

ものづくり・IT 融合化推進技術の研究開発

# MZ Platform データ連携機能

= 導入手順書 =



独立行政法人  
産業技術総合研究所

## = 目次 =

<b>1. データ連携機能を使用するための初期設定</b> .....	<b>1</b>
1.1. 環境設定 (PLATFORM.INIの編集) .....	1
1.2. 接続先の設定 (HOSTSFILE.INIの編集) .....	3
1.3. アクセスポリシーの設定 (ACCESSPOLICY.INIの編集) .....	4
1.4. 各コンポーネントの設定 (属性の編集) .....	6
<b>2. コンポーネント連携機能</b> .....	<b>8</b>
2.1. リモートメソッド・リモートコンポーネント .....	8
2.2. リモートメソッドの設定 .....	8
2.3. リモートコンポーネントの設定 .....	12
<b>3. コンポーネント転送機能</b> .....	<b>15</b>
3.1. PULL型/PUSH型 .....	15
3.2. 様々なコンポーネント転送 .....	17
3.3. 各種転送とプロパティの設定 .....	21
3.4. 転送要求受信コンポーネントを指定した転送 .....	23

## 1. データ連携機能を使用するための初期設定

データ連携機能は、プラットフォーム間でコンポーネントを互いに転送する機能（以下では、コンポーネント転送機能と呼びます）と、相手のプラットフォームにおけるコンポーネントのメソッドを実行する機能（以下では、コンポーネント連携機能またはリモート接続機能と呼びます）との2つから構成されます。これらデータ連携機能を使用するためには、1)各プラットフォームにおける連携に関する基本情報の設定、2)連携先プラットフォームの事前登録、3)連携先プラットフォームに対してのアクセス情報の設定、4)コンポーネント単位でのアクセス情報の設定を行う必要があります。4)のコンポーネント単位でのアクセス権の設定についてはコンポーネントの属性情報を変更することにより行いますが、1)から3)については設定ファイルが用意されており、それらファイル内のプロパティを指定することにより各設定を行います。それぞれの設定方法を以下に示します。

### 1.1. 環境設定（Platform.iniの編集）

Platform.ini にデータ連携機能を使用するための基本情報を設定します。Platform.ini におけるデータ連携機能に関するプロパティの意味と設定は次の通りです。

データ連携機能に関するプロパティ

プロパティ名	意味・設定
UseDataCooperation	データ連携機能の使用・不使用を指定するプロパティです。 <b>true</b> （使用する）に変更します。
UseOnlyLightDataCooperation	旧バージョンのデータ連携機能の使用・不使用を指定するプロパティです。 <b>true</b> （使用しない）に設定します（旧バージョンを併用する場合、 <b>false</b> に設定します）
PlatformName	プラットフォーム名を設定するプロパティです。一意なプラットフォーム名を設定します。
LocalhostAddress	プラットフォームが起動しているPCのローカルホストアドレスを設定するプロパティです。
RMIPort	<b>RMI</b> レジストリが使用するポート番号を設定するプロパティです。この項目の記入は任意であり、未記入の場合はデフォルトのポート番号（1099）が自動的に設定されます。
AccessPolicyFile	アクセスポリシーを記述したファイルを指定するプロパティです。未記入の場合にはデフォルトのファイル名（AccessPolicy.ini）が設定されます（詳細については[1.3アクセスポリシーの設定（AccessPolicy.iniの編集）]をご覧ください）。

プラットフォーム名を“Platform01”、ローカルホストアドレスを“xxx.example.com”、アクセスポリシーファイル名を“AccessPolicy01.ini”とした場合の Platform.ini の記述例を以下に示します。

```
LogLevel=1
LookAndFeel=
RuntimeLocale=
ComponentListFile_ja=etc/PlatformComponents_ja.ini
ComponentListFile_en=etc/PlatformComponents_en.ini
ComponentListFile=etc/PlatformComponents_en.ini
ComponentInformationFolder=components
BinaryDataAutoSave=true
UseDataCooperation=true
UseOnlyLightDataCooperation=true
BrokerAddress=
PlatformName=Platform01
localhostAddress=xxx.example.comy ←Windows のコンピュータ名による設定も可能
java.rmi.server.codebase=
java.security.policy=etc/java.policy
brokermonitor.broker_access_log=
brokermonitor.broker_debug_log=
RMIPort= ←ポート番号が未記入=デフォルトポート(1099)
CombinativeComponentsFolder=AP_DATA_COMB
AccessPolicyFile=etc/AccessPolicy01.ini
```

※各プロパティについての詳細は、「詳細設定説明書」に記されておりますのでそちらをご覧ください。

## 1.2. 接続先の設定 (hostsfile.iniの編集)

hostsfile.ini に、データ連携機能を利用するプラットフォーム群の情報を記入します。このファイルは、接続先となる外部プラットフォームのプラットフォーム名とホスト名、ポート番号を前もって関連付けておくための設定ファイルです。ここに記入されるホスト名とは、各プラットフォームの Platform.ini において LocalhostAddress プロパティに設定されている値のことであり、完全に一致していなければなりません。以下に、書式を示します。

**<Platform Name>\_<Host Name (or IP Address)>:Port Number**

プラットフォーム名とホスト名の間は、タブで区切ります。

ポート番号は、ホスト名の後ろに「:」(コロン)を加えて記入します。

(ポート番号を省略する場合には、コロンも省略すること)

各プラットフォームの設定は 1 行で完結し、行間に空の行は入れないようにします。

hostsfile.ini の記述例 :

```
#
#プラットフォーム名とローカルホストアドレス or IP アドレスを
#タブ区切りで表記する
#
#ポート番号を指定する場合には、
#アドレスの後ろにコロンをつけて表記する
# (省略可能。省略時にはデフォルトに設定される)
#
#ex)
#PlatformNameA      xxx.example.com
#PlatformNameB      xxx.example.com:3000
#
Platform00      xxx.example.com:3000
Platform01      xxx.example.com ←ポート番号が未記入=デフォルトポート(1099)
Platform02      127.0.0.1:3001
```

### 1.3. アクセスポリシーの設定（AccessPolicy.iniの編集）

AccessPolicy.ini に、hostsfile.ini に記述した連携先のプラットフォームに関するアクセスポリシー（プラットフォーム単位でのアクセス可否）を設定します。

AccessPolicy.iniのプロパティには、以下の表に示すようにコンポーネント連携機能に関するプロパティとコンポーネント転送機能に関するプロパティとがあり、後者の転送機能のプロパティについてはさらにPull型とPush型のプロパティに細分されます（これら連携機能および転送機能の概要については[2.コンポーネント連携機能]および[3.コンポーネント転送機能]をご覧ください）。AccessPolicy.iniにおいて、各プロパティの“=”の後に各機能を許可するプラットフォーム名を記述することによりアクセス制御を行います。以下の表では、AccessPolicy.iniを設定した自プラットフォームを“このPF”、連携先のプラットフォームを“連携先PF”と記述し、各プロパティの意味を説明しています。

アクセス管理に関するプロパティ

プロパティ名	意味
allowAll	以下の全ての連携および転送を許可する連携先 PF 名を示します。
allowRemoteInvocatoinFrom	このプロパティに指定した連携先 PF は、この PF 内のコンポーネントにリモート接続できます。
allowRemoteInvocatoinTo	この PF は、このプロパティに指定した連携先 PF 内のコンポーネントにリモート接続できます。
allowComponenntPullCopyTransfer	このプロパティに指定した連携先 PF は、この PF 内のコンポーネントをコピーし Pull 型で転送させて取得できます。
allowComponenntPullMoveTransfer	このプロパティに指定した連携先 PF は、この PF 内のコンポーネントを Pull 型で転送および移動させて取得できます。この PF からは転送対象のコンポーネントは削除されます。
allowComponenntPushTransfer	このプロパティに指定した連携先 PF は、この PF に対してコンポーネントを Push 型で転送できます。

各プロパティに複数のプラットフォームを指定する場合には、カンマ区切りでプラットフォーム名を記述します。また、自プラットフォームを示す場合には“Local”を指定することができます。さらに、ワイルドカード(\*)を用いて規則性のある名前のプラットフォームに対するアクセス可否を一括して設定することも可能です。

