



目次

- 改訂情報
- はじめに
 - 本書の目的
 - 対象読者
 - 本書の構成
- 概要
 - インポート・エクスポートで扱う情報
 - インポート・エクスポートで扱う日付と時刻の情報
- ファイルフォーマット
 - XMLファイルフォーマット
 - CSVファイルフォーマット
 - 互換性
- インポート
 - XML
 - CSV
 - 更新モード
 - インポート時のデータ検証について
 - ライセンスのインポートについて
 - インポートの依存関係
- エクスポート
 - XML
 - CSV
- オプション
 - XMLインポートオプション
 - XMLエクスポートオプション
 - CSVインポートオプション
 - CSVエクスポートオプション
- 実行方法
 - 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
- 付録
 - アカウントインポートデータ定義書
 - 日付/時刻フォーマットパターンに指定可能な文字
 - トラブルシューティング

変更年月日	変更内容
2013-07-01	初版
2014-01-01	第2版 下記を追加・変更しました <ul style="list-style-type: none">■ 各項目に利用可能なバージョン情報について追記■ 「オプション」にデータ検証オプションの説明を追加■ 「インポート」に「インポート時のデータ検証について」を追加■ 「インポート」に「ライセンスのインポートについて」の説明を追加■ 「インポート」に「インポートの依存関係」を追加
2015-08-01	第3版 下記を追加・変更しました <ul style="list-style-type: none">■ 「概要」に アカウントの削除について追記■ 「付録」の「トラブルシューティング」に「旧バージョンでエクスポートしたXMLファイルのインポートに失敗する」を追加
2015-12-01	第4版 下記を追加・変更しました <ul style="list-style-type: none">■ 「インポート」に アカウントをインポートしただけでは intra-mart Accel Platform を利用できない旨を追記
2016-04-01	第5版 下記を追加・変更しました <ul style="list-style-type: none">■ 「エクスポート」にパスワードの保存方式が「ハッシュ化」方式である場合にパスワードを出力できない旨の警告を追加

本書の目的

本書ではアカウント情報のインポート・エクスポート機能の詳細について説明します。

説明範囲は以下のとおりです。

- アカウント情報のインポート・エクスポートの概要
- アカウント情報のインポート・エクスポートファイルのデータフォーマット
- アカウント情報のインポート・エクスポートの実行方法
- アカウント情報のインポート・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 アカウント を管理する運用担当者
- アカウント情報のインポート・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

- **概要**

アカウント情報のインポート・エクスポートで扱う情報について説明します。

- **ファイルフォーマット**

2つのファイル形式(XML形式とCSV形式)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**

インポート・エクスポートの実行方法について説明します。

項目

- [インポート・エクスポートで扱う情報](#)
- [インポート・エクスポートで扱う日付と時刻の情報](#)

アカウントのインポート・エクスポート機能では、XML形式およびCSV形式でアカウント情報のインポート・エクスポートを行います。

アカウント情報とは、アカウントに関連する以下の情報です。

- [アカウント情報](#)
- [アカウントテーマ情報](#)
- [日付と時刻の形式情報](#)
- [アカウント属性情報](#)
- [アカウント付与ロール情報](#)
- [アプリケーションライセンス情報](#)

アカウントの削除を行う機能は提供されていません。

ただし、既存のアカウントを置き換えることは可能です。

詳しくは、「[インポート](#)」-「[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

扱う情報の制限については「[インポート時のデータ検証について](#)」を参照してください。

アカウント情報

アカウントに対しての設定値を保持します。

アカウントの基本情報およびアカウントライセンス情報で構成されています。

アカウント情報を構成する項目は以下の通りです。

項目	対応するテーブル名	対応するカラム名	説明
ユーザコード	b_m_account_b	user_cd	アカウントを表すユーザコードです。
パスワード	b_m_account_b	password	アカウントに設定されているパスワードです。 パスワードは復号された値が設定されます。
週の開始曜日	b_m_account_b	first_day_of_week	アカウントに設定されている週の開始曜日です。 曜日を表す次の数値が設定されます。 1：日曜 ～ 7：土曜 設定されていない場合は、「-1」が設定され ます。

エンコーディング	b_m_account_b encoding	アカウントに設定されている文字エンコーディングです。 intra-mart Accel Platform では、文字エンコーディングは変更できないため、「UTF-8」が設定されます。
ロケールID	b_m_account_b locale_id	アカウントに設定されているロケールIDです。 ユーザが画面表示等で利用するロケールのIDが設定されます。
タイムゾーンID	b_m_account_b time_zone_id	アカウントに設定されているタイムゾーンIDです。 ユーザが日付と時刻の情報を扱う場合に利用するタイムゾーンのIDが設定されます。
カレンダーID	b_m_account_b calendar_id	アカウントに設定されているカレンダーIDです。 ユーザが日付情報を扱う場合に利用するカレンダー情報のIDが設定されます。
アカウントロック日付	b_m_account_b lock_date	アカウントに設定されているアカウントロック日付です。 アカウントがロックされた日時が設定されます。 ロックされていない場合は設定されません。
ログイン失敗回数	b_m_account_b login_failure_count	アカウントのログイン失敗回数です。 ログインに失敗するたびに、カウントが増加します。 ログインに成功した場合はリセットされ、「0」が設定されます。
備考	b_m_account_b notes	アカウントに設定されている備考です。
有効開始日	b_m_account_b valid_start_date	アカウントの有効開始日です。 アカウントは有効開始日から有効です。
有効終了日	b_m_account_b valid_end_date	アカウントの有効終了日です。 アカウントは有効終了日の前日まで有効です。
アカウントライセンス (非公開)	-	アカウントのライセンス付与状況です。 ライセンスが付与されている場合「true」、付与されていない場合は「false」が設定されます。

アカウントテーマ情報

アカウントに設定されたテーマ情報です。

アカウントテーマ情報は、クライアントタイプごとのテーマIDを保持します。

アカウントテーマ情報を構成する項目は以下の通りです。

項目	対応するテーブル名	対応するカラム名	説明
クライアントタイプID	b_m_account_theme	client_type_id	アカウントテーマ情報に設定されているクライアントタイプIDです。

テーマID	b_m_account_theme theme_id	アカウントテーマ情報に設定されているクライアントタイプに対するテーマのテーマIDです。
-------	----------------------------	---

日付と時刻の形式情報

アカウントに設定された日付と時刻の形式情報です。

日付と時刻の形式情報は、日付や時刻を画面などに表示する場合の形式情報のIDとフォーマットパターンを保持します。

日付と時刻の形式情報を構成する項目は以下の通りです。

項目	対応するテーブル名	対応するカラム名	説明
フォーマットセットID	b_m_account_dtf	format_set_id	日付と時刻の形式情報に設定されているフォーマットセットIDです。 フォーマットセットとは、あらかじめ定義された複数の形式情報のセットです。
ロケールID	b_m_account_dtf	locale_id	日付と時刻の形式情報に設定されているロケールIDです。 フォーマットセットで利用するロケールのIDです。
フォーマットID	b_m_account_dtf	format_id	日付と時刻の形式情報に設定されているフォーマットIDです。 以下のような特定の形式を表すIDです。 日付標準表示、日付簡易表示、時刻標準表示、タイムスタンプ表示
フォーマットパターン	b_m_account_dtf	pattern	日付と時刻の形式情報に設定されている日付/時刻フォーマットパターンです。 表示形式を指定するパターン文字を利用したパターンが設定されます。



