Copyright © 2013 NTT DATA INTRAMART CORPORATION

目次

- 改訂情報
- はじめに
  - 本書の目的
  - 対象読者
  - 本書の構成
  - 用語
    - ショートモジュールID
    - テナントマスタ情報
- セットアップ 概要
  - セットアップとは
  - セットアップの種類
  - セットアップで実行可能な処理
  - セットアップの定義単位
  - セットアップを実行するために用意すべき資源
- SchemaUpdate
  - SchemaUpdate とは
  - スキーマバージョン とは
  - SchemaUpdate の適用対象
  - 動作仕様
  - 差分セットアップ
- システムデータベースセットアップ
  - システムデータベースセットアップ とは
  - システムデータベースセットアップ 仕様
    - システムデータベースセットアップ で実行可能な処理
    - システムデータベースセットアップ 実行準備
    - システムデータベースセットアップ 実行方法
    - システムデータベースセットアップ 実行順序
    - システムデータベースセットアップ におけるトランザクション制御
    - 例外発生時の動作
- テナント環境セットアップ
  - テナント環境セットアップ とは
  - テナント環境セットアップ 仕様
    - テナント環境セットアップ で実行可能な処理
    - セットアップ設定ファイル 実行準備
    - テナント環境セットアップ 実行方法
    - テナント環境セットアップ 実行順序
    - テナント環境セットアップ におけるトランザクション制御
    - 例外発生時の動作
- サンプルデータセットアップ
  - サンプルデータセットアップ とは
  - サンプルデータセットアップ 仕様
    - サンプルデータセットアップ で実行可能な処理
    - サンプルデータセットアップ 実行準備
    - サンプルデータセットアップ 実行方法
    - サンプルデータセットアップ 実行順序
    - サンプルデータセットアップ におけるトランザクション制御
    - 例外発生時の動作
- セットアップ実行結果ログ
  - セットアップ実行結果ログ
    - 出力先

    - 出力内容
    - 出力例
  - インポート処理結果ログ
    - 出力先
- 付録
  - モジュールIDの制限事項
  - セットアップ用SQLファイル / インポートDDL / インポートDML 記法
  - セットアップにおけるマルチデータベース対応
  - IM-Propagation を利用した テナント環境セットアップ実行データの受信
  - フォーマットファイル(xsd)
    - セットアップ設定
    - セットアップ実行結果ログ
  - インポートするデータの依存関係
    - テナント管理機能
  - インポートするデータの登録方針
    - 更新モードについて
    - デフォルトの更新モード
  - 前処理プログラム サンプル
    - サンプル内容
    - Javaクラス サンプル
    - サーバサイドJavaScript サンプル

改訂情報

変更年月日	変更内容
2013-04-01	初版
2014-01-01	第2版　下記を追加・変更しました ■ 「*インポート処理結果ログ* 」にエラー原因の追跡についての説明を追記 ■ 「*インポートするデータの依存関係* 」を追加
2014-04-01	第3版　下記を追加・変更しました ■ 「*セットアップ実行結果ログ* 」ー「*出力先* 」のログ出力先情報を変更 ■ 「*セットアップ 概要* 」ー「*テナント環境セットアップ で実行可能な処理* 」に「前処理」を追記 ■ 「*テナント環境セットアップ* 」ー「*前処理プログラム の作成と配置* 」の説明を追記 ■ 「*サンプルデータセットアップ* 」ー「*前処理プログラム の作成と配置* 」の説明を追記 ■ 「*前処理プログラム サンプル* 」を追加 ■ 「*インポートするデータの登録方針* 」を追加
2014-08-01	第4版　下記を追加・変更しました ■ 「*付録* 」ー「*モジュールIDの制限事項* 」にショートモジュールIDの確認方法へのリンクを追加
2016-12-01	第5版　下記を変更しました ■ DB2に関する記述を削除

はじめに

本書の目的

本書ではシステムデータベースセットアップ、テナント環境セットアップおよびサンプルデータセットアップの仕様、実行方法および資材の管理方法などについて説明します。

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を理解している
- 以下の条件を満たす、セットアップを利用するモジュールの開発者
  - モジュールの概念と開発方法を理解している開発者
  - Java または サーバサイドJavaScript を理解している開発者

本書の構成

本書は以下のような構成となっています。

- *セットアップ 概要*
  この章ではセットアップの概要について説明します。
- *SchemaUpdate*
  この章ではSchemaUpdateについて説明します。
- *システムデータベースセットアップ*
  この章ではシステムデータベースセットアップについて説明します。
- *テナント環境セットアップ*
  この章ではテナント環境セットアップについて説明します。
- *サンプルデータセットアップ*
  この章ではサンプルデータセットアップについて説明します。
- *セットアップ実行結果ログ*
  この章ではセットアップ実行結果ログについて説明します。
- *付録*
  この章ではセットアップに関する補足事項について説明します。

用語

当仕様書で登場する共通的な用語を説明します。

ショートモジュールID

正式なモジュールIDのうち、最後のドット以降の文字列を指します。

例）モジュールID=”jp.co.intra_mart.im_tenant”に対するショートモジュールIDは、”im_tenant”となります。

テナントマスタ情報

テナント運用上、必須となるマスタ情報群です。

以下の情報がテナントマスタ情報に該当します。

- **ロール**
- **アカウント**

- **カレンダー**
  - カレンダー
  - 日付情報セット
  - 日付情報
  - カレンダーマージ
- **メニュー**
  - メニューカテゴリ
  - メニューグループ
- **認可**
  - リソースグループ
  - リソース
  - サブジェクト
  - ポリシー
- **ジョブスケジューラ**

セットアップ 概要

項目

- セットアップとは
- セットアップの種類
- セットアップで実行可能な処理
- セットアップの定義単位
- セットアップを実行するために用意すべき資源

セットアップとは

テナント環境は、様々なモジュールで構成されています。
モジュールはそれぞれ、自身が動作するための動作前提を定義しています。
その動作前提を構築する処理を **セットアップ** と呼びます。

セットアップの種類

セットアップには次の３種類があります。

- **システムデータベースセットアップ**

  intra-mart Accel Platform が起動できる状態を構築する処理です。

- **テナント環境セットアップ**

  intra-mart Accel Platform の運用を行える状態を構築する処理です。

- **サンプルデータセットアップ**

  試用のためのサンプルデータを投入するための処理です。

セットアップで実行可能な処理

セットアップで実行可能な処理は以下のとおりです。

セットアップ	実行可能な処理
システムデータベースセットアップ	DDL
テナント環境セットアップ	前処理 / DDL / DML / 拡張インポート
サンプルデータセットアップ	前処理 / DDL / DML / 拡張インポート

コラム

「前処理」 とは、セットアップの際に、DDL・DMLの発行の前に実行することのできる処理です。
モジュール独自の処理を定義することができます。
セットアップの前提条件を満たしているかチェックを行うなど、DDL・DMLの発行前に実行したい処理がある場合に利用します。

前処理プログラムは、Java / サーバサイドJavaScript どちらでも実装することができます。

intra-mart Accel Platform 2013 Winter 以前 では利用できません。

intra-mart Accel Platform 2014 Spring 以降で利用可能です。

> **コラム**
>
> ここでいう「DML」は、次の二つの処理を包含しています。
>
> - DML の発行
> - テナントマスタ情報のインポート

> **コラム**
>
> **「拡張インポート」** とは、セットアップの際に、DDL・DMLの発行、テナントマスタ情報のインポートの他に実行することのできる処理です。
> モジュール独自の処理を定義することができます。
>
> 拡張インポートは、Java / サーバサイドJavaScript どちらでも実装することができます。

セットアップの定義単位

システムデータベースセットアップ、テナント環境セットアップは、モジュールのバージョン単位で定義することが可能です。
モジュールのバージョンアップで新機能が追加された場合などに対応することができます。

> **コラム**
>
> 正確には、スキーマバージョン単位で定義可能です。スキーマバージョンについては後述の「*SchemaUpdate* 」の章を参照してください。

サンプルデータセットアップは、モジュール単位で定義することが可能です。

セットアップを実行するために用意すべき資源

セットアップを実行するには、次の資源をモジュール単位で用意する必要があります。

- セットアップ設定ファイル

  セットアップ時に、どのインポートファイルを読み込むか、どの拡張インポートを実行するかを定義した設定ファイルです。

- 前処理プログラム

  セットアップ時に実行される前処理です。モジュールが独自に定義可能です。

- インポートファイル

  セットアップ時にインポートする情報を定義したファイルです。

- 拡張インポート

  セットアップ時に実行される処理です。モジュールが独自に定義可能です。

セットアップの種類、インポートする情報、また実行する処理によって、用意する資源が異なります。 各セットアップの章でそれぞれ具体的に説明します。

SchemaUpdate

セットアップの仕様説明の前提として、セットアップが内部で利用する SchemaUpdate 機能について説明します。

項目

- SchemaUpdate とは
- スキーマバージョン とは
- SchemaUpdate の適用対象
- 動作仕様
- 差分セットアップ

SchemaUpdate とは

SchemaUpdate は、モジュールが定義したセットアップの適用状況を管理します。

SchemaUpdate で管理する情報を利用することで、セットアップの実行時に、既に適用されているセットアップが再実行されてしまわないよう制御することができます。

スキーマバージョン とは

スキーマバージョンは、各モジュールが定める動作前提のバージョニング番号です。

番号の定義ルールは 1 から始まる連番です。中抜けを禁止し、必ず 1 ずつインクリメントして定義します。

モジュールのメジャーバージョン、マイナーバージョン、およびマイクロバージョンとは必ずしも合致しません。

スキーマバージョンの実際の定義方法は、後述の *システムデータベースセットアップ*、および *テナント環境セットアップ* の章を参照してください。

SchemaUpdate の適用対象

セットアップのうち、システムデータベースセットアップとテナント環境セットアップが適用対象となります。

コラム

サンプルデータセットアップは適用対象外です。

理由は次のとおりです。

サンプルデータの投入は試用環境を構築する際に実施するものと想定しています。
試用環境は、モジュールの追加やバージョンアップを適宜行いつつ継続利用されるものではない想定のため、サンプルデータについての差分セットアップは実現していません。

サンプルデータを含めた最新状態の環境を構築する場合は、新規環境構築を適宜行ってください。

動作仕様

- データベースにセットアップ適用状況の管理用テーブル **[im_schema_update]** を作成し、このテーブルで管理します。
  - システムデータベースセットアップによるセットアップ適用状況はシステムデータベースで管理します。
  - テナント環境セットアップによるセットアップ適用状況はテナントデータベースで管理します。
- [im_schema_update] テーブルは、セットアップ（サンプルデータセットアップを除く）の実行時に自動作成されます。
  システムデータソース、テナントデータソースが同一の場合は、共用のテーブルをひとつだけ作成します。
- セットアップ適用状況は、モジュール・スキーマバージョン単位、かつ次のインポート種別単位で管理します。
  - 前処理プログラム
  - DDL

- DML（処理としては、DML発行とテナントマスタ情報インポートを包含します。）
- 拡張インポート

コラム

[im_schema_update] では、レコード単位で次の情報を管理します。

カラム名	カラムの説明	登録値の例
module_id	セットアップしたモジュールのショートモジュールID	“im_tenant” / “im_master”
import_type	セットアップしたインポート種別	“SYSTEM_DDL” / “TENANT_DDL” / “TENANT_DML” / “TENANT_EXTENSION”
schema_version	セットアップしたスキーマバージョン	1 / 2 / 3 / ･･･

＜カラム「import_type」の登録値の説明＞

- “SYSTEM_DDL”
  システムデータベースセットアップのインポート種別「DDL」がセットアップ済みであることを表します。
- ““TENANT_PREPROCESSING““
  テナント環境セットアップのインポート種別「前処理」が実行済みであることを表します。
- “TENANT_DDL”
  テナント環境セットアップのインポート種別「DDL」がセットアップ済みであることを表します。
- “TENANT_DML”
  テナント環境セットアップのインポート種別「DML」がセットアップ済みであることを表します。
- “TENANT_EXTENSION”
  テナント環境セットアップのインポート種別「拡張インポート」がセットアップ済みであることを表します。

差分セットアップ

運用開始後、war を構成するモジュールの追加やバージョンアップが発生した際、 システムデータベースセットアップとテナント環境セットアップでは、SchemaUpdate が管理する情報を利用し、各モジュールのセットアップのうち、未セットアップのもののみ適用します。