自プラットフォームおよび Platform01 に対してすべての転送および連携を許可し、Platform02 に対してコンポーネントのリモート接続のみを許可する場合の AccessPolicy.ini の記述例を以下に示します。

```
# 各種アクセスを許可するプラットフォーム名を記述する。
# -カンマ(,)区切りで複数の PF を列挙することが可能。
# -ワイルドカード(*)による完全一致や部分一致で指定することが可能。
# -local=Platform.ini で設定した自 PF のプラットフォーム名
#
allowAll=local, Platform01
allowRemoteInvocationFrom=Platform02
allowRemoteInvocationTo=
allowComponentPullCopyTransfer=
allowComponentPullMoveTransfer=
allowComponentPushTransfer=

# 以下、Broker 版用 (RMI 版では未使用)
#
datamanagement.default_componentcooperation_policy=true
datamanagement.default_componentpulltransfer_policy=true
datamanagement.default_componentpushtransfer_policy=true
```

#### 1.4. 各コンポーネントの設定（属性の編集）

転送対象となるコンポーネントの属性値を編集し、そのコンポーネントへのアクセス可否を制御します。以下の表に示すように、転送および連携の際に用いる公開名の属性、コンポーネント連携（リモート接続）に関する属性、コンポーネント転送（Push 型および Pull 型）に関する属性があります。

データ連携機能に関するコンポーネント属性

プロパティ名	意味
ComponentPublicName	コンポーネントの転送およびリモート接続の際に、自プラットフォームおよび連携先プラットフォームに公開する名称を示します。この名称を指定して転送およびリモート接続を行います。
AllowRemoteInvocation	連携先プラットフォームからこのコンポーネントへリモート接続することを許可する属性です。この属性が <b>true</b> かつ公開名が設定されている場合に、連携先プラットフォームからこのコンポーネントのメソッドを実行できます。
AllowPushTransfer	このコンポーネントを連携先プラットフォームへ <b>Push</b> 型転送することを許可する属性です。この属性が <b>true</b> かつ公開名が設定されている場合に転送できます。
AllowPullTransfer	このコンポーネントを( <b>Pull</b> 型で)転送するよう連携先プラットフォームから要求された場合に、転送することを許可する属性です。この属性が <b>true</b> かつ公開名が設定されている場合に転送されます。

これら属性の設定には、まずビルダー画面上で左側のコンポーネント上で右クリックをして「属性情報設定...」を選択します。



以下のようなダイアログが表示されますので、各属性を設定します。

MZ コンポーネント属性情報		閉じる
ComponentKey	サブルーチン6	<input type="checkbox"/> NULL
ComponentID	6	
ComponentPublicName		<input checked="" type="checkbox"/> NULL
AllowRemoteInvocation	<input type="radio"/> true <input checked="" type="radio"/> false	
AllowPullTransfer	<input type="radio"/> true <input checked="" type="radio"/> false	
AllowPushTransfer	<input type="radio"/> true <input checked="" type="radio"/> false	
ComponentKeys	日本語 : <input type="text"/> 英語 : <input type="text"/>	<input type="checkbox"/> NULL

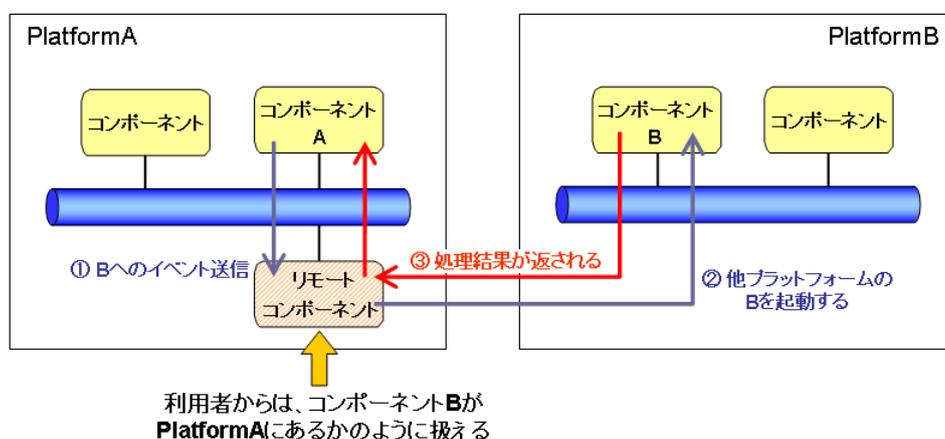
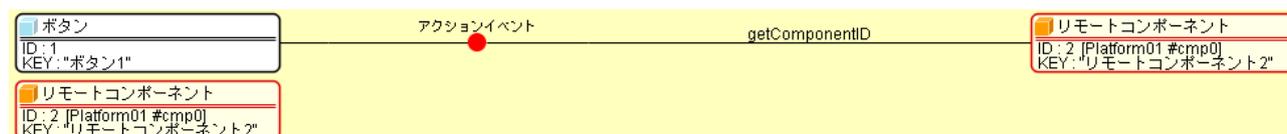
設定

属性情報設定画面（サブルーチンの場合）

## 2. コンポーネント連携機能

異なるプラットフォームに存在するコンポーネントを連携させる（リモート接続）ためには、リモートメソッドまたはリモートコンポーネントを用います。リモートメソッドとリモートコンポーネントの違いについては以下の節で述べます。

リモートメソッドおよびリモートコンポーネントはリモートで起動するメソッドの情報を保持するコンポーネントです。ビルダー上でリモートメソッドおよびリモートコンポーネントはオレンジ色の四角で表示され、外部のプラットフォームにあるコンポーネントのメソッドを、あたかも手元のプラットフォームに存在するコンポーネントのメソッドのように起動することができます。



### 2.1. リモートメソッド・リモートコンポーネント

他のプラットフォームに存在するコンポーネントのメソッドをリモート起動するためのコンポーネントとして、リモートメソッドとリモートコンポーネントを用意しております。これらの違いは次のとおりです。

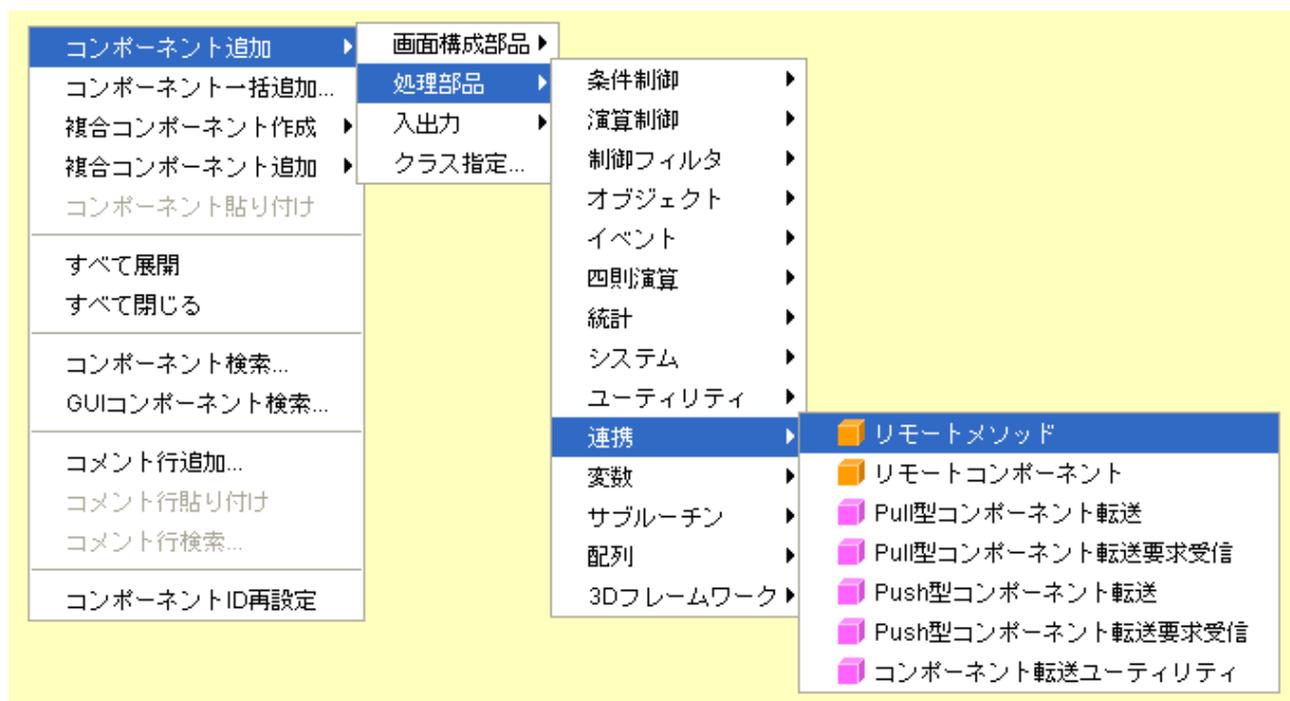
- ・リモートメソッド： リモートで起動するメソッドの情報を一つだけ持つコンポーネント。
- ・リモートコンポーネント： リモートで起動するメソッドの情報を複数持つことが可能なコンポーネント。

