

SIT Information

Friday, May 01, 2015 1:51 PM

SIT (A.K.A. ITest/WSTest)

SIT Test Creation

<http://is.factset.com/ISIntegrations/>

The main page shows a list of last test runs at the step level.

SIT API

- Dashboard: <http://is.factset.com/Core/Dashboard#/applications>
- Example Result: <http://is.factset.com/ISIntegrations/api/results/ResultById/194>
- Example Test run detail: <http://is.factset.com/ISIntegrations/#/16/TestRunDetail/374>

Getting steps by product key

- HTTP GET
- <http://is.factset.com/ISIntegrations/API/step/byProduct/16>
- returns the three steps entered for the "IS Core" product from the same product tree used by Justifier
- Getting steps by Plan and Case are not yet implemented

Getting full step detail

- HTTP GET
- <http://is.factset.com/ISIntegrations/API/step/byId/1> - returns the full detail of the step

Running a step

- HTTP POST
- <http://is.factset.com/ISIntegrations/API/run>
- The JSON to post is in a RunRequest object

```
{
  Environment: "string:required",
  Product: "int:nullable",
  Plan: "int:nullable",
  Case: "int:nullable",
  Step: "array[int]:nullable"
}
```

Example:

```
{
  "Environment": "Development",
  "Product": 16,
  "Plan": 77423
}
```

Or

```
{
  "Environment": "Development",
  "Product": 3311,
```

```
"Case": 115230,  
}
```

Or

```
{  
  "Environment": "Development",  
  "Product": 16,  
  "Step": [1,2]  
}
```

Are all examples of valid ways to make a request.

- Environment values: "CoreDevelopment", "Development", "QualityAssurance", "Staging", "Production"
- The other fields are all optional but you must provide a value in at least one them or there will be nothing to run
- Running a step is async so the success message you get back just tells you that the step was queued to be run

Return

POST will return "true".

Recent Test Results

<http://is.factset.com/Cockpit/pages/ListBuilder.html?entity=SITRunResults&view=recent>

SIT Queue UI

<http://is.factset.com/cockpit#/queues>

SIT Demo

Wednesday, May 06, 2015 1:05 PM

Is.factset.com/Isintegrations

<http://is.factset.com/ISIntegrations/>

SIT RPD Product

Upside down

Group Body Types

Status only time limit.

Response time tests - store in Monster

SIT QA Plugin

Tuesday, May 05, 2015 8:30 AM

SITQAPLugin

Source code: <https://gitlab.factset.com/app-qa-automation/SITQAPLugin>

Plugin Deployment Locations

PROD:

\\TELLUSGRIDB01\QueuePlugins\TELLUS_SIT_PLUGIN

DEV:

\\TELLUSGRIDB03\QueuePlugins\TELLUS_SIT_PLUGIN

To install plugin, simply copy to those directories.

SITQAPLUGIN Logging

SITQAPLUGIN log: <\\tellusgridb03\Tellus\SITQAPLugin\logs\SITQAPLugin.log>

SIT log: \\TELLUSGRIDB03\QueuePlugins\TELLUS_SIT_PLUGIN\PluginManager.log

SIT Documentation Ken

Tuesday, September 15, 2015 9:08 AM

Create a profile first

- **SIT down and stay a while--You've Got SIT todo** (from SIT Manual)
 - SIT is located at <http://is.factset.com/ISIntegrations/>
 - What it does
 - Web service output - exact matches, JSON schema checks, XML Schema checks, regEx matches, status only, duration
 - REST - REST Support is underway the plan is to support general REST and ISAPI based REST, more on this when it is available
 - Database Queries - row counts, schema checks, duration
 - Database XML Results - Schema check, duration
 - Database Table Schemas - Schema only, matches on TSQL Types
 - USE VS TEST FOR SG TESTS
They are also able to use VS Tests to write automated selenium grid tests in C#. SIT will import each method in your VSTest DLL and make it a SIT Test Step.
 - SIT Manual <http://is.factset.com/Docs/DevOps/SITManual.html>
 - SIT Notifications <http://is.factset.com/core/notifications/?notification=934>
The short version is.
 - Run tests after successful build
 - REST Plugin (includes IS API rules)
 - User guides have been published
 - Increased Lima integration
 - Versioning TFS
 - Ken/Tony work on scenarios

I was actually about to announce today that we have integration with our TFS builds. The fdsCrm team starting using it already. Developers can setup SIT tests to run after a successful build. It did not occur to me that AppQA would set this up for TFS builds. Let me know if we need to discuss this. Since the IS Queue Reader is running on the tellus servers it is possible that you could intercept build notification messages from our builds and create a plugin that will process them...or we could enhance our process to give you a way of knowing what TFS builds concern you.

