Problem

How to use CDB bulkData NT configuration in my code?

Solution

Primary the configuration in CDB for BulkData(NT) was ment to be used for configuring stream/flows that Receiver/Sender bulk data component is going to create at start up. In this case the names of the stream and flows are equal to the corresponding names in the CDB. We can say that the stream/flow name and the configuration name is the same.

With new bulk data it is possible to distinguis between stream/flow name and the configuration name. For this purpose it was added functionality to easy use the CDB configuration also directly from C++ API. In such a way we can define default configuration(s) for streams/flows that can be used in someones application/component.

In the new bulk data have been introduced 4 classes for configuring streams/flows:

- `SenderStreamConfiguration`
- `SenderFlowConfiguration`
- `ReceiverStreamConfiguration`
- `ReceiverFlowConfiguration`

Those classes contains member attributes per configuration. The values can be fed in different way: grammatically with setter methods, or from CDB using configuration parse.

The best is to look into an example how to retrieve a configuration = how to use the C++ API:
Let's assume that we have in CDB (alma branch) under folder `!DefaultCorrBDNTCfg` in file `!DefaultCorrBDNTCfg.xml` following configuration:

```
...
<ReceiverStream Name="DefaultCORRStreamCfg">
        <ReceiverFlow Name="DefaultCORRSpectralDataFlowCfg"
            cbReceiveProcessTimeoutSec="0.02"
        >
        </ReceiverFlow>
        <ReceiverFlow Name="DefaultCORRXYZFlowCfg"/>
  </ReceiverStream>
..
```

and we would like to use this configuration for our stream(s) and flow(s).
First we have to obtain the (XML) data, in the standard way from the CDB:

```
CDB::DAL_ptr dal_p = getContainerServices()->getCDB();
ACE_CString CDBpath="alma/";
CDBpath +=   "DefaultCorrBDNTCfg"; //node name in CDB like
char *xmlNode = dal_p->get_DAO(CDBpath.c_str());
```

So now we have the configuration XML string from CDB which should be fed to the parser:

```
AcsBulkdata::BulkDataConfigurationParser *parser_m = new AcsBulkdata::BulkDataConfigurationParser(
"DefaultCorrBDNTCfg" );
//if you want to have more than one BulkDataConfigurationParser (=more than one cfg node in CDB), should each
have unique name / could be component name.
parser_m->parseReceiverConfig(xmlNode);    //xmlNode can be also a XML string
```

Now, we have to obtain proper the configuration from the parser, in this case for ReceiverStream (`ReceiverStreamConfiguration`) and ReceiverFlow (`ReceiverFlowConfiguration`), but it is equivalent for the sender side (`SenderStreamConfiguration` / `SenderFlowConfiguration`)

```
ReceiverStreamConfiguration*  rcvStrCfg = parser_m->getReceiverStreamConfiguration("DefaultCORRStreamCfg");  //
get Receiver Stream cfg
// now we create receiver stream with the configuration
AcsBulkdata::BulkDataNTReceiverStream<MyCallback> *stream =  new AcsBulkdata::
BulkDataNTReceiverStream<MyCallback>("MyStream", *rcvStrCfg);
```

And similar we get the configuration for the ReceiverFlow:

```
ReceiverFlowConfiguration *rcvSpecDataFlowCfg =  parser_m->getReceiverFlowConfiguration("DefaultCORRStreamCfg",
"DefaultCORRSpectralDataFlowCfg");

BulkDataNTReceiverFlow * specDataFlow = stream->createFlow("SpectralDataFlow", *rcvSpecDataFlowCfg);
```

-- BogdanJeram - 11 Jan 2012

Related articles

- How can more people do development with ACS on the same machine without disturbing each other?
- Which ports are used by ACS?
- Problems connecting to ACS servers on a remote machine: bad /etc/hosts