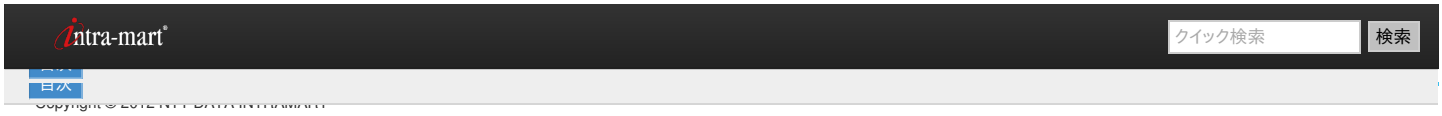

CORPORATION

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
  - サービスセレクタ設定
  - initializer 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ブ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 認可リソースマッパー定義設定
  - 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
  - IPアドレス取得元設定
  - 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
  - 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
- 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
- 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
- IM-ContentsSearch
  - 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
- ログ

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

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)	なし
設定場所	WEB-INF/resin-web.xml

```
<web-app xmlns="http://caucho.com/ns/resin" xmlns:resin="urn:java:com.caucho
  <log-handler name="" class="jp.co.intra_mart.common.platform.log.handler.JD
  <!-- im_service(im_asynchronous) -->
  <resource jndi-name="jca/work" type="jp.co.intra_mart.system.asynchronous.ir
  <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

- 互換モジュール
  - 汎用設定ファイル

索引

タグ名	log-handler

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.handler.JD
</web-app>
```

必須項目	×
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

タグ名	resource

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.asynchronous.ir
</web-app>
```

必須項目	×
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

【属性】

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 ■ 「*サービスセレクタ設定* 」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 ■ 「*サー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

JSPに関する設定です。

【設定項目】

```
<web-app>
  <jsp>
    ...
  </jsp>
</web-app>
```

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	recycle-tags

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】

```
<web-app>
  <jsp>
    <recycle-tags>false</recycle-tags>
  </jsp>
</web-app>
```

必須項目	×	
複数設定	×	
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)	

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

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	デフォルト値
jndi-name	このデータベース設定のJNDI名を設定します。	○	なし

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>
```

必須項目	○
複数設定	×

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グイングループ設定」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」に関数説明を更新 ■ 「*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
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

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	×
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

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。

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
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。パスワードの暗号

```
C:\resin-pro-4.x.xx>resin password-encrypt <パスワード>
password: {RESIN}WFpCuWyYSFzuOK1Gda0TxQ==
```

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必要が

```
[resin-pro-4.x.xx]# bin/resinctl password-encrypt <パスワード>
password: {RESIN}WFpCuWyYSFzuOK1Gda0TxQ==
```

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いる必

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

【設定項目】

```
<web-app xmlns="http://caucho.com/ns/resin" xmlns:resin="urn:java:com.caucho
  ...
  <database jndi-name="jdbc/default">
    <driver>
      ...
      <password>
        <resin:Password value="{RESIN}WFpCuWyYSFzuOK1Gda0TxQ==" />
      </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

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	×
複数設定	×
設定値・設定する内容	コネクションの最大数
単位・型	数値
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

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はメモ
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、レ
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
intra-mart Accel Platform では、機能によりSQLの発行や内容が回数が異なりま
その為、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仕組み
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設定する場合、128等あ

【設定項目】

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

必須項目	×
複数設定	×
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

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	×
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継ぎま
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、その時に異なるセッショ

【設定項目】

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

必須項目	×
複数設定	×
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す。

【設定項目】

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

必須項目	×
複数設定	×

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定をしま
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分間のずれが
> 例えば、セッションタイムアウトを30分に設定した場合、セッションタイムア

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。ヘルス機
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、Resi
が可能となります。

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を定
また、「メーター」とWeb管理コンソールとなる[/resin-admin]を設定する事により、/r

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。ユーザはクラス
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、ヒープ機

/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必要が

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

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

   Windowsの例

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

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
   admin_user : admin
   admin_password : {SSHA}HTfP60Ceq0K0IAvN31wQtBxtql9D+8Wo
   ```

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ファイル

   ```
   # Permit access to /resin-admin from non-local network ip-addresses
   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 にアクセスしてく
   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 /proc/net
> Windowsでは使用できません。

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てい

- **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、ヒープダンプの取得、
- **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て、システムに異常が発生した場合にどのようなタスク

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
- **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
- **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プール
- **<health:CpuHealthCheck>** - CPU使用率が異常に上昇していないか
- **<health:HealthSystemHealthCheck>** - ヘルスシステムが正常に機能してい
- **<health:HeartbeatHealthCheck>** - クラスタのハートビートが正常に機能して

- **<health:JvmDeadlockHealthCheck>** - デッドロックが発生していないか
- **<health:MemoryPermGenHealthCheck>** - JavaVMのPermGen領域が正常
- **<health:MemoryTenuredHealthCheck>** - JavaVMのTenured領域が正常か
- **<health:TransactionHealthCheck>** - コミットの失敗はないか

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

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うにな

```
<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、様々な情報を取集した
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」を返
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待ち時
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

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

> 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ん。
> PDFレポートの「CPU Profile」には「A CPU profile was not generated

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」を返した場合

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状態であ
- <health:JvmDeadlockHealthCheck> - JavaVMによってデッドロック状態である

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。この要素はデフォルト値

属性

属性名	説明	型
enabled	ヘルスシステムの有効/無効。	bool
startup-delay	サーバ起動時にチェックの開始を遅延させる時間	*期間*
period	チェックの間隔。	*期間*
recheck-period	再チェックの間隔。	*期間*
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。	int

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、環境によっては負

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す。
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、または適切に実です。
OK	1	ヘルスチェックは正常な状態であることを報告しました。
WARNING	2	ヘルスチェックは警告の域に達しているか、重要なことが
CRITICAL	3	ヘルスチェックは重要の域に達していることを報告しまし
FATAL	4	ヘルスチェックは致命的な状態であることを報告しました

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

設定例

```
<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が個別にチェックされます

属性

属性名	説明	型
enabled	チェックの有無	boole
warning-threshold	ヘルスステータス「WARNING」のしきい値	int(ハ 0-10
critical-threshold	ヘルスステータス「CRITICAL」のしきい値	int(ハ 0-10

状態

ヘルスステータス	状態
WARNING	CPUのどれかが *warning-threshold* を超過した場合。
CRITICAL	CPUのどれかが *critical-threshold* 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、または時間がた

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行えな 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 *freeze-timeout* の期間内に行わ
FATAL	ヘルスチェックが許容される期間内に完了しなかっ 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信されて
WARNING	クラスタの既知のメンバーからのハートビートが最後に受信 合。

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 *warning-period* の期間内に期限切れになる場合。

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。空き領域が低下しすき

> 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ない場
> 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

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最小量
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 MBeanを検索します。

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。また 合。
WARNING	空きメモリが、ガベージコレクション実行後に *free-warning*
CRITICAL	空きメモリが、ガベージコレクション実行後に *memory-free*

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。空き領域が低下しすぎ

> コラム
>
> JavaVMにTenuredメモリプールが存在しない場合はJavaVMでヒープ

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最小量
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 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。または 合。
WARNING	空きメモリが、ガベージコレクション実行後に *free-warning*
CRITICAL	空きメモリが、ガベージコレクション実行後に *memory-free*

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のリクエストを行

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（*url* が URI の場合に使用しま
ping-port	チェック対象のポート（*url* が URI の場合に使用しま
url	チェック対象のURLまたはURI
socket-timeout	ソケットコネクションのタイムアウト時間
regexp	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。またはHTT 合。

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>
  <regexp>^2|^3</regexp>
</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	boolean
fatal-test	trueと評価された場合にステータスをFATALするEL式。	EL式
critical-test	trueと評価された場合にステータスをCRITICALするEL式。	EL式
warning-test	trueと評価された場合にステータスをWARNINGするEL式。	EL式

状態

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

設定例

```
<health:ExprHealthCheck>
  <critical-test>${mbean('java.lang:type=Threading').ThreadCount > 100}</critic
</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の状態
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ではあ
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されます

<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	型
objectName	JMX MBeanの名前。	Strin
operation	メソッドの名前。	Strin
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。	int
param	適切な型に変換されるメソッドのパラメータ。	boole

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。ログ

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フォーマットの
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す。
log	ヒープダンプをResinの内部データベースに格納されるのに加みます。

設定例

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

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るのに加えてに書き込みます。

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。ロ

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。(**RUNNABLE** の状態
log	スレッドダンプをResinの内部データベースに格納されるのに加書き込みます。

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。*dir* が設定されている場合は、相対パス
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、コマント
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めにAc

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	${resin.ho gen.php
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</report>
  <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

属性

属性名	説明	型	デ ル
log	PDFレポートに加えて、サーバ·ログに書き込みます。	boolean	fa
type	スコアボードレポートのタイプ。	String	re
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	tr

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クするために必要な全ての情報を想定しており、シ
このアクションは通常、予期しない問題に対して実行されることを想定しています。

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ルに記

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実行す
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限り、
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。複数のコンディション

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、その日の時間のような

<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どを抑

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、コンフィグプロ

属性

属性名	説明	型	デフォルト
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。これは頻繁な実行が必要の

