

Contents

Administering Load Test Suite (LTS)	1
Load Test Suite (LTS) Solution Overview	2
Load Test Suite Configuration	2
Copy Source Files to the Application/Batch Folder	2
Create Load Testing User	2
Install OneStream Studio	2
PowerShell Setup	3
Create Folder for Test Suite	3
Extract XF MarketPlace Package	3
Unzip Test Scripts	3
Update PowerShell Scripts	3
Update Server Configuration	3
Update Test Sequence xml Files	3
Update Metadata files	4
Import Application Metadata Zip Files	4
Update Scenario Mapping	5
Update Workflow Profiles	5
Update Batch Scripts	5
Update Data Management Sequence	6
Update Data Management Step	6
Update Cube Views	8
Pre-Built Load Test Suite Tests	9
PowerShell Scripts	9
Sequence Files	10
Using Load Test Suite	11
Test Data Management Jobs	11
Run Tests	11
Analyze Results	11
Modifying MarketPlace Solutions	11

Administering Load Test Suite (LTS)

This guide describes the necessary steps to setup and use the Load Test Suite (LTS) Solution.

Load Test Suite (LTS) Solution Overview

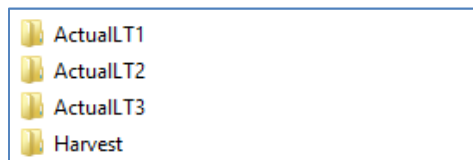
The Load Test Suite (LTS) Solution provides business users with an automated method to accurately test and view an application's overall system performance during data loads resulting from large numbers of simultaneous users and transactions. The Load Test Suite determines whether the application can handle peak usage periods.

Load Test Suite Configuration

The Load Test Suite requires multiple steps to configure and run and this section provides the detailed steps required for the Load Test Suite Solution.

Copy Source Files to the Application/Batch Folder

During Load Testing, OneStream replicates the Workflow Processing load through Batch Processing. In order to run the Batch Process, files normally used during a Workflow Import are placed in the Application/Batch ActualLT1 – LT3 Folders on the server under the appropriate naming. During the Load Testing, the Workflow files are copied from the ActualLT1-ActualLT3 Folders into the Harvest Folder from which they are pulled for the Batch Processing.



The OneStream Load Tester requires one file per Workflow process to be replicated during Load Testing. Multiple Workflows/Scenarios are used during Load Testing to ensure these Workflows are not loading to the same target Entities/Scenario combinations.

See the *OneStream XF Design and Reference Guide* for Batch Loading File naming requirements.

Create Load Testing User

The OneStream Load Test Suite replicates users logging onto the selected application through a single user instead of creating multiple users in the system. The default tests included in the Load Test Suite are set up to use the *Test1* user.

The user needs to be configured as follows:

- 1.) Create a user named *Test1*
- 2.) Add *Test1* to the *Administrator* Security Group
- 3.) Set the user's local password to *123*
- 4.) Login as the *Test1* user
- 5.) Reset the *Test1* password to *Password1* when prompted
- 6.) Set the user's Workflow POV to match the administrator's Workflow POV
- 7.) Set the user's POV to match the administrator's POV
- 8.) Logoff *Test1* user

Install OneStream Studio

The Load Test Suite uses OneStream Studio API's during Load Testing, so ensure Studio is installed on the client machine being used. This must run on a client machine and not a server, or load balancing will not work properly during testing.

Test to confirm the local Studio install can access the environment being load tested.

PowerShell Setup

The Load Test scripts used in testing are Microsoft PowerShell, so ensure PowerShell is setup/configured to run on the client machine being used. Details on setting up PowerShell for OneStream integration can be found in the PowerShell (*POW*) Solution located in the XF MarketPlace.