コラム

日付と時刻の形式情報については、ドキュメント「[日付と時刻の形式](#)」を参照してください。



コラム

表示形式を指定するパターン文字については、以下を参照してください。

「[日付/時刻フォーマットパターンに指定可能な文字](#)」

アカウント属性情報

アカウントに設定された属性情報です。

アカウント属性情報は、アプリケーションで利用される任意の名前と値のセットです。

アカウント属性情報を構成する項目は以下の通りです。

項目	対応するテーブル名	対応するカラム名	説明
属性名	b_m_account_attr_b	attr_name	アカウント属性情報に設定されている属性名です。
属性値	b_m_account_attr_b	attr_value	アカウント属性情報に設定されている属性名に対する属性値です。

アカウント付与ロール情報

アカウントに付与されたロール情報です。

アカウント付与ロール情報は、付与されたロールIDと有効期間を保持します。

アカウント付与ロール情報を構成する項目は以下の通りです。

項目	対応するテーブル名	対応するカラム名	説明
ロールID	b_m_account_role_b	role_id	アカウント付与ロール情報に設定されているロールIDです。
有効開始日	b_m_account_role_b	valid_start_date	アカウント付与ロール情報に設定されている有効開始日です。 アカウントに付与されたロールは有効開始日から有効です。
有効終了日	b_m_account_role_b	valid_end_date	アカウント付与ロール情報に設定されている有効終了日です。 アカウントに付与されたロールは有効終了日の前日まで有効です。

アプリケーションライセンス情報

アカウントに付与されたアプリケーションライセンス情報です。

アプリケーションライセンス情報は、ライセンス付与されたアプリケーションのプロダクトIDを保持します。

アプリケーションライセンス情報を構成する項目は以下の通りです。

項目	対応するテーブル名	対応するカラム名	説明
プロダクトID	(非公開)	-	アプリケーションライセンス情報に設定されているプロダクトIDです。

インポート・エクスポートで扱う日付と時刻の情報

アカウントのインポート・エクスポート機能で扱う「アカウントロック日付」などの日付と時刻の情報は、データ上文字列として設定されます。

そのため、日付と時刻の情報に変換するために以下のフォーマットパターンの指定が必要です。

- 日付/時刻フォーマットパターン

日付と時刻を扱うデータのフォーマットパターン。

「[アカウント情報](#)」の「アカウントロック日付」で利用されます。

インポート／エクスポート実行時に、[オプション](#)の [日付/時刻フォーマットパターン](#) で指定可能です。

指定しなかった場合、デフォルトのフォーマットパターン「yyyy-MM-dd HH:mm:ss.SSS」が利用されま
す。

- 日付フォーマットパターン

日付のみを扱うデータのフォーマットパターン。

「[アカウント情報](#)」の「有効開始日」「有効終了日」および「[アカウント付与ロール情報](#)」の「有効開始
日」「有効終了日」で利用されます。

インポート／エクスポート実行時に、[オプション](#)の [日付フォーマットパターン](#) で指定可能です。

指定しなかった場合、デフォルトのフォーマットパターン「yyyy-MM-dd」が利用されます。

コラム

変換する場合のタイムゾーンは、システム・デフォルトのタイムゾーンが利用されます。

項目

- XMLファイルフォーマット
- CSVファイルフォーマット
- 互換性

この章では、アカウントのインポート・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。

XMLファイルフォーマット

`<account-data>` タグ内に1つのアカウントに関連する情報をすべてを記述します。

[アカウント情報](#) を `<account-data>` タグに記述します。

`cd` 属性にユーザコードを記述します。

タグ内には、パスワード、週の開始曜日、エンコーディング、ロケールID、タイムゾーンID、カレンダーID、アカウントロック日付、ログイン失敗回数、備考、有効開始日、有効終了日、アカウントライセンスを記述します。

[アカウントテーマ情報](#) を `<theme-ids>` タグに記述します。

設定数の数だけ `<theme-info>` タグを記述します。

`client-type-id` 属性にクライアントタイプIDを、`theme-id` 属性にテーマIDを記述します。

[日付と時刻の形式情報](#) を `<date-time-formats>` タグに記述します。

`<date-time-formats>` タグには、`format-set-id` 属性にフォーマットセットIDを、`locale-id` 属性にロケールIDを記述します。

設定数の数だけ `<date-time-format>` タグを記述します。

`id` 属性にフォーマットID、`pattern` 属性にフォーマットパターンを記述します。

[アカウント属性情報](#) を `<account-attributes>` タグに記述します。

設定数の数だけ `<account-attribute>` タグを記述します。

`key` 属性に属性名、`value` 属性に属性値を記述します。

[アカウント付与ロール情報](#) を `<account-roles>` タグに記述します。

設定数の数だけ `<account-role>` タグを記述します。

`<account-role>` タグには、`id` 属性にロールIDを指定します。

`<role-valid-start-date>` タグで有効開始日を指定します。

`<role-valid-end-date>` タグで有効終了日を指定します。

[アプリケーションライセンス情報](#) を `<application-licenses>` タグに記述します。

設定数の数だけ `<application-license>` タグを記述します。

`id` 属性にプロダクトIDを記述します。

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

以下はXMLファイルの例です。

```

<root xmlns="http://intra-mart.co.jp/system/admin/account/account-data">
  <account-data cd="aoyagi">
    <password>aoyagi_password</password>
    <theme-ids>
      <theme-info theme-id="im_theme_dropdown_blue" client-type-id="pc" />
    </theme-ids>
    <date-time-formats locale-id="ja" format-set-id="IM_DATETIME_FORMAT_SET_JA_BASE">
      <date-time-format pattern="yyyy/MM/dd" id="IM_DATETIME_FORMAT_DATE_STANDARD" />
      <date-time-format pattern="MM/dd" id="IM_DATETIME_FORMAT_DATE_SIMPLE" />
      <date-time-format pattern="H:mm" id="IM_DATETIME_FORMAT_TIME_STANDARD" />
      <date-time-format pattern="H:mm:ss" id="IM_DATETIME_FORMAT_TIME_TIMESTAMP" />
    </date-time-formats>
    <account-attributes>
      <account-attribute value="5" key="im_menu.global_nav.dropdown.limitation" />
    </account-attributes>
    <account-roles>
      <account-role id="im_workflow_user">
        <role-valid-start-date>1900-01-01</role-valid-start-date>
        <role-valid-end-date>3000-01-01</role-valid-end-date>
      </account-role>
    </account-roles>
    <account-license>true</account-license>
    <application-licenses>
      <application-license id="CSUSD" />
    </application-licenses>
  </account-data>
  <account-data cd="ueda">
    <password>ueda_password</password>
  </account-data>
</root>

```

CSVファイルフォーマット

アカウント情報、アカウントテーマ情報、日付と時刻の形式情報、アカウント属性情報、アカウント付与ロール情報、アプリケーションライセンス情報の情報をそれぞれ1行で記述します。