属性

属性名	説明	型	デフォルト
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 のパラメータ

属性

なし

設定例

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

<health:IfUptime>

起動してからの時間の長さに応じてマッチします。

属性

属性名	説明	型	デフォルト
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無効に

<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できま

<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。

属性名	説明
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。
regexp	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}" regexp="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」、「WARNIN

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のライフサイクル状態

<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。これは一般的にエラ

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
  - ヘルスメーター名
- JMXメーター

- <health:JmxMeter>
- <health:JmxDeltaMeter>
- 統計分析
- <health:AnomalyAnalyzer>
- <health:IfHealthEvent>

ヘルスメーターはResinの管理コンソール(/resin-admin)に視覚的にわかり易くグラ

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の詳細な

ヘルスメーター名

ヘルスメ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。
ヘルスメーターの統計はResinクラスタの各メンバー間で共有されているため、クラ

名前に含まれるパイプ文字は、ヘルスメーターをドリルダウンに分類して管理コンソ

```
<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** をベースにヘルスメーター名
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つのキーを使用し、

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

JMXメーター

JMX MBean属性の数値をヘルスメ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	デフルト
name	管理コンソールに表示するヘルスメーターの名前	String	なし
objectName	JMX MBean の名前	String	なし
attribute	サンプリングするMBeanの属性	String	なし

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

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す。

属性

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

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、<health-eve

属性

属性名	説明	型	デフォルト
regexp	*<health-event>* が一致する正規表現。	java.util.regex.Pattern	なし（必須）

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。pdf-re

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。また、再

```
<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ルアド

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」レベルのメッセ
- **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に設計されています。

- **Resinのバグレポート** - Resin自体のサポートのために、スナップショットがあれば
- **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場合、wat
- **CPUとパフォーマンス** - JavaVMが予想外にCPUを使用した場合、レポートで主
- **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き止める

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。異常なメモリ配分の

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」の順に

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の内部プロキシーキャッシュと分散キャッシ
きな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「self」

クラスローダーのヒープダンプ

ヒープダンプには、クラスローダのメモリ使用量を個別に表示する項目があります。
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ことが

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ト
いくつもの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から

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マージし

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るので

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つのスレッドの
この場合「**http://*:8444-17**」という名前のスレッドがSocksSocketImplを所有して
11」）

```
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://*:8444-1
http://*:8444-10
  waiting on java.net.SocksSocketImpl@4782b18d owned by [126] http://*:8444-1
http://*:8444-11
  waiting on java.net.SocksSocketImpl@4782b18d owned by [126] http://*:8444-1
   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ません。
PDFレポートの「CPU Profile」には「A CPU profile was not generated during

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プロセスによってF
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
全ての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発生しw

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は自動的に
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しいRe

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。実行環境

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。管理するResin

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Resinインス

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 オーバーのを無効
    - 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アプリケーションの

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、ログフォーマ
ログ設定は、ログハンドラとロガーレベルの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、ファイル
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> は名前付きロガーのログレベルを設定します。<logger> は一般的に複数
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「com.fo

「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クラス
す。
例えば、<logger name="example.hogwarts" ...> は "example.hogwarts.Systen

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の便利

名前	意味
""	すべてをデバックします。
com.caucho.ejb	EJBの処理。
com.caucho.jsp	JSPのデバック。
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」レベル
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す。

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起動の失
実行を妨げる障害。
warning	log.warning("...")	ブラウザにレスポンスコード500を
info	log.info("...")	Webアプリケーションの起動など
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ない開発
finer	log.finer("...")	開発者が使用する詳細なデバッグ
finest	log.finest("...")	詳細なトレースを行うデバック。
all		全てのメッセージがログに記録さ

<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、各ロ
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ファ

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「%」のコードを含めるこ

コード	意味
%a	曜日（省略）
%A	曜日（詳細）
%b	月（省略）
%B	月（詳細）
%c	Javaロケールの日付
%d	月（数字2桁）

コード	意味
%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] {%{thread}}"/>
```

```
[22:50:11.648] WebApp[/doc] starting
[22:50:11.698] http listening to *:8080
[22:50:11.828] hmux 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。ハンド

```
<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にな

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

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クログの出力を有効に

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

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文字列

フォーマット文字列の例

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

ログ EL 変数

変数	説明
${log.level}	ログレベル。
${log.name}	ロガー名。
${log.shortName}	ロガー名を短くしたもの（”co はなく “Foo”）
${log.message}	ログメッセージ。
${log.millis}	イベントが発生した時間。19
${log.sourceClassName}	ロギングを要求したクラス名 ない場合があります。）
${log.sourceMethodName}	ロギングを要求したメソッド名 ない場合があります。）
${log.threadID}	ロギングの要求が発信され 別子を取得。
${log.thrown}	ロギングの要求に関連付け java.lang.Throwableを取得
${thread}	現在のスレッドの名前。
${request}	サーブレットリクエストの値。
${session}	HTTPセッション。
${cookie['JSESSIONID']}	リクエスト クッキーの値。

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.%s]"
      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/caucho/au
[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とがで
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。別の名前を付けること

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、以下のレ

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の構文を使って設定で

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>
  </logger>

</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。フォーマッタは

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たい場合

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>
```

MyFormatter.java

```
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、HMTPおよび syslog サービスなどに一般的なログパターンでメッセ
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。エージェントは、
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。LogEve
ラのクラス名は 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。メールの数を抑え

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	1m（1分）
mail-interval-min	メールメッセージの最小間隔	1h（1時間）
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ることが

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、cron、auth

severityには、emerg、alert、crit、err、warning、notice、info、debugが使用可能です

「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。ロール

ResinのログローテーションはJDK logging、HTTPアクセスログ、標準入出力ログて

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。Resinのデフ

「rollover-size」は最大サイズを指定するために使用し、バイト（50000）、キロバイト

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バーされ
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バーが

「rollover-period」は期間を指定するために使用し、日（15D）、週（2W）、月（1M）、

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）新しいロ

「archive-format」は、アーカイブファイルの名前を指定します。これは、通常の文字、
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」が一日
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パスに

ロールオーバーの無効

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きな数を

```
<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。アーカイ

```
<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>の子としてst
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てはい

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。構文は「archi
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）、時間
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（50000）、K（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>の出力ログ設定がオー

```
<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>の子としてst
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てはい

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。構文は「archi
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）、時間
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（50000）、K（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>の出力ログ設定がオー

```
<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ます。<h

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	レスポンスヘッド。
%{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。

パターン	説明
%T	リクエストの処理が完了するまでにかかった時間。(秒)
%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}i\" \"%{User-Agent}i\""
```

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。構文は「
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
auto-flush	要求のたびにメモリバッファーをフラッシュする場合は true
auto-flush-time	メモリ バッファー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 ログに記録さ
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。(パフォーマンスに影
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ログファイルの最大サイズ。バイト(S す。
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 アクセス ロ
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ように設

パス	説明
ファイルシステムのパス	ファイルに出力します。
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行うため

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-agent-config
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://intra-mart.co.jp/system/service/client/config/networ
  <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

タグ名	id

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・設定する内容	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、予期しないエラ
> 必ず、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

タグ名	bind-port

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ます。

【設定項目】

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

必須項目	○
複数設定	×
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

タグ名	port-range

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て利用
例えば、bind-portに5200、port-rangeに2が設定されており、5200番ポートが既に

【設定項目】

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

必須項目	○
複数設定	×
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

タグ名	preferIPv4Stack

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、必ずtr

ユニキャスト設定

タグ名	TCP

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	UDP

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、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ます。
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を行い

【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	○
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-config"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://intra-mart.co.jp/system/platform/config/server-conte
  <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

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クトリに

【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 が起動しなくなる

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る際に利

【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

タグ名	base-url

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を生成する際等に利用されます。
Web Server と Web Application Server でURLが異なる場合等に、クライアント(

【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テキストパ
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

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/config
  <charset>UTF-8</charset>
  <class-name>jp.co.intra_mart.system.service.client.file.StorageLocalFileImpl</
  <unable-charactors>
    <unable-charactor>\</unable-charactor>
    <unable-charactor>:</unable-charactor>
    <unable-charactor>*</unable-charactor>
    <unable-charactor>?</unable-charactor>
    <unable-charactor>&quot;</unable-charactor>
    <unable-charactor>&lt;</unable-charactor>
    <unable-charactor>&gt;</unable-charactor>
    <unable-charactor>|</unable-charactor>
  </unable-charactors>
  <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>
    <conpatible-directory-name>public</conpatible-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</synchronize-key>
  </synchronize-info>
</storage-config>
```

リファレンス

リファレンス

文字コード設定

タグ名	charset

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のみと

ストレージAPI実装クラス設定

タグ名	class-name

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

必須項目	○
複数設定	×
設定値・設定する内容	ストレージ対する読み込み、書き込み等の実装ク 指定されるクラスは、jp.co.intra_mart.foundati 標準で提供されている実装はローカルファイルI ています。 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もので

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-charactor>\</unable-charactor>
    <unable-charactor>:</unable-charactor>
    <unable-charactor>*</unable-charactor>
    <unable-charactor>?</unable-charactor>
    <unable-charactor>&quot;</unable-charactor>
    <unable-charactor>&lt;</unable-charactor>
    <unable-charactor>&gt;</unable-charactor>
    <unable-charactor>|</unable-charactor>
  </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設定しま
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-charactor