これにより、既に実施済みのセットアップを再度実行してしまったり、差分適用すべきセットアップ内容を利用者自らが意識して適用しなければならない状況を回避します。

セットアップの際は、[im_schema_update] テーブルで保持されているスキーマバージョンよりも大きいスキーマバージョンのセットアップ用ファイルが存在する場合、それらを差分セットアップ対象とみなします。

システムデータベースセットアップ

項目

- システムデータベースセットアップ とは
- システムデータベースセットアップ 仕様
  - システムデータベースセットアップ で実行可能な処理
  - システムデータベースセットアップ 実行準備
    - インポートファイル の作成と配置
  - システムデータベースセットアップ 実行方法
  - システムデータベースセットアップ 実行順序
  - システムデータベースセットアップ におけるトランザクション制御
  - 例外発生時の動作

システムデータベースセットアップ とは

intra-mart Accel Platform の起動時からシステムが参照する必要のあるシステムデータベースのテーブルを、システムデータベースに作成する処理です。

たとえば、「im_tenant」モジュールでは、システムデータベースセットアップで次のようなテーブルを作成しています。

- im_administrator（システム管理者テーブル）
- im_tenant_info（テナント情報テーブル）

これらは、 intra-mart Accel Platform が起動してからテナント環境セットアップが実行開始されるまでにアクセスする必要があり、かつシステムデータベースで管理されるべき情報を保持するテーブルです。
このようなテーブルを必要とするモジュールは、システムデータベースセットアップを実行するための資材を用意する必要があります。

> **注意**
>
> システムデータベースセットアップは上記の性質上、NTTデータ イントラマート社が提供する製品でのみ実行するセットアップです。
>
> 独自開発などの場合は、システムデータベースセットアップの定義の追加は行わないでください。

> **注意**
>
> intra-mart Accel Platform における「システムデータベース」は、
> intra-mart WebPlatform / intra-mart AppFramework における「システムデータベース」とは扱いが異なることに注意してください。
>
> それぞれのデータベースの対応は下表のようになります。

intra-mart Accel Platform	intra-mart WebPlatform / intra-mart AppFramework
システムデータベース	（なし）
テナントデータベース	ログイングループデータベース
シェアードデータベース	システムデータベース

システムデータベースセットアップ 仕様

システムデータベースセットアップ で実行可能な処理

セットアップで実行できる処理は、システムデータベースに対する DDL の発行のみです。

システムデータベースセットアップ 実行準備

システムデータベースセットアップを実行するモジュールは、セットアップ用SQLファイルを用意する必要があります。

例：モジュールID=”jp.co.intra_mart.im_tenant”の場合、次のようなSQLファイルを用意してください。

WEB-INF/conf/products/import/system/im_tenant/import-im_tenant-1.sql

セットアップ用SQLファイルの配置先、および命名の規則について説明します。

- セットアップ用SQLファイルの配置先ディレクトリ規則は次のとおりです。

  WEB-INF/conf/products/import/system/ %ショートモジュールID%

- セットアップ用SQLファイルの命名規則は次のとおりです。

  import-%ショートモジュールID%-%スキーマバージョン%.sql

> **コラム**
>
> セットアップ用SQLファイルの初回作成の場合、スキーマバージョンは必ず”1”としてください。
>
> 以降、バージョンアップなどのタイミングで追加セットアップが必要となったときは、順次スキーマバージョンを1ずつインクリメントしてセットアップ用SQLファイルを作成してください。

> **コラム**
>
> セットアップ用SQLファイルは、システムデータベースの種類に応じた定義をすることができます。
> セットアップ用SQLファイルのファイル名の末尾にデータベースサフィックスを付加することで対応します。
>
> 実際のモジュールを例にとって説明します。
>
> 「サービス機構モジュール（jp.co.intra_mart.im_service）」は、スキーマバージョン＝1で次のセットアップ用SQLファイルを定義しています。
>
> - import-im_service-1.sql
> - import-im_service-1_oracle.sql
> - import-im_service-1_sqlserver.sql
>
> この場合のシステムデータベースセットアップは次のように動作します。
>
> | システムデータベースの種類 | システムデータベースセットアップ時に読み込まれるセットアップ用SQLファイル |
> |---|---|
> | Oracle Database | import-im_service-1_oracle.sql |
> | PostgreSQL | import-im_service-1.sql |
> | Microsoft SQL Server | import-im_service-1_sqlserver.sql |
>
> マルチデータベース対応の詳細については *セットアップにおけるマルチデータベース対応* を参照してください。

セットアップ用SQLファイルの記法については *セットアップ用SQLファイル / インポートDDL / インポートDML 記法* を参照してください。

インポートファイル の作成と配置

セットアップ設定ファイルに記述したインポートファイルを作成します。

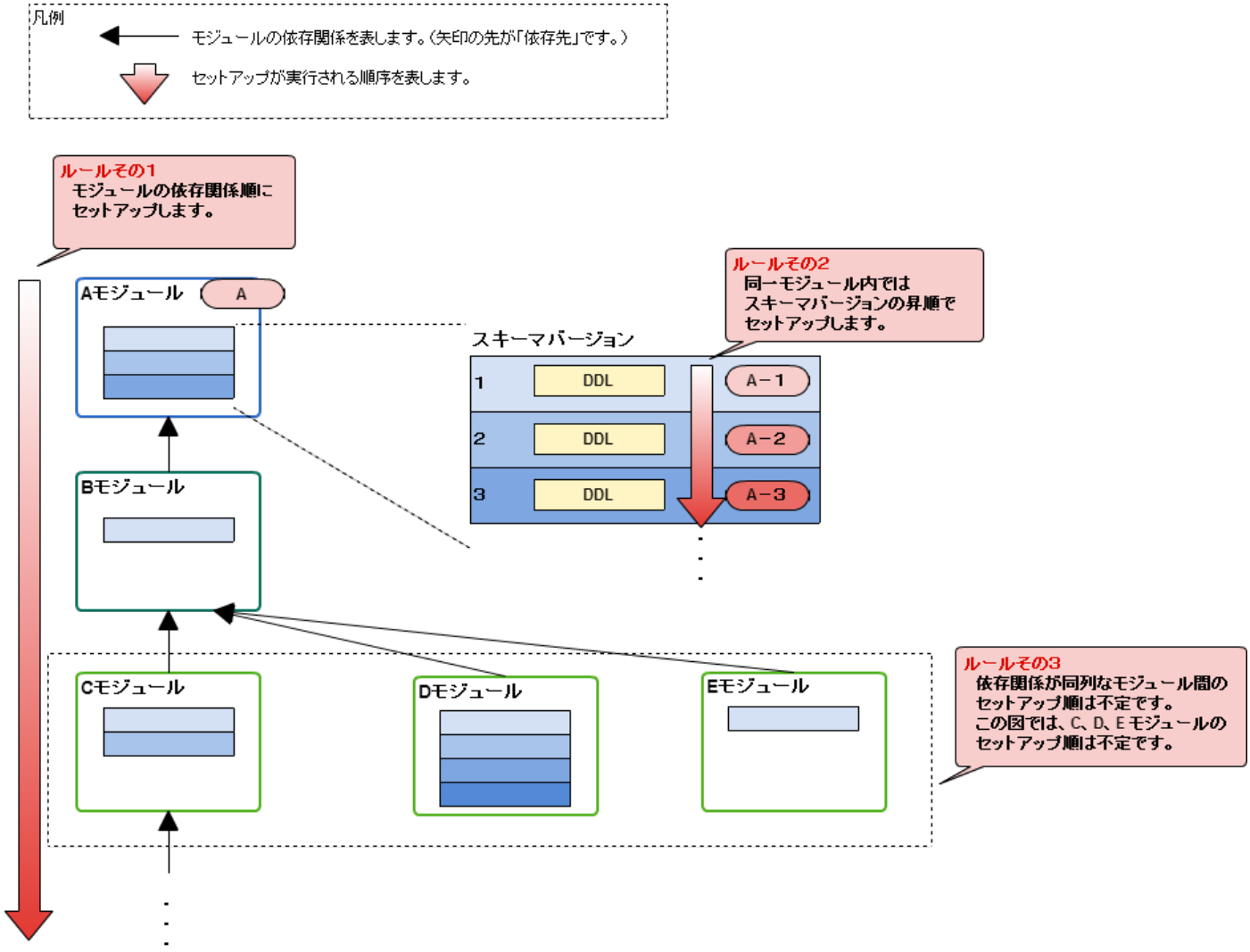
作成したら、セットアップ設定ファイルで指定したパスに配置します。
配置先は **システムストレージ** です。パブリックストレージではないので注意してください。

システムデータベースセットアップ 実行方法

ここまでで説明したセットアップ用SQLファイルが用意されている場合、定義したDDLは intra-mart Accel Platform の起動時に発行

されます。

システムデータベースセットアップ 実行順序



> **コラム**
>
> SchemaUpdate 機能を利用しているので、セットアップ済みのシステムデータベースセットアップはスキップされます。
> 同じ処理が2度以上実行されることはありません。

システムデータベースセットアップ におけるトランザクション制御

システムデータベースセットアップ では、トランザクションの開始・終了処理は行いません。
すべてのDDLは、トランザクション制御なしで発行されます。

例外発生時の動作

システムデータベースセットアップの実行中に例外が発生した場合、システムデータベースセットアップは即時停止し、後続のセットアップ処理は実行しません。

> **コラム**
>
> システムデータベースセットアップのすべての処理が正常に完了しない限り、 intra-mart Accel Platform の起動が正しく行えません。
>
> システムデータベースセットアップ途中で例外が発生した場合は、後続のセットアップを続ける意味がないため、即時停止します。

テナント環境セットアップ

項目

- テナント環境セットアップ とは
- テナント環境セットアップ 仕様
  - テナント環境セットアップ で実行可能な処理
  - セットアップ設定ファイル 実行準備
    - セットアップ設定ファイル の作成と配置
    - 前処理プログラム の作成と配置
    - インポートファイル の作成と配置
    - 拡張インポート の作成と配置
  - テナント環境セットアップ 実行方法
  - テナント環境セットアップ 実行順序
  - テナント環境セットアップ におけるトランザクション制御
  - 例外発生時の動作

テナント環境セットアップ とは

デプロイされた各モジュールが動作するための動作前提を構築する処理です。

テナント環境セットアップを実行することにより、 intra-mart Accel Platform の運用を行える状態を構築します。

warの初回デプロイ後、またはモジュールの追加やバージョンアップを行った場合は、 intra-mart Accel Platform の起動後にテナント環境セットアップを実行する必要があります。

> **注意**
>
> warの初回デプロイ後、またはモジュールの追加やバージョンアップを行った状態で
> テナント環境セットアップを実行していない場合、 intra-mart Accel Platform が
> 正常に運用できない場合があるので注意してください。

モジュールの動作に必要な事前処理（例：テーブル作成など）が存在する場合、モジュール開発者は、テナント環境セットアップ用の資材を準備する必要があります。

テナント環境セットアップ 仕様

テナント環境セットアップ で実行可能な処理

テナント環境セットアップで実行できる処理は、次の通りです。

- 前処理プログラムの実行
- テナントデータベースに対するDDLの発行
- テナントデータベースに対するDMLの発行
- テナントマスタ情報のインポート
- 拡張インポートの実行

セットアップ設定ファイル 実行準備

セットアップ設定ファイル の作成と配置

テナント環境セットアップを実行するモジュールは、セットアップ設定ファイルを用意する必要があります。

例：モジュールID=”jp.co.intra_mart.im_tenant”の場合、次のxmlファイルを用意してください。

WEB-INF/conf/products/import/basic/im_tenant/import-im_tenant-config-1.xml

セットアップ設定ファイルの配置先、および命名の規則について説明します。

- セットアップ設定ファイルの配置先ディレクトリ規則は次のとおりです。

  WEB-INF/conf/products/import/basic/ %ショートモジュールID%

- セットアップ設定ファイルの命名規則は次のとおりです。

  import-%ショートモジュールID%-config-%スキーマバージョン%.xml

> **コラム**
>
> セットアップ設定ファイルの初回作成の場合、スキーマバージョンは必ず”1”としてください。
>
> 以降、バージョンアップなどのタイミングで追加セットアップが必要となったときは、順次スキーマバージョンを1ずつインクリメントしてセットアップ設定ファイルを作成してください。

セットアップ設定ファイルには、次の定義を記述します。

- 発行するDDLを定義したインポートDDLパス
- 発行するDMLを定義したインポートDMLパス
- インポートするテナントマスタ情報のインポートXMLパス
- 実行する拡張インポートのパス