1列目にはデータ種別を指定します。2列目にはユーザコードを指定します。それ以降の列はデータ種別毎に設定値を記述します。

アカウント情報 のデータ種別は `account-data` です。

3列目以降は、パスワード、週の開始曜日、エンコーディング、ロケールID、タイムゾーンID、カレンダーID、アカウントロック日付、ログイン失敗回数、備考、有効開始日、有効終了日、アカウントライセンス の順に列に記述します。

アカウントテーマ情報 のデータ種別は `theme-ids` です。

3列目以降は、クライアントタイプID、テーマID の順に、設定値の数だけ繰り返して記述します。

日付と時刻の形式情報 のデータ種別は `date-time-formats` です。

3列目には、フォーマットセットID、4列目には、ロケールIDを指定します。

5列目以降は、フォーマットID、フォーマットパターン の順に、設定値の数だけ繰り返して記述します。

アカウント属性情報 のデータ種別は `account-attributes` です。

3列目以降は、属性名、属性値 の順に、設定値の数だけ繰り返して記述します。

[アカウント付与ロール情報](#) のデータ種別は `account-roles` です。

3列目以降は、ロールID、有効開始日、有効終了日 の順に、設定値の数だけ繰り返して記述します。

[アプリケーションライセンス情報](#) のデータ種別は `application-licenses` です。

3列目以降は、プロダクトIDを設定値の数だけ繰り返して記述します。

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

以下はCSVファイルの例です。

```
account-data,aoyagi,aoyagi_password,,,,,,,,,true
theme-ids,aoyagi,pc,im_theme_dropdown_blue
date-time-
formats,aoyagi,IM_DATETIME_FORMAT_SET_JA_BASE,ja,IM_DATETIME_FORMAT_DATE_STANDARD,yyyy/MM/dd

account-attributes,aoyagi,im_menu.global_nav.dropdown.limitation,5
account-roles,aoyagi,im_workflow_user,1900-01-01,3000-01-01
application-licenses,aoyagi,CSUSD
account-data,ueda,ueda,,,,,,,,,false
```

互換性

intra-mart Accel Platform で利用するインポートファイルは、intra-mart WebPlatform/AppFramework で利用するインポートファイルとは互換性はありません。

intra-mart WebPlatform/AppFramework で利用していたインポートファイルをそのまま利用したい場合、互換機能を利用してください。

互換機能を利用するためには、「[互換ガイド](#)」を参照してください。

項目

- XML
- CSV
- 更新モード
- インポート時のデータ検証について
- ライセンスのインポートについて
- インポートの依存関係

アカウントのインポートはXML形式とCSV形式で行うことが可能です。
どちらの形式でも同様にマスタ情報の新規登録、更新できます。

この章では、それぞれの形式における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。

 注意

intra-mart Accel Platform を利用する際は、アカウントだけでなく、対応するIM-共通マスタのユーザのプロファイル情報が存在している必要があります。例えば、新規アカウントをインポートしただけでは、「ユーザメンテナンス」画面を表示した際に「プロファイル：ユーザ情報取得に失敗しました。」というメッセージのエラーが発生します。

ユーザのプロファイル情報のインポートに関しては、「[IM-共通マスタ インポート・エクスポート仕様書](#)」を参照してください。

XML

`<account-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。

また、XML形式の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。

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

CSV

連続して同じユーザコードが指定されている行を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「[CSV ファイルフォーマット](#)」を参照してください。

また、CSV形式の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。

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

 注意

CSV形式のインポートは intra-mart Accel Platform 2013 Summer 以降で利用可能です。

 コラム

Excel を利用して、**CSV**を編集する場合について

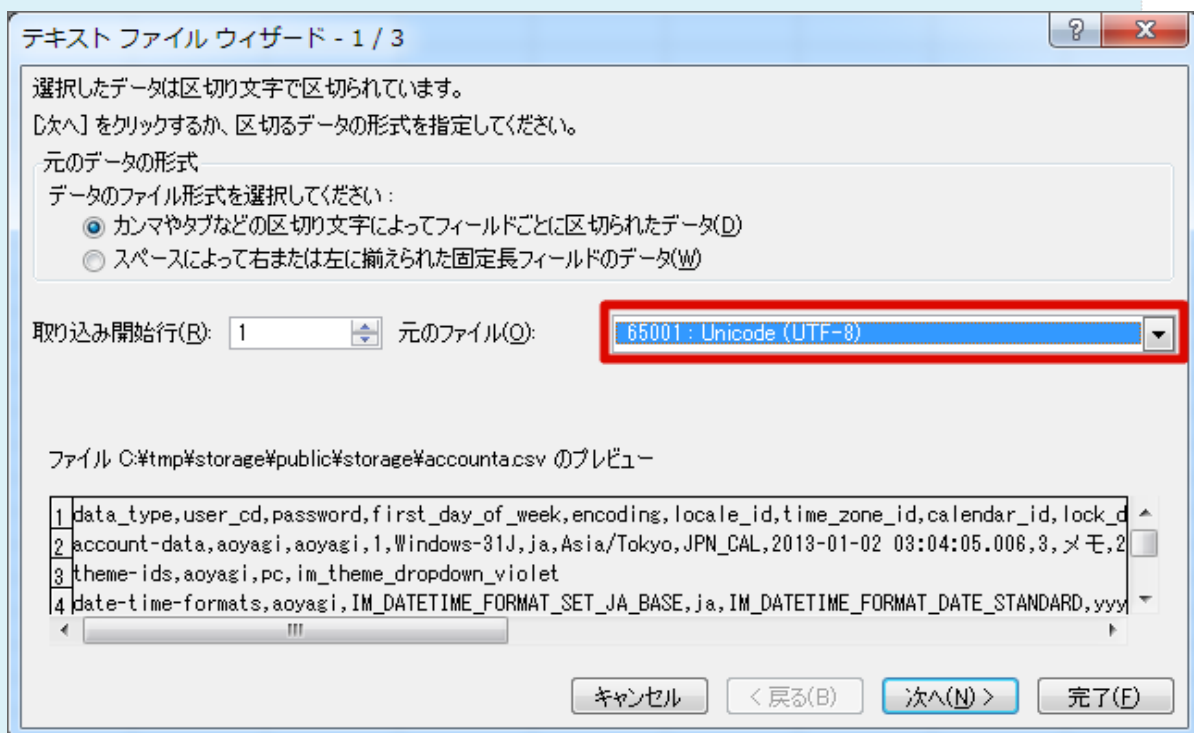
オプションを指定せずにエクスポートした CSV形式のファイルをExcel で直接開いた場合、日付/時刻データが自動的に変換される場合があります。

