



Copyright © 2014 NTT DATA INTRAMART CORPORATION

# 目次

---

- 1. 改訂情報
- 2. はじめに
  - 本書の目的
  - 対象読者
  - 本書の構成
- 3. メニュー機能
  - 目的
  - メニューの種類
  - メニューの構成
  - メニューにおける認可の仕様
- 4. インポート・エクスポートの概要
  - インポート・エクスポート機能
  - インポート・エクスポートで扱う情報
  - インポート・エクスポート手順
- 5. ファイルフォーマット
  - メニューグループカテゴリ
  - メニューグループ
  - 互換性
- 6. インポート
  - 更新モード
  - メニューグループカテゴリ
  - メニューグループ
  - インポートの依存関係
- 7. エクスポート
  - メニューグループカテゴリ
  - メニューグループ
- 8. オプション
  - インポートオプション
  - エクスポートオプション
- 9. 実行方法（メニューグループカテゴリ）
  - ジョブスケジューラを利用する
  - Javaから実行する
  - スクリプト開発モデルプログラムから実行する
- 10. 実行方法（メニューグループ）
  - ジョブスケジューラを利用する
  - Javaから実行する
  - スクリプト開発モデルプログラムから実行する
- 11. 付録
  - メニューインポートデータ定義書

変更年月日	変更内容
2014-05-01	初版
2014-08-01	第2版 下記を追加しました <ul style="list-style-type: none"><li>■ 「<a href="#">メニュー グループに対する参照権限</a>」にURLでの直接アクセスについて説明を追記</li><li>■ 「<a href="#">URLに対する実行権限</a>」にURLでの直接アクセスについて説明を追記</li></ul>
2015-04-01	第3版 下記を追加しました <ul style="list-style-type: none"><li>■ 「<a href="#">メニュー機能</a>」の「<a href="#">メニュー グループ カテゴリ</a>」の表の列に導入バージョンを追加</li><li>■ 「<a href="#">メニュー機能</a>」の「<a href="#">メニュー グループ カテゴリ</a>」の表に新たに追加されたIDを追加</li></ul>
2018-04-01	第4版 下記を追加しました <ul style="list-style-type: none"><li>■ 「<a href="#">メニュー情報</a>」に新規ノード (icon-16、icon-32、icon-48) の情報を追加</li><li>■ 「<a href="#">XMLファイルフォーマット</a>」に新規ノード (icon-16、icon-32、icon-48) のサンプルを追加</li><li>■ 「<a href="#">メニューインポートデータ定義書</a>」に新規ノード (icon-16、icon-32、icon-48) の情報を追加</li></ul>

## 本書の目的

---

本書ではメニュー情報のインポート・エクスポート機能の詳細について説明します。

説明範囲は以下の通りです。

- メニュー情報のインポート・エクスポートの概要
- メニュー情報のインポート・エクスポートファイルのデータフォーマット
- メニュー情報のインポート・エクスポートの実行方法
- メニュー情報のインポート・エクスポートの実行オプション

## 対象読者

---

本書では次の利用者を対象としています。

- intra-mart Accel Platform のメニューを管理する運用担当者
- メニュー情報のインポート・エクスポート機能を利用したアプリケーションを開発する開発者

## 本書の構成

---

- [メニュー機能](#)

メニュー機能について説明します。

- [インポート・エクスポートの概要](#)

メニューのインポート・エクスポートの概要について説明します。

- [ファイルフォーマット](#)

インポートファイルのデータフォーマットについて説明します。

- [インポート](#)

インポートの処理について説明します。更新モードについても説明します。

- [エクスポート](#)

エクスポートの処理について説明します。

- [オプション](#)

インポート・エクスポート実行時に指定可能なオプションについて説明します。

- [実行方法（メニューグループカテゴリ）](#)

メニューグループカテゴリのインポート・エクスポート実行方法について説明します。

- [実行方法（メニューグループ）](#)

メニューグループのインポート・エクスポート実行方法について説明します。

## 項目

- 目的
- メニューの種類
- メニューの構成
  - メニューグループカテゴリ
  - メニューグループ
  - メニュー情報
    - メニューフォルダ
    - メニューアイテム
    - 外部メニューリンク
- メニューにおける認可の仕様
  - メニューグループカテゴリ、メニューグループに対しての管理権限
  - メニューグループに対しての参照権限
  - URLに対しての実行権限

## 目的

メニューとは、業務画面へアクセスするためのショートカットとなるリンクを集めたものです。

一般ユーザ視点から見ると、業務画面へのアクセス性を向上させることができます。

メニューからアクセスすることで直接 URL を指定してアクセスする事なく、一般ユーザが行おうとしている業務のスタート地点に素早く到達できることを目的としています。

管理者視点から見ると、テナントごとに利用者の行動パターンに則した動線設計でショートカットを構成することができます。

テナントによって異なるログインユーザごとの業務特性を考慮したメニューを管理者が設定できるようになり、ログインユーザがよく使用する業務画面へアクセスしやすくなることを目的としています。

## メニューの種類

intra-mart Accel Platform のメニューには以下の種類があります。

### 1. グローバルナビ

サイト内の各ページに共通して設置されているサイト内の各コンテンツをログインユーザに案内するためのメニューです。

現在アクセスしているページの場所に関わらず、画面の上部に表示され、目的のページへ素早くアクセスすることができます。

ここには全ての画面ではなく、ログインユーザがよく使うと思われる業務画面への入り口をメニュー管理者（テナント管理者からメニューに関する権限を与えられた管理者）が厳選して登録しておくことで、ログインユーザの利便性を向上させます。

初期状態では、モジュール製作者が作成したデフォルトのメニュー構成が登録され、メニュー管理者が自由に編集することができます。

### 2. サイトマップ

サイト内の各業務画面への入り口をすべて表示して、利用可能なサービスの一覧をログインユーザに提供するメニューです。

初期状態では、モジュール製作者が作成したデフォルトのメニュー構成が登録され、メニュー管理者が自由に編集することができます。

### 3. マイメニュー

ログインユーザが intra-mart Accel Platform 上でアクセス中のページを簡単にブックマークとして記録しておく機能です。

ブラウザのブックマークとは別に intra-mart Accel Platform 上でブックマークを管理します。

ログインユーザ自身で、ブックマークの登録や編集を自由に行うことができます。



画面における各メニューの配置

## メニューの構成

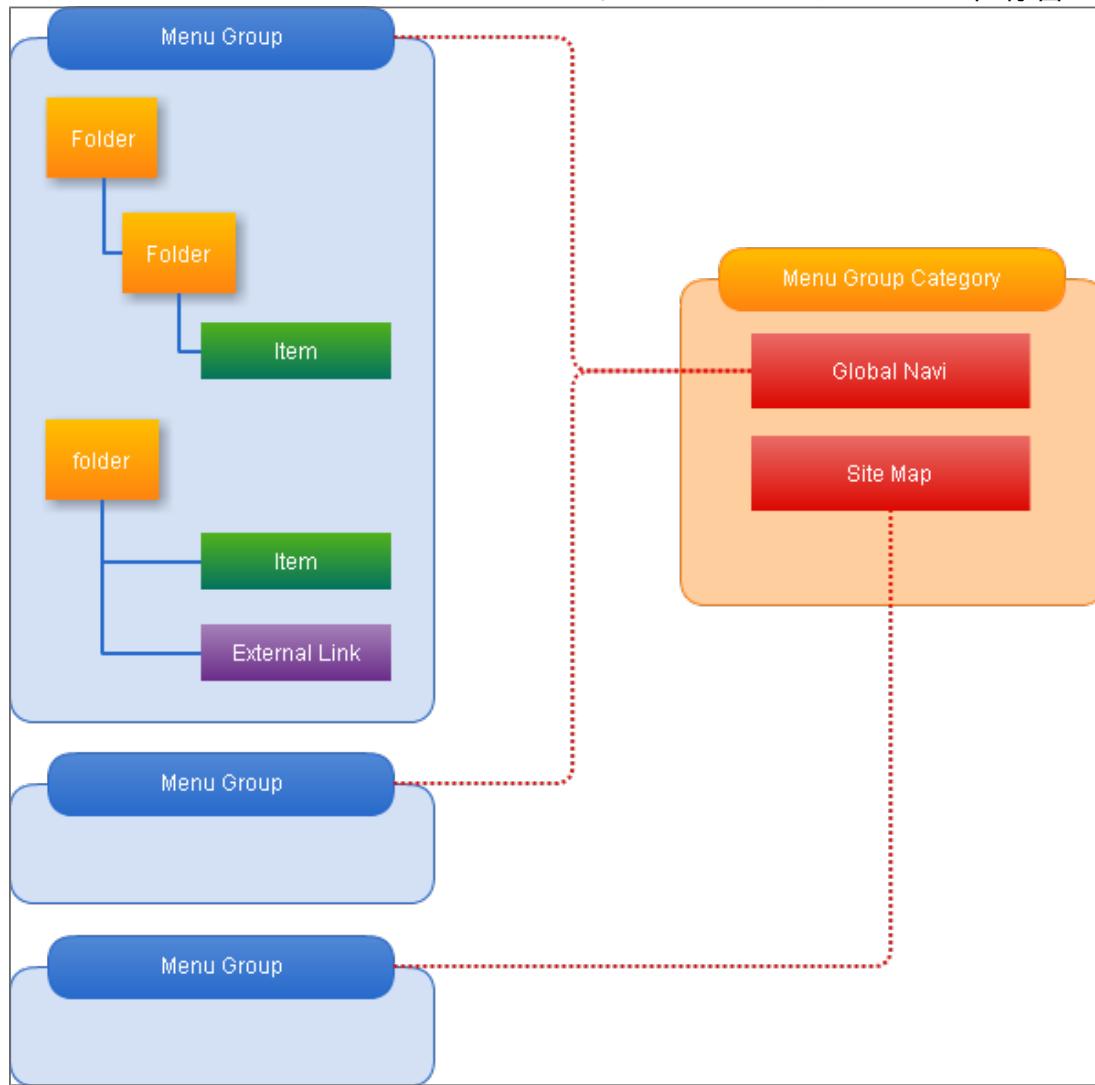


図 メニューの構成

## メニュー グループ カテゴリ

複数のメニュー グループの集合体で、モジュール、クライアントタイプ（PC・スマートフォン）などの単位で管理することができます。（図中の **Menu Group Category**）