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。

【設定項目】

```
<storage-config>
  ...
  <unable-charactors>
    <unable-charactor>\</unable-charactor>
    <unable-charactor>:</unable-charactor>
    <unable-charactor>*</unable-charactor>
    <unable-charactor>?</unable-charactor>
    <unable-charactor>&quot;</unable-charactor>
    <unable-charactor>&lt;</unable-charactor>
    <unable-charactor>&gt;</unable-charactor>
    <unable-charactor>|</unable-charactor>
  </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します。 標準で禁止している文字は : * ? " < > | が設定されています。
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用でき

ストレージ設定

タグ名	storage-info

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

タグ名	root-path-name

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。
${システムプロパティのキー} 形式で記述を行うことによりシステムプロパティの値
例えば、Resinを利用した場合は、Resinがインストール（展開）された位置を${resi
その為 ${resin.home}/storage と記述した場合はResinがインストール（展開）され
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していしま
/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くださ

【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を指定
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に作成

【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します。
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,
> この設定は、intra-mart Accel Platform において利用される内部項目で

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に作成

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ディレクト
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に有効
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、そちらが有効にな

【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	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-directory-name
> この設定は、intra-mart Accel Platform において利用される内部項目で

グループストレージディレクトリ名設定

タグ名	group-directory-name

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に作成

【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	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-directory-name
> グループストレージは過去の互換性および、将来的な利用の為の予約項
> この設定は、intra-mart Accel Platform において利用される内部項目で

設定ファイル配置用ディレクトリ名設定

タグ名	configuration-directory-name

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に作成

【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	×
設定値・設定する内容	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します。

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-directory-name
設定ファイル配置用ディレクトリ名設定は、将来的な利用の為の予約項目
この設定は、intra-mart Accel Platform において利用される内部項目で

互換モジュール用ディレクトリ名設定

タグ名	compatible-directory-name

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に作成

【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	×
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します。
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スト
互換モジュール用ディレクトリ名設定は、将来的な利用の為の予約項目
この設定は、intra-mart Accel Platform において利用される内部項目で

ストレージディレクトリ名設定

タグ名	storage-directory-name

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領域内で
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、テンポラリス

【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	storage-info

> **注意**
>
> この設定は、intra-mart Accel Platform において利用される内部項目で

テンポラリストレージディレクトリ名設定

タグ名	temporary-directory-name

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に作成
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されま

【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	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-directory-name
> この設定は、intra-mart Accel Platform において利用される内部項目で

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す。
この項目は将来的な利用の為の予約項目です。

【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

サービスセレクタ設定

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設定しま

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-selecter-config.xsd
設定場所	WEB-INF/conf/service-selecter-config/service-selecter-

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

リファレンス

サービス設定

タグ名	services

各サービスの選出方法を設定します。

【設定項目】

```
<server-selecter-config>
 <services>
   <service service-id="XXXXX" />
 </services>
</server-context-config>
```

必須項目	○
複数設定	×

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-selecter-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。(複数起動可 ります。)

コラム

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

- ServerManager：「server.service.controller」
- QueueService：「server.service.queue.management」
- TaskService：「server.service.task.management」
- JobSchedulerService：「server.service.job_scheduler」

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選出し
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サー

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を起動
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サー

【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	invalid-hosts

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しませ

【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	host

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を指定

【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	サーバアドレスを指定します。
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	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(.)単位で区切られた範囲内
  > - 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

- 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「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(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を指しま
> 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

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

 <java-script-api>
  <global-function-class>sample.common_libs.global_fnction.GlobalFnction#s
  <global-function-script>sample/common_libs/global_fnction#global_fnction<
  <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</tag-clas
  <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	java-script-api

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

 <java-script-api>
  <global-function-class>sample.common_libs.global_fnction.GlobalFnction#s
  <global-function-script>sample/common_libs/global_fnction#global_fnction<
  <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	グローバル関数を実装したJava 行関数
	global-function-script	グローバル関数を定義する js フ 関数
	api-class	オブジェクトを定義した Javaク
	api-script	オブジェクトを定義した js ファイ 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

タグ名	global-function-class

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.GlobalFnction#sa
</java-script-api>
```

必須項目	×
複数設定	○
設定値・設定する内容	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

タグ名	global-function-script

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

【設定項目】

```
<java-script-api>
  <global-function-script>sample/common_libs/global_fnction#global_fnction1<
</java-script-api>
```

必須項目	×
複数設定	○
設定値・設定する内容	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

タグ名	api-class

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・設定する内容	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/initializer-
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://intra-mart.co.jp/system/secure/product/initializer/co

  <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-class	IMARTタグを定義したJavaクラスのパス
	tag-script	IMARTタグを定義js ファイルのパス#実行関数
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	tag-class

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	×
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	tag-script

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	×
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	initializer

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

  <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

タグ名	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.ImSampleTag<
</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	デフォルト値
MBean	MBeanServerに登録するかどうかを指定します。	×	false

起動時初期化js ファイルのパス

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

必須項目	×

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

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}.xml

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

リファレンス

クライアントタイプの設定

タグ名	client-type-info

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_type.Client
</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.multi_de 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・デフォルトのクライアント

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

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

概要

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定しま
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 (xsd)	WEB-INF/schema/date-time-format-config.xsd
設定場所	WEB-INF/conf/date-time-format-config/{任意のファイル名

```
<?xml version="1.0" encoding="UTF-8"?>
<date-time-format-config
  xmlns="http://www.intra-mart.jp/system/i18n/datetime/date-time-format-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/system/i18n/datetime/date-time-fo
  <format-set default="true">
    <format-set-id> IM_DATETIME_FORMAT_SET_EN_BASE </format-set-id>
```

```
  <format-set-id>IM_DATETIME_FORMAT_SET_EN_BASE</format-set-id>
  <locale>en</locale>
  <format type="date">
    <format-id>IM_DATETIME_FORMAT_DATE_STANDARD</format-id>
    <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-id>
    <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-id>
    <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-set-id>
  <locale>ja</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 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>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

タグ名	format-set

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	必須
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

タグ名	format-set-id

「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・設定する内容	「日付と時刻の形式」の識別子
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・設定する内容	ロケールID
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	format-set

【属性】

属性名	説明		必須	デフォルト値
type	date	日付用のフォーマットとして扱われます。	○	なし
	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

タグ名	format-id

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-id>
      .....
    </format>
  </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	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	デフォルト値
default	true の場合、初期状態におけるデフォルト値となります。	×	false

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-co
  <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	デフォルト値
name	ロケール ID を設定します。	○	なし
default	true の場合、システム・デフォルトのロケールになります。	×	false

> **注意**
>
> 運用開始後の変更は推奨しません。運用開始前に利用するロケールを決
>
> default="true" のロケールが複数ある場合、設定ファイルの先頭から検索
> default="true" のロケールが見つからない場合、設定ファイルで先頭の

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}.xml

```
<time-zone-config
  xmlns="http://www.intra_mart.co.jp/system/i18n/timezone/config/time-zone-conf
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra_mart.co.jp/system/i18n/timezone/config/ti
  <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 によって定義

【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・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 ID で

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

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に関す

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ブディレク
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いる任意の

```
<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

文字エンコーディングの設定

タグ名	charset

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 Platform に

【設定項目】

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

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 の設定値
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の文字
> 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	javascript

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	×
複数設定	×
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	compiler

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	×
複数設定	×
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。 • *true* を設定した場合、JavaScriptをJavaクラスへコン 上にキャッシュし、以後のプログラム実行は、すべてク ソースの変更が一切反映されなくなります（ソースのん）。この設定は、運用環境向き設定です。 • *false* を設定した場合、JavaScriptをインタプリタモー す。開発をスムーズに進めることができますが、常にソ きません。この設定は、開発環境向き設定です。 IM-JugglingのビルドウィザードからWARファイルを生成 **テスト環境**」を選択していると、souorce-config.xmlの設 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	×
複数設定	×
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最適化レ きくなります。0を設定した場合は、最適化はされません。 プログラムを最適化して実行する事は、パフォーマンスの かつ複雑な機能な為、プログラムの内容によって通常て 最適化レベルを適用する場合は、その適用範囲を必要

> **注意**
>
> intra-mart Accel Platform および関連するアプリケーション製品は全て最
> 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	view

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	compiler

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・設定する内容	View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	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。 • *true* を設定した場合、HTMLをコンパイルして実行し キャッシュされ、以後のプログラム実行は、すべて中間 が、ソースの変更が一切反映されなくなります（ソース ん）。この設定は、運用環境向き設定です。 • *false* を設定した場合、HTMLをインタプリタモードで 発をスムーズに進めることができますが、常にソースを せん。この設定は、開発環境向き設定です。 IM-JugglingのビルドウィザードからWARファイルを生成 **テスト環境**」を選択していると、souorce-config.xmlの設 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

タグ名	escapeXml

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です。

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・設定する内容	XML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	XMLエスケープの有効・無効を指定します。
exclusion	XMLエスケープ処理の対象外とする文字列を指定します
delimiter4exclusion	XMLエスケープ処理の対象外とする文字列のデリミタ指

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エスケープを行う設定

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

必須項目	×
複数設定	×
設定値・設定する内容	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

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

ID属性を自動的に付加する設定

タグ名	replicateNameToId

<imart type="hidden"> タグ使用時に、name属性に指定された値をid属性に付加
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下のよう

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

【設定項目】

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

必須項目	×
複数設定	×
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	• *true* を設定した場合、name属性に設定された値をid属性に • *false* 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に対して
replicateNameToId は対応しているプログラムソースにのみ使用するよ

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 Server に

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.xsd
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
  xmlns="http://intra-mart.co.jp/system/database/config/data-source-mapping-co
  xsi:schemaLocation="http://intra-mart.co.jp/system/database/config/data-sourc
 <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>xxxxxxxxxxxxx</resource-ref-name>
  <database-name>xxxxxxxxxxxxx</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て利用するデ
	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	×	
複数設定	○	
設定値・設定する内容	connect-id	シェアードデータベースの接続ID
	resource-ref-name	シェアードデータベースとして利用する
	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

タグ名	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	×	
複数設定	○	
設定値・設定する内容	tenant-id	このデータベース設定を利用するテナン
	resource-ref-name	テナントデータベースとして利用するデ
	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

タグ名	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です。

【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リソース参
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-data-sourc

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ベースの

【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自動的に判
親タグ	system-data-source, shared-data-source, tenant-data-sour

> **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異なる場

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>xxxxxxxxxxxxx</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 (xxxxxxxxxxxxx)
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 (xxxxxxxxxxxxx)
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
  - メールヘッダ設定
  - 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.xml WEB-INF/conf/javamail-config/{任意のファイル名}.xml