それぞれの設定方法について以下に示します。

### 2.2. リモートメソッドの設定

#### (1) リモートメソッドの追加

ビルダー画面をマウス右クリックすることにより表示されるポップアップメニューの「コンポーネント追加」より、[処理部品] - [連携] - [リモートメソッド]を順に選択することにより追加できます。



## (2)メソッド情報の設定

新規追加されたリモートメソッドは、参照先のメソッドに関する以下の情報が全てデフォルト値（未設定）になっております。

- ・連携先プラットフォーム名（String／空文字）
- ・連携先コンポーネント公開名（String／空文字）
- ・連携先コンポーネント ID（int／-1）
- ・呼び出しメソッド名（String／空文字）
- ・呼び出しメソッドの戻り値の型（String／空文字）
- ・呼び出しメソッドの引数の型の配列（String[]／要素数 0 の空配列）

これらのメソッド情報を編集は、次の2つの方法で行うことができます。

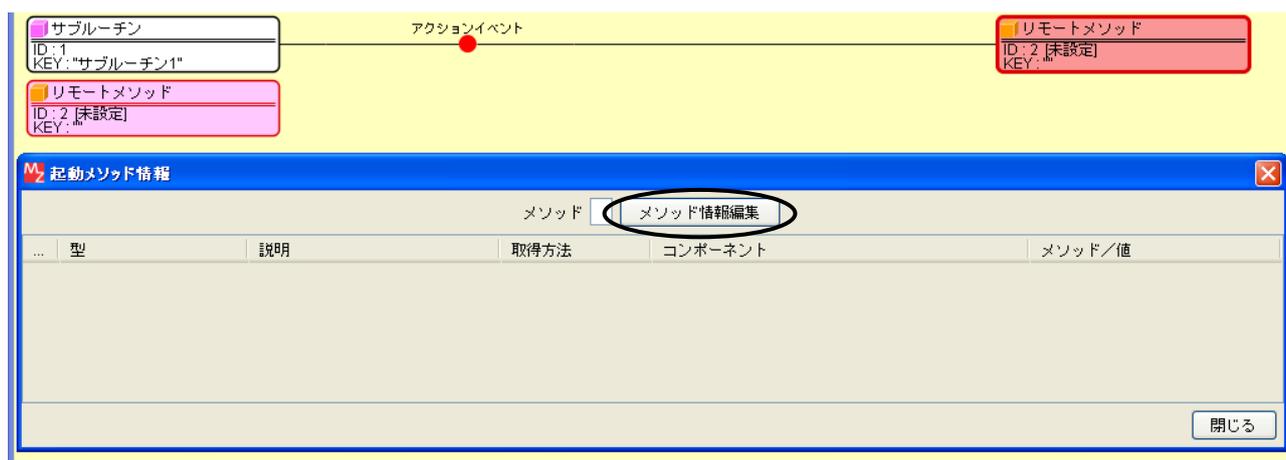
### ◇メソッド情報入力画面から編集する

コンポーネント上から表示されるポップアップメニューの「メソッド情報編集...」項目を選択することで、下右図のような「メソッド情報入力」ダイアログが表示されます。



「プラットフォーム名」は呼び出すコンポーネントが配置されているプラットフォームの Platform.ini に記載されている名前です。「コンポーネント公開名」はその呼び出し先であるプラットフォーム上のコンポーネントの公開名です。「コンポーネント ID」は呼び出し先であるプラットフォーム上におけるコンポーネント ID ですが、コンポーネント公開名を設定している場合には省略できます。「メソッド名」には実行するメソッドの名前を指定します。「メソッド戻り値」および「メソッド引数」には、それぞれのクラス名を記入します（例：int、java.lang.Long、java.lang.String 等）。メソッド引数については、引数のないメソッドを呼び出す場合に限り省略できます。各欄へ値を入力して「設定」ボタンを押下すると、コンポーネントにメソッド情報が設定されます。

また、「メソッド情報入力」ダイアログは、起動メソッドの接続コンポーネントから表示されるポップアップメニュー「起動メソッド設定...」を押下した際に表示される「起動メソッド情報」ダイアログからも呼び出すことができます。メソッド名が表示されているテキストフィールドの右にある、「メソッド情報編集」ボタンを押下します。



#### ◇属性設定画面から編集する

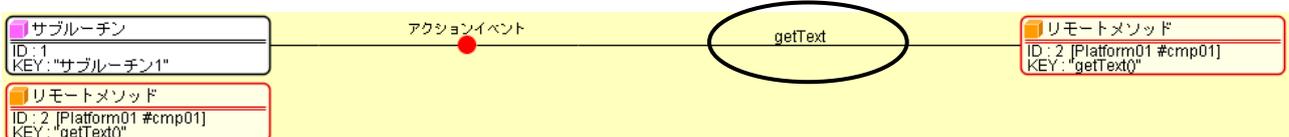
引数のないメソッドを呼び出す場合に限り、属性設定画面からも全てのメソッド情報値を編集することができます。引数があるメソッドについては、引数以外の値を編集することが可能です。

M2 コンポーネント属性情報	
ComponentKey	<input type="checkbox"/> NULL
ComponentID	2
ComponentPublicName	<input checked="" type="checkbox"/> NULL
AllowRemoteInvocation	<input type="radio"/> true <input checked="" type="radio"/> false
AllowPullTransfer	<input type="radio"/> true <input checked="" type="radio"/> false
AllowPushTransfer	<input type="radio"/> true <input checked="" type="radio"/> false
ComponentKeys	日本語 : <input type="text"/> 英語 : <input type="text"/> <input type="checkbox"/> NULL
BrokerName	<input type="checkbox"/> NULL
PlatformName	Platform01 <input type="checkbox"/> NULL
RemoteComponentID	-1
RemoteComponentPublicName	cmp01 <input type="checkbox"/> NULL
MethodName	getText <input type="checkbox"/> NULL
ReturnType	java.lang.String <input type="checkbox"/> NULL
IsLight	<input checked="" type="radio"/> true <input type="radio"/> false
IsAsync	<input type="radio"/> true <input checked="" type="radio"/> false
RemoteComponentName	<input type="checkbox"/> NULL
RemoteComponentKey	<input type="checkbox"/> NULL

