SolarWinds Task Factory

Created on: 23 August 2022

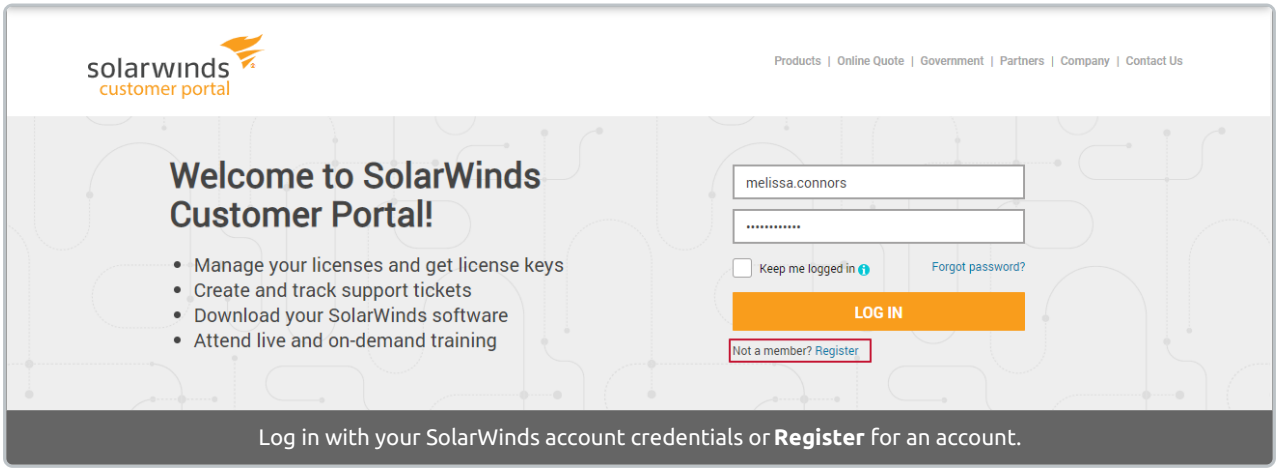
Table of Contents

SolarWinds Account Management

Last Modified on 05 August 2022

SolarWinds Customer Portal

As of version 2021.8, you must use the SolarWinds Customer Portal to manage your account and product licenses.



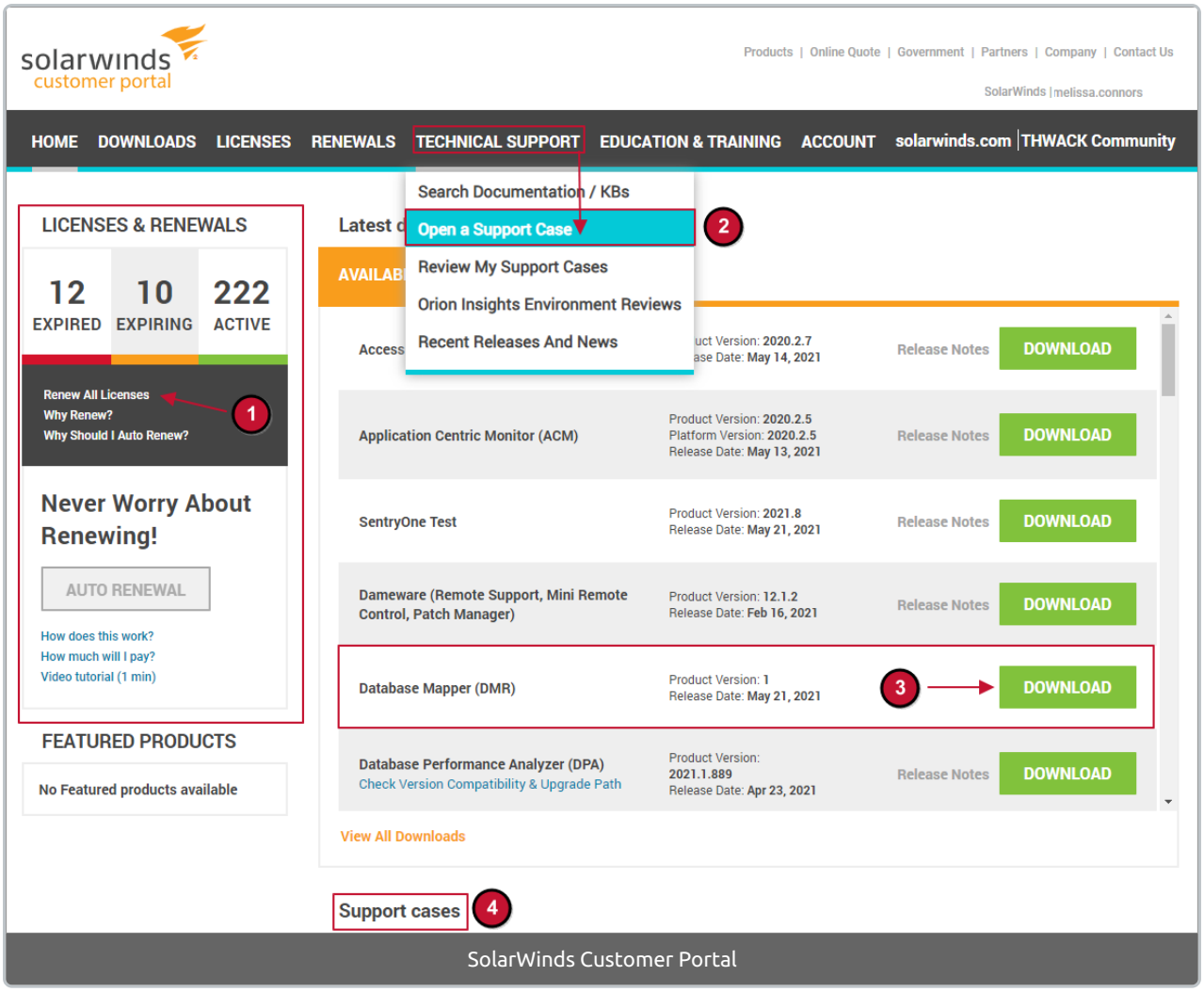
Log in with your SolarWinds account credentials or **Register** for an account.

Additional Information: For more information about the updated product licensing, see the SQL Sentry, Database Mapper, and Task Factory licensing articles.

SolarWinds Self-Service Options

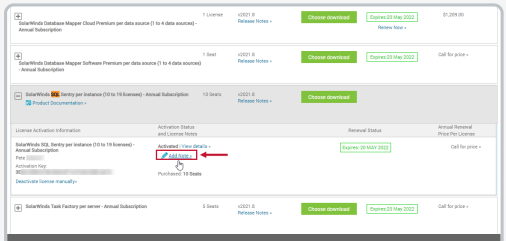
Once you are logged into the SolarWinds Customer Portal, you have a several important self-service options available, including:

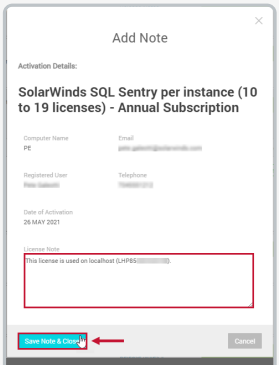
1. **License** and **renewal** information and management options are on the left.
2. The T**echnical Support** option on the top menu provides a link to **Open a Support Case**.
3. Links to product **downloads** and release notes are available in the center.
4. Your **open support cases** appear below the product downloads section.



SolarWinds Customer Portal

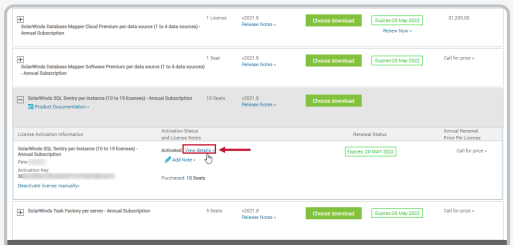
Note: For SQL Sentry licensing, you can use the **Add Note** feature to keep track of the license keys installed on your server installations.


Add Note

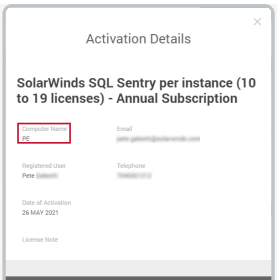


Add Note window

Note: The **View Details** option displays more information about license key activation, including the computer name used for activation.



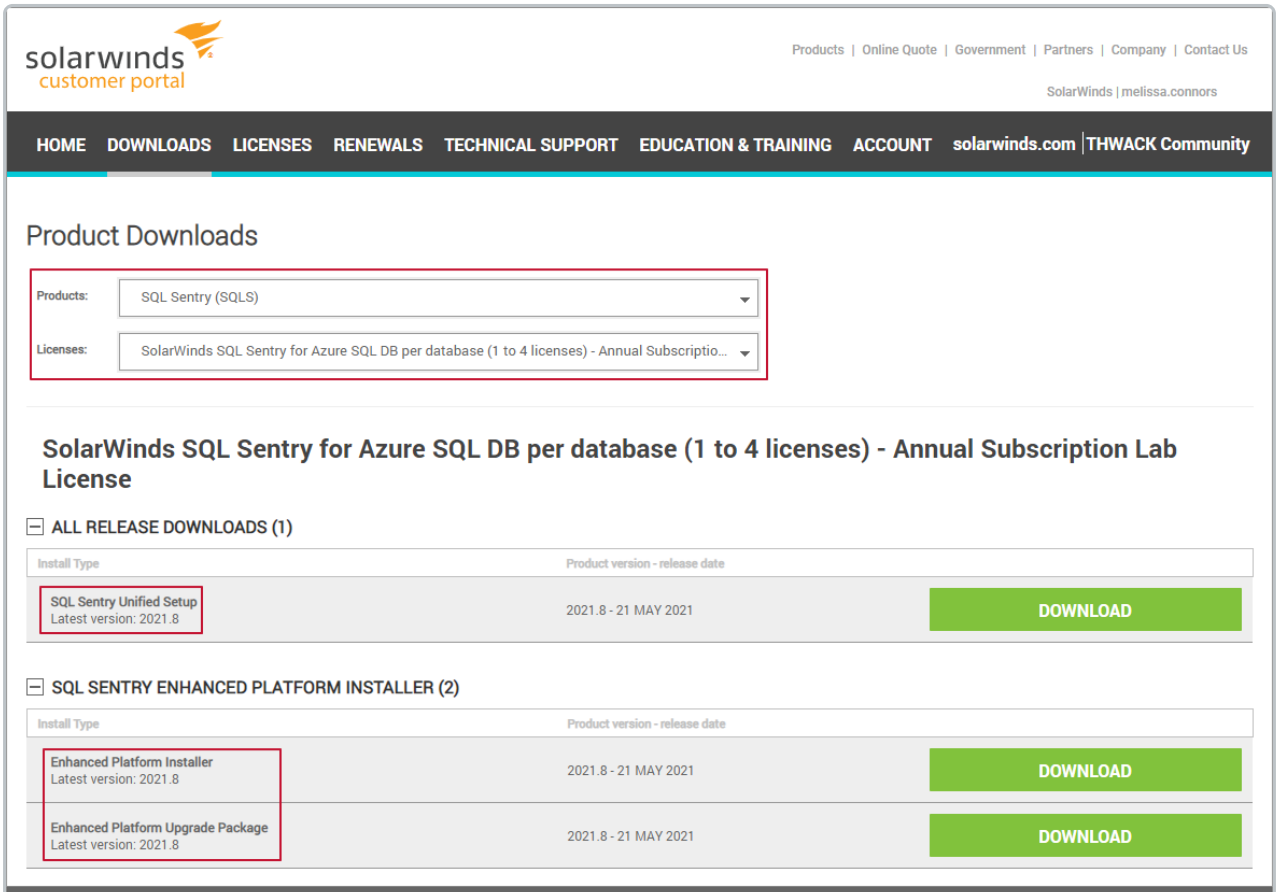
View Details



Activation Details

SolarWinds Product Downloads

On the **Product Downloads** view, select your products and licenses from the drop-down menus to get the links for your products. In the example below, there are multiple downloads for the different installation and upgrade options for SQL Sentry (e.g. *Unified Setup* vs. *Enhanced Platform Installer*).



SQL Sentry product download options

my.sentryone.com

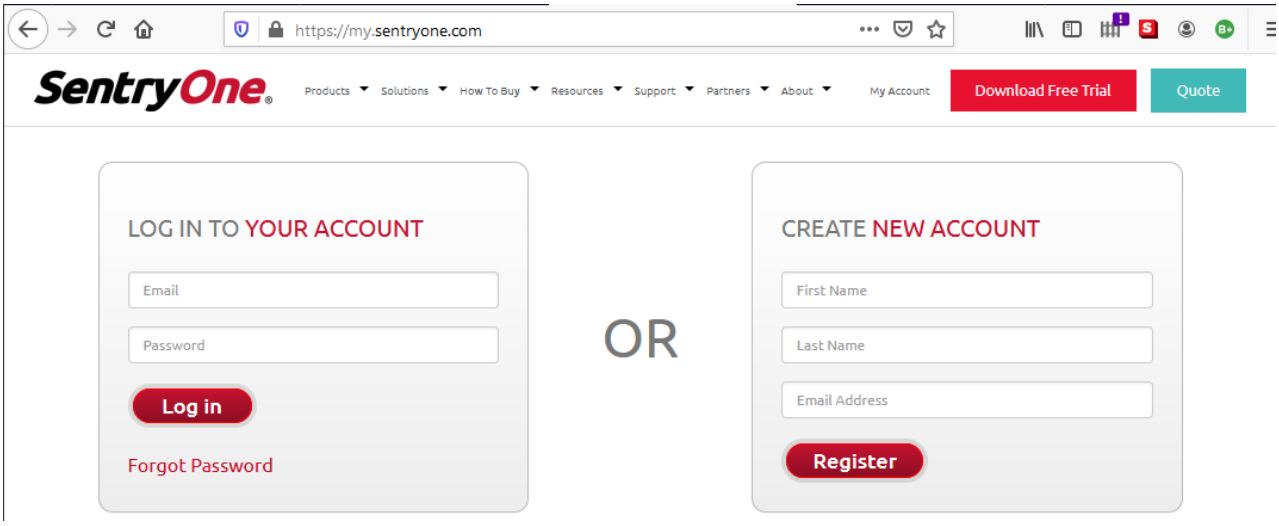
Deprecated: As of version 2021.8, you must use the SolarWinds Customer Portal to manage your account and product licenses.

The my.sentryone.com page described below is only for SentryOne branded products before version

2021.8.

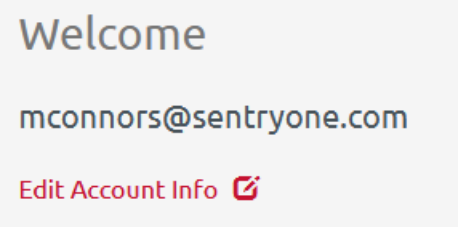
SentryOne Account

Go to my.sentryone.com to log into your account, create a new account, or get help with a forgotten password.



How do I edit my SentryOne account information?

Select **Edit Account Info** from the **Welcome** banner.



The **Edit Account Profile** screen lets you update your name, contact address, email address, and password information.

SentryOne Products

Where do I download SentryOne installation files?

The **Downloads** section contains links to download all the latest versions of products applicable to your account.

Downloads

Product	Version	Released	Links
SQL Sentry Unified Setup	2020.14.2	2020-07-14	Download | Change List | Installation Instructions
Plan Explorer	2020.8.17	2020-05-18	Download | Change List
Task Factory	2020.1.1.0	2020-01-01	Download
Pragmatic Workbench	2020.1.1.0	2020-01-28	Download
Workbench Server	2020.1.1.0	2020-01-28	Download
SentryOne Test Visual Studio Extension	2020.1.1.0	2020-01-28	Download
LegiTest	2020.1.1.0	2020-01-28	Download
SentryOne Document Remote Agent	2020.1.1.0	2020-01-28	Download

Note: If you want to use the Enhanced Platform Installer (EPI) version of SQL Sentry, see the EPI article and SQL Sentry Enhanced Platform Installer table of downloads.

SQL Sentry Enhanced Platform Installer

Product	Version	Released	Links
Enhanced Platform Installer *Command Line Interface *SentryOne Portal Packaged with EPI. Minimum version 20.0 required.*	2020.14.2	2020-07-14	Download | Installation Instructions **See important v20 upgrade information in the Installation Instructions**
Enhanced Platform Upgrade Package *Minimum version 20.0.0 required in order to use this upgrade package.*	2020.14.2	2020-07-14	Download | Installation Instructions **See important v20 upgrade information in the Installation Instructions**

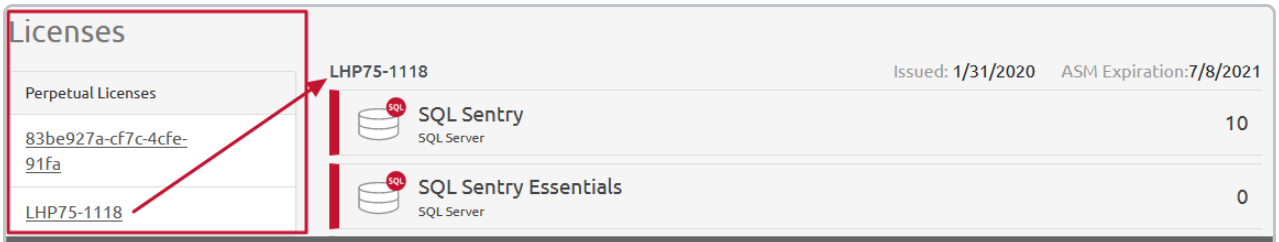
How do I get the AWS Marketplace and Azure Marketplace images for SentryOne?

Select the **AWS Marketplace VM** or **Azure Marketplace VM** link from the **Resources** menu on the sidebar.

SentryOne Licensing

Where can I view my SentryOne licensing information?

On the **Licenses** sidebar box, there are options for **Perpetual Licenses**, **Evaluations**, and **Subscriptions**. Select the applicable category, then select the license key to get additional information.



Example of a license for LHP75-1118 to monitor 10 targets with SQL Sentry

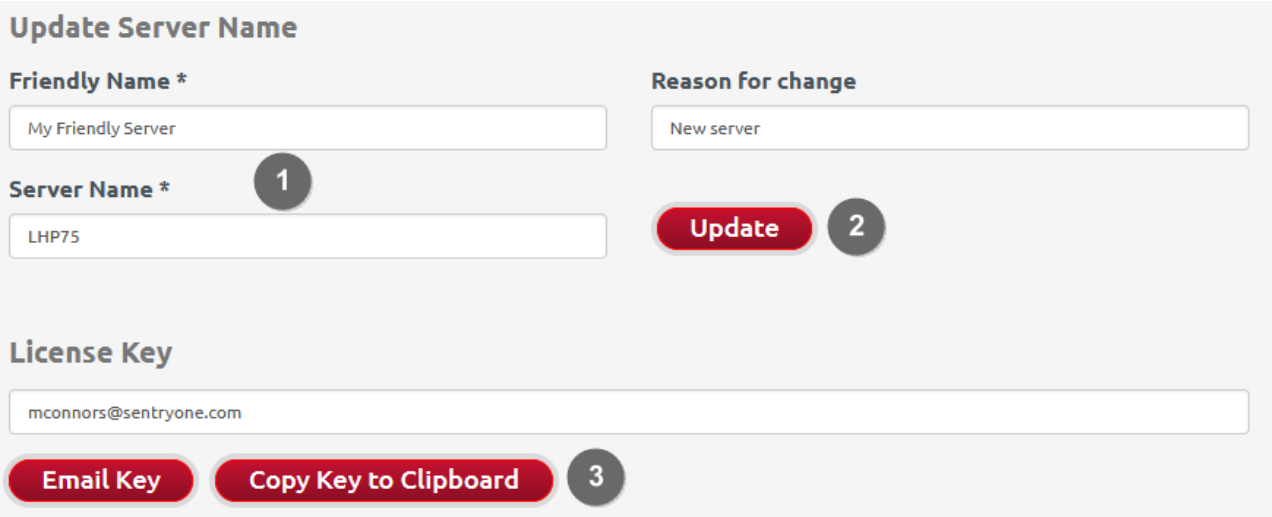
How do I add subscriptions to my SentryOne account?

Select **Subscriptions** from the Licenses sidebar box. You will need to contact the SentryOne sales team if your account is not set up for subscription-based licensing.

How do I update the server name on a SentryOne license key for SQL Sentry?

Select the applicable **license key** from the **Licenses** sidebar box, then:

1. Go to the **Update Server Name** section and enter the new **Friendly Name** and **Server Name** for the license key. You may optionally enter a reason for requesting this change.
2. Select the **Update** button.
3. Select **Email Key** or **Copy Key to Clipboard** to get a copy of the license key text.



Additional Information: See The Hardware Key and Applying a New License sections of the License Management article for additional instructions on using the new license key.

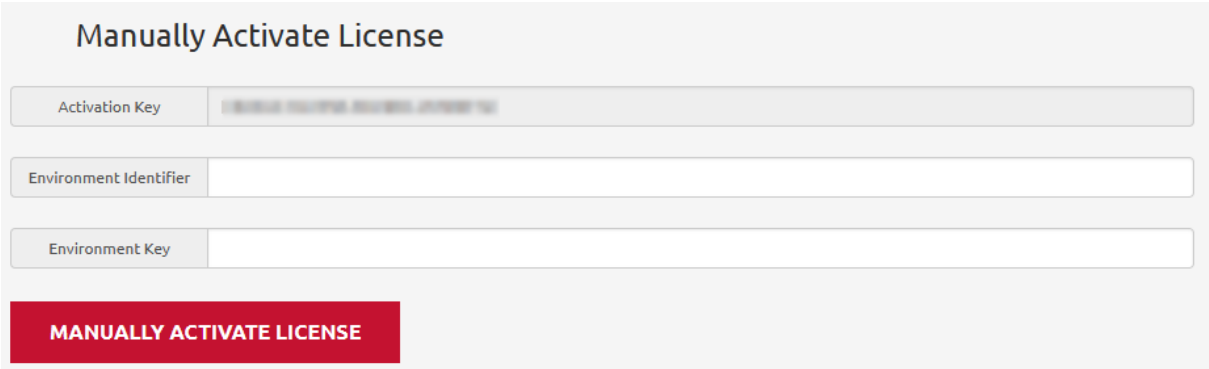
How do I manually activate a license for SentryOne Task Factory, SentryOne Document, or Workbench products?

This option is needed for machines that do not have internet access to automatically activate a license during product setup.

1. For SentryOne Document, Task Factory, SentryOne Test, BI xPress, DBA xPress, DOC xPress, LegiTest,

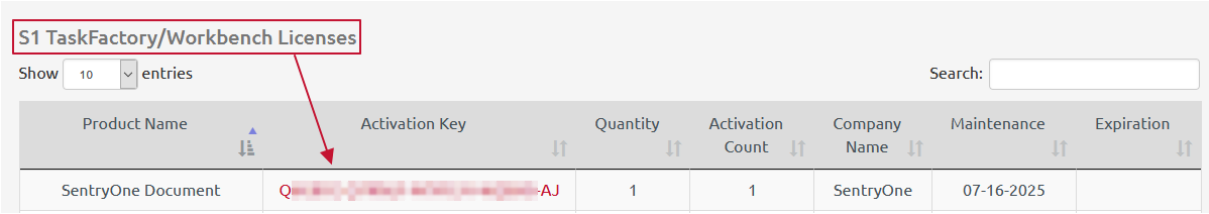
DTS xChange, and Pragmatic Workbench products, go to the **S1 TaskFactory/Workbench Licenses** section.

2. Select the applicable **Activation Key** from the table. This opens the **Manage License** screen.
3. Go to the **Manually Activate License** section.
4. Enter the **Environment Identifier** and **Environment Key** for the license you are activating.
5. Select the **Manually Activate License** button.

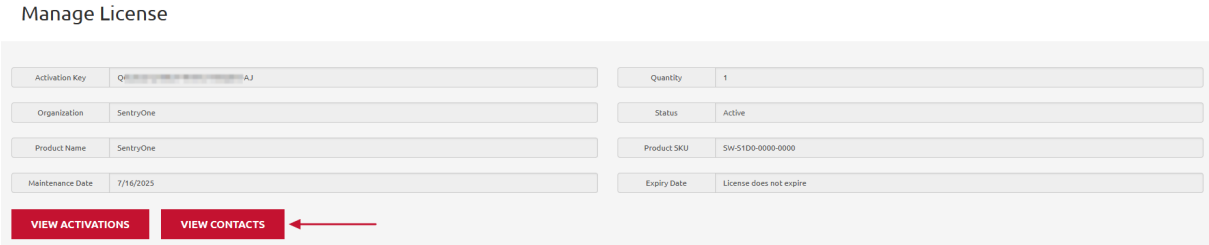


How do I manage SentryOne license contacts?

1. For SentryOne Document, Task Factory, SentryOne Test, BI xPress, DBA xPress, DOC xPress, LegiTest, DTS xChange, and Pragmatic Workbench products, go to the **S1 TaskFactory/Workbench Licenses** section.
2. Select the applicable **Activation Key** from the table. This opens the **Manage License** screen.



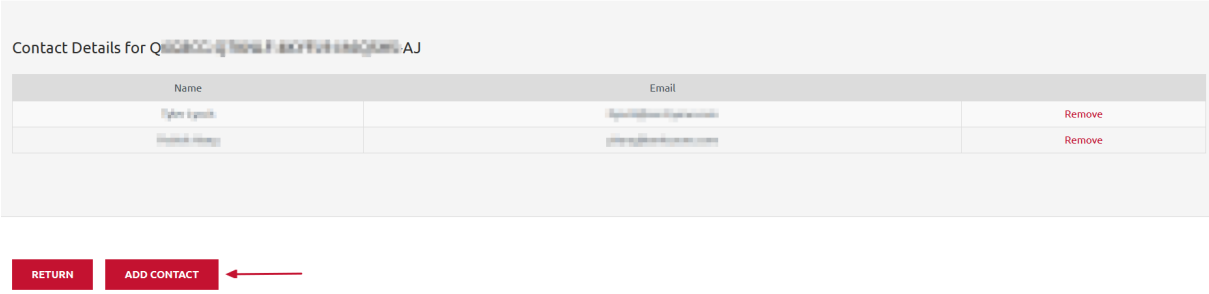
3. On the **Manage License** view, select **View Contacts**.



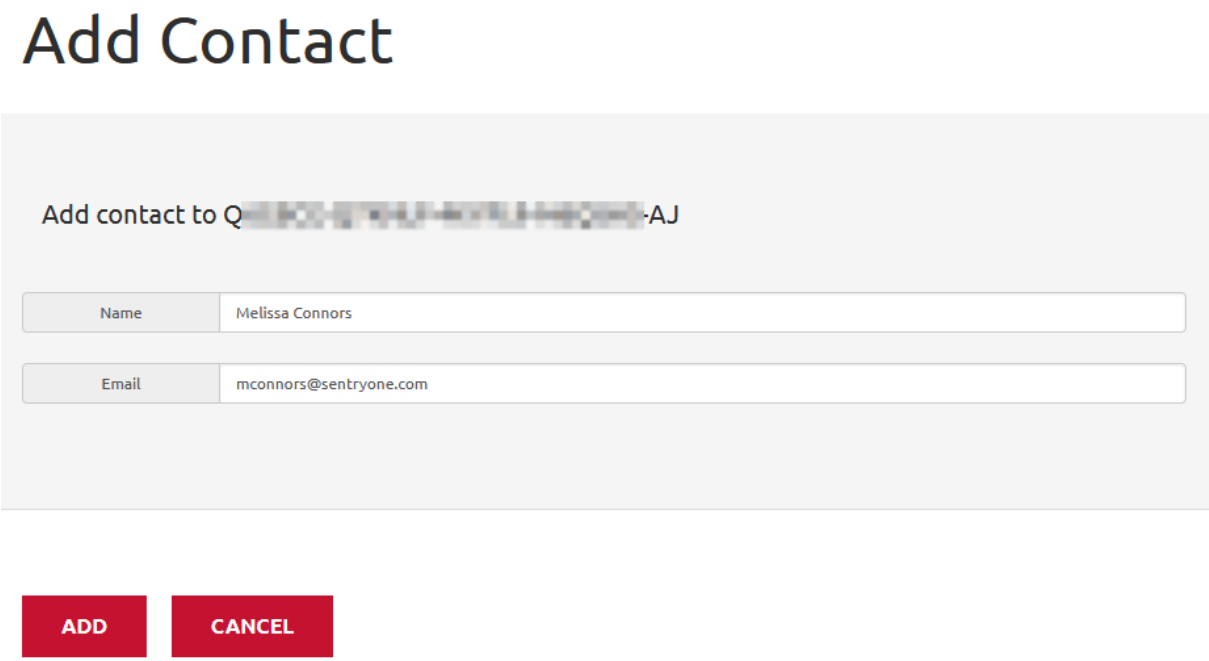
From the **Contact Details** view, you can **add** or **remove** contacts.

1. To add a new contact, select **Add Contact** from the **Contact Details** view.

Manage License



2. On the **Add Contact** form, enter the **Name** and **Email** information, then select **Add**.



> ⚠ **Important:** For Database Mapper (previously known as SentryOne Document), you must add users as contacts to the license in addition to adding them as users to Organization Settings. The **Name** and **Email** must match what is entered in **Organization Settings**.

How do I view license activation history for SentryOne Task Factory, SentryOne Document, or Workbench products?

1. For SentryOne Document, Task Factory, SentryOne Test, BI xPress, DBA xPress, DOC xPress, LegiTest, DTS xChange, and Pragmatic Workbench products, go to the **S1 TaskFactory/Workbench Licenses** section.
2. Select the applicable **Activation Key** from the table. This opens the **Manage License** screen.
3. Select **View Activations** from the **Manage License** screen. This displays the **Environment Key**, **Date Last Activated**, and **Environment Identifier information**.

Task Factory Release Notes

Last Modified on 23 August 2022

For more information on the additional steps SolarWinds is taking to improve our security posture and policies, please reference this blog by SolarWinds President and CEO Sudhakar Ramakrishna.

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Download: For more information about Task Factory pricing or downloading a trial, see the Task Factory product page.

2022

Version 2022.3 [August 23, 2022]

Features

- **63497**: **Rest Task** - Add functionality to just read response headers.
- **68456**: **Rest Source** - Add ability to parse headers in the Pagination tab.
- **68864**: **Dynamics Connection Manager**- Implement OAuth2 authentication type.
- **69159**: Updated the Task Factory certificate.

Fixes

- **6853**: **TF Compression Task** not producing a valid error stack.
- **6872**: **Delete or Move Email Source Messages Task** - Unable to select Email OAuth2 option in connection manager dropdown.
- **67253**: **Email Source** - Reading DateTime variable incorrectly in German Format.
- **68164**: **Excel Source** - Add support for "Special" formatted cells.
- **68637**: **Secure FTP** - Unable to download full file.
- **68773**: **TF Surrogate Key Transform** not saving custom seed value
- **68775**: **Secure FTP** - Error - "AutoFix: Using implicit SSL/TLS because port is 990."
- **68810**: **TF Secure FTP** - Fixed Cipher Issues.

2021

Version 2021.18 [September 20, 2021]

Features

- 68142: Add support for Task Factory on Azure Data Factory.

Fixes

- 66781: **Secure FTP** - Implement binary download in FTPES.
- 66872: SF Case 00066610 Excel Connection v20.21.0.
- 68071: Duplicate Connection Managers in Advanced Derived Columns causing issues.
- 68134: **Secure FTP** - IBM Mainframe server is hanging on new versions of Task Factory.
- 68228: **Dynamics CRM Connection Manager** - error "EnableRegionalDisco" is disabled.
- 68480: **Salesforce Source** - Bulk mode not including columns from convertCurrency() or format().

Version 2021.8 [May 25, 2021]

Notes

- This release is for the rebranding of SentryOne Task Factory to **SolarWinds Task Factory**. In addition to logos and icons being updated, the licensing has been switched to SolarWinds licensing. Starting with version 2021.8, your license and software downloads are available through the SolarWinds Customer Portal. Older versions (prior to the 2021.8 rebranding) are still managed through your SentryOne account.

Version 2021.2.3 [April 8, 2021]

Notes

- This release includes changes for the REST OAuth2 Connection Manager as described below.

Fixes

- 67443: **REST OAuth2 Connection Manager**
  - *Could not unlock OAuth2 component* error in version 2021.2

Version 2021.2 [March 9, 2021]

Notes

- There's an updated EULA for this version.
- This release includes changes for Salesforce Source and Secure FTP as described below.

Features

- 66414: **Salesforce Source**
  - Add support for SOQL *toLabel(fields)*
    - **Additional Information:** See the SOQL and SOSL Reference guide from Salesforce.

Fixes

- 67026: **Secure FTP**
  - Failure to connect to mainframe server using binary.
- 64017: **Salesforce Source**
  - Expected result when traversing through two relationship queries is incorrect when in *Normal* mode.
- 63808: **Salesforce Source**
  - Not accepting rows with special characters when using *Bulk Query* mode.
    - ℹ **Note:** There was no issue in this scenario when setting to *Normal* mode.

2020

Version 2020.21.2 [November 23, 2020]

Fixes

- 66450: Issue saving Dynamics Connections Managers

Version 2020.21 [October 29, 2020]

Fixes

- Fix 61110: Date displaying incorrectly in preview mode of Salesforce source with DB_Date as data type.
- Fix 64814: Rest Source - error " *Last Page Detection in the Pagination settings is empty*" when setting Last Page Detection to End paging when response contains error message
- Fix 65300: Incorrect date being read from TF Excel

Version 2020.18 [August 25, 2020]

Features

- Feature 64637: Implement OAuth2 for Office365 Email

Fixes

- Fix 64909: Excel Reader not reading column values in tags
- Fix 64873: Pack Data Transform - " *Cannot remove a table that has existing relations*" error when looping
- Fix 64908: Google Drive Connection Manager - error when trying to get a Refresh Token " *Could not acquire connection*"

Version 2020.11.1 [May 28, 2020]

Features

- Feature 63930: Added check for .NET 4.5 requirement to installation package
- Feature 63970: Add Basic Authentication to Token Getter

Fixes

- Fix 55131: Pack data Transform - not resetting after a loop - error - item with the same key has already been added
- Fix 61870: Address Verification - ExecutionEngineException with USPS engine
- Fix 63231: Excel Source handling date formats with asterisks incorrectly
- Fix 63828: Advanced Derived Column Transform - When you delete the error output constraint the component will still be set to "Redirect Row" even if you change the option in "Advanced"

Version 2020.1.4 [April 7, 2020]

- Bug fixes and technical improvements

Version 2020.1.3 [March 24, 2020]

- Item 62641: Update EULA

Features

- Feature 61583: Include Dimension Merge in Task Factory ADF
- Feature 61685: Update Salesforce to use latest API
- Feature 61989: Update Chilkat to latest version
- Feature 62301: Add cookie container to REST Task
- Feature 63050: Add support for reading Salesforce Source / Destination / Connection from SSIS Provider

Fixes

- Fix 58658: Google Cloud config file encoding scopes twice creating a %2520 causing a failure for invalid_scope
- Fix 62062: Advanced Derived Column Transform - error - Unsupported format [J] when trying to get Julian Date from GetDatePart()
- Fix 62735: Dimension Merge SCD - 2019 doesn't work because of missing Microsoft.SqlServer.Types reference
- Fix 62555: Task factory- Do not remove Newton Soft 7
- Fix 62602: SharePoint Source - Remove IDs From Results option causes certain characters to get excluded from the results
- Fix 62632: Help links do not work for some components
- Fix 63146: Getting object reference error when using the upload directory to server action
- Fix 63242: Email Source - Attachments not showing up in Attachments column when email has a logo

Version 2020.1.1 [February 11, 2020]

Features

- Feature 59979: Excel Destination - Added the ability to create an Excel file in the Excel Destination.

