intra-mart WebPlatform/AppFramework Ver.7.0

BPM 説明資料

2008/08/22 初版

<< 変更履歴 >>

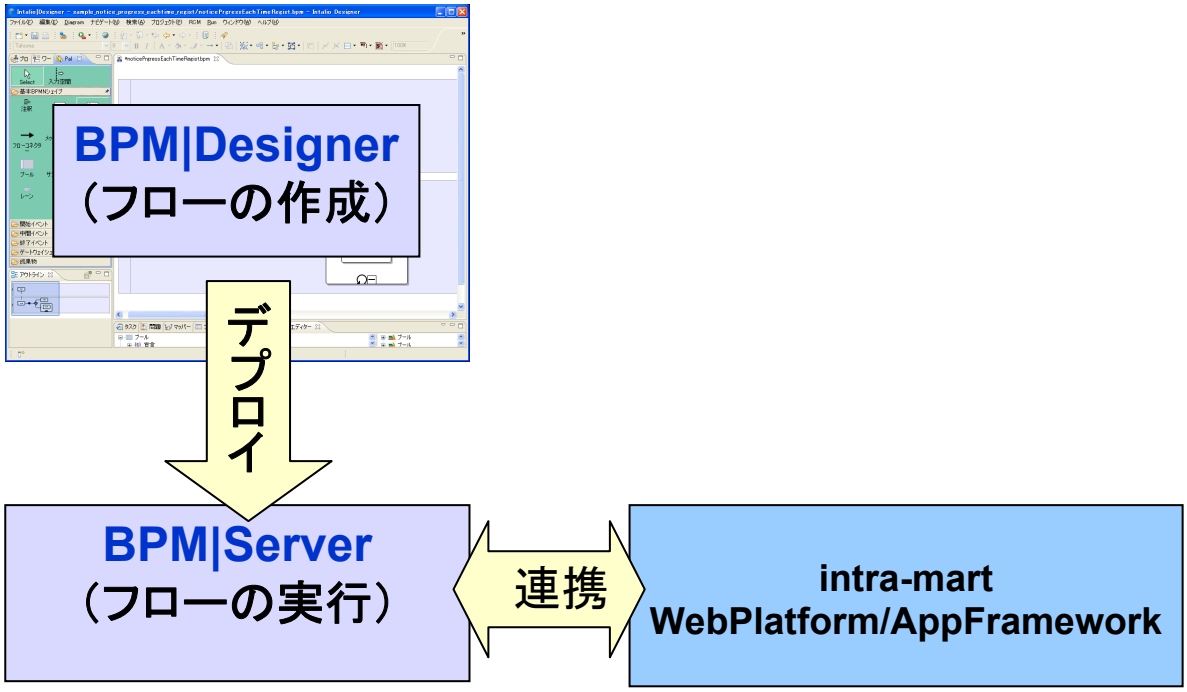
変更年月日	変更内容
2008/08/22	初版

<< 目次 >>

1 はじめに 1
2 Secutiry Provider 2
2.1 概要 2
2.2 IM-BPMとintra-mart間の権限連携 2
2.2.1 概要 2
2.2.2 権限の種類 3
2.2.3 権限IDフォーマット 6
2.2.4 1 ユーザ当りのBPM|Server上でのロール総数 6
2.2.5 期間化情報 7
2.3 BPM管理画面のアクセス制御 7
2.3.1 権限の概要 8
2.3.2 設定 10
2.3.3 注意点 11
3 TokenHandler 12
3.1 概要 12
3.2 トークン 12
3.2.1 トークンに含まれる情報 12
3.2.2 トークンAPI (intra-mart) 13
3.2.3 トークンAPI(BPM|Server) 13
3.2.4 トークンの利用箇所 13
3.3 秘密鍵 14
3.3.1 秘密鍵の設定(intra-mart) 14
3.3.2 秘密鍵の設定(BPM|Server) 14
3.4 TokenHandlerの変更 15
3.4.1 変更に必要な要件 15
3.4.2 intra-mart側のTokenHandlerを変更する場合 15
3.4.3 BPM|Server側のTokenHandlerを変更する場合 16

1 はじめに

intra-mart WebPlatform/AppFramework ver7.0 エンタープライズ for BPM では、次のような仕組みでビジネスフローが実行されます。



BPM|Designer で作成されたビジネスフローは、BPM|Server へデプロイすることで実行可能な状態になります。
intra-mart WebPlatform/AppFramework(以下、「intra-mart」と表記します)では、ヒューマンタスクを処理するための画面や WebService 等を実装することが可能で、BPM|Server と連携して処理を実行します。

本書では、BPM|Server と intra-mart が連携する際の仕様について記述します。
BPM|Designer や BPM|Server のインストール方法や、フローの作成方法については、別冊のセットアップガイドや操作ガイドを参照してください。

