Load Test Suite Guide

PV600 SV100

Copyright © 2023 OneStream Software LLC. All rights reserved.

Any warranty with respect to the software or its functionality will be expressly given in the Subscription License Agreement or Software License and Services Agreement between OneStream and the warrantee. This document does not itself constitute a representation or warranty with respect to the software or any related matter.

OneStream Software, OneStream, Extensible Dimensionality and the OneStream logo are trademarks of OneStream Software LLC in the United States and other countries. Microsoft, Microsoft Azure, Microsoft Office, Windows, Windows Server, Excel, .NET Framework, Internet Information Services, Windows Communication Foundation and SQL Server are registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. DevExpress is a registered trademark of Developer Express, Inc. Cisco is a registered trademark of Cisco Systems, Inc. Intel is a trademark of Intel Corporation. AMD64 is a trademark of Advanced Micro Devices, Inc. Other names may be trademarks of their respective owners.

Table of Contents

Administering Load Test Suite 1

Load Test Suite Solution Overview 2

Configure Load Test Suite 3

- Create Load Testing Suite User 3
- Install Client API 3
- PowerShell Setup 3
- Create Folder for Test Suite 4
- Extract MarketPlace Package 4
- Unzip Test Scripts 4
- Update PowerShell Scripts 5
- Update Server Configuration 5
- Update Test Sequence XML Files 5
  - Logon 6
  - Open Application 6
- Update Metadata Files 6
  - Scenario Name 7
- Update Workflow Profiles (configured based on customer setup) 7
  - Cube Name 7
  - Parent Workflow 7

Import Transformation Profile (Actual) 7
Import Data Source (Actual) 8
Workflow Profile 8
Parent Workflow Profile 8
Update Scenario Mapping 8
Copy Source Files to the Application/Batch Folder 9
Import Data Sources Business Rule 10
Import Application Metadata .Zip Files 11
Installer Part A 11
Installer Part B 11
Update Batch Scripts 11
Update Cube Views 11
Update Data Management Sequence 12
Update Data Management Step 12
Test Data Management Jobs 13
Pre-Built Load Test Suite Tests 14
PowerShell Scripts 14
Sequence Files 17
Using Load Test Suite 18
Run Tests 18

Import and Analyze Results 19
View all Results 21
Optimal Display Settings 22
Modifying MarketPlace Solutions 23

Administering Load Test Suite

This guide describes how to set up and use Load Test Suite.

Load Test Suite Solution Overview

The Load Test Suite 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. Load Test Suite determines if an application can handle peak usage periods.

Configure Load Test Suite

This section describes how to configure and run Load Test Suite.

Create Load Testing Suite User

Load Test Suite replicates users logging onto an application through one, not multiple, system users. The default tests provided are set up to use the Test1 user.

Configure users by performing these steps:

1. Create a user named *Test1*.
2. Add *Test1* to the **Administrator Security** Group.
3. Set the user's local password to *123*.
4. Log in as *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 Cube POV to match the administrator's POV.
8. Log off *Test1* user.

Install Client API

Load Test Suite uses OneStream Client APIs during load testing, so ensure the client API is installed on your machine. This must run on a client machine and not a server, otherwise load balancing will not work properly during testing. Test to confirm the local Client API install can access the environment being load tested.

PowerShell Setup

The Load Test scripts used in testing are Microsoft PowerShell, so ensure PowerShell is configured to run on the client machine being used. Information about for setting up PowerShell for OneStream integration is in the PowerShell (POW) Solution on the MarketPlace.

The Elevation level for Windows may need to be adjusted, as the PowerShell tool requires "run as an administrator" accessibility.

Create Folder for Test Suite

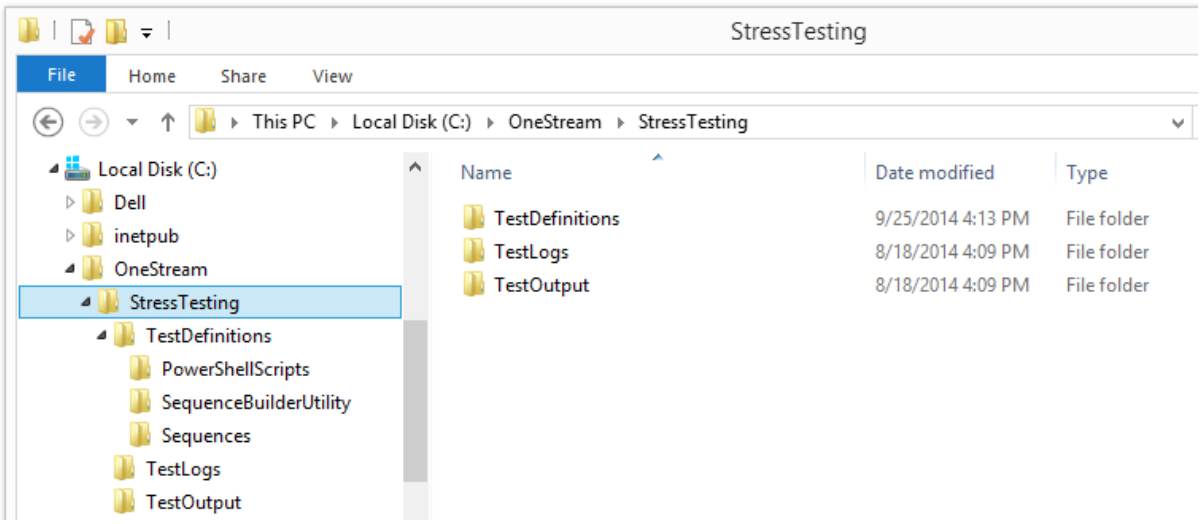
Load Test Suite uses the *C:\OneStream* folder on the test client machine. Create this folder if it does not already exist on the client machine running the load testing.