> **コラム**
>
> セットアップ設定ファイルで定義するインポート対象ファイルを総称して、 **「インポートファイル」** と表記します。
>
> インポートファイルをさらに分類し、記述されている内容や形式によって次のように表記します。
>
> - インポートDDL
>
>   DDL文を記述したSQL形式のインポートファイルのことです。
>   拡張子は **「sql」** です。
>
> - インポートDML
>
>   DML文を記述したSQL形式のインポートファイルのことです。
>   拡張子は **「sql」** です。
>
> - インポートXML
>
>   XML形式のインポートファイルのことです。
>   テナントマスタのインポート情報を定義するファイルに該当します。
>   拡張子は **「xml」** です。

セットアップ設定ファイルの記述例を示します。
下記では、セットアップ設定ファイルで定義可能なすべての情報を盛り込んでいます。

```
<import-data-config
 xmlns="http://intra_mart.co.jp/system/service/provider/importer/config/import-data-config"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://intra_mart.co.jp/system/service/provider/importer/config/import-data-config import-data-config.xsd ">

 <!-- 前処理プログラム ※不要な場合、削除してください。 -->
 <preprocessors>

  <!-- Javaクラスを指定する場合 ※複数定義が可能です。 [0..*] -->
  <!-- jp.co.intra_mart.system.service.importer.preprocessing.ImportPreprocessor を継承したJavaクラスの完全修飾クラス名を指定します。 -->
  <preprocessor-class>jp.co.intra_mart.foo.bar.SampleImportPreprocessor</preprocessor-class>

  <!-- サーバサイドJavaScriptを指定する場合 ※複数定義が可能です。 [0..*] -->
  <!-- WEB-INF/jssp/srcなど、サーバサイドJavaScriptのルートからの相対パスを、拡張子".js"をつけて指定します。 -->
  <!-- 指定したサーバサイドJavaScriptにおける execute 関数が実行されます。 -->
```

```
  <!-- execute 関数は処理結果を Boolean 型で返却する必要があります。 -->
  <preprocessor-class>sample/sample_preprocessor.js</preprocessor-class>

</preprocessors>

<!-- データベース系 ※不要な場合、削除してください。 -->
<database>

  <!-- インポートDDL ※複数定義が可能です。 [0..*] -->
  <!-- SystemStorageからの相対パスを指定します。 -->
  <create-file>products/import/basic/module_id/ddl.sql</create-file>

  <!-- インポートDML ※複数定義が可能です。 [0..*] -->
  <!-- SystemStorageからの相対パスを指定します。 -->
  <!-- 基本情報と国際化情報を分割して定義します。 -->
  <insert-file>products/import/basic/module_id/dml.sql</insert-file>
  <insert-file>products/import/basic/module_id/dml_ja.sql</insert-file>
  <insert-file>products/import/basic/module_id/dml_en.sql</insert-file>
  <insert-file>products/import/basic/module_id/dml_zh_CN.sql</insert-file>

</database>

<!-- テナントマスタ系 ※不要な場合、削除してください。 -->
<tenant-master>
  <!-- 各ファイルパスには、SystemStorageからの相対パスを指定します。 -->

  <!-- ロール ※複数定義が可能です。 [0..*] -->
  <!-- 基本情報と国際化情報を分割して定義します。 -->
  <role-file>products/import/basic/module_id/role.xml</role-file>
  <role-file>products/import/basic/module_id/role_ja.xml</role-file>
  <role-file>products/import/basic/module_id/role_en.xml</role-file>
  <role-file>products/import/basic/module_id/role_zh_CN.xml</role-file>

  <!-- アカウント ※複数定義が可能です。 [0..*] -->
  <!-- 基本情報のみ定義します。（アカウントには国際化情報がないためです。） -->
  <account-file>products/import/basic/module_id/account.xml</account-file>

  <!-- カレンダー -->

  <!-- カレンダー:カレンダー ※複数定義が可能です。 [0..*] -->
  <!-- 基本情報と国際化情報を分割して定義します。 -->
  <calendar-file>products/import/basic/module_id/calendar.xml</calendar-file>
  <calendar-file>products/import/basic/module_id/calendar_ja.xml</calendar-file>
  <calendar-file>products/import/basic/module_id/calendar_en.xml</calendar-file>
  <calendar-file>products/import/basic/module_id/calendar_zh_CN.xml</calendar-file>

  <!-- カレンダー:日付情報セット ※複数定義が可能です。 [0..*] -->
  <!-- 基本情報と国際化情報を分割して定義します。 -->
  <calendar-day-set-file>products/import/basic/module_id/calendar-day-set.xml</calendar-day-set-file>
  <calendar-day-set-file>products/import/basic/module_id/calendar-day-set_ja.xml</calendar-day-set-file>
  <calendar-day-set-file>products/import/basic/module_id/calendar-day-set_en.xml</calendar-day-set-file>
  <calendar-day-set-file>products/import/basic/module_id/calendar-day-set_zh_CN.xml</calendar-day-set-file>

  <!-- カレンダー:日付情報 ※複数定義が可能です。 [0..*] -->
  <!-- 基本情報と国際化情報を分割して定義します。 -->
  <calendar-day-file>products/import/basic/module_id/calendar-day.xml</calendar-day-file>
  <calendar-day-file>products/import/basic/module_id/calendar-day_ja.xml</calendar-day-file>
  <calendar-day-file>products/import/basic/module_id/calendar-day_en.xml</calendar-day-file>
  <calendar-day-file>products/import/basic/module_id/calendar-day_zh_CN.xml</calendar-day-file>

  <!-- カレンダー:カレンダーマージ ※複数定義が可能です。 [0..*] -->
  <!-- 基本情報のみ定義します。（カレンダー:カレンダーマージには国際化情報がないためです。） -->
  <calendar-merge-file>products/import/basic/module_id/calendar-merge.xml</calendar-merge-file>

  <!-- メニュー -->

  <!-- メニュー:メニューカテゴリ ※複数定義が可能です。 [0..*] -->
  <!-- 基本情報と国際化情報を分割して定義します。 -->
  <menu-group-category-file>products/import/basic/module_id/menu-group-category.xml</menu-group-category-file>
  <menu-group-category-file>products/import/basic/module_id/menu-group-category_ja.xml</menu-group-category-
```

```
    <menu-group-category-file>products/import/basic/module_id/menu-group-category_ja.xml</menu-group-category-file>
    <menu-group-category-file>products/import/basic/module_id/menu-group-category_en.xml</menu-group-category-file>
    <menu-group-category-file>products/import/basic/module_id/menu-group-category_zh_CN.xml</menu-group-category-file>

    <!-- メニュー:メニューグループ※複数定義が可能です。 [0..*] -->
    <!--  基本情報と国際化情報を分割して定義します。 -->
    <menu-group-file>products/import/basic/module_id/menu-group.xml</menu-group-file>
    <menu-group-file>products/import/basic/module_id/menu-group_ja.xml</menu-group-file>
    <menu-group-file>products/import/basic/module_id/menu-group_en.xml</menu-group-file>
    <menu-group-file>products/import/basic/module_id/menu-group_zh_CN.xml</menu-group-file>

    <!-- 認可 -->

    <!-- 認可:リソースグループ※複数定義が可能です。 [0..*] -->
    <!--  基本情報と国際化情報を分割して定義します。 -->
    <authz-resource-group-file>products/import/basic/module_id/authz-resource-group.xml</authz-resource-group-file>
    <authz-resource-group-file>products/import/basic/module_id/authz-resource-group_ja.xml</authz-resource-group-file>
    <authz-resource-group-file>products/import/basic/module_id/authz-resource-group_en.xml</authz-resource-group-file>
    <authz-resource-group-file>products/import/basic/module_id/authz-resource-group_zh_CN.xml</authz-resource-group-file>

    <!-- 認可:リソース※複数定義が可能です。 [0..*] -->
    <!--  基本情報と国際化情報を分割して定義します。 -->
    <authz-resource-file>products/import/basic/module_id/authz-resource.xml</authz-resource-file>
    <authz-resource-file>products/import/basic/module_id/authz-resource_ja.xml</authz-resource-file>
    <authz-resource-file>products/import/basic/module_id/authz-resource_en.xml</authz-resource-file>
    <authz-resource-file>products/import/basic/module_id/authz-resource_zh_CN.xml</authz-resource-file>

    <!-- 認可:サブジェクトグループ※複数定義が可能です。 [0..*] -->
    <!--  基本情報と国際化情報を分割して定義します。 -->
    <authz-subject-group-file>products/import/basic/module_id/authz-subject-group.xml</authz-subject-group-file>
    <authz-subject-group-file>products/import/basic/module_id/authz-subject-group_ja.xml</authz-subject-group-file>
    <authz-subject-group-file>products/import/basic/module_id/authz-subject-group_en.xml</authz-subject-group-file>
    <authz-subject-group-file>products/import/basic/module_id/authz-subject-group_zh_CN.xml</authz-subject-group-file>

    <!-- 認可:ポリシー※複数定義が可能です。 [0..*] -->
    <!--  基本情報のみ定義します。（ポリシーには国際化情報がないためです。） -->
    <authz-policy-file>products/import/basic/module_id/authz-policy.xml</authz-policy-file>

    <!-- ジョブスケジューラ※複数定義が可能です。 [0..*] -->
    <!--  基本情報と国際化情報を分割して定義します。 -->
    <job-scheduler-file>products/import/basic/module_id/job-scheduler.xml</job-scheduler-file>
    <job-scheduler-file>products/import/basic/module_id/job-scheduler_ja.xml</job-scheduler-file>
    <job-scheduler-file>products/import/basic/module_id/job-scheduler_en.xml</job-scheduler-file>
    <job-scheduler-file>products/import/basic/module_id/job-scheduler_zh_CN.xml</job-scheduler-file>

  </tenant-master>

  <!-- 拡張インポート※不要な場合、削除してください。 -->
  <extends-import>

    <!-- Javaクラスを指定する場合※複数定義が可能です。 [0..*] -->
    <!--  jp.co.intra_mart.foundation.security.ExtendsImport を継承したJavaクラスの完全修飾クラス名を指定します。 -->
    <extends-import-class>jp.co.intra_mart.foo.bar.SampleExtendsImporter</extends-import-class>

    <!-- サーバサイドJavaScriptを指定する場合※複数定義が可能です。 [0..*] -->
    <!--  WEB-INF/jssp/srcなど、サーバサイドJavaScriptのルートからの相対パスを、拡張子".js"をつけて指定します。 -->
    <!--  指定したサーバサイドJavaScriptにおける doImport 関数が実行されます。 -->
    <!--  doImport 関数の第一引数にはテナントIDを渡します。 -->
    <!--  doImport 関数は処理結果を Boolean 型で返却する必要があります。 -->
    <extends-import-class>sample/sample_import.js</extends-import-class>

  </extends-import>
```

```
</import-data-config>
```

> **コラム**
>
> - セットアップ設定ファイルで定義された情報は、定義順に（上から順に）処理されます。
> - 「database」タグ内は必ず次の順序で定義する必要があります。
>
> 1. create-file
> 2. insert-file
>
> - 「tenant-master」タグ内に定義可能なタグ群は順不同で自由に定義可能です。
> インポート順序によっては情報の不整合が発生する場合があるため、適宜調整してください。
>
> 例
>
> メニューリソースはメニューグループのインポートで登録されますが、この作業の実施前にメニューリソースに対する認可ポリシーをインポートしようとすると、メニューリソースが存在せずに例外が発生する可能性があります。
>
> セットアップ設定ファイルのフォーマットファイルについては *フォーマットファイル(xsd)* を参照してください。

前処理プログラム の作成と配置

セットアップ設定ファイルに記述した前処理プログラムを作成します。
前処理プログラムの実装については *前処理プログラム サンプル* を参照してください。

作成したら、セットアップ設定ファイルで指定したパスに配置します。

セットアップ設定ファイルで前処理プログラムを記述していない場合は当作業は不要です。

前処理プログラムの配置先パッケージおよびパスについての規則は特に設けません。
各モジュールにおける適当なパッケージおよびパスで定義してください。

インポートファイル の作成と配置

セットアップ設定ファイルに記述したインポートファイルを作成します。

作成したら、セットアップ設定ファイルで指定したパスに配置します。
配置先は **システムストレージ** です。パブリックストレージではないので注意してください。

セットアップ設定ファイルでインポートファイルを記述していない場合は当作業は不要です。

インポートファイルの配置先、および命名の規則について説明します。

インポートファイルの配置先ディレクトリ規則は次のとおりです。

<STORAGE_PATH>/system/storage/products/import/basic/%ショートモジュールID%[/任意のディレクトリ]

