

OpenRTM-aist (C++) - バグ #707

OpenRTM-aist(C++)-v1.0.0-RC1 PublisherNew.cppの不具合

2009/06/15 13:39 - 匿名ユーザー

ステータス:	終了	開始日:	2009/06/05
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:			
説明			
PublisherNew.cppのpushAll()関数・pushFifo()関数・pushSkip()関数で、BUFFER_FULLの際、出力データが抜ける不具合あり。			

履歴

#1 - 2009/06/15 13:45 - 匿名ユーザー

- ファイル PublisherNew.cpp.diff を追加
- 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

●テスト条件

- exsampleのSeqIOよりSeqIn/Outを使用し、以下の手順で操作した。
 - SeqOutから連番1〜出力するようコンパイル。
 - SeqOut/InをEclipseに配置、ポート接続、All Activate。(type:new、m_skipn:9)
 - SeqInのDeactivate -> Activate を繰り返す。
この時、SeqIn出力の連番表示を確認。

■RTCログ名称の不具合現象

- ・RTCログのログ名称が"PublisherPeriodic"になっている。

●原因

- ・コンストラクタで、rtclog("PublisherPeriodic") となっていた。

●対処

- ・rtclog("PublisherNew") に修正した。

■write()関数の不具合現象

- ・SeqInのDeactivate -> Activate時に、BUFFER_FULLから復帰した後のデータが、ひとつ前のデータを表示している。

●原因

- ・write()時のデータ書き込みを行っていなかった。

●対処

- ・PublisherNew.cpp write()関数で、BUFFER_FULL時にm_buffer->write()を追加。

★修正前

```
if (m_retcode BUFFER_FULL) {
RTC_DEBUG(("write(): InPort buffer is full."));
return BUFFER_FULL;
}
```

★修正後

```
if (m_retcode BUFFER_FULL) {
RTC_DEBUG(("write(): InPort buffer is full."));
//追加部位 --->
CdrBufferBase::ReturnCode ret(m_buffer->write(data, sec, usec));
m_task->signal();
//追加部位 <---
return BUFFER_FULL;
}
```

●動作確認

- ・テスト条件で操作し、データ表示抜けがない事を確認した。

■pushAll()関数の不具合現象

- SeqInのDeactivate -> Activate時に、BUFFER_FULLとなったデータが表示から抜けている。
- 正常終了時にPORT_ERRORになっている。

●原因

- put()エラー時に、m_buffer->advanceRptr()を実行している為、読み出しポインタがずれていた。
- return PORT_OK が抜けていた。

●対処

- PublisherNew.cpp pushAll()関数で、put()エラー時にreturnするよう修正。

★修正前

```
{
RTC_TRACE(("pushAll()"));
try {
while (m_buffer->readable() > 0) {
cdrMemoryStream& cdr(m_buffer->get());
ReturnCode ret(m_consumer->put(cdr));

if (ret == SEND_FULL)
{
return SEND_FULL;
}
}
return CONNECTION_LOST;
}
return PORT_ERROR;
}

m_buffer->advanceRptr();
}
catch (...)
```

★修正後（不具合調査当初からの修正分全て）

```
{
RTC_TRACE(("pushAll()"));
try {
while (m_buffer->readable() > 0) {
cdrMemoryStream& cdr(m_buffer->get());
ReturnCode ret(m_consumer->put(cdr));

if (ret == SEND_FULL)
{
return SEND_FULL;
}
//追加部位 --->
else if (ret != PORT_OK)
{
return ret;
}
//追加部位 <---
{
return CONNECTION_LOST;
}
return PORT_ERROR;
}

m_buffer->advanceRptr();
}
//追加部位 --->
return PORT_OK;
//追加部位 <---
}
catch (...)
```

●動作確認

- テスト条件で操作し、データ表示抜けがない事を確認した。

■pushFifo()関数の不具合現象

- SeqInのDeactivate -> Activate時に、BUFFER_FULLとなったデータが表示から抜けている。

●原因

- put()エラー時に、m_buffer->advanceRptr()を実行している為、読み出しポインタがずれていた。

●対処

- PublisherNew.cpp pushFifo()関数で、put()エラー時にreturnするよう修正。

```

★修正前 {
RTC_TRACE(("pushFifo()"));
try {
cdrMemoryStream& cdr(m_buffer->get());
ReturnCode ret(m_consumer->put(cdr));

if (ret == SEND_FULL)
{
return SEND_FULL;
}
{
return CONNECTION_LOST;
}
return PORT_ERROR;
}

m_buffer->advanceRptr();

return ret;
}
catch (...)

```

```

★修正後 {
RTC_TRACE(("pushFifo()"));
try {
cdrMemoryStream& cdr(m_buffer->get());
ReturnCode ret(m_consumer->put(cdr));

if (ret == SEND_FULL)
{
return SEND_FULL;
}
//追加部位 --->
else if (ret != PORT_OK)
{
return ret;
}
//追加部位 <---
{
return CONNECTION_LOST;
}
return PORT_ERROR;
}

m_buffer->advanceRptr();

return ret;
}
catch (...)

```

●動作確認

- ・テスト条件で操作し、データ表示抜けがない事を確認した。

■pushSkip()関数の不具合現象

- ・データがスキップされて表示しない。

●原因

- ・スキップ処理が入っていない。

●対処

- ・PublisherNew.cpp pushSkip()関数の、ロジックを見直した。

```

★修正前 {
RTC_TRACE(("pushSkip()"));
try {
cdrMemoryStream& cdr(m_buffer->get());
m_buffer->advanceRptr(m_skipn);
return m_consumer->put(cdr);
}
catch (...) {
return CONNECTION_LOST;
}
return PORT_ERROR;
}

```

```

★修正後 {
//修正部位 --->
static int leftskip; // 残りのスキップ数

RTC_TRACE(("pushSkip()"));
try
{
    ReturnCode ret(PORT_OK);
    int preskip(m_buffer->readable() + leftskip);
    int loopcnt(preskip/(m_skipn +1));
    int postskip(m_skipn - leftskip);
    for (int i(0); i < loopcnt; ++i)
    {
        m_buffer->advanceRptr(postskip);
        const cdrMemoryStream& cdr(m_buffer->get());
        ret = m_consumer->put(cdr);
        if (ret != PORT_OK)
        {
            m_postskip); // 読み出しポインタを戻す
            return ret;
        }
        postskip = m_skipn +1;
    }
    >advanceRptr(m_buffer->readable());
    if (loopcnt == 0)
    {
        // Not put
        leftskip = preskip % (m_skipn +1);
    }
    else
    {
        if ( m_retcode != PORT_OK ) // 前回は異常終了
        {
            // put NG after
            leftskip = 0;
        }
        else
        {
            // put OK after
            leftskip = preskip % (m_skipn +1);
        }
    }
    return ret;
}
//修正部位 <---
}
catch (...)
{
    return CONNECTION_LOST;
}
return PORT_ERROR;
}

```

●動作確認

- ・スキップ数単位でデータ（10の倍数）が表示されることを確認した。
- ・テスト条件で操作し、データ表示抜けがない事を確認した。

以上

#2 - 2009/06/15 13:47 - 匿名ユーザー

- ステータスを 解決 から 終了 に変更

ファイル

PublisherNew.cpp.diff	2.17 KB	2009/06/15	匿名ユーザー
-----------------------	---------	------------	--------