設定

### (3) 起動メソッドの設定

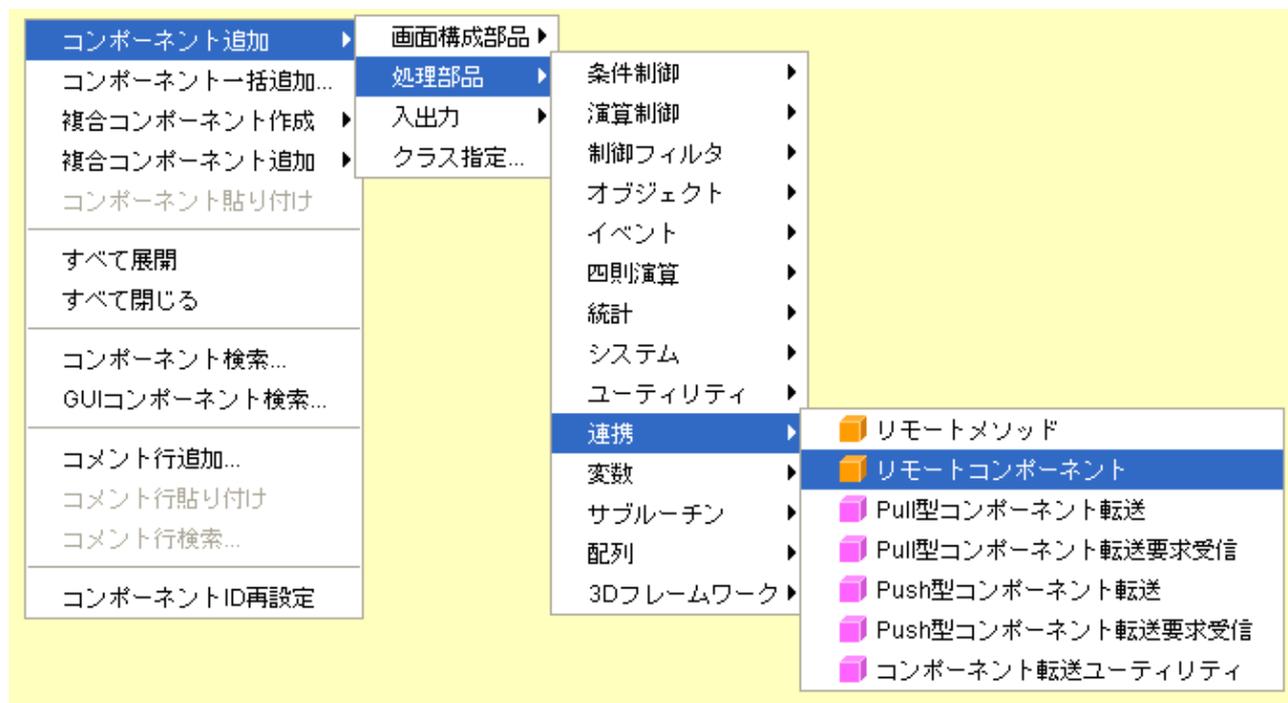
(2)でメソッド情報を設定した場合、接続先コンポーネント設定時に自動的に起動メソッドが設定されます。



## 2.3. リモートコンポーネントの設定

### (1) リモートコンポーネントの追加

ビルダー画面をマウス右クリックすることにより表示されるポップアップメニューの「コンポーネント追加」より、[処理部品] - [連携] - [リモートコンポーネント]を順に選択することにより追加できます。



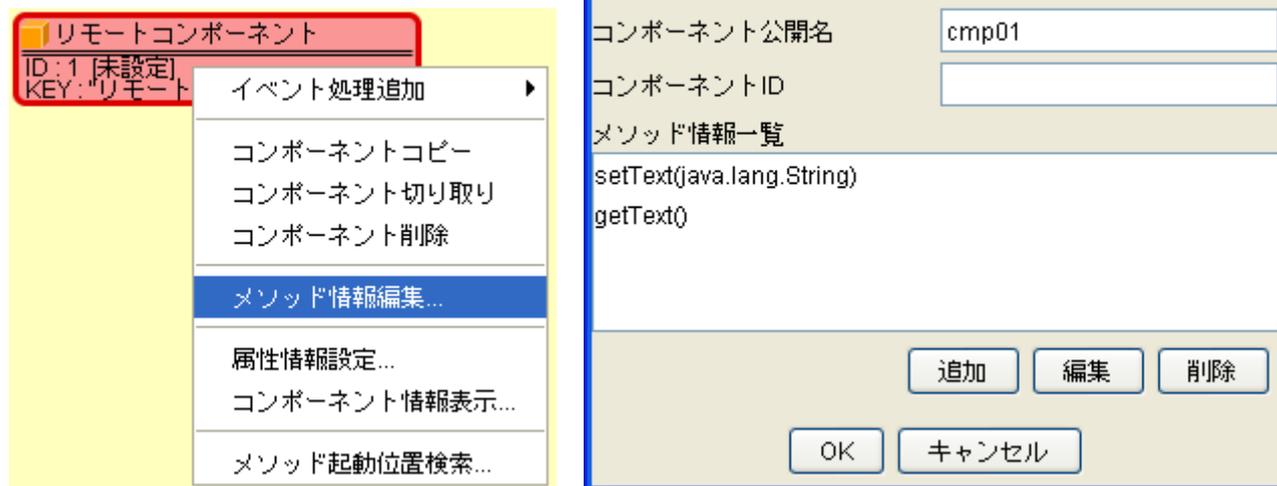
### (2) メソッド情報の設定

新規追加されたリモートコンポーネントは、参照先のメソッドに関する以下の情報が全てデフォルト値（未設定）になっております。

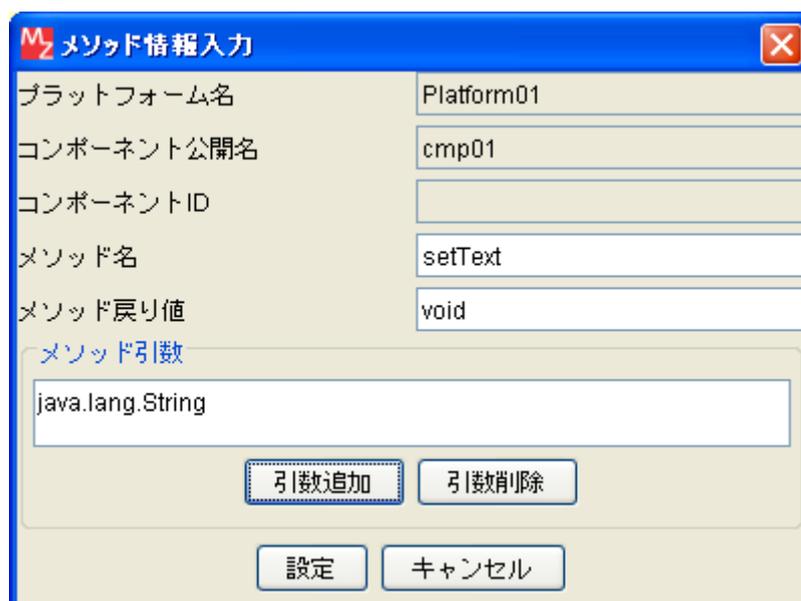
- ・連携先プラットフォーム名（String／空文字）
- ・連携先コンポーネント公開名（String／空文字）
- ・連携先コンポーネント ID（int／-1、オプション）
- ・呼び出しメソッド情報（複数）：
  - ・呼び出しメソッド名（String／空文字）
  - ・呼び出しメソッドの戻り値の型（String／空文字）
  - ・呼び出しメソッドの引数の型の配列（String[]／要素数 0 の空配列）

