○

ホストの設定

タグ名	host
-----	------

プロキシサーバのIPアドレスを設定します。

【設定項目】

```
<proxy enable="true">
  <host>XXX.XXX.XXX.XXX</host>
</proxy>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	文字列
省略時のデフォルト値	なし
親タグ	proxy

ポート番号の設定

タグ名	port
-----	------

プロキシサーバのポート番号を設定します。

【設定項目】

```
<proxy enable="true">
  <port>XXXX</port>
</proxy>
```

必須 x
項目

複数 x
設定

設定 なし
値・
設定
する
内容

単 数値(0-)
位・
型

省略 なし
時の
デ
フォ
ルト
値

親タ proxy
グ

【属性】

属性名	説明	必
xsi:nil	ポート番号を指定する場合、この属性を削除してください。 ポート番号を指定しない場合、trueを設定してください。	x

ユーザ名の設定

タ username
グ
名

プロキシ使用時にユーザ認証が必要な場合、ユーザ名を設定します。

【設定項目】

```
<proxy enable="true">
  <username>XXXXXXXX</username>
</proxy>
```

必須 x
項目

複数 x
設定

設定 なし
値・
設定
する
内容

単位・型	文字列
省略時のデフォルト値	なし
タグ	proxy

パスワードの設定

タグ	password
名前	

プロキシ使用時にユーザ認証が必要な場合、パスワードを設定します。

【設定項目】

```
<proxy enable="true">
  <password>XXXXXXXX</password>
</proxy>
```

必須項目	x
複数設定	x
設定値・設定する内容	なし
単位・型	文字列
省略時のデフォルト値	なし
タグ	proxy

ワークステーションの設定

タグ	workstation
名前	

NTLM認証を行っている場合、ワークステーションを設定します。

【設定項目】

```
<proxy enable="true">
  <workstation>XXXXXXXX</workstation>
</proxy>
```

必須項目	x
複数設定	x
設定値・設定する内容	なし
単位・型	文字列
省略時のデフォルト値	なし
親タグ	proxy

ドメインの設定

タグ名	domain
-----	--------

NTLM認証を行っている場合、ドメインを設定します。

【設定項目】

```
<proxy enable="true">
  <domain>XXXXXXXX</domain>
</proxy>
```

必須項目	x
複数設定	x

設定値・設定する内容	なし
単位・型	文字列
省略時のデフォルト値	なし
親タグ	proxy

Amazon SNSの設定

タグ名	asns
-----	------

Amazon SNSを使用するための設定します。

【設定項目】

```
<im-notice-mobile-config>
  <asns name="asns-apns">
    .....
  </asns>
</im-notice-mobile-config>
```

必須項目	☐
複数設定	☐
設定値・設定する内容	Amazon SNSを使用するために必要な設定をしてください
単位・型	なし
省略時のデフォルト値	なし
親タグ	im-notice-mobile-config

【属性】

属性名	説明
name	任意の名前を設定してください。 pushタグのengine要素で通知方法を設定する際に使用

エンドポイントの設定

タグ名	endpoint
-----	----------

エンドポイントを設定します。

【設定項目】

```
<im-notice-mobile-config>
  <asns name="asns-apns">
    <endpoint>sns.ap-northeast-1.amazonaws.com</endpoint>
  </asns>
</im-notice-mobile-config>
```

必須項目	x
複数設定	x
設定値・設定する内 容	使用するリージョンに対応するAmazon SNSサ ー各リージョンのエンドポイントは AWS document
単位・型	文字列
省略時のデフォルト 値	空文字
親タグ	asns

Amazon リソースネームの設定

タグ platform-application-arn
名

Amazonリソースネームを設定します。

【設定項目】