```
<?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/javama

 <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.HalfKar
 </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

タグ名	smtp-server

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	×
複数設定	×

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IDと同
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ため、必

【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です。 APIからのメール送信時にこのIDを指定することで、 にIDを指定しなかった場合、テナントIDと同じIDのSM 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アドレスを指定しま
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

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証明書を信頼済みて
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て使用

【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	×
複数設定	×
設定値・設定する内容	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による暗号化通信を利用し 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。TLS/STA falseを設定してください。 この設定を有効にする場合はenable属性も有効に設定し

SMTP Authentication設定

タグ名	auth

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。SMTP Au 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

メールヘッダ設定

タグ名	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します。
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

タグ名	debug

JavaMailにデバック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

コンテンツタイプ群

タグ名 content-type

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を指定しま
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	file

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	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さい。
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タイプを記ます。

リスナ群設定

タグ名	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	×
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

タグ名	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.MailSendL
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_mart.found

【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ドが定

【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・設定する内容	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。
親タグ	javamail-config

【属性】

属性名	説明	必須	デフォルト値
locale	ロケールIDを指定します。	×	なし

文字コード設定

タグ名	charset

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・設定する内容	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

タグ名	mime-encoding

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・設定する内容	MIMEエン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

ボディエンコード設定

タグ名	content-transfer-encoding

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・設定する内容	ボディのエン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

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。
リクエストのを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を提

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

```
<?xml version="1.0" encoding="UTF-8"?>
<request-control-config
   xmlns="http://intra-mart.co.jp/system/servlet/filter/xml/request_control/request-c
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://intra-mart.co.jp/system/servlet/filter/xml/request_co
 <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	thread

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関しては
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	×
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	inspection

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は無効

【設定項目】

```
<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	なし
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	デフォルト値
keep-alive	処理中のリクエストの有効期間を指定します。	○	なし

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クエリの

【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	デフォルト値
queue	処理待ちを行うリクエストの最大数を設定します。	○	なし

リクエストクエリの制限

タグ名	query

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	length

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レスポンスコード 4
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さが小さ 処理を行います。

リクエスト処理の直列化

タグ名	synchronized

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発生しな
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	デフォルト値
queue	処理待ちを行うリクエストの最大数を設定します。	○	なし

直列処理を行うリクエストクエリの制限

タグ名	query

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けます。

【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

タグ名	length

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並列処理

【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・設定する内容	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	デフォルト値
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。	○	なし

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(xsd)	WEB-INF/schema/response-header-config.xsd
設定場所	WEB-INF/conf/response-header-config/default.xml

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

リファレンス

静的なヘッダーの指定

タグ名	static-response-header

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"/>

</response-header-config>
```

必須項目	×
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 のいずれかを指定
format	type 属性に date を指定した際、value を解析するため 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

動的なヘッダーの指定

タグ名	dynamic-response-header

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_m

</response-header-config>
```

必須項目	×
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】

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_mar た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 のいずれかを指定し
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

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-c
  <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、すでに

【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 の機能
>
> 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にくくな
> 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負荷が減少

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 の処理

【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 の機能

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伝搬処
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処理が

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

概要

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定です。

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、受信側の

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め、デー
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れてい
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行いま

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

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うにして

```
<?xml version="1.0" encoding="UTF-8"?>
<propagation-receivers-config xmlns="http://www.intra-mart.jp/propagation/rec
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/propagation/receivers-config pro
 <receiver source="jp.co.intra_mart.foundation.sample.model.SampleModel" op
  <decoder class="jp.co.intra_mart.system.sample.propagation.SampleModelPr
   <params>
    <param key="key1">value1</param>
    <param key="key2">value2</param>
   </params>
  </decoder>
  <procedure class="jp.co.intra_mart.system.sample.propagation.SampleModel
   <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

タグ名	receiver

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処理内容

【設定項目】

```
<propagation-receivers-config>
 <receiver source="jp.co.intra_mart.foundation.sample.model.SampleModel" op
  .....
 </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修飾クラ です。
operationType	データ送信側が IM-Propagation に渡すデータの処理内容です

コラム

source 属性には IM-Propagation 用のデータが格納されたクラスの完
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受け取る

注意

データ送信側設定の sender タグに設定された source と operationTy
存在しないクラスや operationType が指定されている場合は、設定内容

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

【設定項目】

```
<propagation-receivers-config>
  <receiver source="jp.co.intra_mart.foundation.sample.model.SampleModel" op
    <decoder class="jp.co.intra_mart.system.sample.propagation.SampleModelPr
  </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
親タグ	receiver

【属性】

属性名	説明	必須	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フェー
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。

データ変換クラスパラメータの設定

タグ名	params

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" op
  <decoder class="jp.co.intra_mart.system.sample.propagation.SampleModelPr
   <params>
    .....
   </params>
  </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
親タグ	decoder

データ変換クラスのパラメータ値の設定

タグ名	param

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" op
  <decoder class="jp.co.intra_mart.system.sample.propagation.SampleModelPr
   <params>
    <param key="key1">value1</param>
    <param key="key2">value2</param>
   </params>
  </decoder>
 </receiver>
</propagation-receivers-config>
```

必須項目	×
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、データ処理クラスを

【設定項目】

```
<propagation-receivers-config>
  <receiver source="jp.co.intra_mart.foundation.sample.model.SampleModel" op
    <procedure class="jp.co.intra_mart.system.sample.propagation.SampleModel
  </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
親タグ	receiver

【属性】

属性名	説明	必須	デフォルト値
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フェース
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.SampleModel" op
    <procedure class="jp.co.intra_mart.system.sample.propagation.SampleModel
      <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
省略時のデフォルト値	なし
親タグ	procedure

データ処理クラスのパラメータ値の設定

タグ名	param

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" op
    <procedure class="jp.co.intra_mart.system.sample.propagation.SampleModel
      <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

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です。

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、送信側の

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め、デー
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-Propagat
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/{任意のファ

> **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うにして

```
<?xml version="1.0" encoding="UTF-8"?>
<propagation-senders-config xmlns="http://www.intra-mart.jp/propagation/senc
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/propagation/senders-config prop
 <sender source="jp.co.intra_mart.foundation.sample.model.SampleModel" ope
  <encoder class="jp.co.intra_mart.system.sample.propagation.SampleModelPr
   <params>
    <param key="key1">value1</param>
    <param key="key2">value2</param>
   </params>
  </encoder>
  <required-procedure class="jp.co.intra_mart.system.sample.propagation.Sam
 </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処理内容

【設定項目】

```
<propagation-senders-config>
  <sender source="jp.co.intra_mart.foundation.sample.model.SampleModel" ope
    .....
  </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 です。
operationType	データ送信側が IM-Propagation に渡すデータの処理内容です

コラム

operationType 属性には IM-Propagation によりいくつかの候補が用意
自モジュール内で完結する伝搬処理の場合は、独自の operationType

注意

source 属性に定義されていないクラスが IM-Propagation に渡された場
送ら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	encoder

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

【設定項目】

```
<propagation-senders-config>
  <sender source="jp.co.intra_mart.foundation.sample.model.SampleModel" ope
    <encoder class="jp.co.intra_mart.system.sample.propagation.SampleModelPr
  </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	sender

【属性】

属性名	説明	必須	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フェース

データ変換クラスパラメータの設定

タグ名	params

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.SampleModel" ope
    <encoder class="jp.co.intra_mart.system.sample.propagation.SampleModelPr
      <params>
        .....
      </params>
    </encoder>
  </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	encoder

データ変換クラスのパラメータ値の設定

タグ名	param

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.SampleModel" ope
  <encoder class="jp.co.intra_mart.system.sample.propagation.SampleModelPr
   <params>
    <param key="key1">value1</param>
    <param key="key2">value2</param>
   </params>
  </encoder>
 </sender>
</propagation-senders-config>
```

必須項目	×
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

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スを指

【設定項目】