メニュー グループ カテゴリは、メニュー設定画面上で追加・変更・削除することはできません。

intra-mart Accel Platform をインストールした直後の初期状態では、以下のメニュー グループ カテゴリが用意されています。

メニュー グループ カテゴリ			導入バージョン
ID	用途	モジュールID	
im_global_nav_pc	PC 用のグローバルナビ	jp.co.intra_mart.im_tenant	2012 Autumn
im_global_nav_sp	スマートフォン用のグローバルナビ	jp.co.intra_mart.im_mobile_framework	2012 Autumn
im_sitemap_pc	PC 用のサイトマップ	jp.co.intra_mart.im_tenant	2012 Autumn
im_sitemap_sp	スマートフォン用のサイトマップ	jp.co.intra_mart.im_mobile_framework	2012 Autumn

メニュー グループ カテゴリ			導入バージョン
ID	用途	モジュールID	
im_personal_settings_pc	PC 用のテーマで右上に表示される個人設定メニュー	jp.co.intra_mart.im_tenant	2012 Autumn
im_site_help_pc	PC 用のテーマで右上に表示されるヘルプドロップダウンのメニュー	jp.co.intra_mart.im_tenant	2014 Winter

メニュー グループ カテゴリには認可により権限を設定することができます。

詳しくは [メニューにおける認可の仕様](#) を参照してください。

## メニュー グループ

メニューの集合体を管理するものです。 (図中の **Menu Group**)

業務画面へのリンクや階層構造を持つために、メニュー グループ内で、メニュー フォルダ、メニュー アイテム、外部メニュー リンクを管理することができます。

メニュー グループには認可により権限を設定することができます。

詳しくは「[メニューにおける認可の仕様](#)」を参照してください。

## メニュー 情報

メニューを構成する最小単位の情報です。

メニュー種別によってメニュー フォルダ、メニュー アイテム、外部メニュー リンクに分類されます。

## メニュー フォルダ

メニュー グループ内のメニューの階層構造を構成するものです。 (図中の **Folder**)

メニュー種別は「**folder**」です。

メニュー フォルダ自身には認可による権限を設定できず、配下に登録された一部、または、全てのメニュー アイテムが表示状態になるとき、メニュー フォルダが表示されます。

また、全てのメニュー アイテムが非表示状態になるときは、メニュー フォルダは表示されません。

## メニュー アイテム

業務画面の入り口へアクセスするためのリンクを構成するものです。 (図中の **Item**)

メニュー種別は「**item**」です。

## 外部メニュー リンク

外部メニューを表示するためのリンクを構成するものです。 (図中の **External Link**)

リンクを構成するための情報は、外部メニューを提供しているサーバから取得します。

外部メニュー リンクは、実際に表示されるときに外部メニュー連携機能を利用して、外部メニューを提供しているサーバから最新の情報を取得して表示されます。

詳しくは、「[セットアップガイド](#)」 - 「[外部メニュー連携](#)」を参照してください。

メニュー種別は「**external**」です。

- 外部メニュー リンクの利用

メニュー設定画面で外部メニューリンクを利用するには、メニュープロバイダの設定で、`editable` 属性を「true」に設定します。

詳しくは、「設定ファイルリファレンス」の「[外部メニュー連携接続先設定](#)」 - 「[メニュー プロバイダの設定](#)」を参照してください。

また、外部メニューリンクが利用可能なのは、次のメニュー グループ カテゴリに属するメニュー グループのみです。

- `im_global_nav_pc`
- 外部メニューの取得先の設定

取得先のプロバイダID、メニューIDを以下の書式でURLに設定します。

`im-external-link://<プロバイダID>/<メニューID>`



### 注意

外部メニュー リンクを利用できるのは、2013 Autumn(Eden) 以降です。

## メニューにおける認可の仕様

メニューの登録、更新、削除、および、参照は、認可（IM-Authz）機構を利用して制御されます。

メニューに関わる認可（IM-Authz）機構を利用した制御は以下の通りです。

認可（IM-Authz）機構の詳細については「[認可仕様書](#)」を参照してください。

- メニューグループに対しての管理権限  
リソースタイプ「メニュー グループ」、アクション「管理」がこの設定に該当します。
- メニューグループに対しての参照権限  
リソースタイプ「メニュー グループ」、アクション「参照」がこの設定に該当します。
- URLに対しての実行権限  
リソースタイプ「画面・処理」、アクション「実行」がこの設定に該当します。

## メニュー グループ カテゴリ、メニュー グループに対する管理権限

メニュー グループに対する管理権限を付与することで、対象となるメニュー グループの更新、および、削除可能です。

メニュー グループの登録を行うためには、登録を行うメニュー グループ カテゴリへの管理権限が必要です。

このため、メニュー グループのインポートを行う場合は必ずメニュー グループの登録、および、更新が可能なユーザで行ってください。

ジョブ機能を利用してインポートを実行する場合、すべてのメニュー グループが管理可能なものとして扱われます。

- 例1：メニュー グループ カテゴリ「グローバルナビ（PC用）」にメニュー グループを登録する場合  
メニュー グループの認可リソース「グローバルナビ（PC用）」への管理権限を持つユーザである必要があります。
- 例2：メニュー グループ カテゴリ「グローバルナビ（PC用）」のメニュー グループ「Top」を更新する場合  
メニュー グループの認可リソース「Top」への管理権限を持つユーザである必要があります。



## コラム

メニュー グループ カテゴリの認可リソース（図中の①）に管理権限が付与されている場合、認可（IM-Authz）機構の認可設定の継承により、その配下のメニュー グループ（図中の②）すべてが管理可能です。

ただし、特定のメニュー グループの管理権限に「禁止」の権限設定されている場合は管理できません。

リソース	アクション	組織
メニュー グループ	管理 > 参照 >	サンプル会社
① グローバルナビ(PC用)	管理 > 参照 >	✓ ✗
Top	管理 > 参照 >	✓ ✗
テナント管理	管理 > 参照 >	✗ ✗
サンプル	管理 > 参照 >	✓ ✗
グローバルナビ(スマートフォン用)	管理 > 参照 >	✗ ✗
ViewCreator	管理 >	✗

## メニュー グループに対する参照権限

メニュー グループに対して参照権限を付与することで、そのメニュー グループを参照可能です。

例えば、標準テーマでは画面上部にグローバルナビが表示されますが、そのメニューには、参照権限のあるメニュー グループだけが表示されます。



## 注意

メニュー グループの参照権限がない場合、メニュー グループは表示されなくなりますが、メニュー アイテムに設定されている URL で直接アクセスすることはできます。

URL に対する権限制御については、「[URL に対する実行権限](#)」を参照してください。

## URL に対する実行権限

メニュー のデータ種別が「item」であるメニュー アイテムは、メニュー アイテムが持つ URL への実行（閲覧）権限があるもののみ表示されます。

ただし、URL が外部サイトである場合は常に表示されます。



## コラム

メニューアイテムが持つURLへの実行権限は、メニュー設定画面のメニューアイテム編集時に「権限設定」ボタンをクリックすることで確認可能です。  
URLへの実行権限がない場合、ブラウザから直接URLを入力してアクセスした場合でも、アクセスが遮断されます。

The screenshot shows the 'Menu Item Settings' dialog box. At the top, there's a title bar with the application logo and menu items like 'Top', 'テナント管理', 'サンプル', and 'サイトマップ'. Below the title bar, the main area is titled 'メニューアイテムの編集' (Edit Menu Item). A sub-section title 'メニュー設定' (Menu Settings) is visible. On the left, a sidebar shows a tree structure of menu items under 'グローバルメニュー' (Global Menu), including 'Top', 'テナント', 'サンプル', and 'サイトマップ'. The main form contains the following fields:

- メニューアイテムID**: tenant\_menu\_settings
- メニューアイテム名**:
  - 日本語\*: メニュー
  - 英語: Menu
  - 中国語(中華人民共和国): 菜单
- URL\***: tenant/menu/settings 権限設定
- 呼び出し方法**: GET
- 引数**: A table with columns 'キー' (Key) and '値' (Value). It has a header row with '+' (Add) and '-' (Delete) buttons. An empty row is shown.
- アイコン画像**:
  - ファイルパス: コンテキストパス配下のURLを入力してください。
  - CSS Sprites: imui://csssprites/ クラス名を入力してください。
- IFRAME表示**:
- ポップアップ表示**:

At the bottom right of the dialog is a '更新' (Update) button.

### 項目

- インポート・エクスポート機能
- インポート・エクスポートで扱う情報
- インポート・エクスポート手順
  - メニューグループカテゴリ
    - インポート
    - エクスポート
  - メニューグループ
    - インポート
    - エクスポート

## インポート・エクスポート機能

メニューには、メニューグループカテゴリ、メニューグループの2つのインポート・エクスポート機能があります。

メニューのインポート・エクスポートでサポートしているファイルフォーマットは、XML形式のみです。

メニューグループカテゴリ、メニューグループのインポートにおいて、削除を行う機能は提供されていません。

ただし、既存のメニューグループを置き換えることは可能です。

詳しくは、「[インポート](#)」 - 「[メニューグループ](#)」 - 「[メニューグループの更新モード](#)」 - 「[replace](#)」を参照してください。

## インポート・エクスポートで扱う情報

メニューのインポート・エクスポートでは、XML形式でメニュー関連情報のインポート・エクスポートを行います。

メニューのインポート・エクスポートで扱う情報は次の通りです。

- メニューグループカテゴリ
  - メニューグループカテゴリ情報
  - メニューグループカテゴリ表示名情報
- メニューグループ
  - メニューグループ情報
  - メニュー情報
  - メニュー引数情報
  - メニュー表示名情報
  - メニューグループカテゴリ内包情報

## インポート・エクスポート手順

### メニューグループカテゴリ

#### インポート

1. インポートファイルを作成します。  
インポートファイルはXML形式で作成します。  
詳しくは「[ファイルフォーマット](#)」 - 「[メニュー グループ カテゴリ](#)」を参照してください。
2. インポートファイルをパブリックストレージに配置します。
3. インポートファイルをインポートします。  
インポートには、次の実行方法があります。
  - 「[ジョブスケジューラを利用する](#)」
    - 「[ジョブ](#)」 - 「[インポート](#)」
    - 「[ジョブネット](#)」 - 「[インポート](#)」
  - 「[Javaから実行する](#)」 - 「[インポート](#)」
  - 「[スクリプト開発モデルプログラムから実行する](#)」 - 「[インポート](#)」

## エクスポート

メニュー グループ カテゴリのエクスポートを行う手順は次の通りです。

1. エクスポートします。  
エクスポートには、次の実行方法があります。
  - 「[ジョブスケジューラを利用する](#)」
    - 「[ジョブ](#)」 - 「[エクスポート](#)」
    - 「[ジョブネット](#)」 - 「[エクスポート](#)」
  - 「[Javaから実行する](#)」 - 「[エクスポート](#)」
  - 「[スクリプト開発モデルプログラムから実行する](#)」 - 「[エクスポート](#)」
2. パブリックストレージにエクスポートファイルが出力されます。

