

文章
姚鑫 · 五月 17, 2021 阅读大约需分钟

第三章 执行测试

第三章 执行测试

示例: 执行测试

现在使用%UnitTest.Manager.RunTest执行单元测试。方法是:

1. 在包含单元测试的名称空间中打开终端;在本例中为用户。如果终端未在正确的命名空间中打开,请使用ZNC更改命名空间。
2. 将^UnitTestRoot全值设置为包含导出测试类目录的父级。

```
DHC-APP>Set ^UnitTestRoot="d:\Temp"
```

3. 使用方法%UnitTest.Manager.RunTest执行测试。

```
DHC-APP>do ##class(%UnitTest.Manager).RunTest("test")
```

4. IRIS从XML文件加载测试类,编译类,执行测试,从服务器删除测试代码,并向终端发送报告。

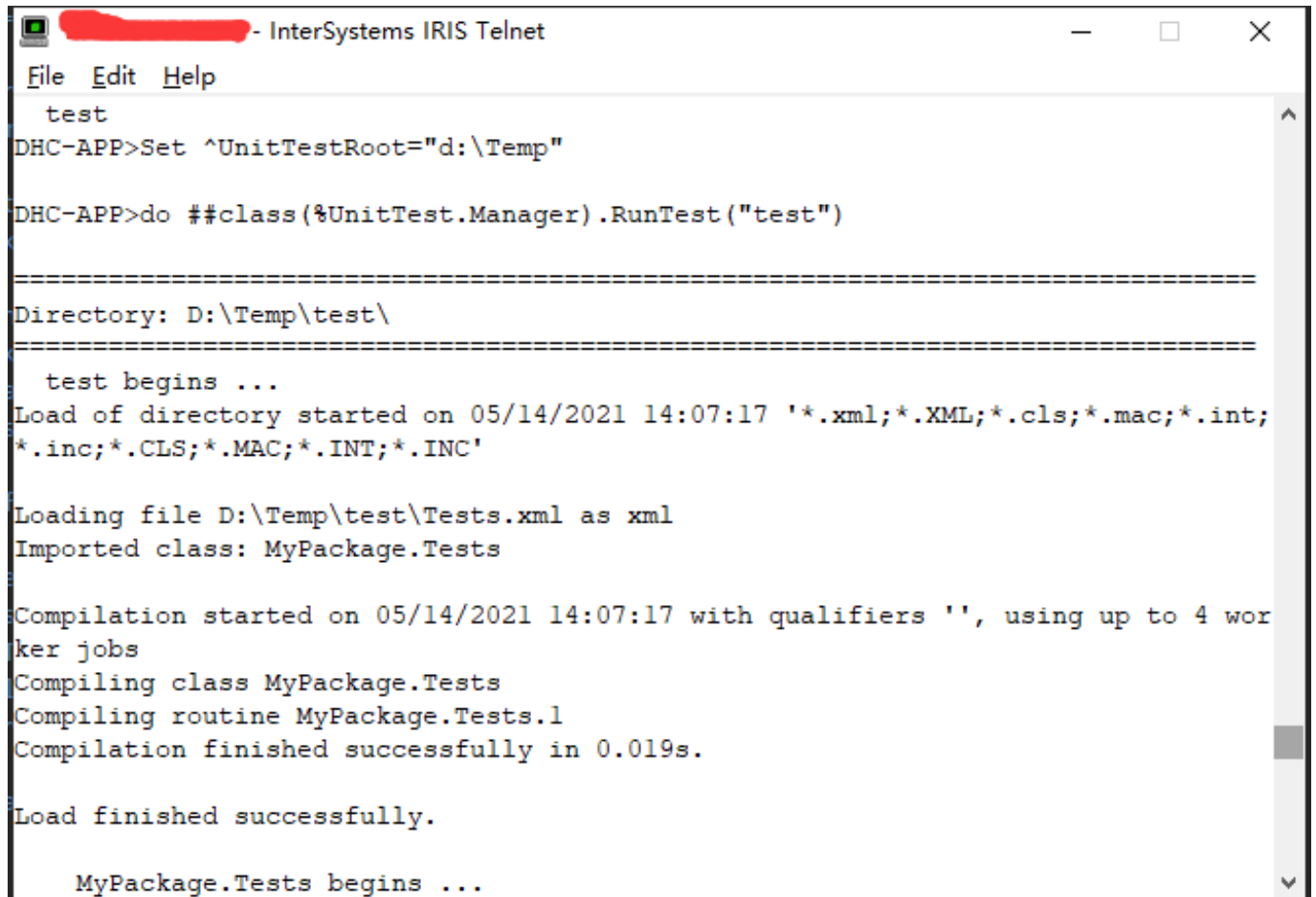
```
HC-APP>do ##class(%UnitTest.Manager).RunTest("test")
```

```
=====  
Directory: D:\Temp\test\  
=====  
test begins ...  
Load of directory started on 05/14/2021 14:07:17 '*.xml;*.XML;*.cls;*.mac;*.int;*.inc  
;*.CLS;*.MAC;*.INT;*.INC'  
  
Loading file D:\Temp\test\Tests.xml as xml  
Imported class: MyPackage.Tests  
  
Compilation started on 05/14/2021 14:07:17 with qualifiers '', using up to 4 worker j  
obs  
Compiling class MyPackage.Tests  
Compiling routine MyPackage.Tests.1  
Compilation finished successfully in 0.019s.  
  
Load finished successfully.  
  
MyPackage.Tests begins ...  
TestAdd() begins ...  
AssertEquals:Test Add(2,2)=4 (passed)  
AssertNotEquals:Test Add(2,2)!=5 (passed)  
LogMessage:Duration of execution: .000061 sec.
```

```
TestAdd passed
MyPackage.Tests passed
test passed
```