Create Folder for Test Suite

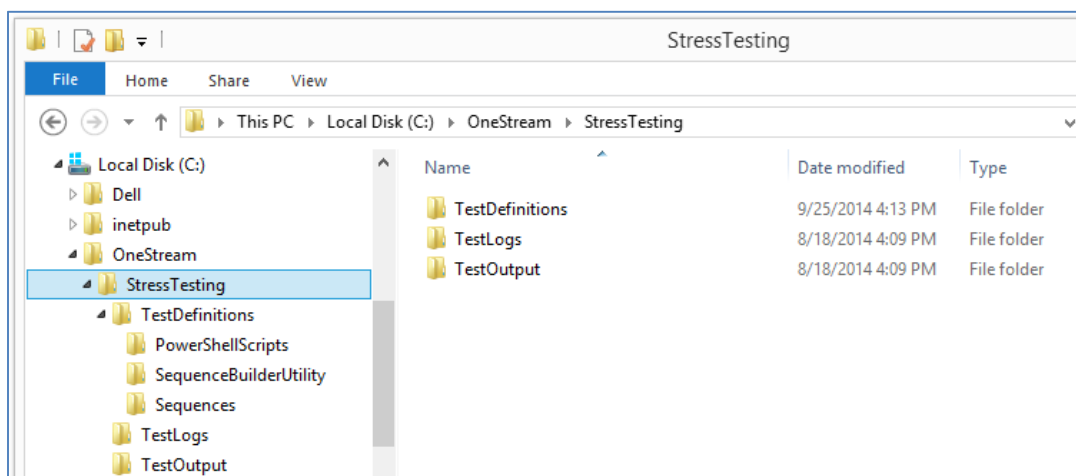
The Load Test Suite is setup to use the *C:\OneStream* folder on the test client machine. This folder needs to be created if it does not already exist on the client machine running the Load Testing.

Extract XF MarketPlace Package

The Load Test Suite Package contains four zip files. Extract the contents of the Load Test Suite Package to the *C:\OneStream* folder created in the previous step.

Unzip Test Scripts

Extract the *XFT LoadTester ClientScriptFilesAndFolders (LTS).zip* and unzip it to the *C:\OneStream* folder created in a prior step.



Update PowerShell Scripts

Each PowerShell Script (*C:\OneStream\StressTesting\TestDefinitions\PowerShellScripts*) included in the Load Test Suite Solution needs to be updated to ensure the proper Studio client API paths are specified on line three of the script. See the example below:

```
1 #Note: refer to PowerShell setup instructions in "C:\XF\DevelopmentTools\Configuration\Windows PowerShell Notes.docx"  
2 #Studio Install Path  
3 Add-Type -Path "C:\Program Files (x86)\OneStream Software\OneStreamStudio\OneStreamClientApi.dll"  
4 #Development Matching Path  
5 #Add-Type -Path "C:\XF\Source\Client\Windows\OneStreamClientApi\bin\Debug\OneStreamClientApi.dll"
```

Update Server Configuration

The default tests in the Load Test Suite are set to use two Gen/Stage/Data Management servers and one Consolidation server. Make sure the test environment has this configuration prior to running the first round of testing.

Update Test Sequence xml Files

The test sequence files included in the Load Test Suite use default values for both WebServer and Application. These must be updated with the correct names from the test environment in order to operate

correctly during Load Testing. The Load Test Suite provides an editing tool for the sequence files located in the `C:\OneStream\StressTesting\TestDefinitions\SequenceBuilderUtility` folder.

Once the editor is opened, browse to the sequence file that needs to be updated which is located in `C:\OneStream\StressTesting\TestDefinitions\Sequences`.