Fixes

- Fix 60394: Unpack Data Transform - sorting by *Column Index In Data*
- Fix 60584: Rest Source - variable that has hard coded value doesn't show a preview when applied to a Token Path for a column
- Fix 60664: SFTP - Remote File Directory is duplicated when using *Reuse same connection*
- Fix 60842: TF Rest Destination - Capture group numbers must be less than or equal to Int32.MaxValue
- Fix 61057: Terminator Destination - selected columns that have to be scrolled to are no longer selected when component is closed
- Fix 61670: RefreshTokenHeaders are not being set correctly when *RefreshToken* is called in OAuth2 connection manager
- Fix 61880: Allow Excel Destination with no source attached

2019

Version 2019.4.4 [December 19,2019]

Features

- Feature 59949: USPS Address Verification - Added new columns for the LI attribute

Fixes

- Fix 58534: Could not find file **C:\Program Files (x86)\SentryOne\Task Factory\TimezoneDB** using default database.
- Fix 59955: Secure FTP - SFTP Use binary transfer option needs to be brought over to Chilkat
- Fix 60840: Update EULA

Version 2019.4.3 [November 23, 2019]

Fixes

- Fix 60249: Fixed issue with SoftSecure and ADF

Version 2019.4.2 [October 24, 2019]

Features

- Feature 53552: Allow Sharepoint Destination to map more than just ID column when using a Delete action

- Feature 58161: Rest Source - Add support for variable syntax in the headers tab to be <@User::VariableName>
- Feature 58162: Unpack Data Transform and Placeholder Source - add support to auto populate Code Page and Length for certain data types

Fixes

- Fix 57915: Excel Destination - DT_NTEXT columns not being written into Excel Destination only in VS - works outside
- Fix 58781: Data Cleansing Transform failing with String not recognized as Boolean data type when there is no Boolean data type from Source
- Fix 59002: Debug not set for Email Source that results in component failing

Version 2019.4.1 [September 24, 2019]

Features

- Feature 46648: New password protection encryption types have been added to the Compression Task
- Feature 51559: New Execute Powershell Task
- Feature 53027: Deprecated LinkedIn, Marketo, and Hadoop components
- Feature 56417: Components that use DEBUG converted to use the Debug property
- Feature 56431: Return non-zero activation failures for licensemanger.commandline
- Feature 56453: USPS Address Verification replacing the encrypted version(s) with individual unencrypted versions
- Feature 56460: USPS Address Verification Small Town Default change
- Feature 56743: Rest Task - add support for sensitive parameters
- Feature 56744: Dynamics CRM Destination - show primary key on UI
- Feature 57779: SFTP connection manager

Fixes

- Fix 56295: Dynamics Source crashes when clicking Get Changes without an entity selected
- Fix 56716: Advanced Execute Process Task Error with valid cmdline after upgrade from 2018.4.5. to 2019.1.2.
- Fix 56742: SFTP error - Number of bytes in FileName is not equal to the number of bytes requested
- Fix 57466: Secure FTP "Authentication failed or partial success" after updating to 2019.1.3
- Fix 57480: Hash Columns transformed to Null once they pass through Error Output Destination
- Fix 58352: Fix issue with salesforce destination debug mode

Version 2019.1.4 [August 30, 2019]

Features

- Feature 53026: Update Facebook and Twitter configuration files
- Feature 53527: Rename "Download Task" to "HTTP File Download Task"
- Feature 56965: PGP Task doesn't default to Cast5

Fixes

- Fix 46636: XML Output Destination - TFREQUIRED is opposite of what is expected based on the XSD
- Fix 51513: Rest OAuth2 the Connection Manager's Token Getter resizes the textboxes when populating with a long app secret
- Fix 52012: Rest OAuth2 Connection Manager not encoding URLs
- Fix 52970: Invalid Refresh Token causes Google Drive Source's UI to corrupt
- Fix 53182: XML Output Destination - omit unmapped elements
- Fix 53914: Dynamics Source requires validation to ensure at least one column is selected for output
- Fix 54329: Update variable syntax in the Dynamics Source
- Fix 54784: Unable to use drop-down to select a country in Advanced Email and SMS Task
- Fix 55416: Advanced Derived Column Transform - error - Microsoft.SQL.DTS.Runtime.Wrapper.IDTSVariables100 - when targeting 2017 with later version of SSIS installed
- Fix 55455: File Properties Task File Contents - error - File contents property not set because file is larger than 50 MB
- Fix 55729: Rest Task - error when trying to open the component after deleting the connection manager
- Fix 55892: "No results returned" when filtering for a SharePoint List using an underscore in the SharePoint Source component
- Fix 56178: Rest OAuth Connection Manager not allowing creation of connector

Version 2019.1.3 [August 5, 2019]

Fixes

- Fix 55044: Salesforce Source invalid cast error

Version 2019.1.2 [June 21, 2019]

Fixes

- Fix 52014: Dynamics CRM - Update all timeout bindings to obey manually set timeout in connection manager
- Fix 53021: Improve Salesforce Source performance

Version 2019.1.1 [June 4, 2019]

Features

- Feature 44762: Update Bouncy Castle for Task Factory

- Feature 46645: Add Functionality to the TF File Gateway Task to allow no character to be used to denote Text Columns
- Feature 46657: Allow Rest Source to Ignore Response Errors
- Feature 47214: Email Source configuration attachments to a single directory option
- Feature 47351: Add cipher selection to PGP Task
- Feature 48823: Set Rest Destination Valid Status Codes to match Rest Source
- Feature 49596: SalesForce Source Supports Nested Queries
- Feature 49794: Integrate Telemetry Logger with license check
- Feature 50010: Excel Power Refresh connection managers can now filter for xlsm files
- Feature 50114: Update Placeholder Source to use WPF
- Feature 50140: Modify TF Advanced Lookup Cache Connection Manager, checkboxes to radio buttons
- Feature 50239: Add TokenStore options to OAuth2 Connection Manager
- Feature 50526: Add Support For SSIS 2019 / Visual Studio 2019
- Feature 50605: Update Timezone Conversion DB's
- Feature 50607: Update Salesforce Soap API to 45
- Feature 51528: Convert FTP from edtFtpNetPro to Chilkat

Fixes

- Fix 42369: Advanced Package Execution throws a "Method not found" error when attempting to execute a package located in the SSIS Catalog from Visual Studio 2017
- Fix 43867: TF Advanced Lookup Cache Connection Manager - Package Upgrade Failure
- Fix 44168: TF XML Generator Transform is crashing Visual Studio after selecting an XSD file.
- Fix 45318: TF Excel Destination is corrupting the Excel file when attempting to insert into style template
- Fix 46297: Dynamics CRM Source error when custom Option Set is deleted and orphaned record still exists
- Fix 46654: TF NULL Handler is locking the memory of NTEXT columns, when attempting to convert blank values to NULL
- Fix 46656: TF Dynamics CRM - Fix connecting through web content proxy
- Fix 46892: Changes to filter can damage Fact Table Destination
- Fix 46894: TF Advanced Execute Package - urn could not be resolved when using 'out of process' Execution Mode
- Fix 47496: TF Data Flow Nugget - Object reference not set to an instance of an object when using a multicast
- Fix 47539: Error Output Description throws Bad Lineage ID error if data passing through component contains colon
- Fix 48652: Dynamics CRM connection manager unable to connect to NA2 instance
- Fix 48896: Advanced Installer not replacing files
- Fix 48964: Hash Transform can't reorder the columns

- Fix 48975: Advanced Aggregator Transform component breaks if more than one Group By column is selected
- Fix 49572: Task Factory installer's uses "Sentry One" for the directory and start menu location instead of "SentryOne"
- Fix 49668: Update Chilkat for SFTP
- Fix 50221: Adv Lookup cache conn mgr has :MEMORY: in the custom cache location's textbox
- Fix 50746: Update address verification license details to replace with "any application" / "task factory" check
- Fix 51142: Rest Task is stripping out \r characters from the postbody property
- Fix 51369: SalesForce Source tool unexpectantly changing dates when pulling 'date' values in local time
- Fix 51514: SCD Component does not display boxes for Row Change Detection Tab on RDPt
- Fix 51583: Advanced Execute Process Task does not recognize Japanese Characters
- Fix 51640: Upsert Destination throws obj reference error when creating the Insert Row Count Variable
- Fix 51869: The Address Key is not being returned by the API in the 4.1 Address Verification Database Update
- Fix 52018: Email Source is not filtering correctly with "contains"
- Fix 52388: Rest Oauth2 Connection Manager - refresh tab not sending headers with request

2018

Version 2018.4.3 [February 4, 2019]

New Features:

- New 45349: Address Verification can now use connection managers

Updated Features:

- Update 45961: Hash Transform can now output to Byte Arrays
- Update 45664: SalesForce Connection Manager now Includes Retry Settings

Fixes:

- Fix 45491: Advanced Lookup now allows DT_IMAGE columns to be outputted
- Fix 45819: After Upgrading to Visual Studio 2017 15.9 or later, Task Factory Components do not show up in SSIS Toolbox
- Fix 47170: Dynamics CRM Source throws errors when "Get Changes" is selected
- Fix 47185: Rest Source fails to output DT_TimeStamp data type
- Fix 46508: Email Source incorrectly shows 'Could not set Column' as informational message instead of warning
- Fix 46605: Upsert Destination fails in row by row mode when a DT_IMAGE column is present
- Fix 46964: SalesForce Sources throws error when switching to an invalid connection manager

Version 2018.3.3 [August 9, 2018]

New Features:

- New 1232: Task Factory now supports installing on Server Core installations
- New 1394: Oracle Upsert Destination includes the ability to define temp tables
- New 4030: REST Source connection manager now includes support for Survey Monkey configuration files
- New 8242: Amazon S3 Task
    - Manage your S3 environment using the S3 Task
- New 11962: Unpack Data Transform now supports XML unpacking without needing a root xpath query
- New 11992: Rest Source now supports OAuth2 certificate authentication
- New 13928: Rest Task
    - Pull results from an endpoint and store it within a variable for pre-processing data
- New 15200: Rest Source and Unpack Data Transform now allows empty strings to be used within JSON token paths
- New 15652: REST connection manager now allows specifying the security protocol
- New 16131: ADF Licensing for TF
- New 16289: REST Source and Destination can now return header values within the Data Flow
- New 16300: Rest Source can now ignore certification validation errors during authentication
- New 17204: Allow REST JSON parser to ignore formatting on strings that contain dates
- New 17238: Azure Storage Source
    - Supports Blob, Table and File modes for Azure Storage
- New 17239: Azure Storage Destination
    - Supports Blob, Table and File modes for Azure Storage
- New 17585: Address Verification improvements
    - Improved encryption security
    - Change the name of the "Invalid Address Output" to "Unverified Output" to limit confusion
    - Update the Response Status table to include missing response codes
- New 18136: Amazon S3 Source
    - Extract data from S3 using the Task Factory Amazon S3 Source component
- New 18137: Amazon S3 Destination
    - Load data to S3 using the Task Factory Amazon S3 Destination component

Updated Features:

- Update 16287: Improved documentation within Compression task to include descriptions of which usage types are allowed for each mode
- Update 17020: Improved documentation for the Advanced Execute Package task
- Update 17076: Advanced Email and SMS provider list
- Update 17408: Secure FTP's 'Stop package on failure?' property is now set to 'True' by default
- Update 17439: Columns added to Rest Source's output should include incremental numbers to avoid

same-name columns
- Update 17607: Update to Task Factory's Software Requirements
- Update 17729: Improved documentation for licensing requirements on clustered servers

Fixes:

- Fix 1186: Advanced Aggregate Transform is not passing Sort properties through
- Fix 4003: Advanced Aggregator Transform ignores the group by command in provided package
- Fix 16044: Salesforce Source returns NULL values when subquery is used to define a column in SOQL query
- Fix 16063: Replace Unwanted Character Transform not escaping hyphens correctly
- Fix 16320: Marketo Source concatenates the access_token in the URL
- Fix 16322: SalesForce Source incorrectly previews data within a dot notation query
- Fix 16341: Oracle Upsert cannot create temp destination tables when using row flag compare
- Fix 17018: Excel Source throws an "Index was outside the bounds of the array" error for a cell that contains 0.0%
- Fix 17023: Email Source returns NULL values from Exchange Mail server
- Fix 17069: When executing a Task Factory package from a LegiTest Group on a machine without a Task Factory license, the package throws a "Task Factory is not regiserted on this machine" error
- Fix 17241: When the Data Cleansing Transform input's NULL values into a Date column, the NULL value is unable to be input into the SQL Table
- Fix 17243: Dimension Merge's output column selection isn't displaying any of the columns
- Fix 17475: Advanced Lookup throws "System.FormatException: Input string was not in a correct format"
- Fix 17481: Dimension Merge transform that uses a composite key with bit values throws "Duplicate Business Key in Source_System: NullKeyStruct" error
- Fix 17482: The Output Column CarrierRouteRateSortIndicator from the Address Verification Transform outputs data that differs from the documentation
- Fix 17491: Filter Rows transform's expression causes data corruption within Excel Destination
- Fix 17610: Dynamics CRM Source creates endless loop if the batch size is smaller than the row count received
- Fix 17615: Upsert Destination freezes when handling the properties for tables containing large amounts of columns
- Fix 17647: When decompressing a GZip file to a folder destination, the Compression task throws "Could not decompress" error
- Fix 17731: Excel Source not reading certain formats
- Fix 17871: When attempting to view the metadata for the LoginHistory object, Salesforce Source throws an "Object Reference" error

Version 2018.2.3 [April 27, 2018]

> ⚠ **Important:** The 2018.2.3 release for Task Factory will be the last supported release for SQL Server 2008 and 2008 R2.

New Features:

- New 16092: Improved the Unpacked Transform's ability to handle errors when NULL or empty data is passed into it
- New 16265: Added a select all columns option to the Error Output Description.

Updated Features:

- Update 16064: SFTP Task has been updated to the latest version of ChilKat

Fixes:

- Fix 15093: LinkedIn Social OAuth Connection Manager is out of date
- Fix 15696: Advanced Conditional Split Transform does not save column description
- Fix 15705: Advanced Email and SMS task is truncating emails with top-level domains longer than 4 characters
- Fix 15718: Fact Table Destination fails when mapping multiple destination columns to single lookup
- Fix 15765: Salesforce Destination throws "component has detected potential metadata corruption" when upgrading from BIDS 2008 to SSDT 2017
- Fix 16007: Task Factory does not install within the correct directories for Visual Studio / SSDT 2017
- Fix 16010: Unpack Data Transform throws HRESULT: 0xC00470B2 error when selected a column containing packed data
- Fix 16038: Unpacker converts values to Date format before setting to a string data type
- Fix 16046: Password fields in legacy Task Factory connection managers cause Visual Studio to freeze before resolving cursor in password field
- Fix 16050: Advanced Email and SMS task doesn't save the connection manager information when renamed
- Fix 16078: LinkedIn / REST OAuth 2.0 connection manager proxy information is not copied to the Get Token form
- Fix 16080: NULL Handler Transform does not allow XML as a replacement value
- Fix 16238: Dynamics Destination does not display an error when there are no column mappings
- Fix 16271: SharePoint Source and Destination throws Object Reference Error when selecting a list before a connection manager
- Fix 16272: Fact Destination closes when clicking the destination column mappings before a connection manager
- Fix 16274: Marketo Destination throws "Object Reference Error" when selecting a destination object before a connection manager
- Fix 16286: Pagination Settings continue to make calls to endpoint when Last Page Detection is set to

'End when result set contains zero rows' and no rows are returned to REST Source

- Fix 16292: LinkedIn Source's "View Api Document" link no longer directs users to an invalid webpage.
- Fix 16308: Fact Table Destination throws an ArgumentException error when using a variable to do a lookup more than once
- Fix 16417: Packdata Transform inconsistent output between DT_TEXT and DT_NTEXT
- Fix 16972: SharePoint Source throws an "Unable to cast object" error when more than one filter is applied
- Fix 16988: Data Validation Transform throws Object Reference Error when opening the component
- Fix 17067: Hash Transform does not keep the algorithm when adding a new Output Column or clicking OK

Version 2018.1.4 [February 8, 2018]

New Features:

- New 11892: SharePoint Destination now auto maps columns
- New 13965: REST Destination can now send a file as part of the POST Body

Updated Features:

- Update 13975: Help documentation now includes detailed list of Task Factory related help videos

Fixes:

- Fix 12504: Sharepoint Source Returning NULL for Column
- Fix 12517: SharePoint Source doesn't update column Description metadata when connection manager is updated
- Fix 12569: Replace Unwanted Characters 'Remove unprintable characters' Action strips valid characters from text
- Fix 13956: TF Advanced Lookup Transform will not accept a Boolean Data Type
- Fix 13971: Advance Cleansing Task Transforming Names Incorrectly
- Fix 13980: File Gateway Source - not reporting end of rowset
- Fix 13984: XML Output Destination creates empty elements
- Fix 13985: Upsert Destination cannot compare timestamps without updating destination column
- Fix 14003: Excel Source Serialization Error when Sheet Name is not found
- Fix 14022: Excel Source Component - Does not allow you to return one row.
- Fix 14030: Sharepoint Source pulling wrong column name resulting in NULL values
- Fix 14455: Add reserved-word Date header to rest connections
- Fix 14573: PGP Task displays password in plain text within the properties window
- Fix 14574: Fact Table Destination outputs NULL values when join conditions use Date data types
- Fix 14581: REST Oauth tokens can expire during long running operations
- Fix 14586: Upsert Task errors when attempting to update / insert to a table set to T-SQL Always

Encrypted
- Fix 14592: Excel Destination may corrupt the file when writing to pre-existing documents
- Fix 14593: Pack Data Transform does not remove mapping lines within UI when using 'Remove Mapping' option
- Fix 14597: Pack Data 'Generate Template' does not properly generate Parent to child relationships in .Liquid For Loop
- Fix 14609: Pack Data Transform passes NULL values when packing sub-level inputs
- Fix 14646: SFTP Task fails when running two SFTP tasks in parallel, and "Reuse Connection" is enabled within the connection manager
- Fix 14649: When special characters (#00,<) are included, a SharePoint Source throws "an error occurred while attempting to parse"
- Fix 14653: CRM Destination throws Object Reference error when writing to entity referenced upstream in data flow
- Fix 14656: Excel Source defaults back to original Column Names after every change within the component
- Fix 15126: Excel Source automatically converts date time columns set to 24 hour format to 12 hour format
- Fix 15127: Move or Delete Email will not fail when targeting a non-existent folder
- Fix 15128: Fact Table Destination throws error when using a Case Sensitive server and a Case Sensitive database
- Fix 15164: XML Output Destination does not always include specific attributes in the XML output
- Fix 15608: Newer TF components do not display the version number in the title bar
- Fix 15630: REST Source uses TLS 1.2 to "Preview Data" but uses TLS 1.0 when executing the package.
- Fix 15632: Address Verification Transform throws Exception from HRESULT: 0xC0010009 error when attempting to replace a column
- Fix 15676: Dimension Merge SCD crashes Visual Studio when current row value is blank on the SCD2 Date Handling page

2017

Version 2017.4.5 [November 6, 2017]

New Features:

- New 7365: REST Source can handle paging between web calls when page number and total pages are passed in
- New 10270: Oracle Upsert Destination now includes Refresh Metadata button
- New 10377: Pack Data transform now warns user when a column exists within the template that does not exist in the input
- New 11953: Salesforce Connection Manager now includes option to attempt retries on failures to connect or execute

- New 12006: Upsert Destination's error output now provides higher level of detail
- New 13042: Salesforce Destination now includes ability to set Batch or Bulk Size
- New 13044: XML Destination now includes option to not write an empty documentation when no rows are provided
- New 13059: XML Generator transform now includes UTF- NO BOM Encoding option
- New 13536: Task Factory is now compatible with SSDT for Visual Studio 2017

Updated Features:

- Update 9143: Dimension Merge Slowly Changing Dimension transform now includes option to ignore changes to SCD0 columns
- Update 13519: REST Source Header now supports the same variable syntax as other fields in REST Source
    - The REST Source Header section now accepts Namespace::VariableName as the syntax for replacement
- Update 11979: Online help documentation now includes a rolling list of release notes that include all product release notes within a single file

Fixes:

- Fix 9814: Column names insert at end of expression when adding from main page of Advanced Derived Column
- Fix 11024: Data Cleansing transform throws "Datatype mismatch between [Date] and output date format parameter [MMddyyyy]" error when converting from DT_DBTIMESTAMP to MMddyyyy
- Fix 12018: Expression engine does not correctly format the Day_Of_Week, GetDiskFreeSpace, GetDiskSize, Log10, SetDatePart and Sleep functions
- Fix 12039: The Test Expression button is disabled when using Expression task
- Fix 12477: Packages containing Advanced Lookup Cache connection managers fail to deploy to the SSIS Catalog
- Fix 12481: Social Media connection manager fails to connect to Facebook with "Could not find access_token" error
- Fix 12512: Replace Unwanted Characters is not using Unicode values correctly
- Fix 12530: REST Source hides any errors that are thrown if no error output is attached
- Fix 12570: File Gateway Source causes devenv.exe to hang and build memory pressure when selecting attached file
- Fix 12571: File Gateway Source does not reinitialize metadata properly when changing files
- Fix 12573: Dynamic CRM Destination can throw "The request channel timed out while waiting for a reply" when executing
- Fix 13058: XML Generator Transform drops lowest level of elements when closed and re-opened multiple times
- Fix 13097: SalesForce Destination does not use proxy settings during post execute phase

- Fix 13434: DT_DBTIMESTAMP2 variables not included as options for Auditing Values
- Fix 13488: Pack Data Transform does not retrieve existing values correctly and replaces them with NULL values
- Fix 13489: During execution, File Gateway source throws "File Validation Failed When Checking File Properties" error when validating a file with size between 1 and 1023 bytes
- Fix 13506: REST Destination not associating lineage ID's correctly when square brackets are present in POST Body
- Fix 13507: Upsert Destination's Refresh Metadata button does not work when on the update method tab
- Fix 13509: REST Source OAuth Connection Manager fails to authenticate if no token is specified
- Fix 13512: XML Output Destination's metadata does not re-initializing after columns are removed
- Fix 13524: Oracle Upsert doesn't find unique indexes
- Fix 13535: Excel Source throws "IsolatedStorageException" error when executed from a SQL Server Agent Job
- Fix 13929: Parameters defined within a REST OAuth2 configuration file are not replacing all fields within the REST OAuth2 connection manager
- Fix 13951: Expressions Editor is not accessible from within the SFTP task
- Fix 13991: Advanced Aggregate transform populates valid data with NULLs

Version 2017.3.3 [August 7, 2017]

New Features:

- New 8765: Move the preview for dynamics source to a pop up
- New 10293: Rest Source and Destination can now pass cookies during authentication
- New 10342: Dimension Merge SCD now includes sorting choices for GUIDs
- New 10390: Included the ability to preview Owner object columns in Salesforce Source UI
- New 10914: Advanced Execute Package task can now use project level parameters when in Project Deployment Model
- New 11025: Dynamics CRM Source can now extract the data from a view on an entity
- New 11039: Rest Source and Destination now support OAUth1 and Basic rest connections
- New 11098: XML Generation Transform can now automap columns
- New 11157: Within SFTP task, changing 'Stop package on failure?' now sets the 'FailPackageOnFailure' property to the same value
- New 11184: Added USPS Address Verification Component
- New 11643: Allow accessing Memory Optimized tables in Upsert
- New 11644: Check against zombie transactions
- New 11650: Rest Source Component Headers Variable/Parameter Syntax
- New 11974: REST Connection now allows selecting of REST config file
- New 11975: REST Config files now include header section

- New 11987: SMTP Connection Manager can now reuse same connection across the SSIS package execution
- New 12004: Marketo Destination now includes the ability to add custom parameters
- New 12401: Task Factory SSIS 2017 support
- New 2846: Licensing now works within environments that use FIPS compliant algorithms for encryption

Updated Features:

- Update 11964: Pack - Unpack Columns transform now supports removing byte order marks in xml unpack
- Update 11976: REST Destination "Test API" button now includes warning when clicked
    - Including a warning helps prevent unintentional REST calls

Fixes:

- Fix 1605: BI xPress Auditing Framework within Workbench throws "System.Exception" errors when being applied to packages containing Task Factory Components
- Fix 4407: Dimension Merge SCD transform throws "Array dimensions exceeded supported range error" when initial load of 200+ million rows
- Fix 10278: Dimension Merge SCD Incorrectly Sending Data Down New Path When SCD2 date is invalid
- Fix 10812: When in Project Deployment mode, child packages executed by the Advanced Execute Package Task do not use any of their referenced project parameters during execution
- Fix 11033: XML Generator throws an "item with same key" error when inside a for-each loop
- Fix 11657: Fact Table Destination Lookups are re-written when opened in 2017.1 versions
- Fix 11659: Email Source FilterDelimiter property does not work
- Fix 11664: Upsert Destination throwing Object Reference Error for SQL_Variant data type
- Fix 11667: Excel Source sometimes throws "Object Reference" error when executing
- Fix 11672: Advanced Derived Column shows invalid expressions as valid
- Fix 11890: Excel Source is not able to read formatted rows with a common formatid
- Fix 11891: Fact Table Destination fails to loads when objects are missing
- Fix 11960: Pack - Unpack Columns transform throws 'Column must have a code page greater than zero' error when new columns are added
- Fix 11961: Dynamics CRM Source takes a long time to respond when connecting to objects with high number of activity parties
- Fix 12014: Excel Source does not parse worksheet properly where row indexes are skipped
- Fix 12036: Salesforce Source incorrectly maps columns in Error Output after truncation occurs
- Fix 12476: Unpack Columns transform does not parse columns with DT_NTEXT or DT_TEXT data types
- Fix 12403: Denied Access to registry key "HKEY_CURRENT_USER"

Version 2017.2.3 [May 1, 2017]

New Features:

- New 1898: Rest Destination
- Rest Destination can be used to send data to a Rest API and read the responses back into an output.
- New 10245: Task Factory UI's compatible with custom scaling in Windows Display Settings
    - Create ability for all Task Factory task/component UI's to work with scaling options in Windows 10 other than 100%. When set to 125% or higher, UI do not adjust, and configurations can not be seen in certain screens. Image of Expression Editor attached to display this behavior. Issue is also known to affect the column mappings tab in the Dim Merge SCD component as well.
- New 10288: Allow renaming of files in Remote Directory path when uploading to SFTP
- New 10389: Added compatibility for string response based on RegEx to REST Source
    - Add functionality for Rest Source to handle alternate response from REST APIs, such as string
- New 10790: E-mail connection manager now minimizes the number of concurrent connections needed during execution

Updated Features:

- Update 8692: REST Source & Destination now includes an auto retry connection functionality
- Update 9168: Rest Source now includes an error response handler
- Update 9175: SalesForce Destination now includes ability to deselect an Assignment Rule
- Update 9812: Help documentation for Advance Execute Process now better explains the difference between "In Process" and "Out of Process"
- Update 9825: Rest Connections now uses .NET WebRequest if .NET 4.5 is installed on the machine
- Update 9829: SalesForce Source now allows the user to refresh metadata when a subquery is present within a where statement
- Update 9831: Hash Transform now defaults to '|' as the column separator
- Update 9837: OAuth2 Rest Connection now defaults to Post
- Update 9853: Help documentation for SalesForce Destination now better describes its Bulk processing option
- Update 10359: Azure ML Batch Task help page to reflect the necessary inclusion of /jobs appended at the end of the post request
- Update 10364: Help documentation for SharePoint Destination now includes details on Download Attachments
- Update 10365: Help documentation for SalesForce Source no includes information on variable usage
- Update 10958: SharePoint Destination now includes additional date data type compatibility
    - DT_DBDATE
    - DT_DATE
    - DT_DBTIMESTAMP2

Fixes:

- Fix 4074: "Replace Null Text Values" option is re-selected when address parse is closed
- Fix 7874: Fixed SFTP connection where TLS 1.2 is not enabled on SSH servers

- Fix 9140: In Upsert mode, SalesForce Destination loses External ID column when refreshing columns
- Fix 9141: UPSERT with AZURE orphans temp tables after execution completes
- Fix 9152: Adding auditing framework removes the mappings from the XML Output Destination
- Fix 9158: Salesforce Destination does not update values in SF Object when passing an empty string
- Fix 9198: Add option for bankers rounding within the expression engine
- Fix 9204: Pack Data Transform does not recognize NULL values for IF statements in Liquid
- Fix 9214: XML Output Destination fails to show validation errors when opening invalid XSD
- Fix 9215: Excel files throw corrupted data error when Excel Destination creates new worksheet
- Fix 9216: Rest Source throws 'Failed to verify exchange signature' error
- Fix 9250: Excel Source does not capture data from Named Ranges not beginning on Row 1 of worksheet
- Fix 9687: The data type of output columns on the TF Dynamics CRM Destination cannot be changed, except for DT_WSTR and DT_NTEXT columns.
- Fix 9764: Salesforce Connection Manager does not allow for dynamic variable usage
- Fix 9810: Unpack Data Transform throws a "could not cast object type 'System.Xml.Linq.XElement' to type 'System.XML.Linq.XAttribute'" error when using attribute based queries
- Fix 9816: Marketo Destination throws an "Object Reference" error when attempting to access the destination object
- Fix 9823: Null Handler Transform populates the column list when setting "NULL or Blank Values" to a user defined value with column parameter
- Fix 9834: TF Upsert Destination Does Not Fail On Table Constraint Violation
- Fix 9836: Excel Source throw an "Object Reference" error and inserts an empty string when the native query returns 'null'
- Fix 9840: Added option for Guid sorting to fix XML Output Destination skipping elements in multiple entity datasets when guids are used for keys
- Fix 9844: XML Output Destination does not follow XSD and write empty elements when there is an empty string or null value in the data set
- Fix 9855: Excel Source does not refresh meta when switching from Raw to Formatted output
- Fix 9862: Secure FTP now disables Local File and Directory Settings when using the 'Check if file exists on server' option
- Fix 10202: Fix reinitialize metadata for pack data transform
- Fix 10283: Making column changes in the File Format tab refreshes the columns in the column tab
- Fix 10284: TF Basic Rest Connection does not use Headers with "Test Connection" on Endpoint tab
- Fix 10287: Dimension Merge Slowly Changing Dimension experiences performance degradation
- Fix 10310: Salesforce Destination does not carry Assignment Rules when applied in Bulk / Bulk Zip modes
- Fix 10320: Excel Destination Object disconnect or does not exist
- Fix 10321: Marketo Destination - Object Reference when populating Destination Object list when proxy is enabled but not connected
- Fix 10323: Marketo Destination - filterType not specified Error when selecting custom Marketo objects

- Fix 10324: TaskFactory FileCombinePath expression returns
- Fix 10344: Excel Source throws Unable to determine the identity of domain
- Fix 10354: Large record sets cause memory pressure
- Fix 10357: TF SharePoint Source throws System.ArgumentException: Item has already been added. Key in dictionary error
- Fix 10915: Data Validation Expression Tester Does Not Leverage Transform Options
- Fix 10920: AesEncrypt expression throws Unable to cast object of type 'System.Byte[]' to type 'System.IConvertible' with provided package
- Fix 10951: Cannot open TF upsert UI
- Fix 10959: Creating a new column in TF Advanced Derived Column throws Exception from HRESULT: 0xC0204019
- Fix 11159: EndsWith Operator in the Advanced Lookup Cache Transform does not work properly
- Fix 11160: Filter Operators for Advanced Lookup Transform are not functioning properly
- Fix 11161: Using IsContainedIn filter in Advanced Cache Lookup Connection Manager breaks the component

Version 2017.1.3 [February 2, 2017]

New Features:

- New 4702: sFTP Task can now remotely rename a file
- New 7797: Excel Power Refresh now includes an execution timeout option
    - ℹ **Note:** this will not stop an execution when a Connection Login dialog is prompted by Excel.
- New 7799: REST Source now supports including the HTTP status information as an output
    - REST Source now includes a new output that will be filled with a single row when anything other than a status of 200 is returned from the rest server
    - This output contains the status code (200, 400, 404, 500, etc), the status description and any data returned from the server
- New 7826: Added support for wildcards (*) in JSON queries
- New 7930: Task Factory help documentation now includes sample Rest Source configurations files for TFS / VSTS
- New 7960: REST Connection and Source now supports the use of configuration files
- New 8308: SharePoint Source now allows the use of custom Caml & Query options
- New 8614: OAuth2 token getter now includes proxy and headers

Updated Features:

- Update 4691: Dynamics CRM Source now includes change data capture (Get Changes) functionality
    - Added an option to the Dynamics CRM Source that allows the use of the Get Changes functionality added in the 2015 Update 1 release of Dynamics CRM.
- Update 8739: Improved product version logging at start up

Fixes:

- Fix 4727: Advanced Derived Column is unable to convert string values to GUIDs
- Fix 4728: Salesforce Source consumes more memory than previous versions on execution
- Fix 7865: Help links within some Task Factory components point to incorrect locations
- Fix 7880: XML Output Destination is not mapping unbounded recurring elements correctly
- Fix 7923: SharePoint Documents unable to upload files over 25 MB
- Fix 7926: Unpack Data is dropping Data Flow expressions after transform has been edited
- Fix 7928: In Visual Studio 2015, Expression Engine throws an object reference error when target SQL Server Version 2014 or below
- Fix 7931: Dimension Merge SCD is dropping records when memory optimized mode is enabled
- Fix 8243: 'Content not allowed in prolog' error thrown upon SalesForce Source execution
- Fix 8322: 'Create new table' option not available when using a database with no tables in Upsert Destination
- Fix 8332: Parsing error thrown with dt_dbtime and dt_dbtime2 datatypes in SSIS 2008
- Fix 8689: SalesForce Destination now uses proxy settings when using bulk process mode
- Fix 8695: Unpack and Rest Source does not return child elements for an XML based column
- Fix 8709: XML Destination writes elements out of order when writing a full element before a grouping occurs
- Fix 8763: Upsert Destination does not fail when a DB constraint is violated and fail component option is set
- Fix 8772: Email Source does not filter correctly when a filter is placed on the 'MessageID'
- Fix 8777: Auditing columns within the Dimension Merge do not allow string variables or string values to be written
- Fix 8778: SalesForce Destination does not send errors down Error Output when in Bulk or BulkZip Mode
- Fix 9146: Unpack Data Transform throws Object Reference when editing File Format properties
- Fix 9147: Unpack Data Transform throws Object Reference when JSON Properties tab is not configured
- Fix 9142: The license expiration date does not appear in the License Manager

2016

Version 2016.4.5 [October 25, 2016]

New Features:

- New 4012: File Properties Task compressed UI description should indicate NTFS compression
    - Updated the File Properties Task description for compressed files to clarify the use of NTFS compression.
- New 6627: Added before and after scripts to Fact Destination, Upsert Destination and Oracle Destination
    - Added Before and After Scripts to Upsert Destination. Oracle Upsert Destination and Fact Table

Destination. The before script is used to execute a sql statement before each buffer is processed. The after script is used to execute a sql statement after each buffer has been sent to the destination.

- New 6654: Created FileGateway Source
  - Create a filegateway source based on the filegateway task that will separate the valid and invalid rows into success and error output
- New 6685: Excel Source
  - Excel Source uses openxml to read an excel worksheet, sending the rows into a data flow
- New 6688: Excel Connection Manager
- New excel connection manager for use with the Task Factory Excel Source and Destination
- New 6691: Excel Destination
  - Excel Destination supports writing data flows into excel files using the openxml format
- New 6693: Refactor Task Factory help doc control flow items
  - Refactor the Task Factory help documentation control flow items to ensure they are up to date and easier readability.
- New 6759: Added Support for TLS 1.1 and 1.2 To Rest Connections
- New 6794: Task Factory - Rest Source - Addition of Namspacing
  - Added the ability to define namespaces for xml queries in both the unpack data transform and the rest source
- New 7344: OAuth 1 Connection Manager to REST Source
  - This can be used to connect to endpoints where OAUTH1 is needed to make rest calls.
- New 7381: Added "Pause Between Round Trips" option to Rest Source, Marketo Source and Marketo Destination
  - A client is experiencing the following issue. After discussing the inquiry, development requested a PBI for the inquiry below: • The Marketo API limits 100 calls in 20 seconds. Each call can return 300 records at a time. • We have around 92000 records to be extracted however the process fails after 30000 records. Is there a way to limit the number of records being read by TF Marketo source? It can help us overcome this problem.
  - Find out if configurable on Marketo
- New 7382: Include 'Rebuild temp table indexes' option in Oracle Upsert Destination
  - This new option gives the ability to rebuild the temp table index after each buffer.
  - This resolves an issue where some Oracle environments may return an 'ORA-26026' error when using the Oracle Upsert Destination.
- New 7388: Added Style / Formatting To Excel Destination Columns
  - Added support for adding style to columns. The styling is based on microsoft excel formatting and all formatting can be customized based on Excel formats.