```
<im-notice-mobile-config>
  <asns name="asns-apns">
    <platform-application-arn>arn:aws:sns:ap-northeast-1:XXXXXXXXX
  </asns>
</im-notice-mobile-config>
```

必須項目	x
複数設定	x
設定値・設定する 内容	Amazon SNSで作成したアプリケーションのApplice 内容
単位・型	文字列
省略時のデフォ ルト値	空文字
親タグ	asns

アクセスキーの設定

タ access-key
グ
名

アクセスキーを設定します。

【設定項目】

```
<im-notice-mobile-config>
  <asns name="asns-apns">
    <access-key>XXXXXXXXXXXXXXXXXXXX</access-key>
  </asns>
</im-notice-mobile-config>
```

必須項目	x
複数設定	x
設定値・設 定する内容	AWSで作成したアクセスキーを設定してください。

単位・型	文字列
省略時のデフォルト値	空文字
親タグ	asns

シークレットキーの設定

タ secret-key
グ
名

シークレットキーを設定します。

【設定項目】

```
<im-notice-mobile-config>
  <asns name="asns-apns">
    <secret-key>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</secret-key>
  </asns>
</im-notice-mobile-config>
```

必須項目	x
複数設定	x
設定値・設定する内容	AWSで作成したシークレットキーを設定してください。
単位・型	文字列
省略時のデフォルト値	空文字
親タグ	asns

Google Cloud Messaging for Androidのプロジェクト番号の設定

タグ名 gcma-project-number

Google Cloud Messaging for Androidのプロジェクト番号を設定します。

【設定項目】

```
<im-notice-mobile-config>
  <asns name="asns-gcm">
    <gcma-project-number>XXXXXXXXXXXX</gcma-project-number>
  </asns>
</im-notice-mobile-config>
```

必須項目	通知手段にAmazon SNS+Google Cloud Mes
複数設定	×
設定値・設定する内容	Google Cloud Messaging for Androidで作成し
単位・型	文字列
省略時のデフォルト値	なし
親タグ	asns

Baidu APIキーの設定

タグ名 baidu-api-key

Baidu APIキーを設定します。

【設定項目】

```
<im-notice-mobile-config>
  <asns name="asns-baidu">
    <baidu-api-key>XXXXXXXXXXXX</baidu-api-key>
  </asns>
</im-notice-mobile-config>
```

必須項目	通知手段にAmazon SNS+Baiduを利用する場合に必須
複数設定	×
設定値・設定する内容	BaiduクラウドプロジェクトのAPIキーを設定してください。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	asns

Google Cloud Messaging for Androidの設定

タグ名

Google Cloud Messaging for Androidを使用するための設定をします。

【設定項目】

```
<im-notice-mobile-config>
  <gcma name="gcma">
    .....
  </gcma>
</im-notice-mobile-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	Google Cloud Messaging for Androidを使用する
単位・型	なし
省略時のデフォルト値	なし
親タグ	im-notice-mobile-config

【属性】

属性名	説明
name	任意の名前を設定してください。 pushタグのengine要素で通知方法を設定する際に使用します。

URLの設定

タグ名

Google Cloud Messaging for AndroidのURLを設定します。

【設定項目】

```
<im-notice-mobile-config>
  <gcma name="gcma">
    <url>https://android.googleapis.com/gcm/send</url>
  </gcma>
</im-notice-mobile-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	「 https://android.googleapis.com/gcm/send 」を設定し る内容
単位・型	文字列
省略時のデフォルト値	空文字
親タグ	gcma

APIキーの設定

タグ名
api-key

APIキーを設定します。

【設定項目】

```
<im-notice-mobile-config>
  <gcma name="gcma">
    <api-key>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
  </gcma>
</im-notice-mobile-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	Google Cloud Messaging for Androidで作成し る内容
単位・型	文字列
省略時のデフォルト値	空文字
親タグ	gcma

プロジェクト番号の設定

タグ名
project-number

プロジェクト番号を設定します。

【設定項目】

```
<im-notice-mobile-config>
  <gcma name="gcma">
    <project-number>XXXXXXXXXXXX</project-number>
  </gcma>
</im-notice-mobile-config>
```