From: Lucy Tancredi

Sent: Thursday, September 10, 2015 12:50 PM

To: Abilash Mohan; Keith W. Baker; Robert L. Groce; Usha Neelakandan; Chris Werner; Bryan A. Ehrlich; Tony Piazza; Manoj K. Yerrasani

Cc: Pooja Krishna Digumarthi; Padmini Rajanala; Jay Militscher; Alisa C. Berry; Gireesh Danduboina; Naveen Nemani; Ken D. Morris

Subject: SIT tests & QA coverage

As you create SIT tests, please ensure you are coordinating with QA / Padmini.

QA will ensure that the automated tests are hooked up to Tellus so that they are run with builds, and they will also be able to decommission manual tests that are duplicates of the SIT tests, freeing them up to expand testing in other areas of IS that are currently untested.

Thank you,

- Lucy

SIT RPD's

Thursday, September 10, 2015

11:21 AM

Documentation:

[RPD:18177928](#)

[RPD:17621058](#) (overrides)

Updated 7/22/2016 - Ken

SIT RPD View

RPD's

[RPD:24138923](#) SIT Master RPD

[RPD:22647517](#) SIT support for job ID

SIT WorkFlow

Thursday, June 04, 2015 2:04 PM

In our last meeting Tony brought up a good question. What are the jobs of each tool and how does that fit into the workflow of the user.

Let's discuss that and come up with an answer.

Notes from last meeting:

Notes:

1. Discussed how a Developer would use SIT
 - a. SIT –
 - i. **Job of SIT:**
 1. user: Developer tool,
 2. workflow: runs the tests that are defined in SIT
 - ii. Create test steps in SIT for an integration test
 - iii. Run at the point of test step creation
 - iv. For IS team hook up to build system
 1. Could hook up to other builds, but need to research per case basis
 - v. Determine which test are Release blockers
 1. Who determines Developer or QA person
 - vi. Add those test to the regression QAI Status Page Template
 1. Who does this SIT by Developer or QA person
 - vii. Need to turn off SIT notify if run by Tellus which has its own e-mail notify
 1. E-mail structures should match Tellus
 - b. Tellus
 - i. **Job of Tellus:**
 1. **user:** QA(Scheduled and adhoc) and Developer (scheduled)
 2. **Workflow:**
 - a. From Job entry page select the IS test and run the job (scheduled and Adhoc now)
 - b. E-mails results
 - i. E-mail structure should match b/w SIT
 - c. Need to send SIT URL to QAI (phase 1)
 - ii. Use to schedule a one time job for each test plan (example <http://tellus.pc.factset.com/calendar.php> May 18th 13:33)
 - iii. Run ad hoc outside SIT (questionable if people would use Tellus instead of SIT itself)
 - iv. Future create reoccurring scheduled job with multiple tests
 - v. Save history of the SIT Tellus runs so:
 1. runs are on the Tellus Calendar
 2. data in Tellus LB
 - a. IS and QAI on the same DB so can share info, but Tellus is on its own DB so sharing an issue
 - c. QAI
 - i. **Job of QAI:**
 1. Manual test execution and collection of manual tests.
 2. Also report Automated test results status.
 3. Tracks all test plans
 - ii. Track reoccurring test runs on a Status Webpage
 - iii. Test Plans and cases are created from SIT
 - iv. Test Step data is not on QAI
 1. Possibly have the data on Test run history (not setup yet) not in the Test manager [example in QAI](#)
 - v. Should QAI be the one place all runs Tellus or SWP are stored...
 - vi. Who sets up SIT test cases on QAI SWP? (see a iv & v above)

SIT data as mentioned above is stored so the tools can see and updated processes. Depending on the answer to how a user will use the Tellus and QAI applications can change what data we need to store.