```
<propagation-senders-config>
  <sender source="jp.co.intra_mart.foundation.sample.model.SampleModel" ope
    <required-procedure class="jp.co.intra_mart.system.sample.propagation.Sam
  </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	sender

【属性】

属性名	説明	必須	デ値
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。	○	な

> **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場合、伝

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

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グを行う

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-jssp-config.xsd
設定場所	WEB-INF/conf/admin-routing-jssp-config/{任意のファイル

```
<?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-
  <file-mapping path="/system/application/main" page="system/application/view/
  <file-mapping path="/system/application/list" page="system/application/view/lis
  <file-mapping path="/system/application/process/create" page="system/applica
  <file-mapping path="/system/application/process/update" page="system/applic
  <file-mapping path="/system/application/process/delete" page="system/applica
</admin-routing-jssp-config>
```

リファレンス

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】

```
<admin-routing-jssp-config>
  <file-mapping path="/system/application/main" page="system/application/view/
  <file-mapping path="/system/application/list" page="system/application/view/lis
</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	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す 値に {＜識別子＞} を記述することでURLの途中の値がリ 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ムを指定
action	page属性に指定されたプログラムの実行前に呼び出す関 from属性を指定した場合はfrom属性に指定されたスクリ from属性が未指定の場合はpage属性に指定されたスク ます。

属性名	説明
from	action属性で指定した関数を呼び出すプログラムを指定し

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パス形
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ムが

【設定項目】

```
<admin-routing-jssp-config>
  <folder-mapping path-prefix="/system/application" folder="system/application"
</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ィレクトリ 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、スクリプト開発モデ
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-mappingを記述

システム管理者用IM-JavaEE Frameworkルーティン

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の処理のマッピングを

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-config.xsd
設定場所	WEB-INF/conf/admin-routing-service-config/{任意のファ

```
<?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-service-con
  <service-mapping path="/system/application/main" application="application" se
</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

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】

```
<admin-routing-service-config>
  <service-mapping path="/system/application/main" application="application" se
</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です。 値に {＜識別子＞} を記述することでURLの途中の値が HttpServletRequest#getAttribute()で取得可能です。 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のサービス名を指定

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行いま
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がマッ

【設定項目】

```
<admin-routing-service-config>
  <application-mapping path-prefix="/system/application" application="applicatio
</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のプログラムのアプリ 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 Frame
> れません。
> 使用するIM-JavaEE Frameworkのサービスに対してservice-mapping

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ピング

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-config.xsd
設定場所	WEB-INF/conf/admin-routing-servlet-config/{任意のファ

```
<?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-servlet-cont
  <servlet-mapping path="/system/application/main" servlet="application/main" />
</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/main" />
</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す。 値に {＜識別子＞} を記述することでURLの途中の値が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を設定
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 ../sch
  <decision-config name="default" combinator="permit-overrides">
    <module class="jp.co.intra_mart.foundation.authz.services.decision.impl.Adr
    <module class="jp.co.intra_mart.foundation.authz.services.decision.impl.Pla
    <module class="jp.co.intra_mart.foundation.authz.services.decision.impl.Sta
  </decision-config>
</authz-decision-config>
```

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結果を

【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 ../sch
  <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予約され

combinator	子要素に指定する情報 <module> タグに指定された
	NotApplicable)。その結果をどのように取り扱うかを
	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 / Block を返
	• deny-overrides
	  ◦ 上から順に評価して最初に Deny / Block を返し
	• first-applicable
	  ◦ 上から順に評価して最初に Permit / Deny / Blo

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の詳細

> **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を実行し

【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 ../sch
  <decision-config name="default" combinator="permit-overrides">
    <module class="jp.co.intra_mart.foundation.authz.services.decision.impl.Sta
  </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クラス名を指定します。

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 ../schem
  <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.xlsx</te
    <path>im_authz/xlsx_export</path>
    <file-name>yyyyMMdd_HHmmss</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

タグ名	policy-update

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	×
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-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	×
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-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するかと設定します。

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	×
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-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します。

【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	×
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	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自動で

【設定項目】

```
<authz-editor-config>
  <resource-group-backup>
    ...
  </resource-group-backup>
</authz-editor-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-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、パブリックストレージの

【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に利用

【設定項目】

```
<authz-editor-config>
  <xlsx-import>
    ...
  </xlsx-import>
</authz-editor-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-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形式のファイルを配置す
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ートから
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ファイルも

【設定項目】

```
<authz-editor-config>
  <xlsx-import>
    <path>im_authz/xlsx_import</path>
  </xlsx-import>
</authz-editor-config>
```

必須項目	×
複数設定	×
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所を、パ 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

Excelインポートコミット間隔設定

タグ名	commit-count

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）を行った後にデ

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ベース・
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）インポート・エ

【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

タグ名	validate-inherits

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ポリシー
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が反
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ルに対

- セルの内容が「未設定（継承した結果、許可）」であり、セルの上位リソースグルー
- セルの内容が「未設定（継承した結果、禁止）」であり、セルの上位リソースグルー

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詳細は、

【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を設定しま
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

タグ名	validate-resource-exists

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・リソースに関す
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合、エ

【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	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するデー
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、エラー扱いとしてインポー

【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、古い順にフ
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値
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、パブリックストレー

【設定項目】

```
<authz-editor-config>
  <xlsx-import>
    <max-size>10</max-size>
  </xlsx-import>
</authz-editor-config>
```

必須項目	×
複数設定	×
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際に利

【設定項目】

```
<authz-editor-config>
  <xlsx-export>
    ...
  </xlsx-export>
</authz-editor-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-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のパ

【設定項目】

```
<authz-editor-config>
  <xlsx-export>
    <template-file>products/im_authz_xlsx_import_export/authz_template.xlsx</t
  </xlsx-export>
</authz-editor-config>
```

必須項目	×
複数設定	×
設定値・設定する内容	エクスポートするExcel（xlsx）ファイルを作成するた します。
単位・型	文字列
省略時のデフォルト値	products/im_authz_xlsx_import_export/authz_t
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	path

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ートから

【設定項目】

```
<authz-editor-config>
  <xlsx-export>
    <path>im_authz/xlsx_export</path>
  </xlsx-export>
</authz-editor-config>
```

必須項目	×
複数設定	×
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所を、パブ 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	file-name

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す。利用
JavaDocを参照してください。

【設定項目】

```
<authz-editor-config>
  <xlsx-export>
    <file-name>yyyyMMdd_HHmmss</file-name>
  </xlsx-export>
</authz-editor-config>
```

必須項目	×
複数設定	×
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。
単位・型	文字列
省略時のデフォルト値	yyyyMMdd_HHmmss
親タグ	xlsx-export

Excelエクスポートファイル拡張子設定

タグ名	file-ext

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りです。

- `{^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	×
複数設定	×
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	max-size

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ます。
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、古い順に
認可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設定で指
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、パブリックストレー

【設定項目】

```
<authz-editor-config>
  <xlsx-export>
    <max-size>10</max-size>
  </xlsx-export>
</authz-editor-config>
```

必須項目	×
複数設定	×
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	xlsx-password

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
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・削除が可能にな

【設定項目】

```
<authz-editor-config>
  <xlsx-export>
    <xlsx-password>intramart</xlsx-password>
  </xlsx-export>
</authz-editor-config>
```

必須項目	×
複数設定	×
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認可

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-config.xsd
設定場所	WEB-INF/conf/authz-partial-policy-edit-config/{任意の

```
<?xml version="1.0" encoding="UTF-8"?>
<authz-partial-policy-edit-config xmlns="http://www.intra-mart.jp/authz/authz-p
  <part-config>
    <part-id>im_authz_impl_router</part-id>
    <caption-cd>CAP.Z.IWP.ROUTER.AUTHZ.PARTCONFIG.TITLE</caption-c
    <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.ResourceGroup
    </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	part-config

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まとまりです。

【設定項目】

```
<authz-partial-policy-edit-config xmlns="http://www.intra-mart.jp/authz/authz-p
  <part-config>
    ...
  </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
省略時のデフォルト値	なし
親タグ	authz-partial-policy-edit-config

設定ID

タグ名	part-id

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プログラム
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を付与して

【設定項目】

```
<authz-partial-policy-edit-config xmlns="http://www.intra-mart.jp/authz/authz-pa
  <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	×
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意味のある
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	caption-cd

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。このコードをも

【設定項目】

```
<authz-partial-policy-edit-config xmlns="http://www.intra-mart.jp/authz/authz-pa
  <part-config>
    <caption-cd>CAP.Z.IWP.ROUTER.AUTHZ.PARTCONFIG.TITLE</caption-cd
  </part-config>
</authz-partial-policy-edit-config>
```

必須項目	×
複数設定	×
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、ユーザ
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/authz-p
  <part-config>
    <resource-groups>
      ...
    </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	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

タグ名	resource-group-id

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可能性

【設定項目】

```
<authz-partial-policy-edit-config xmlns="http://www.intra-mart.jp/authz/authz-p
  <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

タグ名	subject-types

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設定画
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/authz-p
  <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
省略時のデフォルト値	なし
親タグ	part-config

サブジェクトタイプ

タグ名	subject-type-id

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/authz-p
  <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