Extract MarketPlace Package

The Load Test Suite package contains several .zip files. Extract the contents of the package and constituent .zip files.

Unzip Test Scripts

Copy the StressTesting folder from *Supplemental Configuration.zip* and unzip it to the *C:\OneStream* folder created in a previous step.



Update PowerShell Scripts

Update each provided PowerShell script (*C:\OneStream\StressTesting\TestDefinitions\PowerShellScripts*) to ensure the proper Studio client API paths are specified on line 3 of each script. The path will be where the Client API was installed on the client machine. By default this is:

```
C:\ProgramFiles (x86)\OneStreamSoftware\OneStreamStudio\OneStreamClientApi.dll
```

For example:

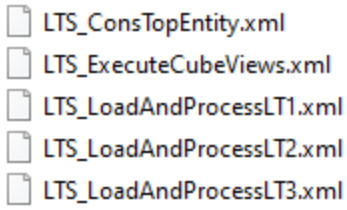
```
1  #Note: refer to PowerShell setup instructions in "C:\XF\DevelopmentTools\Configuration\Windows PowerShell Notes.docx"
2  #Studio Install Path
3  Add-Type -Path "C:\Program Files (x86)\OneStream Software\OneStreamStudio\OneStreamClientApi.dll"
4  #Development Maching Path
5  #Add-Type -Path "C:\XF\Source\Client\Windows\OneStreamClientApi\bin\Debug\OneStrea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

Update Test Sequence XML Files

The test sequence files included in Load Test Suite use default values for WebServer and Application. These files must be updated with the correct names from the test environment to operate correctly during load testing. Use the provided editing tool for the sequence files in *C:\OneStream\StressTesting\TestDefinitions\SequenceBuilderUtility*.



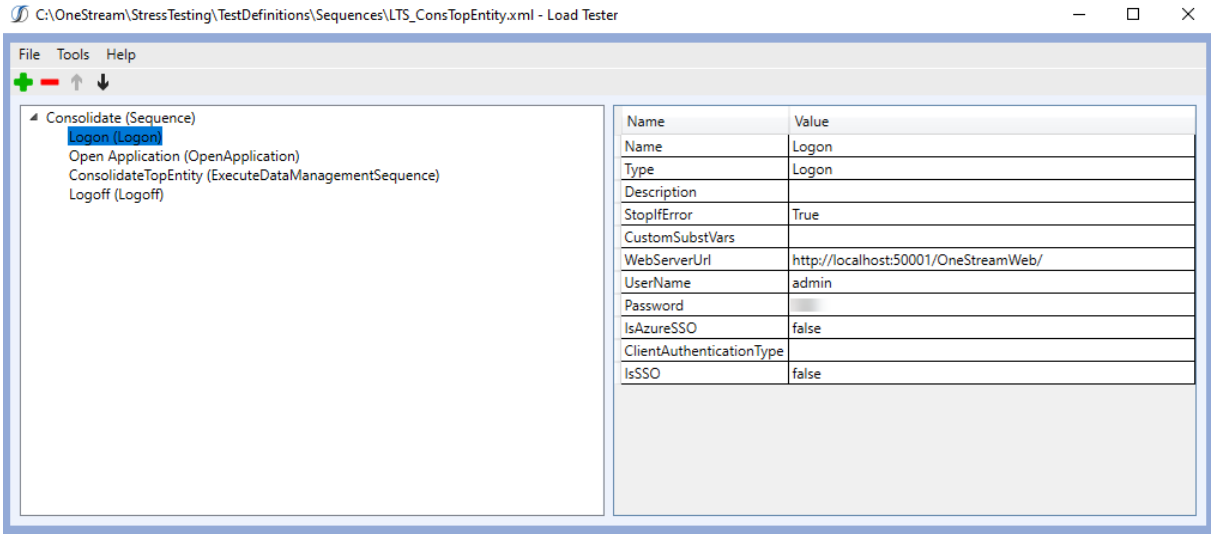
Once the editor is opened, browse to the sequence file that needs to be updated which is located in *C:\OneStream\StressTesting\TestDefinitions\Sequences.*

In the editing tool, update the following actions:

Logon

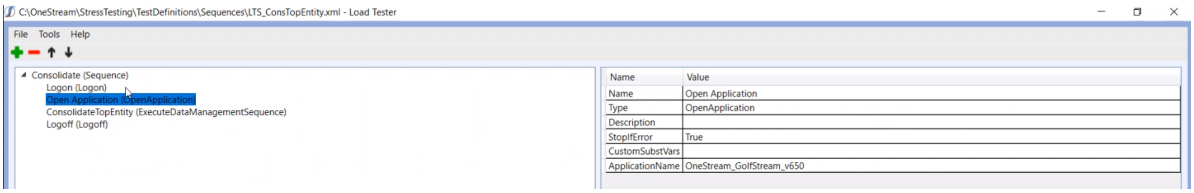
In the *WebServerUrl* field, replace the *WebServerName* with the name of the current Web Server.

> **NOTE:** Update your username and password. Refer to the previous section, Create Load Test Suite user.