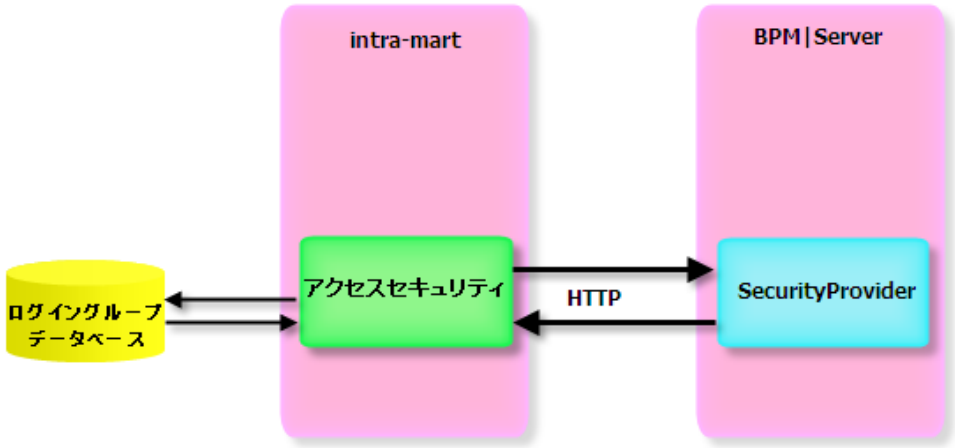
1.1 用語解説

intra-mart Ver7.0 BPM|Server	以下、**BPM|Server** と略します。 **BPM|Server** をインストールしたディレクトリを**% BPM|Server %**と略します。
intra-mart WebPlatform Ver7.0 または intra-mart AppFramework Ver7.0	以下、**iWP/iAF** と略します。 **iWP/iAF** をインストールしたディレクトリを**%IM_HOME%**と略します。

2 Secutiry Provider

2.1 概要

SecurityProvider は、BPM|Server と intra-mart 間のログイン認証・アクセス権限の連携を行う為のモジュールです。
BPM|Server 上で、intra-mart 上のログイン認証・アクセス権限取得のサービスを実行することで、intra-mart システムとの連携を行っています。
これにより intra-mart にログインすることで、BPM|Server に対しシングルサインオンと一貫したアクセス制御が可能となっています。



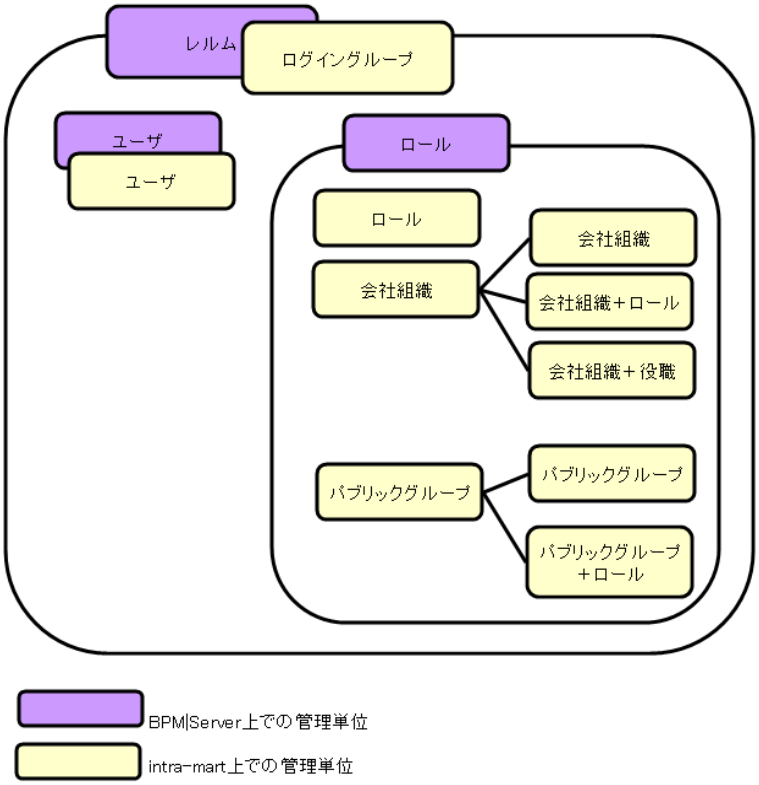
2.2 IM-BPMとintra-mart間の権限連携

IM-BPM では、ビジネスプロセスのヒューマンタスク(またヒューマンタスクを配置したプール)には処理権限の設定を、BPM 管理画面の一部の機能に対してはアクセス権限の設定を行うことが出来ます。
ここでは IM-BPM と intra-mart 間の権限の連携について説明します。

2.2.1 概要

IM-BPM では、権限を「レルム(認証領域)」・「ユーザ」・「ロール」 単位で管理を行っています。
レルム(認証領域)は intra-mart のログイングループにあたります。
IM-BPM での「ロール」権限は、intra-mart の「ロール」権限に「会社組織(役職)」・「パブリックグループ」権限を包括する構造となっています。
BPM|Designer で処理権限を設定する場合は、「ロール」権限・「会社組織(役職)」・「パブリックグループ」権限がそれぞれ独立して設定可能になっているため、通常はこれらの構造を意識する必要はありません。

Copyright 2000-2008 株式会社 NTT データ イントラマート All rights Reserved.