タグ名	callbacks

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

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表示する
めに使用されます。

【設定項目】

```
<authz-partial-policy-edit-config xmlns="http://www.intra-mart.jp/authz/authz-pa
  <part-config>
    <callbacks>
      ...
    </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	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グル

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表示する
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うにする
ます。

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され、認

【設定項目】

```
<authz-partial-policy-edit-config xmlns="http://www.intra-mart.jp/authz/authz-pa
  <part-config>
    <callbacks>
      <resource-group-authorizer>jp.co.intra_mart.system.authz.ResourceGroup
    </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	リソースグループに対する認可設定可否を判断するクラスの完
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権限を
ザに対して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 操作を

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してしま

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-config.xs
設定場所	WEB-INF/conf/authz-protected-resource-config/{任意

```
<?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-resources auth

  <protected-resource uri="service://authz/settings/basic" action="execute" />
  <protected-resource uri="service://authz/settings/parts" action="execute" />
  <protected-resource uri="service://authz/settings/procedure" action="execute"

</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-resources auth

  <protected-resource uri="service://authz/settings/basic" action="execute" />

</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。データベース上に 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定義設

項目
- 概要
- リファレンス
  - リソースマッパー設定

概要

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、同様に紐

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予め用意
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

```
<?xml version="1.0" encoding="UTF-8"?>
<authz-resource-mappers xmlns="http://www.intra-mart.jp/router/authz-resourc
  <mapper name="dev-auto-register" class="jp.co.intra_mart.system.router.authz
  <mapper name="welcome-all" class="jp.co.intra_mart.system.router.authz.user
</authz-resource-mappers>
```

リファレンス

リソースマッパー設定

タグ名	mapper

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.AuthzResourceM

【設定項目】

```
<authz-resource-mappers xmlns="http://www.intra-mart.jp/router/authz-resourc
  <mapper name="welcome-all" class="jp.co.intra_mart.system.router.authz.user
</authz-resource-mappers>
```

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-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-defau のはここで指定した値です。
class	この名前に割り当てるマッパーの実装クラスの完全修飾クラ

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.xsd
設定場所	WEB-INF/conf/authz-resource-type-config/{任意のファイ

```
<?xml version="1.0" encoding="UTF-8"?>
<authz-resource-type-config xmlns="http://www.intra-mart.jp/authz/authz-resou
  <resource-type
    type-class="jp.co.intra_mart.system.router.authz.resourcetype.GeneralService
    model-class="jp.co.intra_mart.system.router.authz.resourcetype.GeneralServic
    cache-class="jp.co.intra_mart.system.router.authz.resourcetype.GeneralServic
</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のセット

【設定項目】

```
<authz-resource-type-config xmlns="http://www.intra-mart.jp/authz/authz-resou
  <resource-type
    type-class="jp.co.intra_mart.system.router.authz.resourcetype.GeneralService
    model-class="jp.co.intra_mart.system.router.authz.resourcetype.GeneralServic
    cache-class="jp.co.intra_mart.system.router.authz.resourcetype.GeneralServic
</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 す。
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の完全修飾 定します。

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/{任意のファイル

```
<?xml version="1.0" encoding="UTF-8"?>
<authz-subject-type-config xmlns="http://www.intra-mart.jp/authz/authz-subject
  <subject-type type-class="jp.co.intra_mart.foundation.authz.subjecttype.im_ma
    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

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クラス

【設定項目】

```
<authz-subject-type-config xmlns="http://www.intra-mart.jp/authz/authz-subject
  <subject-type type-class="jp.co.intra_mart.foundation.authz.subjecttype.im_ma
    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】

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指定しま
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の完全修定します。

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設定を

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.xml 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.standard.Stand
  </cryption>

  <!-- アカウントパスワード用 -->
  <cryption category="account">
    <cryption-class>jp.co.intra_mart.foundation.security.cryption.standard.Stand
  </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.standard.Stand
  </cryption>

</cryption-config>
```

リファレンス

暗号化設定

タグ名	cryption

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	cryption-config

【属性】

属性名	説明	必
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設定します。	○
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から
default="true" の暗号化設定が見つからない場合、設定ファイルで先頭

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	cryption-class

暗号化方式の実装クラスの設定を行います。

【設定項目】

```
<cryption-config>
  <cryption category="default" default="true">
    <cryption-class>jp.co.intra_mart.foundation.security.cryption.standard.Stand
  </cryption>
</cryption-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.securi す。
単位・型	文字列
省略時のデフォルト値	なし
親タグ	cryption

サブジェクトリゾルバ(DeclaredSubjectResolver)拡張

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のう

サブジェクトリゾルバ、DeclaredSubjectResolverについての詳細は認可仕様書を

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.xsd
設定場所	WEB-INF/conf/declared-subject-resolvers/{任意のファイ

```
<?xml version="1.0" encoding="UTF-8"?>
<declared-subject-resolvers xmlns="http://www.intra-mart.jp/authz/declared-su
  <class-name>jp.co.intra_mart.foundation.authz.subjectresolver.im_master.lwp
</declared-subject-resolvers>
```

リファレンス

DeclaredSubjectResolverの追加設定

タグ名	class-name

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/declared-su
  <class-name>jp.co.intra_mart.foundation.authz.subjectresolver.im_master.lwp
</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の完全修飾ク
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(OnDemandSubjectResolver)

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バのう

サブジェクトリゾルバ、OnDemandSubjectResolverについての詳細は認可仕様書

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-resolvers.xsd
設定場所	WEB-INF/conf/ondemand-subject-resolvers/{任意のファ

```
<?xml version="1.0" encoding="UTF-8"?>
<ondemand-subject-resolvers xmlns="http://www.intra-mart.jp/authz/ondeman
  <class-name>sample.TimeDependResolver</class-name>
</ondemand-subject-resolvers>
```

リファレンス

OnDemandSubjectResolverの追加設定

タグ名	class-name

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/ondeman
  <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の完全修
単位・型	文字列
省略時のデフォルト値	なし
親タグ	ondemand-subject-resolvers

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、システム管理

デフォルト (未設定) の状態では javax.servlet.ServletRequest#getRemoteAddr(
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.servlet.S
この場合、代替となるHTTPヘッダ (X-Forwarded-For など) を設定することで、その

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を採用
最後まで取得できなかった場合は、javax.servlet.ServletRequest#getRemoteAd
javax.servlet.ServletRequest#getRemoteAddr() の動作について詳しくは JDK 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/ip-address-detector-config.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/ip-addre
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.intra-mart.jp/http/utility/ip-address-detector-co
 <remote-address>
   <http-header delimiter="," position="first">X-Forwarded-For</http-header>
   <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

タグ名	remote-address

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

タグ名	http-header

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ヘッダ名を定義します。

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を採用
最後まで取得できなかった場合は、javax.servlet.ServletRequest#getRemoteAc

【設定項目】

```
<ip-address-detector-config>
  <remote-address>
    <http-header delimiter="," position="first">X-Forwarded-For</http-header>
    <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評価します。

属性名	説明
position	delimiter で区切られた複数の値から評価に使用する値の場所で 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。範囲外の場合は評価 ん。 正の数を指定した場合は、最初から数えて何番目を評価 決定します。 負の数を指定した場合は、最後から数えて何番目を評価 決定します。

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
使用するロードバランサ、プロキシサーバの実装に応じて、delimiter, pos

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/{任意のファイル名}.xm

```
<?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 ../sche
 <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	cache-target

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	resource-type

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。指定 です。
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、パスワードの定期的な
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.password.
  <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-page>
  <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">!&amp;quot;#$%&amp;amp;&amp;apos;()+,-./:;&amp;lt;=&
   <deny-old-password>false</deny-old-password>
   <deny-userid>false</deny-userid>
  </check-password>
  <password-cryption-class>jp.co.intra_mart.foundation.security.cryption.stanc
  <return-initial-page>false</return-initial-page>
 </group-default>

 <group name="secondary" accessor-class="jp.co.intra_mart.foundation.security
  <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-page>
  <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">!&amp;quot;#$%&amp;amp;&amp;apos;()+,-./:;&amp;lt;=&
   <deny-old-password>false</deny-old-password>
   <deny-userid>false</deny-userid>
  </check-password>
  <password-cryption-class>jp.co.intra_mart.foundation.security.cryption.stanc
  <return-initial-page>false</return-initial-page>
 </group>

 <group name="thirdly" accessor-class="jp.co.intra_mart.foundation.security.pas
  <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-page>
  <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">!&amp;quot;#$%&amp;amp;&amp;apos;()+,-./:;&amp;lt;=&
   <deny-old-password>false</deny-old-password>
   <deny-userid>false</deny-userid>
  </check-password>
  <password-cryption-class>jp.co.intra_mart.foundation.security.cryption.stanc
  <return-initial-page>false</return-initial-page>
 </group>
</password-history>
```

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でテナ
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。

コラム

password-history.xmlはpassword-history-config.xsdを参照していな
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

タグ名	group-default

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.PasswordHisto フェース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="StandardPasswordHistoryAccesso
    ...
  </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

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 jp.co.intra_mart.foundation.security.password.Pas ンタフェース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