Updated Features:

- Update 7341: Error Output Description now uses properties and inputs

Fixes:

- Fix 4988: Uninstalling Task Factory with Workbench installed removes Workbench dependencies from the GAC
- Fix 5501: Int64 Variable Values not available for auditing in the SCD Transform
- Fix 5520: Advanced Derived Column throws "Illegal Characters in Path" error when using the FIleWriteAllByte function
- Fix 6063: SalesForce connection manager throws "Proxy 407 Authentication Error" when using a proxy that doesn't use basic authentication
- Fix 6064: Hash transform updates order when selecting and removing input column(s)
- Fix 6184: TF Sharepoint Destination Column Mismatch between versions.
- Fix 6202: TF Upsert Destination - Unable to contain spaces in table name
- Fix 6203: TF Fact Table Destination unable to use table with Spacing in the table name
- Fix 6216: Using "Deploy Package" option in 2015 data tools throws VS_IsCorrupt error with Advanced Aggregator
- Fix 6220: Excel Refresh Task - task completes successfully even if the data connections are not refreshed successfully.
- Fix 6255: TF SalesForce Source cannot run in Parallel
- Fix 6785: SalesForce Source throws "XML Document Error" for Contact and Account objects
- Fix 6830: Fact Table Destination fails to create temporary table when the same column exists in the source, lookup and destination table
- Fix 6831: Fixed Error in XML Transform Where Child Elements Weren't Being Written When Mapped
- Fix 7342: The Rest Source OAuth 2 Token Getter throws 'Browser No Longer Supported' when getting tokens
- Fix 7355: MS SQL 2016 CU1 update throws error "Could not load file or assembly 'PragmaticWorks.Common&'
- Fix 7356: Within Fact Table Destination, multiple equality lookups on a dimension table changes to the same source column
- Fix 7362: Fixed Issue Where Salesforce Destination passes name of Lookup column twice when used as External ID
- Fix 7374: Email Source doesn't save variable selection when removing selected variable
- Fix 7375: Dimension Merge SCD does not open in VS 2008 when using TF Version 2016.3.7
- Fix 7380: Salesforce Source fails to return data when multiple objects are referenced
- Fix 7395: Pack Data Transform hangs when no rows are passed to the component
- Fix 7406: Upsert Destination throws error when creating temporary tables based on table names which include blank spaces
- Fix 7806: SalesForce Destination throws "Unable to cast object of type 'System.Int64' to type 'System.Byte[]'.". error when using bulk mode
- Fix 7836: SalesForce Destination throws 'Object not set to an instance of an object' error when using Bulk or Bulk Zip mode in combination with the 'Wait for Bulk Results' option

- Fix 6666: Licensing is now more robust and should not issue an error during installation.
- Fix 8689: SalesForce Destination fails when using a proxy in Bulk or BulkZip mode.

Version 2016.3.7 [August 1, 2016]

New Features:

- New 4492: Add support for three part names in upsert / fact table
  - Support for three part names (server).(db).(tablename) in upsert and fact table has been added.
- New 4692: Sharepoint document task now allows check out and check in of a file
  - The SharePoint task now supports the ability to check a file out prior to updating it, and check it in when the update is finished.
- New 4693: Created File Gateway Task
  - The File Gateway task supports validating text files against a set of rules. This enables users to confirm that a file is valid for processing prior to attempting to load it. If the file passes all rules, then the task will succeed, otherwise it fails.
  - Supported rules include:
    - Number of rows
    - File size
    - Number of columns
    - Rows contain the specified number of columns
- New 4705: Upsert Destination now supports Error Row Redirection
  - The Upsert Destination now supports error row redirection. As a buffer of records comes into the component, it is sent as a batch to the database. Any database failure will roll back the batch, and the buffer will be retried a single row at a time. Any individual row failure will be redirected to the error output. Subsequent buffers will be processed as batches, unless there is a failure.
  - This behavior can be enabled using the "How to handle errors" property. Setting it to redirect errors enables the new behavior.
- New 5480: Added Execution Timeout to Rest Source
  - Added Execution Timeout to the rest source to allow the end user to provide a longer running execution time.
- New 5491: Sync 2016 Extensions and Mappings in TF Build
  - The mappings files have been updated to reflect the latest versions and compatibility.
- New 6065: Task Factory Advanced Execute Package Task - Outside Project execution
  - Updated the advanced execute to allow the selection of any environment from the catalog projects.

Fixes:

- Fix 1324: Allow Oracle Upsert to gracefully handle changes in table/views
- Fix 1364: Salesforce Destination outputs incorrect object SalesforceId when bulk processing mode is selected

- Fix 1373: Salesforce Destination fails entire batch when Quote object references other objects with a deleted or non-existent object ID
- Fix 1564: Terminator dest does not retain column selection for output
- Fix 1694: "System.AccessViolationException" Error when loading all TF Connection Managers in 64 bit
- Fix 1944: SF Source queries that returns no rows throws error when closing the component
- Fix 4463: Salesforce proxy reverts to using non-proxy connection if proxy host and credentials are incorrect
- Fix 4482: TypeHandlerFactory fails to load if Microsoft.SqlServer.Types 10.0.0.0 isn't present on the system
- Fix 4667: Adv email and SMS task is unable to use 'user friendly' names appended to email addresses
- Fix 4719: Changing the upstream source component name causes fact table lookups to become invalid
- Fix 4843: Cannot change action in previously configured Sharepoint Source
- Fix 4887: TF Upsert Destination - Cannot upgrade to newest release when upgrading from 2.3.0
- Fix 5450: TF Advanced Derived Column Transform - FileWriteAllBytes
- Fix 5457: Pack Transform displays an error when Generate Template is clicked
- Fix 5499: Sorted GUID business keys throws "Items must be added to the queue in sorted, ascending order" error in memory opt mode
- Fix 5517: Upsert Row By Row doesn't fail a second row in insert only mode
- Fix 6045: Fixed TF SalesForce Source SOQL Relational Query Issue Where Columns Named 'Type' Returned The Wrong Data
- Fix 6263: Lastest TF build is missing assembly Microsoft.SqlServer.DtsMsg for 2014

Version 2016.2.4 [May 4, 2016]

> ⚠ **Important:** Per Salesforce's help documentation, "Starting in June 2016, Salesforce will begin disabling the TLS 1.0 encryption protocol using a phased approach."Customers should note that when this change goes into effect, all versions of Task Factory prior to version 2016.1.7.210 will no longer be able to connect to Salesforce.

New Features:

- New SQL Server 2016
  - Task Factory compatibility with SQL Server 2016. (Currently, Release Candidate 3.)
- New Dynamics Source & Destination
  - (TF-TFS-4292):Dynamics Source and Destination components can now connect to Dynamics CRM 2016 premise and online servers.

Updated Features:

- Update All Components
  - (TF-TFS-4433): All components that use column selection now support select all/ select none

toggle.
    - (TF-TFS-4089): Updated components that threw error when CodeSlice was installed with Task Factory.
    - (TF-TFS-4406): Updated installer to uninstall all third party assemblies.
- Update REST Source
    - (TF-TFS-3334): Improvements to component including: removed password requirement from rest connection, add URL Based paging, allow a choice of POST or GET actions, update code for handling tokens, and created the token getter as a standalone app.
- Update Facebook Source
    - (TF-TFS-4117): Connection manager now allows users to set permissions.

Fixes:

- Fix Address Parse Transform
    - (TF-TFS-4076): Fixed error where AddressLine2 Nulls caused component to fail with object reference error.
- Fix Advanced Lookup Transform
    - (TF-TFS-4296): Fixed issue where input parameters that contain null throws 'Input string was not in a correct format' error.
- Fix Email Source
    - (TF-TFS-3074): Fixed issue where connection credentials set dynamically in a for each loop reused the initial variable values.
- Fix Facebook Source
    - (TF-TFS-4117): Fixed issue where the connection manager couldn't set permissions.
- Fix PGP Encryption Task
    - (TF-TFS-4135): Fixed issue where component did not embed the name of the file in the metadata.
- Fix Secure FTP Task
    - (TF-TFS-4405): Fixed issue where using ftp connection type throws object reference error.
    - (TF-TFS-4175): Fixed issue where SFTP fails to connect and throws 'Setting up fallback kbi prompt based on supplied password' using edtFtpNetPro.
    - (TF-TFS-4176): Fixed issue where certain key files failed authentication even when able to connect using filezilla .
- Fix Salesforce Source
    - (TF-TFS-4133): Fixed issue where relational columns created in altered soql query produced NULL in the output.
- Fix Sharepoint Source
    - (TF-TFS-3847): Fixed issue where calculated columns in Sharepoint still displayed ID's with column name even if remove option is selected
- Fix Upsert Destination
    - (TF-TFS-3876): Fixed issue where using column compare method updated columns with '0' values

to null

Version 2016.1.7.210 [February 10, 2016]

New Features:

- New Azure ML Storage Destination
  - A new data flow component used to send data to an azure blob storage file that will be consumed by an Azure ML Scoring / Predication Web Service.
- New Azure ML Storage Source
  - A new data flow component used to retrieve the results of a scoring / prediction Azure ML Web Service execution.
- New Azure ML Batch Execution Task
  - A new control flow component that control the batch execution of an Azure ML Web Service. Can be used with the Azure ML Storage Destination and Azure ML Storage Source to insert data before a batch execution and then retrieve the results of a batch execution.

Updated Features:

- Update Dimension Merge
  - (TF-TFS-1206): Updated Dimension Merge SCD's auditing to default to an integer and filter variables by datatypes.
  - (TF-TFS-2416): Updated Dimension Merge SCD to allow users the ability to select compound surrogate keys.
- Update Salesforce Connection Manager
  - (TF-TFS-3510): Connection managers are now updated to use TLS 1.2. *Important note: Per Salesforce's help documentation, "Starting in June 2016, Salesforce will begin disabling the TLS 1.0 encryption protocol using a phased approach."
- Update Rest Source Connection Manager
  - (TF-TFS-3334): Rest Source's standard connection manager now allows users to connect using base64 encryption headers.
  - (TF-TFS-3337): Rest Source's connection manager now allows users to add headers to authentication requests.
- Update SFTP Task
  - (TF-TFS-4002): Reintroduced edtFtpNetPro into the connection manager. This can be seen by choosing the SFTP-SSH FTP Legacy connection type.
- Update Unpack Data Transform
  - (TF-TFS-2898): Added the option to select / unselect all columns to be included in the Unpacked output

Fixes:

- Fix Advanced Aggregator

    - (TF-TFS-4134): Fixed error where component produced incorrect results when using MIN and MAX with text values.
- Fix Advanced Execute Task
    - (TF-TFS-3650): Fixed error when executing child packages using Excel source and x86 Microsoft Access Database Engine 2010 Redistributable.
- Fix Advanced Cache Transform
    - (TF-TFS-4050): Fixed error where the component throws "item with same key" error when it was used in for each loop.
- Fix Email Source
    - (TF-TFS-4033): Fixed issue where clicking the test connection button for IMAP closed and crashed the connection manager when port and folder aren't selected
- Fix Excel Power Refresh
    - (TF-TFS-4136): Fixed issue where refresh throws error when Excel used an SSAS cube connection.
- Fix Expression Based Components
    - (TF-TFS-3359): Fixed issue where ExecuteSql method caused 'ExecuteQueryWithParamsDB' is not a member of 'PW.TaskFactory.SharedMethods' error.
    - (TF-TFS-3765): Fixed issue where expression task fails to convert to type double when using an INSERT INTO expression.
- Fix Facebook Source
    - (TF-TFS-4117): Fixed issue where the connection manager couldn't set permissions.
- Fix Dimension Merge SCD
    - (TF-TFS-3432): Fixed bug in Dimension Merge SCD where new row variable counts produced in auditing did not properly set in the post execute event when using memory optimized mode.
    - (TF-TFS-3669): Fixed bug when using compound business keys with dt_timestamp threw an error in 2012 & 14, but executed in 2008.
    - (TF-TFS-3763): Fixed issue where using GUID as business keys sent rows down the invalid input.
    - (TF-TFS-3816): Fixed issue where adding new columns to SCD caused outputs configured to be excluded to become included.
- Fix Secure FTP Task
    - (TF-TFS-3077): Fixed issue where renaming the connection manager caused the component to lose the connection manager selected.
    - (TF-TFS-4121): Fixed issue where package did not fail when trying to download an invalid directory from server using chilkat setting.
    - (TF-TFS-4153): Fixed issue where reusing the connection appended the remote file path.
    - (TF-TFS-4211): Fixed issue in legacy setting where host addresses that ended with "/" threw a "no such host is known" error on connection test.
- Fix Salesforce Source
    - (TF-TFS-2828): Fixed issue where some users were experiencing slower performance with large datasets when compared to pre 4.0 versions.

    - (TF-TFS-2831): Fixed issue where relational columns that contained nulls caused the column to be removed from the metadata.
    - (TF-TFS-3817): Fixed issue where custom SOQL queries using < and a trailing > eliminates the < and > in the previewed query statement.
    - (TF-TFS-4247): Fixed issue where 3 sources in a single data flow that returned more than 1000 rows threw an error.
- Fix Sharepoint Source
    - (TF-TFS-1339): Fixed custom columns set to not be displayed in custom views are being displayed in Sharepoint Source external metadata column
    - (TF-TFS-3476): Fixed issue when adding and removing columns from the Sharepoint source caused "key in dictionary" error.
    - (TF-TFS-3764): Fixed issue where component did not display previews of custom columns that used (Sharepoint) single line of text datatype.
- Fix Unpack Data Transform
    - (TF-TFS-3093): Fixed issue with unpack data tranform in Xml mode where null values were returned with a root query defined.
    - (TF-TFS-3103): Fixed issue with Unpack Data Transform where DT_NTEXT columns were not being included in the unpacked output.
- Fix Upsert Destination
    - (TF-TFS-4019): Fixed issue with the "Rebuild Definition From Mappings" hovered over entire user interface when adjusting window size.
- Fix XML Output Destination
    - (TF-TFS-3075): Fixed issue where user interface didn't display certain elements referenced in a child xsd.
    - (TF-TFS-4004): Fixed the output to include xml declaration when using xsd's that provide the declaration.

2015

Version 2015.3.7.1105 [November 5, 2015]

New Features:

- New Marketo Source
    - A new data flow component used to connect to and retrieve objects such as leads and opportunities for users that employ the Marketo automated marketing software.
- New Marketo Destination
    - A new data flow component that connects to the Marketo automated marketing software and allows users to insert, upsert, update, delete, and duplicate data such as Leads and Opportunities.

Updated Features:

- Update Data Validation Transform
  - (TF-TFS-2415): Custom regular expressions can now be viewed in the user interface when package is viewed by a different user.
- Update Rest Source Connection Manager
  - (TF-TFS-2424): The OAuth2 connection manager is now more configurable to accommodate connections to various OAuth uses. Users can select preconfigured templates including (but not limited to) Google, Paypal, and GoToMeeting.
  - (TF-TFS-2424): The standard connection manager options were expanded to include an (optional) Authentication Endpoints tab.
- Update Salesforce Destination
  - (TF-TFS-1701): Users can now select between Parallel and Serial Concurrency when performing an action.
- Update Update Batch & Delete Batch Transform
  - (TF-TFS-2004): Update and Delete Batch transform user interfaces now match Upsert Destination General tab (column mappings) and includes a Refresh Metadata button.
- Update XML Output Destination
  - (TF-TFS-1778): Component now has the option to create and edit namespaces for elements.

Fixes:

- Fix Advanced Derived Column
  - (TF-TFS-1372): Fixed ToDate function that would error when format option was included in the expression.
  - (TF-TFS-1378): Fixed Millisecond function text formatting within the expression editor.
- Fix Advanced Lookup
  - (TF-TFS-2580): Fixed issue with Norway (Bokmål) region setting sending matched output to the non-matched output.
- Fix Dimension Merge SCD
  - (TF-TFS-1716): Fixed execution 'unable to cast object' error when processing columns with DT_IMAGE (blobcolumn) datatypes.
- Fix Dynamics CRM Source
  - (TF-TFS-1544): Fixed the Account entity that incorrectly displayed and returned the OwnerIdType column as a name rather than a numeric value.
- Fix Dynamics CRM Destination
  - (TF-TFS-2438): Fixed issue with inconsistent results being sent when both success and error outputs were used.
- Fix Email Source
  - (TF-TFS-2079): Corrected user interface filters where the OK and Cancel buttons were reversed.
- Fix Fact Destination
  - (TF-TFS-1583): Fixed error encountered when a lookup used 2 conditions that pointed to the same

source column.

- Fix Rest Connection Manager
  - Fixed an issue with the Basic Authentication where the Network Credentials were not being set correctly.
- Fix SFTP Task
  - (TF-TFS-2231): Fixed issue where component throws Chilkat error when using variables for path and filename to download a file.
  - (TF-TFS-2888): Fixed issue where using the download directory action did not download subdirectory files when the option was selected.
- Fix Salesforce Destination
  - (TF-TFS-1371): Fixed issue where the assignment rules drop-down was disabled for Accounts object even when rules referenced an account.
- Fix Salesforce Source
  - (TF-TFS-1566): Fixed issue where the component did not output different parent/child objects created in a custom query.
- Fix Upsert Destination
  - (TF-TFS-2567): Fixed issue with component not updating from pre-3.0 versions.
- Fix XML Output Destination
  - (TF-TFS-2232): Fixed issue with integer variables mapped to an attribute caused the component to discard the xsd and its mappings, leaving only the attribute using the variable.
  - (TF-TFS-2445): Fixed issue where multiple inputs caused the package to fail after being executed multiple times.

V4

Version 4.3.0.709 [September 3, 2015]

> ⚠ **Important:** *As of the 4.3.0.709 release of Task Factory, the Integration Services 2005 binaries are no longer included within the installer. If you are running SSIS 2005, please use a build prior to 4.3.0.709 to install the 2005 components.

New Features:

- New Rest Source
  - This new component allows users to connect to a web service that utilizes a REST API and extract data in XML or Json format.

Updated Features:

- Update Advanced Cache Lookup Connection Manager
  - (TF-TFS-10951): Updated the user interface for 2012 and 2014 versions and its configurable filter.

- Update Connection Managers
  - (TF-TFS-10531): This update addresses a Microsoft fix where converting third-party connection managers (ie Dynamics, Hadoop, Salesforce, and Sharepoint) to project deployment while using EncryptSensitiveWithPassword encryption threw the following error: "'Salt' attribute for password is missing in the project manifest - SSIS SMO Connection Manager."
- Update Upsert Destination
  - (TF-TFS-11274): Component now includes a Refresh Metadata button and no longer displays error when a column is added or deleted. This allows the package to execute even if upstream metadata adds or deletes a column before reaching the Upsert Destination. (Previous behavior – the package would fail execution.)
- Update Sharepoint Source
  - (TF-TFS-10537): Lists with attachments now have ability download the attachment

Fixes:

- Fix Advanced Conditional Split
  - (TF-TFS-11182): Fixed issue where the default output name didn't save after closing the component.
- Fix Advanced Derived Column
  - (TF-TFS-10897): Corrected issue where user interface didn't display the functions list correctly.
- Fix Advanced Execute Package
  - (TF-TFS-11337): Fixed issue where upgrades that skipped over multiple versions changed the MappingDirection within the package XML code.
- Fix Advanced Lookup Cache Connection Manager
  - (TF-TFS-8324): Fixed issue where input parameters did not retain multiple conditions.
- Fix Data Flow Nugget
  - (TF-TFS-9927): Fixed issue where execution package threw "input lineage ID not found" error when triggering a nugget package that used the native Sort component.
- Fix Dimension Merge SCD
  - (TF-TFS-2197): Fixed issue where component threw error when using multiple sorted business keys in memory optimized mode.
- Fix Fact Destination
  - (TF-TFS-11183): Fixed issue with conditions using hard coded integer values appended the letter "b" to the end.
- Fix Oracle Upsert Destination
  - (TF-TFS-11149): Fixed issue where the component could not create temp tables when accessing a table created by a different user.
- Fix Salesforce Source
  - (TF-TFS-11150): Fixed issue where setting the batch size in normal mode only inserted that number of rows.

    - (TF-TFS-10871): Fixed issue with proxies used in the connection manager throwing an "unable to connect to remote server" error.
- Fix Salesforce Destination
    - (TF-TFS-10822): Fixed issue where certain credentials were throwing a "Given key not present in the dictionary" error when using the Leads object.
- Fix Timezone Conversion
    - (TF-TFS-10877): Fixed issue where setting the batch size in normal mode only inserted that number of rows.

Version 4.2.0.604 [June 4, 2015]

Patch:

- Patch Advanced Lookup Cache
    - (TF-TFS-11045):This patch fixes an issue where the advanced lookup cache manager failed to create indexes thus causing performance degradation.

Version 4.2.0.511 [May 11, 2015]

New Features:

- New Excel Power Refresh Task
    - A control flow component that can be used to refresh Excel Power Query and Power Pivot tables
- New Pack Data Transform
    - The counterpart to the Unpack Data Transform, developers can use a single input or multiple relational inputs to store multiple column and row values within a single destination column. Users can choose to store this data as either XML, JSON, or delimited format.
- New Facebook Source
    - Developers can use SSIS to connect to a Facebook account and extract information such as status updates, friends, comments, and inbox messages.
- New LinkedIn Source
    - This new component gives developers the ability to connect to a LinkedIn account and extract items such as a user's name, updates, connections, and headlines.
- New Twitter Source
    - Developers can use SSIS to connect to a twitter account and extract information such as status updates (known as 'Tweets',) followers, hashtags, retweeted, and 'favorited' messages.

Updated Features:

- Update Dimension Merge Slowly Changing Dimension
    - Memory Optimized Property: This new property was added to help relieve the memory pressure on packages that used extremely large datasets. It is highly recommend that users read the help documentation before incorporating it within packages. (This option is only available to users with

SQL 2012 and higher.)

- Update Dynamics CRM Source and Destination
  - Components are now compatible with Dynamics CRM 2015 premise and online.
  - (TF-TFS-10411) Users can now select N:N custom entities
- Update Fact Table Destination
  - (TF-TFS-9427): Component can perform lookups on views.
- Update SFTP Task
  - (TF-TFS-10322) Component now includes 'Get List of Files with Metadata' option.
- Update Sharepoint Source
  - (TF-TFS-10536) Lists with attachments now displays the attachment rather than a Boolean value.
- Update Unpack Data Transform
  - Added a Test Unpack Columns tab that allows users to test their configuration with real data
  - Component can now unpack columns with JSON data

Features:

- Fix All Components
  - (TF-TFS-7632): All help buttons now open the correct and current online help document.
  - (TF-TFS-9651): All video buttons have been removed from component footers. Videos can now be accessed by clicking the Help button.
- Fix Advanced Derived Column
  - (TF-TFS-2913): Setting a decimal scale to 0 no longer throws an object reference error.
  - (TF-TFS-3469): Syntax Window that displayed multiple parameters now adjusts properly.
  - (TF-TFS-10140): Corrected issue where DateDiff expression's StartDate and EndDate input values being reversed.
  - (TF-TFS-10366): Fixed issue with GetDatePart function formatted to return 2 digit year ("YY") returned 4 instead.
- Fix Advanced Execute Package
  - (TF-TFS-1747): Assignment Direction and Child Package Variable are no longer enabled until a user selects a package.
  - (TF-TFS-2129) The Advanced Execute Package Transform can now replace a second nugget found within the first Nugget.
  - (TF-TFS-3851) Proper error message is now displayed when attempting to connect to an SSIS Catalog but no SSISDB is present.
  - (TF-TFS-4001) Component now throws a more graceful error message when attempting to execute a non-existent package thought to be located on the SSIS Catalog.
  - (TF-TFS-7144) Expression button is no longer fixed in place when resizing the user interface.
- Fix Data Flow Nugget
  - (TF-TFS-4721): Password property no longer displays actual password but instead replaces characters with asterisks

    - (TF-TFS-9100): Component now includes validation to ensure the nugget data flow does not contain another nugget
- Fix Dimension Merge SCD
    - (TF-TFS-9437): Component no longer throws an error when processing DT_DBTIME2 as SCD1 or SCD2 type changes.
- Fix Dynamics CRM
    - (TF-TFS-10420): Fixed issue with Dynamics CRM connection manager did not retain proxy configurations.
- Fix Error Output Destination
    - (TF-TFS-4610): Fixed issue where upstream columns that were deleted after the EOD component was configured remained.
    - (TF-TFS-4612): Component no longer duplicates columns when constraint is removed and reapplied.
- Fix Fact Table Destination
    - (TF-TFS-8833): Source tables that use non-alphanumeric characters no longer throws an error.
    - (TF-TFS-10399): Deleting a lookup and closing the component no longer throws an object reference error.
- Fix Oracle Upsert
    - (TF-TFS-9528): Users can now insert only without needing a primary key.
    - (TF-TFS-9534): Corrected issue where temp tables remained on the Oracle server when execution failed.
- Fix Salesforce Source
    - (TF-TFS-9298): Component no longer throws an XML error when sandbox accounts use the version 30.0 API
    - (TF-TFS-10012): Fixed “escape” issue where '&' displayed as 'amp&'
    - (TF-TFS-10690): Corrected issue with Salesforce source UI not displaying edit query and variable parameter window for environments using 125% or higher display values.
    - (TF-TFS-10692): Fixed previews that contained custom fields, NULL, & actual values throwing a "Given key not present in dictionary" error.
- Fix Secure FTP
    - (TF-TFS-4083): Users no longer have to select “Upload directory from server” to enable the “Include Subdirectories” checkbox.
    - (TF-TFS-9939): Corrected an issue where component would display the error message “Handle not found” when uploading to root directories for some sites.
    - (TF-TFS-9943): Corrected an issue where component would not delete a file for certain ftp types.
    - (TF-TFS-10389): Fixed issue with component’s inability to delete directory files/delete files on certain SFTP sites.
- Fix SharePoint Source
    - (TF-TFS-9717): Hard coding a name filter no longer throws a “one or more fields are not installed

properly” error.
    - (TF-TFS-10011): Fixed issue with preview throwing an error due to trying to return 2 “Modified By” columns in a shared documents list.
- Fix Timer Task
    - (TF-TFS-1393): The user interface text doesn’t overlap when Windows Performance Options Visual Effects are set to "Adjust for best appearance"
- Fix XML Output Destination
    - (TF-TFS-9309): Corrected issue where 3-deep nested child attributes were not being included within the output.
    - (TF-TFS-10010): Fixed issue with component completing execution successfully, but displaying a “Could not remove table” error when execution completed.

Version 4.1.0.115 [January 15, 2015]

New Features:

- New Hadoop WebHDFS Source
    - This new component can be used to stream large files stored in the HDFS of a Hadoop server which can be converted into rows of data within SSIS.
- New Unpack Column Transform
    - The Unpack Column Transform can be used to extract XML or delimited data contained in a single column and “unpacked” or separated into user-defined column(s) to be loaded or merged downstream.

Updated Features:

- Update Advanced Execute Package
    - Update Execution Methods:
        - File System - Executes a child package by loading the package from the file system and executing it in the process of the parent package
        - SQL Server (MSDB, File System) - Executes a child package by copying the package from SQL Server into the parent package and executing it in the process of the parent package.
        - Local Catalog Execution - Will execute a package by creating a project (ispac) in process of the parent package and assigning the child package, project level parameters and project level connection managers to the in-memory project.
        - Server Catalog Execution - Will execute a package via server catalog execution (out of process) and wait on the package to finish execution on the server.

Fixes:

- Fix Advanced Derived Column
    - (TF-TFS-9022): "Too large for current buffer size" error is no longer produced when changing Test Input value to ntext.

    - (TF-TFS-9023): NTEXT columns that use the ToChar function no longer outputs the value twice.
- Fix Advanced Execute Package
    - (TF-TFS-8694): Parent package now properly recognizes environment variables.
- Fix Advanced Lookup Cache
    - (TF-TFS-9100): Advanced Lookup Cache can now process tables that use spaces in their column names.
    - (TF-TFS-9138): Dates in added conditions are now wrapped with ticks instead of pound signs.
    - (TF-TFS-9154): Advanced Lookup Cache no longer ignores "Use first value" when using key and date conditions.
- Fix Compression Task
    - (TF-TFS-8813): Compression task now properly releases connection when the task completes.
- Fix Data Validation
    - (TF-TFS-8827): An error is no longer thrown for MMDDYYYY unless it fails proper validation.
    - (TF-TFS-9078): Data validation now properly refreshes metadata.
- Fix Delete Batch
    - (TF-TFS-9079): Component now properly refreshes and can still be opened even though the configured connection manager is deleted.
- Fix Dynamics CRM Destination
    - (TF-TFS-7846): The Dynamics Destination actions are now consistent in how to handle output.
- Fix Dynamics CRM Source
    - (TF-TFS-4527): Columns now properly return the same output data if using the same source.
    - (TF-TFS-8665): The filter for "Does not equal" now properly filters, and no longer produces the same result as the "Equals" filter.
    - (TF-TFS-9060): Product numbers with hyphens are no longer erroneously processed as dates.
- Fix Error Output Destination
    - (TF-TFS-8853): The correct value for DT_DBDATE columns are now output rather than NULL.
- Fix Fact Table Destination
    - (TF-TFS-8818): Fact Table Destination now properly refreshes lookups and column mappings when a new fact table or connection manager is selected.
    - (TF-TFS-8831): Fact Table Destination now executes as expected when selecting keys that use expressions.
    - (TF-TFS-8952): Adding a new lookup (without configuring it) and returning to the previous lookup no longer prevents you from selecting a new dimension table.
    - (TF-TFS-8987): Adding an exclamation point in the source name no longer produces an error.
    - (TF-TFS-8992): External metadata is refreshed within the UI when component is opened.
    - (TF-TFS-9040): Column names can now properly use spaces.
- Fix Oracle Upsert
    - (TF-TFS-8800): Oracle Upsert no longer throws an error when using an index that uses a tablespace.
- Fix Replace Unwanted Characters

    - (TF-TFS-9032): Replace unwanted characters now replaces rather than append the value when selecting replace column data option.
- Fix SalesForce
    - (TF-TFS-8816): Fixed an issue where when using two different connections in SalesForce, an error messages that were associated with the first connection would persist to the second connection.
    - (TF-TFS-8817): Edit query no longer needs to be checked and re-checked to enable editing when selecting a new connection manager. If checked, you will be able to edit across connection managers.
    - (TF-TFS-8830): Connection managers no longer produce "Unable to find token" error when correct credentials are supplied.
    - (TF-TFS-9025): Destination no longer throws the "Sequence contains no matching element" error.
- Fix Secure FTP
    - (TF-TFS-8832): Folders are no longer included when getting a list of files using SSH connections.
    - (TF-TFS-8834): "Check if File Exists" no longer returns false when the file exists.
    - (TF-TFS-9008): File filters using SSH connections within the SFTP component are now respected and only return the files designated in its configuration.
- Fix SharePoint
    - (TF-TFS-5912): SharePoint Connection Manager no longer throws an "Unspecified Error" when using a User Name and Password.
    - (TF-TFS-8830): Sharepoint Source and Destination connection manager throws “unable to find token” error when Sharepoint Documents with same connection could connect.
    - (TF-TFS-8838): SharePoint Source and Destination now properly refresh when changing connections.
- Fix Upsert Destination
    - (TF-TFS-9096): Upsert Destination now respects different connections set dynamically in a for-each loop.
- Fix XML Output Destination
    - (TF-TFS-9259): XML Output Destination no longer throws an "item with same key" error when used inside a for-each loop.

Version 4.0.0.930 [September 30, 2014]

New Features:

- New Fact Table Destination
    - Developers can perform multiple lookups and insert/update data into a Fact table within a single component.
    - Additional functionality includes: choosing between insert and update actions, configuring table join conditions to dimension tables from their source data, creating column aliases for multiple lookups on the same Dimension, and the ability to define a default value that prevents NULLS from

being added.

- New Hash Transform
  - Users can choose from a selection of algorithms to create hash values from input columns. Other features include: customizing the order of input, the option to enable padding, adding null replacement character(s), and defining separator character(s).

Updated Features:

- Update Address Parse Transform
  - (TF-TFS-8381): Users can select columns to not be included in the output.
  - (TF-TFS-8575): Component now recognizes the following address suffixes: CI = Circle, PY = Parkway, RW = Row, and HY = Highway.
- Update Advanced Execute Package Task
  - (TF-TFS-8351): Details of a package being executed by the component can be seen in the execution results / progress.
- Update Compression Task
  - (TF-TFS-8391): Component now supports Deflate64 compression / decompression.
- Update Secure FTP Task
  - (TF-TFS-7426): Improved performance when using SSH connections.
  - (TF-TFS-7739): Component now performs a check to "Reuse Connections" when that option is selected and a connection is made during execution.
- Update Salesforce Destination
  - (TF-TFS-7582): The component now outputs a column that identifies each row as being inserted, updated, or deleted.
- Update Task Factory Registration
  - (TF-TFS-7697): Registration application now displays the version installed regardless of product being registered.
- Update XML Output Destination
  - (TF-TFS-7746): Component now supports mapping root level elements / attributes

Fixes:

- Fix General Usability
  - (TF-TFS-3827): Components no longer attempt to validate metadata when SSIS is being used in Offline Mode.
- Fix Advanced Derived Column (Expression Engine)
  - (TF-TFS-7752): ToMoney function incorrectly rounded DT_CY (currency) columns.
  - (TF-TFS-7844): Resolved issue where expression engine incorrectly lowered numeric columns by .01.
  - (TF-TFS-7859): Corrected the ToChar function to not display milliseconds when omitting it in the datetime format configuration.
  - (TF-TFS-8435): Resolved issue with various functions missing their example expressions.

    - (TF-TFS-8451): Corrected SysTimeStamp function returning an incorrect date when formatting the year.
    - (TF-TFS-8464): Hex expression incorrectly returned original value rather than Hex value.
    - (TF-TFS-8631): Resolved issue with GetDatePart function throwing an 'Expression not declared' error.
    - (TF-TFS-8712): Resolved issue with expression editor misplacing functions when drag and dropped from the functions list.
- Fix Case Transform
    - (TF-TFS-7698): Resolved issue with Irish surnames (such as O'Connor) not being capitalized properly.
- Fix Data Flow Nugget Transform
    - (TF-TFS-7253): Resolved issue with certain packages throwing a 'key not present in dictionary' error on execution.
- Fix Dimension Merge Slowly Changing Dimension Transform
    - (TF-TFS-8759): Resolved issue where NULL expiry dates were being sent to the Invalid Input although the component is configured to use NULL as its last input expiry date.
- Fix Dynamics CRM Source
    - (TF-TFS-8759): Component now produces a clear and graceful error message when the component's filter is missing a configuration value.
- Fix Email Source
    - (TF-TFS-7584): When configuring the "From" filter, users can choose to use display names or email addresses.
- Fix Error Output Description
    - (TF-TFS-8510): In 2005, connecting the component to a source's error output no longer causes the component to "hang" on execution.
- Fix Oracle Upsert Destination
    - (TF-TFS-7605): Component now outputs binary data from DT_NTEXT input columns instead of "microsoft.sqlserver.dts.pipeline.blobcolumn."
- Fix PGP Task
    - (TF-TFS-7806): Resolved execution issue for using keys created outside of Task Factory.
- Fix Salesforce Source
    - (TF-TFS-8359): Component now refreshes the SOQL query when selecting a new connection manager.
- Fix Sharepoint Source
    - (TF-TFS-6544): Resolved issue with users having to restart BIDS / SSDT for the Remove ID's selection to persist.
    - (TF-TFS-7544): Execution Results / Progress incorrectly displayed error column's description name rather than the actual column name.
    - (TF-TFS-8468): When clicking the Preview Data button, all output columns are displayed.

- Fix Timezone Conversion Transform
  - (TF-TFS-8323): Component made incorrect conversion when using Europe/Istanbul.
- Fix Upsert Destination
  - (TF-TFS-8677): Output columns with DT_CY (currency) datatypes now match native ADO.NET destination precision and scale.
  - (TF-TFS-8796): Resolved issue with component throwing an error when using tables that included spaces within their name.
- Fix XML Generator Transform
  - (TF-TFS-7142): Users receive an error message when closing the component without mapped columns.

V3

Version 3.7.0.702 [July 1, 2014]

New Features:

- New SQL 2014 Support
  - (TFS-5925): Task Factory now supports SQL Server 2014.
- New Dynamics 2013 Support
  - (TFS-5925): Task Factory Dynamics CRM Source and Destination now supports 2013.
- New Error Output Description
  - (TFS-6832): Error output description now captures all primary keys from multiple sources.
- New XML Generator
  - (TFS-7121): XML Generator columns and temp file path can now be configured using variables. New SalesForce Destination
  - (TFS-6962 & 7565): SalesForce Destination now includes BulkZip mode processing. This enables NTEXT and binary data to be sent to SalesForce.com in bulk copy mode as well as the ability to attach files to records.
- New SharePoint Connection Managers
  - (TFS-5339): SharePoint connection managers now support Claims Based Authentication.
- New XML Output Destination
  - (TFS-7120): Created the new XML Output Destination component (contains similar functionality to XML Generator Transform with a few added features).
- New XML Generator
  - (TFS-7123): XML Generator now includes an option for "pretty" output formatting.

Updated Features:

- Update (TFS-6914): Task Factory now includes support for SharePoint 2013.
- Update (TFS-7122): XML Generator Transform / XML Output Destination no longer outputs optional attributes when those values are NULL or empty.

- Update (TFS-7124): XML Generator Transform / XML Output Destination no longer outputs XML prolog and Document Type Declaration.