Open Application

In the *ApplicationName* field, replace *GolfStream* with the current application name.



Update Metadata Files

The Load Test Suite Scenario dimension must be updated so that it imports into the correct scenario being tested. Unzip **Installer Part B** and update **LoadTestScenarios.xml**.

Scenario Name

Replace the instance of

name="Scenarios" with *name="<Your Reporting Scenario Name>"*

Update Workflow Profiles (configured based on customer setup)

Load Test Suite creates ten workflows used for load testing during batch processing. To add workflows to the appropriate area, LoadTester WorkflowProfiles.xml file must be updated so that it will import into the application area being tested.

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 to prevent other users from accessing these workflows.

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. If you use Workflow Channels in your application, ensure that the appropriate channel is set in these workflow profiles.

Cube Name

Replace all instances of

cubeName="XFR_DataSources" with cubeName="<your Cube Name>".

Parent Workflow

Replace all instances of

parentName="XFR_DataSources" with parentName="<Your Default WF Parent Node Name>".

Import Transformation Profile (Actual)

Replace all instances of

attributeValue="LTS_DataSource" with attributeValue="<Your Transformation Profile Name>".

Import Data Source (Actual)

Replace all instances of

attributeValue="LTS_Delimited_Source" with attributeValue="<Your Data Source Name>"

Workflow Profile

Update the profile name to that of your current workflow.

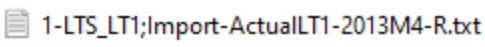
Replace all instances of:

name="LTS_LT{1 - 10}.Import" with name="<Your Profile Name>"

name="LTS_LT{1 - 10}.Forms" with name="<Your Profile Name>"

name="LTS_LT{1 - 10}.Adj" with name="<Your Profile Name>"

Batch file name format should coincide with your profile name. Details on setting up the batch file name is located in the OneStream Design and Reference Guide within the Batch File Name Format Specification - Profile Name section.