2.2.2 権限の種類

2.2.2.1 レルム（認証領域）

IM-BPM 上でのレルム（認証領域）は、intra-mart 上のログイングループを表します。

BPM|Server でのログイングループの取得は、サーバ起動時に、以下のように行われます。

- intra-mart の Application runtime が起動している場合
  intra-mart サーバ上のログイングループ取得サービスを呼び出し、サービスよりログイングループを取得します。
  取得した情報はファイルにキャッシュします。
  キャッシュファイルは、% **BPM|Server %** /conf ディレクトリに「im-group.conf」という名前で作成されます。

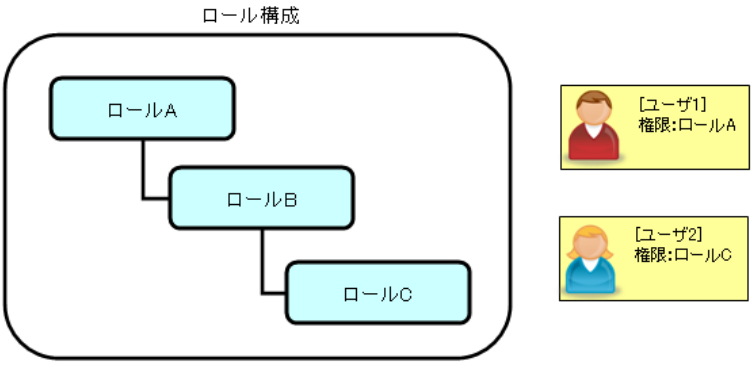
- intra-mart の Application runtime が起動していない場合
  上記キャッシュファイルが存在する場合は、ファイルからログイングループを取得します。
  キャッシュファイルが存在していない場合（初回起動時）は、エラーとなりエラーログが出力されます。

2.2.2.2 ユーザ

IM-BPM 上でのユーザ権限は、intra-mart 上のユーザ権限を表します。

2.2.2.3 ロール

- ロール
  intra-mart 上のロール権限を表します。
  設定したロールを保持するユーザが処理可能となります。
  また、ロールがネスト構造を持っている場合、設定したロールの上位ロールを保持するユーザも処理可能です。絞込み条件としてロールを設定した場合も同様になります。



設定ロールと処理権限の有無

設定ロール	ロール A	ロール B	ロール C
ユーザ1	○	○	○
ユーザ2	×	×	○

※　ユーザ 1 はロール B、ロール C の上位ロールを持っている為、
A,B, C 全てのロールが設定されている処理が可能

■　会社・組織

intra-mart 上の会社・組織権限を表します。
設定した会社・組織に所属するユーザが処理可能となります。
設定した会社・組織の上位組織に所属するユーザには権限はありません。

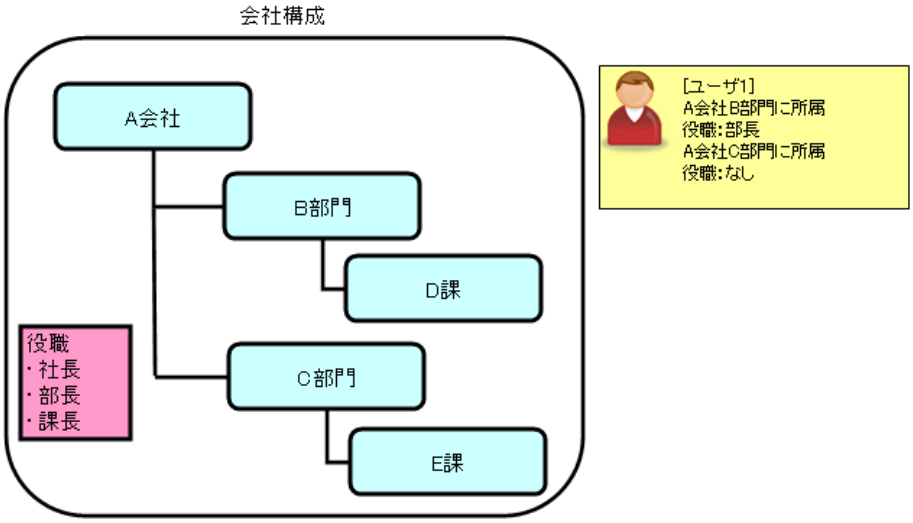
会社・組織にはロールによる絞込みと役職による絞込み設定が可能です。

◆　ロールによる絞込み

会社・組織にロールによる絞込み設定を行った場合、その会社・組織に所属していて、かつ絞込み設定を行ったロールを保持するユーザが処理可能となります。

◆　役職による絞込み

役職による絞込み設定を行った場合、その会社・組織に所属していて、かつ絞込み設定を行った役職を付与されたユーザが処理可能となります。
但し、役職が有効となるのはユーザに役職を付けた会社・組織のみです。



Copyright 2000-2008 株式会社 NTT データ イントラマート All rights Reserved.

