

コア - バグ #3574

Manager::getORB() 等のリファレンスカウントとownershipを整理する

2016/07/19 13:35 - n-ando

ステータス:	終了	開始日:	2016/07/19
優先度:	通常	期日:	
担当者:	n-ando	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

getORB().getPOA().getPOAManager()
などはリファレンスカウントを増やさずに参照を返すため_varで受けるとオブジェクトが削除される問題。
CORBAのマナー的には、関数内でduplicateなどでリファレンスカウントを増やしownershipを保持したまま、参照を返すべき。

安藤様

宮本です。

direct接続したときに発生する不具合ですが、原因を特定できたのでご報告します。

まず原因箇所はOutPortBaseクラスのgetLocalInPort関数の以下の部分でした。

```
CORBA::ORB_var orb = RTC::Manager::instance().getORB();
```

```
PortableServer::POA_var poa = RTC::Manager::instance().getPOA();
```

この部分を以下のように修正したら不具合は発生しなくなりました。

```
CORBA::ORB_var orb = CORBA::ORB::_duplicate(RTC::Manager::instance().getORB());
```

```
PortableServer::POA_var poa = PortableServer::POA::_duplicate(RTC::Manager::instance().getPOA());
```

修正前のようにgetORB関数で取得したポインタをそのまま渡してしまうと、ORB_varのデストラクタでCORBA::release関数を呼び出してリファレンスカウントが1つ減ります。
つまりgetLocalInPort関数を呼び出すたびにリファレンスカウントが減ってしまうのが原因なので、_duplicate関数でリファレンスカウントを増やしてやれば不具合はなくなるはずです。

ソースコードを読んだ結果、以下の部分にも同様の記述があったので修正する必要があると思います。

- ・ CORBA_SeqUtil.cppのrefToVstring関数
- ・ CorbaNamingクラスのコンストラクタ

以上です。

関係しているリビジョン

リビジョン 2757 - 2016/07/20 16:35 - n-ando

[incompat,RELENG_1_2] Mananger::getORB().getPOA.getPOAManager() behaviour has been changed, and new theORB(),thePOA(),thePOAManager operations have been added. For the details, see function reference document in the header. refs #3574

リビジョン 2757 - 2016/07/20 16:35 - n-ando

[incompat,RELENG_1_2] Mananger::getORB().getPOA.getPOAManager() behaviour has been changed, and new theORB(),thePOA(),thePOAManager operations have been added. For the details, see function reference document in the header. refs #3574