これらのメソッド情報を編集は、次のように行います。

コンポーネント上から表示されるポップアップメニューの「メソッド情報編集...」項目を選択することで、下右図のような「メソッド情報編集」ダイアログが表示されます。

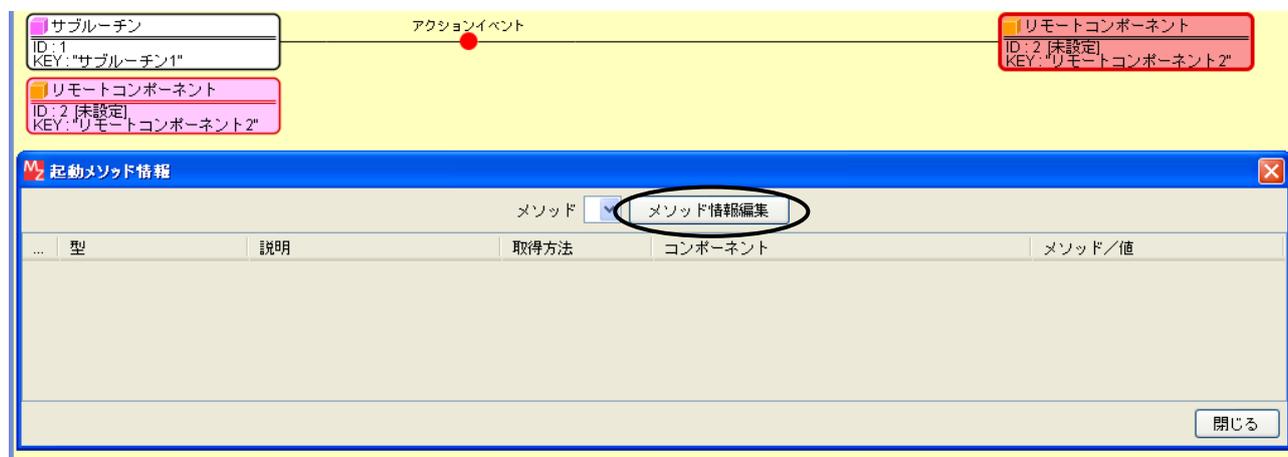


「プラットフォーム名」は呼び出すコンポーネントが配置されているプラットフォームの Platform.ini に記載されている名前です。「コンポーネント公開名」はその呼び出し先であるプラットフォーム上のコンポーネントの公開名です。「コンポーネント ID」は呼び出し先であるプラットフォーム上におけるコンポーネント ID ですが、コンポーネント公開名を設定している場合には省略できます。リモートメソッドとは異なり、起動するメソッドを複数登録できます。登録されているメソッドは「メソッド情報一覧」に表示されます。登録には「追加」ボタンを押します。このとき表示される「メソッド情報入力」ダイアログ上で、リモートメソッドの場合と同じく、メソッド名、メソッド戻り値、メソッド引数を順に設定します。



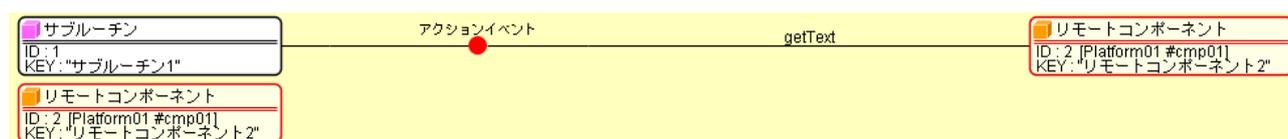
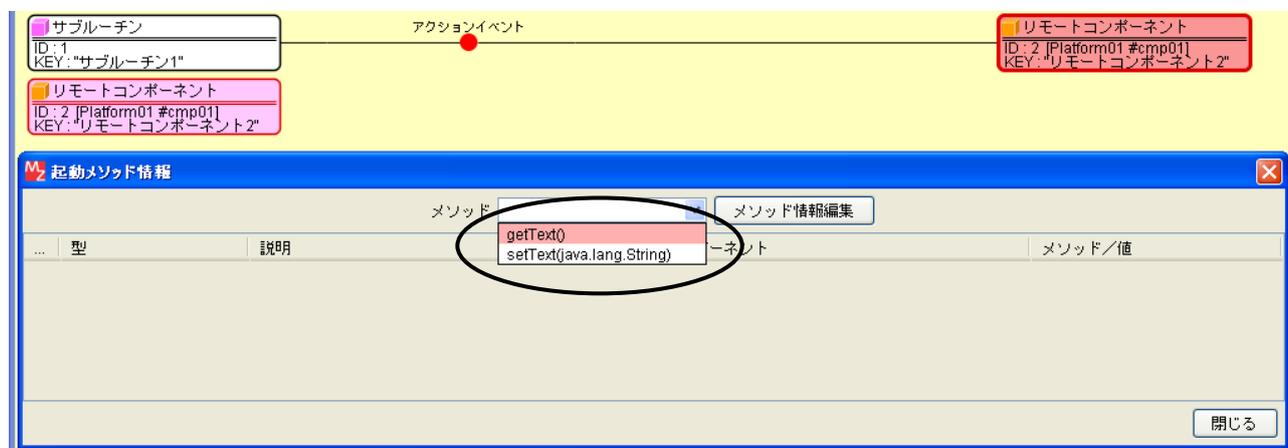
各欄へ値を入力して「設定」ボタンを押下すると、コンポーネントにメソッド情報が設定されます。また、「メソッド情報編集」ダイアログは、起動メソッドの接続コンポーネントから表示されるポップ

アップメニュー「起動メソッド設定...」を押下した際に表示される「起動メソッド情報」ダイアログからも呼び出すことができます。メソッド名が表示されているテキストフィールドの右にある、「メソッド情報編集」ボタンを押下します。



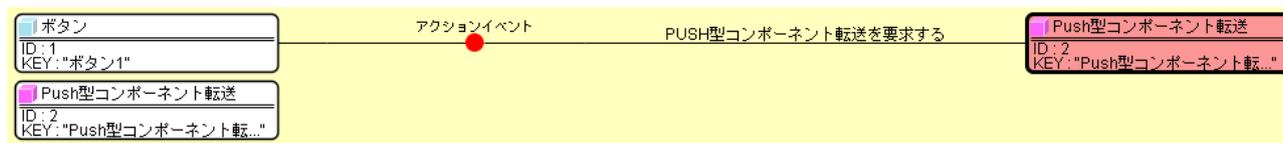
### (3) 起動メソッドの設定

(2)でメソッド情報を設定済みの場合、接続先コンポーネント設定時に「起動メソッド情報」ダイアログのドロップダウンリストに設定済みのメソッドが表示されます。メソッドを選択すると起動メソッドが設定されます。



### 3. コンポーネント転送機能

コンポーネント転送は、ビルダー上にコンポーネント転送コンポーネント(Pull 型と Push 型があります)を配置して、そのコンポーネント転送要求メソッドを起動することで実行します。



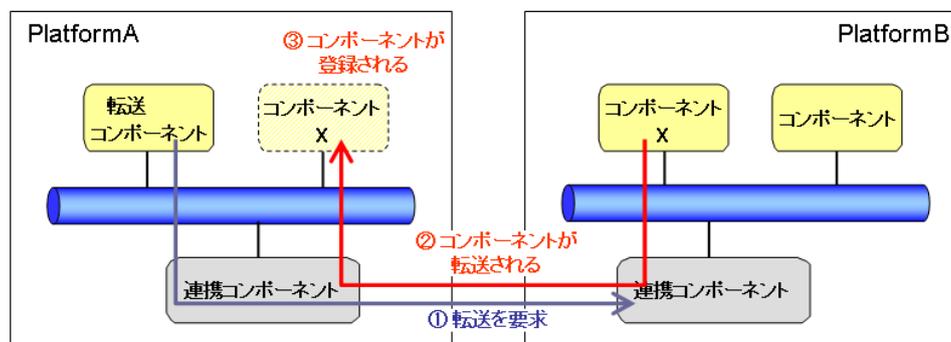
コンポーネント転送コンポーネントは、実体を持った処理コンポーネントであるため、ビルダー上での扱いは一般のコンポーネントと変わりありません。

#### 3.1. Pull型/Push型

コンポーネント転送には、転送要求に対してどちら向きにコンポーネントが転送されるかによって、Pull 型と Push 型の 2 種類があります。

##### 3.1.1. Pull型コンポーネント転送

他プラットフォームからコンポーネントを取り寄せて、手元のプラットフォームに登録する転送を Pull 型コンポーネント転送といいます。Pull 型転送を行うためには、「Pull 型コンポーネント転送」コンポーネントの転送要求メソッドを実行します。



※ここでの「連携コンポーネント」とは、プラットフォーム側のデータ連携に関する一機能を表わす

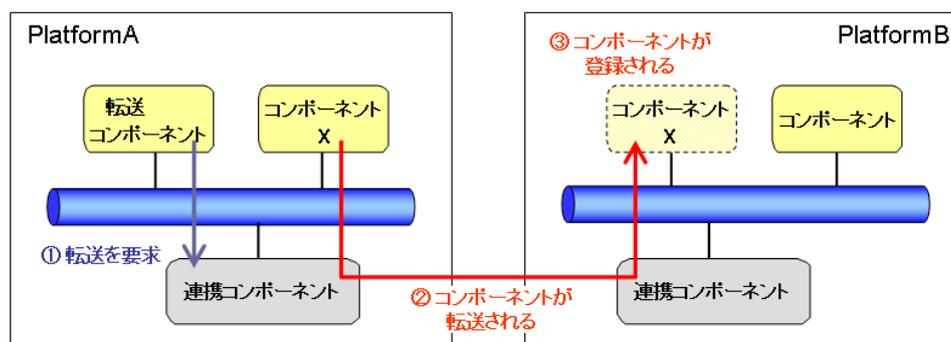
転送要求メソッドには 3.2 で示す転送の形態により次のメソッドがあります。

- PULL 型コンポーネント転送を要求する
- PULL 型コンポーネントコピー転送を要求する
- PULL 型コンポーネント連携コピー転送を要求する
- PULL 型コンポーネント連携置換転送を要求する
- PULL 型コンポーネント連携移動転送を要求する

