

OpenRTM-aist (C++) - バグ #760

OpenRTM-aist(C++)-delete_component時のPOA不一致エラーの問題

2009/07/02 16:07 - 匿名ユーザー

ステータス:	終了	開始日:	2009/07/02
優先度:	通常	期日:	
担当者:	kurihara	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			
ManagerのPOAポリシーを_default_POAから、PERSISTENT・USER_ID設定へ変更した事によりdelete_component時にサーバントアクティベートエラーが発生する。			
(エラーメッセージ)			
omniORB: ERROR -- A servant has been deleted that is still activated. id: root<3> (active)			
omniORB: ERROR -- A servant has been deleted that is still activated. id: root<1> (active)			

履歴

- #1 - 2009/07/13 15:09 - kurihara
- 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

元々は、デフォルトPOAを使用した場合にサービスポートのACTIVATE/DEACTIVATEにてIORが変化するという問題があり、この対処方法としてManagerクラスのgetPOA()で取得できるPOAは、PERSISTENT・USER_IDを設定したPOAを返すようにしていたが、これにより副作用があることが判明した。そこで、以前のようにManager::getPOA()はデフォルトPOAを返すように戻し、CorbaPortクラスにてACTIVATE/DEACTIVATEでIORが変わらないように修正を行った。修正箇所は下記のとおり。

```
$ svn diff Manager.cpp
Index: Manager.cpp
=====
--- Manager.cpp      (リビジョン 1443)
+++ Manager.cpp      (作業コピー)
@@ -923,13 +923,7 @@
     m_pORB = CORBA::ORB_init(argc, argv);
     // Get the RootPOA
     CORBA::Object_var obj = m_pORB->resolve_initial_references("RootPOA");
- PortableServer::POA_var rootPOA = PortableServer::POA::_narrow(obj);
- CORBA::PolicyList policies;
- policies.length(2);
- policies[(CORBA::ULong)0] = rootPOA->create_lifespan_policy(PortableServer::PERSISTENT);
- policies[(CORBA::ULong)1] = rootPOA->create_id_assignment_policy(PortableServer::USER_ID);
- PortableServer::POAManager_var rootManager = rootPOA->the_POAManager();
- m_pPOA = rootPOA->create_POA("persistent", rootManager, policies);
+ m_pPOA = PortableServer::POA::_narrow(obj);
+ if (CORBA::is_nil(m_pPOA))
+ {
+     RTC_ERROR(("Could not resolve RootPOA."));
$ svn diff CorbaPort.h
Index: CorbaPort.h
=====
--- CorbaPort.h      (リビジョン 1443)
+++ CorbaPort.h      (作業コピー)
@@ -491,11 +491,39 @@
 * @endif
 */
 NVList m_providers;
- std::vector<PortableServer::RefCountServantBase*> m_servants;
- std::vector<std::string> m_instance_name;
```

```

-
+
+      /*!
+      * @if jp
+      * @brief Provider
+      * @else
+      * @brief The structure to be stored Provider's information.
+      * @endif
+      */
+      struct ProviderInfo
+      {
+          ProviderInfo(PortableServer::RefCountServantBase* _servant, PortableServer::ObjectId_var _objectid)
+          : servant(_servant),
+            oid(_objectid)
+          {}
+
+          ProviderInfo(const ProviderInfo& pinfo)
+          : servant(pinfo.servant),
+            oid(pinfo.oid)
+          {}
+
+          ProviderInfo operator=(const ProviderInfo& _pinfo)
+          {
+              ProviderInfo pinfo(_pinfo);
+              return pinfo;
+          }
+
+          PortableServer::RefCountServantBase* servant;
+          PortableServer::ObjectId_var oid;
+      };
+      typedef std::map<std::string, ProviderInfo> ServantMap;
+      ServantMap m_servants;
+
+      /*!
+      * @if jp
+      * @brief Consumer
+      * @else
+      * @brief The structure to be stored Consumer information.

```

```

$ svn diff CorbaPort.cpp
Index: CorbaPort.cpp

```

```

=====
--- CorbaPort.cpp      (リビジョン 1443)
+++ CorbaPort.cpp      (作業コピー)
@@ -70,8 +70,7 @@
     return false;
 }

-    PortableServer::ObjectId_var oid =
-        PortableServer::string_to_ObjectId(instance_name);
+    PortableServer::ObjectId_var oid = Manager::instance().getPOA()->servant_to_id(&provider);
     try
     {
         Manager::instance().getPOA()->activate_object_with_id(oid, &provider);
@@ -95,10 +94,7 @@
         CORBA::String_var ior = orb->object_to_string(obj);
         CORBA_SeqUtil::
             push_back(m_providers, NVUtil::newNV(key.c_str(), ior));
-        m_servants.push_back(&provider);
-
-        std::string name(instance_name);
-        m_instance_name.push_back(name);
+        m_servants.insert(std::pair<std::string, ProviderInfo>(instance_name, ProviderInfo(&provider, oid)));
         return true;
     };
@@ -139,13 +135,12 @@
     */
     void CorbaPort::activateInterfaces()
     {
-        for (int i(0), len(m_servants.size()); i < len; ++i)
+        ServantMap::iterator it = m_servants.begin();
+        while(it != m_servants.end())
         {
             try
             {

```

```

-         PortableServer::ObjectId_var oid =
-             PortableServer::string_to_ObjectId(m_instance_name[i].c_str());
-         Manager::instance().getPOA()->activate_object_with_id(oid, m_servants[i]);
+         Manager::instance().getPOA()->activate_object_with_id(it->second.oid, it->second.servant);
    }
    catch(const ::PortableServer::POA::ServantAlreadyActive &)
    {
@@ -153,6 +148,7 @@
        catch(const ::PortableServer::POA::ObjectAlreadyActive &)
        {
        }
+    it++;
    }
}

@@ -165,11 +161,11 @@
    */
    void CorbaPort::deactivateInterfaces()
    {
-    for (int i(0), len(m_servants.size()); i < len; ++i)
+    ServantMap::iterator it = m_servants.begin();
+    while(it != m_servants.end())
    {
        PortableServer::ObjectId_var oid =
            PortableServer::string_to_ObjectId(m_instance_name[i].c_str());
        Manager::instance().getPOA()->deactivate_object(oid);
+        Manager::instance().getPOA()->deactivate_object(it->second.oid);
+    it++;
    }
}

```

#2 - 2009/07/27 11:56 - n-ando

- ステータスを 解決 から 終了 に変更

#3 - 2010/03/09 14:01 - 匿名ユーザー

[r1444](#)