> **コラム**
>
> ショートモジュールIDディレクトリを作成することにより、モジュール間でのインポートファイルの重複を回避します。
>
> ショートモジュールIDディレクトリ配下のディレクトリ作成は必要に応じて実施してください。

インポートファイルの命名規則は次のとおりです。

- 可能な限り、ファイル名からインポートされる情報の概要が推測できるようなファイル名をつけてください。
- あるモジュールのあるスキーマバージョンでどのインポートファイルが処理されるかは、該当のセットアップ設定ファイルを見れば簡単に確認できるため、
  ファイル名にモジュールIDやスキーマバージョンの付加を義務付けることはしません。

インポートDDL、インポートDMLの記法については *セットアップ用SQLファイル / インポートDDL / インポートDML 記法* を参照してください。
インポートXMLの記法については、各テナントマスタインポータの仕様書を参照してください。

コラム

- インポートDML
- インポートXML

上記のインポートファイルは、原則として次の2種類を用意し、セットアップ設定ファイルで定義してください。

1. **基本インポートファイル**
   ロケールに関連しない情報のみを持ったインポートファイルです。 保持する情報は以下です。
   - キー
   - ロケールに関連しないデータ

   基本インポートファイルのファイル名には、ロケールIDサフィックスはつけません。
   例）

   im_workflow-dml.sql
   im_admin-role.xml

2. **多言語インポートファイル**
   ロケールに関連する情報のみを持ったインポートファイルです。 保持する情報は以下です。
   - キー
   - ロケールに関連するデータ

   多言語インポートファイルのファイル名には、ロケールIDサフィックスをつけます。
   また、定義するロケール単位でファイルを分割します。
   例）
   - ja ロケールの情報を定義したインポートファイル

     im_workflow-dml **_ja** .sql
     im_admin-role **_ja** .xml
   - en ロケールの情報を定義したインポートファイル

     im_workflow-dml **_en** .sql
     im_admin-role **_en** .xml
   - zh_CN ロケールの情報を定義したインポートファイル

     im_workflow-dml **_zh_CN** .sql
     im_admin-role **_zh_CN** .xml

インポータ仕様によりロケールに関連する情報と関連しない情報が明確に分けることのできない場合、基本インポートXMLには **「en」** の情報を定義してください。
またこの場合は多言語インポートファイルの用意は不要です。

このように管理を行うことで、以下のメリットを見込んでいます。

1. モジュール資源の翻訳を実施する際に、インポートファイルに関する翻訳対象の特定が容易になります。
2. モジュール資源の翻訳を実施する際に、インポートファイルに関する翻訳元言語の特定が容易になります。

コラム

製品化する上では、基本的には３言語（日本語、英語、中国語（簡体字））分のマスタデータ・サンプルデータを提供します。

コラム

テナントマスタ用インポートXMLのフォーマットファイル(xsd)は、基本インポートXMLと多言語インポートXMLを同じスキーマで作成できるように設計されています。

これは以下のメリットを見込んでいます。

1. セットアップ時に利用するインポータと、テナント管理機能「ジョブネット設定」より実行可能なテナントマスタ情報インポートを、同一のインポータでまかなうことができます。
2. セットアップの場合は、基本インポートXML / 多言語インポートXML のように分割管理ができます。
3. テナント管理機能からの実行の場合は、基本＋多言語のインポートXMLを作成し、多言語情報を包含したテナントマスタ情報を一括でインポートすることができます。

コラム

インポートDDLおよびインポートDMLは、テナントデータベースの種類に応じた定義をすることができます。
各ファイル名の末尾にデータベースサフィックスを付加することで対応します。

実際のモジュールを例にとって説明します。

「テナント管理機能（jp.co.intra_mart.im_service）」モジュールは、 スキーマバージョン＝1のセットアップ設定ファイルにおいて、次のインポートDDLを定義しています。

```
<!-- 中略 -->

<database>
   <!-- 中略 -->
   <!-- im_password_history -->
   <create-file>products/import/basic/im_tenant/im_password_history/im_password_history-ddl.sql</create-file>
</database>

<!-- 中略 -->
```

また、 <STORAGE_PATH>/system/products/import/basic/im_tenant/im_password_history ディレクトリには、次のファイルを配置しています。

- im_password_history-ddl.sql
- im_password_history-ddl_sqlserver.sql

この場合のテナント環境セットアップは次のように動作します。

テナントデータベースの種類	テナント環境セットアップ時に読み込まれるインポートDDL
Oracle Database	im_password_history-ddl.sql
PostgreSQL	im_password_history-ddl.sql
Microsoft SQL Server	im_password_history-ddl_sqlserver.sql

セットアップ設定ファイル側には、データベースサフィックスを付けずに定義することに注意してください。

マルチデータベース対応の詳細については *セットアップにおけるマルチデータベース対応* を参照してください。

拡張インポート の作成と配置

セットアップ設定ファイルに記述した拡張インポートを作成します。

作成したら、セットアップ設定ファイルで指定したパスに配置します。

セットアップ設定ファイルで拡張インポートを記述していない場合は当作業は不要です。

拡張インポートの配置先パッケージおよびパスについての規則は特に設けません。

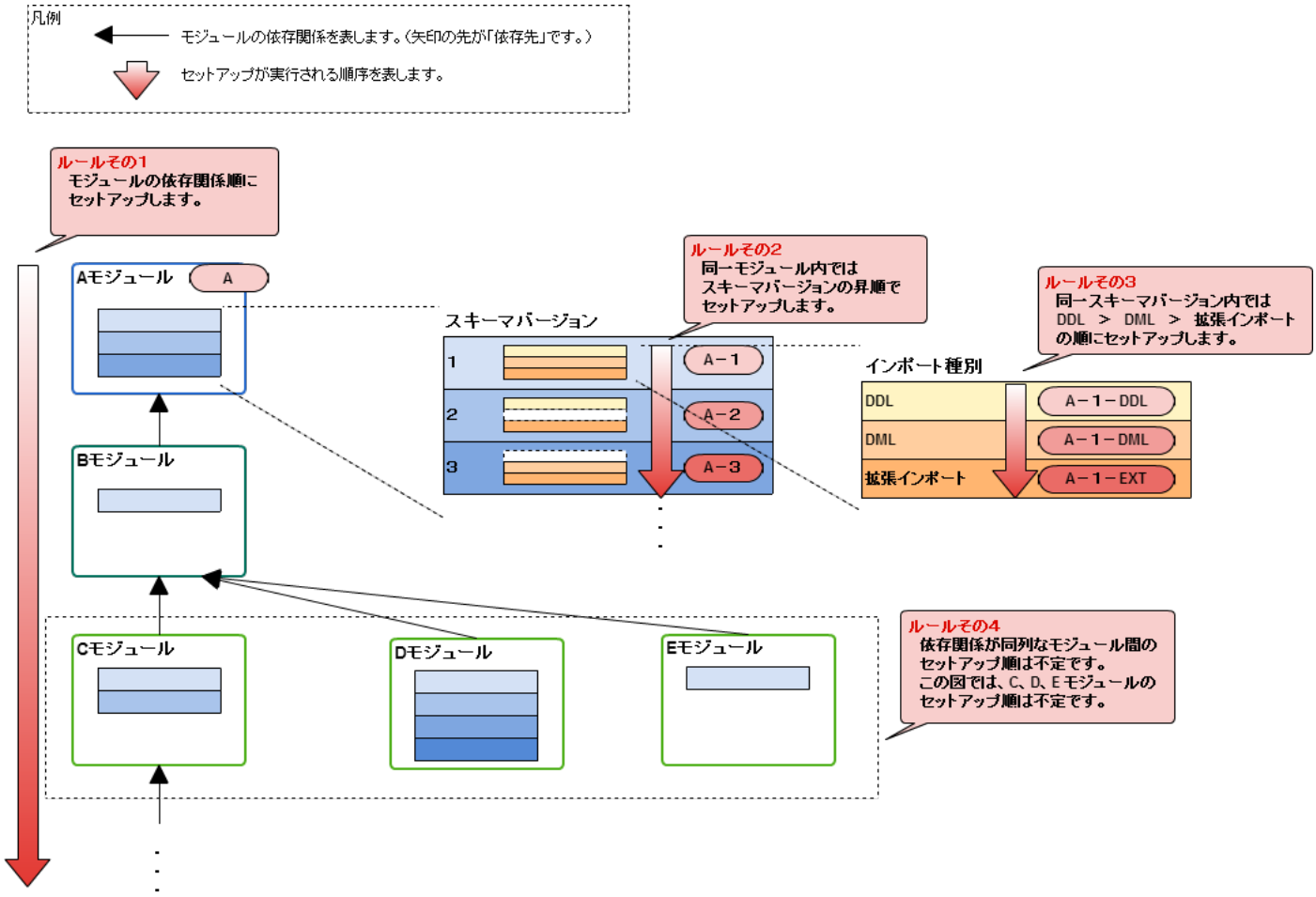
各モジュールにおける適当なパッケージおよびパスで定義してください。

> **コラム**
>
> インポートファイル群はセットアップでのみ利用されるため、モジュールやスキーマバージョンを意識した構成をとります。
> それに対して拡張インポートについては、セットアップでのみ利用されるとは限りません。
>
> 拡張インポートを単体で実行するAPI（ jp.co.intra_mart.foundation.security.ExtendsImportManager ）が用意されており、セットアップのほか、アプリケーションの任意処理で利用されている可能性もあります。
>
> そのため、インポートファイルの場合のようなセットアップに特化した命名・配置規則は、拡張インポートでは採用していません。

テナント環境セットアップ 実行方法

システム管理者でログインし、テナント環境セットアップを実行します。

テナント環境セットアップ 実行順序



> **コラム**
>
> SchemaUpdate 機能を利用しているので、セットアップ済みのテナント環境セットアップはスキップされます。
> 同じ処理が2度以上実行されることはありません。

テナント環境セットアップ におけるトランザクション制御

テナント環境セットアップ 実行時のトランザクション単位は次のとおりです。

- 前処理

トランザクション制御は行いません。トランザクション制御が必要な処理を前処理プログラムで行う場合は、独自にトランザクション制御実装を行う必要があります。

- DDL

  トランザクション制御は行いません。

- DML（DML/テナントマスタ情報のインポート）

  スキーマバージョン単位でトランザクション制御します。

- 拡張インポート

  トランザクション制御は行いません。トランザクション制御が必要な処理を拡張インポートで行う場合は、独自にトランザクション制御実装を行う必要があります。

例外発生時の動作

テナント環境セットアップの実行中に例外が発生した場合、テナント環境セットアップは即時停止し、後続のセットアップ処理は実行しません。

コラム

テナント環境セットアップのすべての処理が正常に完了しない限り、テナントの運用が正しく行えません。

テナント環境セットアップ途中で例外が発生した場合は、後続のセットアップを続ける意味がないため、即時停止します。

コラム

DML（DML/テナントマスタ情報のインポート）で例外が発生した場合、例外発生したモジュール・スキーマバージョンのDMLによって登録された情報はすべてロールバックされます。
この状態で再度テナント環境セットアップを実行すると、セットアップは例外が発生したモジュール・スキーマバージョンのDMLから再開します。

そのため、DMLで例外が発生した場合は、例外の原因を特定し、問題を解決した後にテナント環境セットアップを再実行することで、環境の再構築を行うことなくセットアップを続行することが可能です。

DDLはトランザクション制御を行っていないため、DDL発行中に例外が発生した場合はセットアップの続行は基本的にはできません。

拡張インポートに関しては、拡張インポートによる処理内容次第となります。

サンプルデータセットアップ

サンプルデータセットアップの仕様は、テナント環境セットアップの仕様と似通っています。
説明の中で使用する用語などについても同様ですので、この章を読み進める前に
テナント環境セットアップ を一読されることを推奨します。

項目

- サンプルデータセットアップ とは
- サンプルデータセットアップ 仕様
  - サンプルデータセットアップ で実行可能な処理
  - サンプルデータセットアップ 実行準備
    - セットアップ設定ファイル の作成と配置
    - 前処理プログラム の作成と配置
    - インポートファイル の作成と配置
    - 拡張インポート の作成と配置
  - サンプルデータセットアップ 実行方法
  - サンプルデータセットアップ 実行順序
  - サンプルデータセットアップ におけるトランザクション制御
  - 例外発生時の動作