Fixes:

- Fix (TFS-3840): Metadata information is now included for Error OUtput Description within help documentation.
- Fix (TFS-4068): Secure FTP Task no longer fails to download files located in subfolders using SFTP-SSH connection option.
- Fix (TFS-6908): Delete and Move Emails Task no longer generates an "Unable to cast object" error when package is deployed to the SSIS Catalog.
- Fix (TFS-7024): Advanced Derived Column no longer generates the error "LineageID not found upstream" even when all upstream components are refreshed and saved.
- Fix (TFS-7044): SharePoint Destination no longer throws an "Object reference" error when selecting an action before selecting a list.
- Fix (TFS-7119): Corrected an issue where Null Handler Transform did not convert all nulls to user defined values.
- Fix (TFS-7121): Users can now configure XML Generator Transform's temp file location with a variable or project parameter.
- Fix (TFS-7235): Fixed and issue where XML Generator Transform did not allow mapping elements for certain XSD files.
- Fix (TFS-7272): Dynamics CRM Destination batch textbox no longer stays fixed in place when the UI is resized.
- Fix (TFS-7273): Dynamics CRM Destination no longer outputs "same key already added"
- Fix (TFS-7278): XML Generator Transform no longer incorrectly outputs Nvarchar(max) datatypes as "Microsoft.SqlServer.Dts.Pipeline.BlobColumn" instead of the actual value.
- Fix (TFS-7315): Fixed an issued where Dimension Merge SCD incorrectly processed null values even when column was not selected as "Nullable" in the Row Change Detection tab.
- Fix (TFS-7338): Advanced Lookup Cache Transform no longer throws "Object Reference" error when using fixed input parameters in a ranged lookup.
- Fix (TFS-7393): Updated Dimensions Merge SCD help documents to improve component descriptions.
- Fix (TFS-7431): Upsert Destination no longer throws an error when reserved keywords are used in a table name within the Merge Statement.
- Fix (TFS-7434): IfNull and IfNullorEmpty functions now include configuration information and examples within the expression engine UI.
- Fix (TFS-7605): Fixed an issue where Oracle Upsert incorrectly output "microsoft.sqlserver.dts.pipeline.blobcolumn" instead of an actual value when using NTEXT input.

Version 3.6.0.328 [April 2, 2014]

Updated Features:

- Update (TFS-none): XML Generator now includes an option to configure a temp file which significantly improves the component's performance when producing large XML files.
- Update (TFS-6723): Advanced Email and SMS Task now includes the ability to create an error log that is configurable in the component's properties.
- Update (TFS-6836): Dynamics CRM Source now includes ActivityParty as a selectable entity.

Fixes:

- Fix (TFS-6236): Timezone Conversion Transform incorrectly converted the 2014 DST dates for Santiago (listed as America/Santiago in the UI).
- Fix (TFS-6289): SharePoint Source and Destination UI incorrectly displayed the encoded custom destination columns.
- Fix (TFS-6315): SalesForce Source and Destination produced an XML error when using the Geolocation custom fields (these custom fields were recently introduced by SalesForce).
- Fix (TFS-6448): SalesForce Destination failed to insert/upsert/update data when Opportunity object and bulk process mode were used together.
- Fix (TFS-6546): Expression engine's "RegMatch" function incorrectly produced the error "specified argument was out of the range of valid values" when ignore case parameter was set to True.
- Fix (TFS-6566): Expression engine's "RegExtract" and "RegReplace" functions contained an error in UI's example expression.
- Fix (TFS-6591): Case Transform/Data Cleansing Transform improperly failed to capitalize addresses with "NE" directional suffix.
- Fix (TFS-6602): Expression engine's "FileWriteAllBytes" function improperly displayed "conversion from 'String' to '1-dimensional array of Byte'" error on execution.
- Fix (TFS-6630): File Properties Task incorrectly produced an "Illegal characters in path" error when using a valid Excel connection manager.
- Fix (TFS-6634): SharePoint Destination incorrectly produced an "invalid value for column being updated" error when a custom column was included in the update/create
- Fix (TFS-6640): Dynamics Destination improperly retained a batch size of 500 on execution even when changed in the UI.
- Fix (TFS-6649): Expression engine's "IsNullOrEmpty" function does not appear in the expression editor's UI.
- Fix (TFS-6652): Upsert Destination improperly displayed "Different datatype" warning even though the datatypes were the same.
- Fix (TFS-6653): Upsert Destination advanced editor would crash the UI when selecting "Component Properties" tab followed by "Column Mappings" tab.
- Fix (TFS-6677): SCD Dimension Merge did not support datetimeoffset datatypes for SCD1 and SCD2 changes *(datetimeoffset is NOTsupported as effective and expiry dates within the component).
- Fix (TFS-6716): SalesForce connection manager failed to display the user credentials when it was closed and re-opened.

- Fix (TFS-6717): Expression engine's "GetConnectionProp" function contained an error in the function's example.
- Fix (TFS-6724): Expression engine's "Least" function incorrectly observed first value as an input rather than the case flag value.
- Fix (TFS-6732): Expression engine's "Pv" (present value) function was missing the configuration details, description, and example in the UI.
- Fix (TFS-6738): Oracle Upsert incorrectly produced an "ORA-00600: internal" error when processing various number of rows.
- Fix (TFS-6758): Dynamics Source incorrectly produced an error when configuring a filter that returned no rows.
- Fix (TFS-6790): Timezone conversion improperly converted all millisecond values to .000 when output column datatype was set to DT DBTIMESTAMP or DT DATE.
- Fix (TFS-6793): Download File Task incorrectly produced "OK - No Content" error when the web service returned a 204 message thus causing the task to fail.
- Fix (TFS-6837): Expression engine's "Fv" (future value) function example was missing from the UI.
- Fix (TFS-6854): SalesForce Source was unable to connect to a remote server when the sandbox web service URL contained more than 1 separator "." that preceded "salesforce.com".
- Fix (TFS-6890): Upsert Destination component failed to upgrade when BIxPress Auditing Framework was applied to packages.

Version 3.6.0.1018 [October 18, 2013]

⚠ **Important:** If you are using any instances of Dimension Merge that use a Millisecond granularity and are using the SQL DateTime Max as the "last record's expiry date", this build could potentially cause issues with your data.

We have updated SQL DateTime Max to the correct value of 12/31/9999 23:59:59.997 from the incorrect value 12/31/23:59:59.999. This will cause your Dimension Merge to create new rows of data.

New Features:

- New Timezone Conversion Transform Allows SSIS developers to convert a date from one Timezone to another using the TZ Database for the IANA (see more).
- New Oracle Upsert Destination An almost exact duplicate of our very popular Upsert Destination which allows SSIS developers to upsert data into an Oracle table (see more).
- New SharePoint Documents Tasks Allows SSIS developers to upload and download documents from a SharePoint documents list (see more).

Updated Features:

- Update Dimension Merge now has support for Type 0 Dimensions. It also now supports millisecond

precision levels.

Version 3.5.0.329 [April 2, 2013]

New Features:

- New Dynamics CRM Source Allows SSIS developers to retrieve data from Dynamics CRM Servers (see more).
- New Dynamics CRM Destination Allows SSIS developers to send data to Dynamics CRM Servers (see more).

Updated Features:

- Update SharePoint Connection Manager can now connect to online SharePoint servers (Live, Office 365).
- Update SalesForce Destination now contains the ability to set column level external id's for referenced objects.

Version 3.2.0.0 [January 23, 2013]

> ⚠ **Important:** Please read about our licensing changes here.

New Features:

- New XML Generator Transform Allows SSIS developers to create XML documents based on XML's Schema (see more).

Fixes:

- Fix Fixed a Data Flow Nugger Transform issue where variables from the selected data flow task were not showing up on the "Map Variables" step.
- Fix Fixed an Upsert Destination issue where opening the UI in disconnected mode caused an error (0x80131904) to be thrown.
- Fix Fixed an Upsert Destination issue where Upsert was not properly updating data where the destination column was a date column, the destination was null and the incoming date was 1-1-1900.
- Fix Fixed a SalesForce Source issue where the "Include Deleted / Archived Records" flag was being ignored when reinitializing metadata.

Version 3.1.0.0 [November 26, 2012]

New Features:

- New Data Flow Nugget
    - Brings reusable packages to SSIS (see more).

Updated Features:

- Update In Email Source, the BodyText property was changed to DT_NTEXT.
- Update Added support for Bulk Inserts, Updates and Upserts in SalesForce Destination.
- Update In Terminator Destination, a UI was added to allow users to choose included columns.
- Update Added the ability to choose the SSL type in connection manager for SFTP Task.

Fixes:

- Fix Fixed an Error Output Description issue with running it in SQL 2012.
- Fix Fixed an Expression Based Tasks issue with DR_Decimal columns scale resetting to zero.
- Fix Fixed To_Date function in Expressions Based Tasks.
- Fix Fixed an issue with deleting connection managers in packages that have Expression Based Tasks contained in them.
- Fix Fixed the Abort function in Expression Based Tasks.
- Fix Fixed the SETVARIABLE function in Expression Based Tasks.
- Fix Fixed the IIF statement in Expression Based Tasks.
- Fix Fixed an Expression Based Tasks issue where the datatype does not show up in the UI.
- Fix Fixed a Data Validation Task with "Value does not fall within the expected range".
- Fix Fixed an SFTP Task issue with incorrectly calculating time-outs.
- Fix Fixed the error "Value was either too large or too small for an unsigned byte"e; in Dimension Merge SCD.
- Fix Fixed a SalesForce Source UI issue where SQL statement was enabled by default causing metadata issues.

V2

Version 2.3.0.0 [May 14, 2012]

Updated Features:

- Update All Tasks are now supported in SQL 2012.

Fixes:

- Fix Fixed an Address Parse issue where duplicate fields were created when the alias of the output column was changed.
- Fix Fixed a SalesForce Source issue with parsing sub-queries.

Version 2.2.0.0 [March 5, 2012]

New Features:

- New HTTP Download Task Download files from the internet without FTP over HTTP.
- New Advanced Execute Process Task Execute processes as if you were using the command line SSIS.

Allows for variable replacement in the commands. Allows multiple steps per instance of the task.

- New Advanced Execute Package Task Execute child packages from within SSIS. Allows for the mapping of variables between the two packages.

Updated Features:

- Update The Surrogate Key Transform has been updated to allow the user to save the sequence value to a database as well as reset sequence values and the recycling of sequence values.
- Update The notification pane for Dimension Merge SCD was updated for easier viewing.
- Update Added more granular address data including Street Number, Street Name, Street PreDirection, Street Post Direction, Suite Type, Suite Number, Box Type, Box Type Number, Box, Box Number in the Address Parse Transform.
- Update Added Error Handling Output choice to the Address Parse Transform UI.
- Update Added support in Advanced Email and SMS for variable replacement place holders in the subject.
- Update Added Support in Advanced Email and SMS for the SMTP connection to not have a user name or password supplied.
- Update Added the ability in SalesForce Destination to refresh the columns from the SalesForce object without having to drop and recreate the destination.
- Update Added the ability in SalesForce Destination to grab the SalesForce ID of newly inserted rows.
- Update SalesForce Source UI improved to make it easier to use.
- Update Added error output to SalesForce Source.
- Update Added an option in SalesForce Source to use only the object name rather than forcing the use of a query.
- Update Added the ability to copy any action to all columns in Null Handler Transform, Case Transform, RegEx Replace Transform, Replace Unwanted Characters Transform, and Trim Plus Transform.
- Update Updated Upsert Destination to handle null GUIDs values.
- Update Updated Upsert Destination to handle null values in Column Compare mode.
- Update Updated PGP Task to handle more encryption types.

Fixes:

- Fix Fixed a Dimension Merge SCD issue where changing the SCD column type caused the window to refresh and lose it's place.
- Fix Fixed Dimension Merge SCD to allow December 31, 1999 as a valid expiry date.
- Fix Fixed an Email Source error where not selecting the connection manager caused the UI to fail.
- Fix Prevented SMTP connection managers from being selected in the Email Source UI, this would cause the source to fail.
- Fix Fixed a Data Validation Task issue where opening the task caused a "string was not recognized" error.
- Fix Fixed a Data Validation Task issue where opening the task caused a "object not set to a reference" error.
- Fix Fixed a SalesForce Destination error that was throwing the message """" is not valid for the type

xsd:double". "'' is not a valid value for the type xsd:date".

- Fix Fixed a SalesForce Destination issue that caused the message "EXCEEDED_IT_LIMIT: record limit reached" error.
- Fix Fixed a SalesForce Destination error that caused the message "Dictionary cannot be null. Parameter name: d".
- Fix Fixed a SalesForce Destination issue where the External ID was not saving correctly.
- Fix Fixed the SalesForce Destination error output to work correctly.
- Fix Fixed an XML Destination issue where clicking the "Move row up" button caused an "object not set to a reference error".
- Fix Fixed an issue that did not allow he user to select a variable as a replacement value in Null Handler Transform, Case Transform, RegEx Replace Transform, Replace Unwanted Characters Transform, and Trim Plus Transform.
- Fix Fixed an SFTP issue where downloading text files in ASCII was not adding in the newline characters.
- Fix Fixed an Error Output Description error that was causing "An entry with the same key already exists".

Version 2.0.1.0 [May 17, 2011]

Fixes:

- Fix Fixed a SalesForce.com Connection Manager/Source/Destination issue where running the connection manager again the sandbox server where the source and destination could not retrieve objects from the server.

Version 2.0.0.0 [April 19, 2011]

New Features:

- New Dimension Merge SCD Transform A "Kimball Method" component that is superior to the native Slowly Changing Dimension Transform.
- New XML DestinationAllows the user to export XML within SSIS from source data.
- New Replace Unwanted Characters Transform Allows the user to replace unwanted / invalid characters in a data source.
- New NULL Handler Transform Allows the user to replace NULL or blank data from source data.
- New RegEx Replace Transform Allows the user to replace data from source data with a regular expression
- New TrimPlus Transform Allows the user to trim characters from the beginning and end of source data.
- New Case Transform Allows the user to properly case source data using predefined algorithms.

Updated Features:

- Update Added the option of using variables as part of the SOQL query in SalesForce.com Source.

Fixes:

- Fix Fixed a SalesForce.com Source issue with not retrieving all the SalesForce columns.

V1

Version 1.5.8.0 [February 1, 2011]

Fixes:

- Fix Fixed an Upsert Destination issue where upgrading from version 1.4 to 1.5 caused the task to not open correctly.
- Fix Fixed a SalesForce.com Destination issue where invalid characters were apart of the XML document being sent to SalesForce.

Version 1.5.7.0 [January 7, 2011]

Updated Features:

- Update Added support for use of the DTC for Upsert Destination, Update Batch, and Delete Batch.
- Update Added new feature to Email Source to allow variables to be used in filters.

Fixes:

- Fix Fixed an issue with Upsert Destination, Update Batch, and Delete Batch where changing the name of a connection manager caused the task to reset.
- Fix Fixed a Compression Task issue with compression using BZ2: "[Execute] Error: Attempted to read or write protected memory. This is often an indication that other memory is corrupt.".
- Fix Fixed a SharePoint Connection Manager issue with not being able to use SharePoint Source or Destination with windows authentication.

Version 1.5.2.0 [December 6, 2010]

New Features:

- New SharePoint Source SharePoint Source is used to retrieve data from a SharePoint server.
- New SharePoint Destination SharePoint Destination is used to send data to a SharePoint server.

Updated Features:

- Update Added support for Updates, Deletes, and Upserts in SalesForce.com Destination.
- Update Added support for pulling embedded images out of emails in Email Source.
- Update Added the ability to replace NULL text values from sources with empty strings in Address Parse. This fixes an issue where passing the string "NULL" into an address cause the address not to be parsed correctly

Version 1.4.10.0 [October 15, 2010]

Updated Features:

- Update Added support for sandbox URLs in SalesForce.com Connection Manager.

Version 1.4.9.0 [September 21, 2010]

Updated Features:

- Update Added support for proxies with SalesForce.com Connection Manager.
- Update Added Email Source support for deleting messages off the server (IMAP ONLY).

Fixes:

- Fix Fixed an Email Source issue where an error would occur during processing which caused an EndOfRowSet error but did not display the true error.

Version 1.4.4.0 [September 10, 2010]

Updated Features:

- Update Added a new SFTP Task action "Check if file exists on server" which allows the user to verify a file exists on a server.
- Update Added a File Filter to the "Download directory from server" which allows the user to define a filter on the SFTP task to only download files that match a

Fixes:

- Fix Fixed an SFTP Task error where downloading the directory from the server was not working for SSH.
- Fix Fixed an issue where the new installation package was not properly installing the Task Factory files on some Windows 2003 machines.

Version 1.4.0.1 [August 9, 2010]

New Features:

- New PGP Task With the use of Open PGP Technology, the PGP Task can be used to encrypt and decrypt files as well as creature and verify signature files using PGP.

Updated Features:

- Update The new updates to Upsert Destination allow the user to choose their update model. This will allow the user greater control of how updates occur in the destination table and see the number of rows updated.
- Update Provided a way in Upsert Destination to update the "Last Update" column in the destination table.
  Update Upsert Destination can now compare the data in selected columns to only perform an update if this columns do not match. This allows the user to also get an accurate count on how many rows were

updated.

- Update Upsert Destination can no perform updates by comparing a "last updated" column.

Fixes:

- Fix Fixed an Upsert Destination "null data" issue when the identity column key data is null.
- Fix Fixed an issue in Cleansing Transform where the OK and Cancel button's slide would not float when re-sizing the window.
- Fix Fixed an Upsert Destination issue that did not allow mapping to read only columns.

Version 1.3.4.0 [June 30, 2010]

New Features:

- New Address Parse Transform Used to parse unformatted address data into USPS standardized address data.
- New Data Cleansing Transform Used to cleanse data from a source by applying up to 16 different data cleansing algorithms to the data.
- New SalesForce.com Source Used to retrieve data from SalesForce.com objects.
- New SalesForce.com Destination Used to send data to SalesForce.com object destination.

Updated Features:

- Fix Fixed an issue where Delete Batch Transform allowed a negative time-out.
- Fix Fixed an issue where Surrogate Key allowed a zero increment.
- Fix Fixed an issue where Upsert Destination did not turn red when an error occurred.
- Fix Fixed an Issue where Delete Batch Transform failed in the middle of a transform yet the package finished running instead of stopping.
- Fix Fixed an issue where changing the connection manager in Upsert, Update Batch, and Delete Batch did not change the tables available in the task to reflect the new connection manager.
- Fix Fixed an issue where LastWriteTimeUtc did not function correctly in File Properties Task.

Version 1.1.4.0 [November 10, 2009]

New Features:

- New Surrogate Key Transform Used to add a unique identifier to a data flow. Allows the use of a variable to determine seed and increment.
- New Email Source Used alongside an IMAP / POP3 connection to read e-mails from a mailbox. Supports most e-mail providers, including Gmail.
- New Delete Batch Transform Will delete a batch of rows in a table based on the source and selected keys.
- New Update Batch Transform Will update a batch of rows in a table based on the source and selected keys.

- New Advanced Email and SMS Used to send HTML e-mails and/or SMS messages from within an SSIS Package.
    - Allows for the use of package and system level variables to be used within the messages.
- New Terminator Destination Stops the flow of a data flow task without any configuration.

Version 1.0.3.0 [September 29, 2009]

Fixes:

- Fix Fixed an issue where the installer did not copy files when the client tools for 2005 are installed by themselves.
- Fix Fixed an issue where the help for Data Validation Form was unreadable.
- Fix Moved the Task Action for the Compression Task to the general tab.
- Fix Fixed an issue for Upsert Destination where text columns for NTEXT columns were garbled text.

Version 1.0.0.0 [September 9, 2009]

New Features:

- New File Properties Task Used to read the properties (Creation time, Is File in use, Hidden, Read-only, et cetera) of an existing file.
- New Compression Task Used to compress or decompress files and directories.
    - Supports *.Zip, *.Z, and *.Bz2 compression algorithms as well as self-extracting executables.
- New Secure FTP Task Used to interact with a secure FTP site.
    - Supports SSH as well as Implicit and Explicit SFTP.
    - Supports HTTP, SOCKS4, SOCKS4a, and SOCKS5 proxies.
- New Upsert Destination Used to conditionally insert or update data into a table.
- New Data Validation Transform Validates data from a source with the use of regular expressions.
    - Regular Expression editor included as part of the UI.
    - Outputs the results of the data check in each row of the transform output.

Task Factory System Requirements

Last Modified on 30 December 2021

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Download: For more information about Task Factory pricing or downloading a trial, see the Task Factory product page.

Note: For a comprehensive view of the updates to SolarWinds Task Factory, see the Release Notes.

Hardware Requirements

Component	Minimum Requirements	Recommended Requirements
Memory	4 GB	At least 8 GB
Processor Speed	1.4 GHz	2.0 GHz or higher
Process Type	Not Applicable	• AMD Opteron • AMD Athlon 64 • Intel Xeon with Intel EM64T support • Intel Pentium IV with EM64T support

Note: Hardware requirements can vary based on workloads and other processes.

Software Requirements

The following is required to use Task Factory:

- Microsoft Visual Studio
- Microsoft .NET Framework 4.5
- SQL Server Integration Services 2012, 2014, 2016, 2017, or 2019.

- SQL Server Data Tools (SSDT) 2012, 2014, 2016 (for Visual Studio 2015), 2017 (for Visual Studio 2015 or 2017), or Visual Studio 2019.

> **Note:**
>
> - SSDT is a development environment requirement, however, it's not required on a server installation.
> - Task Factory works on any edition of Microsoft Visual Studio (Community, Professional, or Enterprise).

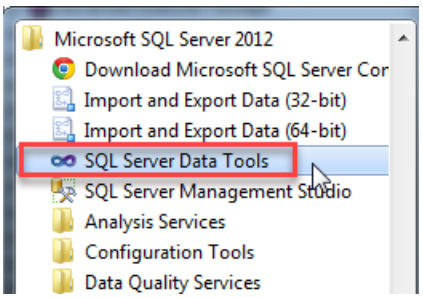
The following Windows Versions are supported by Task Factory:

64-bit (x64) Operating Systems:	32-bit (x86) Operating Systems:
Windows Server 2012	Windows Server 2012
Windows Server 2016	Windows 7
Windows Server 2019	Windows 8
Windows 7	Windows 10
Windows 8	
Windows 10	

Verifying SSDT or BIDS Installation

Complete the following steps to see if SSDT or BIDS is installed:

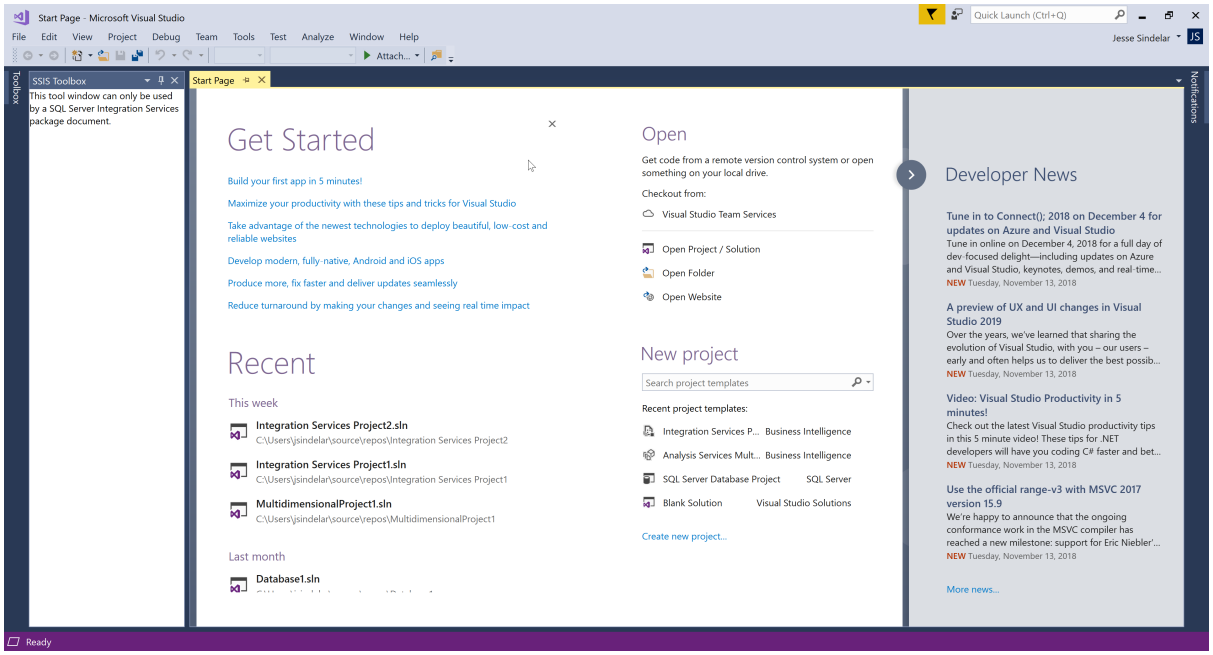
1. Open **Start** > **All Programs** > **Microsoft SQL Server** 2005, 2008, 2012, or 2014. You should be able to see **Business Intelligence Development Studio** or **SQL Server Data Tools** if it's installed.



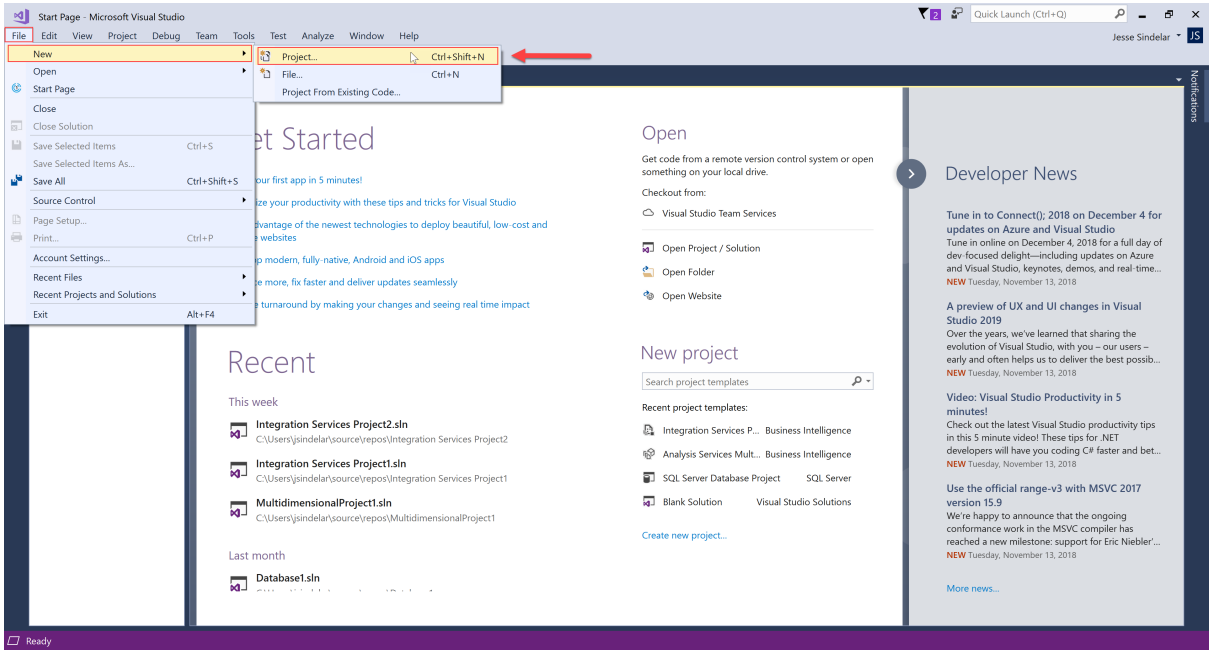
Verifying SSIS Installation

Complete the following steps to see if SSIS components are installed:

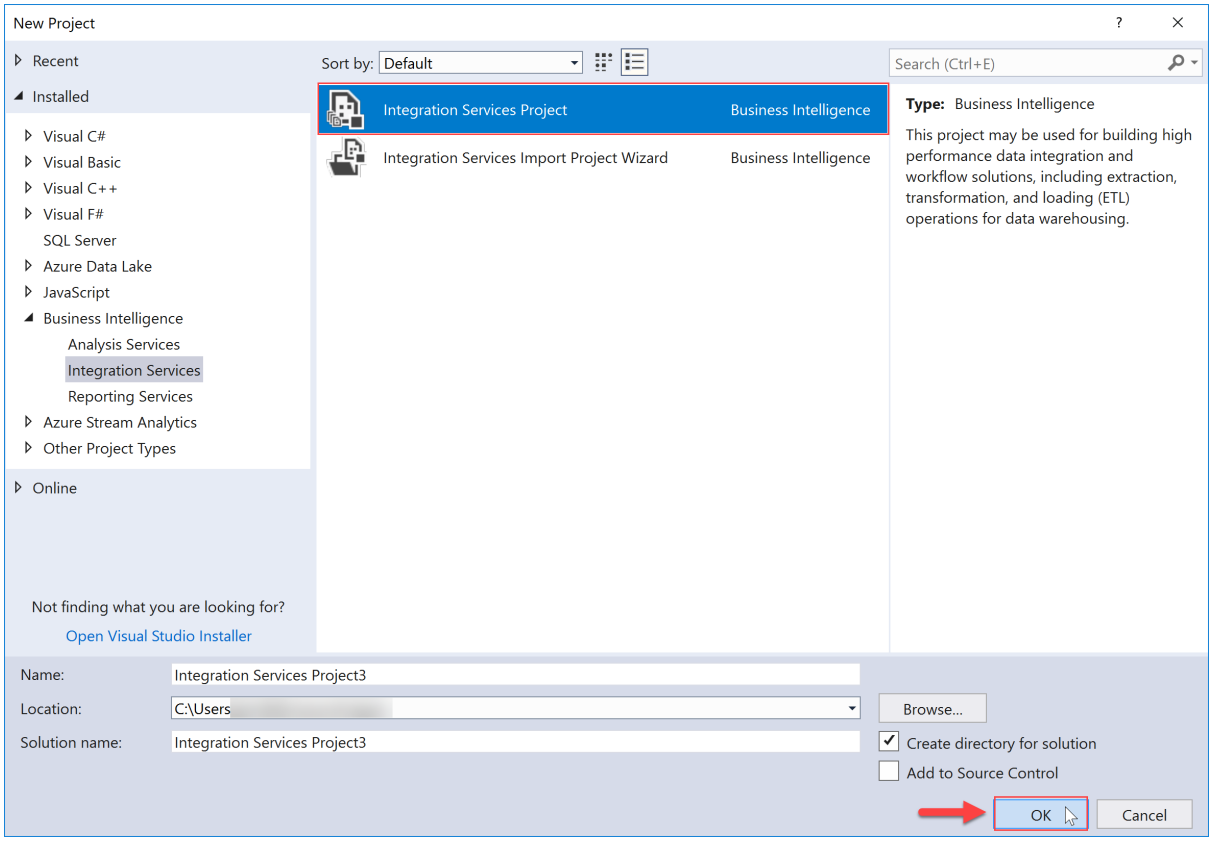
1. Open **BIDS** or **SQL Server Data Tools**.



2. Once open, select **File** > **New** > **Project**.



3. In the **New Project** dialog box, select **Business Intelligence Projects** on the left hand side. If you see **Integration Services Project** on the right hand side, then the SSIS components are installed.



Additional Requirements

Oracle Upsert Destination

The following are Oracle Upsert Destination Requirements:

For 64-bit Task Factory:	For 32-bit Task Factory:
• 64-bit Oracle Data Access Components (ODAC) 11g with Oracle Developer Tools for Visual Studio • 64-bit Oracle 12c Managed Data Access Client	• 32-bit Oracle Data Access Components (ODAC) 11g with Oracle Developer Tools for Visual Studio • 32-bit Oracle 12c Managed Data Access Client

Expression Based Components

Control Flow (e.g. Expression Task) and Data Flow (e.g. Advanced Aggregator Transform, Case Transform, Null Handler Transform, etc.) Components require read and write access to the **C:\Windows\TEMP** directory. This directory is used to store temporary code during package executions. If read or write access is denied to this directory, an error message similar to the following may be encountered:

System.Exception: file 'C:\Windows\TEMP\xyz.0.vb' could not be found no input sources specified at...

Additional Information: See the Task Factory Expression Based Components Require Read/Write

Access to C:\Windows\TEMP article for more details.

Installing Task Factory

Last Modified on 23 August 2022

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Download: For more information about Task Factory pricing or downloading a trial, see the Task Factory product page.

Prerequisites

Additional Information: See the System Requirements article for details on what's needed to install Task Factory.

Additional Information

- See the Release Notes for details on Task Factory versions
- For installing on Azure, see the Enabling Task Factory on Azure-SSIS IR in ADF article

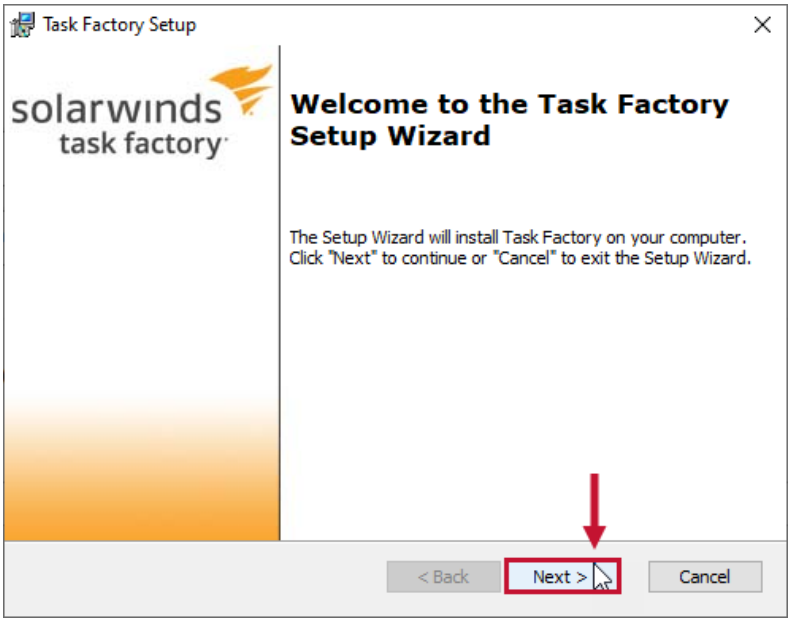
Installing Task Factory

Note: Task Factory must be installed on the server where the packages will be executing.

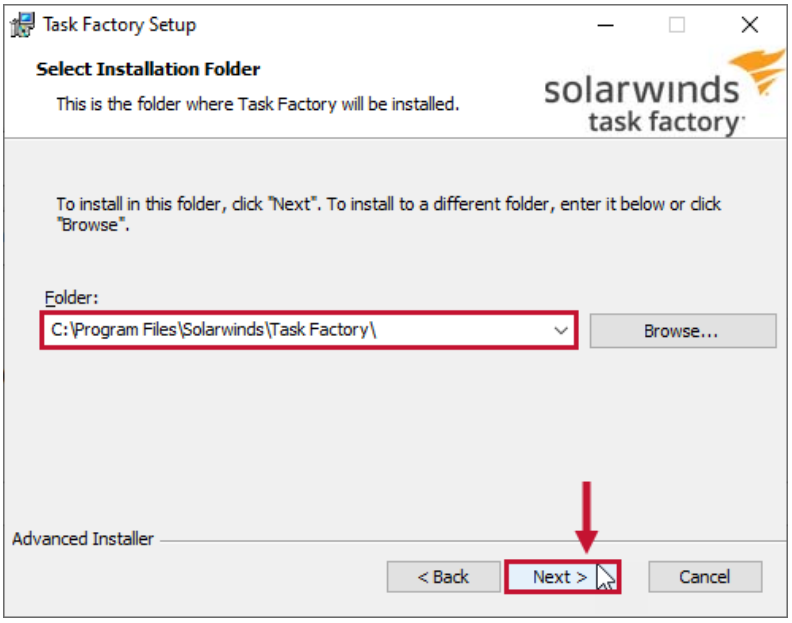
Task Factory Versions 2022.3 or later

1. Start the Task Factory installer by running the **TaskFactory.msi** file.

2. Select **Next** to continue with the Task Factory Setup installer.

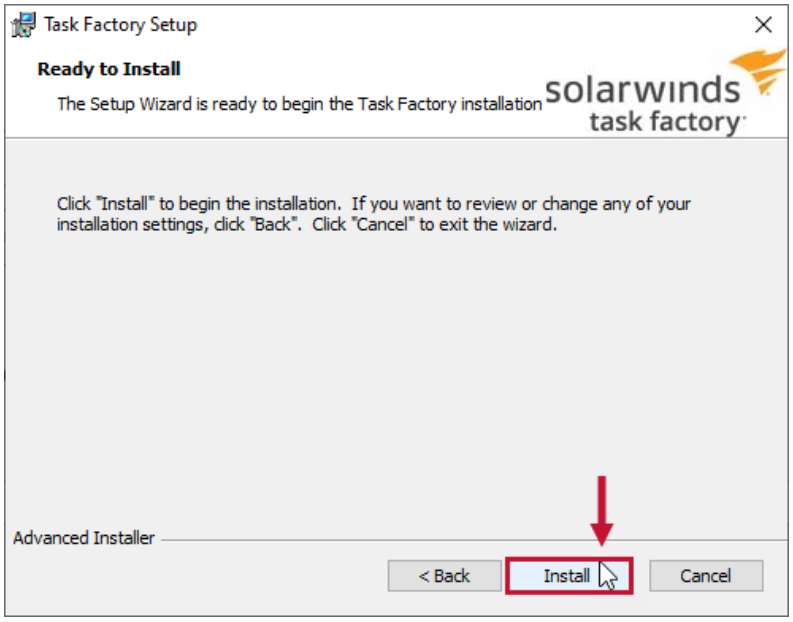


3. Select **Next** to Install Task Factory in the default installation path.



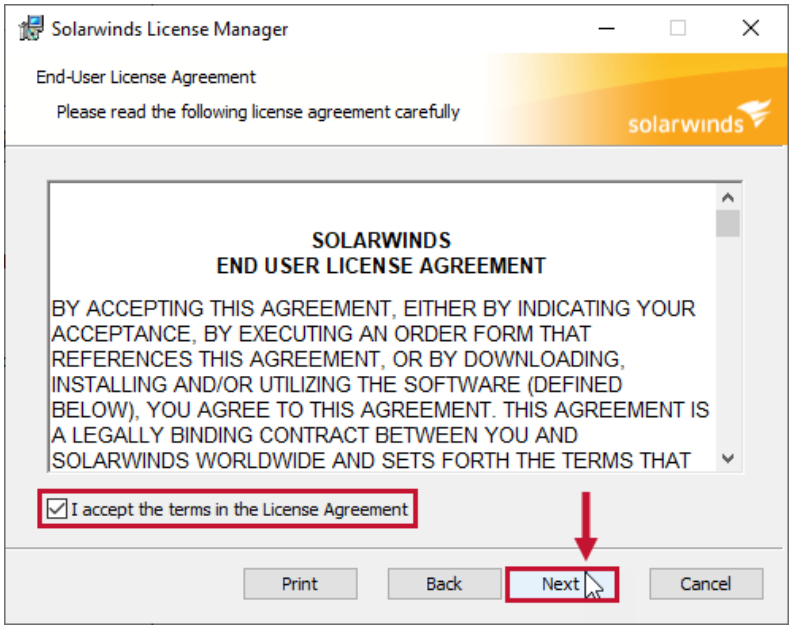
Note: You can change the Task Factory Installation folder by selecting **Browse**, and choosing a different location.