Use the following URL to view the result:

```
http://172.18.18.159:52773/csp/sys/%25UnitTest.Portal.Indices.cls?Index=3&$NAMESPACE=
DHC-APP
All PASSED
```



```
InterSystems IRIS Telnet
File Edit Help
test
DHC-APP>Set ^UnitTestRoot="d:\Temp"

DHC-APP>do ##class(%UnitTest.Manager).RunTest("test")

=====
Directory: D:\Temp\test\
=====

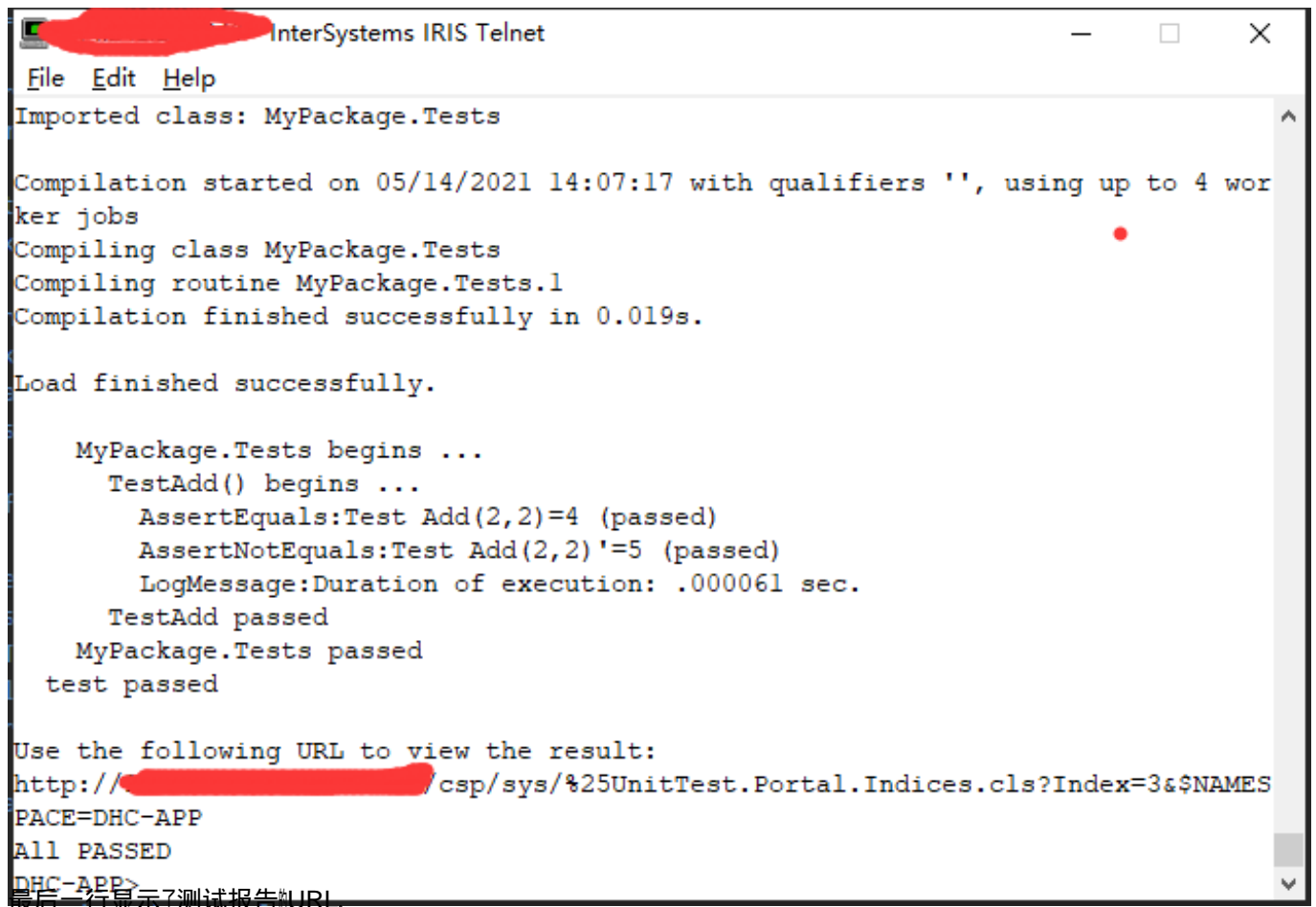
test begins ...
Load of directory started on 05/14/2021 14:07:17 '*.xml;*.XML;*.cls;*.mac;*.int;
*.inc;*.CLS;*.MAC;*.INT;*.INC'

Loading file D:\Temp\test\Tests.xml as xml
Imported class: MyPackage.Tests

Compilation started on 05/14/2021 14:07:17 with qualifiers '', using up to 4 wor
ker jobs
Compiling class MyPackage.Tests
Compiling routine MyPackage.Tests.1
Compilation finished successfully in 0.019s.

Load finished successfully.

MyPackage.Tests begins ...
```



```
InterSystems IRIS Telnet
File Edit Help
Imported class: MyPackage.Tests

Compilation started on 05/14/2021 14:07:17 with qualifiers '', using up to 4 worker jobs
Compiling class MyPackage.Tests
Compiling routine MyPackage.Tests.1
Compilation finished successfully in 0.019s.

Load finished successfully.

MyPackage.Tests begins ...
  TestAdd() begins ...
    AssertEquals:Test Add(2,2)=4 (passed)
    AssertNotEquals:Test Add(2,2) !=5 (passed)
    LogMessage:Duration of execution: .000061 sec.
  TestAdd passed
MyPackage.Tests passed
test passed

Use the following URL to view the result:
http://[redacted]/csp/sys/%25UnitTest.Portal.Indices.cls?Index=3&$NAMES
PACE=DHC-APP
All PASSED
DHC-APP>
```

