

OpenRTM-aist (C++) - バグ #838

PortBaseを自動変数として作成した場合のメモリリーク問題

2009/07/31 14:44 - n-ando

ステータス:	終了	開始日:	2009/07/31
優先度:	通常	期日:	
担当者:	n-ando	進捗率:	100%
カテゴリ:		予定工数:	2.00時間
対象バージョン:			
説明			
PortBaseを通常のRTCのメンバ変数として宣言・使用した場合ではなく、動的に生成、あるいは自動変数として生成した場合にメモリリークが発生する現象が発見された。			

履歴

#1 - 2009/07/31 14:51 - n-ando

- ステータス を 新規 から 終了 に変更
- 進捗率 を 0 から 100 に変更

PortBaseのコンストラクタにおいて、以下の変更を行うことでメモリリークが消滅したようである。

```
        : rtclog(name)
    {
        m_profile.name = CORBA::string_dup(name);
-       m_objref = RTC::PortService::_duplicate(this->_this());
+       m_objref = this->_this();
        m_profile.port_ref = m_objref;
        m_profile.owner = RTC::RTOBJECT::_nil();
    }
```

onExecute内でCorbaPortを自動変数として宣言、使用した結果、100回程度の呼び出しで数kB程度メモリ使用量RESが増加がしていたが、上記の変更を施すことによって、数千回の呼び出しでも増加量がほぼ0kBとなった。

CORBAの関数呼び出しおよび引数受け渡し規則に準じるならば、_this()呼び出しの戻り値の所有権は呼び出し側にあり、それを_var変数として受け取る際にはduplicate()は必要ない。

また、自動変数として宣言され、registerPortされないIPortBaseは、RTOBJECTによってdeactivateされないの、削除時にServantが残りメモリリークの原因にもなる。したがって、デストラクタで以下のようにdeactivateを行うように変更した。

```
PortBase::~PortBase()
{
+   RTC_TRACE(("~PortBase()"));
+   try
+   {
+       PortableServer::ObjectId_var oid = _default_POA()->servant_to_id(this);
+       _default_POA()->deactivate_object(oid);
+   }
+   catch (...)
+   {
+       RTC_WARN(("Unknown exception caught."));
+   }
}
```