会社・組織の設定権限と処理権限有無

設定権限	A 会社	A 会社 B 部門	A 会社 B 部門 役職:部長	A 会社 C 部門	A 会社 C 部門 役職:部長	A 会社 B 部門 D 課	A 会社 C 部門 E 課
ユーザ1	×	○	○	○	×	×	×

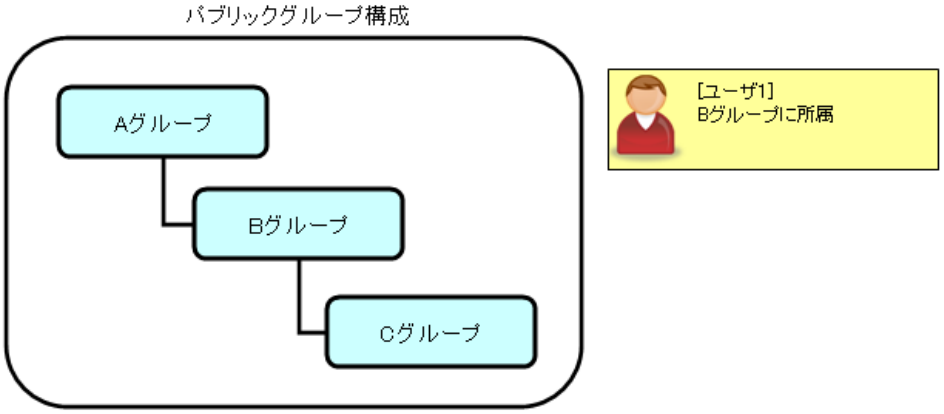
■ パブリックグループ

intra-mart 上のパブリックグループ権限を表します。
設定したパブリックグループに所属しているユーザが処理可能となります。
設定したパブリックグループの上位グループに所属するユーザには権限はありません。

パブリックグループにはロールによる絞込み設定が可能です。

◆ ロールによる絞込み

パブリックグループにロールによる絞込み設定を行った場合、そのパブリックグループに所属していて、かつ絞込み設定を行ったロールを保持するユーザが処理可能となります。



パブリックグループの設定権限と処理権限有無

設定権限	A グループ	B グループ	C グループ
ユーザ1	×	○	×

2.2.3 権限IDフォーマット

IM-BPM では、各種 ID をログイングループ情報を含んだ形式で保持します。
またロールIDは、ロールの種別を表すプレフィックスと複合キー（会社コード・組織コードなど）を区切る為のセパレータ($)をつけることで intra-mart 上での各権限を表現しています。

各種権限 ID のフォーマットは以下のようになっています。

※ログイングループが default の場合

Prefix	種別	intra-mart 上の値	IM|BPM 上での値
なし	ユーザ	aoyagi	default¥aoyagi
0	ロール	level2	default¥0level2
1	組織	comp_sample_01 dept_sample_20	default¥1comp_sample_01$dept_sample_20
	組織 ＋ 役職で絞込み	comp_sample_01 dept_sample_20 ps001	default¥1comp_sample_01$dept_sample_20$2ps001
	組織 ＋ ロールで絞込み	comp_sample_01 dept_sample_20 level3	default¥1comp_sample_01$dept_sample_20$0level3
2	役職	ps001	default¥会社コード$組織コード$2ps001 ※会社組織に対する絞込み条件としてのみ設定
3	パグリックグループ	sample_public public_team_a	default¥3sample_public$public_team_a
	パグリックグループ+ロールで絞込み	sample_public public_team_a level2	default¥3sample_public$public_team_a$0level2

2.2.4 1 ユーザ当りのBPM|Server上でのロール総数

1 ユーザ当りの BPM|Server 上でのロール総数は、以下の計算式より算出できます。

※ロール数にはサブロールも含みます。

保持ロール総数 ＝ ロール数 ＋ 所属組織数 + 所属パブリックグループ数 ＋
(所属組織数 ×ロール数) + 役職数 ＋
(所属パグリックグループ数 ×ロール数)

大量のロールを保持、または大量の会社組織・パブリックグループに所属する場合、BPM|Server 上では膨大なロール ID を保持することとなる為、パフォーマンスの劣化を招く可能性があります。

Copyright 2000-2008 株式会社 NTT データ イントラマート All rights Reserved.