また、日本語などの2バイト文字が含まれていた場合は、文字化けが発生します。

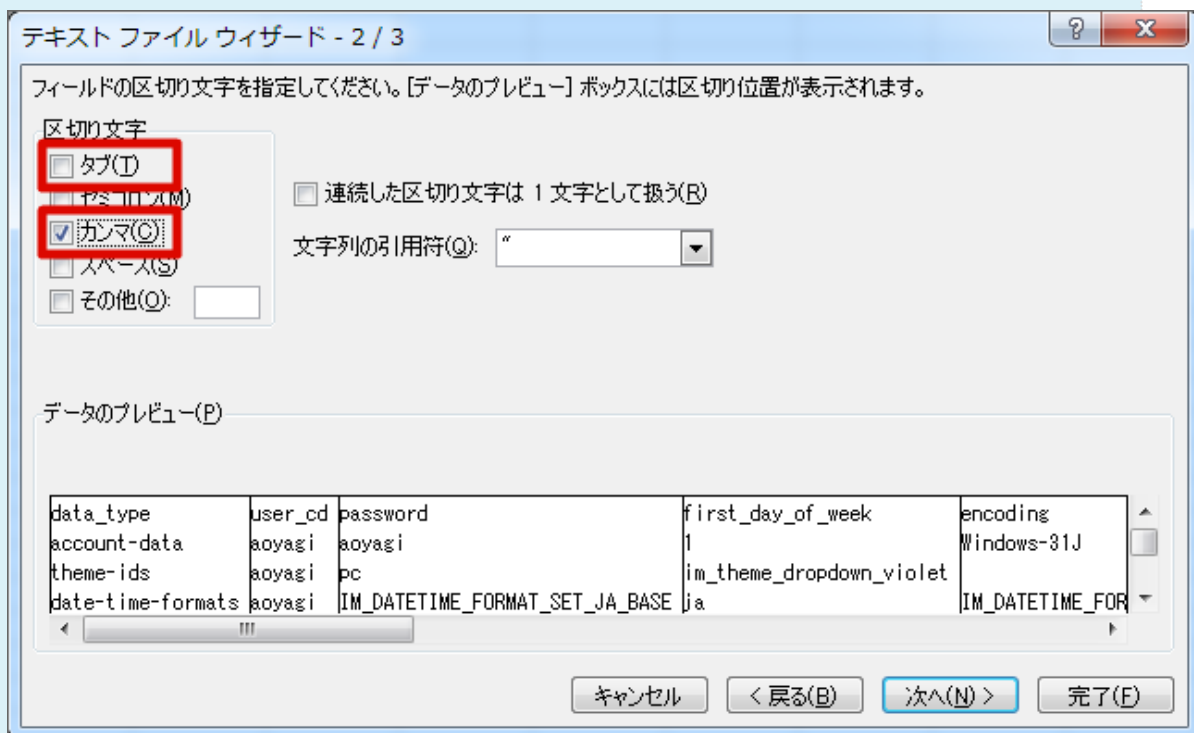
そのためファイルを直接開くのではなく、全てのデータを文字列として扱うようにしてください。

例えば、Excel 2013 では、以下の手順で文字列として扱うことが可能です。

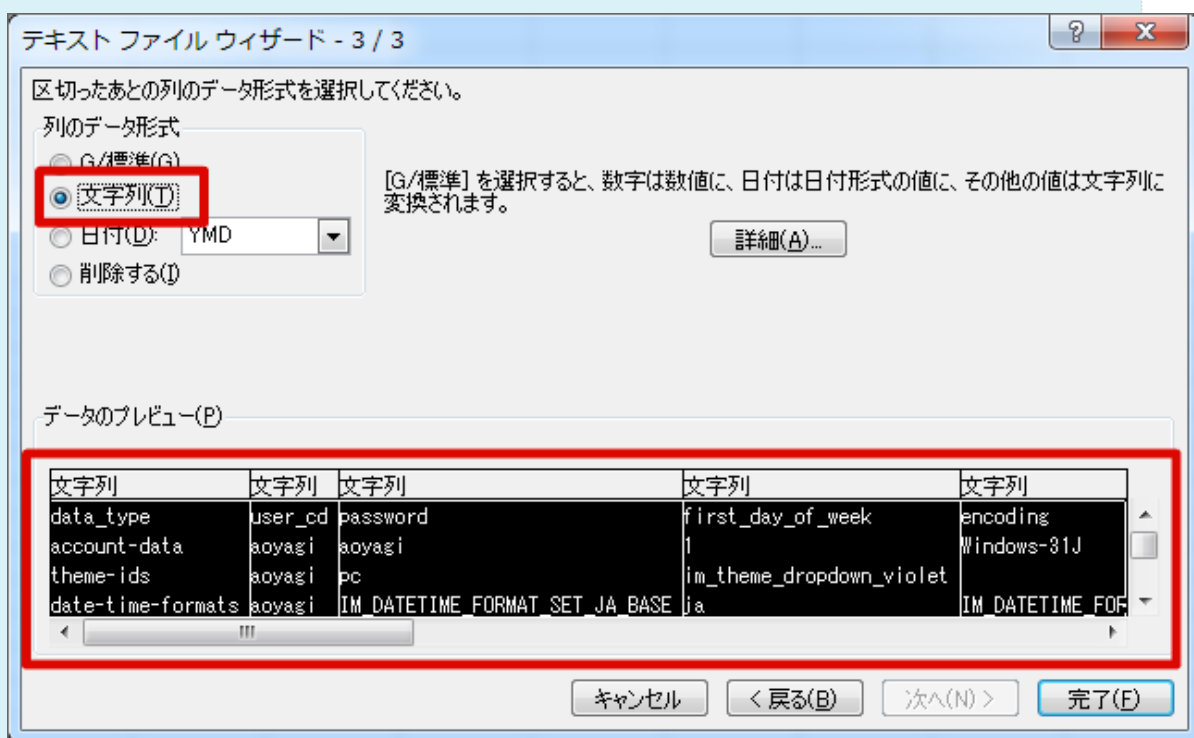
1. 新規に空のファイルを作成します。
2. メニューから「データ」-「外部データの取込」-「テキストファイル」を選択します。
3. テキスト ファイル ウィザード 1/3 で元のファイルで「65001 : Unicode (UTF-8)」を選択し、「次へ」をクリックします。
(2バイト文字が含まれている場合)



4. テキスト ファイル ウィザード 2/3 で区切り文字を「タブ」のチェックを外し、「カンマ」にチェックをして「次へ」をクリックします。



5. テキスト ファイル ウィザード 3/3 でデータのプレビューの表の全ての列を選択し（Shiftキーを押しながら選択できます）、「文字列」を選択して「完了」をクリックします。



6. データの取り込み ダイアログボックスで、「OK」をクリックします。

また、編集したファイルをCSV形式で保存した場合、文字エンコーディングが「Windows-31J」で保存されます。

そのため、インポートする場合は [エンコーディング](#) オプションに「Windows-31J」を指定して実行してください。

更新モードを利用することで、インポートファイルのデータがデータベース上に存在する場合（更新を行う場合）のデータの更新方法を変更できます。

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

XMLインポートの場合は、`<account-data>` タグに `update-mode` 属性を指定することでモードを設定します。CSVインポートの場合は、「*CSVインポートオプション*」の `update-mode` を指定することでモードを設定します。