## メニュー グループ

---

### インポート

メニュー グループ のインポートを行う手順は次の通りです。

1. インポートファイルを作成します。  
インポートファイルはXML形式で作成します。  
詳しくは「[ファイルフォーマット](#)」 - 「[メニュー グループ](#)」を参照してください。
2. インポートファイルをパブリックストレージ直下に配置します。
3. インポートファイルをインポートします。  
インポートには、次の実行方法があります。
  - 「[ジョブスケジューラを利用する](#)」
    - 「[ジョブ](#)」 - 「[インポート](#)」
    - 「[ジョブネット](#)」 - 「[インポート](#)」
  - 「[Javaから実行する](#)」 - 「[インポート](#)」
  - 「[スクリプト開発モデルプログラムから実行する](#)」 - 「[インポート](#)」

メニュー グループのエクスポートを行う手順は次の通りです。

1. エクスポートします。

エクスポートには、次の実行方法があります。

- 「ジョブスケジューラを利用する」
  - 「ジョブ」 - 「エクスポート」
  - 「ジョブネット」 - 「エクスポート」
- 「Javaから実行する」 - 「エクスポート」
- 「スクリプト開発モデルプログラムから実行する」 - 「エクスポート」

2. パブリックストレージにエクスポートファイルが出力されます。

## 項目

- メニューグループカテゴリ
  - XMLファイルフォーマット
    - メニューグループカテゴリ情報
    - メニューグループカテゴリ表示名情報
- メニューグループ
  - XMLファイルフォーマット
    - メニューグループ情報
    - メニューグループカテゴリ内包情報
    - メニュー情報
    - メニュー引数情報
    - メニュー表示名情報
- 互換性

この章では、インポート・エクスポート機能で利用するファイルのフォーマットについて説明します。

ファイルフォーマットの詳細については「[メニューインポートデータ定義書](#)」を参照してください。

メニューのインポート・エクスポートでサポートしているファイルフォーマットは、XML形式のみです。

## メニューグループカテゴリ

### XMLファイルフォーマット

以下はXMLファイルの例です。

<menu-group-category-data> タグ内に 1 つのメニューグループカテゴリに関連する情報をすべて記述します。

```
<root xmlns="http://intra-mart.co.jp/im_menu/menu-group-category-data">
  <menu-group-category-data category="menu-group-category-1" selectable="true">
    <display-names>
      <display-name locale="ja">メニューグループカテゴリ 1 </display-name>
      <display-name locale="en">menu group category 1</display-name>
    </display-names>
  </menu-group-category-data>
  <menu-group-category-data category="menu-group-category-2" selectable="true">
    <display-names>
      <display-name locale="ja">メニューグループカテゴリ 2 </display-name>
      <display-name locale="en">menu group category 2</display-name>
    </display-names>
  </menu-group-category-data>
  <menu-group-category-data category="menu-group-category-3" selectable="true">
    <display-names>
      <display-name locale="ja">メニューグループカテゴリ 3 </display-name>
      <display-name locale="en">menu group category 3</display-name>
    </display-names>
  </menu-group-category-data>
</root>
```

インポートしたメニューグループカテゴリは、認可設定画面で確認することができます。

リソース	アクション	認証		組織		ロール							
		ゲストユーザー	認証済みユーザー	サンプル会社	その他会社	テナント管理者	認可管理者	メニュー管理者	メニュー運用管理者	アカウント管理者	ロール管理者	カレンダーマネジメント	ジョブ管理者
メニューグループ		▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼
サイトマップ(PC用)		参照	>	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
個人設定	管理	>	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
	参照	>	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
サイトマップ(スマートフォン用)	管理	>	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
	参照	>	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
ViewCreator	管理	>	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗
	参照	>	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
メニューグループカテゴリ1	管理	>	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
	参照	>	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗
メニューグループカテゴリ2	管理	>	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
	参照	>	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
メニューグループカテゴリ3	管理	>	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗
	参照	>	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗

「XMLファイルの例」をインポートした時の認可リソース

**i コラム**

上記「XMLファイルの例」のように、英語、日本語の表示名しかない場合、個人設定のロケールを中国語にすると、メニューグループカテゴリの認可リソース名はテナントロケールの表示名で表示されます。

### メニューグループカテゴリ情報

<menu-group-category-data> タグにメニューグループカテゴリ情報を記述します。

項目	ノード名	説明
メニューグループカテゴリ ID	category	システム内部で使用されるメニューグループカテゴリの識別子です。
選択可能フラグ	selectable	メニュー設定画面において、メニューグループカテゴリのセレクトボックスで選択可能かどうかを決める設定です。
		指定する値 説明
		true 選択可能
		false 選択不可能
更新可能フラグ	editable	※現在、使用されておりません。設定しないでください。

### メニューグループカテゴリ表示名情報

メニュー グループ カテゴリに設定された表示名です。

ロケールごとの表示名を保持します。

<display-names> タグ配下には、ロケールの数だけ <display-name> タグを記述します。

項目	ノード名	説明
表示名	<display-name>	メニュー グループ カテゴリに設定されている言語別の表示名です。 ロケールごとに設定可能です。
ロケールID	locale	メニュー グループ カテゴリ 表示名のロケールです。ロケールIDで設定します。

## メニュー グループ

### XMLファイルフォーマット

以下は XML ファイルの例です。

<menu-group-data> タグ内に 1 つのメニュー グループに 関連する情報をすべてを記述します。

```

<root xmlns="http://intra-mart.co.jp/im_menu/menu-group-data">
  <menu-group-data id="menu-group-1">
    <category id="im_sitemap_pc"></category>
    <menu-item menu-id="menu-group-1" sort-number="1" type="folder" method="get" use-iframe="false" use-popup="false">
      <display-names>
        <display-name locale="ja">メニュー グループ 1 </display-name>
        <display-name locale="en">menu group 1</display-name>
      </display-names>
      <menu-item menu-id="menu-item-1-1" sort-number="0" type="item" url="xxxx/xxx/xxx" method="get" use-iframe="false" use-popup="false">
        <display-names>
          <display-name locale="ja">メニュー アイテム 1 - 1 </display-name>
          <display-name locale="en">menu item 1-1</display-name>
        </display-names>
      </menu-item>
      <menu-item menu-id="menu-item-1-2" sort-number="1" type="item" url="xxxx/xxx/xxx" method="get" use-iframe="false" use-popup="false">
        <display-names>
          <display-name locale="ja">メニュー アイテム 1 - 2 </display-name>
          <display-name locale="en">menu item 1-2</display-name>
        </display-names>
      </menu-item>
      <menu-item menu-id="menu-folder-1-3" sort-number="2" type="folder" url="empty" method="get" use-iframe="false" use-popup="false">
        <display-names>
          <display-name locale="ja">メニュー フォルダ 1 - 3 </display-name>
          <display-name locale="en">menu folder 1-3</display-name>
        </display-names>
        <menu-item menu-id="menu-item-1-3-1" sort-number="1" type="item" url="xxxx/xxx/xxx" image-path="" method="get" use-iframe="true" use-popup="false" icon-16="" icon-32="" icon-48="">
          <display-names>
            <display-name locale="ja">メニュー アイテム 1 - 3 - 1 </display-name>
            <display-name locale="en">menu item 1-3-1</display-name>
          </display-names>
        </menu-item>
      </menu-item>
    </menu-group-data>
    <menu-group-data id="menu-group-2">
      <category id="im_sitemap_pc"></category>
      <menu-item menu-id="menu-group-2" sort-number="2" type="folder" url="empty" method="get" use-iframe="false" use-popup="false">
        <display-names>
          <display-name locale="ja">メニュー グループ 2 </display-name>
          <display-name locale="en">menu group 2</display-name>
        </display-names>
        <menu-item menu-id="menu-item-2-1" sort-number="0" type="item" url="xxxx/xxx/xxx" image-path="" method="get" use-iframe="false" use-popup="false" icon-16="" icon-32="" icon-48="">
          <display-names>
            <display-name locale="ja">メニュー アイテム 2 - 1 </display-name>
            <display-name locale="en">menu item 2-1</display-name>
          </display-names>
        </menu-item>
      </menu-item>
    </menu-group-data>
  </root>

```

上記のXMLファイルをインポートした場合、画面上のメニューは以下の通りです。

「XMLファイルの例」をインポートした結果



### 注意

メニューアイテムや外部メニューリンクの配下にメニューアイテムを配置しないでください。  
配置した場合、強制的にメニューfolダに変換されてしまいます。



### コラム

メニューfolダの配下にメニューアイテムを配置することで、メニューの階層構造が構築されます。  
上記の例ではメニューID `menu-folder-1-3` のメニューfolダ配下に、メニューID `menu-item-1-3-1` のメニューアイテムが設定されています。



### コラム

上記「XMLファイルの例」のように、英語、日本語の表示名しかない場合、個人設定のロケールを中国語にすると、メニューはテナントロケールの表示名で表示されます。

## メニューグループ情報

<menu-group-data> タグにメニュー情報を記述します。

項目	ノード名	説明
メニューグループID	<code>id</code>	システム内部で使用されるメニューグループの識別子です。
ルートのメニューID	<code>menu-id</code>	メニューグループ内のルートメニューの識別子です。



## 注意

メニュー グループ ID と ルート の メニュー ID は一致させる必要があります。

## メニュー グループ カテゴリ 内包 情報

メニュー グループ がどのメニュー グループ カテゴリに属しているかの情報です。

<category> タグにメニュー グループ カテゴリ 情報を記述します。

項目	ノード名	説明
メニュー グループ カテゴリ ID	<code>id</code>	メニュー グループ と 紐付けられているメニュー グループ カテゴリ の ID です。

## メニュー 情報

<menu-item> タグにメニュー 情報を記述します。

項目	ノード名	説明
メニュー ID	<code>menu-id</code>	システム 内部 で 使用されるメニュー の 識別子 です。
ソート番号	<code>sort-number</code>	メニュー が 表示されるときのソート順 を 決める 値 です。
メニュー の データ種別	<code>type</code>	メニュー の データ種別 を 指定します。
<b>指定する 値 データ種別</b>		
	<code>item</code>	メニュー アイテム
	<code>folder</code>	メニュー フォルダ
	<code>external</code>	外部 メニューリンク
URL	<code>url</code>	メニュー の リンク先 URL です。
<b>パス種別 例</b>		
	相対パス	<code>menu/sitemap</code>
	絶対パス	<code>http://www.intra-mart.jp</code>
画像ファイルパス	<code>image-path</code>	メニュー の アイコン 画像 の パス です。 コンテキスト パス 配下 の URL を 指定します。
コピー元メニュー ID	<code>original-menu-id</code>	メニュー を コピー して 作成する 場合に、 コピー 元 の メニュー ID を 指定します。
HTTP メソッド	<code>method</code>	メニュー から アクセスするときの リクエスト 方式 を 指定します。 指定できる 値 は、 <code>get</code> と <code>post</code> です。