2.2.5 期間化情報

intra-mart から権限に関する期間化を伴った情報を取得する際の検索基準日は、デフォルトではシステム日付となります。
intra-mart 構成ファイル「**% BPM|Server %** /conf/im-config.properties」 にて、検索基準日の指定が可能です。
設定については、「BPM|Server 設定ガイド 3.2.2 im-config.properties」を参照して下さい。

- intra-mart 構成ファイルに基準日の設定がある場合
  設定した基準日が検索基準日となります。

- intra-mart 構成ファイルに基準日の設定がない場合
  システム日付が検索基準日となります。

(※)検索基準日の設定が有効となるのは、BPM|Server から intra-mart への認証処理が行われる BPM|管理画面のみとなりますので、ご注意ください。

SecurityProvider では、各種権限に対し以下の期間化情報を参照します。

種別	期間化情報
ユーザ	アカウントの有効期間 プロファイルの有効期間
ロール	アカウントの有効期間 アカウントロールの有効期間
組織	プロファイルの期間 組織の期間 所属組織の期間
パブリックグループ	プロファイルの期間 パブリックグループの期間 所属パブリックグループの期間

期間化についての詳細は、「アドミニストレータ・ガイド」「アプリケーション共通マスタ説明資料」を参照してください。

2.3 BPM管理画面のアクセス制御

BPM|Server では BPM 管理画面の各機能に対しロールベースでアクセス制御を行っています。

対象機能は以下になります。

- ログイン権限
- プロセス有効化権限
- プロセス無効化権限
- デプロイ権限
- アンデプロイ権限
- プロセス詳細有効化権限
- プロセス詳細無効化権限
- インスタンス削除権限
- インスタンス全削除権限

2.3.1 権限の概要

ここでは、権限設定が可能な各機能について説明します。

2.3.1.1 ログイン権限

BPM|Server　BPM 管理画面へのログイン権限です。

intra-mart　メニュー[BPM][管理]メニューに設定した権限と同期が取れている必要があります。

デフォルトでは，bpmadmin(BPM 管理者権限)、bpmprocesspanager（BPM プロセス管理者権限）が設定されています。

2.3.1.2 プロセス有効化/無効化権限

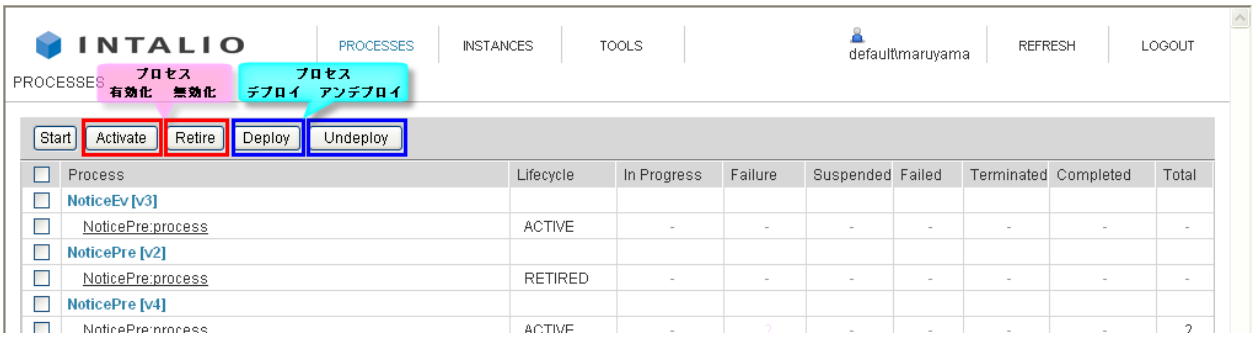
プロセスのライフサイクルを「Active」（有効）・「Retired」（無効）に変更できる権限です。