サンプルデータセットアップ とは

デプロイされた各モジュールを試用するためのサンプルデータを投入する処理です。

試用環境を構築する際に実行するセットアップで、本番環境では実施する必要はありません。

モジュールの開発者は、開発しているモジュールを動作させるためのサンプルデータを利用者に提供する場合、 サンプルデータセットアップを実行するための資材を用意する必要があります。

サンプルデータセットアップ 仕様

サンプルデータセットアップ で実行可能な処理

サンプルデータセットアップで実行できる処理は、次の通りです。

- 前処理プログラムの実行
- テナントデータベースに対するDDLの発行
- テナントデータベースに対するDMLの発行
- テナントマスタ情報のインポート
- 拡張インポートの実行

サンプルデータセットアップ 実行準備

セットアップ設定ファイル の作成と配置

サンプルデータセットアップを実行するモジュールは、セットアップ設定ファイルを用意する必要があります。

例：モジュールID=”jp.co.intra_mart.im_tenant”の場合、次のようなxmlファイルを用意してください。

WEB-INF/conf/products/import/sample/import-im_tenant-config.xml

セットアップ設定ファイルの配置先、および命名の規則について説明します。

- セットアップ設定ファイルの配置先ディレクトリ規則は次のとおりです。

  WEB-INF/conf/products/import/sample

- セットアップ設定ファイルの命名規則は次のとおりです。

  import-%ショートモジュールID%-config.xml

> **コラム**
>
> サンプルデータセットアップのセットアップ設定ファイルには、スキーマバージョンによるバージョニング管理は適用しません。
>
> モジュールが定義するサンプルデータは、そのモジュールの最新バージョンに合わせて常に最新状態となるよう管理してください。

> **コラム**
>
> サンプルデータセットアップのセットアップ設定ファイルの配置先ディレクトリには、 テナント環境セットアップの場合とは異なり、ショートモジュールIDディレクトリを設けません。
>
> サンプルデータセットアップは SchemaUpdate によるセットアップ適用状況の管理の対象外のため、特定のモジュールにおけるセットアップ設定ファイルは最大でもひとつしか作成されません。
>
> そのため、モジュールIDディレクトリを用意して分類する必要がありません。

セットアップ設定ファイルには、次の定義を記述します。（テナント環境セットアップと同じ構成です。）

- 発行するDDLを定義したインポートDDLパス
- 発行するDMLを定義したインポートDMLパス
- インポートするテナントマスタ情報のインポートXMLパス
- 実行する拡張インポートのパス

セットアップ設定ファイルの記述例を示します。

セットアップ設定ファイルの記述例を示します。
下記では、セットアップ設定ファイルで定義可能なすべての情報を盛り込んでいます。

```
<import-data-config
 xmlns="http://intra_mart.co.jp/system/service/provider/importer/config/import-data-config"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://intra_mart.co.jp/system/service/provider/importer/config/import-data-config import-data-
config.xsd ">

 <!-- データベース系 ※不要な場合、削除してください。 -->
 <database>

  <!-- インポートDDL ※複数定義が可能です。 [0..*] -->
  <!--  SystemStorageからの相対パスを指定します。 -->
  <create-file>products/import/sample/module_id/ddl.sql</create-file>

  <!-- インポートDML ※複数定義が可能です。 [0..*] -->
  <!--  SystemStorageからの相対パスを指定します。 -->
  <!--  基本情報と国際化情報を分割して定義します。 -->
  <insert-file>products/import/sample/module_id/dml.sql</insert-file>
  <insert-file>products/import/sample/module_id/dml_ja.sql</insert-file>
  <insert-file>products/import/sample/module_id/dml_en.sql</insert-file>
  <insert-file>products/import/sample/module_id/dml_zh_CN.sql</insert-file>

 </database>

 <!-- テナントマスタ系 ※不要な場合、削除してください。 -->
 <tenant-master>
  <!--  各ファイルパスには、SystemStorageからの相対パスを指定します。 -->

  <!-- ロール ※複数定義が可能です。 [0..*] -->
  <!--  基本情報と国際化情報を分割して定義します。 -->
  <role-file>products/import/sample/module_id/role.xml</role-file>
  <role-file>products/import/sample/module_id/role_ja.xml</role-file>
  <role-file>products/import/sample/module_id/role_en.xml</role-file>
```

```
    <role-file>products/import/sample/module_id/role_zh_CN.xml</role-file>

    <!-- アカウント ※複数定義が可能です。 [0..*] -->
    <!--  基本情報のみ定義します。（アカウントには国際化情報がないためです。） -->
    <account-file>products/import/sample/module_id/account.xml</account-file>

    <!-- カレンダー -->

    <!-- カレンダー:カレンダー ※複数定義が可能です。 [0..*] -->
    <!-- 基本情報と国際化情報を分割して定義します。 -->
    <calendar-file>products/import/sample/module_id/calendar.xml</calendar-file>
    <calendar-file>products/import/sample/module_id/calendar_ja.xml</calendar-file>
    <calendar-file>products/import/sample/module_id/calendar_en.xml</calendar-file>
    <calendar-file>products/import/sample/module_id/calendar_zh_CN.xml</calendar-file>

    <!-- カレンダー:日付情報セット ※複数定義が可能です。 [0..*] -->
    <!--  基本情報と国際化情報を分割して定義します。 -->
    <calendar-day-set-file>products/import/sample/module_id/calendar-day-set.xml</calendar-day-set-file>
    <calendar-day-set-file>products/import/sample/module_id/calendar-day-set_ja.xml</calendar-day-set-file>
    <calendar-day-set-file>products/import/sample/module_id/calendar-day-set_en.xml</calendar-day-set-file>
    <calendar-day-set-file>products/import/sample/module_id/calendar-day-set_zh_CN.xml</calendar-day-set-file>

    <!-- カレンダー:日付情報 ※複数定義が可能です。 [0..*] -->
    <!--  基本情報と国際化情報を分割して定義します。 -->
    <calendar-day-file>products/import/sample/module_id/calendar-day.xml</calendar-day-file>
    <calendar-day-file>products/import/sample/module_id/calendar-day_ja.xml</calendar-day-file>
    <calendar-day-file>products/import/sample/module_id/calendar-day_en.xml</calendar-day-file>
    <calendar-day-file>products/import/sample/module_id/calendar-day_zh_CN.xml</calendar-day-file>

    <!-- カレンダー:カレンダーマージ ※複数定義が可能です。 [0..*] -->
    <!--  基本情報のみ定義します。（カレンダー:カレンダーマージには国際化情報がないためです。） -->
    <calendar-merge-file>products/import/sample/module_id/calendar-merge.xml</calendar-merge-file>

    <!-- メニュー -->

    <!-- メニュー:メニューカテゴリ ※複数定義が可能です。 [0..*] -->
    <!--  基本情報と国際化情報を分割して定義します。 -->
    <menu-group-category-file>products/import/sample/module_id/menu-group-category.xml</menu-group-category-
file>
    <menu-group-category-file>products/import/sample/module_id/menu-group-category_ja.xml</menu-group-category-
file>
    <menu-group-category-file>products/import/sample/module_id/menu-group-category_en.xml</menu-group-category-
file>
    <menu-group-category-file>products/import/sample/module_id/menu-group-category_zh_CN.xml</menu-group-
category-file>

    <!-- メニュー:メニューグループ ※複数定義が可能です。 [0..*] -->
    <!--  基本情報と国際化情報を分割して定義します。 -->
    <menu-group-file>products/import/sample/module_id/menu-group.xml</menu-group-file>
    <menu-group-file>products/import/sample/module_id/menu-group_ja.xml</menu-group-file>
    <menu-group-file>products/import/sample/module_id/menu-group_en.xml</menu-group-file>
    <menu-group-file>products/import/sample/module_id/menu-group_zh_CN.xml</menu-group-file>

    <!-- 認可 -->

    <!-- 認可:リソースグループ ※複数定義が可能です。 [0..*] -->
    <!--  基本情報と国際化情報を分割して定義します。 -->
    <authz-resource-group-file>products/import/sample/module_id/authz-resource-group.xml</authz-resource-group-
file>
    <authz-resource-group-file>products/import/sample/module_id/authz-resource-group_ja.xml</authz-resource-group-
file>
    <authz-resource-group-file>products/import/sample/module_id/authz-resource-group_en.xml</authz-resource-group-
file>
    <authz-resource-group-file>products/import/sample/module_id/authz-resource-group_zh_CN.xml</authz-resource-
group-file>

    <!-- 認可:リソース ※複数定義が可能です。 [0..*] -->
    <!--  基本情報と国際化情報を分割して定義します。 -->
    <authz-resource-file>products/import/sample/module_id/authz-resource.xml</authz-resource-file>
    <authz-resource-file>products/import/sample/module_id/authz-resource_ja.xml</authz-resource-file>
```

```
    <authz-resource-file>products/import/sample/module_id/authz-resource_en.xml</authz-resource-file>
    <authz-resource-file>products/import/sample/module_id/authz-resource_zh_CN.xml</authz-resource-file>

    <!-- 認可:サブジェクトグループ※複数定義が可能です。 [0..*] -->
    <!--  基本情報と国際化情報を分割して定義します。 -->
    <authz-subject-group-file>products/import/sample/module_id/authz-subject-group.xml</authz-subject-group-file>
    <authz-subject-group-file>products/import/sample/module_id/authz-subject-group_ja.xml</authz-subject-group-file>
    <authz-subject-group-file>products/import/sample/module_id/authz-subject-group_en.xml</authz-subject-group-
file>
    <authz-subject-group-file>products/import/sample/module_id/authz-subject-group_zh_CN.xml</authz-subject-group-
file>

    <!-- 認可:ポリシー※複数定義が可能です。 [0..*] -->
    <!--  基本情報のみ定義します。（ポリシーには国際化情報がないためです。） -->
    <authz-policy-file>products/import/sample/module_id/authz-policy.xml</authz-policy-file>

    <!-- ジョブスケジューラ※複数定義が可能です。 [0..*] -->
    <!--  基本情報と国際化情報を分割して定義します。 -->
    <job-scheduler-file>products/import/sample/module_id/job-scheduler.xml</job-scheduler-file>
    <job-scheduler-file>products/import/sample/module_id/job-scheduler_ja.xml</job-scheduler-file>
    <job-scheduler-file>products/import/sample/module_id/job-scheduler_en.xml</job-scheduler-file>
    <job-scheduler-file>products/import/sample/module_id/job-scheduler_zh_CN.xml</job-scheduler-file>

  </tenant-master>

  <!-- 拡張インポート※不要な場合、削除してください。 -->
  <extends-import>

    <!-- Javaクラスを指定する場合※複数定義が可能です。 [0..*] -->
    <!--  jp.co.intra_mart.foundation.security.ExtendsImport を継承したJavaクラスの完全修飾クラス名を指定します。 -->
    <extends-import-class>jp.co.intra_mart.foo.bar.SampleExtendsImporter</extends-import-class>

    <!-- サーバサイドJavaScriptを指定する場合※複数定義が可能です。 [0..*] -->
    <!--  WEB-INF/jssp/platform/srcなど、サーバサイドJavaScriptのルートからの相対パスを、拡張子".js"をつけて指定します。 -->
    <!--  指定したサーバサイドJavaScriptにおける doImport 関数が実行されます。 -->
    <!--  doImport 関数の第一引数にはテナントIDを渡します。 -->
    <!--  doImport 関数は処理結果を Boolean 型で返却する必要があります。 -->
    <extends-import-class>sample/sample_import.js</extends-import-class>

  </extends-import>

</import-data-config>
```

コラム

- セットアップ設定ファイルで定義された情報は、定義順に（上から順に）処理されます。
- 「database」タグ内は必ず次の順序で定義する必要があります。
  1. create-file
  2. insert-file
- 「tenant-master」タグ内に定義可能なタグ群は順不同で自由に定義可能です。
  インポート順序によっては情報の不整合が発生する場合があるため、適宜調整してください。

  例

  メニューリソースはメニューグループのインポートで登録されますが、この作業の実施前にメニューリソースに対する認可ポリシーをインポートしようとすると、メニューリソースが存在せずに例外が発生する可能性があります。