Parent Workflow Profile

Replace all instances of

parentName="LTS_LT{1 - 10}" with parentName="<Your Parent Profile Name>".

Update Scenario Mapping

The load testing portion of the Load Test Suite creates multiple scenarios, to avoid affecting reporting data. To map these scenarios, navigate to the *Supplemental Configuration.zip* file.

There are three options in which scenario mapping may occur.

The first option is to customize *LoadTestAnalyzer_TransformationRules_GolfStream.xml* in the *LoadTester Transformation Rules* folder. Once updated, this file must replace the *TransformationRules.xml* file in the *Installer Part A.zip* folder.

The second option is to customize each sample transformation rule given your preferred cube dimensions, by performing the following steps. Example transformation rules are in *Sample Files.zip*.

1. Open the Transformation Rules screen and select the main reporting Scenario map.

2. Select in the toolbar, browse to your customized .trx file, and then click **OK** to import the map.

The third option is to use the default .trx file that includes the minimum scenarios required. Perform these steps:

1. Extract the TRX file from within the *LoadTester Transformation Rules* folder and import it into the main reporting Scenario's Transformation Rules.

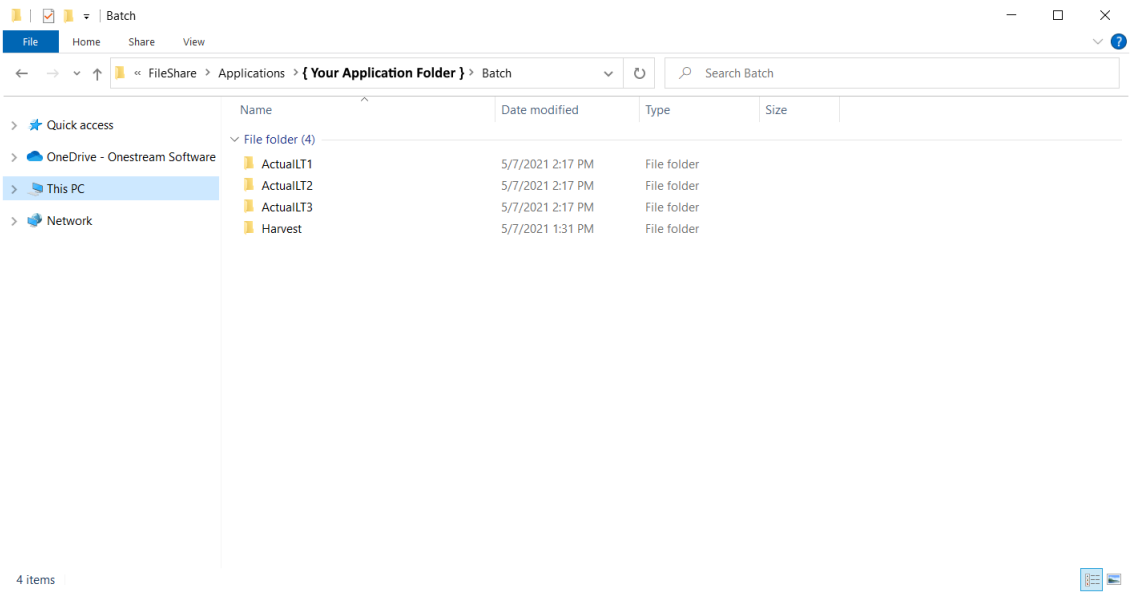
2. Open the Transformation Rules screen and select the main reporting Scenario map.

3. Select in the toolbar, browse to the TRX file, and then click **OK** to import the map.

Copy Source Files to the Application/Batch Folder

Create the ActualLT1, ActualLT2, and ActualLT3 folders shown below, before running the batch process. Each folder represents the load test scenarios created in a Load Test application.

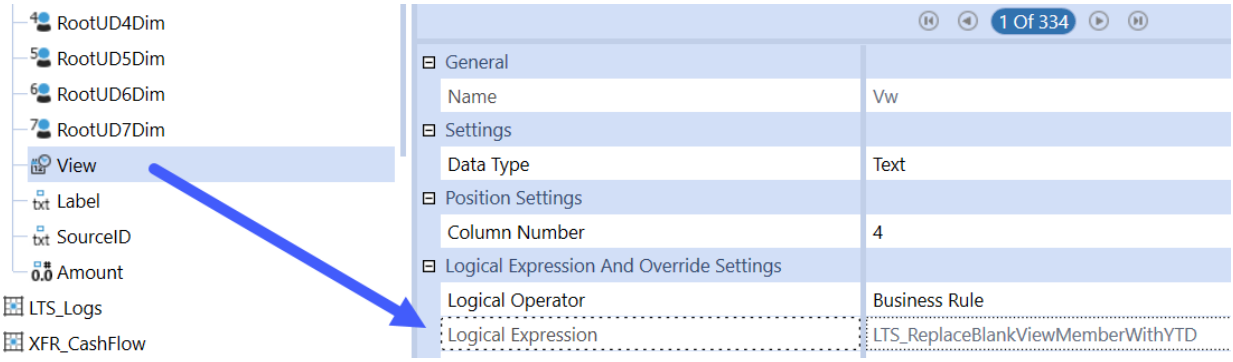
During load testing, the workflow processing load is replicated through batch processing. To run the batch process, files used during a workflow import are placed in the Application/Batch ActualLT1 – LT3 folders on the server in the appropriately named folders. During load testing, the workflow files are copied from the ActualLT1-ActualLT3 folders into the Harvest folder where they are used for batch processing.