ログインユーザに処理権限がある場合は、プロセス一覧画面のツールバーに「Activate」「Retire」ボタンが表示されます。

2.3.1.3 デプロイ/アンデプロイ権限

プロセスをデプロイまたはアンデプロイできる権限です。

ログインユーザに処理権限がある場合は、プロセス一覧画面のツールバーに「Deploy」「Undeploy」ボタンが表示されます。

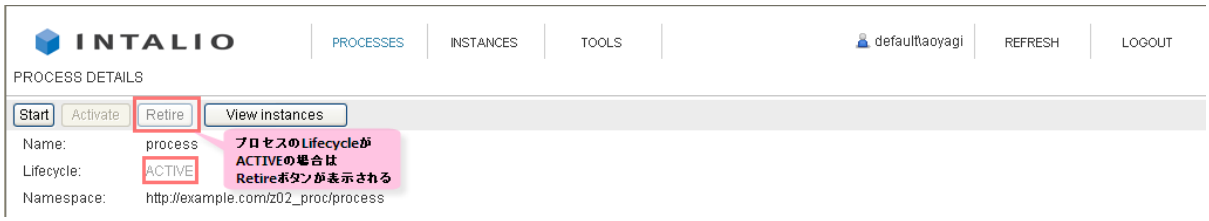


<BPM 管理者画面　プロセス一覧>

2.3.1.4 プロセス詳細有効化/無効化権限

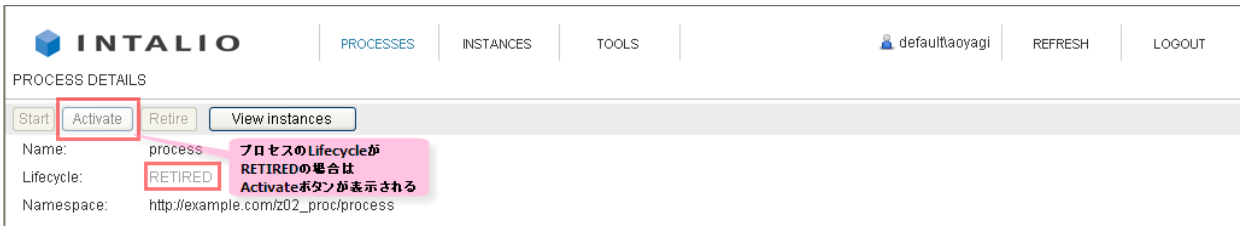
プロセス詳細画面で、プロセスのライフサイクルを「Active」（有効）・「Retired」（無効）に変更できる権限です。

ログインユーザに処理権限がある場合は、プロセス詳細画面のツールバーに「Activate」「Retire」ボタンが表示されます。



Copyright 2000-2008 株式会社 NTT データ イントラマート All rights Reserved.

<BPM 管理者画面 プロセス詳細①>



<BPM 管理者画面 プロセス詳細②>

2.3.1.5 インスタンス削除権限

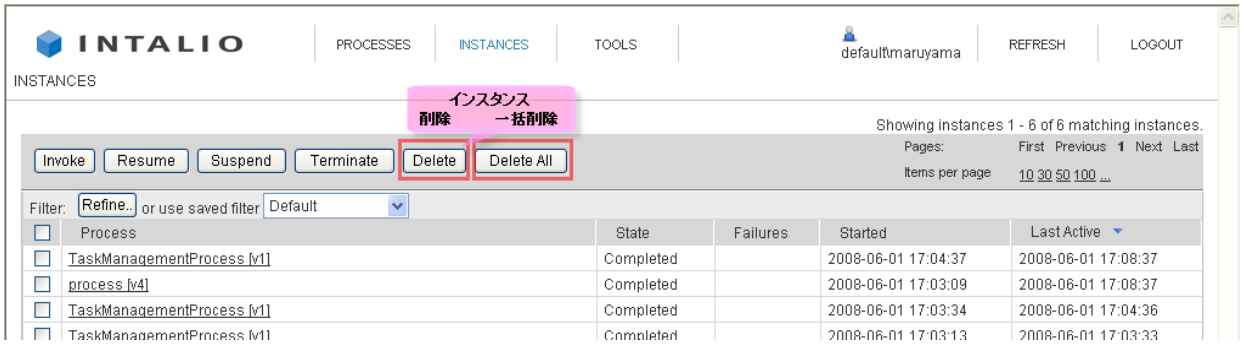
インスタンスを削除できる権限です。

ログインユーザに処理権限がある場合は、インスタンス一覧画面のツールバーに「Delete」ボタンが表示されます。

2.3.1.6 インスタンス一括削除権限

インスタンスを一括削除できる権限です。

ログインユーザに処理権限がある場合は、インスタンス一覧画面のツールバーに「Delete All」ボタンが表示されます。



<BPM 管理者画面 インスタンス一覧>

2.3.2 設定

権限設定は、bean 定義ファイル「**% BPM|Server %** /conf/console/ console-springapp-servlet.xml」の bean id が**"imConsoleRBACQuery"** タグ要素内で行います。
各処理権限は property 要素・name 属性に設定された処理 ID に該当する list 要素内に、権限を付与するロール ID を value 要素内に任意のロール数分記述します。

```
<property name="処理 ID">
  <list>
    <value>ロール ID_1</value>
    <value>ロール ID_2</value>
    <value>ロール ID_3</value>
  </list>
</ property >
```

以下に各処理に該当する処理 ID とデフォルト設定値を示します。

処理名	処理 ID	デフォルト設定値
ログイン権限	loginRoles	bpmadmin (BPM 管理者権限) bpmprocessmanager （BPM プロセス管理者権限）
プロセス有効化権限	activateActionRoles	bpmadmin
プロセス無効化権限	retireActionRoles	bpmadmin
デプロイ権限	deployActionRoles	bpmadmin
アンデプロイ権限	undeployActionRoles	bpmadmin
プロセス詳細有効化権限	pdActivateActionRoles	bpmadmin
プロセス詳細無効化権限	pdRetireActionRoles	bpmadmin
インスタンス削除権限	instancesDeleteActionRoles	bpmadmin
インスタンス全削除権限	instancesDeleteAllActionRoles	bpmadmin

■ ログイン権限の変更例
ログイン権限を「bpmadmin」「bpmprocessmanager」から「sampleloginrole」に変更

【変更前】

```
<!-- this bean is a ImConsoleRBACQuery  -->
<bean id="imConsoleRBACQuery"
  class="jp.co.intra_mart.bpm.server.security_provider.console.ImConsoleRBACQueryFileImpl">
  <!-- Login section granted role -->
  <property name="loginRoles">          ・・・・ログイン権限処理 ID
    <list>
      <value>bpmadmin</value>
      <value>bpmprocessmanager</value>
    </list>
  </property>
～  以下  省略
```


Copyright 2000-2008 株式会社 NTT データ イントラマート All rights Reserved.