4. Select **Install** to install Task Factory on your machine.

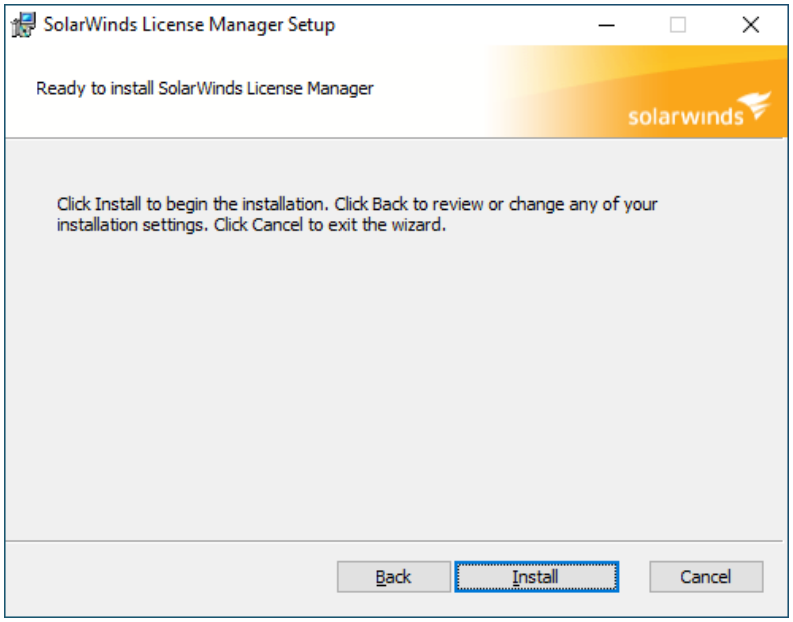


5. The Task Factory installation includes the **SolarWinds License Manager**. If the Setup Wizard seems stuck on the **Installing SolarWinds License Manager** step, check to see if there is a UAC warning or other window that requires focus to continue. Once the **SolarWinds License Manager Setup Wizard** appears, select **Next** to continue.

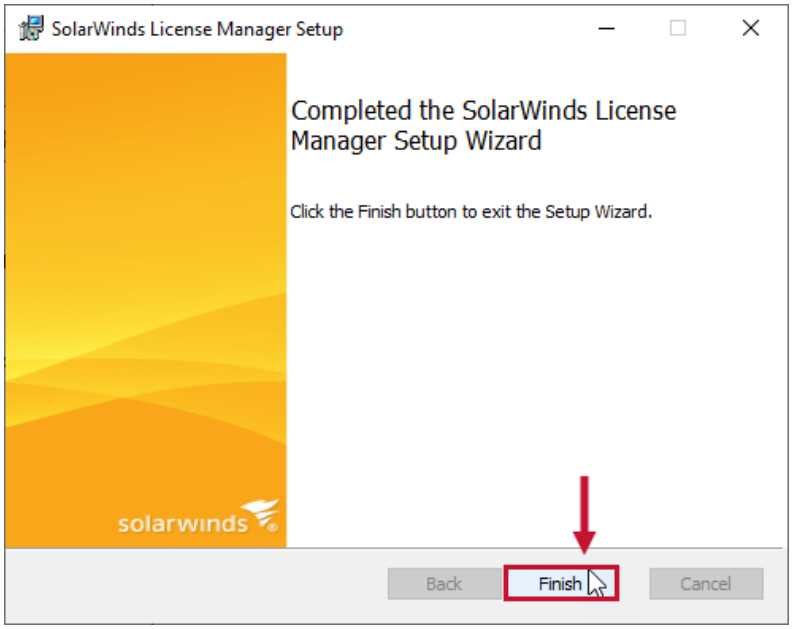
6. Check the **I accept the terms in the License Agreement** box, then select **Next** to continue.



7. Select **Install** to start the installation process for the **SolarWinds License Manager**.



8. Select **Finish** to close the **SolarWinds License Manager Setup** screen and complete the Task Factory installation.



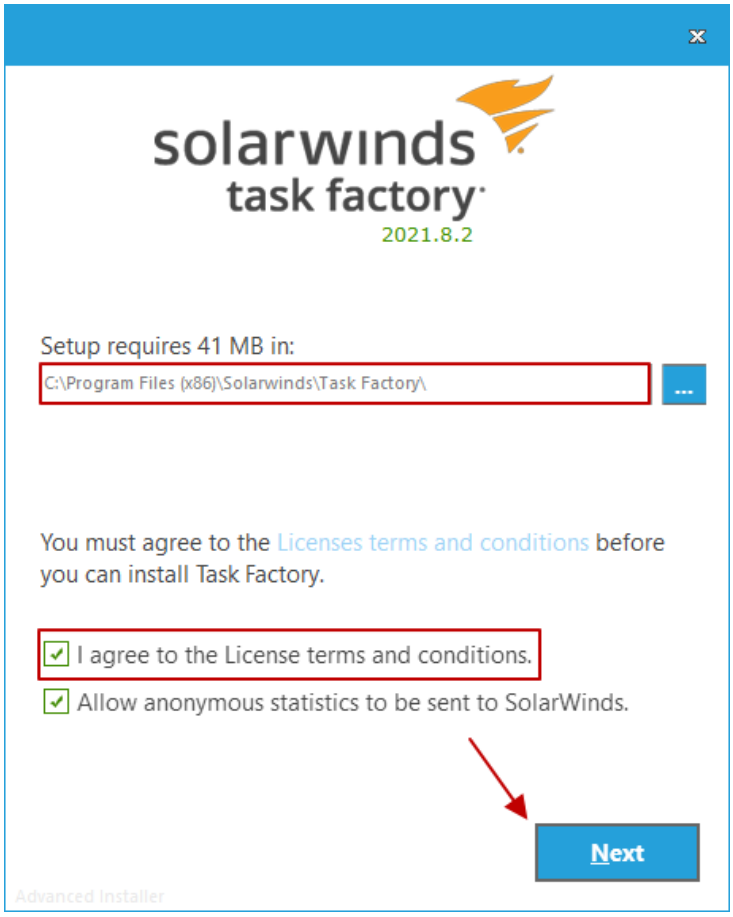
Success: The SolarWinds Task Factory installation is complete.

Task Factory Versions 2021.18 or earlier

Complete the following steps to install Task Factory:

1. Start the **Task Factory installer** by running the **TaskFactory.exe** file.

2. Select your install destination from the drop-box, then select the checkbox to **agree to the license terms and conditions**. Select **Next** to continue.

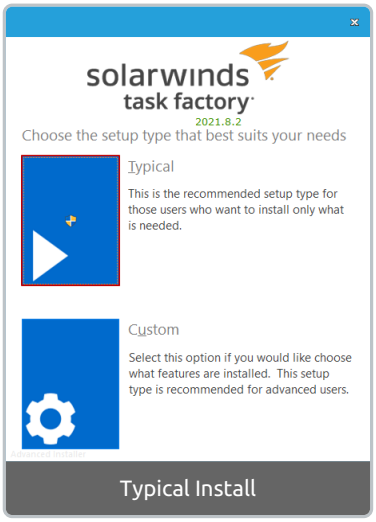


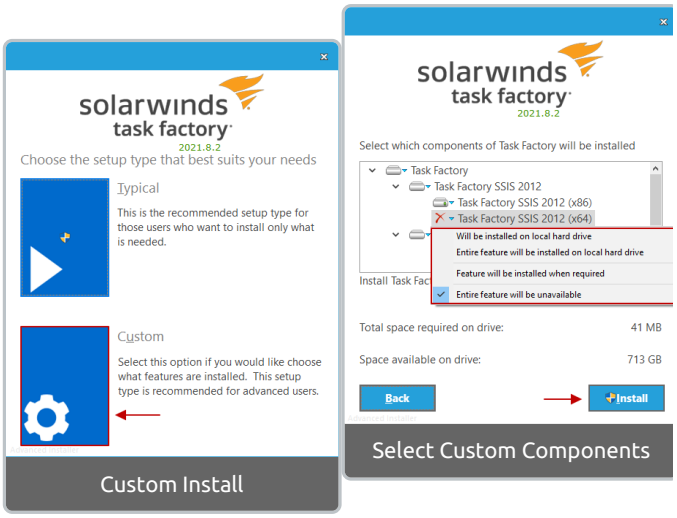
3. You may see a screen reminding you that you need a SolarWinds license for this version of Task Factory. Please ensure that you have your license from the SolarWinds Customer Portal before proceeding. Select **Install** to continue.

4. Select the **Typical** or **Custom** setup type, then select **install**.

> **Note:** The **Typical** setup is recommended for most customers.

> **Important:** If you choose the **Custom** installation for the 2016 version, you must include the 2012 and 2014 installations (which are selected by default).

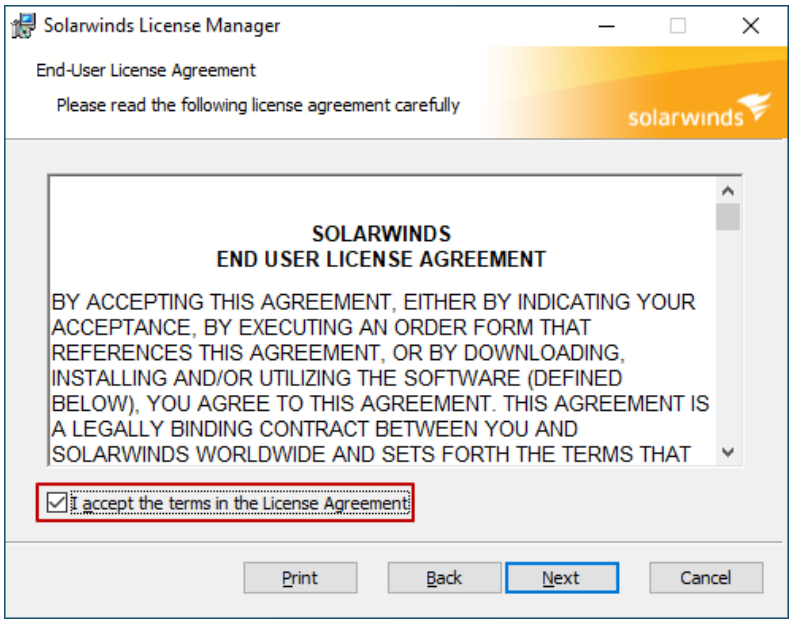




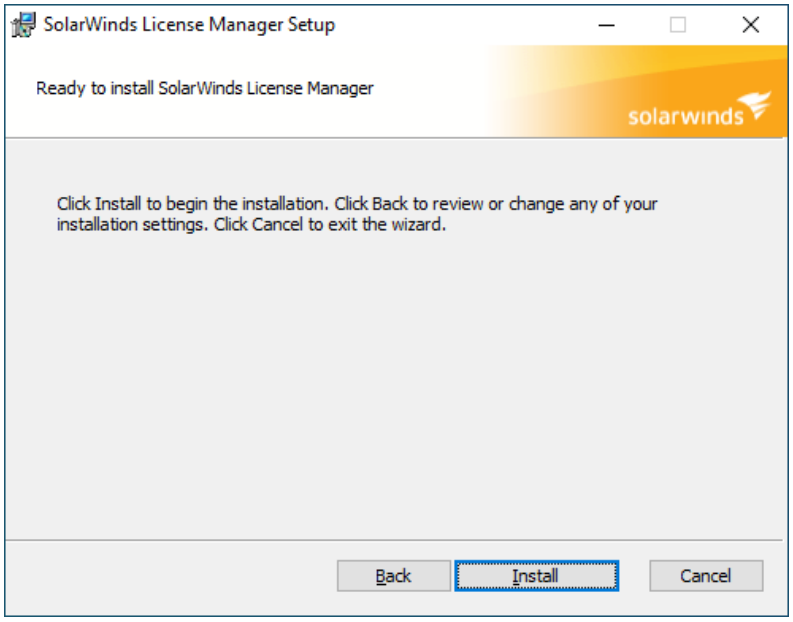
> **Note:** The installation process may take several minutes. You can monitor the status through the installer.

5. The Task Factory installation includes the **SolarWinds License Manager**. If the Setup Wizard seems stuck on the **Installing SolarWinds License Manager** step, check to see if there is a UAC warning or other window that requires focus to continue. Once the **SolarWinds License Manager Setup Wizard** appears, select **Next** to continue.

6. Check the **I accept the terms in the License Agreement** box, then select **Next** to continue.

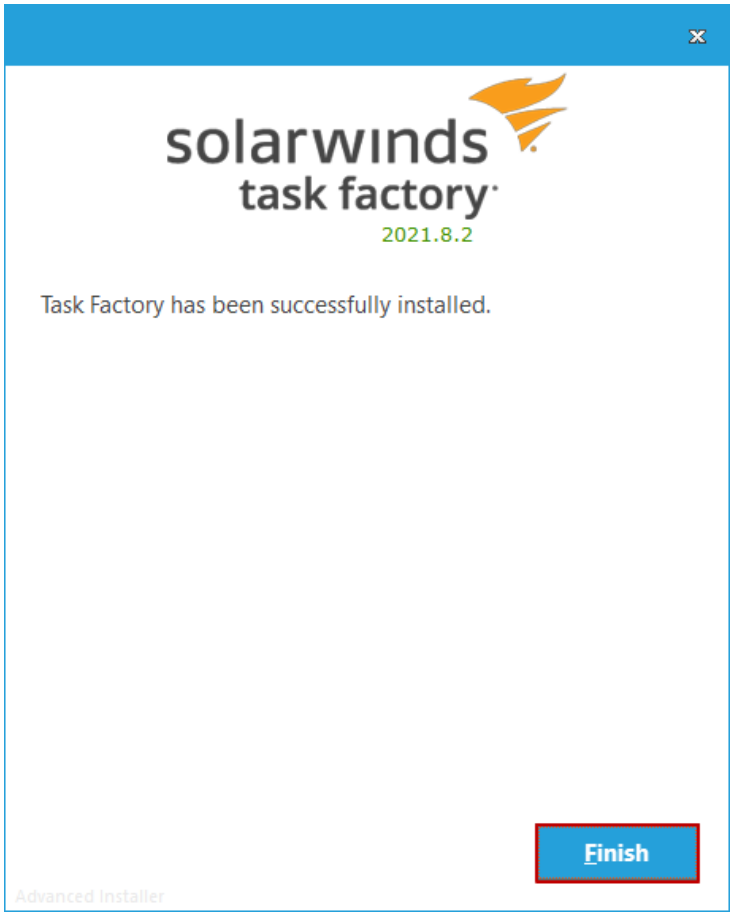


7. Select **Install** to start the installation process for the **SolarWinds License Manager**.



8. Select **Finish** to close the **SolarWinds License Manager Setup** screen and return to the **Task Factory Setup Wizard**.

9. After the installation is complete, the confirmation window displays. Select **Finish** to complete the installation.



Success: The SolarWinds Task Factory installation is complete.

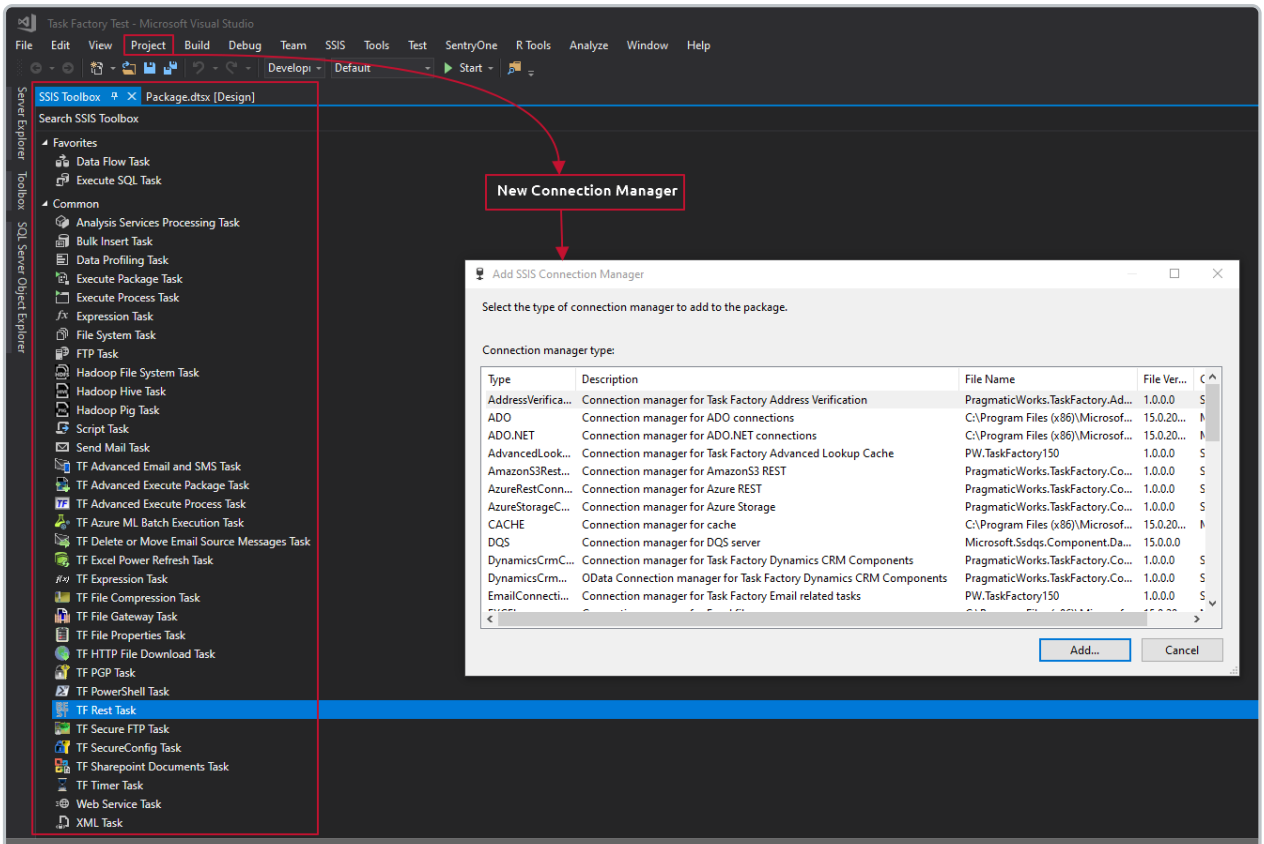
What's next?

Task Factory Licensing

See the Task Factory Licensing article to apply your license or activate your trial.

Task Factory in SSIS Toolbox

See the Task Factory Connection Managers article for instructions on adding new connection managers in Visual Studio.



Example of Task Factory in the Visual Studio SSIS Toolbox and SSIS Connection Manager available for SSIS packages.

Note: If you do not see these options in Visual Studio, ensure you have installed Task Factory to an environment that meets the system requirements.

- SQL Server Data Tools (SSDT) for Visual Studio must be installed.
- The **New Connection Manager** option for the **Add SSIS Connection Manager** displays in Visual Studio for **Integration Services** type projects. See the SQL Server Integration Services Projects download from Microsoft.

If your Task Factory components aren't displaying in the SSIS toolbox after installation, you may not be targeting a supported version of SSIS. See the **Task Factory Items not showing up** tab in the Common Errors article for more information.

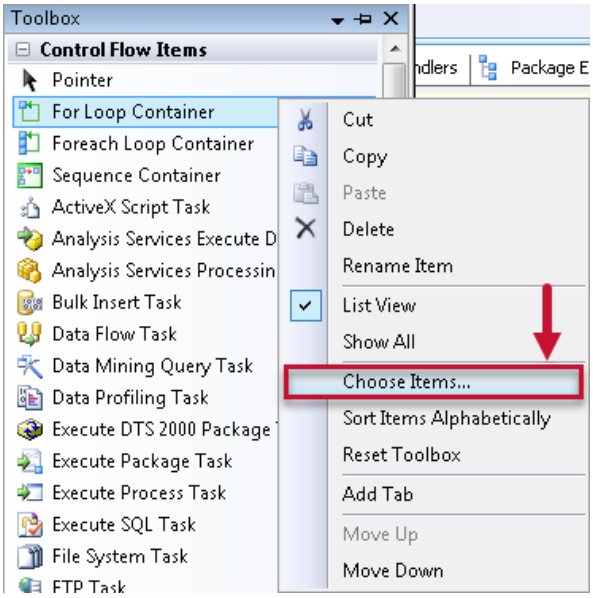
Adding Task Factory components in BIDS/Visual Studio 2010 or lower

Before you can use the Task Factory components in BIDS, you must add them to your toolbox.

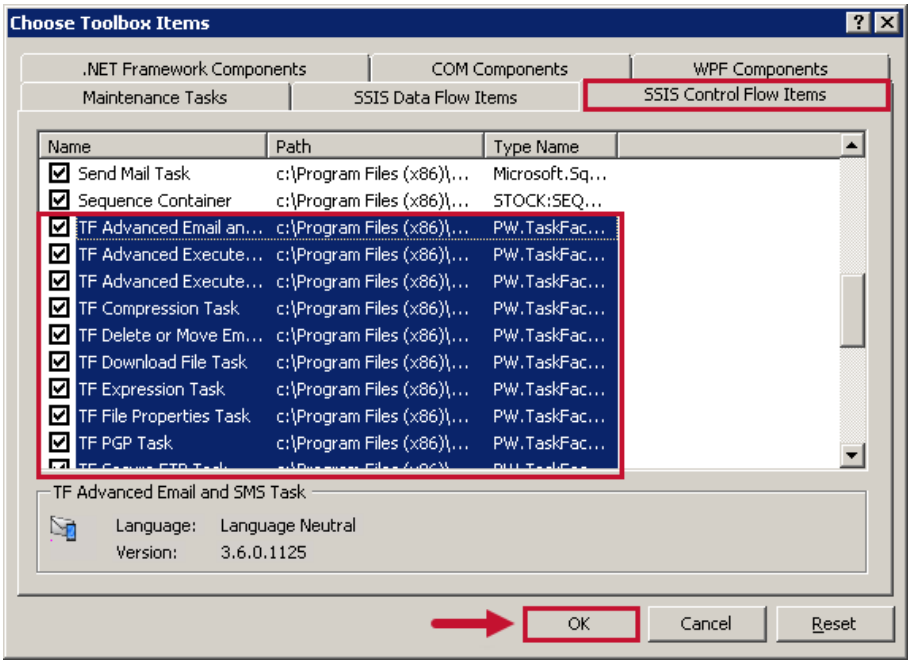
Note: Adding Task Factory components in BIDS is not necessary with SQL Server Data Tools (2012 or higher).

Complete the following steps to add the Task Factory components to your toolbox:

1. Open an SSIS package, then open your toolbox (**View** > **Toolbox**).
2. Right click the toolbox window and select **Choose Items...**



3. Select the **SSIS Control Flow Items** tab and select all deselected tasks that begin with **TF**. Next, follow the same step with the **SSIS Data Flow Items** tab.
4. After you have selected all Task Factory **Control Flow** and **Data Flow** items, select **OK** to save and confirm your selections.

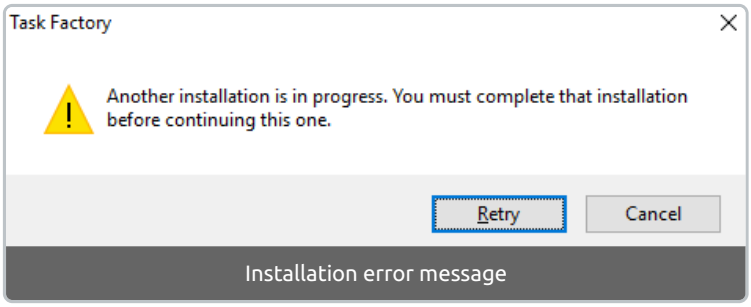


Troubleshooting

Another installation is in progress

If you receive this popup while installing Task Factory:

1. Check **Task Manager** to see if another installation of Task Factory is running. If so, choose **End Task**.
2. Restart the machine.

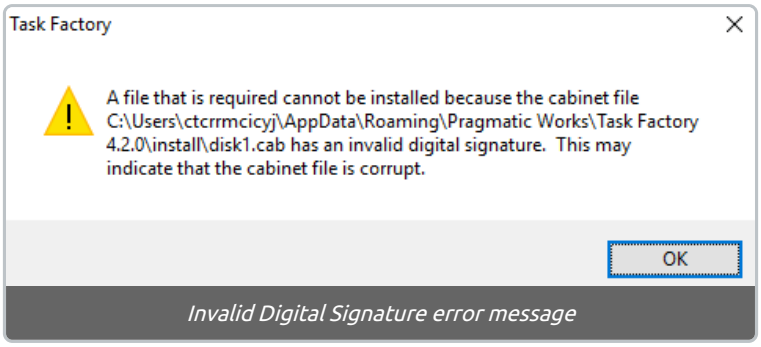


Installation error message

Task Factory Installer Fails To Complete Due To Invalid Digital Signature

See the article above for information about resolving issues related this error message:

A file that is required cannot be installed because the cabinet file C:\Users\...file.cab has an invalid digital signature. This may indicate that the cabinet file is corrupt.



Invalid Digital Signature error message

Creating a Task Factory Installation Log

If you're experiencing other errors while installing Task Factory, see the article above for instructions to create a log file to determine where the installer is failing.

Enabling Task Factory on Azure-SSIS IR in ADF

Last Modified on 08 February 2022

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Introduction

These instructions walk through the steps to create an Azure-SSIS Integration Runtime (IR) and install SolarWinds Task Factory during the start-up process of the Integration Runtime.

If you encounter any issues while installing or activating Task Factory in your Azure SSIS IR, contact our support team at support.sentryone.com.

System Requirements

You need the following prerequisites to set up Task Factory for your Azure Data Factory SSIS Integration Runtime:

- Azure subscription
- Azure SQL Database or Managed Instance Server
- Microsoft Azure Storage ExplorerEdition Restrictions

Notes on Components

The following components are incompatible on an Azure Data Factory SSIS Integration Runtime:

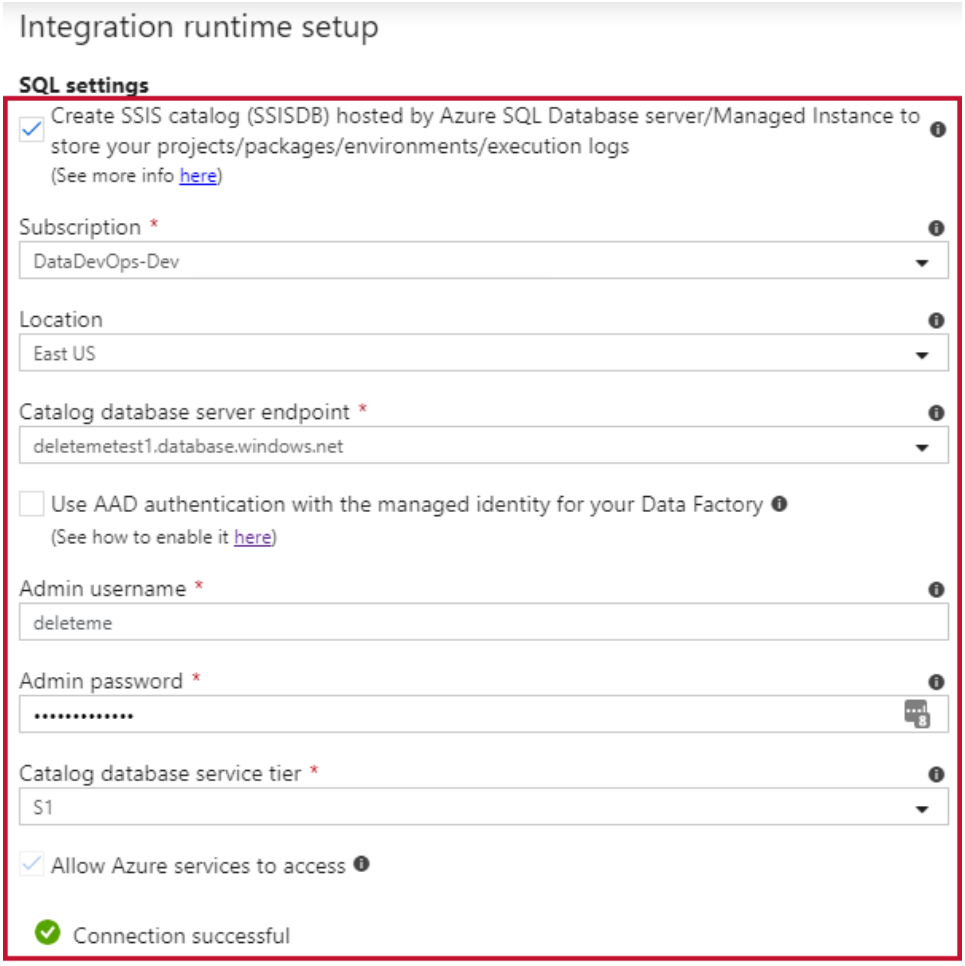
- USPS Address Verification
- Dimension Merge SCD Transform ℹ **Note:** Dimension Merge SCD Transform can be used in Task Factory for Azure on versions 2020.1.3 or higher.
- Task Factory Excel Power Refresh Task
  - ℹ **Note:** You need to install Excel on the Azure Data Factory Integration Runtime to enable this component. For more information, see the Customize setup for Azure SSIS integration runtime MSDN article.

Notes on Trials and Licensing

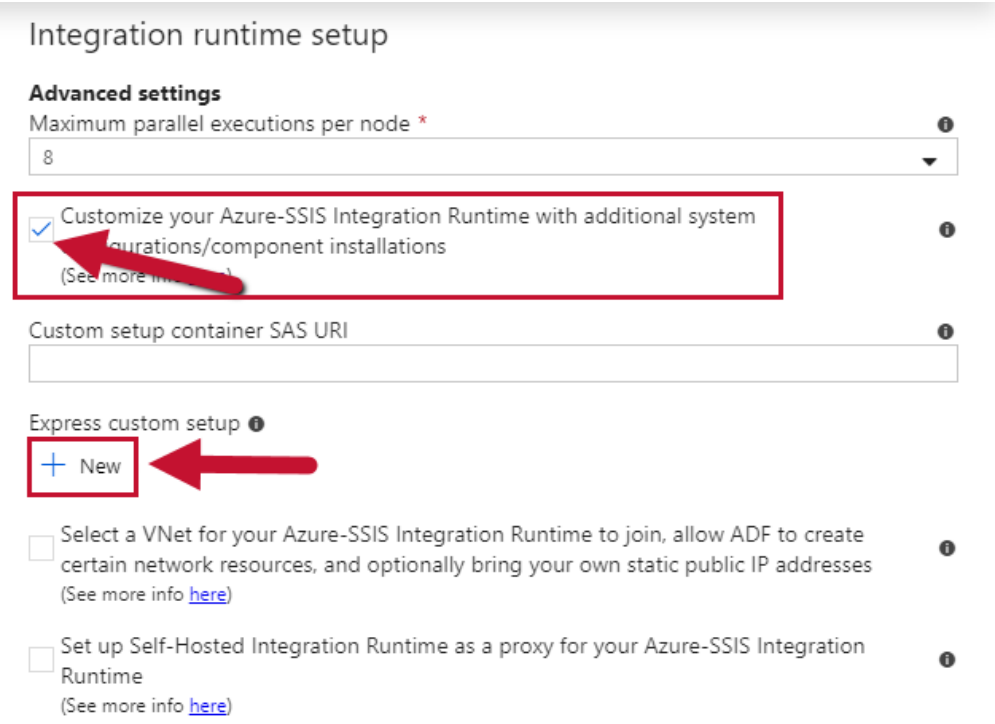
⚠ **Important:** Before activating your Task Factory license, you need to obtain an ADF-ready license from the SolarWinds sales department. Contact your sales represenative for more information. Select the license based on the number of nodes in your SSIS Integration Runtime.

Configuring Task Factory on Azure-SSIS Integration Runtime

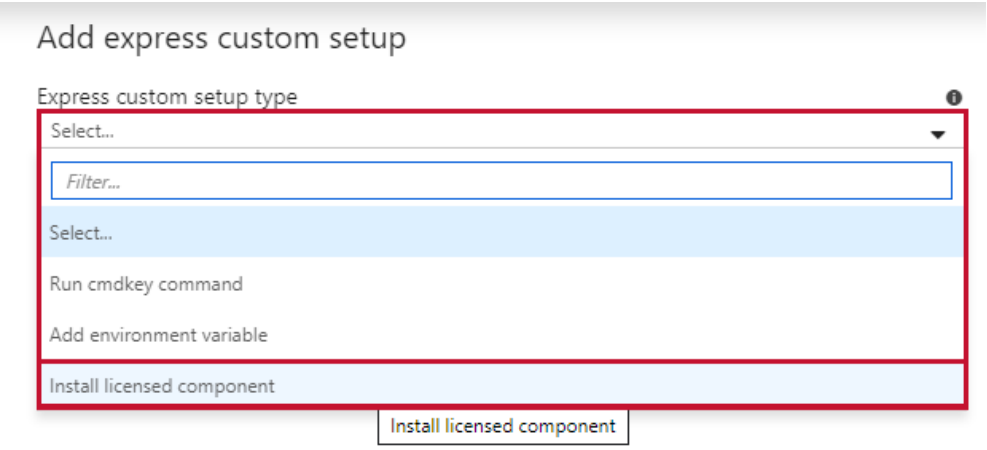
1. Configure your settings on the Initial **Integration runtime setup** screen, and then continue to **Advanced settings**.



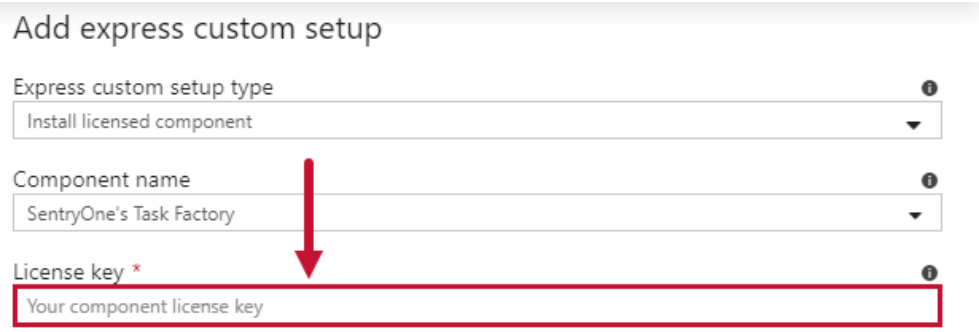
2. Select the **Customize your Azure-SSIS Integration Runtime with additional system configurations/components installations** checkbox, and then select the **+New** link under **Express custom setup** to open the Add express custom setup window.



3. Select **Install licensed component** from the drop-down list, and then select **Task Factory**.



4. Paste your Task Factory License key into the **License Key** box to complete the configuration.



🏆 **Success:** Every time you start your Integration Runtime, each node will have Task Factory automatically installed and activated without any further steps from you.