**IFRAME使用フラグ**`use-iframe`

リンク先を表示するときにIFRAMEを使うかどうかを指定します。

**指定する****値 説明**

指定する	値 説明
true	IFRAMEを使う
false	IFRAMEを使わない

**ポップアップ使用フラグ**`use-popup`

リンク先を表示するときにポップアップで表示するかどうかを指定します。

**指定する値 説明**

指定する値 説明
true ポップアップで表示する
false ポップアップで表示しない

**説明**`<description>`

メニューの説明です。

**16x16サイズ用アイコンのアイコンID**`icon-16`

メニューの16x16サイズ用アイコンのアイコンIDです。システムが16x16サイズのアイコンを要求している画面で利用されるアイコンIDを指定します。

**32x32サイズ用アイコンのアイコンID**`icon-32`

メニューの32x32サイズ用アイコンのアイコンIDです。システムが32x32サイズのアイコンを要求している画面で利用されるアイコンIDを指定します。

**48x48サイズ用アイコンのアイコンID**`icon-48`

メニューの48x48サイズ用アイコンのアイコンIDです。システムが48x48サイズのアイコンを要求している画面で利用されるアイコンIDを指定します。

- メニューフォルダ

メニュー情報を構成する項目のうち、メニューフォルダで使用されるのは以下の通りです。

- メニューID
- ソート番号
- メニューのデータ種別
- 画像ファイルパス
- 説明
- 16x16サイズ用アイコンのアイコンID
- 32x32サイズ用アイコンのアイコンID
- 48x48サイズ用アイコンのアイコンID
- メニューアイテム

メニューアイテムでは全ての項目を使用します。

- 外部メニューリンク

メニュー情報を構成する項目のうち、使用されるのは以下の通りです。

- メニューID

- ソート番号
- メニューのデータ種別
- URL



## コラム

16x16サイズ用アイコンのアイコンIDと32x32サイズ用アイコンのアイコンIDと48x48サイズ用アイコンのアイコンIDは、intra-mart Accel Platform 2018 Spring(Skylark)以降で利用可能です。

## メニュー引数情報

<arguments> タグ配下に、メニュー引数の数だけ <argument> タグを記述します。

項目	ノード名	説明
キー	key	引数のキーを指定します。
値	value	引数の値を指定します。

## メニュー表示名情報

メニューに設定された表示名です。  
ロケールごとの表示名を保持します。

<display-names> タグ配下にロケールの数だけ <display-name> タグを記述します。

項目	ノード名	説明
ロケールID	locale	メニュー表示名のロケールです。ロケールIDで設定します。
表示名	<display-name>	メニューに設定されている言語別の表示名です。 ロケールごとに設定可能です。

## 互換性

intra-mart Accel Platform で利用するメニューのインポートファイルは、intra-mart WebPlatform/AppFramework のメニューで利用するインポートファイルと互換性がありません。

## 項目

- [更新モード](#)
  - [merge](#)
  - [replace](#)
  - [revive](#)
- [メニュー グループ カテゴリ](#)
  - [XML](#)
  - [メニュー グループ カテゴリの更新モード](#)
    - [merge](#)
    - [replace](#)
- [メニュー グループ](#)
  - [XML](#)
  - [メニュー グループ の更新モード](#)
    - [merge](#)
    - [replace](#)
    - [revive](#)
- [インポートの依存関係](#)

メニューのインポートはXML形式で行うことが可能です。

マスタ情報の新規登録、更新を行うことができます。

この章では、インポートがどのように行われるかを説明します。

また、更新モードによる更新方法の違いについて説明します。

## 更新モード

更新モードを利用することで、インポートの目的にあわせたデータの更新を行うことができます。

更新モードには、**merge**、**replace**、**revive** が提供されています。

`update-mode` 属性を指定することでモードを設定します。

各更新モードの仕様は次の通りです。

実際の動作や用途については、[メニュー グループ カテゴリ の更新モード](#)、[メニュー グループ の更新モード](#)を参照してください。

メニュー グループ カテゴリ、メニュー グループ のインポートにおいて、削除を行う更新モードは提供されていません。

### merge

インポートファイルのデータとデータベース上のデータをマージして更新します。

インポートファイルに存在しない項目は既存のデータをそのまま設定します。

### replace

インポートファイルのデータに存在しない情報は未設定の値（デフォルト値）で更新します。

インポートファイルに存在しない項目は既存のデータを未設定に更新します。

インポートファイルのデータと同じキーのデータがデータベースに存在する場合、何もしません。

同じキーのデータがデータベースに存在しない場合、インポートファイルのデータが新規追加されます。

ただし、一部のデータに関しては、インポート対象データの親となるレコードがデータベースに存在する場合、何もしません。



### 注意

更新モード **revive** は、intra-mart Accel Platform 2014 Spring(Granada) 以降で利用可能です。

## メニュー グループ カテゴリ

### XML

メニュー グループ カテゴリは [メニュー グループ](#) とは独立してインポートすることができます。

<menu-group-category-data> タグを 1 レコードとしてインポートします。

ファイルのフォーマットについては「[XML ファイルフォーマット](#)」を参照してください。

インポート時の動作をオプションとして指定が可能です。

インポート時に利用可能なオプションについては、「[インポートオプション](#)」を参照してください。

## メニュー グループ カテゴリ の 更新モード

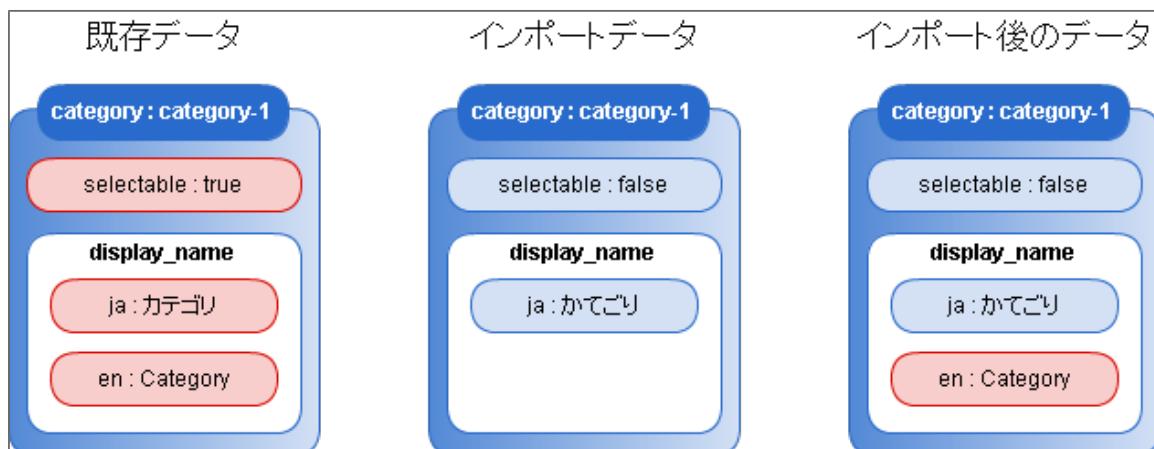
メニュー グループ カテゴリ の 更新モードには *merge*、*replace* が提供されています。

`update-mode` 属性は、<menu-group-category-data> タグに指定します。

特に指定していない場合、*merge* モードで動作します。

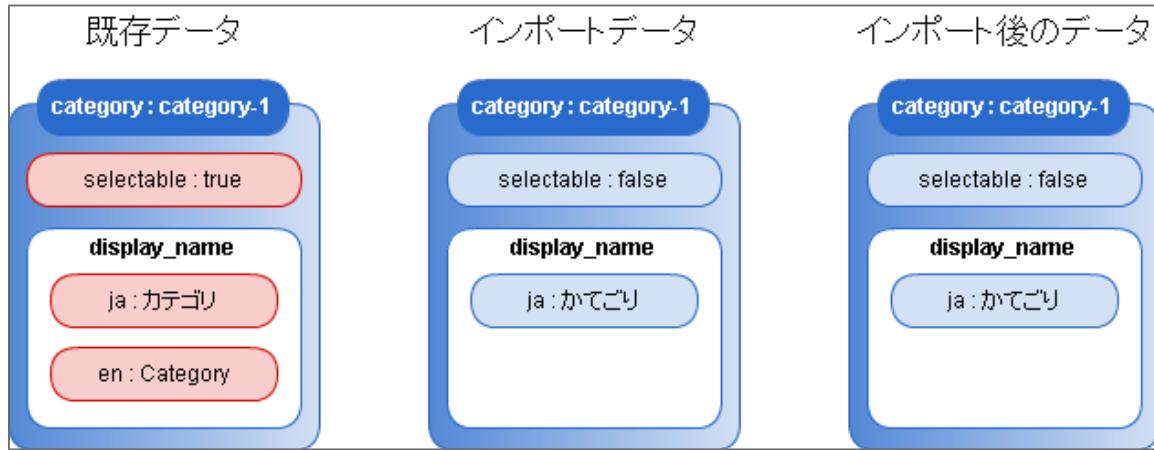
### merge

属性や表示名を更新したり、未登録ロケールの表示名を追加する場合に利用します。



### replace

既存の表示名を削除して、新たに登録する場合に利用します。



## メニュー グループ

### XML

メニュー グループのインポートは、メニュー グループをルートとした配下のメニュー階層構造全体を1件のインポートデータとして取り扱います。

[メニュー グループ カテゴリ](#)とは独立してインポートすることができます。

<menu-group-data> タグを1トランザクションとして、以下の情報をインポートします。

ファイルのフォーマットについては「[XML ファイルフォーマット](#)」を参照してください。

- [メニュー グループ 情報](#)
- メニューの情報（[メニュー情報](#)と[メニュー引数情報](#)と[メニュー表示名情報](#)）
- [メニュー グループ カテゴリ 内包情報](#)