特に指定していない場合は、*merge* モードで動作します。

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です。



インポート時のデータ検証について

インポート実行時には登録・更新処理を実行する前にデータの検証を行います。データ検証にて条件に一致しない場合はインポートに失敗します。

データ検証の実行の有無は「*オプション*」にて変更可能です。

データ検証内容については以下の通りです。

アカウント情報の検証

- ユーザコード

ユーザコードは1文字以上の文字列を必ず指定する必要があります。

ユーザコードは半角英字、数字、アンダースコア、ハイフン、アットマーク、ドット、プラス、または、エクスクラメーションに限定されます。

ユーザコードは100文字以内である必要があります。

- パスワード

パスワードはパスワード履歴管理で設定されている制約で許容されている値である必要があります。



コラム

パスワード履歴管理の設定については「[設定ファイルリファレンス - パスワード履歴管理設定](#)」を参照してください。

- 週の開始曜日

-1 (未設定)、1 (日曜日)、2 (月曜日)、3 (火曜日)、4 (水曜日)、5 (木曜日)、6 (金曜日)、または、7 (土曜日) のみ指定可能です。

- ロケールID

ロケールマスタに定義されているロケールのロケールIDを指定する必要があります。



コラム

ロケールマスタの設定については「[設定ファイルリファレンス - ロケールマスタ](#)」を参照してください。

- タイムゾーンID

タイムゾーンマスタに定義されているタイムゾーンのタイムゾーンIDを指定する必要があります。



コラム

タイムゾーンマスタの設定については「[設定ファイルリファレンス - タイムゾーン](#)」を参照してください。

- カレンダーID

カレンダー情報として登録されているカレンダーのカレンダーIDを指定する必要があります。

- アカウントロック日付

規定のフォーマットを指定する必要があります。詳細は「[インポート・エクスポートで扱う日付と時刻の情報](#)」を参照してください。

システム期間情報で設定されているの日付を指定する必要があります。



コラム

システム期間情報の設定については「[設定ファイルリファレンス - システム期間情報の管理](#)」を参照してください。

- ログイン失敗回数

ログイン失敗回数は0から99999の範囲で指定する必要があります。

- 備考

備考は63文字以内で指定する必要があります。

- 有効開始日

規定のフォーマットを指定する必要があります。詳細は「[インポート・エクスポートで扱う日付と時刻の情報](#)」を参照してください。

システム期間情報で設定されている日付を指定する必要があります。



コラム

システム期間情報の設定については「[設定ファイルリファレンス - システム期間情報の管理](#)」を参照してください。

- 有効終了日

規定のフォーマットを指定する必要があります。詳細は「[インポート・エクスポートで扱う日付と時刻の情報](#)」を参照してください。

システム期間情報で設定されている日付を指定する必要があります。



コラム

システム期間情報の設定については「[設定ファイルリファレンス - システム期間情報の管理](#)」を参照してください。

アカウントテーマ情報の検証

- クライアントタイプID

クライアントタイプマスタに定義されているクライアントタイプのクライアントタイプIDを指定する必要があります。



コラム

クライアントタイプマスタの設定については「[設定ファイルリファレンス - クライアントタイプマスタ](#)」を参照してください。

- テーマID

テーマ設定に定義されているテーマのテーマIDを指定する必要があります。

指定したテーマがサポートするクライアントタイプにクライアントタイプIDが含まれている必要があります。



コラム

intra-mart Accel Platform が標準で提供するテーマについては「[intra-mart が提供するテーマの一覧](#)」を参照してください。

日付と時刻の形式情報の検証

- フォーマットセットID

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を指定する必要があります。

**コラム**

日付と時刻の形式マスタの設定については「[設定ファイルリファレンス - 日付と時刻の形式マスタ](#)」を参照してください。

- フォーマットID

フォーマットIDは100文字以内である必要があります。

- フォーマットパターン

フォーマットパターンは100文字以内である必要があります。

アカウント属性情報の検証

- 属性名

属性名は255文字以内である必要があります。

- 属性値

属性値は255文字以内である必要があります。

アカウント付与ロール情報の検証

- ロールID

ロール情報として登録されているロールのロールIDを指定する必要があります。

- 有効開始日

規定のフォーマットを指定する必要があります。詳細は「[インポート・エクスポートで扱う日付と時刻の情報](#)」を参照してください。

システム期間情報で設定されている日付を指定する必要があります。

**コラム**

システム期間情報の設定については「[設定ファイルリファレンス - システム期間情報の管理](#)」を参照してください。

- 有効終了日

規定のフォーマットを指定する必要があります。詳細は「[インポート・エクスポートで扱う日付と時刻の情報](#)」を参照してください。

システム期間情報で設定されている日付を指定する必要があります。

**コラム**

システム期間情報の設定については「[設定ファイルリファレンス - システム期間情報の管理](#)」を参照してください。

アプリケーションライセンス情報の検証

- プロダクトID

プロダクトIDは100文字以内である必要があります。

ライセンスの最大登録数の制限などにより、アカウントライセンスや、アプリケーションライセンス情報のインポートに失敗した場合は、インポートは失敗扱いとはならず、処理を続行します。
この場合、以下のようなログが出力されます。

```
[WARN] j.c.i.s.a.a.d.AbstractAccountInfoImporter - [W.IWP.IMPORTER.ACCOUNT.20001] ユーザ (sample) にアカウントライセンスが付与できませんでした。  
jp.co.intra_mart.system.secure.license.exception.OverMaxLicenseException: [E.IWP.LICENSE.USER.00001]  
ライセンス数の上限を超えた為、登録できませんでした。
```

```
[WARN] j.c.i.s.a.a.d.AbstractAccountInfoImporter - [W.IWP.IMPORTER.ACCOUNT.20003] ユーザ (sample) にアプリケーションライセンス (*****) が付与できませんでした。  
jp.co.intra_mart.system.secure.license.exception.OverMaxLicenseException: [E.IWP.LICENSE.USER.00001]  
ライセンス数の上限を超えた為、登録できませんでした。
```

インポートの依存関係

アカウントをインポートする際にあらかじめ登録が必要なデータは、以下の通りです。

- 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。

項目

- XML
- CSV

アカウントのエクスポートはXML形式とCSV形式で行うことが可能です。
どちらの形式でも同様にすべてのマスタ情報を出力できます。

この章では、それぞれの形式におけるエクスポートがどのように行われるかを説明します。



注意

パスワードの保存方式が「ハッシュ化」方式である場合、エクスポート実行時にパスワードの出力が行われません。

パスワードの保存方式については「[システム管理者操作ガイド](#)」-「[パスワード保存方式設定](#)」を参照してください。

XML

すべてのアカウントとそのアカウントに関連する情報を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、XML形式の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。
XML形式のエクスポート時に利用可能なオプションについては、「[XMLエクスポートオプション](#)」を参照してください。

CSV

すべてのアカウントとそのアカウントに関連する情報をCSV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「[CSVファイルフォーマット](#)」を参照してください。

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



注意

CSV形式のエクスポートは intra-mart Accel Platform 2013 Summer 以降で利用可能です。

項目

- XMLインポートオプション
- XMLエクスポートオプション
- CSVインポートオプション
- CSV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。

XMLインポートオプション

この項では、XMLファイルで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。

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
日付/時刻フォーマットパターン	date-time-format-pattern	文字列	yyyy-MM-dd HH:mm:ss.SSS	2013 Summer
日付/時刻フォーマットロケールID	date-time-locale-id	文字列	en	2013 Summer
日付フォーマットパターン	date-format-pattern	文字列	yyyy-MM-dd	2013 Summer
日付フォーマットロケールID	date-locale-id	文字列	en	2013 Summer
XML検証フラグ	validate-xml	真偽値	true	2012 Autumn
データ検証フラグ	validate-data	真偽値	true	2013 Winter