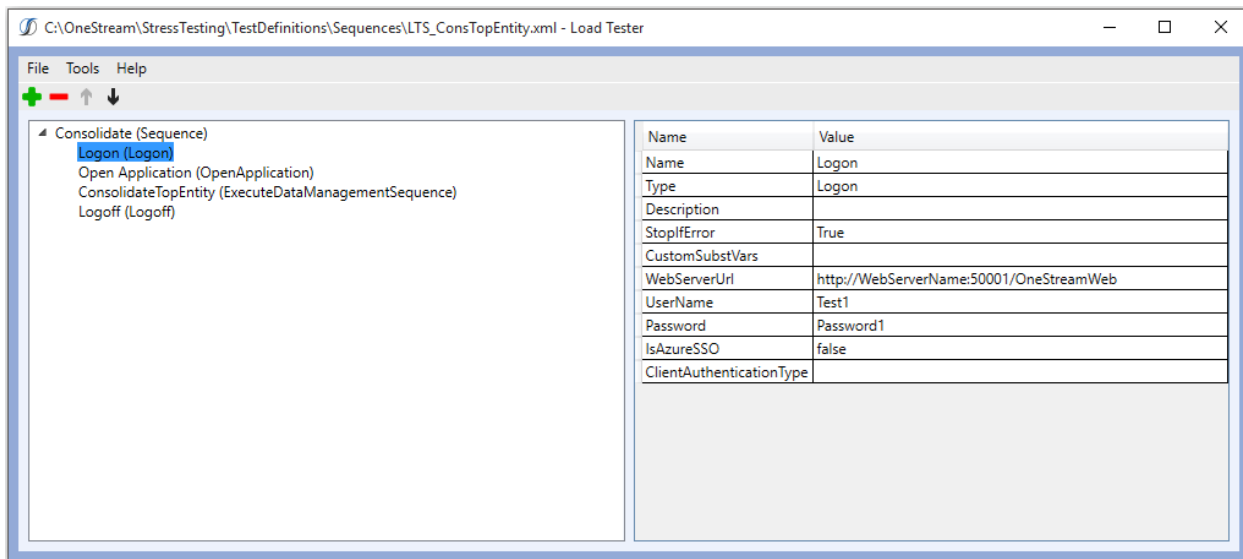
In the editing tool, update the following actions:

Logon

In the `WebServerUrl` field, replace the `WebServerName` with the name of the current Web Server

Open Application

In the `ApplicationName` field, replace `GolfStream` with the current application name



Update Metadata files

The Load Test Suite creates ten Workflows used for load testing during Batch Processing. In order to add the Workflows to the appropriate area, the `XFT LoadTesterWorkflowProfiles (LTS).xml` file needs to be updated, so that it will import into the correct area of the application being tested.

Cube Name

Replace all instances of `cubeName="GolfStream"` with `cubeName="<CurrentCubeName>"`

Parent Workflow

Replace all instances of `parentName="GolfStream"` with `parentName="<ParentNodeInDefaultWF>"`

Scenario Name

Replace `type="Scenario" name="MainScenarioDim"` with `type="Scenario" name="<CurrentReportingScenarioName>"`

Import Application Metadata Zip Files

Import the zip files containing the Load Test Analyzer and Load Tester Application using the Load/Extract page on the Application Tab. These files contain the metadata for not only performing the Load Testing, but also analyzing the results after the testing is completed.


Load Test Analyzer

The *XFT LoadTestAnalyzer App (LTS).zip* file contains metadata to setup a Workflow used to import and analyze the Load Testing log files.

Load Testing

The *XFT LoadTester App (LTS).zip* file contains metadata used to setup the Load Testing components in the OneStream Application.

Update Scenario Mapping

The Load Testing portion of the Load Test Suite creates three Scenarios for loading test data so as not to affect reporting data. These Scenarios need to be mapped, and the mapping is included in the *XFT LoadTester SuppFiles (LTS).zip* file. Extract the TRX file from the zip and import it into the main reporting Scenario's Transformation Rules. Open the Transformation Rules screen and select the main reporting Scenario map. Select  in the toolbar, browse out to the TRX file, and then click ok to import the map.

Update Workflow Profiles

The Load Test Suite uses ten Workflow Profiles named LT1 through LT10 to run Batch Processing for Load Testing. The Workflows' Import Profile must be updated to use the Data Source and Transformation Rules associated with the Batch Import files copied to the application/batch folder.

Profiles: LT1.Import through LT10.Import

The Workflow Profile Access and Maintenance Security Groups are set to the values stored in the Default Workflow Template. These should be secured to the *Administrator* Security Group in order to prevent other users from accessing these Workflows.

In order for the data to load correctly, ensure that the Data Source for the Load Test Workflows uses the *Current DataKey* setting for the Scenario Dimension.

The Workflow profiles LT1 – LT10 have the Workflow Channel set to Standard by default. If using Workflow Channels in your application ensure that the appropriate channel is set in these WF profiles.

