

CN REST - Bug #7746

Node registration update fails when <contactSubject> spans multiple lines

2016-04-19 23:55 - Mark Servilla

Status:	In Progress	Start date:	2016-04-26
Priority:	Normal	Due date:	
Assignee:	Rob Nahf	% Done:	30%
Category:	d1_cn_noderegistry	Estimated time:	0.00 hour
Target version:			
Story Points:			
Description			
Node registration update fails when the string in the node registration document (attached) is on a separate line from the XML element tags. Apparently, parsing this field results in a subject string that does not match the LDAP record correctly, and the result is 401 not authorized exception:			
CN=Lisa Stillwell A15851,O=University of North Carolina at Chapel Hill,C=US,DC=cilogon,DC=org			
results in:			
<?xml version="1.0" encoding="UTF-8"?>			
CN=Lisa Stillwell A15851,O=University of North Carolina at Chapel Hill,C=US,DC=cilogon,DC=org is not a Registered Subject			
Correcting to:			
CN=Lisa Stillwell A15851,O=University of North Carolina at Chapel Hill,C=US,DC=cilogon,DC=org			
succeeds.			
Subtasks:			
Task # 7750: apply business rules on the CN that Subject strings will be stripped of le...			New

History

#1 - 2016-04-20 00:04 - Rob Nahf

- Category set to d1_cn_noderegistry

The exception thrown is called when cnIdentify.getSubjectInfo(subject) returns a NotFound.

Calling <https://cn-sandbox.test.dataone.org/cn/v2/accounts>,

I see Lisa's subject string listed as

```
CN=Lisa Stillwell A15851,O=University of North Carolina at Chapel Hill,C=US,DC=cilogon,DC=org
```

```
Lisa  
Stillwell  
true
```

without space characters leading the string.

In the Node document, after the new line, there are also 8 space characters preceding the non-whitespace part, so it is seeing the them as different subjects.

I think the following in the Node document would work as well.

CN=Lisa Stillwell A15851,O=University of North Carolina at Chapel Hill,C=US,DC=cilogon,DC=org

Subject elements need to accept whitespace (internally at least), so I wonder if we need to tighten up validation to disallow leading and trailing whitespace, assuming it's possible?

#2 - 2016-04-20 00:30 - Rob Nahf

- Status changed from New to In Progress

- % Done changed from 0 to 30

#3 - 2016-04-20 17:59 - Rob Nahf

I tested all three subject strings and all of these pass when we call with libclient's CNode.getSubjectInfo(subject):

newlines and leading spaces: "\n CN=Lisa Stillwell A15851,O=University of North Carolina at Chapel Hill,C=US,DC=cilogon,DC=org\n"
leading spaces: " CN=Lisa Stillwell A15851,O=University of North Carolina at Chapel Hill,C=US,DC=cilogon,DC=org"
trimmed: "CN=Lisa Stillwell A15851,O=University of North Carolina at Chapel Hill,C=US,DC=cilogon,DC=org"
newlines only: "\nCN=Lisa Stillwell A15851,O=University of North Carolina at Chapel Hill,C=US,DC=cilogon,DC=org\n"

current speculations:

Maybe the serialization / deserialization done by the client normalizes the string, where calling the identityService directly(?) does not?

Maybe the xslt parsing rules on subject and contactSubject are different?

#4 - 2016-04-20 21:27 - Rob Nahf

cn-rest calls the identity manager directly, bypassing serialization of the request. (implementation class is org.dataone.service.cn.impl.v2.SystemMetadataDaoMetacatImpl in cn_identity_manager). The remaining question is figuring out how serialization process trims whitespace.

#5 - 2016-04-21 21:27 - Rob Nahf

Using the TypeMarshaller, and TypeFactory.clone, I was not able to use serialization to remove leading newlines and other whitespace, as I thought it would.

It's unclear to me why calling cn.getSubjectInfo using a subject that has leading and trailing whitespace doesn't throw an exception when updateNode does...

Parking this issue for now. Below is code for a couple of the tests I did.

```
@Test
public void testSubjectWhitespace() throws ServiceFailure, ClientSideException, IOException {
    org.dataone.client.v1.CNode cn = D1NodeFactory.buildNode(org.dataone.client.v1.CNode.class, new DefaultHttpMultipartRestClient(), URI.create("https://cn-sandbox.test.dataone.org/cn"));
    Subject subject = TypeFactory.buildSubject("\n CN=Lisa Stillwell A15851,O=University of North Carolina at Chapel Hill,C=US,DC=cilogon,DC=org\n");
    try {
        cn.getSubjectInfo(null, subject);
    } catch (NotAuthorized | NotImplemented | NotFound | InvalidToken e) {
        System.out.println(e.getDescription());
        Assert.fail("Should not be not authorized");
    }
}

@Test
```

```

public void testNodeParsing() throws InstantiationException, IllegalAccessException, IOException, JiBXException
{
    org.dataone.service.types.v1.Node node = TypeMarshaller
        .unmarshalTypeFromFile(org.dataone.service.types.v1.Node.class, "/Users/rnahf/Downloads/DFC-node.xml");
    org.dataone.service.types.v1.Node serializedNode = TypeFactory.clone(node);
    List<Subject> sl = serializedNode.getContactSubjectList();
    for (Subject s: sl) {
        System.out.printf("{%s}\n",s.getValue());
    }

    TypeMarshaller.marshallTypeToFile(
        TypeFactory.buildSubject(
            "\n          CN=Lisa Stillwell A15851,O=University of North Carolina at Chapel Hill,C=US,DC=
cilogon,DC=org\n"),
        "/Users/rnahf/Downloads/DFC-subject.xml");
    Subject serializedSubject = TypeMarshaller.unmarshalTypeFromFile(Subject.class, "/Users/rnahf/Downloads/DF
C-subject.xml");
    System.out.printf("{%s}\n",serializedSubject.getValue());
}

```

Files

DFC-node.xml	1.21 KB	2016-04-19	Mark Servilla
--------------	---------	------------	---------------