OneStream Load Test Suite 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.

Import Data Sources Business Rule

To process each import file successfully, a data transformation business rule may be needed to handle NULL or blank values. Add the appropriate *DataSources.xml* file as indicated by your technical consultant. This file is in the Parser Rules.zip folder and *Supplemental Configuration.zip*.



Import Application Metadata .Zip Files

Zip the folders (*Installer part A, Installer Part B)* and import the .zip files containing the Load Test Analyzer and Load Tester applications using the Load/Extract tool on the Application tab. These files contain the metadata for load testing and analyzing test results.

Installer Part A

Installer Part A.zip contains metadata to set up a Workflow to import and analyze the load testing log files.

Installer Part B

Installer Part B.zip contains metadata to set up the load testing components in a OneStream application.

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. Load Test Suite is designed to work with text files but other file types may be supported in the future.

Update Cube Views

The cube views included in Load Test Suite simulate the users viewing data in the application. The cube view in the *LTS_LoadTest Group* will not have the same dimensionality as the applications being tested, so perform these updates:

- Add valid accounts to the rows, or use shared rows/columns from an existing report
- Update the POV

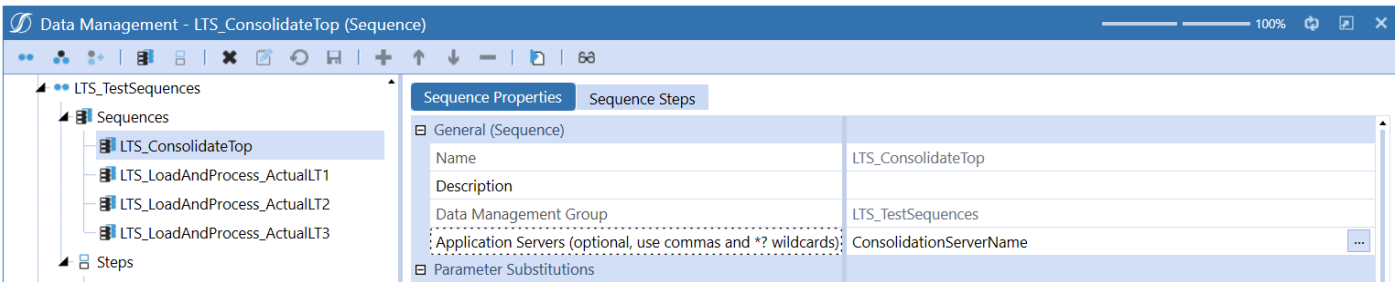
NOTE: Do not change formula in the Entity POV.

- Update BRString Rule variables on the POV tab. These values 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.
  They should return a random Entity each time one is run.

Update Data Management Sequence