セットアップ設定ファイルのフォーマットファイルについては *フォーマットファイル(xsd)* を参照してください。

前処理プログラム の作成と配置

セットアップ設定ファイルに記述した前処理プログラムを作成します。
前処理プログラムの実装については *前処理プログラム サンプル* を参照してください。

作成したら、セットアップ設定ファイルで指定したパスに配置します。

セットアップ設定ファイルで前処理プログラムを記述していない場合は当作業は不要です。

前処理プログラムの配置先パッケージおよびパスについての規則は特に設けません。
各モジュールにおける適当なパッケージおよびパスで定義してください。

インポートファイル の作成と配置

セットアップ設定ファイルに記述したインポートファイルを作成します。

作成したら、セットアップ設定ファイルで指定したパスに配置します。
配置先は **システムストレージ** です。パブリックストレージではないので注意してください。

セットアップ設定ファイルでインポートファイルを記述していない場合は当作業は不要です。

インポートファイルの配置先、および命名の規則について説明します。

インポートファイルの配置先ディレクトリ規則は次のとおりです。

<STORAGE_PATH>/system/storage/products/import/sample/%ショートモジュールID%[/任意のディレクトリ]

> **コラム**
>
> ショートモジュールIDディレクトリを作成することにより、モジュール間でのインポートファイルの重複を回避します。
>
> ショートモジュールIDディレクトリ配下のディレクトリ作成は必要に応じて実施してください。

インポートファイルの命名規則はテナント環境セットアップと同様です。

インポートDDL、インポートDMLの記法については *セットアップ用SQLファイル / インポートDDL / インポートDML 記法* を参照してください。
インポートXMLの記法については、各テナントマスタインポータの仕様書を参照してください。

> **コラム**
>
> - インポートDML
> - インポートXML
>
> 上記のインポートファイルは、原則として次の2種類を用意し、セットアップ設定ファイルで定義してください。
>
> 1. **基本インポートファイル**
> 2. **多言語インポートファイル**
>
> 詳細はテナント環境セットアップと同様です。

> **コラム**
>
> インポートDDLおよびインポートDMLは、システムデータベースの種類に応じた定義をすることができます。
> 各ファイル名の末尾にデータベースサフィックスを付加することで対応します。
>
> 詳細はテナント環境セットアップと同様です。

拡張インポート の作成と配置

セットアップ設定ファイルに記述した拡張インポートを作成します。

作成したら、セットアップ設定ファイルで指定したパスに配置します。

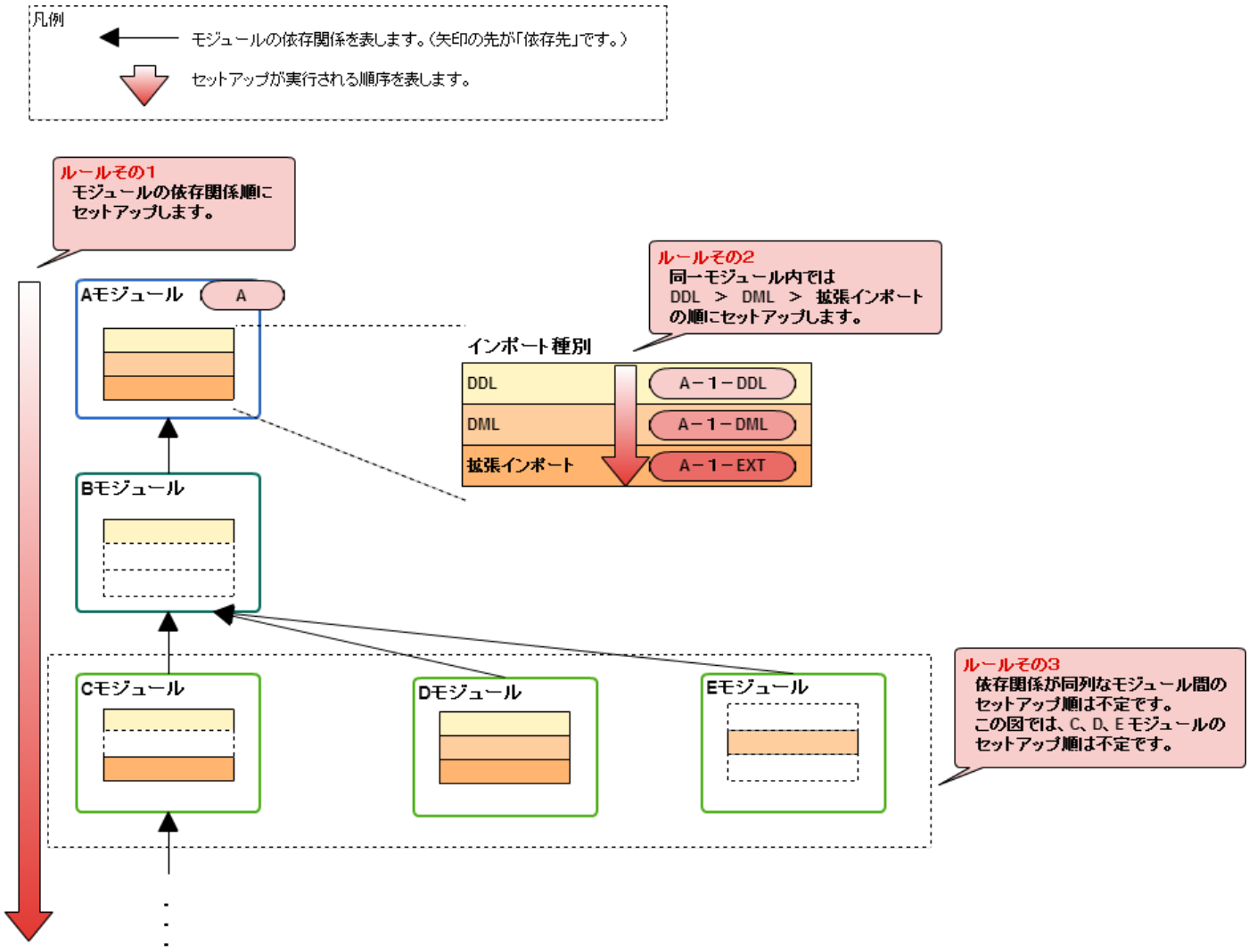
セットアップ設定ファイルで拡張インポートを記述していない場合は当作業は不要です。

拡張インポートの配置先パッケージおよびパスについての規則は特に設けません。
各モジュールにおける適当なパッケージおよびパスで定義してください。

サンプルデータセットアップ 実行方法

システム管理者でログインし、サンプルデータセットアップを実行します。

サンプルデータセットアップ 実行順序



> **注意**
>
> サンプルデータセットアップは、 SchemaUpdate によるセットアップ状況管理の対象外のため、セットアップ実行のたびにすべてのモジュールのサンプルデータセットアップが実行されます。

サンプルデータセットアップ におけるトランザクション制御

サンプルデータセットアップ 実行時のトランザクション単位は次のとおりです。

- 前処理

  トランザクション制御は行いません。トランザクション制御が必要な処理を前処理プログラムで行う場合は、独自にトランザクション制御実装を行う必要があります。

- DDL

  トランザクション制御は行いません。

- DML（DML/テナントマスタ情報のインポート）

  スキーマバージョン単位でトランザクション制御します。

- 拡張インポート

トランザクション制御は行いません。トランザクション制御が必要な処理を拡張インポートで行う場合は、独自にトランザクション制御実装を行う必要があります。

例外発生時の動作

サンプルデータセットアップの実行中に例外が発生した場合、テナント環境セットアップとは異なり、後続のセットアップ処理を継続実行します。

> **コラム**
>
> 理由は、サンプルデータをもつモジュールが新規追加された場合、環境の再構築をせずとも
> サンプルデータセットアップの再実行により情報を最新化する可能性を残すためです。
>
> ただこの場合は、すでにインポート済みの情報に関しては一意制約違反などの例外が多量に発生してしまう恐れがあります。
> また、既存モジュールのバージョンアップの場合は想定したサンプルデータが投入されない恐れもあります。
>
> そのため、開発などの限定的な用途で利用されることを想定しています。

セットアップ実行結果ログ

項目

- セットアップ実行結果ログ
  - 出力先
  - 出力内容
  - 出力例
- インポート処理結果ログ
  - 出力先

セットアップ実行結果ログ

テナント環境セットアップ、サンプルデータセットアップの実行結果ログです。
このファイルではモジュール毎のセットアップの履歴確認や、セットアップ失敗時の詳細ログ参照のための処理実行ID確認を行うことができます。

出力先

システムストレージに出力されます。

- intra-mart Accel Platform 2013 Winter 以前
  - テナント環境セットアップの場合

    <STORAGE_PATH>/system/storage/import_result/basic/import-result-detail-data_yyyy-MM-dd_HH-mm-ss.xml
  - サンプルデータセットアップの場合

    <STORAGE_PATH>/system/storage/import_result/sample/import-result-detail-data_yyyy-MM-dd_HH-mm-ss.xml
- intra-mart Accel Platform 2014 Spring 以降
  - テナント環境セットアップの場合

    <STORAGE_PATH>/system/storage/import_result/basic/<テナントID>/import-result-detail-data_yyyy-MM-dd_HH-mm-ss.xml
  - サンプルデータセットアップの場合

    <STORAGE_PATH>/system/storage/import_result/sample/<テナントID>/import-result-detail-data_yyyy-MM-dd_HH-mm-ss.xml

出力内容

セットアップ内部のインポート処理の単位で、以下の情報が出力されます。

タグ名：属性名	概要	詳細
import-result-detail-data : success	処理結果	処理成功の場合は”true”、処理失敗の場合は”false”
module-id	モジュールID	セットアップ対象のモジュールID
execute-id	処理実行ID	インポート処理の実行ID
import-type	インポート種別	“DDL”、”DML”、”EXTENSION”（拡張インポート）
importer-id	インポータID	インポート処理を特定するID
target-name	インポートファイル名	インポート情報・インポート処理を特定するファイル名
message	エラーメッセージ	例外発生した場合のみ出力されるエラーメッセージ

出力例

- セットアップ実行に成功した場合

```
<?xml version="1.0" encoding="UTF-8"?>
<root xmlns="http://intra-mart.co.jp/system/service/provider/updater/import-result-detail-data">
  <import-result-detail-data success="true">
    <module-id>im_tenant</module-id>
    <execute-id>5becb3e4-e67a-4933-93b2-286377c54f9b</execute-id>
    <import-type>DDL</import-type>
    <importer-id>jp.co.intra_mart.system.service.importer.SQLFileImporter</importer-id>
    <target-name>products/import/basic/im_tenant/im_tenant_common/im_tenant_common-ddl.sql</target-name>
  </import-result-detail-data>
  <import-result-detail-data success="true">
    <module-id>im_portal</module-id>
    <execute-id>97ab409c-78f8-4ae3-bc82-8084ecc4ed12</execute-id>
    <import-type>DML</import-type>
    <importer-id>jp.co.intra_mart.import.StandardAuthzResourceXmlImporter</importer-id>
    <target-name>products/import/basic/im_portal/im_portal-authz-resource_zh_CN.xml</target-name>
  </import-result-detail-data>
  <import-result-detail-data success="true">
    <module-id>im_portal</module-id>
    <execute-id>674e2988-2fde-41fa-bc43-4b9530b4d709</execute-id>
    <import-type>EXTENSION</import-type>
    <importer-id>jp.co.intra_mart.system.service.importer.ExtendsImporter</importer-id>
    <target-name>jp.co.intra_mart.foundation.portal.common.portletadmin.mbeans.PortletImport</target-name>
  </import-result-detail-data>
</root>
```

- セットアップに失敗した場合

該当の処理について、次のようなログが出力されます。（失敗した処理部分のみを抜粋。）
処理結果が false となり、エラーメッセージが出力されます。

```
<import-result-detail-data success="false">
  <module-id>im_workflow</module-id>
  <execute-id>36b27536-5a95-46d2-9752-489b261b20a1</execute-id>
  <import-type>DML</import-type>
  <importer-id>jp.co.intra_mart.import.StandardAuthzPolicyXmlImporter</importer-id>
  <target-name>products/import/basic/im_workflow/im_workflow-authz-policy.xml</target-name>
  <message>[E.IWP.IMPORTEXPORT.IMPORTER.10001] データのインポートに失敗しました。 実行クラス = jp.co.intra_mart.system.authz.services.admin.batch.imex.policy.PolicyXmlImporter, 実行ID = 36b27536-5a95-46d2-9752-489b261b20a1</message>
</import-result-detail-data>
```

> **コラム**
> インポート処理単位での成功/失敗の情報がこのファイルから確認できます。
> より詳しい情報を確認したい場合は、下記の「インポート処理結果ログ」を確認します。

インポート処理結果ログ

テナント環境セットアップ、サンプルデータセットアップでは、複数のインポータを実行することで各種データのインポートを実現しています。
セットアップに失敗した場合、個々のインポート処理の結果ログを参照することにより、例外の原因をつきとめるための判断材料を確認できます。