履歴

#1 - 2016/07/19 13:45 - n-ando

getORB()を利用している関数

```
CORBA_SeqUtil.h: CORBA::ORB_ptr orb = ::RTC::Manager::instance().getORB();  
CorbaPort.h: CORBA::ORB_ptr orb = Manager::instance().getORB();
```

```

CorbaPort.h:          CORBA::ORB_ptr orb = Manager::instance().getORB();
CorbaPort.h:          CORBA::ORB_ptr orb = ::RTC::Manager::instance().getORB();
Manager.h:           CORBA::ORB_var  getORB();
InPortCorbaCdrConsumer.cpp:  CORBA::ORB_ptr orb = RTC::Manager::instance().getORB();
InPortCorbaCdrConsumer.cpp:  CORBA::ORB_ptr orb = RTC::Manager::instance().getORB();
InPortCorbaCdrProvider.cpp:  CORBA::ORB_ptr orb = ::RTC::Manager::instance().getORB();
Manager.cpp:          CORBA::ORB_ptr Manager::getORB()
Manager.cpp:          RTC_TRACE("Manager::getORB()");
ManagerServant.cpp:        obj = m_mgr.getORB()->resolve_initial_references("omniINSPOA");
ManagerServant.cpp:        obj = m_mgr.getORB()->resolve_initial_references((char*)"omniINSPOA");
ManagerServant.cpp:        ior = m_mgr.getORB()->
ManagerServant.cpp:        mobj = m_mgr.getORB()->string_to_object(mgrloc.c_str());
ManagerServant.cpp:        ior = m_mgr.getORB()->object_to_string(RTM::Manager::_duplicate(mgr));
NamingManager.cpp:        ior = RTC::Manager::instance().getORB()->object_to_string(obj.in());
NamingManager.cpp:        getORB()->string_to_object(iorstr.c_str());
NamingManager.cpp:        ior = RTC::Manager::instance().getORB()->object_to_string(obj.in());
NamingManager.cpp:        getORB()->string_to_object(iorstr.c_str());
NamingManager.cpp:        ior = RTC::Manager::instance().getORB()->object_to_string(obj.in());
NamingManager.cpp:        getORB()->string_to_object(iorstr.c_str());
NamingManager.cpp:        orb = CORBA::ORB::_duplicate(m_manager->getORB());
OutPortBase.cpp:        CORBA::ORB_var orb = RTC::Manager::instance().getORB();
OutPortCorbaCdrConsumer.cpp:  CORBA::ORB_ptr orb = ::RTC::Manager::instance().getORB();
OutPortCorbaCdrConsumer.cpp:  CORBA::ORB_ptr orb = RTC::Manager::instance().getORB();
OutPortCorbaCdrProvider.cpp:  CORBA::ORB_ptr orb = ::RTC::Manager::instance().getORB();
RTOject.cpp:        m_pORB(CORBA::ORB::_duplicate(manager->getORB())),
RTOject.cpp:        m_portAdmin(manager->getORB(), manager->getPOA()),

```

getPOA()を利用している関数

```

CorbaPort.h:          m_oid = Manager::instance().getPOA()->servant_to_id(m_servant);
CorbaPort.h:          getPOA()->activate_object_with_id(m_oid, m_servant);
CorbaPort.h:          obj = Manager::instance().getPOA()->id_to_reference(m_oid);
CorbaPort.h:          m_oid = Manager::instance().getPOA()->servant_to_id(m_servant);
CorbaPort.h:          getPOA()->activate_object_with_id(m_oid, m_servant);
CorbaPort.h:          Manager::instance().getPOA()->deactivate_object(m_oid);
Manager.h:          PortableServer::POA_ptr getPOA();
Manager.h:          PortableServer::POAManager_ptr getPOAManager();
Manager.cpp:          if(CORBA::is_nil(this->getPOAManager()))
Manager.cpp:          this->getPOAManager()->activate();
Manager.cpp:          PortableServer::POA_ptr Manager::getPOA()
Manager.cpp:          RTC_TRACE("Manager::getPOA()");
Manager.cpp:          PortableServer::POAManager_ptr Manager::getPOAManager()
Manager.cpp:          RTC_TRACE("Manager::getPOAManager()");
OutPortBase.cpp:        PortableServer::POA_var poa = RTC::Manager::instance().getPOA();
RTOject.cpp:        m_pPOA(PortableServer::POA::_duplicate(manager->getPOA())),
RTOject.cpp:        m_portAdmin(manager->getORB(), manager->getPOA()),
RTOjectStateMachine.cpp:        PortableServer::POA_ptr poa = RTC::Manager::instance().getPOA();

```

getPOAManager()を利用している関数

```

Manager.h:          PortableServer::POAManager_ptr getPOAManager();
Manager.cpp:          if(CORBA::is_nil(this->getPOAManager()))
Manager.cpp:          this->getPOAManager()->activate();
Manager.cpp:          PortableServer::POAManager_ptr Manager::getPOAManager()
Manager.cpp:          RTC_TRACE("Manager::getPOAManager()");

```

#2 - 2016/07/20 16:36 - n-ando

- ステータスを新規から解決に変更

- 進捗率を0から100に変更

getORB, getPOA, getPOAManagerの振る舞いを変更。すべて、内部で参照をduplicateすることにした。
 加えて、theORB, thePOA, thePOAManager関数を追加。これらはduplicateせずに参照を返すのみの関数。

#3 - 2016/08/09 15:03 - n-ando

- ステータスを解決から終了に変更