## メニュー グループ の 更新 モード

メニュー グループ の 更新 モードには [merge](#)、[replace](#)、[revive](#) が提供されています。

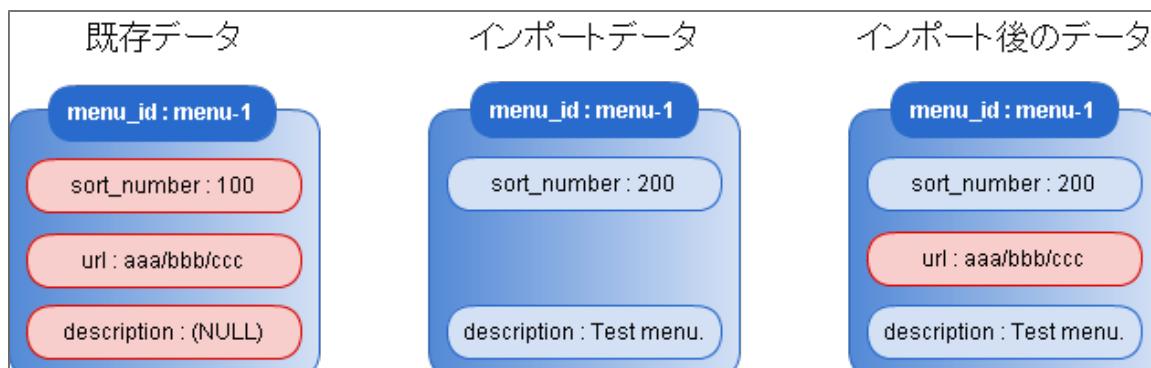
`update-mode` 属性は、<menu-group-data> タグに指定します。

特に指定していない場合、[merge](#) モードで動作します。

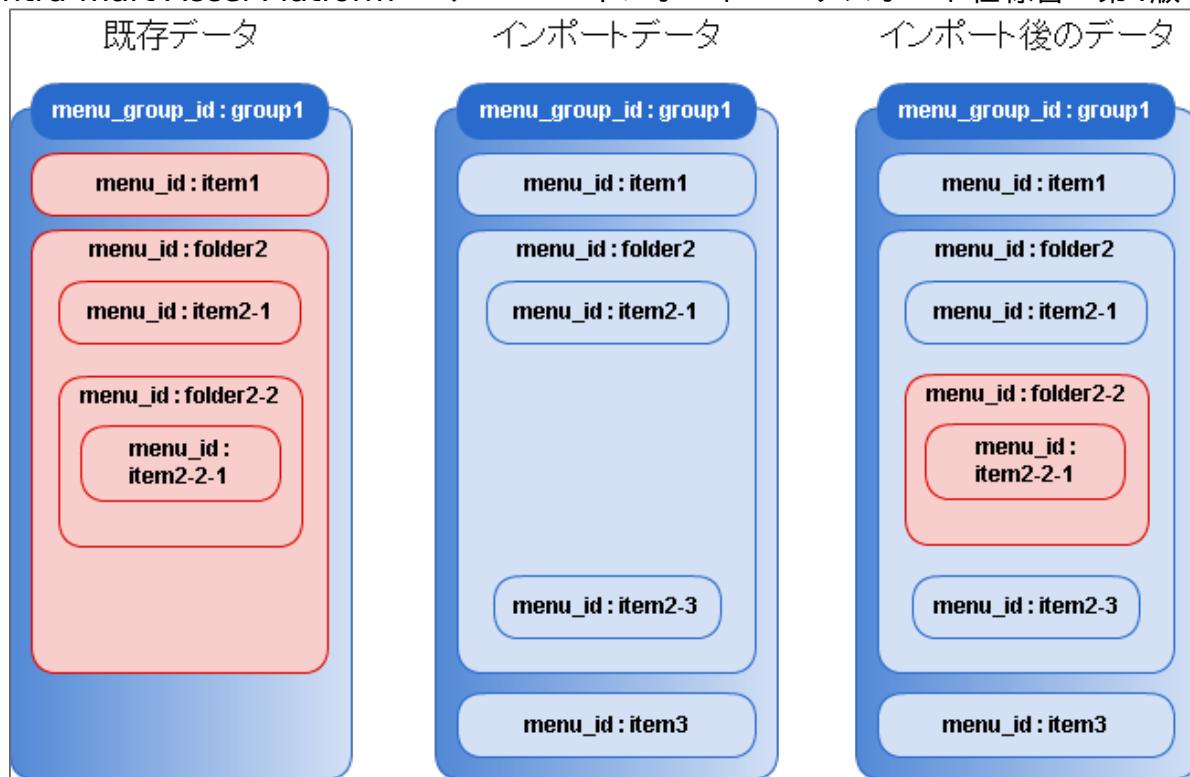
ただし、テナント環境セットアップ時におけるテナントマスタ情報のインポートで実施されるメニュー グループ のインポートについては、`update-mode` 属性を指定しない場合、[revive](#) モードで動作します。