Update Batch Scripts

During Load Testing, OneStream replicates the Workflow Processing load through Batch Processing. The batch scripts that are called during this process copy the files that were placed in the Application/Batch ActualLT1 – LT3 Folders. This copy process is pre-configured to copy text (*.txt) files. If the files to be used in testing are of a different type these scripts need to be updated to use the correct file extension. The Extensibility scripts *LTS_CopyFilesforBatchLT1 – LT3* need to have line 30 updated to use the correct file extension.

```

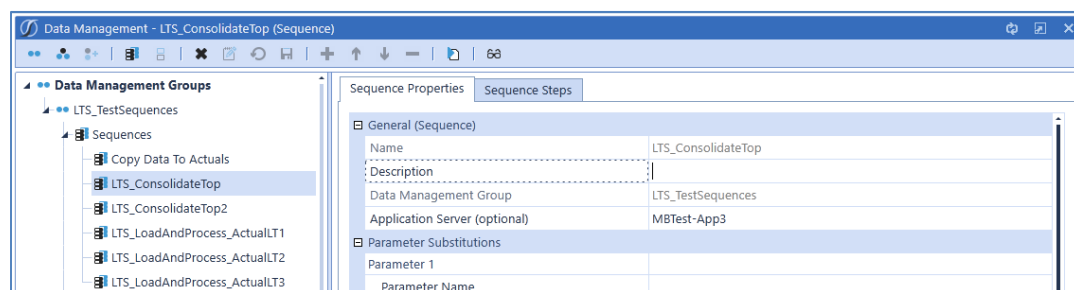
19 Namespace OneStream.BusinessRule.Extender.LTS_CopyFilesforBatchLT1
20 Public Class MainClass
21     Public Function Main(ByVal si As SessionInfo, ByVal globals As BRGlobals, ByVal api As Object, ByVal args As ExtenderArgs) As Object
22     Try
23         system.Threading.Thread.Sleep(5000)
24
25         'Get source directory & Harverst directory
26         Dim sourceDir As String = BRAP1.Utilities.GetFileShareFolder(si, filesharefoldertypes.Batch , Nothing) & "\ActualLT1"
27         Dim targetDir As String = BRAP1.Utilities.GetFileShareFolder(si, filesharefoldertypes.BatchHarvest , Nothing)
28
29         'Get list of files to copy
30         Dim fileList = Directory.GetFiles(sourceDir, "*.txt")

```

Update Data Management Sequence

Load Testing includes running consolidations during testing. These consolidations are called from a Data Management Sequence. The Load Test Suite expects all consolidations to perform on one specific server during Load Testing. In order to ensure that all consolidations run on the correct server, the Data Management Sequence must specify the name of the consolidation server.

Select the *LTS_ConsolidateTop* Sequence from the Data Management screen. Enter the name of the consolidation server being used in the *Application Server* field.



Update Data Management Step

The *LTS_ConsolTopSpecificServer* Data Management Step contains the settings for running a consolidation. The Data Unit settings need to be updated to the correct values for the system being tested.