We had a few open ended questions that we'll have to review after the SIT conference this week. ;)

SIT Protocol SF

Monday, October 12, 2015 12:29 PM

- After Jason creates the "spec" for integration points....
- His engineering team will develop the actual web service calls between SF apps and FDS apps in order to either transact or sync data
- We plan to create automated service tests to ensure each "conversation" works correctly
- Jason's team and Ken's team (he's the QA Automation Mgr) are to create a protocol for defining those tests in coordination with Priyanka (Alisa's QA Lead in Hyd)
- With that protocol in place, Jason's team will then create tests associated with the spec
- Ken is hiring an engineer in QA to take over that testing work, also following that protocol. But since that person isn't yet in place, it's important we establish that protocol first and then have Jason follow it until we take over.

Folks Priyanka Jhawar

SIT Protocol SF

From the specification created for Sales Force

Engineering team develops actual web service calls between SF apps and FDS apps in order to either transact or sync data

Those web service calls for either transact or sync data need to have test cases written for them.

Create test cases for API end points

SF consultant will define what the test cases are

The QA Analyst will create the test cases (Priyanka Jhawar)

SIT Test cases will be automated by the automation scripiter (still to be hired)
Create test cases to ensure each "conversation" works correctly.

The QA Analyst will view the results

If there are any changes to the

What Have we Here?

[SIT](#) down and stay a while--You've Got SIT todo

System integration testing (SIT) is exactly what it sounds like and so we named our system as it is. In the Information Systems department SIT evolved out of a need to test web service endpoints but once the idea got its grip on us we kept going--keep reading to understand everything it can do. The SIT application is designed with developers in mind. We want to make it easy for developers to test their own services and to test the services they consume. The Information Systems department hopes that SIT can be used to form a contract between service consumers and service suppliers and we understand that many of us wear both hats.

Today SIT can be used to test

- Web service output - exact matches, JSON schema checks, XML Schema checks, regex matches, status only, duration
- REST - REST Support is underway the plan is to support general REST and ISAPI based REST, more on this when it is available
- Database Queries - row counts, schema checks, duration
- Database XML Results - Schema check, duration
- Database Table Schemas - Schema only, matches on TSQL Types

SIT can also be used to automate Visual Studio Based tests but when we present this option we want to be careful and specific. There is no need for SIT to automate the running of Unit Tests your dev environment and your build system should handle this for you naturally and if it does not I might suggest you are not using the proper tools. However, we have gone with the idea that we can use VS Test to write integration tests and we might suggest that you go this route if you are trying to perform some sort of integration test that SIT does not support directly. We are also able to use VS Tests to write automated selenium grid tests in C#. SIT will import each method in your VSTest DLL and make it a SIT Test Step. These features are documented, read on.

Getting Started

Like I said earlier, we wrote SIT with developers in mind and we hope to make it relatively easy for developers to adopt. Let's start at the beginning, let's start where you will land first.

SIT will remember the last product you set in the Product control but it will only remember it if you chose the product for the control. This allows people to share URLs and jump around but only save the changed product when they do so with a purpose. On your first visit no product will have been chosen and the screen will be blank like it is above. To get started search for the product you are interested in. If other developers have already entered some test steps you will see a the screen change. Here is what the control looks like in the middle of a search.

And after you have selected a product that already has test results.

Many of the features on this page will be better understood after you have entered and run a test step so we are not going to dive in right now but feel free to poke around if you are following along in the app while you read this document.

We are going to focus on building tests for web services for the first part of this document. This is to give you an idea of the flow of the SIT portal and how developing tests works. Later we will have a look at the other test types.

Manage Profiles

All tests require a profile whether it be a web service test or one of the other test types. In the application bar, just above the Product Search, you will find a link that reads "Manage Profiles." Let's have a look at the empty profile form used to make a new profile. You can get to this form by following the "Manage Profiles" link and then using the "Add Product Profile" action button. In SIT all of these actions types will be found across from the "Product Search" and they all have the green plus icon



set to a slightly smaller size.

This form covers a bunch of the basic setup for your test steps and allows you to make certain assumptions when your test is running. Let's go over each field and we will discuss some of the changes that occur based on some of the values you choose.

- Name - Just a name to help you remember what this profile is for
- Profile Type - (Web, Database, VSTest) what type of test will this profile be used for? This is important as it drives what "Environment" data you enter and what plugins you can use on your test steps (more on this later.)
- Authentication Type - (Windows, Lima) this also drives some of the fields in the "Environment" data. IS Engineers will choose "Windows" unless they know otherwise as we have a few exceptions, Software Engineering will choose Lima unless they know otherwise.