初期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。

【設定項目】

```
<password-history>
  <group-default accessor-class="jp.co.intra_mart.foundation.security.password.
    <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	false 初期ログイン時にパスワード変更を要求しません。 true 初期ログイン時にパスワード変更を要求します。
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.password.
    <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.password.
    <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.password.
    <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を設定しま
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.password.
    <password-expire-page>/user/password/expire</password-expire-page>
  </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定したい 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デフォルト 変更画面として扱われます。

パスワード文字列チェック

タグ名	check-password

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.password.
    <check-password enable="true">
      ...
    </check-password>
  </group-default>
</password-history>
```

必須項目	×
複数設定	×
設定値・設定する内容	パスワード文字列に対する有効チェックの内容を設定します。
単位・型	なし
省略時のデフォルト値	親タグが group-default 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	必須
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

タグ名	password-cryption-class

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.password.
    <password-cryption-class>jp.co.intra_mart.foundation.security.cryption.stand
  </group-default>
</password-history>
```

必須項目	×
複数設定	×
設定値・設定する内容	jp.co.intra_mart.foundation.security.cryption.Cryption
単位・型	文字列 (xxxxxxxxxxxxx)
省略時のデフォルト値	親タグが group-default の場合、jp.co.intra_mart.founda 親タグが group の場合、group-default タグ内の設定値

親タグ	group-default, group

パスワード変更後ログイン画面表示

タグ名	return-initial-page

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、ログイン画面に戻るかどう

【設定項目】

```
<password-history>
  <group-default accessor-class="jp.co.intra_mart.foundation.security.password.
    <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表示します。 true　パスワード変更後にログイン画面を表示します。
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

タグ名	check-password-length

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、なし 親タグが group/check-password の場合、group-default/ch
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。

属性名	説明
max	パスワードの最大の長さを設定します。

パスワード利用可能文字(英文字)

タグ名 allow-latin-letters

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 (xxxxxxxxxxxxx)
省略時のデフォルト値	親タグが group-default/check-password の場合、A-Za-z 親タグが group/check-password の場合、group-default/ch
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)の制約を行うかどうか true の場合は、少なくとも1文字は設定した文字が必要

パスワード利用可能文字(数字)

タグ名 allow-number

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 (xxxxxxxxxxxxx)
省略時のデフォルト値	親タグが group-default/check-password の場合、0-9 親タグが group/check-password の場合、group-default/ch
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)の制約を行うかどうかを true の場合は、少なくとも1文字は設定した文字が必要

パスワード利用可能文字(その他)

タグ名　allow-extra-char

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">!&amp;quot;#$%&amp;amp;&amp;apos;()+,-./:;&amp;lt;=&
    </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 (xxxxxxxxxxxxx)
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-default/ch
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)の制約を行うかどうか true の場合は、少なくとも1文字は設定した文字が必要

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利用を許可
	true　パスワード履歴に存在するパスワードの利用を禁止
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、false 親タグが group/check-password の場合、group-default/ch

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許可します。 true　ユーザコードと同じパスワードの利用を禁止します。
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、false 親タグが group/check-password の場合、group-default/ch
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を行いま

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/{任意のファイル名}.xml

```
<?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 ../sche
  <authz-default uri="service://application/default" action="execute" />
  <file-mapping path="/user/application/main" page="user/application/view/main'
  <file-mapping path="/user/application/list" page="user/application/view/list" fro
    <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/application/
    <authz uri="service://application/update" action="execute" />
  </file-mapping>
  <file-mapping path="/user/application/process/update" page="user/application/
    <authz uri="service://application/update" action="execute" />
  </file-mapping>
  <file-mapping path="/user/application/process/delete" page="user/application/
    <authz uri="service://application/update" action="execute" />
  </file-mapping>
</routing-jssp-config>
```

リファレンス

デフォルト認可設定

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	×
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	必
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、file-mappingタグ、folder-mappingタ

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/main"
  <file-mapping path="/user/application/list" page="user/application/view/list" fro
    ...
  </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「*」を指定することが可能で 値に {<識別子>} を記述することでURLの途中の値が 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ムを指
action	page属性に指定されたプログラムの実行前に呼び出す from属性を指定した場合はfrom属性に指定されたスクリ from属性が未指定の場合はpage属性に指定されたスク す。
from	action属性で指定した関数を呼び出すプログラムを指定
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ッ

> **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パス形

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ムが

【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ィレクトリ 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マッピングに 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、スクリプト開発モデ
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-mappingを記述

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/list" fror
    <authz mapper="sample-mapper">
      ...
    </authz>
  </file-mapping>
  <file-mapping path="/user/application/process/create" page="user/application/a
    <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 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設定で設定 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folde

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/list" fror
    <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

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定を行い

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/routing-service-config.xsd
設定場所	WEB-INF/conf/routing-service-config/{任意のファイル名}.x

```
<?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-config ../sc
  <authz-default uri="service://application/default" action="execute" />
  <service-mapping path="/user/application/main" application="application" serv
  <service-mapping path="/user/application/list" application="application" service
    <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="applica
    <authz uri="service://application/update" action="execute" />
  </service-mapping>
  <service-mapping path="/user/application/process/update" application="applica
    <authz uri="service://application/update" action="execute" />
  </service-mapping>
  <service-mapping path="/user/application/process/delete" application="applica
    <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-service-config>
  <authz-default uri="service://application/default" action="execute" />
</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	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設定で設定の値です。

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、application-r

サービスマッピング設定

タグ名	service-mapping

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" serv
  <service-mapping path="/user/application/list" application="application" service
    ...
  </service-mapping>
  <service-mapping path="/user/application/process/create" application="applica
    ...
  </service-mapping>
</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	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です。 値に {<識別子>} を記述することでURLの途中の値が HttpServletRequest#getAttribute()で取得可能です。 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のサービス名を指定
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ッピン

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行いま
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がマッ

【設定項目】

```
<routing-service-config>
  <application-mapping path-prefix="/user/application" application="application"
    ...
  </application-mapping>
</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	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のプログラムのアプリケ 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マッピング

> **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 Frame
> れません。
> 使用するIM-JavaEE Frameworkのサービスに対してservice-mapping

認可設定

タグ名	authz

認可設定

【設定項目】

```
<routing-service-config>
  <service-mapping path="/user/application/list" application="application" service
    <authz mapper="sample-mapper">
      ...
    </authz>
  </service-mapping>
  <service-mapping path="/user/application/process/create" application="applica
    <authz uri="service://application/update" action="execute" />
  </service-mapping>
</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	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設定で設定の値です。

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-service-config>
  <service-mapping path="/user/application/list" application="application" service
    <authz mapper="sample-mapper">
      <param key="foo" value="FOO" />
      <param key="bar" value="BAR" />
    </authz>
  </service-mapping>
</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	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

サー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
  - サー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、フォワード先のサーブレットパスへのマッピングを行うための設定を行

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/{任意のファイル名}.x

```
<?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-config ../sc
  <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" />
    <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="application/
    <authz uri="service://application/update" action="execute" />
  </servlet-mapping>
  <servlet-mapping path="/user/application/process/update" servlet="application
    <authz uri="service://application/update" action="execute" />
  </servlet-mapping>
  <servlet-mapping path="/user/application/process/delete" servlet="application/
    <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-servlet-config>
  <authz-default uri="service://application/default" action="execute" />
</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
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設定で設定の値です。

> **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タグを

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】

```
<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能です。 値に {<識別子>} を記述することでURLの途中の値が 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。
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ッピン

注意

認可によるアクセス制御について

サーブレットに対して認可によるアクセス制御を行う場合、直接アクセ
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を追加す

- web.xml に定義されたサーブレットに対するURL（servlet-mapping
- <routing-servlet-config> タグ内 <servlet-mapping> タグの path
- <routing-servlet-config> タグ内 <servlet-mapping> タグの servl

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へ直接

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としま

```
<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-mappi

```
<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は指
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

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/
  <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。
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設定で設定 の値です。

コラム

authz-defaultタグが指定されていて、かつ、servlet-mappingタグにauth

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-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	×
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からアクセスする事で

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.Stan
    <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.sh
    <validator code="RoleUser" class="jp.co.intra_mart.foundation.security.shortc
    <validator code="Script" class="jp.co.intra_mart.foundation.security.shortcut.S
  </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)は存在し
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

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)の実装クラスを設

【設定項目】

```
<short-cut-config>
  <short-cut-accessor>
    <short-cut-accessor-class>jp.co.intra_mart.foundation.security.shortcut.Stan
  </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.ShortCutA
単位・型	文字列 (xxxxxxxxxxxxx)
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を設
単位・型	文字列 (xxxxxxxxxxxxx)
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パスを設定し
単位・型	文字列 (xxxxxxxxxxxxx)
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	validator

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ログラム

【設定項目】

```
<short-cut-config>
  <short-cut-accessor>
    <validator code="RegExpUser" class="jp.co.intra_mart.foundation.security.sh
  </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.ShortCutValid 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/validdate/config/system-va
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra_mart.co.jp/foundation/admin/validdate/co
  <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	min-date

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	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	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	max-date

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	必須	デフォルト値
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値を元
内部的には、正当性のチェックには SecureTokenManager を利用して行います。
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

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。正当ではない

> **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.Secure
> 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 (xsd)	WEB-INF/schema/token-filtering-target-config.xsd
設定場所	WEB-INF/conf/token-filtering-target-config/{任意のファイ