【変更後】

```
<!-- this bean is a ImConsoleRBACQuery  -->
<bean id="imConsoleRBACQuery"
  class="jp.co.intra_mart.bpm.server.security_provider.console.ImConsoleRBACQueryFileImpl">
  <!-- Login section granted role -->
  <property name="loginRoles">          ・・・・ログイン権限処理 ID
    <list>
      <value> sampleloginrole </value>
    </list>
  </property>
～  以下  省略
```

2.3.3 注意点

- 設定ファイルを編集した場合
  変更内容を反映させるには、BPM|Server の再起動が必要です。

- intra-mart でログイングループ情報を変更した場合
  変更内容を BPM|Server に反映させるには、BPM|Server の再起動が必要です。

3 TokenHandler

3.1 概要

Security Provider により管理される権限情報は、TokenHandler を通じてトークンとして利用されます。
トークンは intra-mart WebPlatform/AppFramework と BPM|Server 間において、シングルサインオン、プロセス一覧の取得、プロセスの操作等を行う際に利用されます。
本章では TokenHandler によるトークン管理に関して説明します。

3.2 トークン

トークンは BPM|Server に対してユーザ認証が行われた事を証明するキーです。
BPM|Server が提供する xforms, フォーム・リレーション等を利用し、ビジネス・プロセスを実行する際に必要となります。

3.2.1 トークンに含まれる情報

トークンには、ユーザを一意に特定する為の情報が格納されています。
また、TokenHandler では、トークンの正当性を保障する為、トークン内部にハッシュ値を含めた形で暗号化を行っています。
トークンに含まれる情報を下記に示します。

キー	説明
issued	発行日 ミリ秒で測定した、現在時刻と協定世界時の UTC1970 年 1 月 1 日午前 0 時との差
user	BPM|Server 上で利用されるユーザ ID グループ ID + "¥" + ユーザ ID
roles	BPM|Server 上で利用されるロール(","区切り) intra-martとBPM|Server 上のロール表現に関しては、前章[権限 ID フォーマット]を参照してください。
userid	intra-mart 上で利用されるユーザ ID
fullName	intra-mart 上で利用されるユーザ名 ユーザのデフォルトロケールで取得されるユーザ名
timestamp	トークン生成時間 ミリ秒で測定した、現在時刻と協定世界時の UTC1970 年 1 月 1 日午前 0 時との差
nonce	トークン生成時間を元に作成されるランダムな値
digest	権限情報の値を、timestamp, nonce, 秘密鍵を通じてハッシュ化した値

- 秘密鍵に関しては後述「秘密鍵」をご参照下さい。

Copyright 2000-2008 株式会社 NTT データ イントラマート All rights Reserved.

3.2.2 トークンAPI (intra-mart)

intra-mart では、トークンの生成を行う為の API が提供されます。

- JavaEE 開発モデル

  jp.co.intra_mart.foundation.bpms.token.TokenManager

```
TokenManager tokenManager = new TokenManager("default");
String token = tokenManager.createToken("aoyagi");
```

- スクリプト開発モデル

  TokenManager

```
var tokenManager = new TokenManager("default");
var token = tokenManager.createToken("aoyagi");
```

3.2.3 トークンAPI(BPM|Server)

BPM|Server 上では、トークンの解析を行う為の API は WebService 形式で提供されます。

- トークン生成用 WebService

  http://%BPM サーバ%:8080/axis2/services/TokenService?wsdl

3.2.4 トークンの利用箇所

トークンは、下記の画面、処理において利用されます。

- プロセス、未処理、処理済、通知タスクの取得
- プロセスの開始
- 未処理タスクの保留
- 未処理タスクの保留取消
- 未処理タスクの一時保存
- 通知一覧の確認

上記処理を行う場合、TaskManager API を通して処理が行われます。
TaskManager API では、内部処理で TokenManager を利用しトークンの作成を行う為、通常トークンの発行を行う必要はありません。

3.3 秘密鍵

トークンの生成に利用する秘密鍵の設定です。
トークンに含まれるダイジェストは、そのトークンの正当性を示す為の不可逆なハッシュ値となります。
ダイジェストは、権限情報、トークンの生成時間、トークンの生成時間を基にした乱数値、そして秘密鍵を利用し作成されます。
秘密鍵は、intra-mart、及び BPM|Server 共に設定を行う必要があります。
また、それぞれの秘密鍵は同一の値が設定されている必要があります。