Cube

Set to default reporting Cube

Entity Filter

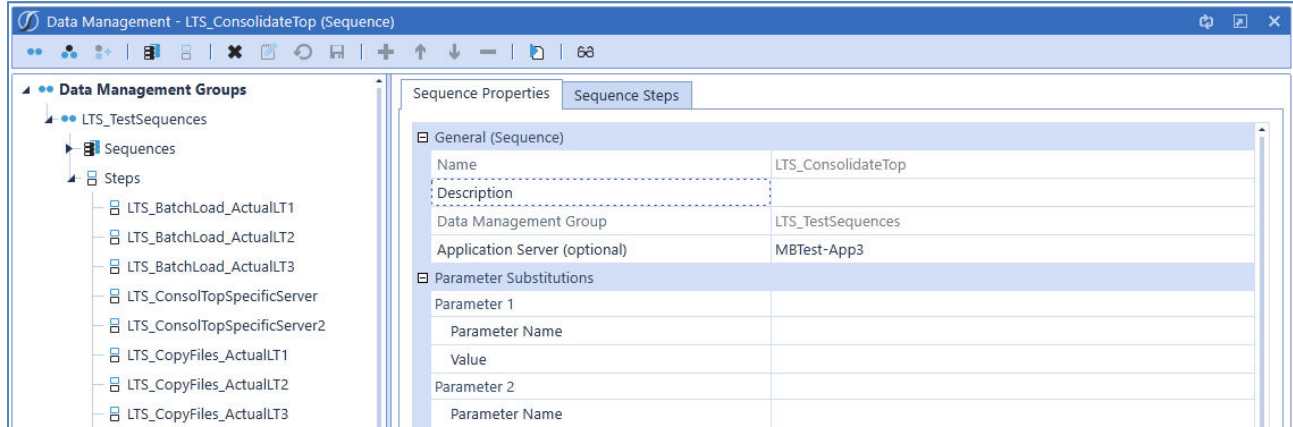
Set to Top Level Entity Parent in reporting Cube

Scenario

Reporting Scenario being consolidated

Time

Reporting Period being consolidated



Update Cube Views

The Cube Views included in the Load Test Suite are used to simulate the users viewing data in the application. The Cube Views in the *LTS_LoadTest* Group will not have the same dimensionality as the applications being tested and will need the following updates:

- Add valid accounts to the rows, or use shared rows/columns from an existing report
- Update the POV
 - **Note:** Do not change formula in the Entity POV
- Update BRString Rule variables on the POV tab
 - These values are used to randomly select an Entity when running the Cube View load tests.
 - Replace *[CorpEntities]* with the name of the current Entity Dimension
 - Replace *[GolfStream]* with the name of the Top Entity Dimension Member used for reporting
 - Replace *[Houston]* with a valid Base Entity Member in the current Entity Dimension
- Verify that each Cube View runs correctly when ran manually
 - The Cube Views should return a random Entity each time one is run

Pre-Built Load Test Suite Tests












The Load Test Suite contains pre-built tests consisting of PowerShell scripts and OneStream sequence xml files. Together, these files allow the Load Test Suite to call functions within OneStream and replicate an actual user load in the system.

PowerShell Scripts

A series of pre-configured tests are included in the *PowerShellScripts* folder. PowerShell scripts are ran from the PowerShell interface on the client machine. These scripts replicate any number of users calling one to many sequence files.

- LTS_1_CubeView replicates:
 - i. Ten users calling LTS_ExecuteCubeViews sequence
- LTS_1_LoadProc_LT1 replicates:
 - i. One user calling LTS_LoadAndProcessLT1 sequence
- LTS_10_CubeView_LoadProc_LT1_Cons replicates:
 - i. Ten users calling LTS_ExecuteCubeViews sequence
 - ii. One user calling LTS_LoadAndProcessLT1 sequence
 - iii. One user calling LTS_ConsTopEntity sequence
- LTS_10_CubeView_LoadProc_LT1_LT2_Cons replicates:
 - i. Ten users calling LTS_ExecuteCubeViews sequence
 - ii. One user calling LTS_LoadAndProcessLT1 sequence
 - iii. One user calling LTS_LoadAndProcessLT2 sequence
 - iv. One user calling LTS_ConsTopEntity sequence
- LTS_10_CubeView_LoadProc_LT1_LT2_LT3_Cons replicates:
 - i. Ten users calling LTS_ExecuteCubeViews sequence
 - ii. One user calling LTS_LoadAndProcessLT1 sequence
 - iii. One user calling LTS_LoadAndProcessLT2 sequence
 - iv. One user calling LTS_LoadAndProcessLT3 sequence
 - v. One user calling LTS_ConsTopEntity sequence
- LTS_60_CubeView_LoadProc_LT1_Cons replicates:
 - i. Sixty users calling LTS_ExecuteCubeViews sequence
 - ii. One user calling LTS_LoadAndProcessLT1 sequence
 - iii. One user calling LTS_ConsTopEntity sequence
- LTS_60_CubeView_LoadProc_LT1_LT2_Cons replicates:
 - i. Sixty users calling LTS_ExecuteCubeViews sequence
 - ii. One user calling LTS_LoadAndProcessLT1 sequence
 - iii. One user calling LTS_LoadAndProcessLT2 sequence
 - iv. One user calling LTS_ConsTopEntity sequence
- LTS_60_CubeView_LoadProc_LT1_LT2_LT3_Cons replicates:
 - i. Sixty users calling LTS_ExecuteCubeViews sequence
 - ii. One user calling LTS_LoadAndProcessLT1 sequence
 - iii. One user calling LTS_LoadAndProcessLT2 sequence
 - iv. One user calling LTS_LoadAndProcessLT3 sequence
 - v. One user calling LTS_ConsTopEntity sequence
- LTS_100_CubeView_LoadProc_LT1_Cons replicates:
 - i. One hundred users calling LTS_ExecuteCubeViews sequence
 - ii. One user calling LTS_LoadAndProcessLT1 sequence
 - iii. One user calling LTS_ConsTopEntity sequence
- LTS_100_CubeView_LoadProc_LT1_LT2_Cons replicates:

- i. One hundred users calling LTS_ExecuteCubeViews sequence
- ii. One user calling LTS_LoadAndProcessLT1 sequence
- iii. One user calling LTS_LoadAndProcessLT2 sequence
- iv. One user calling LTS_ConsTopEntity sequence
- LTS_100_CubeView_LoadProc_LT1_LT2_LT3_Cons replicates:
 - i. One hundred users calling LTS_ExecuteCubeViews sequence
 - ii. One user calling LTS_LoadAndProcessLT1 sequence
 - iii. One user calling LTS_LoadAndProcessLT2 sequence
 - iv. One user calling LTS_LoadAndProcessLT3 sequence
 - v. One user calling LTS_ConsTopEntity sequence

Name	Date modified	Type	Size
 LTS_1_CubeView	9/19/2014 9:34 AM	Windows PowerShell Script	7 KB
 LTS_1_LoadProc_LT1	9/19/2014 9:34 AM	Windows PowerShell Script	7 KB
 LTS_10_CubeView_LoadProc_LT1_Cons	9/19/2014 9:34 AM	Windows PowerShell Script	7 KB
 LTS_10_CubeView_LoadProc_LT1_LT2_Cons	9/19/2014 9:34 AM	Windows PowerShell Script	7 KB
 LTS_10_CubeView_LoadProc_LT1_LT2_LT3_Cons	9/19/2014 9:35 AM	Windows PowerShell Script	7 KB
 LTS_60_CubeView_LoadProc_LT1_Cons	9/19/2014 9:35 AM	Windows PowerShell Script	7 KB
 LTS_60_CubeView_LoadProc_LT1_LT2_Cons	9/19/2014 9:35 AM	Windows PowerShell Script	7 KB
 LTS_60_CubeView_LoadProc_LT1_LT2_LT3_Cons	9/19/2014 9:35 AM	Windows PowerShell Script	7 KB
 LTS_100_CubeView_LoadProc_LT1_Cons	9/19/2014 9:39 AM	Windows PowerShell Script	7 KB
 LTS_100_CubeView_LoadProc_LT1_LT2_Cons	9/19/2014 9:40 AM	Windows PowerShell Script	7 KB
 LTS_100_CubeView_LoadProc_LT1_LT2_LT3_Cons	9/19/2014 9:40 AM	Windows PowerShell Script	7 KB

Sequence Files

Sequence files are called from PowerShell scripts and contain the commands used to execute processes within OneStream.

LTS_ConsTopEntity

Executes a top level consolidation using a Data Management Job

LTS_ExecuteCubeViews

Executes two Cube Views with a random wait time between them to simulate users viewing reports. It then loops and re-runs the process which replicates one user running four Cube Views.

LTS_LoadAndProcessLT1

Executes a Batch Load using a Data Management Job. The first Data Management Sequence called copies the Batch Files into the Batch Folder and then executes the Batch Process running four parallel file groups.

LTS_LoadAndProcessLT2

Executes a Batch Load through a Data Management Job. The first Data Management Sequence called copies the Batch Files into the Batch Folder and then executes the Batch Process running four parallel file groups.