### merge

既存のメニューを更新したり、メニューを追加したい場合に利用します。

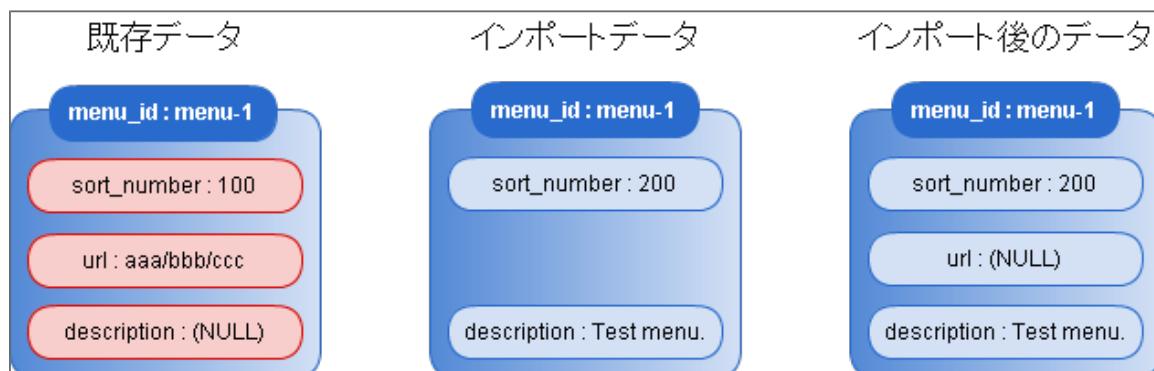


インポートファイルとデータベース上に同じメニュー グループ ID が存在する場合、配下のメニュー階層をマージします。

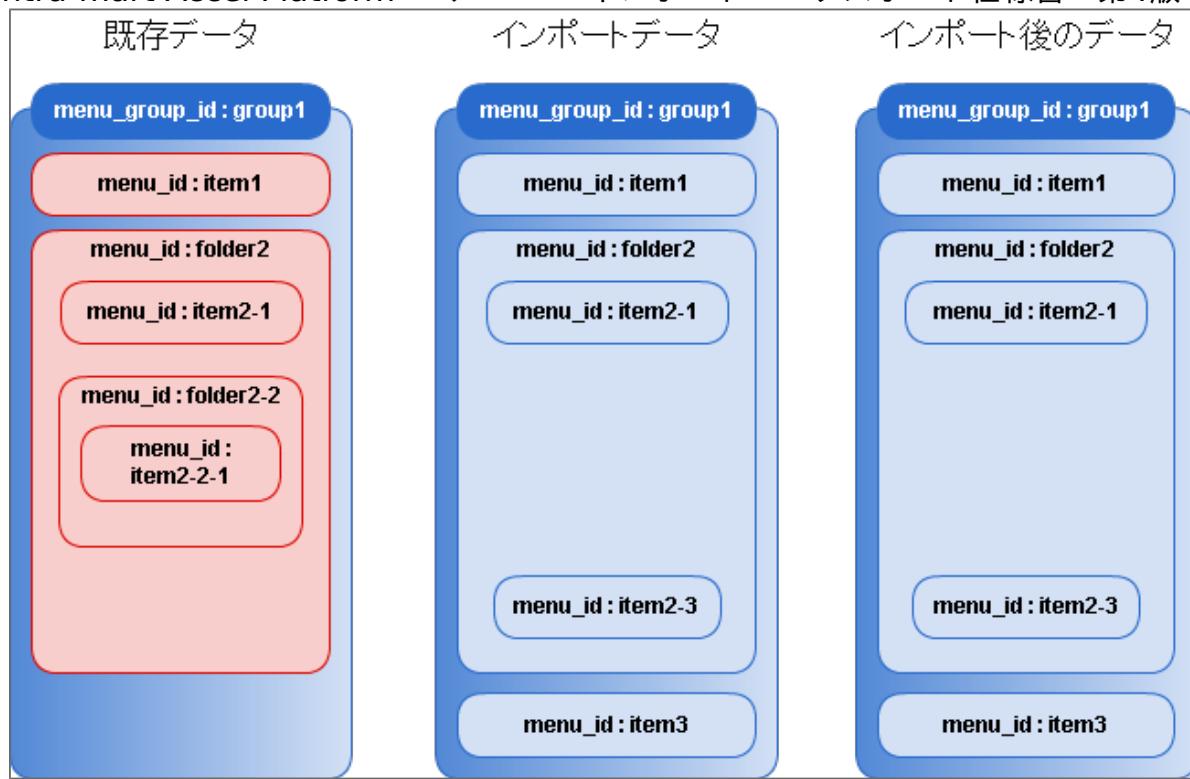


### replace

既存のメニューを削除して、インポートファイルのメニューで置き換える場合に利用します。

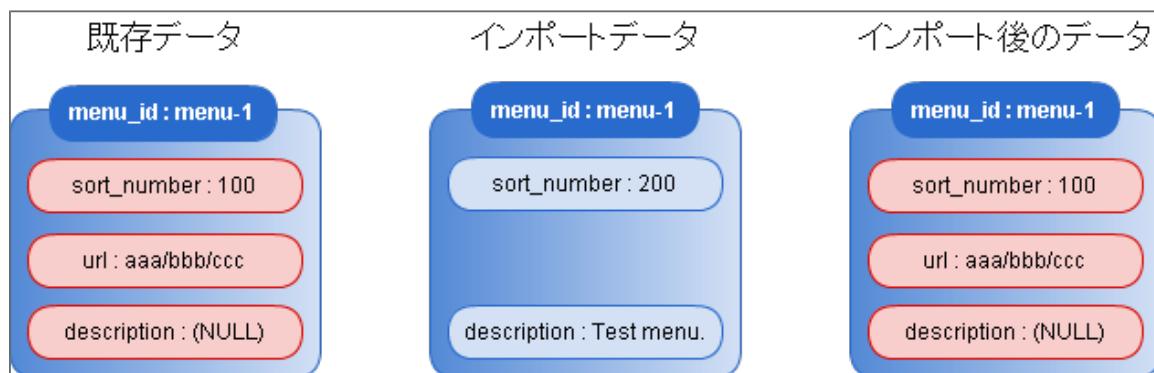


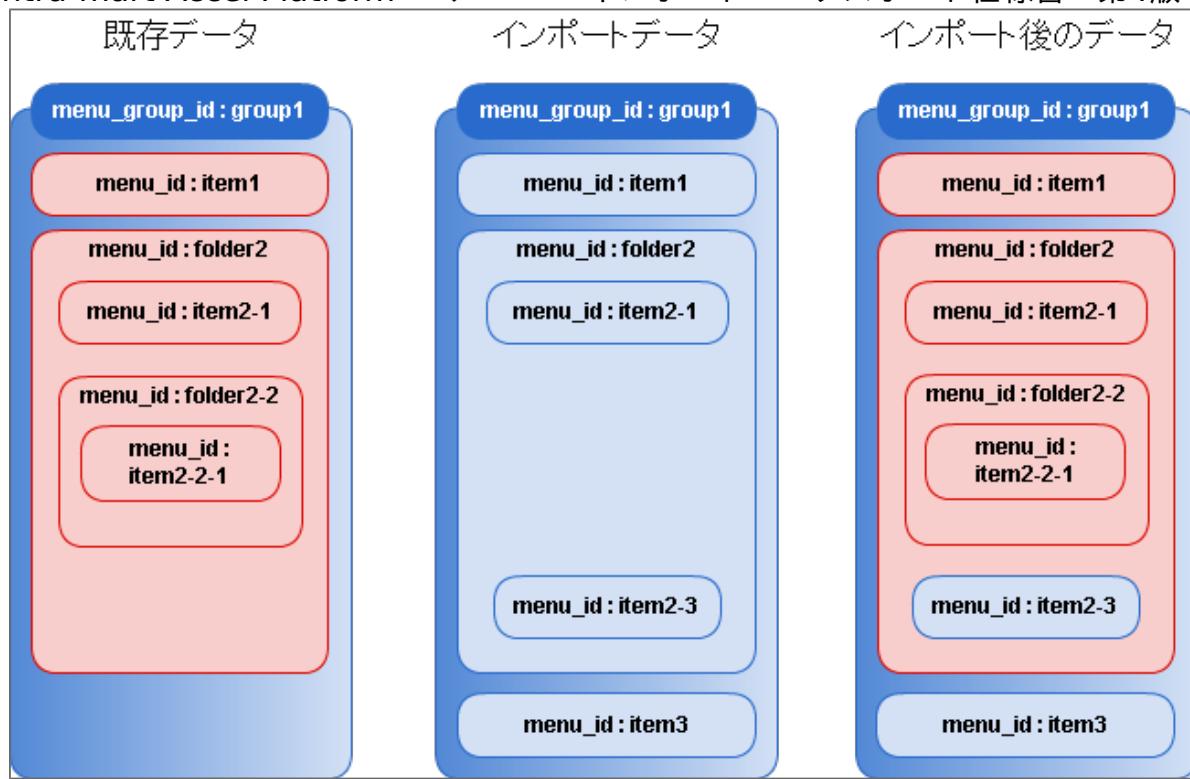
インポートファイルとデータベース上に同じメニューグループIDが存在する場合、メニュー階層の子メニューを全て削除してから、インポートファイルのデータを登録します。



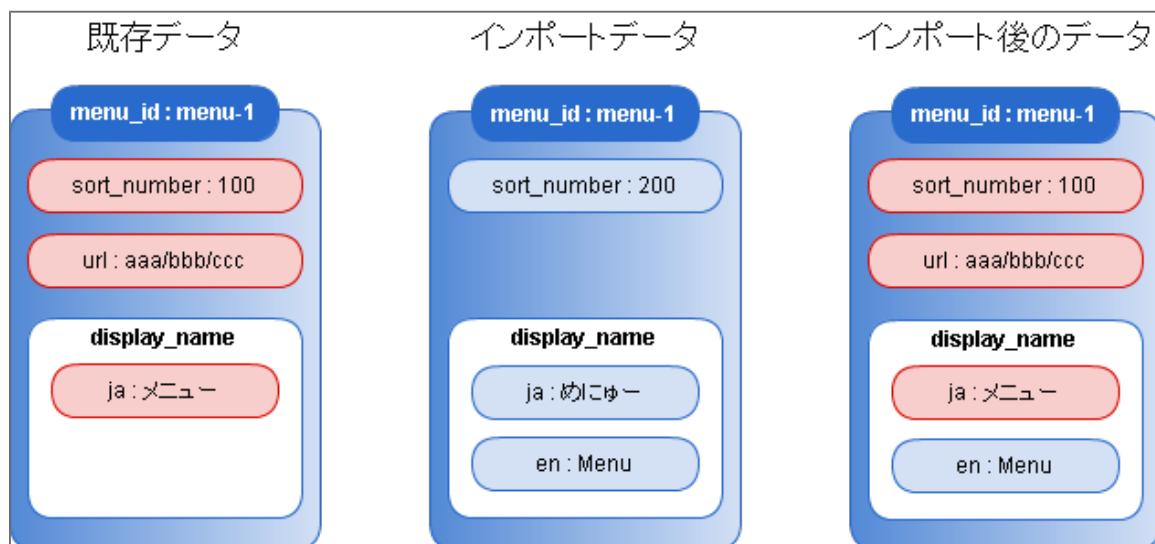
## revive

既存のメニューは更新せずに、メニューの追加のみを行いたい場合に利用します。





以下のような場合、未登録ロケールの表示名を追加します。



## インポートの依存関係

メニューグループをインポートする際、あらかじめメニューグループカテゴリを登録する必要があります。

#### 項目

- メニューグループカテゴリ
  - XML
- メニューグループ
  - XML

メニューのエクスポートはXML形式で行うことが可能です。

すべてのマスタ情報を出力することができます。

この章では、エクスポートがどのように行われるかを説明します。

## メニューグループカテゴリ

---

### XML

すべてのメニューグループカテゴリとそれに関連する情報をXML形式でファイルに出力します。

ファイルのフォーマットについては「[XMLファイルフォーマット](#)」を参照してください。

## メニューグループ

---

### XML

すべてのメニューグループと、その配下に登録されているメニューに関連する情報をXML形式でファイルに出力します。

ファイルのフォーマットについては「[XMLファイルフォーマット](#)」を参照してください。

また、エクスポート時の動作をオプションとして指定が可能です。

エクスポート時に利用可能なオプションについては、「[エクスポートオプション](#)」を参照してください。

### 項目

- インポートオプション
- エクスポートオプション

インポート・エクスポートで扱うファイルなどの設定や、インポート・エクスポート処理の動作を変更するためのオプションが用意されています。

この章では、インポート・エクスポート時に使用できるオプションについて説明します。

## インポートオプション

この項では、インポートを行う際に使用可能なオプションについて説明します。



### コラム

この項で説明されているインポートオプションは、メニューグループカテゴリ、メニューグループの両方で使用可能です。

### 設定可能項目一覧

名前	キー名	型	デフォルト値	導入バージョン
エンコーディング	encoding	文字列	UTF-8	2012 Autumn
ファイルパス	file	文字列	(なし)	2012 Autumn
コミット件数	commit-count	数値	0	2012 Autumn
XML検証フラグ	validate-xml	真偽値	true	2012 Autumn

#### エンコーディング

##### キー名 encoding

インポートするXMLファイルの文字エンコーディングを指定します。

#### ファイルパス

##### キー名 file

インポートするXMLファイルのパス（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。

#### コミット件数

## キー名 commit-count

インポート処理で、コミットを行うまでのデータ件数を指定します。

コミット件数に「0」（デフォルト値）が指定された場合は、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。



### 注意

`commit-count` を指定した場合、インポート実行元で管理しているトランザクションがコミットされる可能性があります。

## XML検証フラグ

### キー名 validate-xml

インポートするXMLファイルの構文を検証するかどうかを指定します。

#### 指定する値 説明

true XML構文の検証を行います。（デフォルト値）

false XML構文の検証を行いません。

## エクスポートオプション

この項では、エクスポートを行う際に使用可能なオプションについて説明します。



### コラム

この項で説明されているエクスポートオプションは、メニューグループカテゴリ、メニューグループの両方で使用可能です。

## 設定可能項目一覧

名前	キー名	型	デフォルト値	導入バージョン
エンコーディング	encoding	文字列	UTF-8	2012 Autumn
ファイルパス	file	文字列	(なし)	2012 Autumn
XML整形フラグ	format-xml	真偽値	false	2012 Autumn
ルートタグ名	root-tag-name	文字列	root	2012 Autumn
書き込み件数	flush-count	数値	5000	2012 Autumn

キー名 encoding

エクスポートするXMLファイルの文字エンコーディングを指定します。

ファイルパス

キー名 file

エクスポートするXMLファイルのパス（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。

XML整形フラグ

キー名 format-xml

エクスポートするXMLファイルを整形するかどうかを指定します。

指定する値 説明

true XMLの整形を行います。

false XMLの整形を行いません。（デフォルト  
値）

ルートタグ名

キー名 root-tag-name

エクスポートするXMLファイルのルートタグ名を指定します。

書き込み件数

キー名 flush-count

エクスポートするXMLファイルに一度に書き込むデータ件数を指定します。

## 項目

- ジョブスケジューラを利用する
  - ジョブ
    - インポート
    - エクスポート
  - ジョブネット
    - インポート
    - エクスポート
- Javaから実行する
  - インポート
  - エクスポート
- スクリプト開発モデルプログラムから実行する
  - インポート
  - エクスポート

この項では、メニュー グループ カテゴリのインポート・エクスポートを実行する手段を紹介します。



### 注意

インポートしたデータをエクスポートする場合、またはエクスポートしたデータをインポートする場合は、「エンコーディング」などの対応するオプションは同じ値を指定する必要があります。

## ジョブスケジューラを利用する

ジョブスケジューラの機能を利用してインポート・エクスポートを実行する方法を紹介します。

ジョブスケジューラの詳細については「[ジョブスケジューラ仕様書](#)」を参照してください。

intra-mart Accel Platform ではメニュー グループ カテゴリのインポート・エクスポートを行うためのジョブ・ジョブネットを提供しています。

この項では、intra-mart Accel Platform が標準で提供しているメニュー グループのインポート・エクスポートを行うジョブ・ジョブネットの情報を紹介します。