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。

日付/時刻フォーマットパターン

キー名 date-time-format-pattern

インポートするXMLファイル内の日時項目のフォーマットを指定します。

「[アカウント情報](#)」の「アカウントロック日付」で利用します。

フォーマットに使用可能なパターン文字については「[日付/時刻フォーマットパターンに指定可能な文字](#)」を参照してください。

日付/時刻フォーマットロケールID

キー名 date-time-locale-id

インポートするXMLファイル内の日時項目のロケールIDを指定します。

[日付/時刻フォーマットパターン](#)と一緒に指定します。

「[アカウント情報](#)」の「アカウントロック日付」で利用します。

日付フォーマットパターン

キー名 date-format-pattern

インポートするXMLファイル内の日付項目のフォーマットを指定します。

「[アカウント情報](#)」の「有効開始日」「有効終了日」および「[アカウント付与ロール情報](#)」の「有効開始日」「有効終了日」で利用します。

フォーマットに使用可能なパターン文字については「[日付/時刻フォーマットパターンに指定可能な文字](#)」を参照してください。

キー名 date-locale-id

インポートするXMLファイル内の日付項目のロケールIDを指定します。
[日付フォーマットパターン](#)と一緒に指定します。

「[アカウント情報](#)」の「有効開始日」「有効終了日」および「[アカウント付与ロール情報](#)」の「有効開始日」「有効終了日」で利用します。

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。

データ検証フラグ

キー名 validate-data

インポートするデータの検証を行うかどうかを指定します。

指定する値	説明
true	データの検証を行います。（デフォルト値）
false	データの検証を行いません。

XMLエクスポートオプション

この項では、XMLファイルで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。

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
読み込み件数	fetch-count	数値	5000	2012 Autumn

名前	キー名	型	デフォルト値	導入バージョン
日付/時刻フォーマットパターン	date-time-format-pattern	文字列	yyyy-MM-dd HH:mm:ss.SSS	2013 Summer
日付/時刻フォーマットロケールID	date-time-locale-id	文字列	en	2013 Summer
日付フォーマットパターン	date-format-pattern	文字列	yyyy-MM-dd	2013 Summer
日付フォーマットロケールID	date-locale-id	文字列	en	2013 Summer
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

エンコーディング

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。

読み込み件数

キー名 [fetch-count](#)

データベースからエクスポート対象のデータを一度に読み込む件数を指定します。

日付/時刻フォーマットパターン

キー名 [date-time-format-pattern](#)

エクスポートするXMLファイル内の日時項目のフォーマットを指定します。

「[アカウント情報](#)」の「アカウントロック日付」で利用します。
フォーマットに使用可能なパターン文字については「[日付/時刻フォーマットパターンに指定可能な文字](#)」を参照し

日付/時刻フォーマットロケールID

キー名 date-time-locale-id

エクスポートするXMLファイル内の日時項目のロケールIDを指定します。
[日付/時刻フォーマットパターン](#)と一緒に指定します。

「[アカウント情報](#)」の「アカウントロック日付」で利用します。

日付フォーマットパターン

キー名 date-format-pattern

エクスポートするXMLファイル内の日付項目のフォーマットを指定します。

「[アカウント情報](#)」の「有効開始日」「有効終了日」および「[アカウント付与ロール情報](#)」の「有効開始日」「有効終了日」で利用します。

フォーマットに使用可能なパターン文字については「[日付/時刻フォーマットパターンに指定可能な文字](#)」を参照してください。

日付フォーマットロケールID

キー名 date-locale-id

エクスポートするXMLファイル内の日付項目のロケールIDを指定します。
[日付フォーマットパターン](#)と一緒に指定します。

「[アカウント情報](#)」の「有効開始日」「有効終了日」および「[アカウント付与ロール情報](#)」の「有効開始日」「有効終了日」で利用します。

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。

CSVインポートオプション

この項では、CSVファイルで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。

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	2013 Summer
ファイルパス	file	文字列	(なし)	2013 Summer
コミット件数	commit-count	数値	0	2013 Summer
日付/時刻フォーマットパターン	date-time-format-pattern	文字列	yyyy-MM-dd HH:mm:ss.SSS	2013 Summer
日付/時刻フォーマットロケールID	date-time-locale-id	文字列	en	2013 Summer
日付フォーマットパターン	date-format-pattern	文字列	yyyy-MM-dd	2013 Summer
日付フォーマットロケールID	date-locale-id	文字列	en	2013 Summer
データ検証フラグ	validate-data	真偽値	true	2013 Winter
ヘッダ行有無フラグ	with-header	真偽値	false	2013 Summer
区切り文字	delimiter-code	文字列	, (半角カンマ)	2013 Summer
引用符	quote-code	文字列	" (半角ダブルクォー ト)	2013 Summer
CSVフォーマットパターン	csv-format-pattern	文字列	standard	2013 Summer
更新モード	update-mode	文字列	merge	2013 Summer
null代替文字列	null-string	文字列	(空文字列)	2013 Summer

キー名 encoding

インポートするCSV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CSV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。