LTS_LoadAndProcessLT3

Executes a Batch Load through a Data Management Job. The first Data Management Sequence called copies the Batch Files into the Batch Folder and then executes the Batch Process running four parallel file groups.

Using Load Test Suite

Test Data Management Jobs

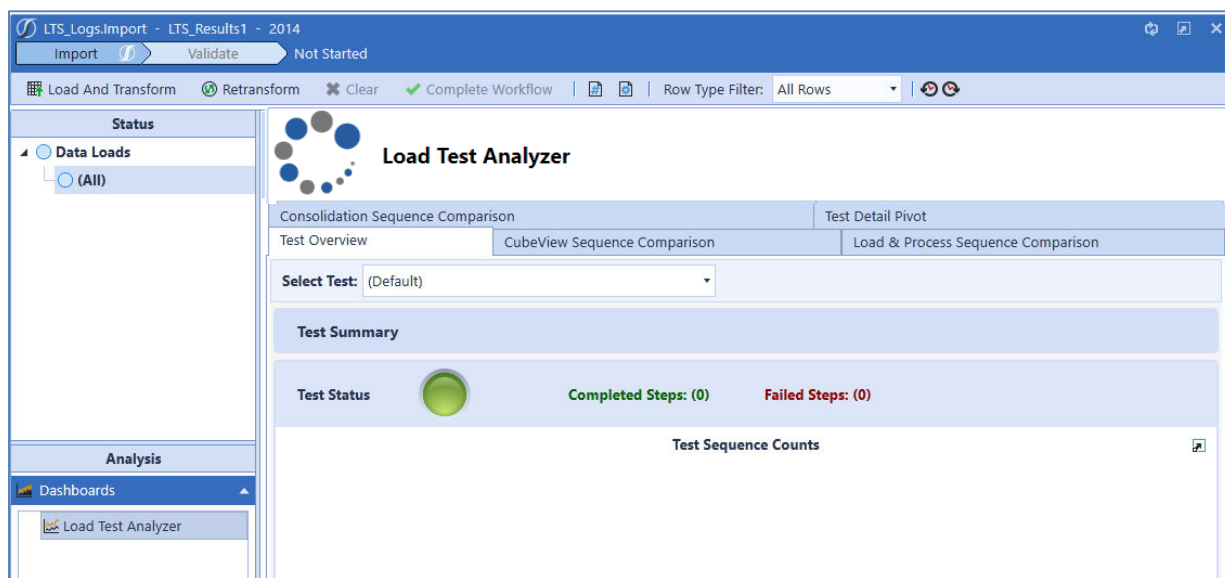
Prior to running the full Load Test Suite, manually run the four sequences in the *LTS_TestSequences* Group in order to ensure they work properly and no errors occur.

Run Tests

Once the Load Test Suite has been configured and the Data Management Jobs have been tested, the Test Suite is ready to be run. The tests (*PowerShell scripts*) can be retrieved from the PowerShell interface.

Analyze Results

The tests will create log files in the *C:\OneStream\StressTesting\TestLogs* folder after they run which can then be imported into the *LTS_Logs* Workflow. After importing the result log files into OneStream, users can review/analyze the results from the Dashboard Analysis section of the Workflow.



Modifying MarketPlace Solutions

A few cautions and disclaimers when modifying a MarketPlace Solution:

- Major changes to Business Rules or custom tables within a MarketPlace Solution will not be supported through normal channels as the resulting solution is significantly different from the core solution.
- If changes are made to any Dashboard object or Business Rule, consider renaming it or copying it to a new object first. This is important because if there is an upgrade to the MarketPlace Solution in the future and the customer applies the upgrade, this will overlay and wipe out the changes. This also applies when updating any of the standard reports and Dashboards.
- If modifications are made to a MarketPlace Solution, upgrading to later versions will be more complex depending on the degree of customization. Simple changes such as changing a logo or colors on a Dashboard do not impact upgrades significantly. Making changes to the custom database tables and Business Rules, which should be avoided, will make an upgrade even more complicated.