引数には、転送したいコンポーネントの公開名 (複数の場合はリスト)、転送先プラットフォーム名、転送要求受信コンポーネント (転送要求受信コンポーネンについては[3.4転送要求受信コンポーネントを指定した転送]をご覧ください) 等を指定します。

### 3.1.2. Push型コンポーネント転送

手元のプラットフォームにあるコンポーネントを送り出して、他プラットフォームに登録する転送をPush型コンポーネント転送という。Push型転送を行うためには、「Push型コンポーネント転送」コンポーネントの転送要求メソッドを実行します。



転送要求メソッドには 3.2 で示す転送の形態により次のメソッドがあります。

- ・ PUSH 型コンポーネント転送を要求する
- ・ PUSH 型コンポーネントコピー転送を要求する
- ・ PUSH 型コンポーネント連携コピー転送を要求する
- ・ PUSH 型コンポーネント連携置換転送を要求する
- ・ PUSH 型コンポーネント連携移動転送を要求する

引数には、転送したいコンポーネントの公開名（複数の場合はリスト）、転送先プラットフォーム名、転送要求受信コンポーネント（転送要求受信コンポーネンについては[3.4転送要求受信コンポーネントを指定した転送]をご覧ください）等を指定します。

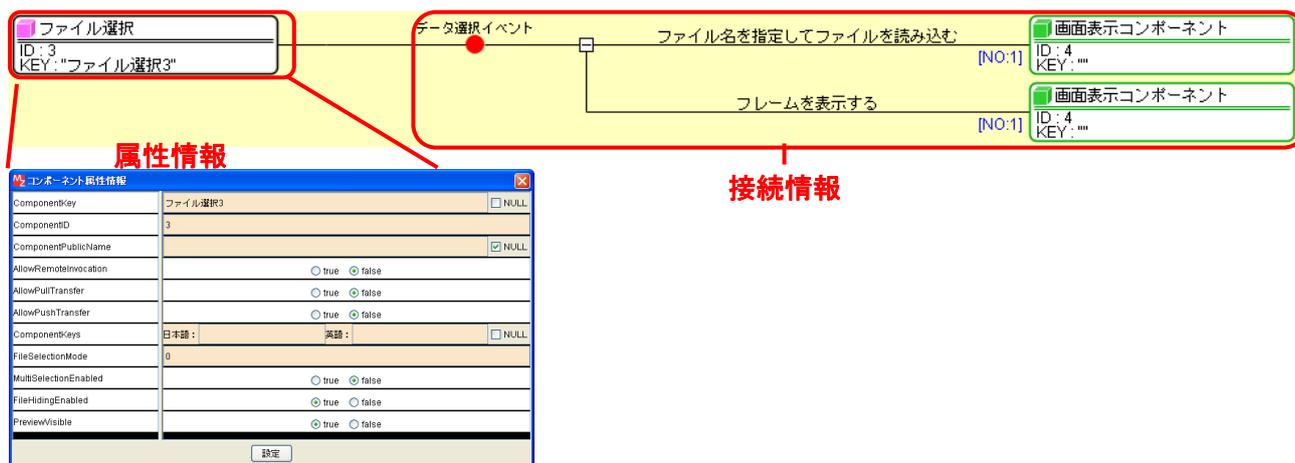
### 3.2. 様々なコンポーネント転送

コンポーネント転送コンポーネントには、様々な種類の転送要求メソッドが存在しています。

これらは、転送されるコンポーネント情報の範囲や、転送に伴う各プラットフォームでの挙動の違いによって区別されています。

#### ■ 転送されるコンポーネント情報の範囲

コンポーネント情報には、コンポーネントそのもののプロパティを示す「属性情報」と、そのコンポーネントから発生するイベントや起動メソッドの設定を示す「接続情報」があります。



#### ■ 転送に伴う挙動のパターン

現在用意されている5種のコンポーネント転送について、転送に伴って起こる挙動をまとめると次のようになります。

転送の名称	属性情報の転送	接続情報の転送	コンポーネント連携生成		転送対象の削除
			転送先	転送元	
コンポーネント転送	○	×	×	×	×
コンポーネントコピー転送	○	○	×	×	×
コンポーネント連携コピー転送	○	○	○	×	×
コンポーネント連携置換転送	○	○	○	○	×
コンポーネント連携移動転送	○	○	○	○	○