最后一行显示了测试报告的URL。

注意:以这种方式运行测试会在它们执行后从InterSystems IRIS中删除它们。如果在执行测试后返回到Atelier查看测试，将看到一个指示，表明Atelier中可见文件与服务器不同步。可以点击重新编译该类，以将代码添加到服务器。

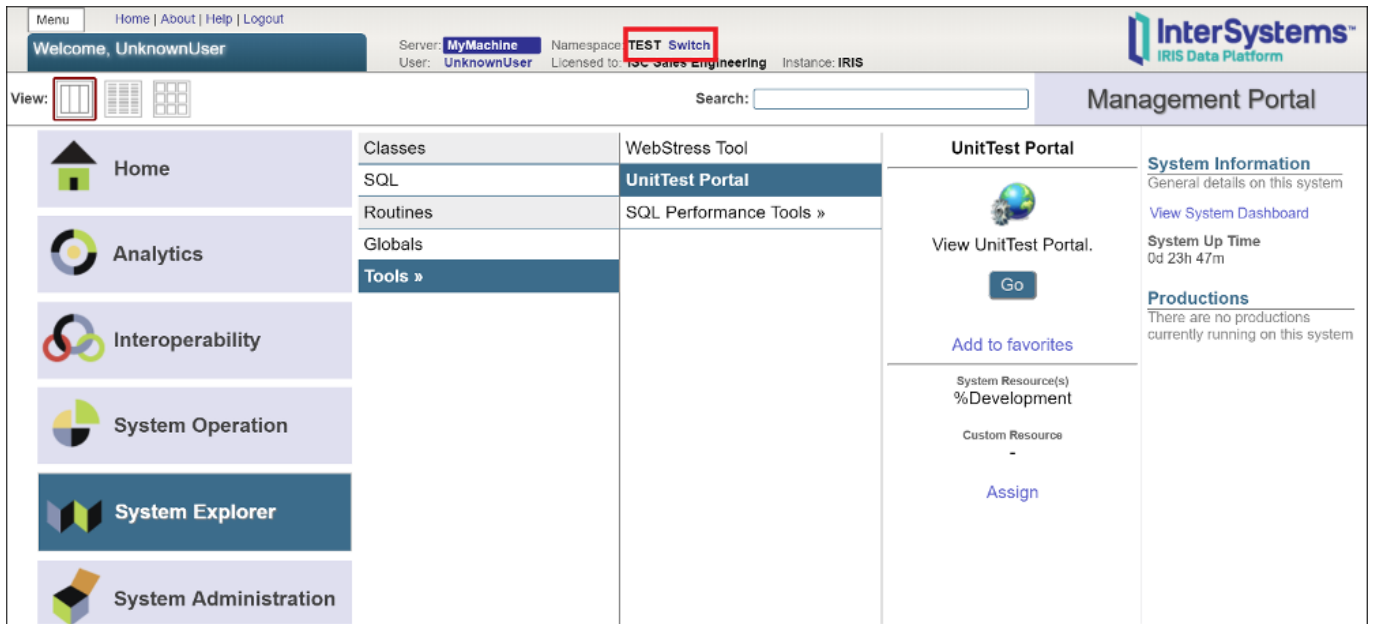
如果使用的是.cls文件而不是XML文件，则必须向RunTest提供/loadudl限定符。

```
USER>do ##class(%UnitTest.Manager).RunTest("mytests", "/loadudl")
```