Creating Your Data Factory

Additional Information: Use the SAS URL you created as you go through the Provision the Azure-SSIS Integration Runtime in Azure Data Factory MSDN tutorial. The specific step you use the URL on is **Step Four (Advanced Settings)**.

Note: The tutorial doesn't mention that you need an Azure SQL Server instance that doesn't have an SSISDB database on it. If you have an SSISDB database on the SQL Server you provide during the process of creating your Azure IR , the IR creation process fails.

Every time you want to create an SSIS IR, you need to delete the current SSISDB or create a new server without an SSISDB database.

Follow the tutorial to set up your Data Factory and Azure SSIS IR instance.

After you have created the Data Factory, log directories are created in your blob storage container. Log files are located under the **main.cmd.log** directory. If you had **Storage Explorer** open already, you might need to refresh the window.

Review the log files for any errors. If you don't see any errors, you're good to go.

Note: You shouldn't need to review the Task Factory installation log unless you run a package with Task Factory and receive a specific Task Factory failure message.

Deploying Packages to Azure-SSIS IR

Now that your Azure IR has been created and Task Factory is installed, you should be able to take any package created locally in SSDT, deploy it to Azure IR, and execute it successfully.

See the **Deploying Packages with Task Factory Components within an Azure-SSIS IR** section of the Easily Install Task Factory on an Azure-SSIS IR in Azure Data Factory blog post for instructions.

Additional Information: To deploy your packages to your Azure IR, follow the Deploy and run a SQL Server Integration Services (SSIS) package in Azure MSDN tutorial.

Note: You must have the latest version of SQL Server Management Studio 17.2 or above installed to

follow the tutorial and view the Integration Service Catalog.

Executing Packages within an Azure-SSIS IR

Now that Task Factory is installed and your packages are deployed to the Azure-SSIS IR, you can execute those packages.

See the **Executing Packages with Task Factory Components within an Azure-SSIS IR** section of the Easily Install Task Factory on an Azure-SSIS IR in Azure Data Factory blog post for instructions.

Upgrading Task Factory

Last Modified on 05 January 2022

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Introduction & Notes

Upgrade to the latest version of Task Factory to access the new and updated components released by SolarWinds. For more information about what the newer versions have to offer and the latest version number, see the Release Notes article.

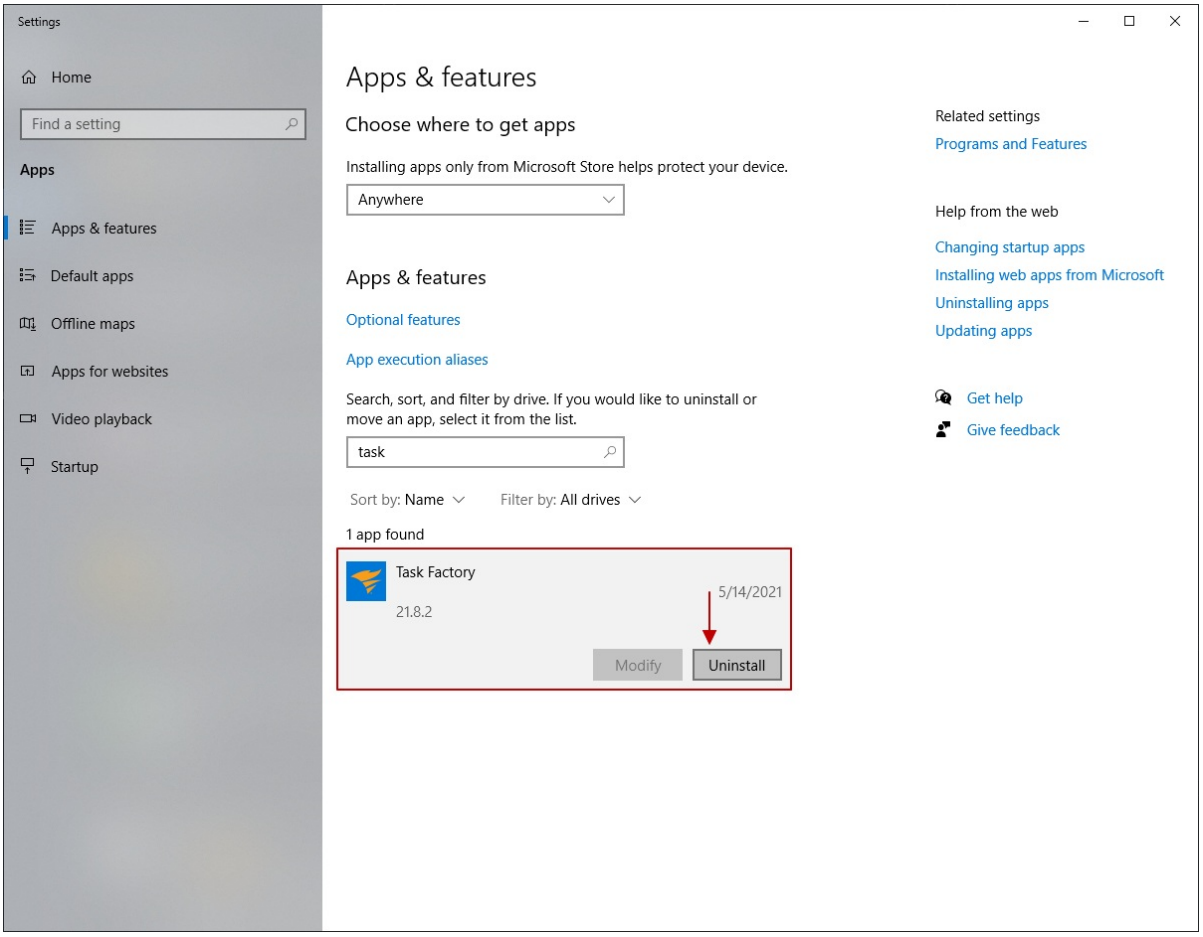
⚠ **Important:**

- Before upgrading your current version of Task Factory, SolarWinds recommends creating a backup of your packages to ensure that you can revert back to a previous version if any issues occur.
- It's recommended that you test upgrades before rolling them out to your production environment. See the Testing the Task Factory Upgrade article for instructions.
- Close any Visual Studio windows before starting an upgrade.

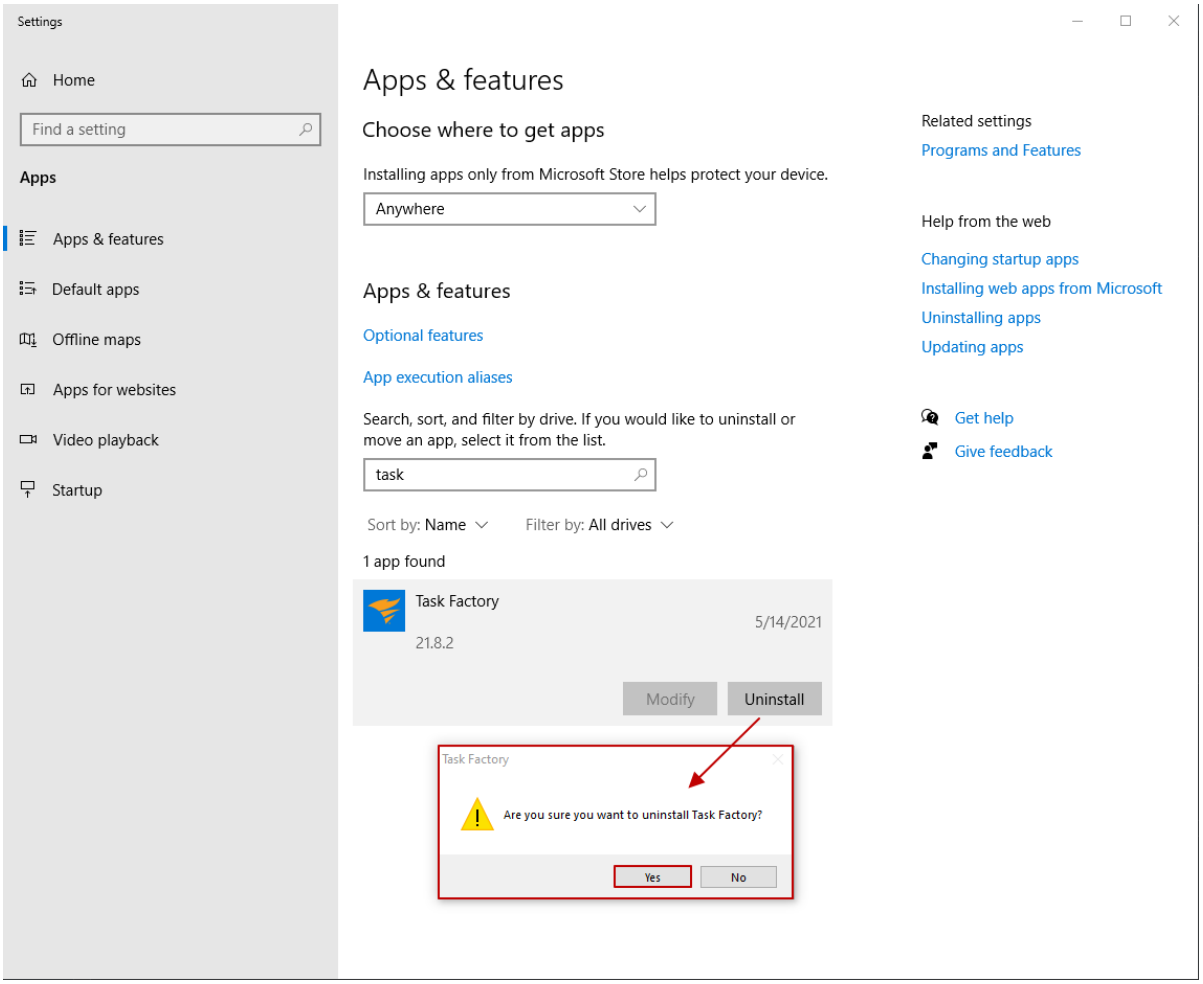
Instructions

1. Uninstall Task Factory

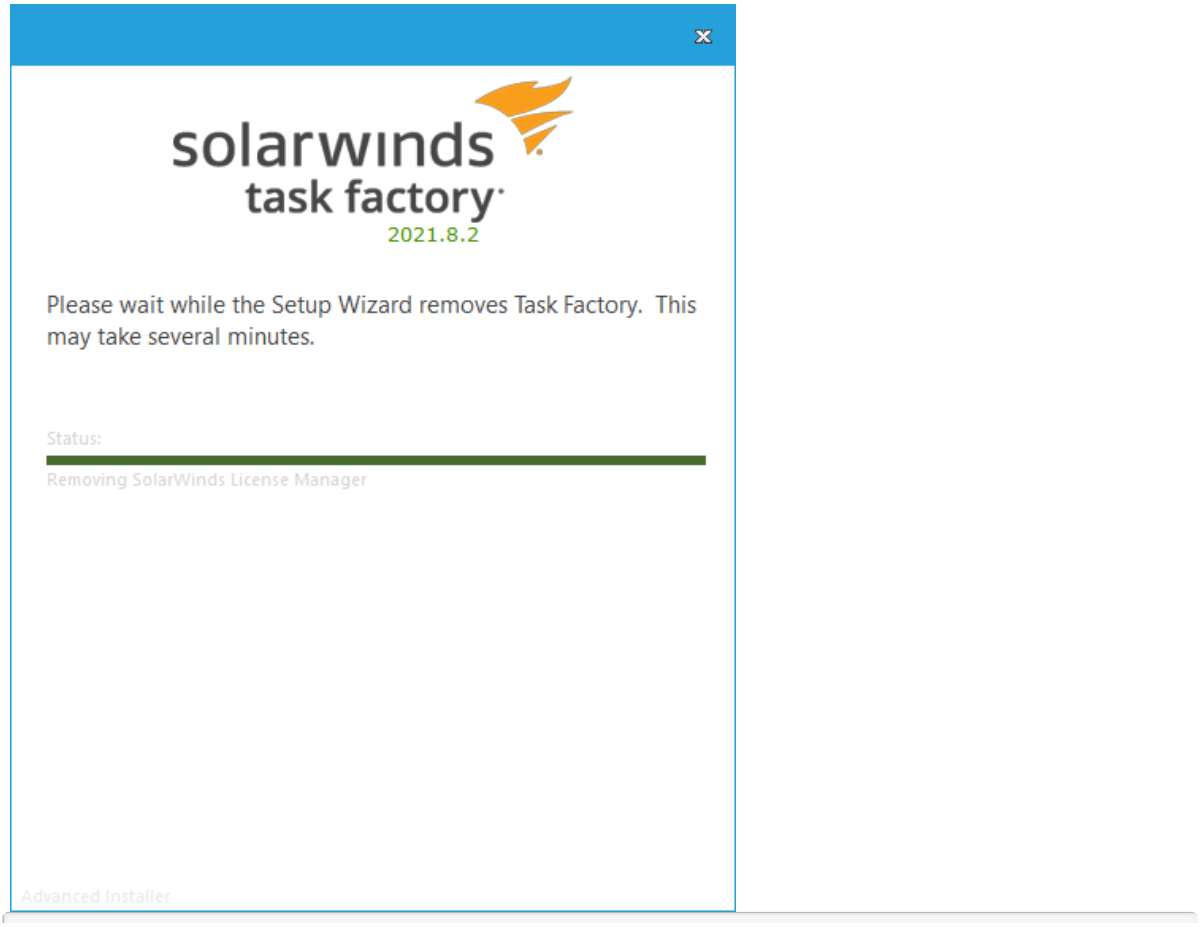
1. Uninstall your current version of Task Factory. Open the **Control Panel** > **Programs** > **Programs and Features**, select **Task Factory**, then select **Uninstall.**



2. Select **Yes** to start the uninstall process.



3. Wait for the **Setup Wizard** to complete the removal process.



> **Note:** The **SolarWinds License Manager** will also be removed during the uninstall process. If you have additional SolarWinds products and need to manage licenses, see the SolarWinds License Manager page to get a standalone copy, if needed.

2. Download Latest Version of Task Factory

Log into the SolarWinds Customer Portal to download the latest version of Task Factory.

> **Note:** Only license administrators can get the latest download from the customer portal above.

3. Install Task Factory

Complete the Installation process for Task Factory to begin using new components.

Troubleshooting

How do I know which version of Task Factory is installed?

Most Task Factory components display the version number next to the name of the component when you are viewing the component editor UI in Visual Studio. For additional ways to find the version number, see the How to Confirm Task Factory Version article.

Manual Deletion of Assemblies after Uninstalling Task Factory

If any assembly files remain after uninstalling Task Factory, they may lead to problems later on when you attempt to run a package. Many different errors can be thrown in this situation. See the Manual Deletion of Assemblies after Uninstall of Task Factory article for all the file locations and instructions to fully remove them.

Task Factory Connection Managers

Last Modified on 13 January 2022

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Introduction

Once Task Factory is installed, you have access to new **Connection Managers**. The **Connection Manager** component stores the settings for a connection that's used within your package and creates the connection during the package runtime. They store your **ConnectionString** to your component and can be used in multiple packages.

Available Connection Managers

The following **Connection Managers** are available in Task Factory:

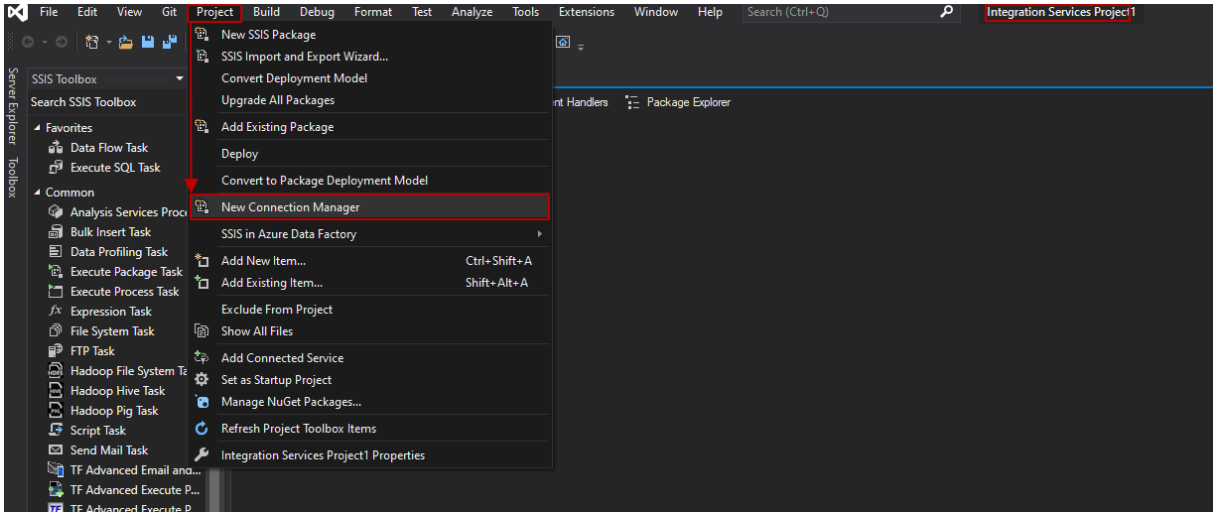
- Advanced Cache Lookup
- Azure Storage
- Dynamics CRM
- Dynamics OData
- Email
- Excel
- Rest Basic
- Rest OAuth1
- Rest OAuth2
- SalesForce.com
- SecureFTP
- SharePoint
- Social OAuth

Note: If you do not see these options in Visual Studio, refer to the Installing Task Factory article for a troubleshooting checklist.

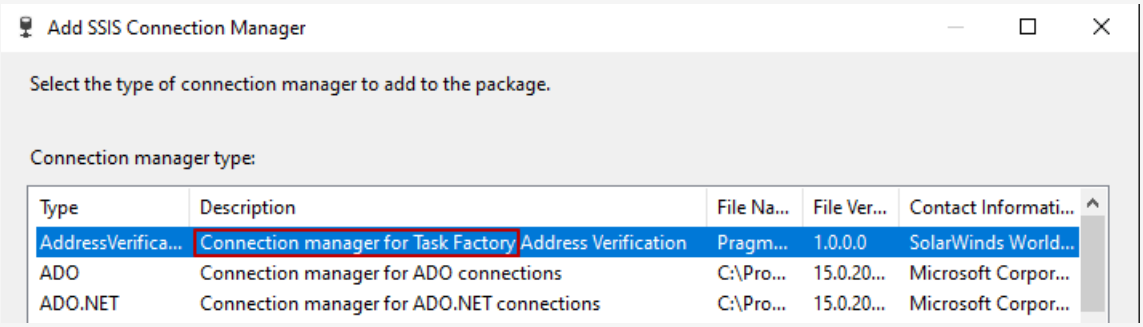
Creating a New Connection Manager

Add a new **Connection Manager** by completing the following process:

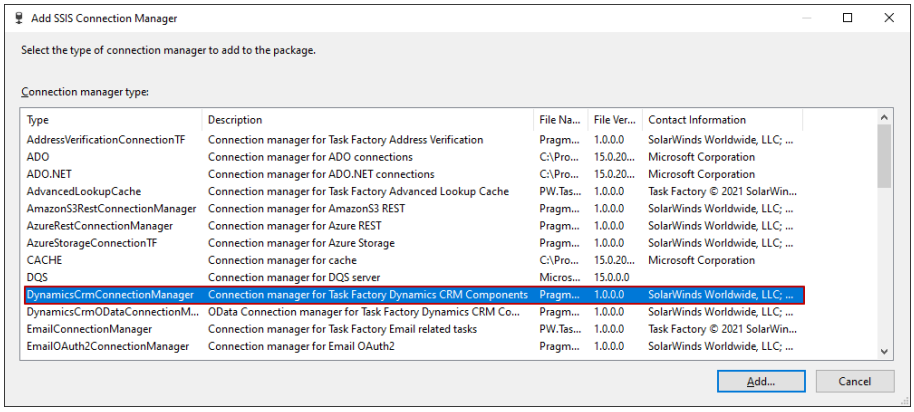
1. Open **SQL Server Data Tools** or **BIDS**. Select **Project** > **New Connection Manager** to open the **Add SSIS Connection Manager** window.



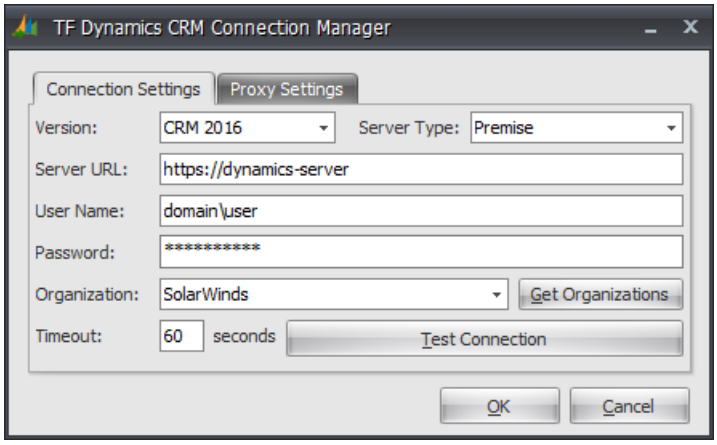
Note: The **Add SSIS Connection Manager** window opens with all available connection managers. A description of **Connection manager for Task Factory** denotes the Task Factory connection managers.



2. Select the desired **Connection Manager**, and then select **Add** to configure that **Connection Manager**.



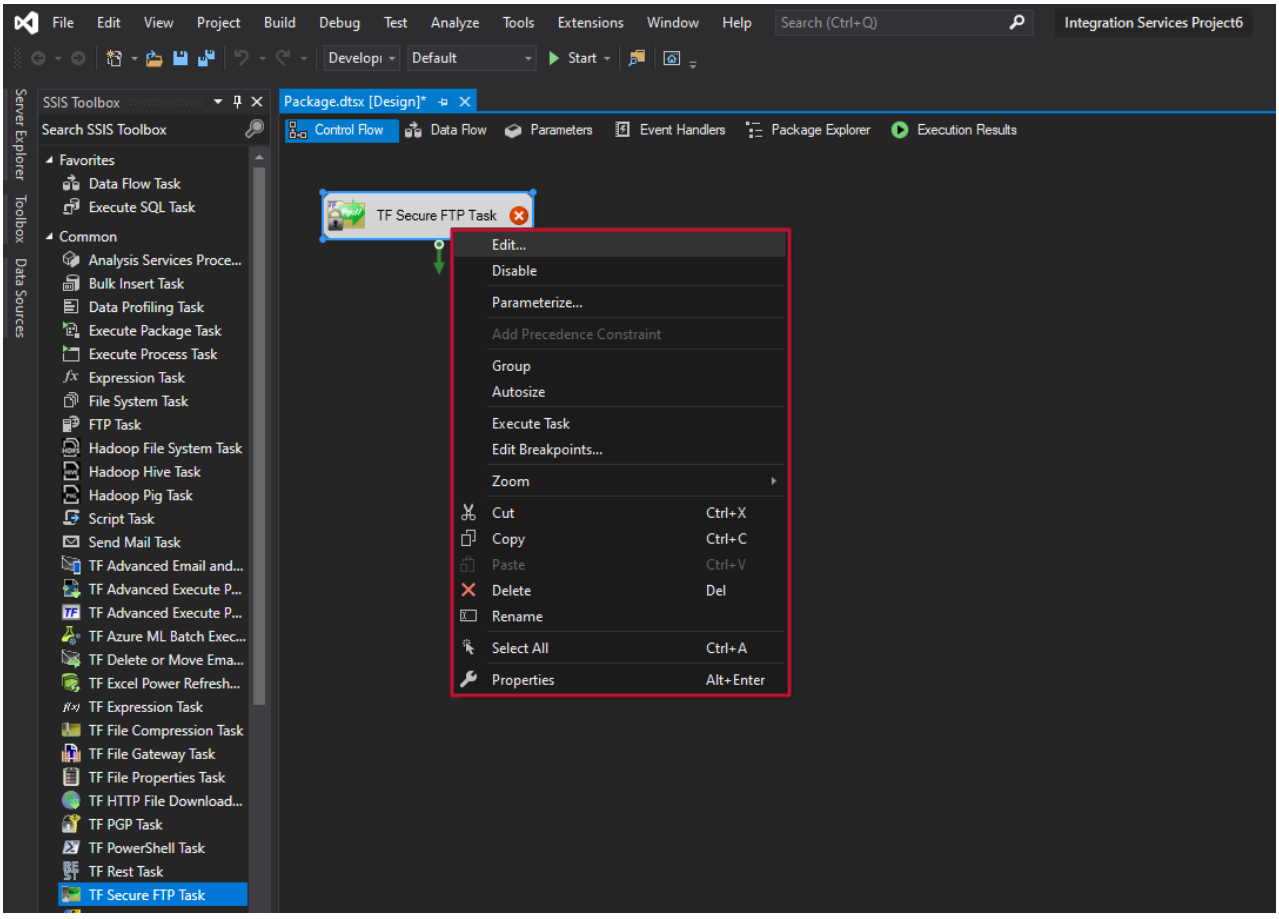
3. Complete the settings and configuration options for the selected **Connection Manager** type (and **Test Connection** if applicable), then select **OK**:



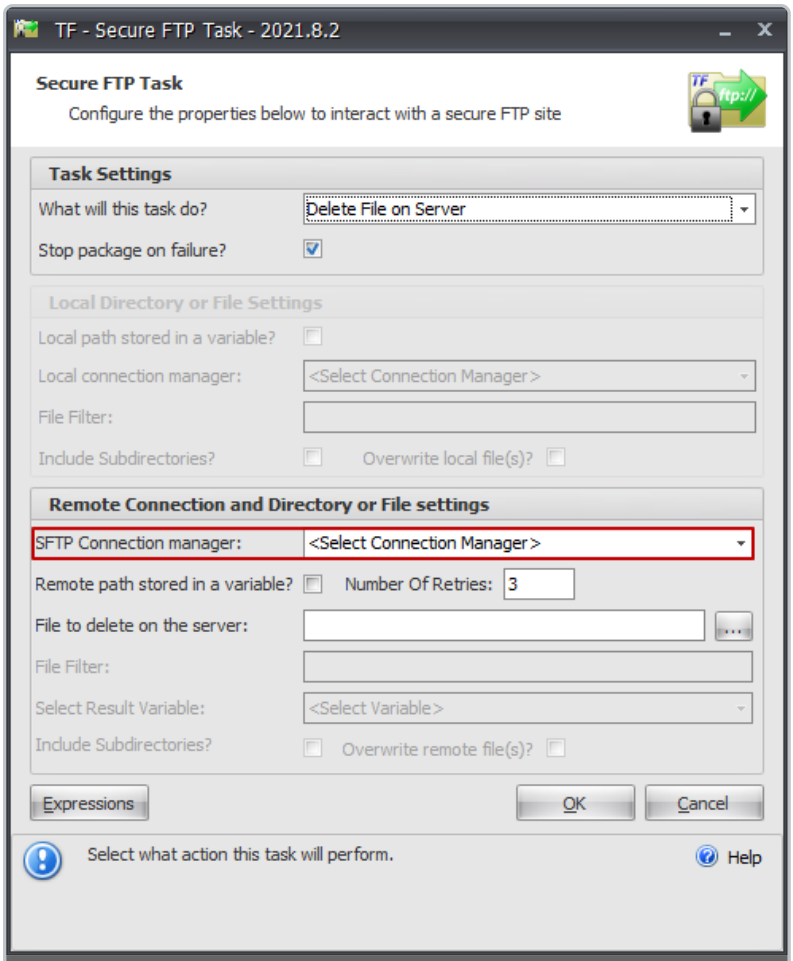
🏆 **Success:** The **Connection Manager** is added. You can now use your **Connection Manager** in tasks.

What's next?

Once a task from the **SSIS Toolbox** is added to your project, you can right-click it to access the context menu options:



For example, select **Edit** to configure the properties of a **Secure FTP Task**. This allows you to set the associated **Connection Manager**:



Select a **Connection Manager** in the task

Note: The drop-down menu also has an option for adding new **Connection Managers**.

Task Factory Licensing

Last Modified on 08 March 2022

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

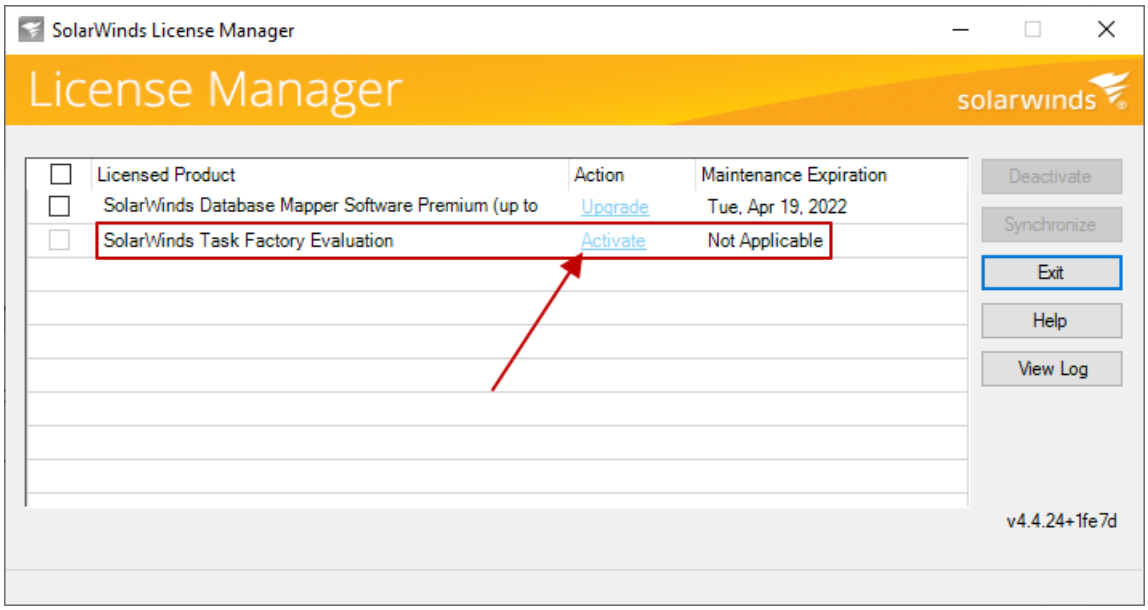
Note: Task Factory can be used in a development environment without a license. A license is required to execute the packages outside of the development environment.

Additional Information: See the Task Factory product page for complete information on licensing options, prices, and models (i.e. which licenses are per server or ADF node).

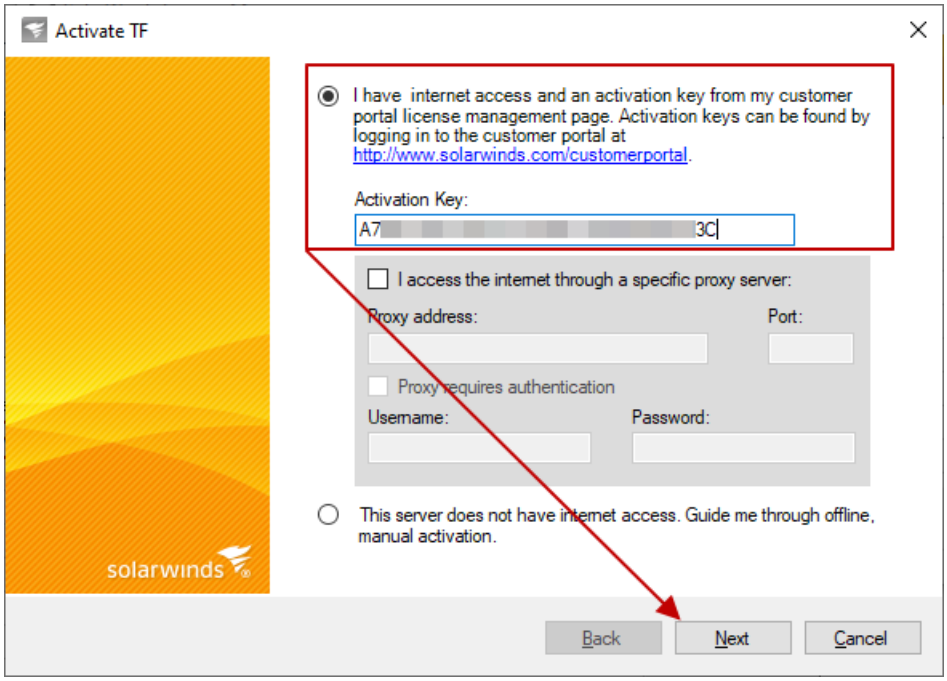
SolarWinds License Manager Online Activation

Your SolarWinds Task Factory license is managed through the SolarWinds License Manager.

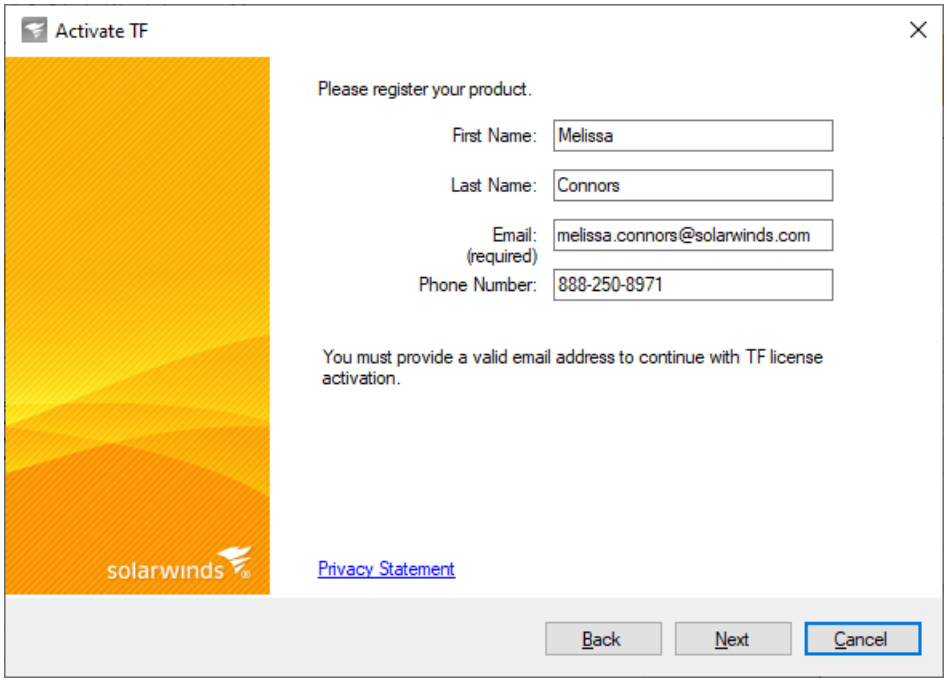
1. Use the **Windows Start** menu to open **SolarWinds License Manager**.
2. Select **Activate** to add your full license. **Note:** Use **Upgrade** if you are updating an existing license, not a trial evaluation.



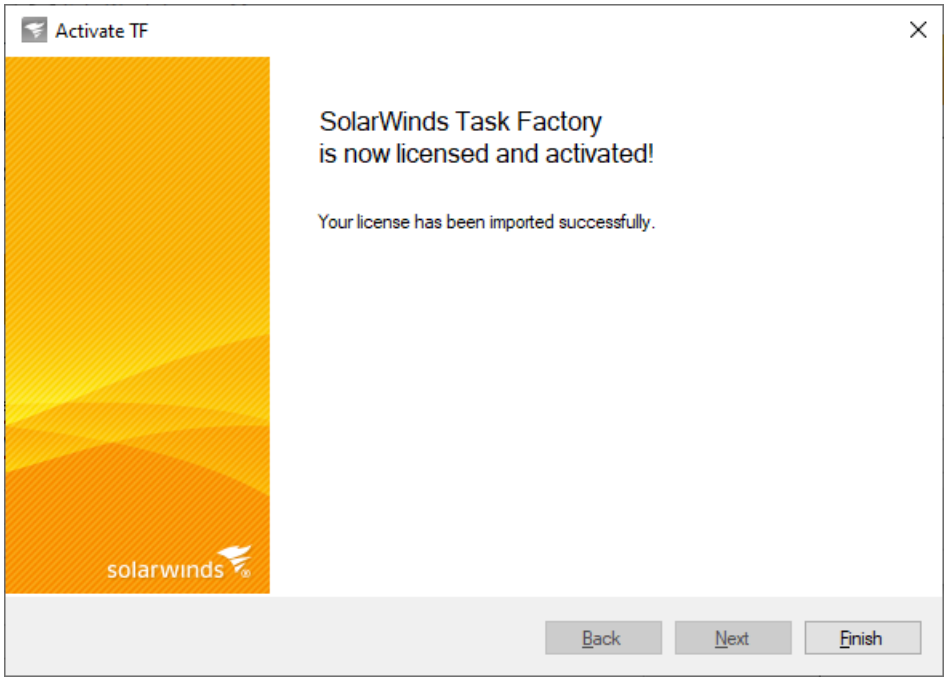
3. Enter the **Activation Key**, then select **Next**. **Note:** There are options to use a proxy server or activate your license offline as well.