```
<?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-config ../s
  <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

タグ名	entry

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へアクセス時にセキュア 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

```
<?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 admin-
  <menu label="%CAP.Z.IWP.SYSTEM.HOME.CONSTRUCTION.TITLE" id="c
    <children>
      <menu href="system/settings/license" label="%CAP.Z.IWP.SYSTEM.HO
      <menu href="system/settings/tenant" label="%CAP.Z.IWP.SYSTEM.HOM
      <menu href="system/setup" label="%CAP.Z.IWP.SYSTEM.HOME.CONS
    </children>
  </menu>
  <menu label="%CAP.Z.IWP.SYSTEM.HOME.ADMINISTRATION.TITLE" id="a
    <children>
      <menu href="master/maintenance/call/user" label="%CAP.Z.IWP.SYSTE
      <menu href="system/service/module" label="%CAP.Z.IWP.SYSTEM.HOI
      <menu href="system/service/status" label="%CAP.Z.IWP.SYSTEM.HOM
      <menu href="system/service/lock" label="%CAP.Z.IWP.SYSTEM.HOME.
      <menu href="system/storage/view" label="%CAP.Z.IWP.SYSTEM.HOME
      <menu href="system/database" label="%CAP.Z.IWP.SYSTEM.HOME.AD
      <menu href="portal/portlet/app/list" label="%CAP.Z.IWP.SYSTEM.HOME
      <menu href="portal/setting" label="%CAP.Z.IWP.SYSTEM.HOME.ADMI
      <menu href="system/async/admin/task_queue_list" label="%CAP.Z.IWP.
      <menu href="system/settings/shared_database" label="%CAP.Z.IWP.SY
    </children>
  </menu>
</admin-global-navi-config>
```

リファレンス

メニュー設定

タグ名	menu

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" id="cons
  <children>
    <menu href="system/settings/license" label="%CAP.Z.IWP.SYSTEM.HOME
    <menu href="system/settings/tenant" label="%CAP.Z.IWP.SYSTEM.HOME.
    <menu href="system/setup" label="%CAP.Z.IWP.SYSTEM.HOME.CONSTR
  </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-global-navi-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。aタグ
id	このメニューのIDを指定します。複数ファイルに渡って合、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が指定されている場
children	メニューの子要素を指定します。子要素にはmenu要素になります。

コラム

label属性に%から始まる値を指定すると、MessageManager によるメッ

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.xsd
設定場所	WEB-INF/conf/admin-home-widget-config/{任意のファイ

```
<?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-config ../
  <widget url="system/home/widgets/status/status" captionUrl="system/home/w
  <widget url="system/home/widgets/module/module" id="module" row="2" col=
  <widget url="system/home/widgets/license/license" id="license" row="3" col="
  <widget url="system/home/widgets/system-info/system-info" id="system-info"
  <widget url="system/home/widgets/bookmark/bookmark" id="bookmark" row=
    <parameter name="bookmark" value="[
      {&quot;href&quot;: &quot;http://www.intra-mart.jp/&quot;, &quot;name&qu
      {&quot;href&quot;: &quot;http://www.intra-mart.jp/imart/partner.portal&quo
      {&quot;href&quot;: &quot;http://www.intra-mart.jp/support/intramart.cgi&qu
      {&quot;href&quot;: &quot;http://imfaq.intra-mart.jp/imfaq/&quot;, &quot;nar
      {&quot;href&quot;: &quot;http://www.intra-mart.jp/download/product/index
      {&quot;href&quot;: &quot;http://demo.intra-mart.jp/&quot;, &quot;name&qu
      {&quot;href&quot;: &quot;http://www.intra-mart.jp/apilist/&quot;, &quot;nar
      ]"/>
  </widget>
  <widget url="system/home/widgets/node/node" captionUrl="system/home/widg
  <widget url="system/home/widgets/tenant/tenant" id="tenant" row="2" col="1"
  <widget url="system/home/widgets/memory/memory" id="memory" row="2" co
</admin-home-widget-config>
```

リファレンス

ホームウィジェット設定

タグ名	widget

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/home/widg
```

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-home-widget-config

【属性】

属性名	説明	必須
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。	×

属性名	説明	必須
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ます。	×

パラメータ設定

タグ名	parameter

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&quot;: &
  ]"/>
```

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	widget

【属性】

属性名	説明	必須	デフォルト値
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

```
<?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 admin-uti
  <menu href="javascript:void(0);" label="%CAP.Z.IWP.SYSTEM.HOME.ADMIN
    <children>
      <menu href="system/administrator/password" label="%CAP.Z.IWP.SYST
      <menu href="system/administrator/locale" label="%CAP.Z.IWP.SYSTEM
    </children>
  </menu>
  <menu href="logout" label="%CAP.Z.IWP.SYSTEM.COMMON.LOGOUT" />
</admin-utility-config>
```

リファレンス

ユーティリティメニュー設定

タグ名	menu

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.ADMINIS
  <children>
    <menu href="system/administrator/password" label="%CAP.Z.IWP.SYSTEM
    <menu href="system/administrator/locale" label="%CAP.Z.IWP.SYSTEM.H
  </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。aタグ(
id	このメニューのIDを指定します。複数ファイルに渡って合、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が指定されている場

属性名	説明
children	メニューの子要素を指定します。子要素にはmenu要 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 によるメッ

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
フォーマットファイル(xsd)	WEB-INF/schema/im-notice-config.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-config ../s

   <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が使用されます。

IM-Noticeへのメッセージ配信処理をジョブ経由で行う場合は、ベースURLの設定が
「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」から設定を行っ

> 注意
>
> ベースURLを指定しない場合、以下のエラーが発生します。
> jp.co.intra_mart.system.notice.exception.NoticeRuntimeException:

【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テキストパ
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URL
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、有効期間が過ぎると

【設定項目】

```
<short-cut-duration-minutes>43200</short-cut-duration-minutes>
```

必須項目	○
複数設定	×
設定値・設定する内容	ショートカットURLの有効期間を設定してください。 「0」を設定した場合、ショートカットURLは作成されません。
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
  - プロジェクト番号の設定
  - 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-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/notice/mobile/config/im-notice-m

  <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>
    <gcma-project-number></gcma-project-number>
  </asns>

  <gcma name="gcma">
    <url>https://android.googleapis.com/gcm/send</url>
    <api-key></api-key>
    <project-number></project-number>
  </gcma>

  <push type="android" engine="gcma" />
  <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	proxy

Amazon SNSやGoogle Cloud Messaging for Androidへ接続するためのプロキシ

【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	デフォルト値
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。	○	false

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>
```

必須項目	×
複数設定	×
設定値・設定する内容	なし
単位・型	数値(0-)
省略時のデフォルト値	なし
親タグ	proxy

【属性】

属性名	説明	必須	デフォルト値
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。	×	true

ユーザ名の設定

タグ名	username

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

パスワードの設定

タグ名	password

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	×
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

ワークステーションの設定

タグ名	workstation

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	×
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

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	×
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

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します。

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	×

複数設定	×
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サービスアク 各リージョンのエンドポイントは AWS documentation - Re
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

タグ名	platform-application-arn

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:XXXXXXXXXXXX:
  </asns>
</im-notice-mobile-config>
```

必須項目	×
複数設定	×
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Application ARN
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

タグ名	access-key

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	×
複数設定	×
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

タグ名	secret-key

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>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</
  </asns>
</im-notice-mobile-config>
```

必須項目	×
複数設定	×
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

プロジェクト番号の設定

タグ名	gcma-project-number

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	○
複数設定	×
設定値・設定する内容	Google Cloud Messaging for Androidで作成したプロジ
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	gcma

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を使用するために必要
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
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	url

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」を設定してくださ
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>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</api-k
  </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で作成したアプリケ

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・設定する内容	GCMAで作成したプロジェクトのプロジェクト番号を設定してくだ
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	push

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タグのname属定してください。

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	×
複数設定	×
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してく
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してくださ
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-c

  <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	address

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ます。そ
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://{ブローカーを* 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	必須	デ 値

属性名	説明	必須	デ値
bind	スタンドアローン構成の場合、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。	○	tr

ping機能の設定

タグ名	ping

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	○
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を設定してくださ
interval-seconds	クライアントにpingを流す間隔を数値（秒）で設定してください。
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値（秒）で設い。

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ーのフロ

【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用したUF ブローカーを利用する場合、「*tcp://{ブローカーを* 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	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スレッドプールのサイズを設
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-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大数を設
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

タグ名	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期間保:
詳細は ZMQ_LINGER: Set linger period for socket shutdown を参照してくださ

【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・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-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(xsd)	WEB-INF/schema/im-notice-mq-broker-config.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-broker-c
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.intra-mart.jp/notice/mq/broker/config/im-notice

  <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	frontend

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設定し
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ポート番
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	backend

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ます。
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を
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

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を設定してください。
interval-seconds	クライアントにpingを流す間隔を数値（秒）で設定してください。

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スレッドプールのサイズを設
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大数を設
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

タグ名	zmq-backend-linger

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保持さ
詳細は ZMQ_LINGER: Set linger period for socket shutdown を参照してくださ

【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

タグ名	zmq-frontend-linger

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期間保
詳細は ZMQ_LINGER : Set linger period for socket shutdown を参照してくださ

【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