日付/時刻フォーマットパターン

キー名 date-time-format-pattern

インポートするCSVファイル内の日時項目のフォーマットを指定します。

「[アカウント情報](#)」の「アカウントロック日付」で利用します。

フォーマットに使用可能なパターン文字については「[日付/時刻フォーマットパターンに指定可能な文字](#)」を参照してください。

日付/時刻フォーマットロケールID

キー名 date-time-locale-id

インポートするCSVファイル内の日時項目のロケールIDを指定します。

[日付/時刻フォーマットパターン](#) と一緒に指定します。

「[アカウント情報](#)」の「アカウントロック日付」で利用します。

日付フォーマットパターン

キー名 date-format-pattern

インポートするXMLファイル内の日付項目のフォーマットを指定します。

「[アカウント情報](#)」の「有効開始日」「有効終了日」および「[アカウント付与ルール情報](#)」の「有効開始日」「有効終了日」で利用します。

フォーマットに使用可能なパターン文字については「[日付/時刻フォーマットパターンに指定可能な文字](#)」を参照してください。

日付フォーマットロケールID

キー名 date-locale-id

インポートするXMLファイル内の日付項目のロケールIDを指定します。

[日付フォーマットパターン](#)と一緒に指定します。

「[アカウント情報](#)」の「有効開始日」「有効終了日」および「[アカウント付与ロール情報](#)」の「有効開始日」「有効終了日」で利用します。

データ検証フラグ

キー名 validate-data

インポートするデータの検証を行うかどうかを指定します。

指定する値	説明
true	データの検証を行います。（デフォルト値）
false	データの検証を行いません。

ヘッダ行有無フラグ

キー名 with-header

インポートするCSVファイルがヘッダ行を含むかどうかを指定します。

指定する値	説明
true	ヘッダ行（最初の1件）を除いてインポートします。
false	最初の1行からデータとみなしてインポートします。（デフォルト値）

区切り文字

キー名 delimiter-code

インポートするCSVファイルの各項目を区切る文字列を指定します。

i コラム

このオプションには、以下の特殊文字を指定できます。

t	タブ文字
r	CR
n	LF
	半角バックスラッシュ
	シュ

! 注意

`csv-format-pattern` オプションを明示的に指定した場合は、このオプションは利用できません。

引用符

キー名 `quote-code`

インポートするCSVファイルの各項目を囲むための引用符の文字列を指定します。

! 注意

`csv-format-pattern` オプションを明示的に指定した場合は、このオプションは利用できません。

CSVフォーマットパターン

キー名 `csv-format-pattern`

インポートするCSVファイルのフォーマット情報を指定します。

区切り文字 (`delimiter-code`)、引用符 (`quote-code`) オプションがあらかじめ定義されたフォーマットのセットを指定できます。

指定する値	説明	区切り文字	引用符
standard	標準的なCSVフォーマットです。(デフォルト値)	, (半角カンマ)	" (半角ダブルクォート)
excel	Excelで使用されるCSVフォーマットです。	, (半角カンマ)	" (半角ダブルクォート)
excel-north-europe	北ヨーロッパのExcelで使用されるCSVフォーマットです。	; (半角セミコロン)	" (半角ダブルクォート)

更新モード

キー名 `update-mode`

インポートするデータの更新モードを指定します。

全件のデータに影響します。1件ごとの指定はできません。

指定する値	説明
merge	既存のデータにマージして更新します。（デフォルト値）
replace	既存のデータを置き換えて更新します。

null代替文字列

キー名 null-string

インポートする項目のうち、`null` として扱う代替文字列を指定します。

CSVエクスポートオプション

この項では、CSVファイルで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。

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	2013 Summer
ファイルパス	file	文字列	(なし)	2013 Summer
読み込み件数	fetch-count	数値	5000	2013 Summer
日付/時刻フォーマットパターン	date-time-format-pattern	文字列	yyyy-MM-dd HH:mm:ss.SSS	2013 Summer
日付/時刻フォーマットロケールID	date-time-locale-id	文字列	en	2013 Summer
日付フォーマットパターン	date-format-pattern	文字列	yyyy-MM-dd	2013 Summer
日付フォーマットロケールID	date-locale-id	文字列	en	2013 Summer
UTF BOM	with-utf-bom	真偽値	false	2013 Summer
ヘッダ行有無フラグ	with-header	真偽値	false	2013 Summer
区切り文字	delimiter-code	文字列	, (半角カンマ)	2013 Summer
引用符	quote-code	文字列	" (半角ダブルクォー ト)	2013 Summer

名前	キー名	型	デフォルト値	導入バージョン
改行文字	newline-code	文字列	\r\n (CRLF)	2013 Summer
CSVフォーマットパターン	csv-format-pattern	文字列	standard	2013 Summer
null代替文字列	null-string	文字列	(空文字列)	2013 Summer

エンコーディング

キー名 `encoding`

エクスポートするCSV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CSV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。

読み込み件数

キー名 `fetch-count`

データベースからエクスポート対象のデータを一度に読み込む件数を指定します。

日付/時刻フォーマットパターン

キー名 `date-time-format-pattern`

エクスポートするCSVファイル内の日時項目のフォーマットを指定します。

「[アカウント情報](#)」の「アカウントロック日付」で利用します。

フォーマットに使用可能なパターン文字については「[日付/時刻フォーマットパターンに指定可能な文字](#)」を参照してください。

日付/時刻フォーマットロケールID

キー名 `date-time-locale-id`

エクスポートするCSVファイル内の日時項目のロケールIDを指定します。

[日付/時刻フォーマットパターン](#) と一緒に指定します。

「[アカウント情報](#)」の「アカウントロック日付」で利用します。

日付フォーマットパターン

キー名 date-format-pattern

エクスポートするXMLファイル内の日付項目のフォーマットを指定します。

「[アカウント情報](#)」の「有効開始日」「有効終了日」および「[アカウント付与ルール情報](#)」の「有効開始日」「有効終了日」で利用します。

フォーマットに使用可能なパターン文字については「[日付/時刻フォーマットパターンに指定可能な文字](#)」を参照してください。

日付フォーマットロケールID

キー名 date-locale-id

エクスポートするXMLファイル内の日付項目のロケールIDを指定します。

[日付フォーマットパターン](#)と一緒に指定します。

「[アカウント情報](#)」の「有効開始日」「有効終了日」および「[アカウント付与ルール情報](#)」の「有効開始日」「有効終了日」で利用します。

UTF BOM

キー名 with-utf-bom

エクスポートするCSVファイルのエンコーディングが UTF-8、UTF-16、UTF-32 のいずれかの場合にBOMを含ませるかどうかを指定します。

指定する値	説明
true	BOMを含めます。
false	BOMを含めません。（デフォルト値）

ヘッダ行有無フラグ

キー名 with-header

エクスポートするCSVファイルにヘッダ行を含ませるかどうかを指定します。

指定する値	説明
true	ヘッダ行（最初の1件）を出力します。
false	ヘッダ行を出力しません。（デフォルト値）

区切り文字

キー名 delimiter-code

エクスポートするCSVファイルの各項目を区切る文字列を指定します。

i コラム

このオプションには、以下の特殊文字を指定できます。

t	タブ文字
r	CR
n	LF
	半角バックスラッシュ シュ

! 注意

`csv-format-pattern` オプションを明示的に指定した場合は、このオプションは利用できません。

引用符

キー名 `quote-code`

エクスポートするCSVファイルの各項目を囲むための引用符の文字列を指定します。

! 注意

`csv-format-pattern` オプションを明示的に指定した場合は、このオプションは利用できません。

改行文字

キー名 `newline-code`

エクスポートするCSVファイルの改行文字列を指定します。

i コラム

このオプションには、以下の特殊文字を指定できます。

r	CR
n	LF

! 注意

`csv-format-pattern` オプションを明示的に指定した場合は、このオプションは利用できません。

CSVフォーマットパターン

キー名 `csv-format-pattern`

エクスポートするCSVファイルのフォーマット情報を指定します。

区切り文字 (`delimiter-code`)、引用符 (`quote-code`)、改行文字 (`newline-code`) オプションがあらかじめ定義されたフォーマットのセットを指定できます。

指定する値	説明	区切り文字	引用符	改行文字
standard	標準的なCSVフォーマットです。(デフォルト値)	, (半角カンマ)	" (半角ダブルクォート)	\r\n (CRLF)
excel	Excelで使用されるCSVフォーマットです。	, (半角カンマ)	" (半角ダブルクォート)	\n (LF)
excel-north-europe	北ヨーロッパのExcelで使用されるCSVフォーマットです。	; (半角セミicolon)	" (半角ダブルクォート)	\n (LF)

null代替文字列

キー名 null-string

エクスポートする項目のうち、**null** として扱う代替文字列を指定します。

項目

- 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

この章では、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 アカウントのインポート・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アカウントのインポート・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

- アカウントインポート（XML形式）

ジョブカテゴリ テナントマスタ > インポート

ジョブID account-import

ジョブ名 アカウントインポート

- アカウントインポート（CSV形式）

ジョブカテゴリ テナントマスタ > インポート(CSV)