必須項目	×
複数設定	×

設定値・設定内容	GCMAで作成したプロジェクトのプロジェクト番号を設定する
単位・型	文字列
省略時のデフォルト値	空文字
親タグ	gcma

通知方法の設定

タグ名

各通知先へ通知を行う際に使用するサービスを設定します。

【設定項目】

```
<im-notice-mobile-config>
  <push type="android" engine="gcma" />
  <push type="ios" engine="asns-apns" />
</im-notice-mobile-config>
```

必須項目	☐
複数設定	☐
設定値・設定する内容	なし
単位・型	識別子
省略時のデフォルト値	なし
親タグ	im-notice-mobile-config

【属性】

属性名	説明
type	通知先 (android/ios) を設定してください。
engine	各通知先への通知方法 (asnsタグまたはgcmaタグ) を設定してください。

通知タイトルの最長文字数の設定

タグ名 max-push-subject-length

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ます。

【設定項目】

```
<im-notice-mobile-config>
  <max-push-subject-length>30</max-push-subject-length>
</im-notice-mobile-config>
```

必須項目	x
複数設定	x

設定値・設定する内容	Push通知受信時に表示するタイトルの最大文字数を設定する
単位・型	数値 (0-)
省略時のデフォルト値	空文字
親タグ	im-notice-mobile-config

通知内容の最長文字数の設定

タグ名	max-push-body-length
-----	----------------------

Push通知受信時に表示する本文の最長文字数を設定します。

【設定項目】

```
<im-notice-mobile-config>
  <max-push-body-length>70</max-push-body-length>
</im-notice-mobile-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	Push通知受信時に表示する本文の最長文字数を設定する
単位・型	数値 (0-)
省略時のデフォルト値	空文字
親タグ	im-notice-mobile-config

IM-Notice MQ設定

- 項目
- 概要
 - リファレンス
 - ポート番号の設定
 - ping機能の設定
 - エンドポイントの設定
 - I/Oスレッドの設定
 - ソケットの最大数の設定
 - リンガーの設定

概要

デスクトップ通知機能を使用するための設定です。

モジュール	デスクトップ通知機能
フォーマットファイル(xsd)	WEB-INF/schema/im-notice-mq-config.xsd
設定場所	WEB-INF/conf/im-notice-mq-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<im-notice-mq-config
  xmlns="http://www.intra-mart.jp/notice/mq/config/im-notice-mq-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/notice/mq/config/im-notice-mq-config"

  <address bind="true">tcp://*:40608</address>
  <ping enable="false" interval-seconds="0" time-to-live-seconds="0" />
  <endpoints>
    <endpoint>tcp://localhost:40608</endpoint>
  </endpoints>
  <zmq-io-threads>1</zmq-io-threads>
  <zmq-max-sockets>1024</zmq-max-sockets>
  <zmq-linger>1000</zmq-linger>

</im-notice-mq-config>
```

リファレンス

ポート番号の設定

タグ
名

サーバがクライアントとの通信を行う際に使用するポート番号を設定します。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を利用し、デフォルトの設定では40608ポートを使用します。

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。

【設定項目】

```
<im-notice-mq-config>
  <address bind="true">tcp://*:40608</address>
</im-notice-mq-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ポート番号を設定してください。 ブローカーを利用する場合、「tcp://{ブローカーホスト名}:ポート番号」で指定します。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	im-notice-mq-config

【属性】

属性名	説明
bind	スタンドアロン構成の場合、trueを設定してください。 ブローカーを利用する構成の場合、falseを設定してください。

ping機能の設定

タグ
名

ping機能に関する設定をします。

【設定項目】