Above we see the Product Profile form in Web Windows mode. First let me take a moment to describe the deal with environments. SIT was originally developed with Information Systems in mind so we used our standard environment labels **but these are just labels** and it has been suggested that we make the labels customizable and I am sure we will but for now just remember, **JUST LABELS**. Just think of these as the different environments that have versions of your application in their various stages of development and use a tab to represent each environment you need to test.

Notice that in Windows Web mode each environment has just two values you need to supply.

- Account - This is the windows account that will be used to run your test if you do not see the account(s) you need please file an RPD under the SIT project
 - Root URL - This is the root url for the current environment for instance say you are testing the HR Portal's Service layer you may want to use <http://iswebdev.pc.factset.com/HR/API/> in a tab for development and <http://iswebqa.pc.factset.com/HR/API/> in a tab for quality assurance
- If your test must use an employee please see [The Morph Module](#) and get this setup for the appropriate **non production** environment in your application. Then for the test step that needs to run as an employee add the X-MorphAs header as described in that Morph Module documentation. If this becomes cumbersome and popular please file an RPD under SIT and request that this be made into a part of the profile.

Lets have a look at Lima Web setup.

Hopefully by the time you are reading this there are five fields on that form rather than three. The five should be.

- Account - The username (directory) used to run the test
 - Root URL - This is the root url for the current environment for instance say you are testing the HR Portal's Service layer you may want to use <http://iswebdev.pc.factset.com/HR/API/> in a tab for development and <http://iswebqa.pc.factset.com/HR/API/> in a tab for quality assurance
 - Serial Number - The serial number to be used to run the test
 - Override - The override build setting to be used to direct this request in FDSA
 - Cluster - The cluster your build is building to (optional) but required depending on your team's configuration if you have no idea try with this blank
- The Database profile option is really designed for Windows Authentication with a SQL Server database you can also use SQL Server Logins by including the information in your connection string but we do not recommend this option for security reasons. Just supply the connections string and include a default database if you do not plan on fully qualifying all of your objects.

Manage Tests

This is the heart of SIT--creating and managing integration tests. Sometimes these tests are for your services and sometimes they are for services you are going to consume. In order to manage these tests SIT piggy backs off of [QAI](#). Traditionally QAI has been a management system for manual QA Testing, recording and organizing test results. More recently automated tests have been recorded here and now SIT tests will take advantage of the organizational structure that QAI provides. A test step is a member of a test case and test cases belong to test plans. When you create a test plan or a test case in SIT it is actually saved in QAI, you will be able to go over to QAI and find it under your product. Inside QAI you are not able to flag a plan or case to SIT this only happens when they are created in SIT. Let's first take a quick look at SIT's Test Plan form.

- Name - The name of the plan you wish to create it should represent what you plan to test at a broad level below your product
 - QA Lead - This will already display you and you should not be able to change this
- No plans are made up of cases. A case is specific thing you are testing. In terms of WebAPI a case might be for a specific controller then later when you make steps you may have one or more of each method the controller offers. The Test Case for looks like: `br />`

- Name - Name should give appropriate meaning yo
 - Tagged To - Allows you to tag another product giving you a way to say "Hey, we are using your API in this manner."
- Alright, you have Test Plan and Test Case containers you are ready to get to the heart of SIT. Its time to make a Test Step.

The Test Step

After clicking on the Test Case name in the display grid you come to a list of Test Steps (an empty list if you just created the case.) Notice at the top right hand side of the screen there are three buttons.



We will go over the "Add Test Step" button in detail. The other two buttons allow you to generate steps. The "RESTful API Step Wizard" only works if you follow the Information Systems API Standards and have a discoverable endpoint available. The "VSTest Step Wizard" is only usefully if you are using the Information Systems TFS instance and have a special integration test build setup for this purpose.

Let's go into detail about your first Web Service Test. In order to define a test that uses the "Raw Web Request" plugin you must have already made a Profile of type "Web." The other prerequisites are that you have made a Test Plan and a Test Case. Now go ahead and click that "Add Test Step" button. You will see.

This form comes to life as you make a few choices.

- Enter a name for this step (make it descriptive)
- Choose a profile
- Choose the "Raw Web Request" plugin

Once you perform these three steps the page will change to look something like.

On the top right you will see your selected profiles information and at the bottom you will see that two more form sections have appeared one for the "Actual" details of the request and one for how to handle "Expected" results of the request. **There are several plugins available and I will not be documenting them all here but we add hover cards to explain them so please use these for plugin specifics we are only going to cover the RawWebRequest plugin in this document.**

As a first step check off the environments you want this test to run in. Then study the "Actual" form section.

