

# OpenRTM-aist (C++) - バグ #2404

## RtORBでコンパイルが通らない

2012/04/08 19:56 - n-ando

ステータス:	終了	開始日:	2012/04/08
優先度:	通常	期日:	
担当者:		進捗率:	100%
カテゴリ:		予定工数:	0.00時間
対象バージョン:			
<b>説明</b>			
trunk (r2330時点) でRtORB (1.0) モードでコンパイルが通らない。 これを修正する。			

### 関係しているリビジョン

リビジョン 2351 - 2012/04/09 12:40 - n-ando

Some modification to support RtORB. refs #2404

リビジョン 2351 - 2012/04/09 12:40 - n-ando

Some modification to support RtORB. refs #2404

リビジョン 2351 - 2012/04/09 12:40 - n-ando

Some modification to support RtORB. refs #2404

### 履歴

#1 - 2012/04/08 19:59 - n-ando

- ダミーのStubファイルが生成されない
  - これにより、Stubをインクルードしているところでエラーになる。
- local\_service/nameservice\_file でrtorbのスタブと既存ファイル名がぶつかるのでエラーになる。
- ec/logical\_time も同様

#2 - 2012/04/08 22:47 - n-ando

RtORBではIDLが1ファイルしか使えないため、すべてのrtm/idl以下のIDLをOpenRTM-aist.idlというファイルにマージしている。整合性のために\*Skel.hをMakefile.amで生成するようにしているが、Stub.hは生成されていなかったため、いくつかのファイルでコンパイルエラーが発生。  
Makefile.amを修正し、\*Stub.hも生成するようにした。

Index: idl/Makefile.am

```
=====
--- idl/Makefile.am      ($(O*c吾 (B 2335)
+++ idl/Makefile.am      (嵌罐$(O-c湯(B)
@@ -81,6 +81,15 @@
     DataPortSkel.h          /
     ExtendedDataTypesSkel.h /
     InterfaceDataTypesSkel.h
+DUMMY_STUBS =                /
+    SDOPackageStub.h        /
+    RTCStub.h                /
+    OpenRTMStub.h           /
+    ManagerStub.h           /
+    BasicDataTypeStub.h     /
+    DataPortStub.h          /
+    ExtendedDataTypesStub.h /
+    InterfaceDataTypesStub.h
ORB_SRC = $(RTORB_SRC)
endif
@@ -170,8 +179,14 @@
    $(top_srcdir)/build/makewrapper.py $$f ; /
done ;
@for f in $(DUMMY_SKELS) ; do /
+    echo cp OpenRTM-aistSkel.h $$f ; /
```

```

+           cp OpenRTM-aistStub.h $$f ; /
+           cp OpenRTM-aistSkel.h $$f ; /
-         done
+         done ;
+         @for f in $(DUMMY_STUBS) ; do /
+           echo cp OpenRTM-aistSkel.h $$f ; /
+           cp OpenRTM-aistSkel.h $$f ; /
+         done ;

```

#3 - 2012/04/08 22:50 - n-ando

ExecutionContextProfile.cpp内でCORBA\_SeqUtil::push\_back を使用している所がコンパイルエラーになる。とりあえず、意味はおなじになるように以下のように修正。

```

--- src/lib/rtm/ExecutionContextProfile.cpp      (revision 2335)
+++ src/lib/rtm/ExecutionContextProfile.cpp      (working copy)
@@ -270,7 +270,12 @@
+
+         return RTC::RTC_ERROR;
+     }
+     Guard guard(m_profileMutex);
+ #ifndef ORB_IS_RTORB
+     CORBA_SeqUtil::push_back(m_profile.participants, rtobj._retn());
+ #else
+     CORBA_SeqUtil::push_back(m_profile.participants,
+                               RTC::RTOBJECT::_duplicate(rtobj.in()));
+ #endif
+     return RTC::RTC_OK;
+ }

```

#4 - 2012/04/08 22:51 - n-ando

RtORBでは bool != CORBA::Boolean か？

とりあえず、以下のように修正。

```

<pre>
Index: src/lib/rtm/ExecutionContextWorker.h

```

```

=====
--- src/lib/rtm/ExecutionContextWorker.h      (revision 2335)
+++ src/lib/rtm/ExecutionContextWorker.h      (working copy)
@@ -138,7 +138,7 @@

```

```

+
+     *
+     * @endif
+     */
-     bool isRunning(void);
+     CORBA::Boolean isRunning(void);
+
+     /*!
+     * @if jp
</pre>

```

#5 - 2012/04/09 12:27 - n-ando

/FileNameservice.cpp内で、CORBA::Object\_var にRTOBJECT代入するとエラーになる。以下のように変更。

```

Index: src/ext/local_service/nameservice_file/FileNameservice.cpp

```

```

=====
--- src/ext/local_service/nameservice_file/FileNameservice.cpp      (revision 2335)
+++ src/ext/local_service/nameservice_file/FileNameservice.cpp      (working copy)
@@ -404,8 +404,13 @@

```

```

        NamingAction::preBind(RTC::RTOBJ_impl* rtobj, coil::vstring& name_)
        {
            coil::vstring name = name_;
+##ifndef ORB_IS_RTORB
            CORBA::Object_var objref =
                RTC::RTOBJ::_duplicate(rtobj->getObjRef());
+##else
+            RTC::RTOBJ_var objref =
+                RTC::RTOBJ::_duplicate(rtobj->getObjRef());
+##endif // ORB_IS_RTORB
            CORBA::String_var ior =
                RTC::Manager::instance().getORB()->object_to_string(objref);
            coil::vstring ns_info;