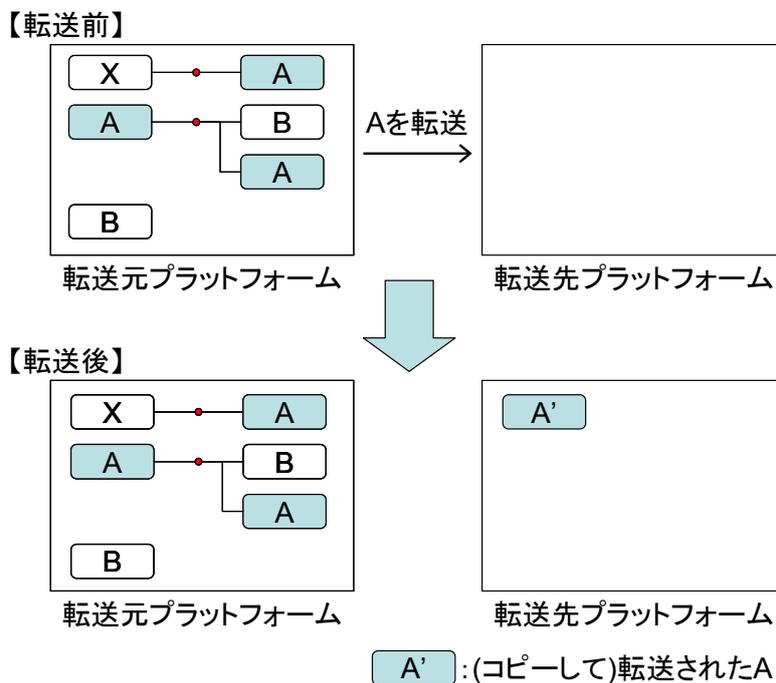
※転送先 = 転送されてきたコンポーネントを受け取るプラットフォーム

※転送元 = コンポーネントを送り出すプラットフォーム

各コンポーネント転送が実際にどのような振る舞いをするかを以下に説明します。

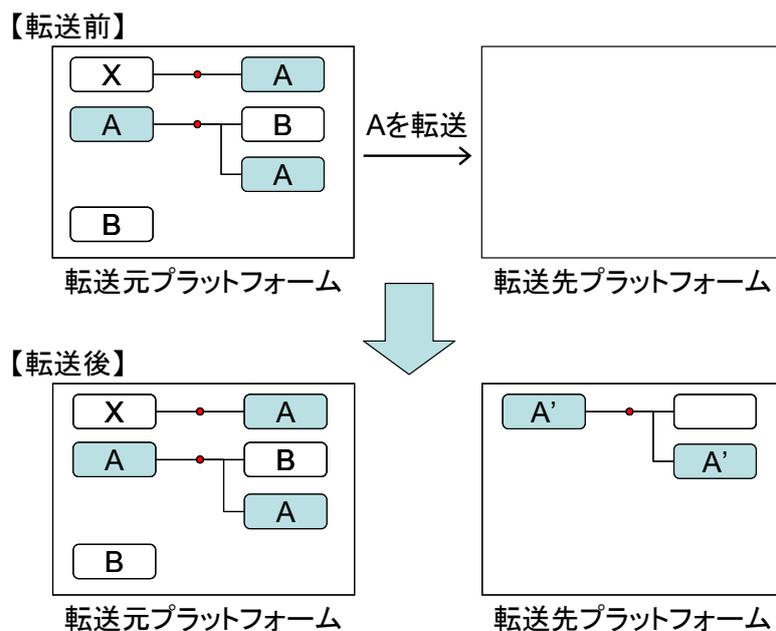
### 3.2.1. コンポーネント転送

最も基本的な転送で、コンポーネントの属性情報のみが転送されます。



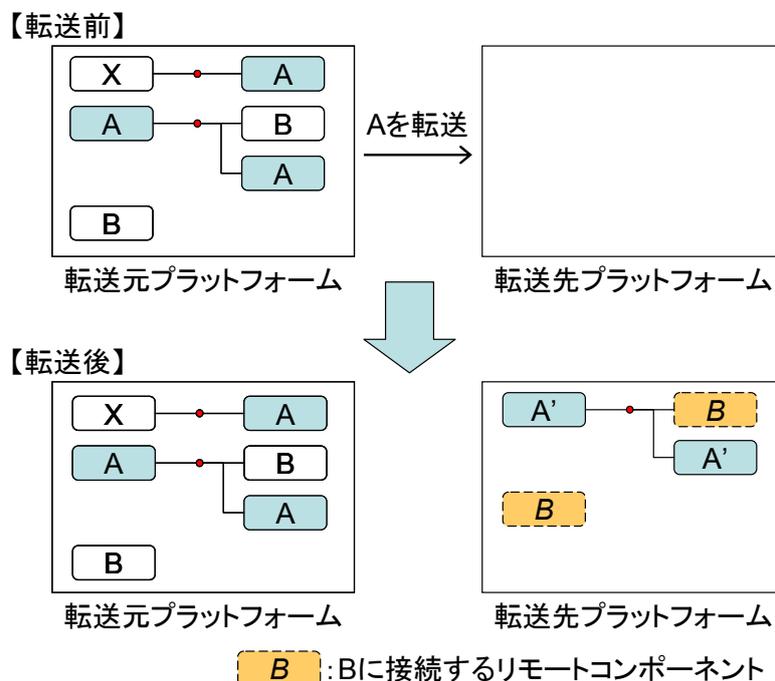
### 3.2.2. コンポーネントコピー転送

コンポーネントの接続情報も転送されます。接続コンポーネントが転送対象ではない場合には、転送先プラットフォームにおいて起動メソッドが空白に置き換わります。



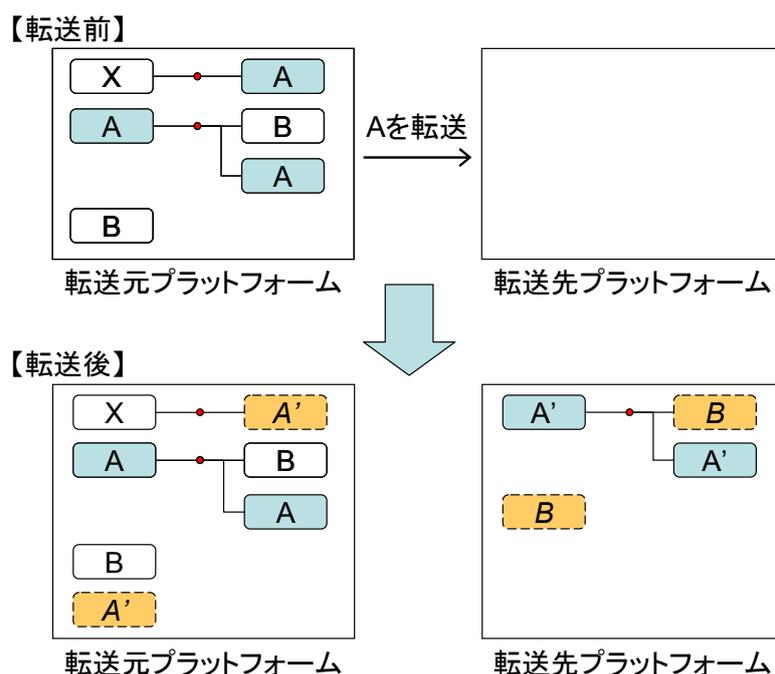
### 3.2.3. コンポーネント連携コピー転送

コンポーネントの接続情報も転送され、転送先で連携が構築されます。接続コンポーネントが転送対象ではない場合には、転送先プラットフォームにおいて、起動メソッドが転送元プラットフォームのコンポーネントへの連携（リモートコンポーネント）に置き換わります。



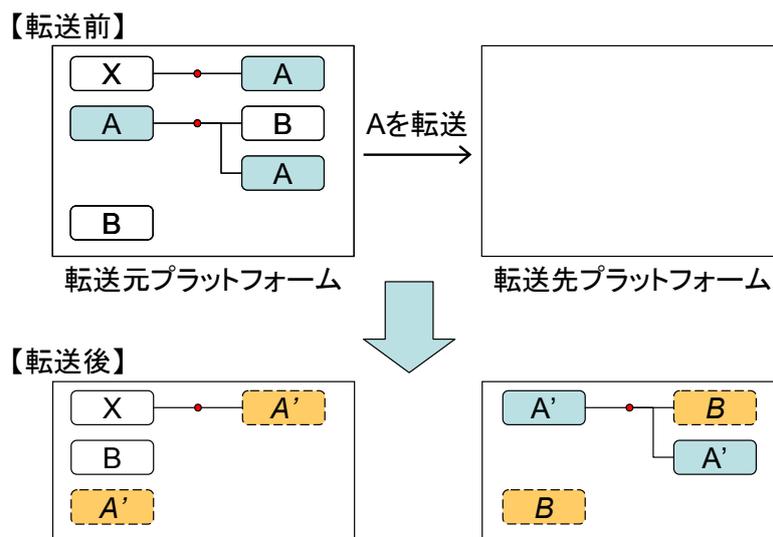
### 3.2.4. コンポーネント連携置換転送

コンポーネントの接続情報も転送され、転送先および転送元で連携が構築されます。接続コンポーネントが転送対象ではない場合には、転送先プラットフォームにおいて、起動メソッドが転送元プラットフォームのコンポーネントへの連携（リモートコンポーネント）に置き換わります。また、転送元プラットフォームにおいて、接続コンポーネントが転送対象である場合には、起動メソッドが転送先プラットフォームのコンポーネントへの連携に置き換わります。



### 3.2.5. コンポーネント連携移動転送

コンポーネントの接続情報も転送され、転送先および転送元で連携が構築されるとともに、転送対象のコンポーネントが転送元から削除されます（移動）。接続コンポーネントが転送対象ではない場合には、転送先プラットフォームにおいて、起動メソッドが転送元プラットフォームのコンポーネントへの連携（リモートコンポーネント）に置き換わります。また、転送元プラットフォームにおいて、接続コンポーネントが転送対象である場合には、起動メソッドが転送先プラットフォームのコンポーネントへの連携に置き換わり、転送対象そのものが削除されます。



### 3.3. 各種転送とプロパティの設定

上述の5種類の転送を行うに当たり、前述のアクセスポリシー（1.3節）およびコンポーネント属性（1.4節）が適切に設定されている必要があります。AccessPolicy.iniのプロパティについては、転送元および転送先プラットフォームにおいて、以下の表の○で示すプロパティに連携先のプラットフォームが設定されている必要があります。

AccessPolicy.ini のプロパティ設定

転送要求の受信側 PF (Pull の場合転送元、Push の場合転送先)	コピー		連携コピー		連携置換		連携移動	
	Pull	Push	Pull	Push	Pull	Push	Pull	Push
allowRemoteInvocatoinFrom	—	—	○	—	○	○	○	○
allowRemoteInvocatoinTo	—	—	—	○	○	○	○	○
allowComponenntPullCopyTransfer	○	—	○	—	○	—	○	—
allowComponenntPullMoveTransfer	—	—	—	—	—	—	○	—
allowComponenntPushTransfer	—	○	—	○	—	○	—	○
転送要求の送信側 PF (Pull の場合転送先、Push の場合転送元)	コピー		連携コピー		連携置換		連携移動	
	Pull	Push	Pull	Push	Pull	Push	Pull	Push
allowRemoteInvocatoinFrom	—	—	—	○	○	○	○	○
allowRemoteInvocatoinTo	—	—	○	—	○	○	○	○
allowComponenntPullCopyTransfer	—	—	—	—	—	—	—	—
allowComponenntPullMoveTransfer	—	—	—	—	—	—	—	—
allowComponenntPushTransfer	—	—	—	—	—	—	—	—