```
<im-notice-mq-config>
  <ping enable="false" interval-seconds="0" time-to-live-seconds="0" />
</im-notice-mq-config>
```

必須項目	○
目	
複数設定	×
設定値・設定する内容	ping機能の設定をしてください。
単位・型	なし
省略時のデフォルト値	なし
親タグ	im-notice-mq-config

【属性】

属性名	説明
enable	ping機能の利用設定を行います。 利用する場合はtrue、利用しない場合はfalseを設定し
interval-seconds	クライアントにpingを流す間隔を数値(秒)で設定して
time-to live-seconds	クライアントがサーバに再接続を行うまでの時間を数い。

エンドポイントの設定

タグ名	endpoint
-----	----------

エンドポイントを設定します。
ブローカーを利用する場合ブローカーのホストまたはIPアドレスとブローカ

【設定項目】

```
<im-notice-mq-config>
  <endpoints>
    <endpoint>tcp://localhost:40608</endpoint>
  </endpoints>
</im-notice-mq-config>
```

必須項目	○
複数設定	○
設定値・設定する内容	クライアントの接続先を設定します。 addressタグで設定したポート番号を使 ブローカーを利用する場合、「tcp://{ブ
単位・型	文字列
省略時のデフォルト値	なし
親タグ	endpoints

I/Oスレッドの設定

タグ名	zmq-io-threads
-----	----------------

ZeroMQで使用されるI/Oスレッドの設定をします。
詳細は [ZMQ_IO_THREADS: Set number of I/O threads](#) を参照してください

【設定項目】

```
<im-notice-mq-config>
  <zmq-io-threads>1</zmq-io-threads>
</im-notice-mq-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	I/O操作を処理するためのZeroMQスレッドプールの
単位・型	数値
省略時のデフォルト値	なし
親タグ	im-notice-mq-config

ソケットの最大数の設定

タグ `zmq-max-sockets`

ZeroMQで使用されるソケットの最大数の設定をします。

【設定項目】

```
<im-notice-mq-config>
  <zmq-max-sockets>1024</zmq-max-sockets>
</im-notice-mq-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ZeroMQのコンテキストで許可されているソケットの最
単位・型	数値
省略時のデフォルト値	なし
親タグ	im-notice-mq-broker-config

リンガーの設定

タグ `zmq-linger`

リンガー期間を設定します。
クライアントのソケットが閉じられた場合、未送信のメッセージは設定された
詳細は [ZMQ_LINGER: Set linger period for socket shutdown](#) を参照し

【設定項目】

```
<im-notice-mq-config>
  <zmq-linger>1000</zmq-linger>
</im-notice-mq-config>
```

必須項目	○
------	---

複数設定	×
設定値・リンガー期間を設定してください。	
設定する内容	
単位・型	数値(秒)(0-)
省略時のデフォルト値	なし
親タグ	im-notice-mq-config

IM-Notice MQブローカー設定

項目
● 概要
● リファレンス
○ ポート番号(フロントエンド)の設定
○ ポート番号(バックエンド)の設定
○ ping機能の設定
○ I/Oスレッドの設定
○ ソケットの最大数の設定
○ リンガー(バックエンド)の設定
○ リンガー(フロントエンド)の設定

概要

ブローカーを含む構成でデスクトップ通知機能を使用するための設定です。

モジュール	デスクトップ通知機能
フォーマットファイル	WEB-INF/schema/im-notice-mq-broker-config.xsd(xsd)
設定場所	WEB-INF/conf/im-notice-mq-broker-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<im-notice-mq-broker-config
  xmlns="http://www.intra-mart.jp/notice/mq/broker/config/im-notice-mq-b"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/notice/mq/broker/config/i

  <frontend>tcp://*:40608</frontend>
  <backend>tcp://*:40609</backend>
  <ping enable="false" interval-seconds="0" />
  <zmq-io-threads>1</zmq-io-threads>
  <zmq-max-sockets>1024</zmq-max-sockets>
  <zmq-backend-linger>1000</zmq-backend-linger>
  <zmq-frontend-linger>1000</zmq-frontend-linger>

</im-notice-mq-broker-config>
```