Load testing includes running consolidations during testing. These consolidations are called from a Data Management Sequence. Load Test Suite expects all consolidations to perform on one specific server during load testing. To ensure that all consolidations run on the correct server, ensure the Data Management Sequence specifies the name of the consolidation server.

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



Update Data Management Step

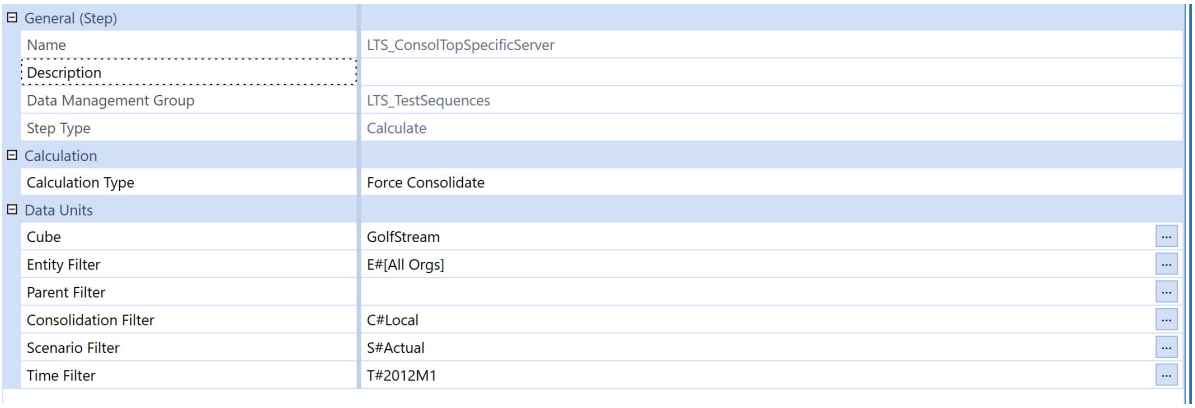
The *LTS_ConsolTopSpecificServer* Data Management Step contains the settings for running a consolidation. The Data Unit settings need to be updated to the correct values for the system being tested.

Cube
Set to the default reporting cube.

Entity Filter
Set to the top level Entity parent in reporting cube.

Scenario
Specify the reporting scenario to consolidate.

Time
Specify the reporting period to consolidate.



⊟ General (Step)	
Name	LTS_ConsolTopSpecificServer
Description	
Data Management Group	LTS_TestSequences
Step Type	Calculate
⊟ Calculation	
Calculation Type	Force Consolidate
⊟ Data Units	
Cube	GolfStream
Entity Filter	E#[All Orgs]
Parent Filter	
Consolidation Filter	C#Local
Scenario Filter	S#Actual
Time Filter	T#2012M1

Test Data Management Jobs

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

If errors occur, ensure the folders are set up and Data Unit sections are set in the Data Management step.

Pre-Built Load Test Suite Tests

Load Test Suite contains pre-built tests consisting of PowerShell scripts and OneStream sequence .xml files. Together, these files allow Load Test Suite to call functions in 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.

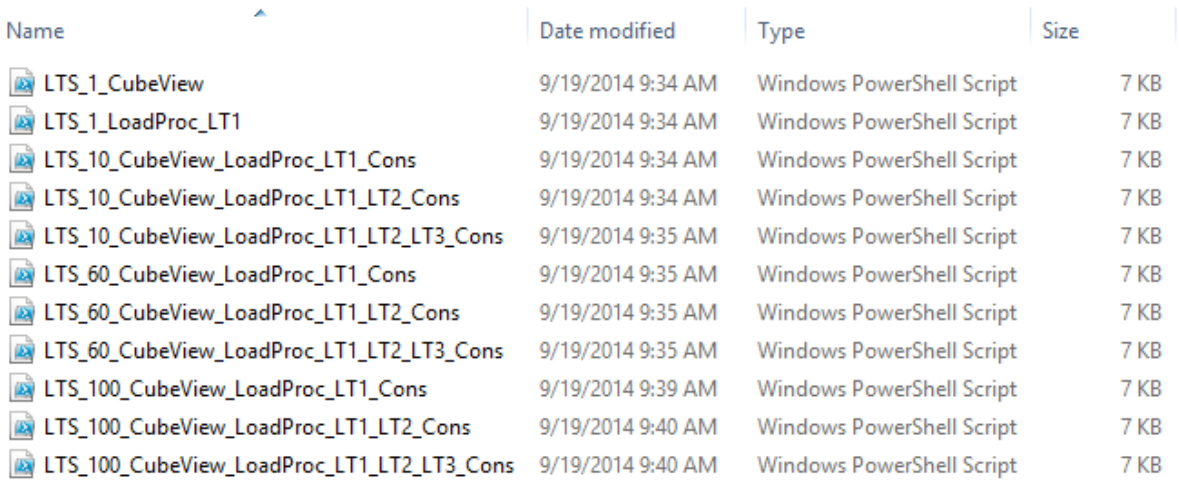
- LTS_1_CubeView replicates:
    1. Ten users calling LTS_ExecuteCubeViews sequence
- LTS_1_LoadProc_LT1 replicates:
    1. One user calling LTS_LoadAndProcessLT1 sequence
- LTS_10_CubeView_LoadProc_LT1_Cons replicates:
    1. Ten users calling LTS_ExecuteCubeViews sequence
    2. One user calling LTS_LoadAndProcessLT1 sequence
    3. One user calling LTS_ConsTopEntity sequence
- LTS_10_CubeView_LoadProc_LT1_LT2_Cons replicates:
