The RSC Resource

When working with distributed testing nodes, tests may require additional resources such as CSV files, data files, or other external files. These resources must be transferred to the remote testing node before the test executes.

Because the resource may be stored at a different location on the remote node, a path that was hardcoded when the test was created may no longer be valid. This can cause the test to fail because the required file cannot be found.

Appvance IQ provides the **RSC DPL** to solve this problem. Resources can be declared in the test's `<resources>` section, and the RSC DPL can then be used to retrieve the actual path of a resource at runtime, regardless of where the resource is stored on the testing node.

Declaring Resources

Resources required by the test, including resources used by other DPLs, should be declared in the `<resources>` section.

For example, suppose a `HashDPL` requires a CSV file. The CSV file can exist anywhere on the local machine, but it must be included in the resources section so that Appvance IQ can transfer it to the remote testing node.

```
<resources>
    <data path="c:\mydata\data.csv"/>
</resources>
```

The `<data>` element represents a file resource.

Resources are assigned an index based on their order in the `<resources>` section. The first resource has index **0**.

For example:

```
<resources>
    <data path="c:\mydata\data.csv"/>
    <data path="c:\mydata\users.csv"/>
</resources>
```

In this example:

- `data.csv` is resource **0**
- `users.csv` is resource **1**

Using RSC with another DPL

Once a resource has been declared, the **RSC DPL** can be used to obtain its actual runtime path.

For example, if a `HashDPL` requires the path to a CSV file:

```
<DataSources>
    <dpl name="mydpl" type="HashDPL">
        <argument name="file"
                  dpl="rsc"
                  value="getDataByIndex"
                  index="0"/>
    </dpl>
</DataSources>
```

Here, the `file` argument of the `HashDPL` receives its value from the RSC DPL.

The following attributes are used:

Attribute	Description
`dpl="rsc"`	Specifies that the RSC DPL is being used.
`value="getDataByIndex"`	Specifies that the resource path should be retrieved by its index.
`index="0"`	Identifies the resource declared at index 0 in the `<resources>` section.

The value returned by RSC is the **actual path of the resource on the testing node**. This means the test does not need to know or depend on the path used by the remote node.

Using RSC in Test Steps

RSC can also be used directly in test step definitions.

For example, a Jython test step may require the path of a resource:

```
<run name="CheckInventory"
     module="JythonModule"
     testclass="testgen4web"
     method="run"
     langtype="jython">
    <argument name="frozenfile"
              dpl="rsc"
              value="getDataByIndex"
              index="0"/>
    <argument name="ownerinfo"
              dpl="values"
              value="getData"/>
</run>
```

In this example, the `frozenfile` argument receives the runtime path of resource **0**.

The test step can therefore access the file without knowing where Appvance IQ stored the resource on the remote testing node.

Recommended Usage

Use the following approach when a test depends on an external file:

1. Add the file to the `<resources>` section.
2. Determine its resource index based on its position in the section. Remember that indexing starts at **0**.
3. Use the RSC DPL with `getDataByIndex` whenever the file path is required.
4. Avoid hardcoding paths to resources that will be used on distributed testing nodes.

This approach allows the same test to run reliably across different testing nodes and environments without requiring changes to file paths.

Summary

The RSC DPL provides a runtime-safe way to access resources transferred to distributed testing nodes. Instead of relying on a hardcoded local or remote path, the test references the resource by its index in the `<resources>` section.

For example:

```
<resources>
    <data path="c:\mydata\data.csv"/>
</resources>
```

The resource can then be referenced anywhere a file path is required:

```
<argument name="file"
          dpl="rsc"
          value="getDataByIndex"
          index="0"/>
```

The RSC DPL resolves the resource to its actual location on the testing node at runtime.

Revision #4
Created 2026-07-31 15:09:37 UTC by Admin
Updated 2026-10-15 15:53:28 UTC by mmora