# OpenRTM-aist (C++) - 機能 #4428
# direct接続Pull型の実装

2018/01/23 09:11 - n-miyamoto

| | | | |
|---|---|---|---|
| **ステータス:** | 終了 | **開始日:** | 2018/01/23 |
| **優先度:** | 通常 | **期日:** | |
| **担当者:** | n-miyamoto | **進捗率:** | 100% |
| **カテゴリ:** | | **予定工数:** | 0.00時間 |
| **対象バージョン:** | | | |
| **説明** | | | |

## 関係しているリビジョン

**リビジョン 3184 - 2018/01/23 09:15 - miyamoto**

[incompat,->RELENG_1_2] refs #4428

**リビジョン 3184 - 2018/01/23 09:15 - miyamoto**

[incompat,->RELENG_1_2] refs #4428

**リビジョン 3187 - 2018/01/23 14:07 - miyamoto**

[compat, ->RELENG_1_2] refs #4428

**リビジョン 3187 - 2018/01/23 14:07 - miyamoto**

[compat, ->RELENG_1_2] refs #4428

**リビジョン 3189 - 2018/01/23 16:55 - miyamoto**

[compat, ->RELENG_1_2] refs #4428

**リビジョン 3189 - 2018/01/23 16:55 - miyamoto**

[compat, ->RELENG_1_2] refs #4428

**リビジョン 3190 - 2018/01/23 16:58 - miyamoto**

[compat, ->RELENG_1_2] refs #4428

**リビジョン 3190 - 2018/01/23 16:58 - miyamoto**

[compat, ->RELENG_1_2] refs #4428

**リビジョン 3193 - 2018/01/23 17:41 - miyamoto**

[compat, ->RELENG_1_2] refs #4428

**リビジョン 3193 - 2018/01/23 17:41 - miyamoto**

[compat, ->RELENG_1_2] refs #4428

**リビジョン 3194 - 2018/01/23 19:20 - miyamoto**

[compat, ->RELENG_1_2] refs #4428

**リビジョン 3194 - 2018/01/23 19:20 - miyamoto**

[compat, ->RELENG_1_2] refs #4428

**リビジョン 3195 - 2018/01/24 08:49 - miyamoto**

[compat, ->RELENG_1_2] refs #4428

**リビジョン 3195 - 2018/01/24 08:49 - miyamoto**

[compat, ->RELENG_1_2] refs #4428

**リビジョン 3196 - 2018/01/24 09:12 - miyamoto**

[compat, ->RELENG_1_2] refs #4428

**リビジョン 3196 - 2018/01/24 09:12 - miyamoto**

[compat, ->RELENG_1_2] refs #4428

## 履歴

**#1 - 2018/01/23 15:33 - n-miyamoto**

*- ファイル write.png を追加*

*- ファイル read.png を追加*

*- ステータス を 担当 から 解決 に変更*

*- 進捗率 を 0 から 100 に変更*

- 追加した関数
  以下の関数を追加した。

- InPortBase
  - getLocalOutPort
    - ローカルのピアOutPortを取得
- InPortConnector
  - setOutPort
    - データをダイレクトに書き込むためのOutPortのサーバントを設定する
- OutPort
  - read
    - データをダイレクトに読み込む
- OutPortBase
  - getListeners
    - リスナホルダを取得する
- OutPortConnector
  - setDirectMode
    - ダイレクト接続モードに設定
  - directMode
    - ダイレクト接続モードかの判定

- データ転送時の処理
  データ書き込み時のシーケンス図は以下のようになっている。

まずOutPortConnectorのdirectMode関数を呼び出してPull型ダイレクト接続かを確認する。
ダイレクト接続に場合は変数m_directValueにデータを格納し、m_directNewDataをtrueにする。
この部分についてはミューテックスにより排他処理をする。
write.png

データ読み込み時のシーケンス図は以下のようになっている。
まずgetDirectData関数を呼び出す。getDirectData関数は、Pull型Direct接続ではない場合は即座にfalseを返す。
Pull型Direct接続の場合はOutPortのread関数を呼び出す。
read関数内では変数m_directValueを変数に格納し、m_directNewDataをfalseにしている。
read関数内はミューテックスにより排他処理をする。
read.png

- コネクタ接続時について
  コネクタ接続時にOutPortConnectorのsetDirectMode関数を呼び出すことにより、Pull型direct接続であることを設定する必要がある。

```
OutPortConnector*
OutPortBase::createConnector(const ConnectorProfile& cprof,
                             coil::Properties& prop,
                             OutPortProvider* provider)
{
(省略)
        if (coil::normalize(prop["interface_type"]) == "direct")
      {
          connector->setDirectMode();
      }
```

またInPortConnctorにOutPortのサーバントを設定する。

```
  ConnectorInfo profile(cprof.name,
                        cprof.connector_id,
                        CORBA_SeqUtil::refToVstring(cprof.ports),
```

```
                    prop);
(省略)
    if (coil::normalize(prop["interface_type"]) == "direct")
    {
        OutPortBase* outport = getLocalOutPort(profile);
        if (outport == NULL)
        {
            RTC_DEBUG(("interface_type is direc "
                "but a peer InPort servant could not be ));btained."
            delete connector;
            return 0;
        }

        connector->setOutPort(outport);

    }
```

#2 - 2018/09/13 09:26 - n-miyamoto

*- ステータス を 解決 から 終了 に変更*

## ファイル

| | | | |
|---|---|---|---|
| write.png | 37.1 KB | 2018/01/23 | n-miyamoto |
| read.png | 44.6 KB | 2018/01/23 | n-miyamoto |