> **コラム**
> 原因の追跡方法については「ログ仕様書 - インポートエラーの原因をログから追跡する」もあわせて参照してください。

出力先

システムストレージに出力されます。

<STORAGE_PATH>/system/storage/log/import-export/<%処理実行ID%>.log

<%処理実行ID%> には、インポート処理の処理実行IDが適用されます。

セットアップで失敗した場合は、失敗したインポート処理の実行IDによってインポート処理結果ログを特定し、内容を参照することで、例外発生の詳細情報を確認できます。

付録

モジュールIDの制限事項

セットアップでは、「ショートモジュールID」をキーとして動作したり、資材を配置したりします。

そのため、セットアップを定義するモジュール間でショートモジュールIDが重複している場合、
予期せぬ順序でセットアップが実行されてしまったり、資材の混在・上書きが発生してしまいます。
セットアップを定義するモジュール間では、ショートモジュールIDが重複しないよう管理してください。

既存のショートモジュールIDの確認を行うには、
「WARファイルに含まれるモジュール情報・ショートモジュールIDの一覧を確認する方法」を参照してください。

セットアップ用SQLファイル / インポートDDL / インポートDML 記法

1. SQL分の終端にはセミコロン";"を記述してください。
   ひとつのSQLファイルには複数のSQL文の記述が可能です。
   - DDLの例

```
CREATE TABLE im_administrator (
    user_cd             VARCHAR(100) NOT NULL,
    password            VARCHAR(255),
    locale_id           VARCHAR(20),
    encoding            VARCHAR(20),
    time_zone_id        VARCHAR(128),
    telephone_number    VARCHAR(50),
    email_address       VARCHAR(256),
    create_user_cd      VARCHAR(100) NOT NULL,
    create_date         TIMESTAMP NOT NULL,
    record_user_cd      VARCHAR(100) NOT NULL,
    record_date         TIMESTAMP NOT NULL,
    PRIMARY KEY (user_cd)
);

CREATE TABLE im_tenant_info (
    tenant_id           VARCHAR(100) NOT NULL,
    locale_id           VARCHAR(20) NOT NULL,
    encoding            VARCHAR(20),
    time_zone_id        VARCHAR(128),
    calendar_id         VARCHAR(20),
    first_day_of_week   DECIMAL(1) NOT NULL,
    home_url            VARCHAR(1024),
    email_address       VARCHAR(256) NOT NULL,
    lock_count          DECIMAL(15) NOT NULL,
    lock_term           DECIMAL(15) NOT NULL,
    create_user_cd      VARCHAR(100) NOT NULL,
    create_date         TIMESTAMP NOT NULL,
    record_user_cd      VARCHAR(100) NOT NULL,
    record_date         TIMESTAMP NOT NULL,
    PRIMARY KEY (tenant_id)
);

 .
 .
 .
```

   - DMLの例

```
insert into imw_m_list_pattern
(pattern_id,locale_id,list_page_type,pattern_name,default_flag,note,update_count,create_user_code,update_user_code
values ('default_pattern_0','ja','0','申請標準','1','申請標準パターン','1','intra-mart','intra-mart',to_timestamp('2012-08-
08 00:00:00', 'yyyy-mm-dd hh24:mi:ss'),to_timestamp('2012-08-08 00:00:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into imw_m_list_pattern
(pattern_id,locale_id,list_page_type,pattern_name,default_flag,note,update_count,create_user_code,update_user_code
```

```
values ('default_pattern_1','ja','1','一時保存標準','1','一時保存標準パターン','1','intra-mart','intra-
mart',to_timestamp('2012-08-08 00:00:00', 'yyyy-mm-dd hh24:mi:ss'),to_timestamp('2012-08-08 00:00:00', 'yyyy-
mm-dd hh24:mi:ss'));
insert into imw_m_list_pattern
(pattern_id,locale_id,list_page_type,pattern_name,default_flag,note,update_count,create_user_code,update_user_code
values ('default_pattern_2','ja','2','未処理標準','1','未処理標準パターン','1','intra-mart','intra-mart',to_timestamp('2012-
08-08 00:00:00', 'yyyy-mm-dd hh24:mi:ss'),to_timestamp('2012-08-08 00:00:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into imw_m_list_pattern
(pattern_id,locale_id,list_page_type,pattern_name,default_flag,note,update_count,create_user_code,update_user_code
values ('default_pattern_3','ja','3','処理済（未完了）標準','1','処理済（未完了）標準パターン','1','intra-mart','intra-
mart',to_timestamp('2012-08-08 00:00:00', 'yyyy-mm-dd hh24:mi:ss'),to_timestamp('2012-08-08 00:00:00', 'yyyy-
mm-dd hh24:mi:ss'));
insert into imw_m_list_pattern
(pattern_id,locale_id,list_page_type,pattern_name,default_flag,note,update_count,create_user_code,update_user_code
values ('default_pattern_4','ja','4','処理済（完了）標準','1','処理済（完了）標準パターン','1','intra-mart','intra-
mart',to_timestamp('2012-08-08 00:00:00', 'yyyy-mm-dd hh24:mi:ss'),to_timestamp('2012-08-08 00:00:00', 'yyyy-
mm-dd hh24:mi:ss'));
insert into imw_m_list_pattern
(pattern_id,locale_id,list_page_type,pattern_name,default_flag,note,update_count,create_user_code,update_user_code
values ('default_pattern_5','ja','5','過去案件標準','1','過去案件標準パターン','1','intra-mart','intra-
mart',to_timestamp('2012-08-08 00:00:00', 'yyyy-mm-dd hh24:mi:ss'),to_timestamp('2012-08-08 00:00:00', 'yyyy-
mm-dd hh24:mi:ss'));
insert into imw_m_list_pattern
(pattern_id,locale_id,list_page_type,pattern_name,default_flag,note,update_count,create_user_code,update_user_code
values ('default_pattern_6','ja','6','参照（未完了）標準','1','参照（未完了）標準パターン','1','intra-mart','intra-
mart',to_timestamp('2012-08-08 00:00:00', 'yyyy-mm-dd hh24:mi:ss'),to_timestamp('2012-08-08 00:00:00', 'yyyy-
mm-dd hh24:mi:ss'));
insert into imw_m_list_pattern
(pattern_id,locale_id,list_page_type,pattern_name,default_flag,note,update_count,create_user_code,update_user_code
values ('default_pattern_7','ja','7','参照（完了）標準','1','参照（完了）標準パターン','1','intra-mart','intra-
mart',to_timestamp('2012-08-08 00:00:00', 'yyyy-mm-dd hh24:mi:ss'),to_timestamp('2012-08-08 00:00:00', 'yyyy-
mm-dd hh24:mi:ss'));
insert into imw_m_list_pattern
(pattern_id,locale_id,list_page_type,pattern_name,default_flag,note,update_count,create_user_code,update_user_code
values ('default_pattern_8','ja','8','確認（未完了）標準','1','確認（未完了）標準パターン','1','intra-mart','intra-
mart',to_timestamp('2012-08-08 00:00:00', 'yyyy-mm-dd hh24:mi:ss'),to_timestamp('2012-08-08 00:00:00', 'yyyy-
mm-dd hh24:mi:ss'));
insert into imw_m_list_pattern
(pattern_id,locale_id,list_page_type,pattern_name,default_flag,note,update_count,create_user_code,update_user_code
values ('default_pattern_9','ja','9','確認（完了）標準','1','確認（完了）標準パターン','1','intra-mart','intra-
mart',to_timestamp('2012-08-08 00:00:00', 'yyyy-mm-dd hh24:mi:ss'),to_timestamp('2012-08-08 00:00:00', 'yyyy-
mm-dd hh24:mi:ss'));

insert into imw_m_mail_template
(mail_id,locale_id,mail_template_name,note,mail_template_path,update_count,create_user_code,update_user_code,cre
values ('processing','ja','処理依頼','処理依頼用の標準メールテンプレー
ト','im_workflow/data/^^^GROUP_ID^^^/^^^MASTER_FILE_DIR^^^/mail/processing_ja.xml','0','intra-mart','intra-
mart',to_timestamp('2012-08-08 00:00:00', 'yyyy-mm-dd hh24:mi:ss'),to_timestamp('2012-08-08 00:00:00', 'yyyy-
mm-dd hh24:mi:ss'));
insert into imw_m_mail_template
(mail_id,locale_id,mail_template_name,note,mail_template_path,update_count,create_user_code,update_user_code,cre
values ('result','ja','処理結果通知','処理結果通知用の標準メールテンプレー
ト','im_workflow/data/^^^GROUP_ID^^^/^^^MASTER_FILE_DIR^^^/mail/result_ja.xml','0','intra-mart','intra-
mart',to_timestamp('2012-08-08 00:00:00', 'yyyy-mm-dd hh24:mi:ss'),to_timestamp('2012-08-08 00:00:00', 'yyyy-
mm-dd hh24:mi:ss'));
insert into imw_m_mail_template
(mail_id,locale_id,mail_template_name,note,mail_template_path,update_count,create_user_code,update_user_code,cre
values ('reference','ja','参照依頼','参照依頼用の標準メールテンプレー
ト','im_workflow/data/^^^GROUP_ID^^^/^^^MASTER_FILE_DIR^^^/mail/reference_ja.xml','0','intra-mart','intra-
mart',to_timestamp('2012-08-08 00:00:00', 'yyyy-mm-dd hh24:mi:ss'),to_timestamp('2012-08-08 00:00:00', 'yyyy-
mm-dd hh24:mi:ss'));
.
.
.
```

2. **SQL文中のコメントには対応していません。** コメントは記述しないでください。

セットアップにおけるマルチデータベース対応

システムデータベースセットアップのセットアップ用SQLファイル、セットアップ設定ファイルで指定するインポートDDLとインポートDDLのファイル名に、
以下のサフィックスを付加することで、利用しているデータベースの種類に応じた DDL / DML の定義が可能です。

データベース	サフィックス	システムデータベースセットアップ用SQLファイルの例	インポートDDL / インポートDML の例
Oracle Database	“oracle”	import-im_tenant-1_oracle.sql	im_admin-ddl_oracle.sql / im_admin-dml_oracle.sql
PostgreSQL	“postgre”	import-im_tenant-1_postgre.sql	im_admin-ddl_postgre.sql / im_admin-dml_en_postgre.sql
Microsoft SQL Server	“sqlserver”	import-im_tenant-1_sqlserver.sql	im_admin-ddl_sqlserver.sql / im_admin-dml_zh_CN_sqlserver.sql

上記のサフィックスが付加されたSQLファイルが存在する場合は優先的に読み込まれ、サフィックスが付加されていないSQLファイルは読み込まれません。

利用しているデータベースに対応するサフィックスが付加されたSQLファイルが存在しない場合は、サフィックスが付加されていないSQLファイルが読み込まれます。

> コラム
> PostgreSQL に対応するサフィックスは、 **“postgre”** です。”postgres” ではないので注意してください。

IM-Propagation を利用した テナント環境セットアップ実行データの受信

IM-Propagation を利用して テナント環境セットアップ実行データを受信するための、受信側の設定について説明します。

- 受信データ定義の設定

  WEB-INF/conf/propagation-receivers-config ディレクトリ配下に以下の xml を配置します。

```
<?xml version="1.0" encoding="UTF-8"?>
<propagation-receivers-config
   xmlns="http://www.intra-mart.jp/propagation/receivers-config"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.intra-mart.jp/propagation/receivers-config propagation-receivers-config.xsd">
   <receiver source="jp.co.intra_mart.system.service.provider.updater.propagation.UpdatedTenant"
operationType="PROC_COMPLETED">
      <decoder class="{任意のデータ変換クラスの完全修飾クラス名}"/>
      <procedure class="{任意のデータ処理クラスの完全修飾クラス名}"/>
   </receiver>
</propagation-receivers-config>
```

  テナント環境セットアップ実行データ受信設定のための属性値について説明します。

  - （receiver タグ）source 属性

    jp.co.intra_mart.system.service.provider.updater.propagation.UpdatedTenant

    - セットアップされたモジュール情報とそのスキーマバージョンを持つモデルです。
    - モジュール情報をリストで持ちます。モジュール情報は以下のとおりです。

プロパティ	説明	例
id	モジュールID	“jp.co.intra_mart.im_core”
shortId	ショートモジュールID	“im_core”
schemaVersion	スキーマバージョン	1