### ジョブ

#### インポート

ジョブカテゴリ テナントマスタ > インポート

ジョブID menu-group-category-import

ジョブ名 メニュー グループ カテゴリ インポート

#### エクスポート

ジョブカテゴリ テナントマスタ > エクスポート

ジョブID menu-group-category-export

ジョブ名 メニューグループカテゴリエクスポート

## ジョブネット

### インポート

ジョブネットカテゴリ テナントマスタ > インポート

ジョブネットID menu-group-category-import-jobnet

ジョブネット名 メニューグループカテゴリインポート

### エクスポート

ジョブネットカテゴリ テナントマスタ > エクスポート

ジョブネットID menu-group-category-export-jobnet

ジョブネット名 メニューグループカテゴリエクスポート



#### コラム

ジョブスケジューラ利用時のオプションについて

ジョブスケジューラを利用してインポート・エクスポートを実行する場合は、ジョブ・ジョブネットのパラメータに「[オプション](#)」を指定します。



#### コラム

ジョブスケジューラ利用時のトランザクション管理について

ジョブスケジューラを利用したインポートでは、オプション `commit-count` を指定しない場合、インポート処理が完了後に一括してコミットを行います。

必要に応じてオプション `commit-count` の値を変更してご利用ください。

## Javaから実行する

JavaのAPIを利用してインポート・エクスポートを実行する方法を紹介します。

### インポート

`DataImportExecutor#importData(String, InputStream, Map)` を利用してインポートを行います。

- 完全修飾クラス名

```
jp.co.intra_mart.foundation.data.importer.DataImportExecutor
```

第1引数にはインポータIDを指定します。インポータIDは以下を利用します。

```
jp.co.intra_mart.import.StandardMenuGroupCategoryXmlImporter
```

第2引数にはインポート元を `InputStream` で指定します。

第3引数にはインポートオプションを `Map<String, Object>` で指定します。

詳細は、「[インポートオプション](#)」を参照してください。

`DataImportExecutor` の詳細については「[DataImportExecutorクラスのAPIドキュメント](#)」を参照してください。

以下はインポートを行うサンプルプログラムです。

```
package sample;

import java.io.IOException;
import java.io.InputStream;
import java.util.HashMap;
import java.util.Map;

import jp.co.intra_mart.foundation.data.OptionKeyName;
import jp.co.intra_mart.foundation.data.exception.DataImporterException;
import jp.co.intra_mart.foundation.data.importer.DataImportExecutor;
import jp.co.intra_mart.foundation.service.client.PublicStorage;

/**
 * メニューグループカテゴリのインポートを行うクラスです。
 */
public class MenuGroupCategoryImporter {

    private static final String IMPORTER_ID =
"jp.co.intra_mart.import.StandardMenuGroupCategoryXmlImporter";

    /**
     * メニューグループカテゴリのインポートを行います。
     * @throws DataImporterException インポートで何らかの例外が発生した場合。
     */
    public void doImport() throws DataImporterException {
        final DataImportExecutor executor = new DataImportExecutor();
        final Map<String, Object> options = new HashMap<String, Object>();
        options.put(OptionKeyName.ENCODING.value(), "UTF-8");
        options.put(OptionKeyName.VALIDATE_XML.value(), true);
        options.put(OptionKeyName.COMMIT_COUNT.value(), 100);
        // パブリックストレージ直下のmenu-group-category.xmlを選択
        final PublicStorage storage = new PublicStorage("menu-group-category.xml");
        try {
            // PublicStorageからInputStreamを取得
            final InputStream stream = storage.open();
            try {
                executor.importData(IMPORTER_ID, stream, options);
            } finally {
                stream.close();
            }
        } catch (final IOException e) {
            throw new DataImporterException(e);
        }
    }
}
```



### 注意

第2引数の `InputStream` を指定した場合は、オプション `file` は利用できません。  
オプション `file` を利用したい場合は、第2引数に `null` を指定してください。



### 注意

メニュー グループ カテゴリ のインポートには認可による管理権限が必要です。  
詳しくは [メニューにおける認可の仕様](#) を参照してください。

## エクスポート

`DataExportExecutor#exportData(String, OutputStream, Map)` を利用してエクスポートを行います。

- 完全修飾クラス名

```
jp.co.intra_mart.foundation.data.exporter.DataExportExecutor
```

第1引数にはエクスポートIDを指定します。エクスポートIDは以下を利用します。

```
jp.co.intra_mart.import.StandardMenuGroupCategoryXmlExporter
```

第2引数にはエクスポート先を `OutputStream` で指定します。

第3引数にはエクスポートオプションを `Map<String, Object>` で指定します。

詳細は、「[エクスポートオプション](#)」を参照してください。

`DataExportExecutor` の詳細については「[DataExportExecutorクラスのAPIドキュメント](#)」を参照してください。

以下はエクスポートを行うサンプルプログラムです。

```

package sample;

import java.io.IOException;
import java.io.OutputStream;
import java.util.HashMap;
import java.util.Map;

import jp.co.intra_mart.foundation.data.OptionKeyName;
import jp.co.intra_mart.foundation.data.exception.DataExporterException;
import jp.co.intra_mart.foundation.data.exporter.DataExportExecutor;
import jp.co.intra_mart.foundation.service.client.file.PublicStorage;

/**
 * メニューグループカテゴリのエクスポートを行うクラスです。
 */
public class MenuGroupCategoryExporter {

    private static final String EXPORTER_ID =
"jp.co.intra_mart.export.StandardMenuGroupCategoryXmlExporter";

    /**
     * メニューグループカテゴリのエクスポートを行います。
     * @throws DataExporterException エクスポートで何らかの例外が発生した場合。
     */
    public void doExport() throws DataExporterException {
        final DataExportExecutor executor = new DataExportExecutor();
        final Map<String, Object> options = new HashMap<String, Object>();
        options.put(OptionKeyName.ENCODING.value(), "UTF-8");
        options.put(OptionKeyName.FORMAT_XML.value(), false);
        options.put(OptionKeyName.FETCH_COUNT.value(), 10);
        // パブリックストレージ直下のmenu-group-category.xmlを選択
        final PublicStorage storage = new PublicStorage("menu-group-category.xml");
        try {
            // PublicStorageからOutputStreamを取得
            final OutputStream stream = storage.create();
            try {
                executor.exportData(EXPORTER_ID, stream, options);
            } finally {
                stream.close();
            }
        } catch (final IOException e) {
            throw new DataExporterException(e);
        }
    }
}

```



### 注意

第2引数の `OutputStream` を指定した場合は、オプション `file` は利用できません。  
オプション `file` を利用したい場合は、第2引数に `null` を指定してください。

## スクリプト開発モデルプログラムから実行する

スクリプト開発モデルのAPIを利用してインポート・エクスポートを実行する方法を紹介します。

`DataImportExecutor#importData(String, ByteReader, Object)` を利用してインポートを行います。

第1引数にはインポータIDを指定します。インポータIDは以下を利用します。

`jp.co.intra_mart.import.StandardMenuGroupCategoryXmlImporter`

第2引数にはインポート元を `ByteReader` で指定します。

第3引数にはインポートオプションをObject形式で指定します。

詳細は、「[インポートオプション](#)」を参照してください。

`DataImportExecutor` の詳細については「[DataImportExecutorオブジェクトのAPIドキュメント](#)」を参照してください。

以下はインポートを行うサンプルプログラムです。

```
var IMPORTER_ID = 'jp.co.intra_mart.import.StandardMenuGroupCategoryXmlImporter';

/**
 * メニューグループカテゴリのインポートを行います。
 */

function doImport() {
    var executor = new DataImportExecutor();
    var options = {
        'encoding': 'UTF-8',
        'validate-xml': true,
        'commit-count': 100
    };
    // パブリックストレージ直下のmenu-group-category.xmlを選択
    var storage = new PublicStorage('menu-group-category.xml');
    // ファイルからデータをインポート
    storage.openAsBinary(function(reader, error) {
        if (error) {
            // ファイルの読み込みに失敗->例外処理
            Logger.getLogger().error(error.message);
            return;
        }

        var result = executor.importData(IMPORTER_ID, reader, options);
        if (result.error) {
            // インポート失敗->例外処理
            Logger.getLogger().error(result.errorMessage);
        }
    });
}
```



### 注意

第2引数の `ByteReader` を指定した場合は、オプション `file` は利用できません。

オプション `file` を利用したい場合は、第2引数に `null` を指定してください。



### 注意

メニュー グループ カテゴリ のインポートには認可による管理権限が必要です。  
詳しくは [メニューにおける認可の仕様](#) を参照してください。

## エクスポート

`DataExportExecutor#importData(String, ByteWriter, Object)` を利用してエクスポートを行います。

第1引数にはエクスポートIDを指定します。| エクスポートIDは以下を利用します。

`jp.co.intra_mart.import.StandardMenuGroupCategoryXmlExporter`

第2引数にはエクスポート元を `ByteWriter` で指定します。

第3引数にはエクスポートオプションをObject形式で指定します。

詳細は、「[エクスポートオプション](#)」を参照してください。

`DataExportExecutor` の詳細については「[DataExportExecutorオブジェクトのAPIドキュメント](#)」を参照してください。

以下はインポートを行うサンプルプログラムです。

```
var EXPORTER_ID = 'jp.co.intra_mart.export.StandardMenuGroupCategoryXmlExporter';

/**
 * メニューグループ カテゴリ のエクスポートを行います。
 */
function doExport() {
    var executor = new DataExportExecutor();
    var options = {
        'encoding': 'UTF-8',
        'format-xml': false,
        'fetch-count': 10
    };
    // パブリックストレージ直下のmenu-group-category.xmlを選択
    var storage = new PublicStorage('menu-group-category.xml');
    // ファイルにデータをエクスポート
    storage.createAsBinary(function(writer, error) {
        if (error) {
            // ファイルの作成に失敗->例外処理
            Logger.getLogger().error(error.message);
            return;
        }

        var result = executor.exportData(EXPORTER_ID, writer, options);
        if (result.error) {
            // エクスポートに失敗->例外処理
            Logger.getLogger().error(result.errorMessage);
        }
    });
}
```



### 注意

第2引数の `ByteWriter` を指定した場合は、オプション `file` は利用できません。

オプション `file` を利用したい場合は、第2引数に `null` を指定してください。

## 項目

- ジョブスケジューラを利用する
  - ジョブ
    - インポート
    - エクスポート
  - ジョブネット
    - インポート
    - エクスポート
- Javaから実行する
  - インポート
  - エクスポート
- スクリプト開発モデルプログラムから実行する
  - インポート
  - エクスポート

この項では、メニュー グループのインポート・エクスポートを実行する手段を紹介します。



### 注意

インポートしたデータをエクスポートする場合、または、エクスポートしたデータをインポートする場合は、「エンコーディング」などの対応するオプションは同じ値を指定する必要があります。

## ジョブスケジューラを利用する

ジョブスケジューラの機能を利用してインポート・エクスポートを実行する方法を紹介します。

ジョブスケジューラの詳細については「[ジョブスケジューラ仕様書](#)」を参照してください。

intra-mart Accel Platform ではメニュー グループ カテゴリ、メニュー グループ のインポート・エクスポートを行うためのジョブ・ジョブネットを提供しています。