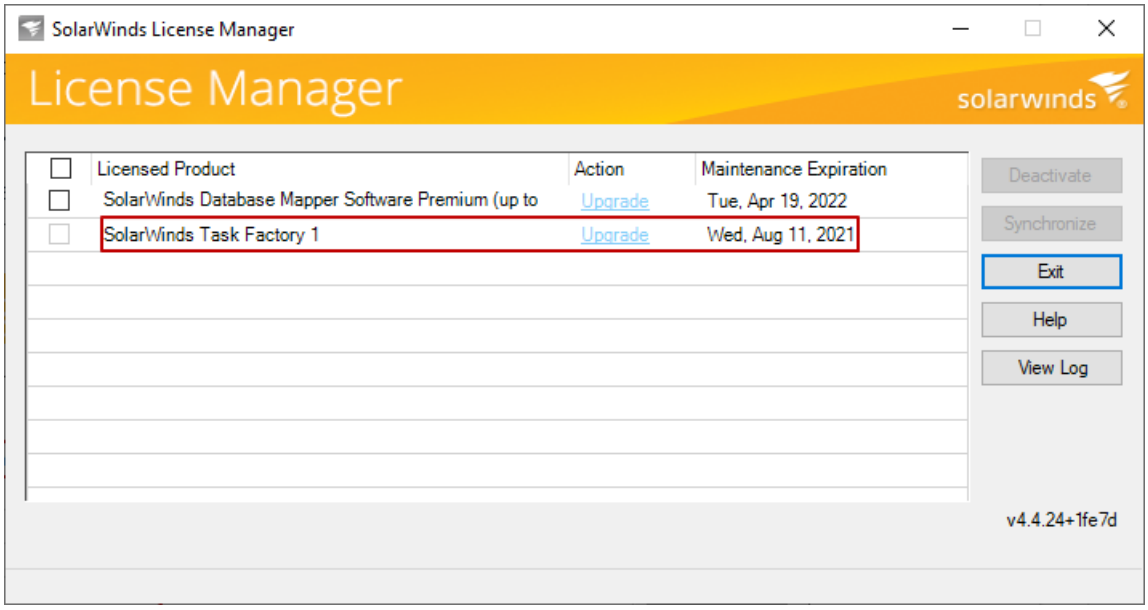
4. You must register Task Factory to continue. Enter your contact information, then select **Next** to continue.



5. Select **Finish** to leave the activation wizard and return to the **SolarWinds License Manager**.



6. This copy of Task Factory is now licensed.



> **Note:** The available **Action** is now **Upgrade**. In this example, there's a *1* after *SolarWinds Task Factory*, indicating that this key activated a license that is good for one server. If you purchase additional licenses, use the **Upgrade** option to apply a new activation key.

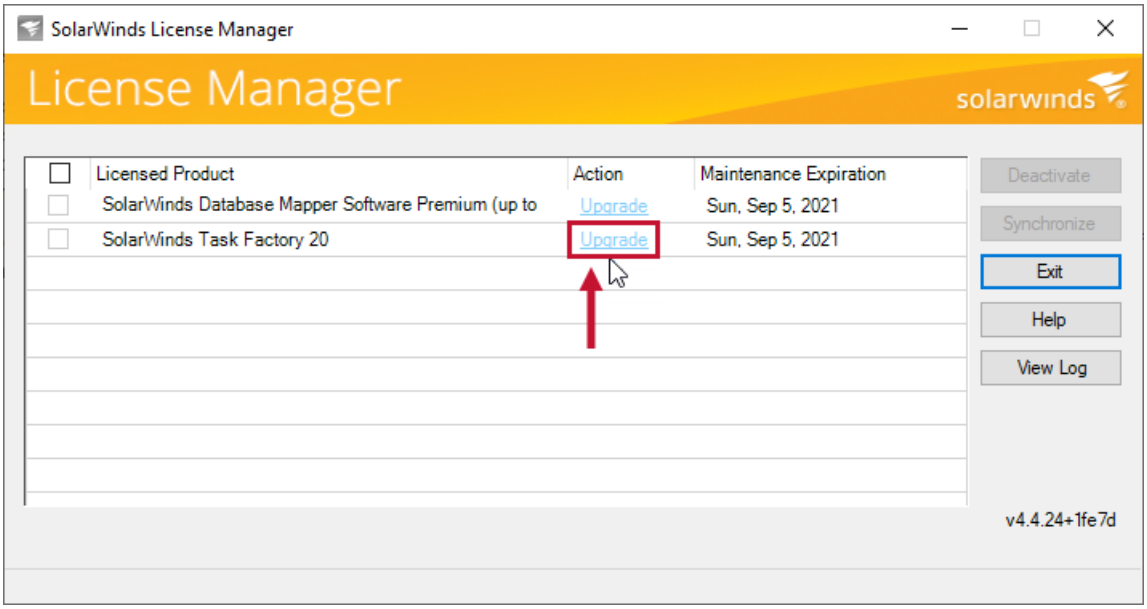
SolarWinds License Manager Offline Activation

Use the **SolarWinds License Manager** to activate a new license key or upgrade your trial license. Activate your license on an offline machine by completing the following steps:

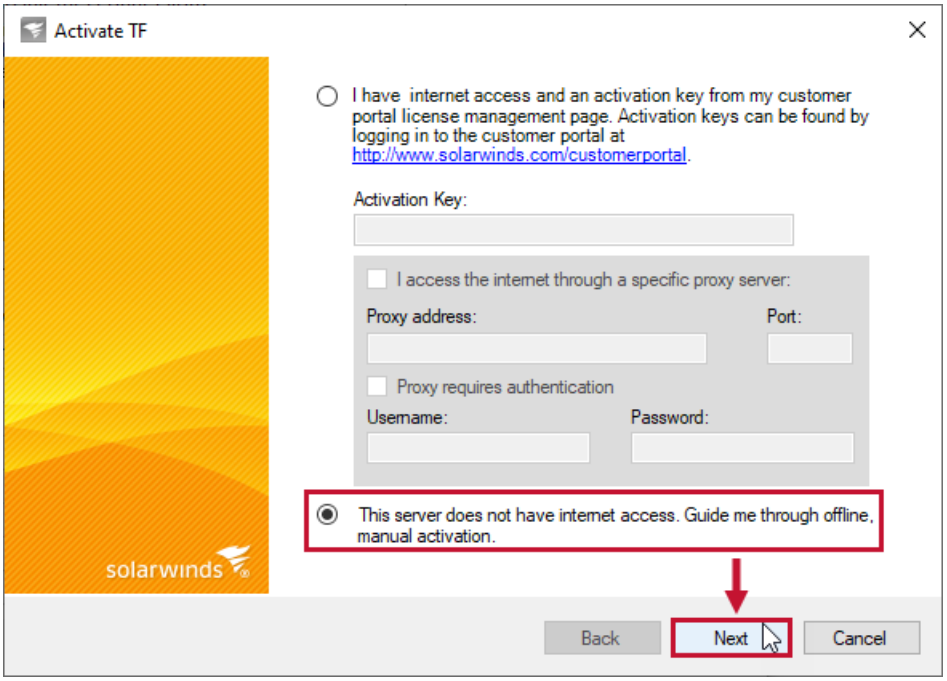
1. Use the **Windows Start** menu to open **SolarWinds License Manager**.

Important: If you have an existing license, the **Action** will have an **Upgrade** option. If you have an evaluation license or just upgraded from a SentryOne licensed version of Task Factory and need to apply a SolarWinds branded license, the **Action** will have an **Activate** option.

2. To apply a license, use the **Upgrade** or **Activate** option.

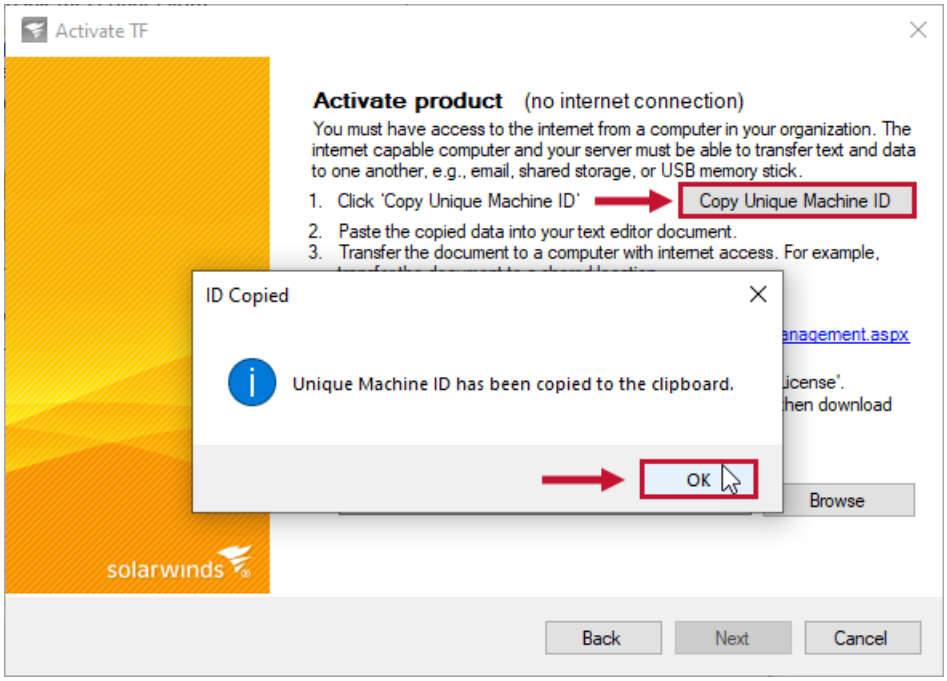


3. Select the **This server does not have internet access option** and then select **Next** to continue.



4. Complete the steps on the Activate Product page:

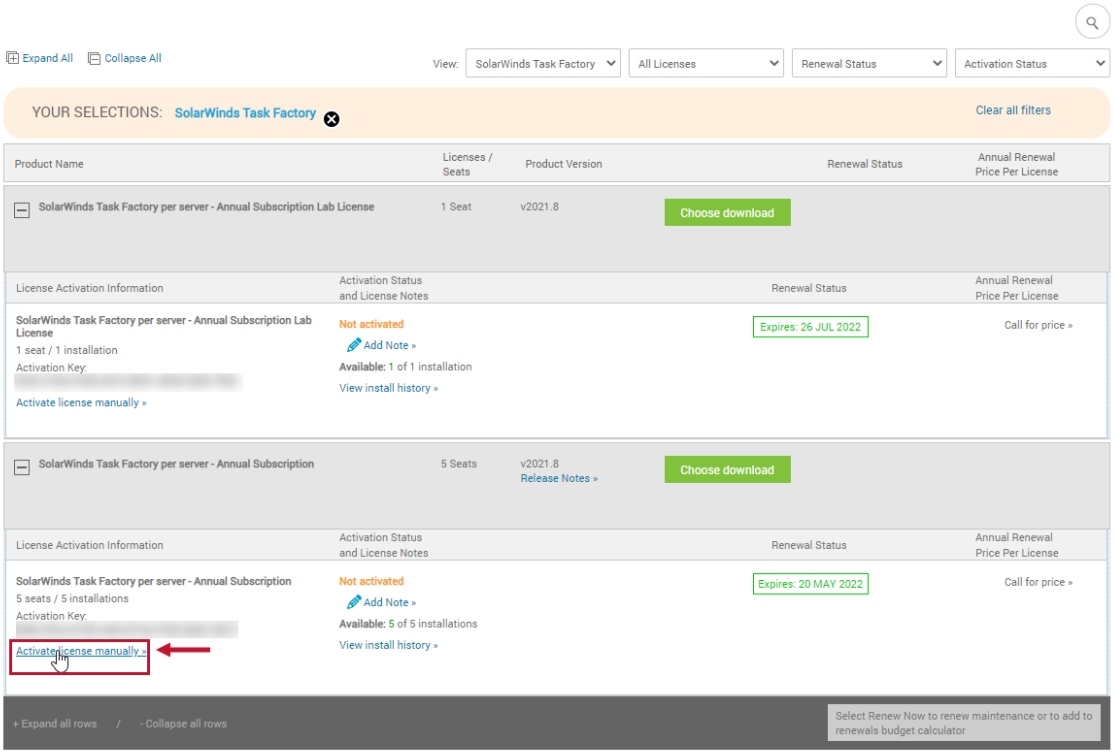
   1. Select **Copy Unique Machine ID** and then paste the id into a text editor. Save the .txt document, and then move the document onto a machine with internet access.



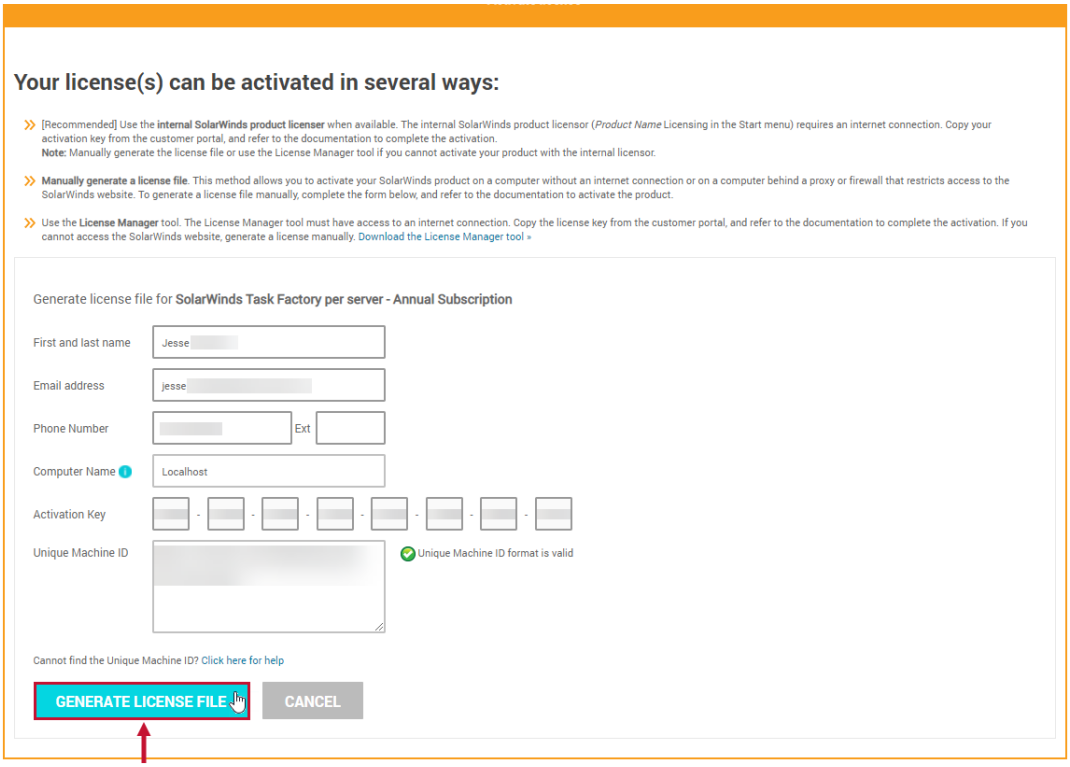
Note: You can transfer the document to a shared location, or copy the document onto a portal drive for transfer.

2. Log into the SolarWinds customer portal, select **Licenses** > **Manage Licenses**, and then navigate to the product you are trying to activate.

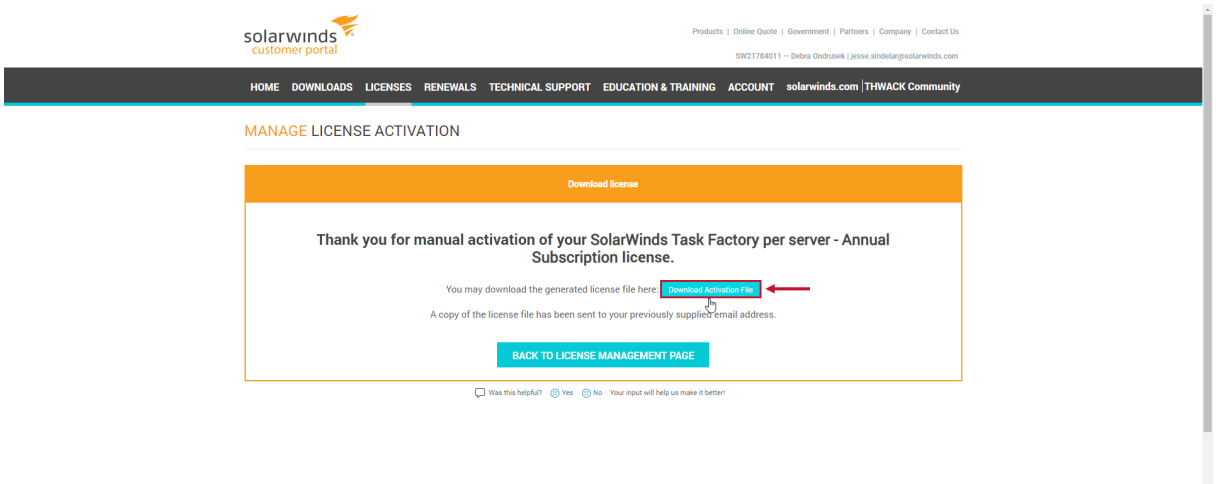
3. Select **Activate license manually** to open the Manage License Activation page.



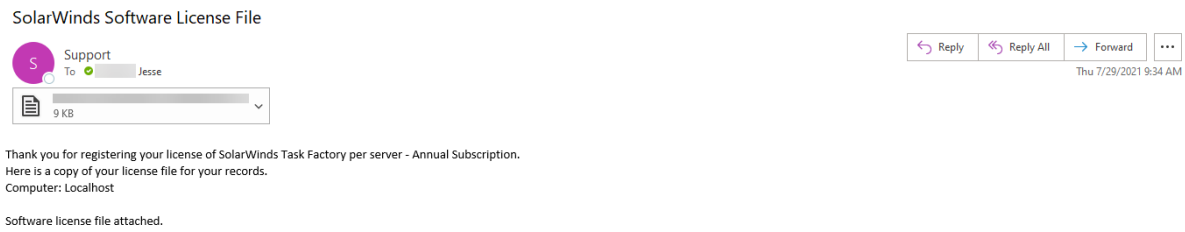
4. Paste the unique machine id from your .txt file into the **Unique Machine ID** field. Enter a computer name for your machine, and then select **Generate License File**.



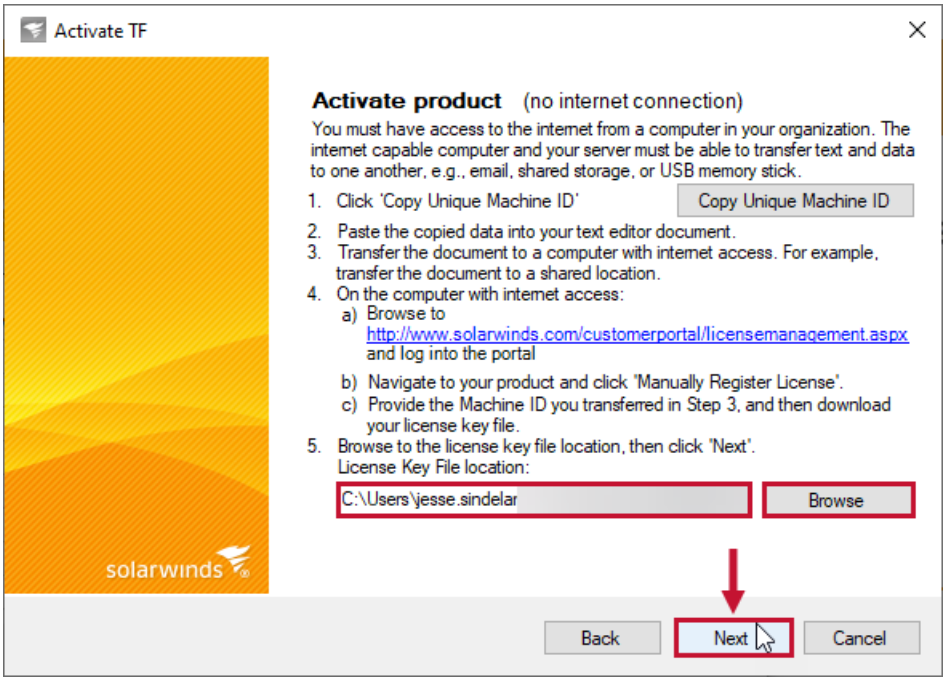
5. Select **Download Activation File**, and then move the file onto your offline machine.



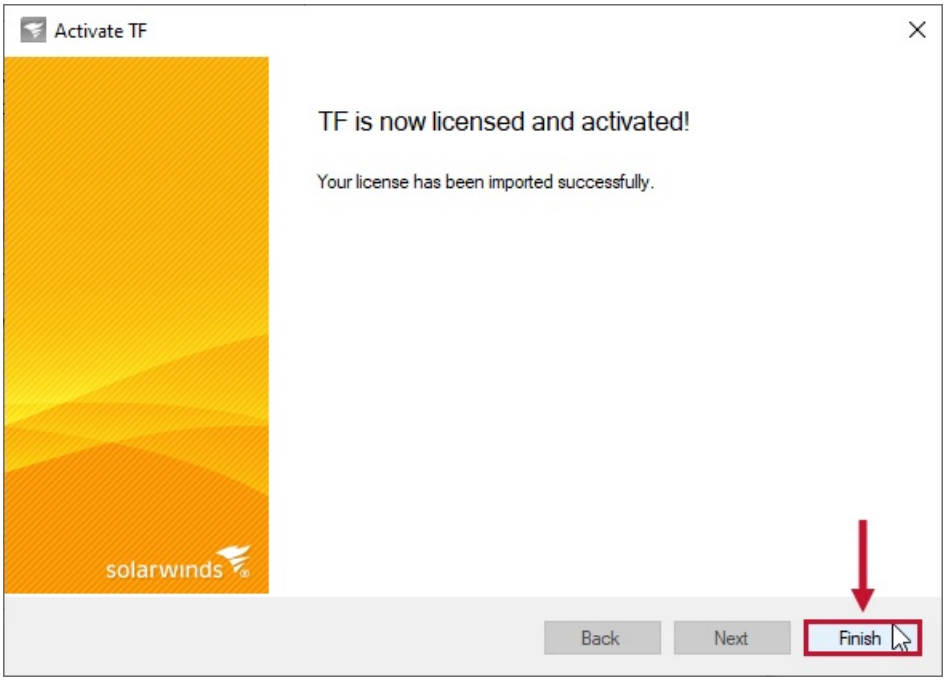
Note: A copy of the download file is emailed to the email associated with your SolarWinds customer account.



5. Select **Browse**, select the location of your license key file, and then select **Next** to continue.

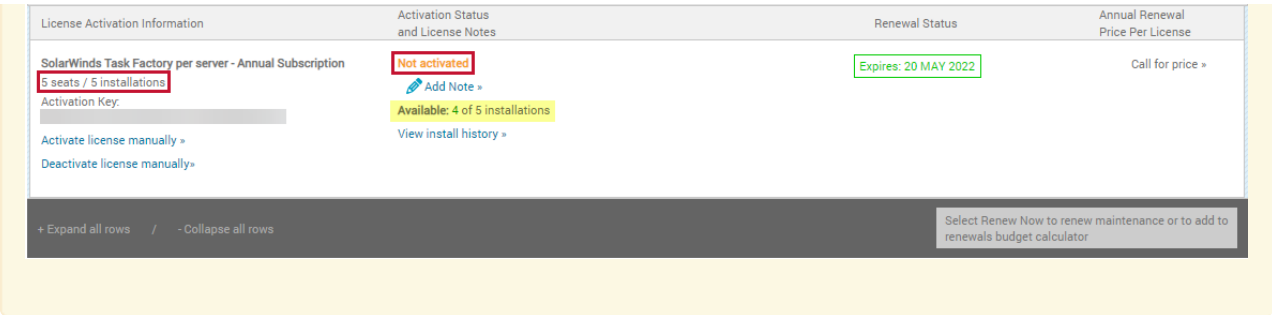


6. Select **Finish** to activate your license.



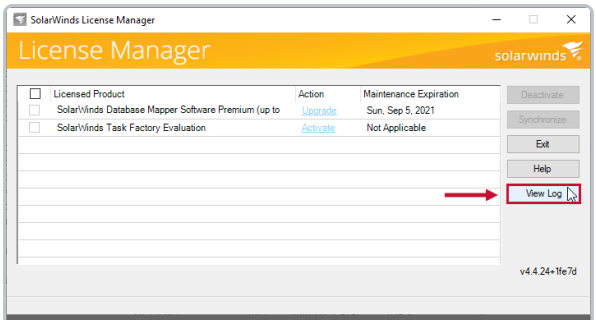
7. Select **Exit** to close the SolarWinds License Manager.

⚠ **Important:** A product that is licensed for multiple installations will display as **Not activated** on the SolarWinds License Management page until all installations have been activated. In the example below, one licensed installation has been activated for Task Factory, and there are 4 out of 5 installations available to be activated.

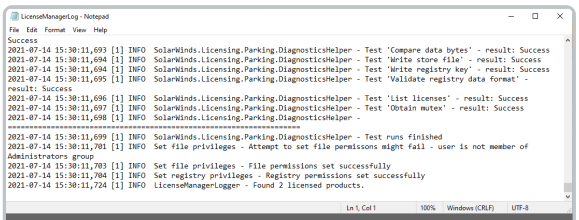


Licensing Errors

Through the course of validating your license in Task Factory, you can run into licensing errors. This section provides an overview for these common errors and ways to troubleshoot them. If you have a licensing issue, it's important to view your log file for more insight into the issue. Select **View Log** to open the LicenseManagerLog.

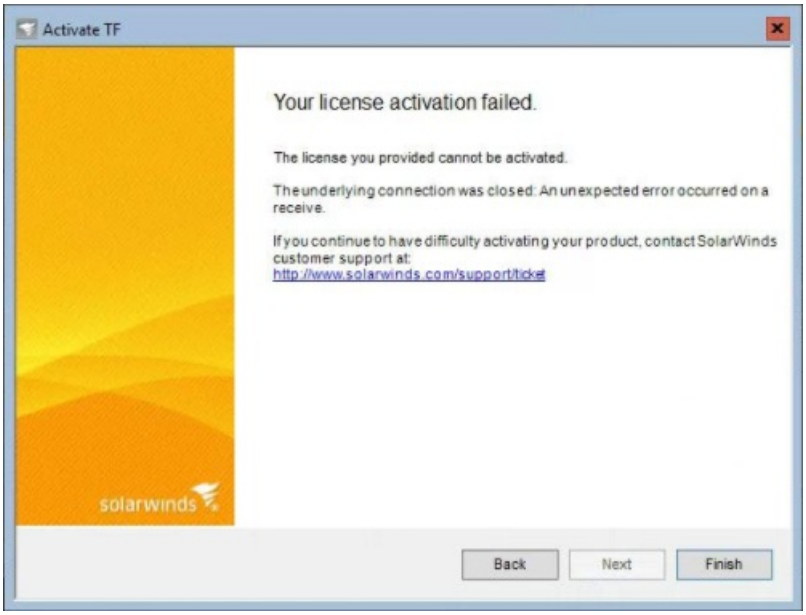


SolarWinds License Manager View Log



LicenseManagerLog file

Online License Activation Failure TLS Version Mismatch

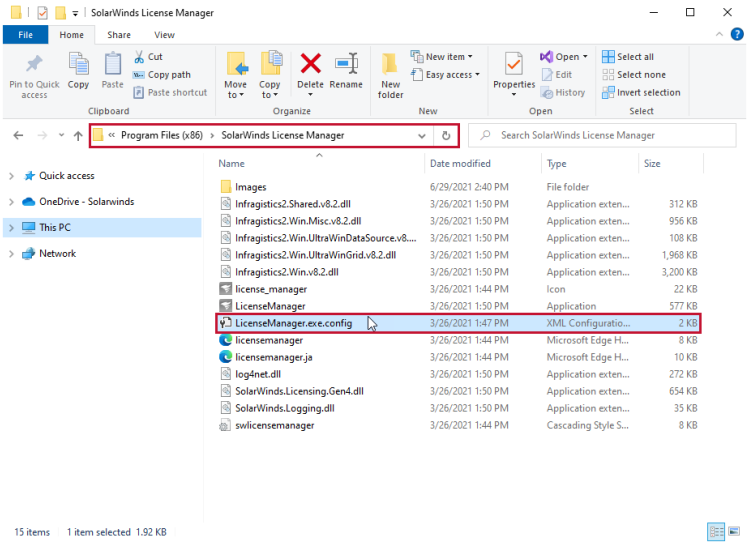


If your log file reports that the client and server can't communicate because they don't have a common algorithm, it's possible that your license manager has a TLS version mismatch. You can troubleshoot this error by completing the following steps:

1. Close the SolarWinds License Manager.

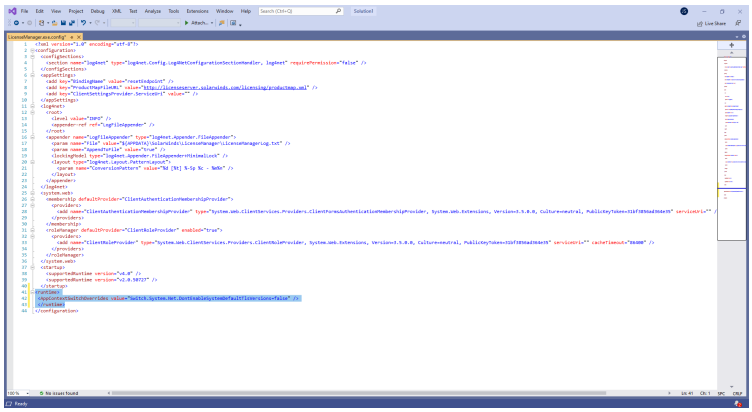
2. Open the SolarWinds License Manager program files at **C:\Program Files (x86)\SolarWinds License Manager**.

3. Open the **LicenseManager.exe.config** configuration file.



4. Add the following code to the .config file before the tag:

```
Switch.System.Net.DontEnableSystemDefaultTlsVersions=false" />
```



5. Save the changes to the .config file, and re-enter your license into the SolarWinds License Manager.

Deprecated License Management

> **Deprecated**: This section applies only to Task Factory versions prior to 2021.8.

SentryOne Task Factory License Manager

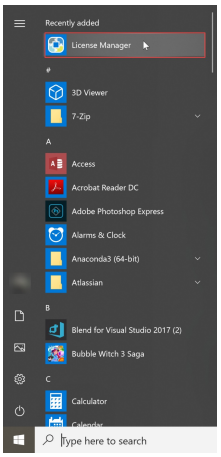
Updating Your SentryOne Task Factory License

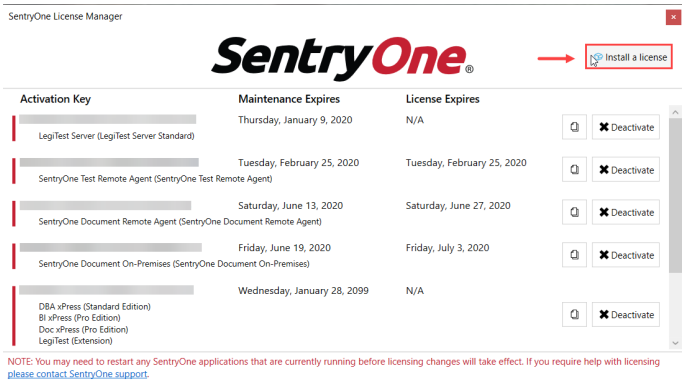
Older versions of Task Factory have different paths to the license manager.

Current path:

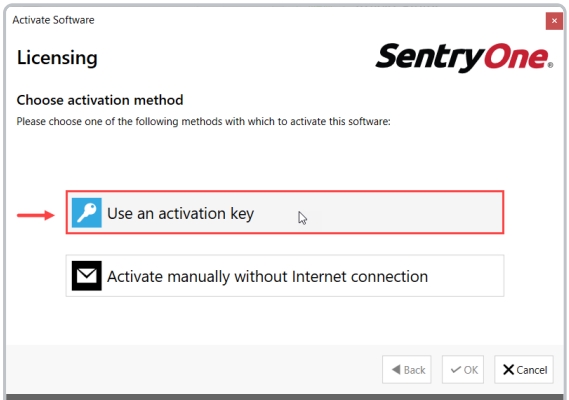
> %Program Files (x86)%\SentryOne\Task Factory\PragmaticWorks.LicenseManager.exe

1. Open the **License Manager**, and then select **Install a License**.





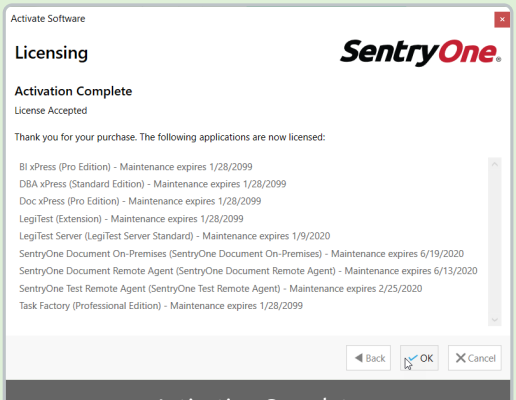
2. Select **Use an activation key**, enter the license key in the **Activation Key** text box, and then select **Activate**.



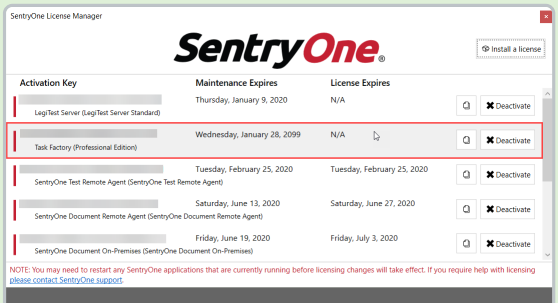
Use an activation key

Activate License

Your license has been activated.



Activation Complete



Active License

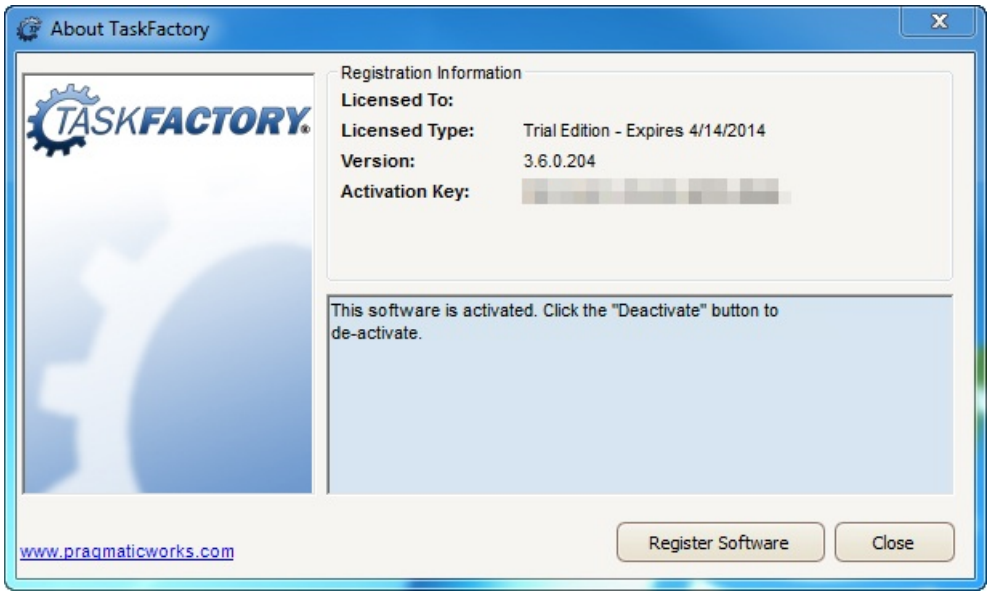
Version 2016.2.1 path:

> %ProgramFiles (x86)%\Pragmatic Works\Task Factory\LicenseManager\PragmaticWorks.LicenseManager.exe

Version 3.2 path:

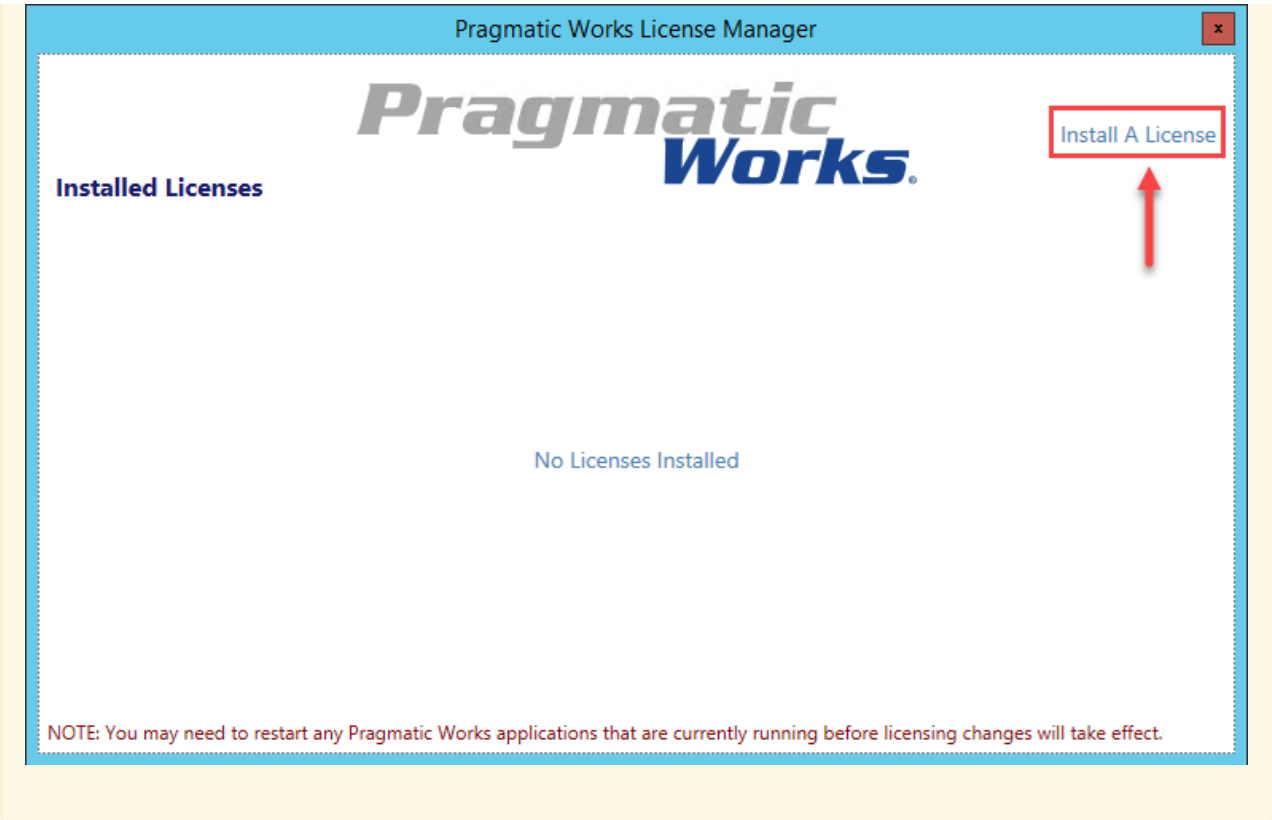
> %ProgramFiles%\Pragmatic Works\Task Factory\TaskFactoryRegistration.exe

1. Before installing Task Factory 3.2, run the Task Factory Registration application (located at %ProgramFiles%\Pragmatic Works\Task Factory\TaskFactoryRegistration.exe), and copy the Activation Key.