示例: UnitTest Portal

运行单元测试将生成测试报告。InterSystems IRIS提供了一个用于查看报告的UnitTest门户。报告按命名空间组织。

可以使用系统资源管理器System Explorer > Tools > UnitTest Portal导航到UnitTest门户。如有必要，请切换到用户命名空间。



示例: 在单元测试门户中查看报告

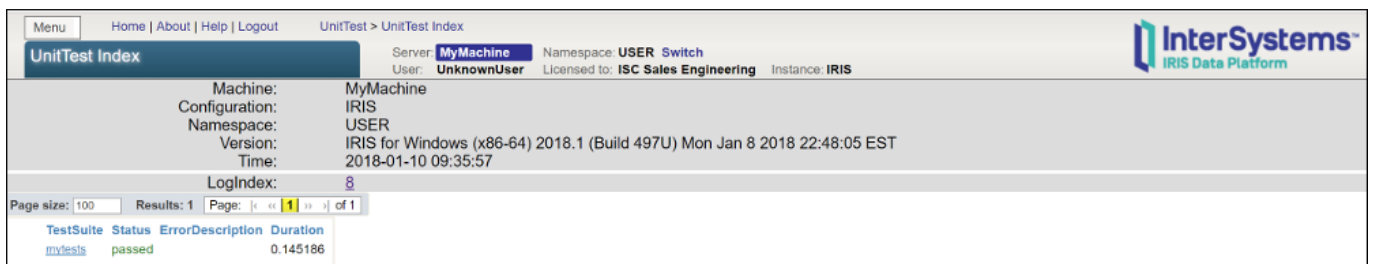
门户将测试结果组织成系列报告。每个测试报告将测试结果组织到一系列超链接页面中。按照链接查找越来越具体的信息。