3.3.1 秘密鍵の設定(intra-mart)

intra-mart 上の TokenHandler で利用される秘密鍵は下記のファイルに設定されています。
%IM_HOME% /conf/bpms-config.xml
デフォルトでは秘密鍵は"mySecret"と設定されています。
秘密鍵の変更を行う場合は、秘密鍵として利用される文字列を置換して下さい。

```
<token-manager>
<service-factory-class>jp.co.intra_mart.foundation.bpms.token.TokenServiceFactoryLocalImpl</service-factory-class>
  <init-param>
    <param-name>secret</param-name>
    <param-value>mySecret</param-value>
  </init-param>
</token-manager>
```

3.3.1.1 注意事項

- bpms-config/token-manager/service-factory-class に、jp.co.intra_mart.foundation.bpms.token.TokenServiceFactoryLocalImpl クラスが指定されている場合は、intra-mart 内部でトークンの作成が行われます。それに対し、jp.co.intra_mart.foundation.bpms.token.TokenServiceFactoryWSImpl クラスが指定されている場合は、トークンの生成は BPM|Server 側で行われる為秘密鍵の指定は必要ありません。
- 秘密鍵の設定を変更した場合、intra-mart の再起動が必要となります。

3.3.2 秘密鍵の設定(BPM|Server)

BPM|Server 上の TokenHandler で利用される秘密鍵は下記のファイルに設定されています。
% BPM|Server % /conf/securityConfig.xml
デフォルトでは秘密鍵は"mySecret"と設定されています。
秘密鍵の変更を行う場合は、秘密鍵として利用される文字列を置換して下さい。

```
<bean id="tokenHandler" class="jp.co.intra_mart.bpm.server.security_provider.ImTokenHandler">
  <property name="secret"><value>mySecret</value></property>
</bean>
```

3.3.2.1 注意事項

- 秘密鍵の設定を変更した場合、BPM|Server の再起動が必要となります。

Copyright 2000-2008 株式会社 NTT データ イントラマート All rights Reserved.

3.4 TokenHandlerの変更

intra-mart の提供する TokenHandler クラスは、設定ファイルにより独自の TokenHandler に差し替える事が可能です。
本項では差し替えに必要な手順、用件を記します。

3.4.1 変更に必要な要件

TokenHandler を通して変換されるトークンは、権限情報の正当性を保証する仕組みである必要があります。
権限情報のトークン化、トークンの権限情報化が行えるよう可逆のトークンを生成する必要があります。
作成されるトークンは必ず一意である必要があります。

3.4.2 intra-mart側のTokenHandlerを変更する場合

パッケージは jp.co.intra_mart.foudnation.bpms.token になります。
TokenServiceFactory インタフェース、TokenService インタフェースを実装したクラスを作成してください。

- TokenServiceFactory インタフェース
  TokenService を生成する為のインタフェースです。
- TokenService インタフェース
  トークンの生成、解析を行う TokenService インタフェースです。
- TokenServiceFactoryLocalImpl クラス
  デフォルトで提供される TokenService 生成用 Factory クラスです。
- TokenServiceLocalImpl クラス
  デフォルトで提供されるトークンの生成、解析を行う TokenService クラスです。

3.4.2.1 bpms-config.xml

%**IM_HOME**% /conf/bpms-config.xml
token-manager/service-factory-class 値を作成した TokenServiceFactory 実装クラスに変更します。
token-manager/init-param 値には、TokenServiceFactory 初期化に必要な値を指定する事が可能です。
secret キー値に関しても、秘密鍵を使用しない実装であれば不要となります。

```
<token-manager>
  <service-factory-class>jp.co.intra_mart.foundation.bpms.token.TokenServiceFactoryLocalImpl</service-factory-class>
  <init-param>
    <param-name>secret</param-name>
    <param-value>mySecret</param-value>
  </init-param>
</token-manager>
```

3.4.3 BPM|Server側のTokenHandlerを変更する場合

パッケージは org.intalio.tempo.security.impl になります。

- TokenHandler クラス

  トークンの生成、解析を行う TokenHandler クラスです。

- jp.co.intra_mart.bpm.server.security_provider.ImTokenHandler クラス

  デフォルトで提供されるトークンの生成、解析を行う TokenHandler を継承したクラスです。

3.4.3.1 securityConfig.xml

% BPM|Server % /conf/securityConfig.xml

beans/bean(@id=tokenHandler)値を作成した TokenHandler 継承クラスに変更します。

beans/bean/property 値には、TokenHandler インスタンス生成時に必要な値を指定する事が可能です。

secret キー値に関しても、秘密鍵を使用しない実装であれば不要となります。

```
<bean id="tokenHandler" class="jp.co.intra_mart.bpm.server.security_provider.ImTokenHandler">
  <property name="secret"><value>mySecret</value></property>
</bean>
```


Copyright 2000-2008 株式会社 NTT データ イントラマート All rights Reserved.

intra-mart WebPlatform/AppFramework Ver.7.0
BPM 説明資料

2008/08/22 初版

Copyright 2000-2008 株式会社 NTT データ イントラマート
All rights Reserved.
TEL: 03-5549-2821
FAX: 03-5549-2816
E-MAIL: info@intra-mart.jp
URL: http://www.intra-mart.jp/