    1. Ten users calling LTS_ExecuteCubeViews sequence
    2. One user calling LTS_LoadAndProcessLT1 sequence
    3. One user calling LTS_LoadAndProcessLT2 sequence
    4. One user calling LTS_ConsTopEntity sequence
- LTS_10_CubeView_LoadProc_LT1_LT2_LT3_Cons replicates:

1. Ten users calling LTS_ExecuteCubeViews sequence
2. One user calling LTS_LoadAndProcessLT1 sequence
3. One user calling LTS_LoadAndProcessLT2 sequence
4. One user calling LTS_LoadAndProcessLT3 sequence
5. One user calling LTS_ConsTopEntity sequence

- LTS_60_CubeView_LoadProc_LT1_Cons replicates:
    1. Sixty users calling LTS_ExecuteCubeViews sequence
    2. One user calling LTS_LoadAndProcessLT1 sequence
    3. One user calling LTS_ConsTopEntity sequence
- LTS_60_CubeView_LoadProc_LT1_LT2_Cons replicates:
    1. Sixty users calling LTS_ExecuteCubeViews sequence
    2. One user calling LTS_LoadAndProcessLT1 sequence
    3. One user calling LTS_LoadAndProcessLT2 sequence
    4. One user calling LTS_ConsTopEntity sequence
- LTS_60_CubeView_LoadProc_LT1_LT2_LT3_Cons replicates:
    1. Sixty users calling LTS_ExecuteCubeViews sequence
    2. One user calling LTS_LoadAndProcessLT1 sequence
    3. One user calling LTS_LoadAndProcessLT2 sequence
    4. One user calling LTS_LoadAndProcessLT3 sequence
    5. One user calling LTS_ConsTopEntity sequence
- LTS_100_CubeView_LoadProc_LT1_Cons replicates:

1. One hundred users calling LTS_ExecuteCubeViews sequence
2. One user calling LTS_LoadAndProcessLT1 sequence
3. One user calling LTS_ConsTopEntity sequence

- LTS_100_CubeView_LoadProc_LT1_LT2_Cons replicates:
  1. One hundred users calling LTS_ExecuteCubeViews sequence
  2. One user calling LTS_LoadAndProcessLT1 sequence
  3. One user calling LTS_LoadAndProcessLT2 sequence
  4. One user calling LTS_ConsTopEntity sequence
- LTS_100_CubeView_LoadProc_LT1_LT2_LT3_Cons replicates:
  1. One hundred users calling LTS_ExecuteCubeViews sequence
  2. One user calling LTS_LoadAndProcessLT1 sequence
  3. One user calling LTS_LoadAndProcessLT2 sequence
  4. One user calling LTS_LoadAndProcessLT3 sequence