転送対象のコンポーネントの属性とそのコンポーネントから接続されている転送対象以外のコンポーネントの属性を次のように設定する必要があります（ComponentPublicName の○は「名称を設定する」の意味、それ以外の○は「true に設定する」の意味です）。

コンポーネントの属性設定

転送対象のコンポーネント	コピー		連携コピー		連携置換		連携移動	
	Pull	Push	Pull	Push	Pull	Push	Pull	Push
ComponentPublicName	○	○	○	○	○	○	○	○
AllowRemoteInvocation	—	—	—	—	○*	○*	○*	○*
AllowPushTransfer	—	○	—	○	—	○	—	○
AllowPullTransfer	○	—	○	—	○	—	○	—
転送対象以外の接続コンポーネント	コピー		連携コピー		連携置換		連携移動	
	Pull	Push	Pull	Push	Pull	Push	Pull	Push
ComponentPublicName	—	—	—	—	○	○	○	○
AllowRemoteInvocation	—	—	—	—	○	○	○	○
AllowPushTransfer	—	—	—	—	—	—	—	—
AllowPullTransfer	—	—	—	—	—	—	—	—

※転送対象以外のコンポーネントから接続されている場合

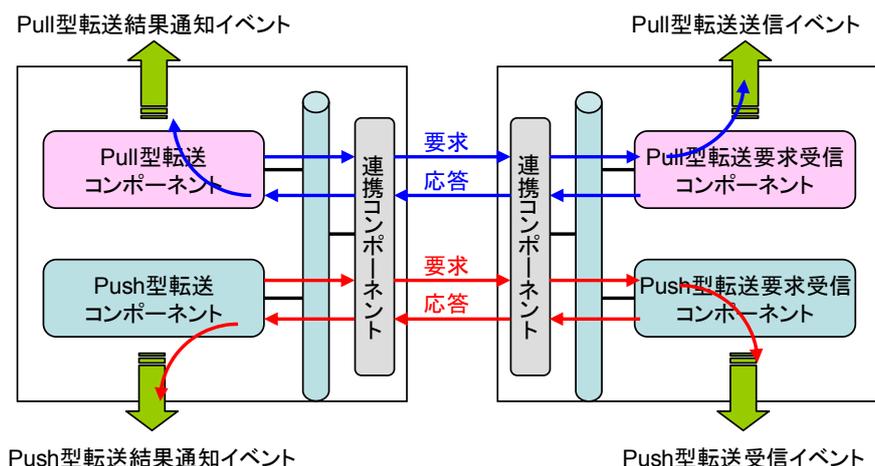
詳細なアクセス管理を行う場合にはこのような複雑な設定が必要ですが、アプリケーション構築中の動作確認等に当たってはすべてを許可する設定（AccessPolicy.ini の allowAll プロパティに連携先プラットフォーム名または“\*”を記入し、コンポーネントについては公開名を設定し、すべての属性を true にする）にすることも可能です。コンポーネントの属性については「コンポーネント転送ユーティリティ」コンポーネントを用いて、転送パターンに対応するメソッドを転送対象コンポーネントを引数に設定して起動することにより、関係するコンポーネントの属性を一括して設定することが可能です。

コンポーネント転送ユーティリティには次のメソッドがあります。

- Pull 型/Push 型コンポーネント連携コピー転送を許可する
- Pull 型/Push 型コンポーネント連携移動転送・連携置換転送を許可する
- コンポーネント公開名一覧をリスト形式で取得する
- 連携先プラットフォーム名一覧をリスト形式で取得する
- プラットフォーム名を取得する

### 3.4. 転送要求受信コンポーネントを指定した転送

Push 型/Pull 型転送コンポーネントの転送要求メソッドの引数には、転送したいコンポーネントの公開名（複数の場合はリスト）および転送先プラットフォーム名だけではなく、転送要求受信コンポーネントの公開名を設定することができます。転送要求受信コンポーネントには、「Pull 型コンポーネント転送要求受信」コンポーネントと「Push 型コンポーネント転送要求受信」コンポーネントがあり、それぞれ、下図に示すように、対応するコンポーネント転送コンポーネント（Pull 型コンポーネント転送、Push 型コンポーネント転送）との組み合わせで使用します。



転送要求受信コンポーネントを指定してコンポーネント転送を行った場合の挙動は以下の通りです。

#### 転送要求受信コンポーネントを指定したコンポーネント転送の挙動

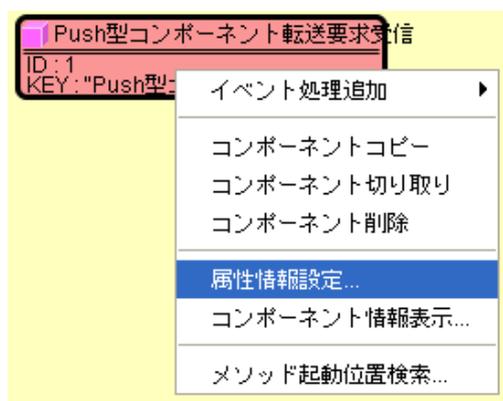
Pull 型転送	コンポーネントは、転送要求メソッドを実行した <u>Pull 型コンポーネント転送コンポーネント</u> が配置されている階層に転送されます。転送対象コンポーネントは、指定プラットフォームのどの階層にあっても転送可能です。
Push 型転送	コンポーネントは、指定プラットフォームの指定した <u>転送要求受信</u> が配置されている階層に転送されます。転送対象コンポーネントは、要求元のプラットフォームのどの階層にあっても転送可能です。
イベント発生	メソッドを実行したコンポーネント転送コンポーネントからは転送結果通知イベントが、イベント発生を許可している転送要求受信コンポーネントからは転送送信/転送受信イベントが発生する。

「Pull 型コンポーネント転送要求受信」コンポーネントと「Push 型コンポーネント転送要求受信」コンポーネントが対応するコンポーネント転送コンポーネントからの要求を受信可能とするためには、公開名と要求受信のターゲットとなることを許可する属性値を設定する必要があります。また、転送要求受信コンポーネントを指定しない場合および他の転送要求受信コンポーネントに向けた転送が行われた際にもイベント発生を行うか否かを設定する属性があります。

## コンポーネント転送要求受信コンポーネントの属性

プロパティ名	意味
ComponentPublicName	コンポーネント転送コンポーネントの転送要求メソッドを実行する際に引数に設定する公開名です。
IsPushTransferTarget (Push 型) IsPullTransferTarget (Pull 型)	転送要求受信コンポーネントが要求受信のターゲットとして使用されることを許可する属性であり、この属性が true で公開名が設定されている場合に、ターゲットとして指定することができます。
AllComponentTransferNotified	転送要求受信コンポーネントを指定しない場合および他の転送要求受信コンポーネントに向けた転送が行われた際にイベントを発生するか否かを指定する属性です。

これら属性の設定には、まずビルダー画面上でコンポーネント上で右クリックをして「属性情報設定...」を選択します。



以下のようなダイアログが表示されますので、各属性を設定します。

**MZ コンポーネント属性情報** ✕

ComponentKey	Push型コンポーネント転送要求受信2	<input type="checkbox"/> NULL
ComponentID	2	
ComponentPublicName		<input checked="" type="checkbox"/> NULL
AllowRemoteInvocation	<input type="radio"/> true <input checked="" type="radio"/> false	
AllowPullTransfer	<input type="radio"/> true <input checked="" type="radio"/> false	
AllowPushTransfer	<input type="radio"/> true <input checked="" type="radio"/> false	
TargetComponentName		<input checked="" type="checkbox"/> NULL
TargetComponentKey		<input checked="" type="checkbox"/> NULL
TargetComponentID	0	
AllComponentTransferNotified	<input type="radio"/> true <input checked="" type="radio"/> false	
IsPushTransferTarget	<input type="radio"/> true <input checked="" type="radio"/> false	

設定