```

#6 - 2012/04/09 12:28 - n-ando

toRepositoryIdが使えない。代わりに、toRepositoryIdOfObjectを使用するように変更。

Index: src/ext/sdo/observer/ComponentObserverConsumer.cpp

```

=====
--- src/ext/sdo/observer/ComponentObserverConsumer.cpp      (revision 2335)
+++ src/ext/sdo/observer/ComponentObserverConsumer.cpp      (working copy)
@@ -684,7 +684,12 @@
        {
            RTC::SdoServiceConsumerFactory& factory
                = RTC::SdoServiceConsumerFactory::instance();
+##ifndef ORB_IS_RTORB
            factory.addFactory(CORBA_Util::toRepositoryId<OpenRTM::ComponentObserver>(),
+##else
+            factory.addFactory(CORBA_Util::
+                                toRepositoryIdOfObject<OpenRTM::ComponentObserver>(),
+##endif
                                ::coil::Creator<
                                    ::RTC::SdoServiceConsumerBase,
                                    ::RTC::ComponentObserverConsumer>,
                                ::coil::Destructor<
                                    ::RTC::SdoServiceConsumerBase,

```

#7 - 2012/04/09 12:34 - n-ando

ローカルにIDLを持っている、ObserverやLogicalTimeECなどのモジュールで、RtORBのIDLコンパイラが生成するスタブのヘッダが\*.hとなるため、既存のファイルとぶつかる。生成ファイルはidlディレクトリに入れるように変更。

Index: src/ext/sdo/observer/Makefile.am

```

=====
--- src/ext/sdo/observer/Makefile.am      (revision 2335)
+++ src/ext/sdo/observer/Makefile.am      (working copy)
@@ -8,7 +8,7 @@
AUTOMAKE_OPTIONS = 1.9

IDL_C = @IDL_C@
-IDLFLAGS = @IDL_FLAGS@ -I$(top_srcdir)/src/lib/rtm/idl
+IDLFLAGS = @IDL_FLAGS@ -I../$(top_srcdir)/src/lib/rtm/idl
LIBS = @LIBS@

AM_CPPFLAGS=-I$(top_srcdir)/src/lib -I$(top_srcdir)/src/lib/coil/include -I$(top_srcdir)/src/lib/rtm/idl
@@ -24,16 +24,20 @@
.idlSkel.cpp:
    echo $(PROVIDER_OBJ)
-    $(IDL_C) $(IDLFLAGS) $<
+    mkdir -p idl
+    cp $< idl/
+    (cd idl ; $(IDL_C) $(IDLFLAGS) $< )
    PYTHONPATH="$(top_srcdir)/build:$(PYTHONPATH)" /
    PATH="$(top_srcdir)/utils/rtm-config:$(PATH)" /

```

```

- $(WRAPPER) $(WRAPPER_FLAGS) --idl-file=$<
+ $(WRAPPER) $(WRAPPER_FLAGS) --idl-file=$< --include-dir=idl/

.idlStub.cpp:
- $(IDLC) $(IDLFLAGS) $<
+ mkdir -p idl
+ cp $< idl/
+ (cd idl ; $(IDLC) $(IDLFLAGS) $< )
  PYTHONPATH="$(PYTHONPATH):$(top_srcdir)/build" /
  PATH="$(top_srcdir)/utils/rtm-config:$(PATH)" /
- $(WRAPPER) $(WRAPPER_FLAGS) --idl-file=$<
+ $(WRAPPER) $(WRAPPER_FLAGS) --idl-file=$< --include-dir=idl/

```

```

#-----
# Source list
@@ -41,7 +45,7 @@

```

```
IDL_SOURCES = ComponentObserver.idl
```

```

OBSERVER_DIST_SRC = ComponentObserverConsumer.cpp
-OBSERVER_NODIST_SRC = $(IDL_SOURCES:.idl=Stub.cpp)
+OBSERVER_NODIST_SRC = idl/$(IDL_SOURCES:.idl=Stub.cpp)
OBSERVER_SRC = $(OBSERVER_DIST_SRC) $(OBSERVER_NODIST_SRC)

```

```

OBSERVER_DIST_H = $(OBSERVER_DIST_SRC:.cpp=.h)
@@ -143,6 +147,7 @@

```

```

clean_skelstub:
  rm -f *Skel.h *Skel.cpp
  rm -f *Stub.h *Stub.cpp
+ rm -rf idl
  rm -f *.hh *SK.cc
  rm -f *C.cpp *C.h *S.cpp *S.h *_T.cpp *_T.h *.inl

```

さらに、インクルードパスが変更されるため、skel\_wrapper.pyに変更を加えて、生成ファイルのインクルードパスのみ、--include-dirオプションで書き換えられるようにした。

Index: utils/rtm-skelwrapper/skel\_wrapper.py

```
=====
```

```
--- utils/rtm-skelwrapper/skel_wrapper.py (revision 2335)
```

```
+++ utils/rtm-skelwrapper/skel_wrapper.py (working copy)
```

```
@@ -86,7 +86,7 @@
```

```
*
*/
```

```

-#include "[include_dir][skel_h]"
+#include "[skel_h]"

```

```

#if defined ORB_IS_TAO
# include "[include_dir][basename]C.cpp"
@@ -139,7 +139,7 @@

```

```
[config_inc]
```

```

#if defined ORB_IS_TAO
-# include "[include_dir][basename]C.h"
+# include "[basename]C.h"
  #elif defined ORB_IS_OMNIORB
  # if defined(WIN32) || defined(_WIN32) || defined(__WIN32__) || defined(__NT__)
  # undef USE_stub_in_nt_dll
@@ -175,7 +175,7 @@

```

```
*
*/
```

```

-#include "[include_dir][stub_h]"
+#include "[stub_h]"

```

```

#if defined ORB_IS_TAO
# include "[include_dir][basename]C.cpp"

```

#8 - 2012/04/09 12:40 - 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