

## 意味のない例外処理を削除する

2016/12/19 15:59 - n-miyamoto

ステータス:	終了	開始日:	2016/12/19
優先度:	通常	期日:	
担当者:	n-miyamoto	進捗率:	100%
カテゴリ:		予定工数:	0.00時間
対象バージョン:	RELEASE_1_2_0		

**説明**

以下のように意味のない箇所で例外処理を行っている部分があるため修正する。

```

try:
    return self._sdoSvcProfiles
except:
    self.__rtcout.RTC_ERROR(OpenRTM_aist.Logger.print_exception())
    raise SDOPackage.InternalError("get_service_profiles()")

```

## 関係しているリビジョン

リビジョン 768 - 2016/12/19 16:01 - miyamoto

[compat,bugfix,-&gt;RELENG\_1\_2] fixed bug. refs #3760

## 履歴

#1 - 2016/12/19 16:04 - n-miyamoto

- 進捗率 を 0 から 50 に変更

以下の部分が該当するため、修正を行った。

- SdoOrganization

```

try:
    guard = OpenRTM_aist.ScopedLock(self._org_mutex)
    self._orgProperty = org_property
    return True
except:
    self.__rtcout.RTC_ERROR(OpenRTM_aist.Logger.print_exception())
    raise SDOPackage.InternalError("add_organization_property()")

```

```

try:
    self._varOwner = sdo
    return True
except:
    self.__rtcout.RTC_ERROR(OpenRTM_aist.Logger.print_exception())
    raise SDOPackage.InternalError("set_owner()")

```

```

try:
    return self._memberList
except:
    self.__rtcout.RTC_ERROR(OpenRTM_aist.Logger.print_exception())
    raise SDOPackage.InternalError("get_members()")

```

```

try:
    self._memberList = sdos
    return True
except:
    self.__rtcout.RTC_ERROR(OpenRTM_aist.Logger.print_exception())
    raise SDOPackage.InternalError("set_members()")

```

```

try:
    self._dependency = dependency
    return True
except:
    self.__rtcout.RTC_ERROR(OpenRTM_aist.Logger.print_exception())
    raise SDOPackage.InternalError("set_dependency(): Unknown.")

```

- SdoConfiguration

```
try:
    guard = OpenRTM_aist.ScopedLock(self._dprofile_mutex)
    self._deviceProfile = dProfile
except:
    self._rtcout.RTC_ERROR(OpenRTM_aist.Logger.print_exception())
    raise SDOPackage.InternalError("Unknown Error")

try:
    if not self._configsets.haveConfig(config_id):
        self._rtcout.RTC_ERROR("No such ConfigurationSet")
        raise SDOPackage.InternalError("No such ConfigurationSet")
except:
    self._rtcout.RTC_ERROR(OpenRTM_aist.Logger.print_exception())
    raise SDOPackage.InternalError("Unknown exception")
```

- RTOBJECT

```
try:
    return self._portAdmin.getPortServiceList()
except:
    self._rtcout.RTC_ERROR(OpenRTM_aist.Logger.print_exception())
```

```
try:
    return self._sdoOwnedOrganizations
except:
    self._rtcout.RTC_ERROR(OpenRTM_aist.Logger.print_exception())
    raise SDOPackage.NotAvailable("NotAvailable: get_owned_organizations")
```

```
try:
    return self._profile.instance_name
except:
    self._rtcout.RTC_ERROR(OpenRTM_aist.Logger.print_exception())
    raise SDOPackage.InternalError("get_sdo_id()")
```

```
try:
    return self._profile.description
except:
    self._rtcout.RTC_ERROR(OpenRTM_aist.Logger.print_exception())
    raise SDOPackage.InternalError("get_sdo_type()")
```

```
try:
    return self._SdoConfigImpl.getDeviceProfile()
except:
    self._rtcout.RTC_ERROR(OpenRTM_aist.Logger.print_exception())
    raise SDOPackage.InternalError("get_device_profile()")
```

```
try:
    return self._sdoSvcProfiles
except:
    self._rtcout.RTC_ERROR(OpenRTM_aist.Logger.print_exception())
    raise SDOPackage.InternalError("get_service_profiles()")
```

```
try:
    return self._sdoSvcProfiles[index].service
except:
    self._rtcout.RTC_ERROR(OpenRTM_aist.Logger.print_exception())
    raise SDOPackage.InternalError("get_service()")
```

```
try:
    return self._SdoConfig
except:
    self._rtcout.RTC_ERROR(OpenRTM_aist.Logger.print_exception())
    raise SDOPackage.InternalError("get_configuration()")
```

```
try:
    return self._sdoOrganizations
except:
    self._rtcout.RTC_ERROR(OpenRTM_aist.Logger.print_exception())
    raise SDOPackage.InternalError("get_organizations()")
```

```
try:
    return self._sdoStatus
except:
```

```
self._rtcout.RTC_ERROR(OpenRTM_aist.Logger.print_exception())
raise SDOPackage.InternalError("get_status_list()")
```

- Timer

```
try:
    self.join()
except:
    pass
```

- TimeValue

```
try:
    res = TimeValue()
except:
    res = OpenRTM_aist.TimeValue()
```

#2 - 2016/12/20 19:07 - n-miyamoto

- 進捗率を 50 から 70 に変更

以下の箇所の例外処理は無理やり変数を置き換えない限り網羅不可能なので削除

- SdoConfiguration

```
try:
    guard = OpenRTM_aist.ScopedLock(self._params_mutex)
    param = copy.copy(self._parameters)
    return param
except:
    self._rtcout.RTC_ERROR(OpenRTM_aist.Logger.print_exception())
    raise SDOPackage.InternalError("Configuration.get_configuration_parameters")
```

```
try:
    config = SDOPackage.ConfigurationSet("", "", [])
    toConfigurationSet(config, configset)
    return config
except:
    self._rtcout.RTC_ERROR(OpenRTM_aist.Logger.print_exception())
    raise SDOPackage.InternalError("Configuration::get_configuration_set()")
```

```
try:
    OpenRTM_aist.CORBA_SeqUtil.push_back(self._organizations, org)
except:
    self._rtcout.RTC_ERROR(OpenRTM_aist.Logger.print_exception())
    raise SDOPackage.InternalError("Configuration.add_organization")
```

```
try:
    OpenRTM_aist.CORBA_SeqUtil.erase(self._orgProperty.properties, index)
    return True
except:
    self._rtcout.RTC_ERROR(OpenRTM_aist.Logger.print_exception())
    raise SDOPackage.InternalError("remove_organization_property_value()")
```

#3 - 2017/02/01 09:18 - n-miyamoto

- ステータスを新規から解決に変更

- 進捗率を 70 から 100 に変更

#4 - 2017/03/19 08:15 - n-ando

- ステータスを解決から終了に変更