この項では、intra-mart Accel Platform が標準で提供しているメニュー グループ カテゴリ、メニュー グループ のインポート・エクスポートを行うジョブ・ジョブネットの情報を紹介します。

### ジョブ

#### インポート

ジョブカテゴリ テナントマスター > インポート

ジョブID menu-group-import

ジョブ名 メニュー グループ インポート

#### エクスポート

ジョブカテゴリ テナントマスター > エクスポート

ジョブID menu-group-export

ジョブ名 メニューグループエクスポート

## ジョブネット

### インポート

ジョブネットカテゴリ テナントマスタ > インポート

ジョブネットID menu-group-import-jobnet

ジョブネット名 メニューグループインポート

### エクスポート

ジョブネットカテゴリ テナントマスタ > エクスポート

ジョブネットID menu-group-export-jobnet

ジョブネット名 メニューグループエクスポート



#### コラム

ジョブスケジューラ利用時のオプションについて

ジョブスケジューラを利用してインポート・エクスポートを実行する場合は、ジョブ・ジョブネットのパラメータに「[オプション](#)」を指定します。



#### コラム

ジョブスケジューラ利用時のトランザクション管理について

ジョブスケジューラを利用したインポートでは、オプション `commit-count` を指定しない場合、インポート処理が完了後に一括してコミットを行います。

必要に応じてオプション `commit-count` の値を変更してご利用ください。

## Javaから実行する

JavaのAPIを利用してインポート・エクスポートを実行する方法を紹介します。

### インポート

`DataImportExecutor#importData(String, InputStream, Map)` を利用してインポートを行います。

- 完全修飾クラス名

```
jp.co.intra_mart.foundation.data.importer.DataImportExecutor
```

第1引数にはインポータIDを指定します。インポータIDは以下を利用します。

```
jp.co.intra_mart.import.StandardMenuGroupXmlImporter
```

第2引数にはインポート元を `InputStream` で指定します。

第3引数にはインポートオプションを `Map<String, Object>` で指定します。

詳細は、「[インポートオプション](#)」を参照してください。

`DataImportExecutor` の詳細については「[DataImportExecutorクラスのAPIドキュメント](#)」を参照してください。

以下はインポートを行うサンプルプログラムです。

```
package sample;

import java.io.IOException;
import java.io.InputStream;
import java.util.HashMap;
import java.util.Map;

import jp.co.intra_mart.foundation.data.OptionKeyName;
import jp.co.intra_mart.foundation.data.exception.DataImporterException;
import jp.co.intra_mart.foundation.data.importer.DataImportExecutor;
import jp.co.intra_mart.foundation.service.client.file.PublicStorage;

/**
 * メニューグループのインポートを行うクラスです。
 */
public class MenuGroupImporter {

    private static final String IMPORTER_ID = "jp.co.intra_mart.import.StandardMenuGroupXmlImporter";

    /**
     * メニューグループのインポートを行います。
     * @throws DataImporterException インポートで何らかの例外が発生した場合。
     */
    public void doImport() throws DataImporterException {
        final DataImportExecutor executor = new DataImportExecutor();
        final Map<String, Object> options = new HashMap<String, Object>();
        options.put(OptionKeyName.ENCODING.value(), "UTF-8");
        options.put(OptionKeyName.VALIDATE_XML.value(), true);
        options.put(OptionKeyName.COMMIT_COUNT.value(), 100);
        // パブリックストレージ直下のmenu-group.xmlを選択
        final PublicStorage storage = new PublicStorage("menu-group.xml");
        try {
            // PublicStorageからInputStreamを取得
            final InputStream stream = storage.open();
            try {
                executor.importData(IMPORTER_ID, stream, options);
            } finally {
                stream.close();
            }
        } catch (final IOException e) {
            throw new DataImporterException(e);
        }
    }
}
```

**注意**

第2引数の `InputStream` を指定した場合は、オプション `file` は利用できません。  
オプション `file` を利用したい場合は、第2引数に `null` を指定してください。

**注意**

メニュー グループのインポートを行う場合は、必ずメニュー グループの登録、および、更新が可能なユーザーで行ってください。  
詳しくは [メニューにおける認可の仕様](#) を参照してください。

## エクスポート

`DataExportExecutor#exportData(String, OutputStream, Map)` を利用してエクスポートを行います。

- 完全修飾クラス名

```
jp.co.intra_mart.foundation.data.exporter.DataExportExecutor
```

第1引数にはエクスポートIDを指定します。 | エクスポートIDは以下を利用します。

```
jp.co.intra_mart.import.StandardMenuGroupXmlExporter
```

第2引数にはエクスポート先を `OutputStream` で指定します。

第3引数にはエクスポートオプションを `Map<String, Object>` で指定します。  
詳細は、「[エクスポートオプション](#)」を参照してください。

`DataExportExecutor` の詳細については「[DataExportExecutorクラスのAPIドキュメント](#)」を参照してください。

以下はエクスポートを行うサンプルプログラムです。

```

package sample;

import java.io.IOException;
import java.io.OutputStream;
import java.util.HashMap;
import java.util.Map;

import jp.co.intra_mart.foundation.data.OptionKeyName;
import jp.co.intra_mart.foundation.data.exception.DataExporterException;
import jp.co.intra_mart.foundation.data.exporter.DataExportExecutor;
import jp.co.intra_mart.foundation.service.client.file.PublicStorage;

/**
 * ロールのエクスポートを行うクラスです。
 */
public class MenuGroupExporter {

    private static final String EXPORTER_ID = "jp.co.intra_mart.export.StandardMenuGroupXmlExporter";

    /**
     * メニューグループのエクスポートを行います。
     * @throws DataExporterException エクスポートで何らかの例外が発生した場合。
     */
    public void doExport() throws DataExporterException {
        final DataExportExecutor executor = new DataExportExecutor();
        final Map<String, Object> options = new HashMap<String, Object>();
        options.put(OptionKeyName.ENCODING.value(), "UTF-8");
        options.put(OptionKeyName.FORMAT_XML.value(), false);
        options.put(OptionKeyName.FETCH_COUNT.value(), 10);
        // パブリックストレージ直下のmenu-group.xmlを選択
        final PublicStorage storage = new PublicStorage("menu-group.xml");
        try {
            // PublicStorageからOutputStreamを取得
            final OutputStream stream = storage.create();
            try {
                executor.exportData(EXPORTER_ID, stream, options);
            } finally {
                stream.close();
            }
        } catch (final IOException e) {
            throw new DataExporterException(e);
        }
    }
}

```



### 注意

第2引数の `OutputStream` を指定した場合は、オプション `file` は利用できません。  
オプション `file` を利用したい場合は、第2引数に `null` を指定してください。

## スクリプト開発モデルプログラムから実行する

スクリプト開発モデルのAPIを利用してインポート・エクスポートを実行する方法を紹介します。

`DataImportExecutor#importData(String, ByteReader, Object)` を利用してインポートを行います。

第1引数にはインポータIDを指定します。インポータIDは以下を利用します。

`jp.co.intra_mart.import.StandardMenuGroupXmlImporter`

第2引数にはインポート元を `ByteReader` で指定します。

第3引数にはインポートオプションをObject形式で指定します。

詳細は、「[インポートオプション](#)」を参照してください。

`DataImportExecutor` の詳細については「[DataImportExecutorオブジェクトのAPIドキュメント](#)」を参照してください。

以下はインポートを行うサンプルプログラムです。

```
var IMPORTER_ID = 'jp.co.intra_mart.import.StandardMenuGroupXmlImporter';

/**
 * メニューグループのインポートを行います。
 */

function doImport() {
    var executor = new DataImportExecutor();
    var options = {
        'encoding': 'UTF-8',
        'validate-xml': true,
        'commit-count': 100
    };
    // パブリックストレージ直下のmenu-group.xmlを選択
    var storage = new PublicStorage('menu-group.xml');
    // ファイルからデータをインポート
    storage.openAsBinary(function(reader, error) {
        if (error) {
            // ファイルの読み込みに失敗->例外処理
            Logger.getLogger().error(error.message);
            return;
        }

        var result = executor.importData(IMPORTER_ID, reader, options);
        if (result.error) {
            // インポート失敗->例外処理
            Logger.getLogger().error(result.errorMessage);
        }
    });
}
```



### 注意

第2引数の `ByteReader` を指定した場合は、オプション `file` は利用できません。

オプション `file` を利用したい場合は、第2引数に `null` を指定してください。

**注意**

メニュー グループのインポートを行う場合は、必ずメニュー グループの登録、および、更新が可能なユーザーで行ってください。  
詳しくは [メニューにおける認可の仕様](#) を参照してください。

## エクスポート

`DataExportExecutor#importData(String, ByteWriter, Object)` を利用してエクスポートを行います。

第1引数にはエクスポートIDを指定します。エクスポートIDは以下を利用します。

`jp.co.intra_mart.import.StandardMenuGroupXmlExporter`

第2引数にはエクスポート元を `ByteWriter` で指定します。

第3引数にはエクスポートオプションをObject形式で指定します。

詳細は、「[エクスポートオプション](#)」を参照してください。

`DataExportExecutor` の詳細については「[DataExportExecutorオブジェクトのAPIドキュメント](#)」を参照してください。

以下はインポートを行うサンプルプログラムです。

```
var EXPORTER_ID = 'jp.co.intra_mart.export.StandardMenuGroupXmlExporter';

/**
 * メニューグループのエクスポートを行います。
 */

function doExport() {
    var executor = new DataExportExecutor();
    var options = {
        'encoding': 'UTF-8',
        'format-xml': false,
        'fetch-count': 10
    };
    // パブリックストレージ直下のmenu-group.xmlを選択
    var storage = new PublicStorage('menu-group.xml');
    // ファイルにデータをエクスポート
    storage.createAsBinary(function(writer, error) {
        if (error) {
            // ファイルの作成に失敗->例外処理
            Logger.getLogger().error(error.message);
            return;
        }

        var result = executor.exportData(EXPORTER_ID, writer, options);
        if (result.error) {
            // エクスポートに失敗->例外処理
            Logger.getLogger().error(result.errorMessage);
        }
    });
}
```



### 注意

第2引数の `ByteWriter` を指定した場合は、オプション `file` は利用できません。

オプション `file` を利用したい場合は、第2引数に `null` を指定してください。

項目

- メニューインポートデータ定義書

## メニューインポートデータ定義書

---

メニューグループカテゴリ、メニューグループのインポート・エクスポートで利用するデータの詳細はメニューインポートデータ定義書として提供します。

メニューインポートデータ定義書は以下からダウンロードできます。

「[im\\_menu\\_import\\_export\\_definition.xls](#)」