  5. 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

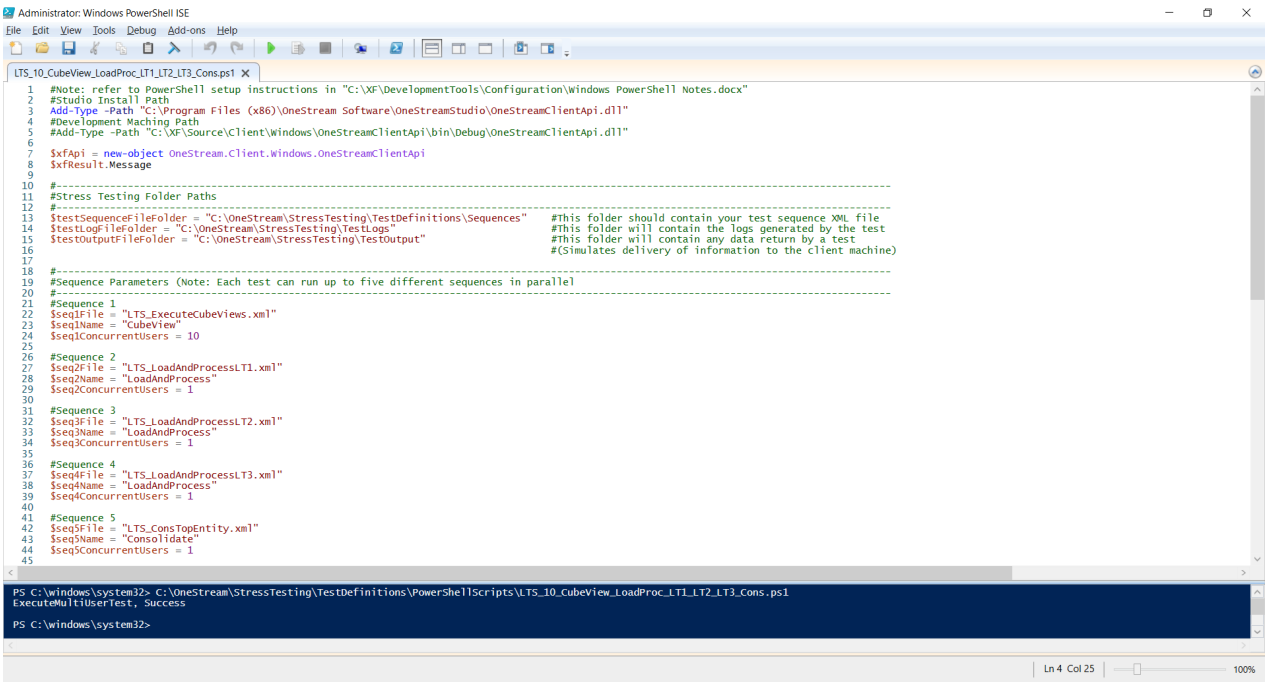
Run Tests

Once you configure Load Test Suite and test the Data Management jobs, run Load Test Suite in Powershell. Open each test script in Powershell ISE and click **Run Script**.

You can retrieve the PowerShell script from:

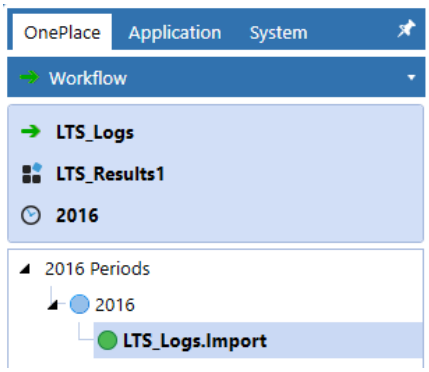
C:\OneStream\StressTesting\TestDefinitions\PowerShellScripts.

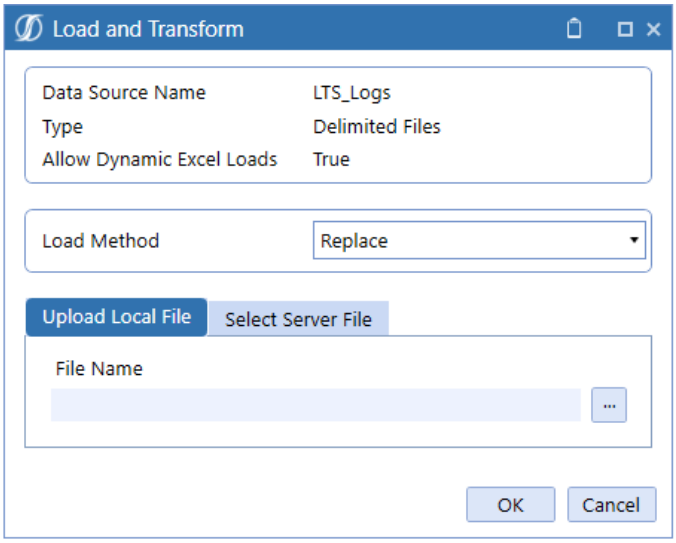
> **NOTE:** PowerShell concurrent user limitation occurs when LTS_LoadAndProcess_ ActualLT1 - LT3 data management jobs and test sequences do not support concurrent users, only one user.



Import and Analyze Results

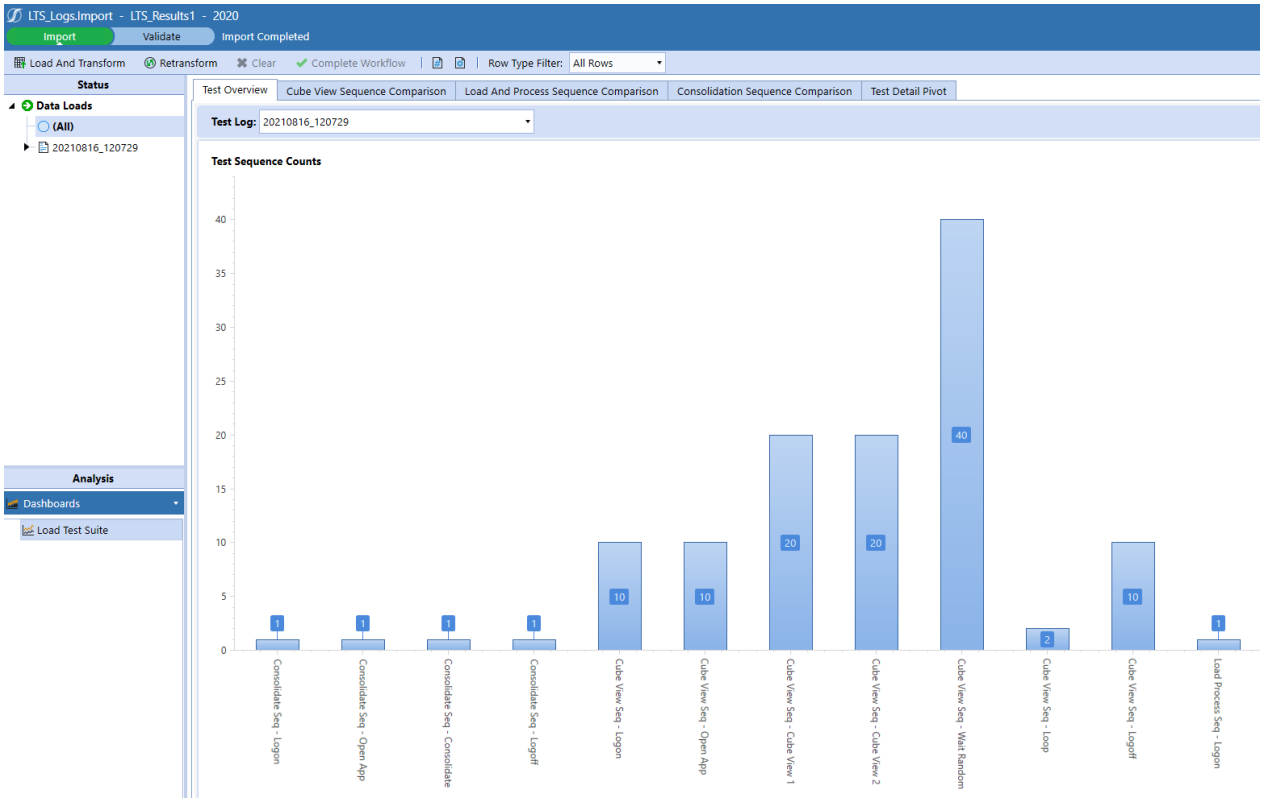
The tests will create log files in the *C:\OneStream\StressTesting\TestLogs* folder after they run which can then be imported into the *LTS_Logs Workflow*. From the import step in the workflow, click **Load And Transform** to upload each .csv file.





After importing the result log files into OneStream, users can review or analyze the results from the Dashboard Analysis section of the Workflow.

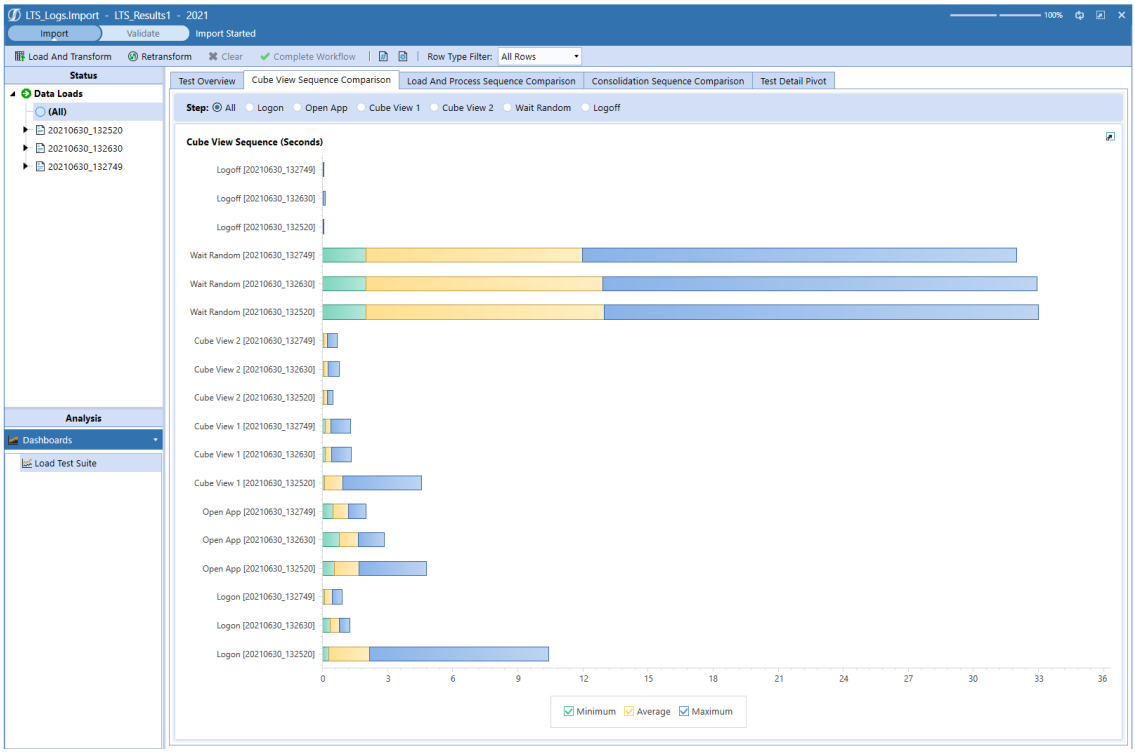
Performance degradation occurs when high concurrent user tests are imported within the same workflow profile, scenario, and year for both the desktop and web clients.



NOTE: If test logs need to be removed or are no longer relevant, highlight the specific test log and click **clear**. This removes the test log from being available in the data load list.

View all Results

By default, the Load Test Suite Comparison tabs will display all results for the test logs imported. To view specific sequence data, select a radio button option from the Step: menu.



Optimal Display Settings

OneStream and MarketPlace solutions frequently require the display of multiple data elements for proper data entry and analysis. Therefore, the recommended screen resolution is a minimum of 1920 x 1080 for optimal rendering of forms and reports.

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.