    - 実際に伝搬されるジェネリックモデルクラスは以下です。

jp.co.intra_mart.foundation.propagation.model.generic.GenericUpdatedTenant

- （receiver タグ）operationType 属性

  “PROC_COMPLETED”

  テナント環境セットアップ実行後に必ず呼び出されます。

> コラム
> IM-Propagation の受信設定の詳細については「intra-mart Accel Platform/ 設定ファイルリファレンス」を参照してください。

フォーマットファイル(xsd)

セットアップ設定

モジュール	インポート・エクスポート
フォーマットファイル(xsd)	WEB-INF/schema/import-data-config.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
   targetNamespace="http://intra_mart.co.jp/system/service/provider/importer/config/import-data-config"
   xmlns:tns="http://intra_mart.co.jp/system/service/provider/importer/config/import-data-config"
   xmlns:jaxb="http://java.sun.com/xml/ns/jaxb" jaxb:version="2.0"
   elementFormDefault="qualified">

<annotation>
 <appinfo>
  <jaxb:schemaBindings>
   <jaxb:package name="jp.co.intra_mart.system.service.provider.importer.config"/>
  </jaxb:schemaBindings>
 </appinfo>
</annotation>

<element name="import-data-config">
 <complexType>
  <sequence>
   <element name="preprocessors" type="tns:preprocessorsType" minOccurs="0" />
   <element name="database" type="tns:databaseConfigType" minOccurs="0"/>
   <element name="tenant-master" type="tns:tenantMasterConfigType" minOccurs="0"/>
   <element name="extends-import" type="tns:extendsImportConfigType" minOccurs="0"/>
  </sequence>
 </complexType>
</element>

<complexType name="preprocessorsType">
 <sequence>
  <element name="preprocessor-class" type="string" minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>

<complexType name="databaseConfigType">
 <sequence>
  <element name="create-file" type="string" minOccurs="0" maxOccurs="unbounded"/>
  <element name="insert-file" type="string" minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>

<complexType name="tenantMasterConfigType">
 <choice minOccurs="0" maxOccurs="unbounded">
  <element name="role-file" type="tns:role-file" />
  <element name="menu-group-category-file" type="tns:menu-group-category-file" />
  <element name="menu-group-file" type="tns:menu-group-file" />
```

```
    <element name="account-file" type="tns:account-file" />
    <element name="calendar-file" type="tns:calendar-file" />
    <element name="calendar-day-set-file" type="tns:calendar-day-set-file" />
    <element name="calendar-day-file" type="tns:calendar-day-file" />
    <element name="calendar-merge-file" type="tns:calendar-merge-file" />
    <element name="job-scheduler-file" type="tns:job-scheduler-file" />
    <element name="authz-resource-group-file" type="tns:authz-resource-group-file" />
    <element name="authz-resource-file" type="tns:authz-resource-file" />
    <element name="authz-subject-group-file" type="tns:authz-subject-group-file" />
    <element name="authz-policy-file" type="tns:authz-policy-file" />
  </choice>
</complexType>

<complexType name="import-file">
  <simpleContent>
    <extension base="string"/>
  </simpleContent>
</complexType>
<complexType name="role-file">
  <simpleContent>
    <restriction base="tns:import-file"/>
  </simpleContent>
</complexType>
<complexType name="account-file">
  <simpleContent>
    <restriction base="tns:import-file"/>
  </simpleContent>
</complexType>
<complexType name="calendar-file">
  <simpleContent>
    <restriction base="tns:import-file"/>
  </simpleContent>
</complexType>
<complexType name="calendar-day-set-file">
  <simpleContent>
    <restriction base="tns:import-file"/>
  </simpleContent>
</complexType>
<complexType name="calendar-day-file">
  <simpleContent>
    <restriction base="tns:import-file"/>
  </simpleContent>
</complexType>
<complexType name="calendar-merge-file">
  <simpleContent>
    <restriction base="tns:import-file"/>
  </simpleContent>
</complexType>
<complexType name="authz-resource-group-file">
  <simpleContent>
    <restriction base="tns:import-file"/>
  </simpleContent>
</complexType>
<complexType name="authz-resource-file">
  <simpleContent>
    <restriction base="tns:import-file"/>
  </simpleContent>
</complexType>
<complexType name="authz-subject-group-file">
  <simpleContent>
    <restriction base="tns:import-file"/>
  </simpleContent>
</complexType>
<complexType name="authz-policy-file">
  <simpleContent>
    <restriction base="tns:import-file"/>
  </simpleContent>
</complexType>
<complexType name="menu-group-category-file">
  <simpleContent>
```

```
    <restriction base="tns:import-file"/>
  </simpleContent>
</complexType>
<complexType name="menu-group-file">
  <simpleContent>
    <restriction base="tns:import-file"/>
  </simpleContent>
</complexType>
<complexType name="job-scheduler-file">
  <simpleContent>
    <restriction base="tns:import-file"/>
  </simpleContent>
</complexType>

<complexType name="extendsImportConfigType">
  <sequence>
    <element name="extends-import-class" type="string" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>

</schema>
```

セットアップ実行結果ログ

モジュール	インポート・エクスポート
フォーマットファイル(xsd)	WEB-INF/schema/import-result-detail-data.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
 elementFormDefault="qualified"
 targetNamespace="http://intra-mart.co.jp/system/service/provider/updater/import-result-detail-data"
 xmlns="http://intra-mart.co.jp/system/service/provider/updater/import-result-detail-data"
 xmlns:jaxb="http://java.sun.com/xml/ns/jaxb" jaxb:version="2.0">

 <xs:annotation>
  <xs:appinfo>
   <jaxb:schemaBindings>
    <jaxb:package name="jp.co.intra_mart.system.service.provider.updater" />
   </jaxb:schemaBindings>
  </xs:appinfo>
 </xs:annotation>

 <xs:element name="import-result-detail-data">
  <xs:complexType>
   <xs:sequence>
    <xs:element minOccurs="0" ref="module-id" />
    <xs:element minOccurs="0" ref="execute-id" />
    <xs:element minOccurs="0" ref="import-type" />
    <xs:element minOccurs="0" ref="importer-id" />
    <xs:element minOccurs="0" ref="target-name" />
    <xs:element minOccurs="0" ref="message" />
   </xs:sequence>
   <xs:attribute name="success" use="required" type="xs:boolean" />
  </xs:complexType>
 </xs:element>

 <xs:element name="module-id" type="xs:string" />
 <xs:element name="execute-id" type="xs:string" />
 <xs:element name="import-type" type="xs:string" />
 <xs:element name="importer-id" type="xs:string" />
 <xs:element name="target-name" type="xs:string" />
 <xs:element name="message" type="xs:string" />

</xs:schema>
```

インポートするデータの依存関係

インポートするデータによっては、他のデータが登録されていることが前提となっていることがあります。
その場合、前提となるデータが未登録のままインポートを実施すると、インポートに失敗する可能性があります。

ここでは、インポートするデータの依存関係（あらかじめ登録が必要なデータ）について説明します。

テナント管理機能

テナント管理機能に関するインポートデータの依存関係は、以下の通りです。
また、複数のデータをインポートする際は、以下の順序で行います。

- ロール
  - サブロールに対する親となるロール
- カレンダー

  インポート前にあらかじめ登録が必要なデータはありません。

- 日付情報セット

  インポート前にあらかじめ登録が必要なデータはありません。

- 日付情報
  - 日付情報セット
- カレンダーマージ設定
  - カレンダー

    - 日付情報セット
- アカウント
    - テーマ（画面テーマ）
    - 日付と時刻の形式
    - カレンダー
    - ロール

> **コラム**
>
> 「日付と時刻の形式」は日付と時刻の形式マスタに定義されているフォーマットセットIDを指定します。

- ジョブ
    - ジョブ、ジョブネットが属するカテゴリ
- 認可（リソースグループ）
    - 親となる認可（リソースグループ）
- 認可（リソース）
    - 親となる認可（リソースグループ）
- 認可（サブジェクトおよびグループ）

  インポート前にあらかじめ登録が必要なデータはありません。

> **注意**
>
> サブジェクトの式に設定されている対象者の実データが登録されていない場合、インポート時にはエラーは発生しませんが、認可の問い合わせを行った際にシステムエラーが発生することがあります。

- メニューグループカテゴリ

  インポート前にあらかじめ登録が必要なデータはありません。

- メニューグループ
    - メニューグループカテゴリ
    - 認可（リソースグループ）

> **コラム**
>
> メニューグループに対する認可リソースを登録するための親となるリソースグループがあらかじめ登録されている必要があります。
> この親リソースグループは、テナント環境セットアップでテナント管理機能のインポートを実行した際に自動的に登録されます。

- 認可（ポリシー）
    - 認可（リソースグループ）
    - 認可（リソース）
    - 認可（サブジェクトおよびグループ）

> **コラム**
>
> メニューグループのインポート時に認可（リソース）が登録されるため、メニューグループのインポート処理後に認可（ポリシー）をインポートします。

インポートするデータの登録方針

更新モードについて

- メニューグループの場合

  メニューグループに用意されている更新モードは以下の通りです。

- 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されます。
- 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未設定となります。
- revive
  - メニューグループについて
    指定したメニューグループIDに紐づくメニューグループが存在する場合、メニューグループの更新を行いません。
  - メニューアイテムについて
    メニューグループ内で指定されているメニューアイテムIDに紐づくメニューアイテムが存在する場合、基本的に更新を行いません（表示名のみ、インポートデータに存在する表示名のうち、まだ存在しないロケールの表示名のみ追加を行います）。

- その他

  その他の更新モードについての詳細は、各データのインポート・エクスポート仕様書を参照してください。

デフォルトの更新モード

- メニューグループ

  メニューグループはインポートによる更新を行う際に、reviveモードによって動作します。
  （更新対象となるデータが存在しない、新規追加処理の場合は、そのまま登録を行います）

- その他

  mergeモードによって動作します。

前処理プログラム サンプル

サンプル内容

このサンプルは 前処理プログラム のJavaクラス・サーバサイドJavaScriptの各サンプルソースになります。 前処理プログラム を実装する場合は以下のサンプルを元に必要な実装を行ってください。

Javaクラス サンプル

Javaで 前処理プログラム を実装する場合は以下のインタフェースを継承したクラスを作成します。

- jp.co.intra_mart.system.service.importer.preprocessing.ImportPreprocessor

```
package jp.co.intra_mart.foo.bar;

import jp.co.intra_mart.system.service.importer.preprocessing.ImportPreprocessingException;
import jp.co.intra_mart.system.service.importer.preprocessing.ImportPreprocessor;

public class SampleImportPreprocessor implements ImportPreprocessor {

  @Override
  public void execute() throws ImportPreprocessingException {
    try {
      // ここに前処理の処理内容を実装します。
      ...
    } catch (final Exception e) {
      throw new ImportPreprocessingException("前処理プログラム失敗時のメッセージ", e);
    }
  }

}
```

ImportPreprocessingExceptionがスローされた場合、セットアップを中断します。

そのため、それ以降の処理は実行されなくなります。
再度セットアップが実行された場合は、例外をスローした前処理プログラムから実行されます。

正常に終了した場合は再度セットアップを行っても実行されません。

> **コラム**
> サンプルデータインポートで前処理プログラムを使用している場合は、インポートが実行されるたびに前処理プログラムが実行されます。

サーバサイドJavaScript サンプル

サーバサイドJavaScriptで 前処理プログラム を実装する場合は execute 関数をもつJsファイルを作成します。

```
function execute() {
  // ここに前処理の処理内容を実装します。
  ...

  // 戻り値にはObject もしくは Boolean 型で処理結果を返却します。
  // 戻り値にBoolean 型を返却する場合は、true : エラー / false : 正常終了 となります。
  return {
    error : false,
    errorMessage : ''
  };
}
```

戻り値をBoolean型で返却する場合は、以下のように返却してください。

- 正常終了の場合 ： true
- 失敗の場合：false

戻り値をObject型で返却する場合は、以下のプロパティを持つオブジェクトを返却してください。

- error：失敗の場合、true（真偽値型）
- errorMessage：失敗時のメッセージ（文字列型）

戻り値に「失敗（ false またはerrorプロパティが true のオブジェクト）」が返却された場合、セットアップを中断します。
そのため、それ以降の処理は実行されなくなります。
再度セットアップが実行された場合は、失敗を返却した前処理プログラムから実行されます。

正常に終了した場合は再度セットアップを行っても実行されません。

> **コラム**
> サンプルデータインポートで前処理プログラムを使用している場合は、インポートが実行されるたびに前処理プログラムが実行されます。