第一页提供了所有测试套件的摘要。在这种情况下，所有测试套件都通过了。



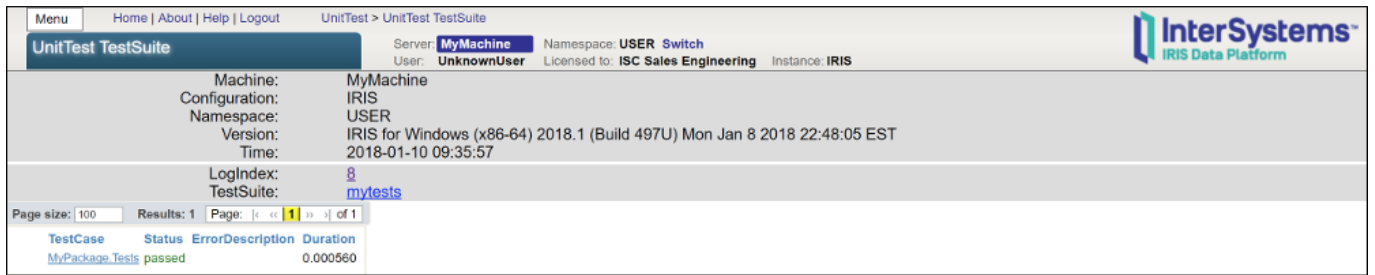
单击要查看的报告ID列中的ID号。

第二个页面显示每个测试套件的结果。在本例中，mytest是测试套件，并且通过了测试。



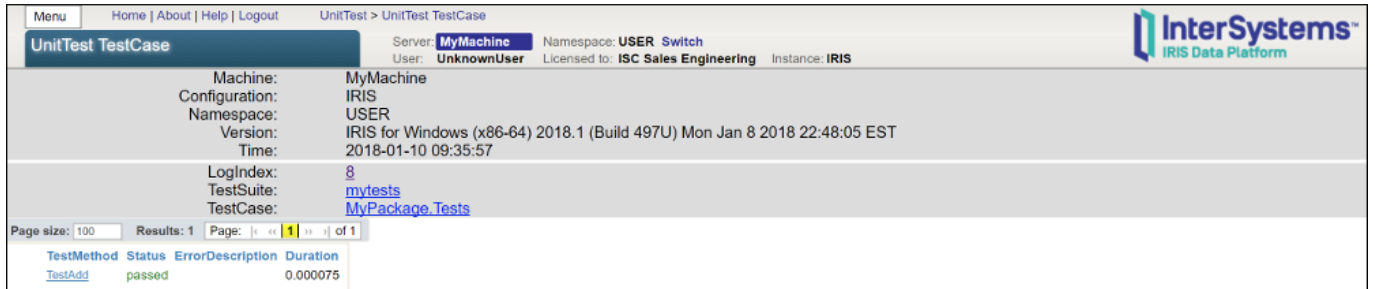
单击 mytests。

第三个页面显示每个测试用例的结果。在本例中，通过了单个测试用例MyPackage.Tests。



单击 MyPackage.Tests

第四页显示了通过测试方法得出的结果，这里通过了单个测试方法TestAdd。



单击 TestAdd。

最后一页显示测试方法中使用的每个AssertX宏的结果。在本例中，AssertEquals和AssertNotEquals都通过了。



设置和拆卸

%UnitTest.TestCase类提供的方法可用于在一个测试或一组测试执行之前设置测试环境，然后在测试完成拆除该环境。以下是对这些方法的说明：

方法	描述
OnBeforeAllTests	在测试类中的任何测试方法执行之前执行一次。可以设置测试环境。
OnAfterAllTests	在测试类中的所有测试方法执行后执行一次。可以破坏测试环境。
OnBeforeOneTest	在测试类中的每个测试方法执行之前立即执行。
OnAfterOneTest	在文本类中的每个测试方法执行后立即执行。

示例：向测试类添加Setup和Tear Down方法

在本例中，将添加一个名为TestEditContact的测试方法。此方法验证MyPackage.Contact类的ContactType属性是否限制为“Personal”或“Business”。添加了一个OnBeforeAllTests方法，该方法在测试执行之前准备数据库。还可以添加一个OnAfterAllTests方法，该方法在测试执行后还原数据库状态。

1. 在Studio中打开MyPackage.Tests(可能要从^UnitTestRoot目录导入它)。
2. 添加OnBeforeAllTests和OnAfterAllTests方法。

```
Method OnBeforeAllTests() As %Status
{
    Do ##class(MyPackage.Contact).Populate(1)
    Return $$$OK
}
```

```
Method OnAfterAllTests() As %Status
{
    Do ##class(MyPackage.Contact).%KillExtent()
    Return $$$OK
}
```

OnBeforeAllTests方法使用单个Contact实例填充数据库。OnAfterAllTests方法从数据库中删除所有Contact实例。

3. 现在将TestEditContact测试方法添加到MyPackage.Tests:

```
Method TestEditContact()
{
    set contact=##class(MyPackage.Contact).%OpenId(1)
    set contact.Name="Rockwell,Norman"
    set contact.ContactType="Friend"
    Do $$$AssertStatusNotOK(contact.%Save(),"ContactType = Friend")
    Set contact.ContactType="Personal"
    Do $$$AssertStatusOK(contact.%Save(),"ContactType = Personal")
}
```

该方法在两种测试执行%Save on

Contact返回的状态值: 为ContactType分配无效值之后和为ContactType分配有效值之后。

4. 将测试导出到c: /unittest /mytest, 覆盖现有的Tests.xml。

源码

[#SQL](#) [#Caché](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)

源 URL: <https://cn.community.intersystems.com/post/%E7%AC%AC%E4%B8%89%E7%AB%A0-%E6%89%A7%E8%A1%8C%E6%B5%8B%E8%AF%95-0>