リファレンス

ポート番号(フロントエンド)の設定

タグ名

ブローカーが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デフォルトの設定では40608ポートを使用します。

【設定項目】

```
<im-notice-mq-broker-config>
  <frontend>tcp://*:40608</frontend>
</im-notice-mq-broker-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ブローカーがクライアントとの通信を行う際に使用する内容
単位・型	文字列
省略時のデフォルト値	なし
親タグ	im-notice-mq-broker-config

ポート番号 (バックエンド) の設定

タグ名

ブローカーがWeb Application Server との通信を行うポート番号を設定しデフォルトの設定では40609ポートを使用します。

【設定項目】

```
<im-notice-mq-broker-config>
  <backend>tcp://*:40609</backend>
</im-notice-mq-broker-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ブローカーがサーバとの通信を行う際に使用するポート番号
単位・型	なし
省略時のデフォルト値	なし
親タグ	im-notice-mq-broker-config

ping機能の設定

タグ名

ping機能の設定をします。

【設定項目】

```
<im-notice-mq-broker-config>
  <ping enable="false" interval-seconds="0" />
</im-notice-mq-broker-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ping機能の設定をしてください。

単位・型	なし
省略時のデフォルト値	なし
親タグ	im-notice-mq-broker-config

【属性】

属性名	説明
enable	ping機能の利用設定を行います。 利用する場合はtrue、利用しない場合はfalseを設定して
interval-seconds	クライアントにpingを流す間隔を数値(秒)で設定してくた

I/Oスレッドの設定

タグ名
zmq-io-threads

ZeroMQで使用されるI/Oスレッドの設定をします。

詳細は [ZMQ_IO_THREADS: Set number of I/O threads](#) を参照してください

【設定項目】

```
<im-notice-mq-broker-config>
  <zmq-io-threads>1</zmq-io-threads>
</im-notice-mq-broker-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	I/O操作を処理するためのZeroMQスレッドプールの
単位・型	数値
省略時のデフォルト値	なし
親タグ	im-notice-mq-broker-config

ソケットの最大数の設定

タグ名
zmq-max-sockets

ZeroMQで使用されるソケットの最大数の設定をします。

【設定項目】

```
<im-notice-mq-broker-config>
  <zmq-max-sockets>1024</zmq-max-sockets>
</im-notice-mq-broker-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	ZeroMQのコンテキストで許可されているソケットの最
単位・型	数値
省略時のデフォルト値	なし

親タグ im-notice-mq-broker-config

リンガー(バックエンド)の設定

タグ zmq-backend-linger
グ
名

サーバ側のリンガー期間を設定します。
サーバのソケットが閉じられた場合、未送信のメッセージは設定された期間
詳細は [ZMQ_LINGER: Set linger period for socket shutdown](#) を参照し

【設定項目】

```
<im-notice-mq-broker-config>  
  <zmq-backend-linger>1000</zmq-backend-linger>  
</im-notice-mq-broker-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	リンガー期間を設定してください。
単位・型	数値(秒)(0-)
省略時のデフォルト値	なし
親タグ	im-notice-mq-broker-config

リンガー(フロントエンド)の設定

タグ zmq-frontend-linger
グ
名

クライアント側のリンガー期間を設定します。
クライアントのソケットが閉じられた場合、未送信のメッセージは設定された
詳細は [ZMQ_LINGER : Set linger period for socket shutdown](#) を参照し

【設定項目】

```
<im-notice-mq-broker-config>  
  <zmq-frontend-linger>1000</zmq-frontend-linger>  
</im-notice-mq-broker-config>
```

必須項目	○
複数設定	×
設定値・設定する内容	リンガー期間を設定してください。
単位・型	数値(秒)(0-)
省略時のデフォルト値	なし
親タグ	im-notice-mq-broker-config