- The URL you enter is relative to the base URL (look in the profile section for details)
 - Http Verb is the verb to use for the request
 - Http headers - Use the Add button to enter as many headers as you need for this request you will get a textbox to enter the name and the value of the header
 - Request Body - If this is a POST or a PUT you can enter the body
- Now that you have entered the details about how to make the request the next step is to setup the criteria needed for pass vs. fail. Study the "Expected" form section.
- Duration (ms) - enter the maximum number of milliseconds the request is allowed to take before the test is considered a fail
 - Http Headers - Enter the required headers that should come back from this request the name is "hard" but for the value you can provide "*" for anything
 - Http Status - The required status for a passed test
 - Body Type - What will you be placing in the "Response Body" field (please use the plugins hover card help for specifics)
 - Line Endings - What do you expect, do you even care?
 - Response Body - Based on the Body Type filed you will leave this blank, put in the expected result for an exact match or used one of the schema options

Other Tips for Testing

Test Step Order

In side a Test Case the steps will be run in order so if you have a step that sets up some data and a future step that needs that data you can do this. You can reorder steps by loading the Test Case grid and using the drag icon at the beginning of the line. If you queue up a Case to run it will run all the valid steps it will not stop because one of them failed (we are looking into adding this feature.) You can not have dependencies across Plans we make no promises about order outside the Case.

Runing Tests

Steps can be run directly from the Test Step editor page or from the Test Step Grid page. You can queue up a case to run from the Case Grid and queue up and entire plan from the Plan Grid. You also have two more options for running tests. You can use [Tellus](#) to define an AdHoc run of a SIT test and the AppQA Team is working allowing you to specify a recurring

schedule for your SIT Tests. If you are in Information Systems and you use our TFS based build systems you can set tests to run after a successful build. You can set this up by using the InformationSystems.Shared.AddBuildTriggerToRunSITTests sproc. You can specify step, case of plan for the @type and provide the proper id with the other required information. To add multiple items to the same build just call the sproc multiple times.

Test Results

The View Results page shows the most recent run of each step but the last column gives you a link to the history of a step. At the present time we are keeping all history but we may purge older results in the future. On the results page you also have links to the Step itself and the result details. On the top corner of the page you also have a link to List Builder. You can analyze your tests using List Builder if that helps. Visiting the run result detail gives you all the information we have gathered about the step run. The top of the page is filled with generic information that is common to every test. The bottom of the middle of the page has details about the specific type of test. The bottom of the page has the detailed log output of the test run.

Tagging Tests

You can tag a Test Case to an application other than the one you are writing the test in. This is useful for writing tests for APIs that you are consuming and gives that API's developers the opportunity to run your test cases whenever they change things. You can find tests that are tagged to your application by using the "View Tagged TestCases" navigation link and you can view tagged test results by using the "Included Tagged Test Results" check box on the results page. You can also view tagged results using List

SIT Status

Friday, July 22, 2016 4:34 AM

Current STATUS

Right now I can confirm that Tellus API should support anything that can be done in the Tellus JEP by 22 July for testing.

However, the JEP does not support scheduling SIT (CRON) because there needs to be a way to get a Job ID. I linked [RPD:22647517](#) that covers that. We need conformation from [@TonyPiazza](#) and [@ChrisWerner](#) on that.

<http://is.factset.com/rpd/summary.aspx?messageId=24138923&commentId=24914579>

SIT ETA's

Friday, July 22, 2016 4:36 AM

- SIT
 - Integrate with QAI at the TestPlan/Test Case Level (Done)
 - Tellus should be able to pull list of SIT tests from QAI (Production OK, dev lists all, needs more testing)
 - Results are written to the Tellus DB (Yes, results are from the plugin for the SIT server)
 - Tellus can not pass results back to SWP (need to know the GroupID)
 - Result page needs to display SIT errors along with SIT test case status. (ETA 3 days)
 - QAI will process SIT results directly if started from QAI itself???(Chris W to confirm)
 - Issue for Tellus seems to be results and not knowing Group ID for results(Pass/Fail).
 - ETA - SIT dispatcher depends having Tellus GroupID - (2 weeks)
 - Link to SWP where the SIT results are at (This means on SIT not Tellus)