ジョブID account-import-csv

ジョブ名 アカウントインポート(CSV)

- アカウントエクスポート（XML形式）

ジョブカテゴリ テナントマスタ > エクスポート

ジョブID account-export

ジョブ名 アカウントエクスポート

- アカウントエクスポート（CSV形式）

ジョブカテゴリ テナントマスタ > エクスポート(CSV)

ジョブID account-export-csv

ジョブ名 アカウントエクスポート(CSV)

ジョブネット

- アカウントインポート (XML形式)

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 account-import-jobnet

ジョブネット名 アカウントインポート

- アカウントインポート (CSV形式)

ジョブネットカテゴリ テナントマスタ > インポート(CSV)

ジョブネットID account-import-csv-jobnet

ジョブネット名 アカウントインポート(CSV)

- アカウントエクスポート (XML形式)

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 account-export-jobnet

ジョブネット名 アカウントエクスポート

- アカウントエクスポート (CSV形式)

ジョブネットカテゴリ テナントマスタ > エクスポート(CSV)

ジョブネットID account-export-csv-jobnet

ジョブネット名 アカウントエクスポート(CSV)

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。

- XML形式のインポートの場合

`jp.co.intra_mart.import.StandardAccountXmlImporter`

- CSV形式のインポートの場合

`jp.co.intra_mart.import.StandardAccountCsvImporter`

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。利用可能なオプションはXML形式とCSV形式で異なります。

詳細は、「[XMLインポートオプション](#)」と「[CSV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リスト](#)」を参照してください。

以下はXML形式で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;

/**
 * アカウントのインポートを行うクラスです。
 */
public class AccountImporter {

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

    /**
     * アカウント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account.xmlを選択
        final PublicStorage storage = new PublicStorage("account.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。

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。

- XML形式のエクスポートの場合

`jp.co.intra_mart.export.StandardAccountXmlExporter`

- CSV形式のエクスポートの場合

`jp.co.intra_mart.export.StandardAccountCsvExporter`

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。利用可能なオプションはXML形式とCSV形式で異なります。

詳細は、「[XMLエクスポートオプション](#)」と「[CSV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リスト](#)」を参照してください。

以下はXML形式で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;

/**
 * アカウントのエクスポートを行うクラスです。
 */
public class AccountExporter {

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

    /**
     * アカウント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account.xmlを選択
        final PublicStorage storage = new PublicStorage("account.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。

- XML形式のインポートの場合

`jp.co.intra_mart.import.StandardAccountXmlImporter`

- CSV形式のインポートの場合

`jp.co.intra_mart.import.StandardAccountCsvImporter`

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。利用可能なオプションはXML形式とCSV形式で異なります。

詳細は、「[XMLインポートオプション](#)」と「[CSV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リスト](#)」を参照してください。

以下はXML形式で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.StandardAccountXmlImporter';

/**
 * アカウント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account.xmlを選択
    var storage = new PublicStorage('account.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。

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。

- XML形式のエクスポートの場合

```
jp.co.intra_mart.export.StandardAccountXmlExporter
```

- CSV形式のエクスポートの場合

```
jp.co.intra_mart.export.StandardAccountCsvExporter
```

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。利用可能なオプションはXML形式とCSV形式で異なります。

詳細は、「[XMLエクスポートオプション](#)」と「[CSV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リスト](#)」を参照してください。

以下はXML形式で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.StandardAccountXmlExporter';

/**
 * アカウント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account.xmlを選択
  var storage = new PublicStorage('account.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

- [アカウントインポートデータ定義書](#)
- [日付/時刻フォーマットパターンに指定可能な文字](#)
- [トラブルシューティング](#)
 - [旧バージョンでエクスポートしたXMLファイルのインポートに失敗する](#)

アカウントインポートデータ定義書

アカウントインポート・エクスポートで利用するデータの詳細はアカウントインポートデータ定義書として提供します。

アカウント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_admin_account_import_export_definition.xls」](#)

日付/時刻フォーマットパターンに指定可能な文字

フォーマットに使用するパターン文字は Java の `SimpleDateFormat` クラス に準拠します。
代表的なパターン文字は以下の通りです。

パターン文字	説明
y	年
M	月
d	月における日
E	曜日
H	時 (24時間表記、0~23)
K	時 (12時間表記、0~11)
h	時 (12時間表記、1~12)
m	分
s	秒
S	ミリ秒
Z	タイムゾーン (RFC822)

トラブルシューティング

旧バージョンでエクスポートしたXMLファイルのインポートに失敗する

以下の原因が考えられます。

1. エクスポートしたXMLファイルに空のタグが含まれている

2. インポートデータで利用する日時データのフォーマットが古い

エクスポートしたXMLファイルに空のタグが含まれている

intra-mart Accel Platform 2013 Winter(Felicia) より、インポート実行時にインポートデータの検証が行われるようになりました。

その検証の中で空のタグが検出されるとインポート処理に失敗する場合があります。

例えば、ロケールID の場合「`<locale-id></locale-id>`」といった空のタグがインポートデータに含まれると以下のエラーが発生します。

- [E.IWP.IMPORTER.ACCOUNT.00001] ロケールマスタに定義されていないロケールIDが指定されています。

回避方法は、以下のいずれかです。

1. タグの値を適切に設定し、再度インポートを行う。
2. 該当のタグを削除し、再度インポートを行う。（インポート対象から除外したい項目である場合）
3. インポート時のデータ検証フラグを「`false`」に設定し、再度インポートを行う。
（インポートデータのデータ検証の実行の有無は「[オプション](#)」 - 「[XMLインポートオプション](#)」 - 「[データ検証フラグ](#)」にて変更可能です。）



コラム

インポートデータの検証の詳細な仕様は「[インポート](#)」 - 「[インポート時のデータ検証について](#)」を参照してください。

インポートデータで利用する日時データのフォーマットが古い

intra-mart Accel Platform 2013 Summer(Damask) より、インポートデータで利用する日時データのフォーマットを明示的に指定するようになりました。

指定されたフォーマットに適さない日時データが検出された場合、インポート処理に失敗します。

例えば、有効開始日が「`<valid-start-date>Jan 1, 1900 12:00:00 AM</valid-start-date>`」の場合、日付フォーマットパターンのデフォルト値「`yyyy-MM-dd`」に適していないため以下のエラーが発生します。

- [E.IWP.IMPORTER.ACCOUNT.00004] 有効開始日の形式が不正です。

回避方法は、以下のいずれかです。

1. 日時データのフォーマットを適切に設定し、再度インポートを行う。
2. インポート時の日付/時刻フォーマットパターン、または、日付フォーマットパターンをインポートデータ内で利用しているフォーマットに設定し、再度インポートを行う。
（インポートデータで利用する日時データのフォーマットは「[オプション](#)」 - 「[XMLインポートオプション](#)」 - 「[日付/時刻フォーマットパターン](#)」、または、「[日付フォーマットパターン](#)」にて変更可能です。）



コラム

日付/時刻フォーマットパターンに指定可能な文字の詳細は「[日付/時刻フォーマットパターンに指定可能な文字](#)」を参照してください。