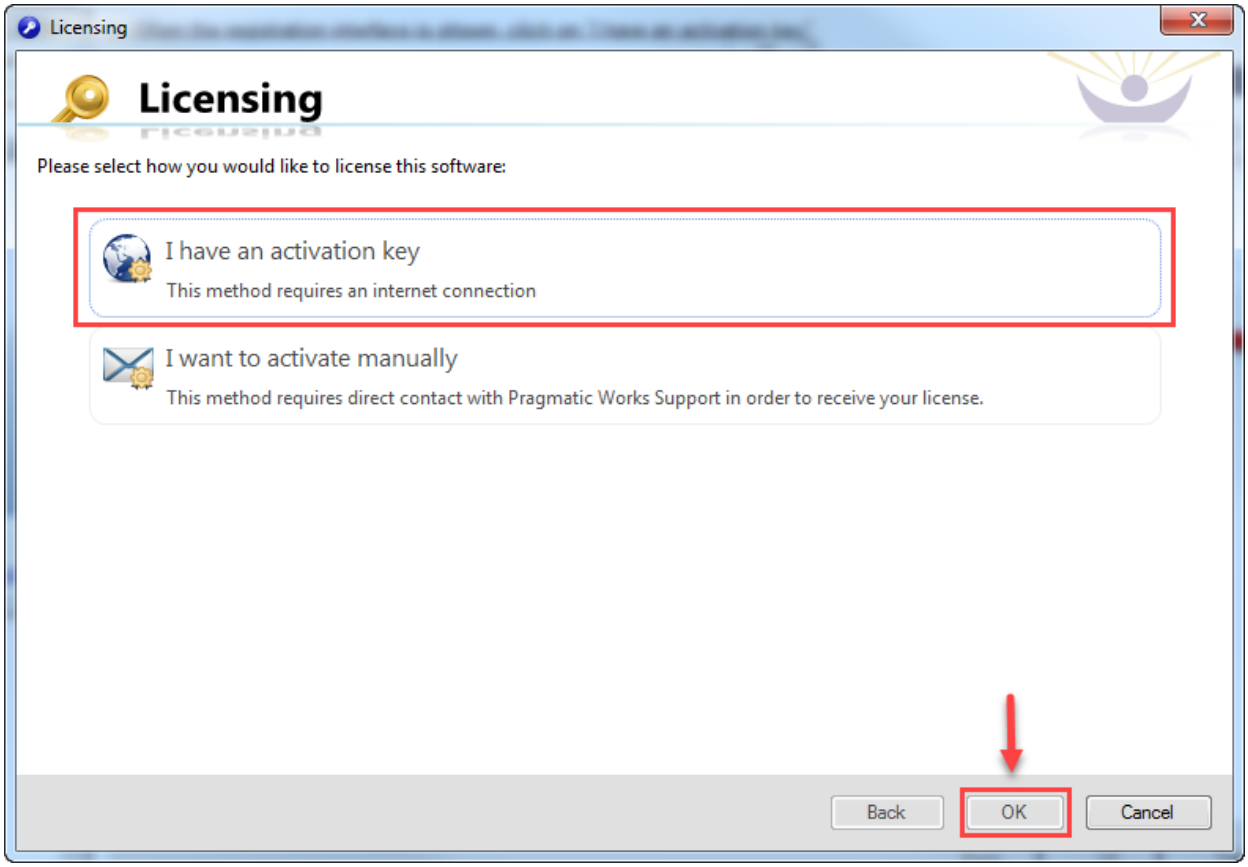


2. After the activation key is copied, uninstall the current version of Task Factory, and then install Version 3.2 (or higher).

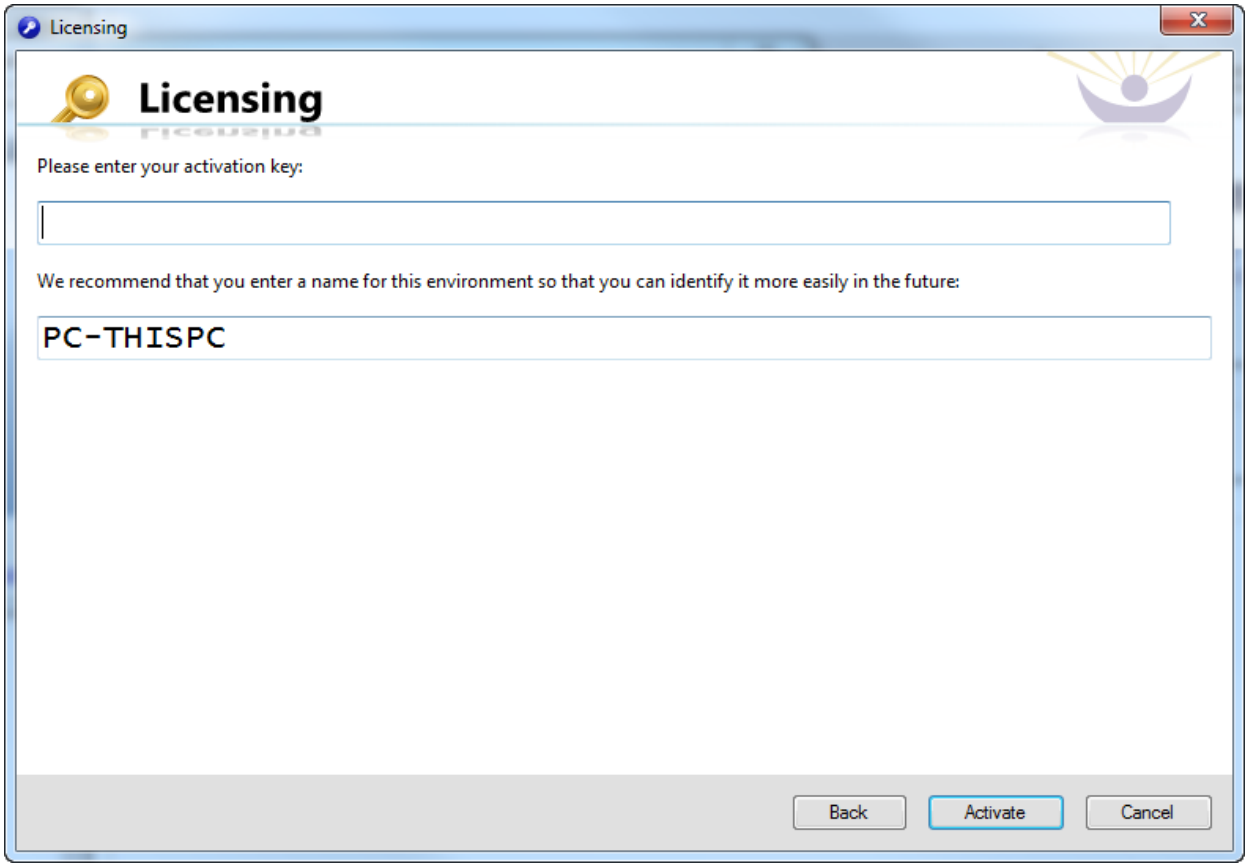
⚠ **Important:** For current versions (2016.2.1 or later), run the new Pragmatic Works License Manager application again (current release is located at %Program Files (x86)%\SentryOne\Task Factory\PragmaticWorks.LicenseManager.exe) then select **Install A License**. All previous versions need to use the same registration tool found in step one.



3. When the **Licensing** window opens, select **I have an activation key,** and then select **OK** to continue.



4. Paste or enter the activation key that was copied, and then select **Activate** to install your license.



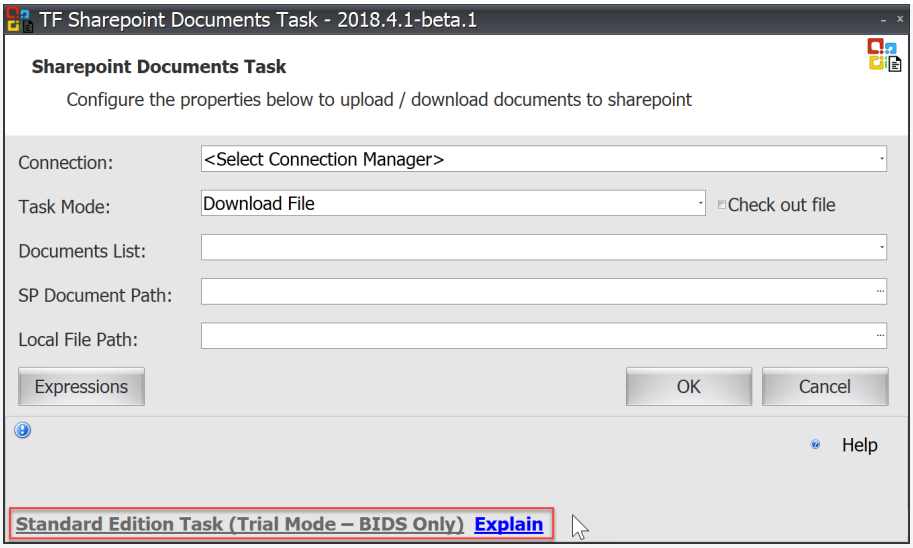
License Activation & Communication Ports

You may need to open a port (or add an exception) on a secure server to allow communication with our licensing servers. This requires the standard **HTTPS** port 443 and the activation URL for the exception is activation.sentryone.com.

Task Factory Licensing Changes

Since the release of version 3.2 of Task Factory we use a licensing technology that requires every license to be updated. This means that if you have a current license, you need to run the **Task Factory Registration tool** with your current license key (if you have one) and update your license.

> **Note:** If your installation of Task Factory shows the (BIDS Only) message at the bottom then your machine does not have a license for the product, and can only be used be for development purposes. A license is needed when you execute an SSIS package containing a Task Factory component outside of BIDS.



Related Support Articles

- Activate your SentryOne Task Factory license
- Deactivate your SentryOne Task Factory license
- Transfer SentryOne Task Factory licenses to new environment

Task Factory Expressions

Last Modified on 13 January 2022

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Unsupported: The **Property Expressions Editor** and **Expression Builder** are launched from Task Factory as a convenience, but they are not part of the Task Factory software. This article shows how they are accessed from Task Factory and provides a simple example of linking them to the options available for the Task Factory component. Please refer to official Microsoft documentation for assistance with these SSIS features.

Introduction

Some components have an **Expressions** button which allows you to build expressions with the property values associated with the task. It uses values in the package with the operators and functions that are available as part of the core SSIS engine. This allows you to build expressions without having to leave the dialog to configure them.

Note: If you're looking for information about the **Expression** task, see the Task Factory Expression article.

Expressions

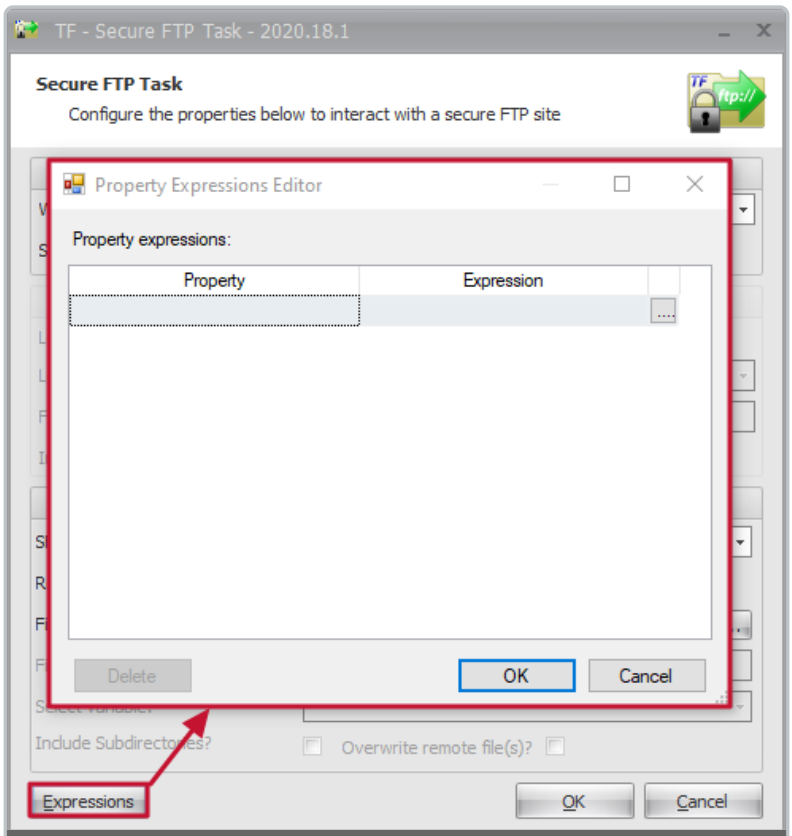
The expressions are built using native expressions in SSIS.

Additional Information: For help with using native expressions in SSIS, see the Integrations Services (SSIS) Expressions article from Microsoft Docs.

Property Expressions Editor

Additional Information: For help with using the **Property Expressions Editor**, see the Use Property Expressions in Packages article from Microsoft Docs.

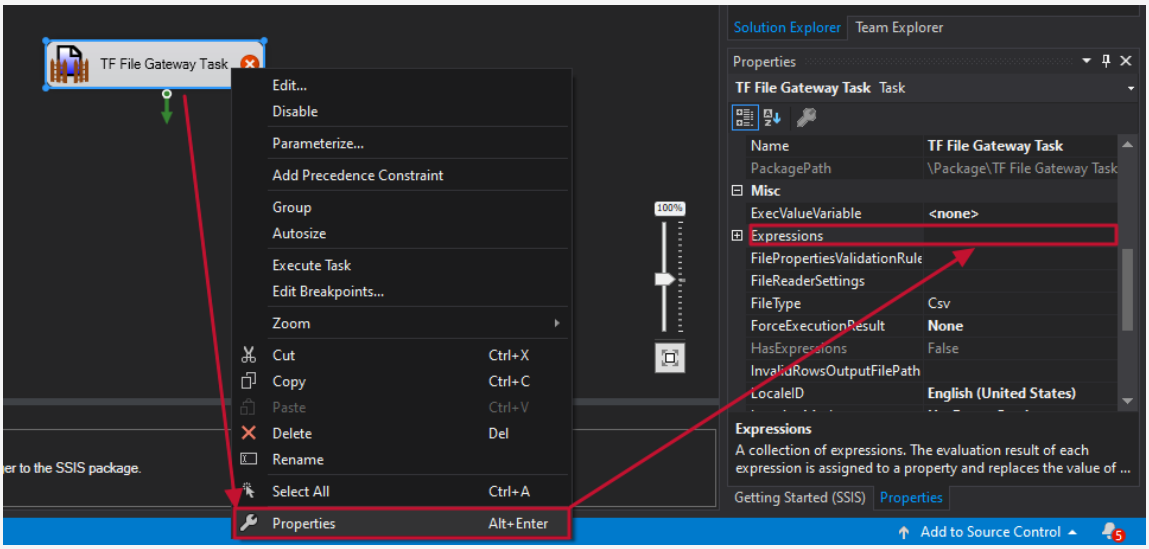
Select the **Expressions** button to open the **Property Expressions Editor**.



Property Expressions Editor

Note: If the Task Factory component doesn't have an **Expressions** button:

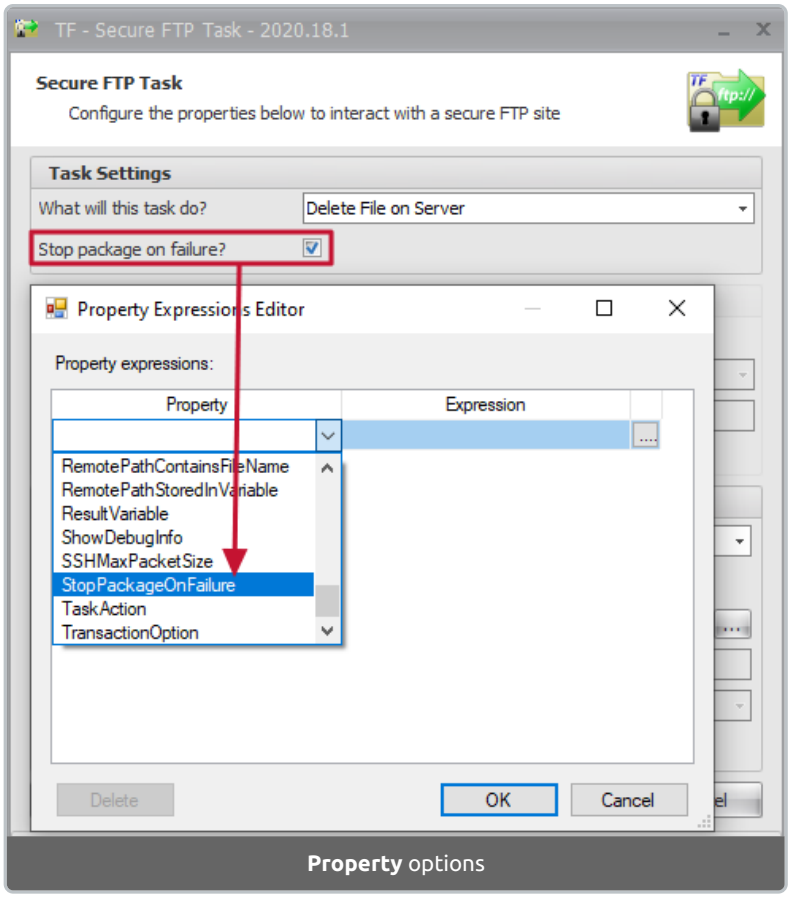
1. Right click the task.
2. Select **Properties**.



3. Select into the **Expressions** option to enable the **ellipsis ···** button.

4. Select the **ellipsis ···** button to launch the **Property Expressions Editor**.

The **Property Expressions Editor** provides the ability to parameterize any of the options available for the Task Factory component (in this example, the **Secure FTP Task**):

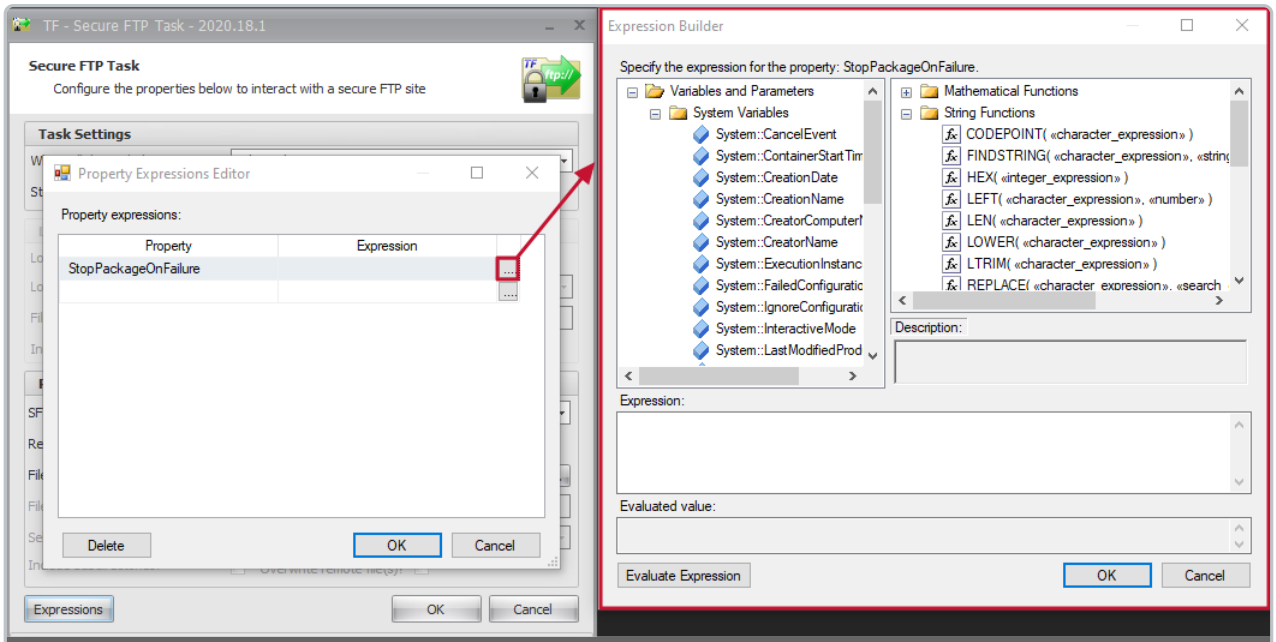


Property options

Expression Builder

Additional Information: For help with the **Expression Builder**, see the Expression Builder article on Microsoft Docs.

Select the ··· to open the **Expression Builder**. This allows you to set properties of SSIS tasks dynamically based on values in the package.



Expression Builder

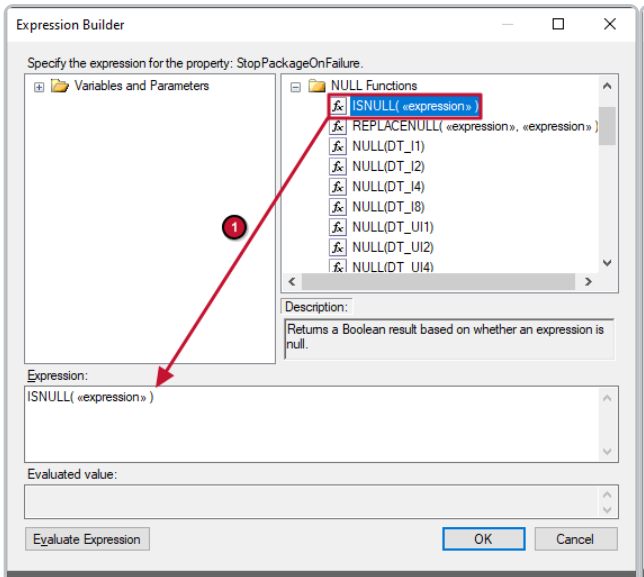
Building Expressions

Additional Information: See the following articles on Microsoft Docs for help with the **System Variables**, **Functions**, **Operators**, and **Casting**:

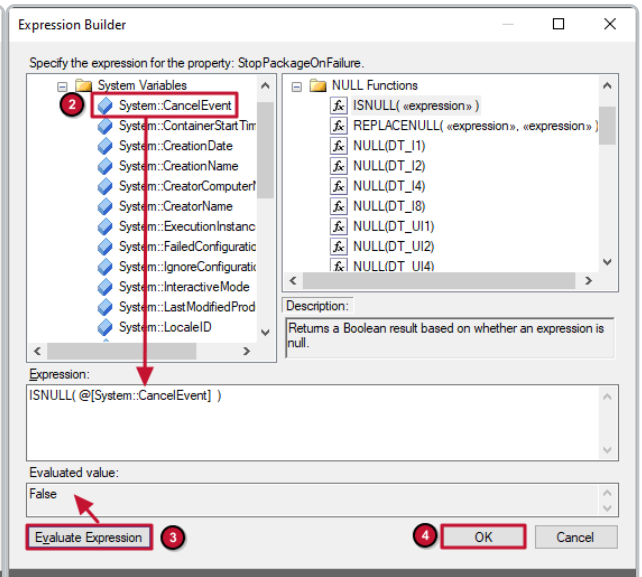
- System Variables for Packages
- Functions (SSIS Expression)
- Operators (SSIS Expression)
- Cast (SSIS Expression)

Drag and drop the options to build an expression. For example:

1. Drag the *ISNULL* function into the **Expression** box
2. Drag the *System::CancelEvent* system variable into the *<>*
3. Select **Evaluate Expression** to verify the syntax and expression
4. Select **OK** to complete building the expression

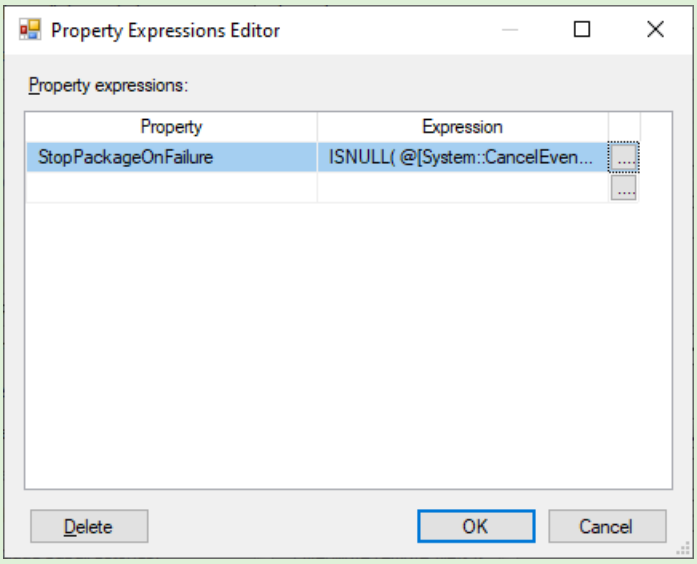


Drag a function into the **Expression**



Build, evaluate, and complete the **Expression**

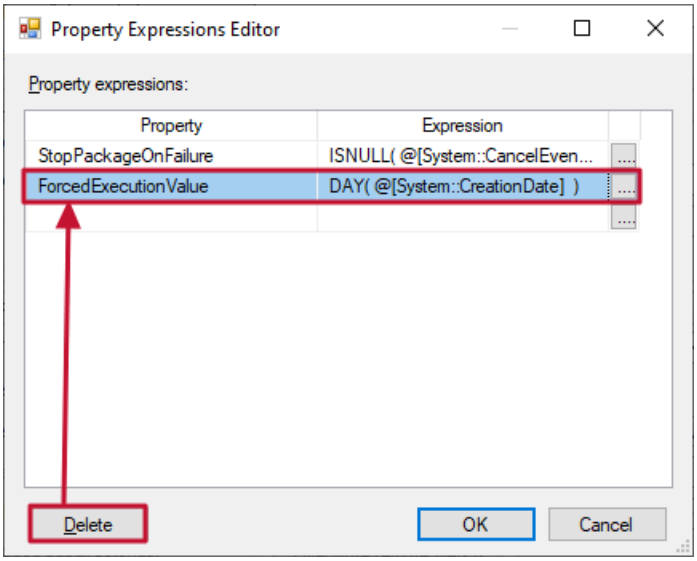
🏆 **Success:** You've built an expression!



ℹ **Note:** Select the ··· on the next line to add another expression.

Deleting Expressions

Highlight an expression, then select **Delete** to remove it:

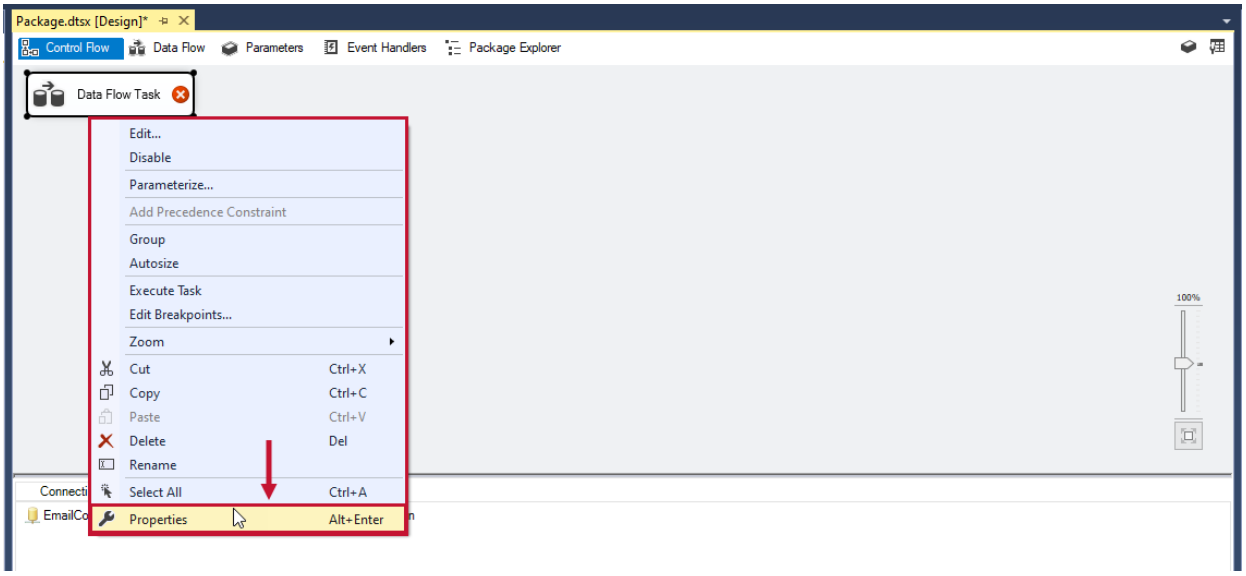


Using Expressions in Task Factory Data Flow Tasks

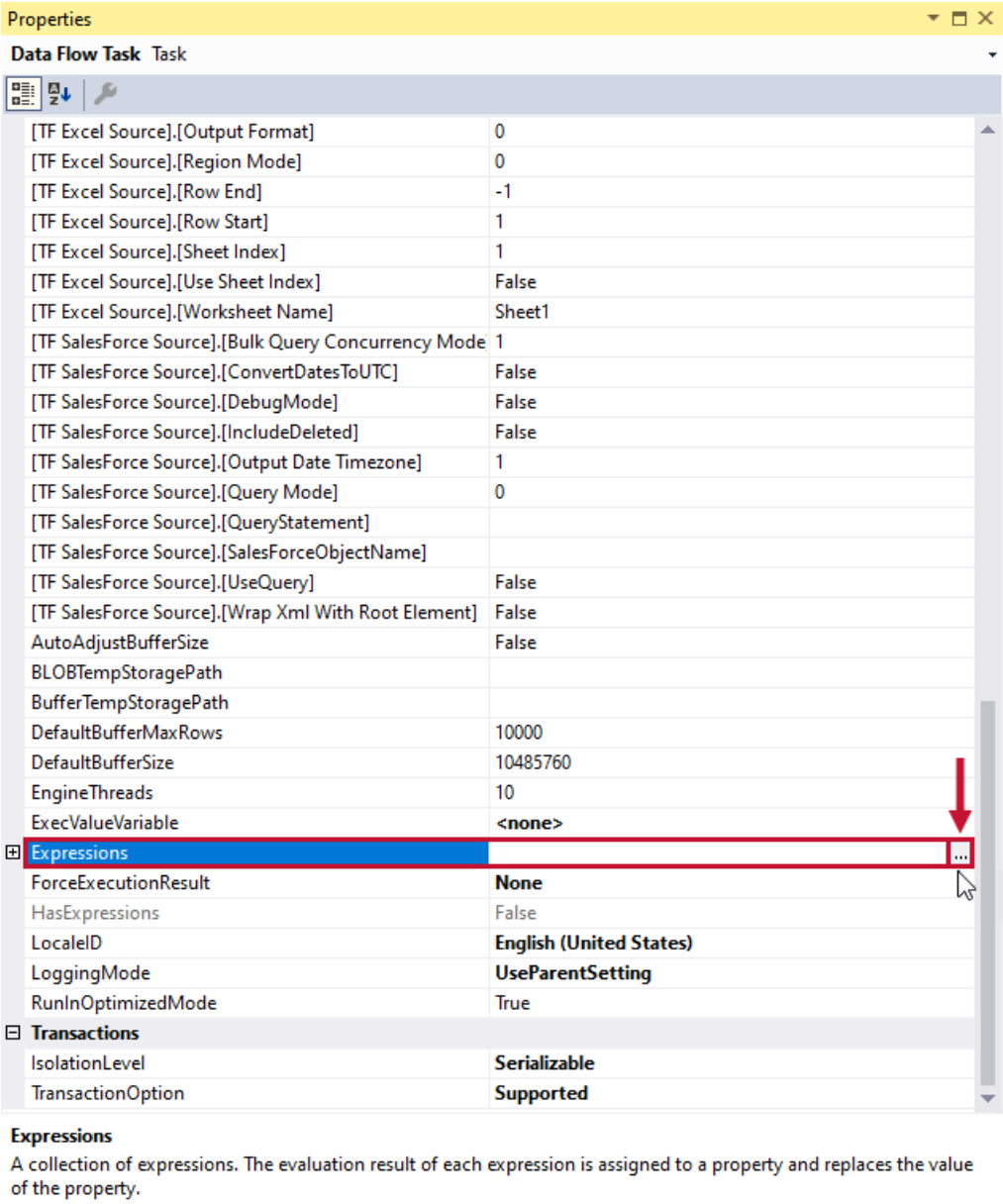
In many of the Task Factory Data Flow Components, you can use the Data Flow Properties and the Property Expressions Editor to select a variable input for your Task Factory Data Flow task.

To use a variable in your Task Factory Data Flow component, complete the following steps:

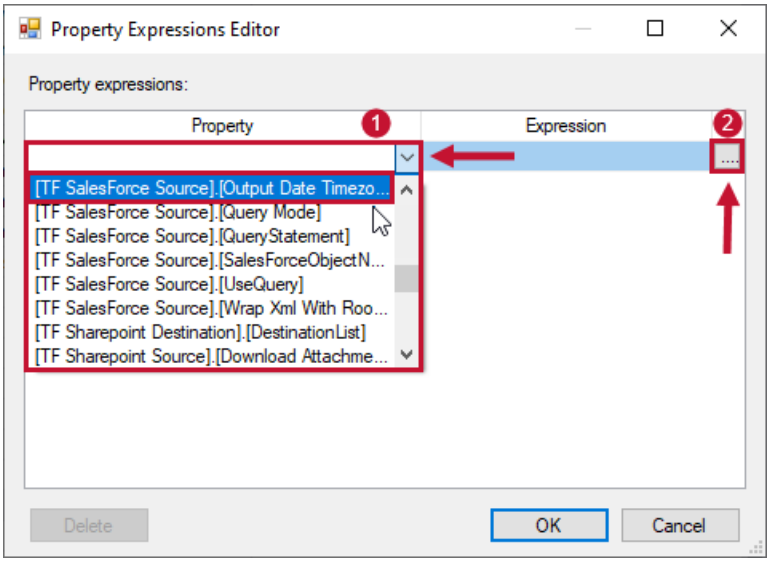
1. Right click your Data Flow, and then select **Properties** to open the Properties window of your Data Flow Task.



2. Select the ••• on the Expressions column to open the Property Expressions Editor window.

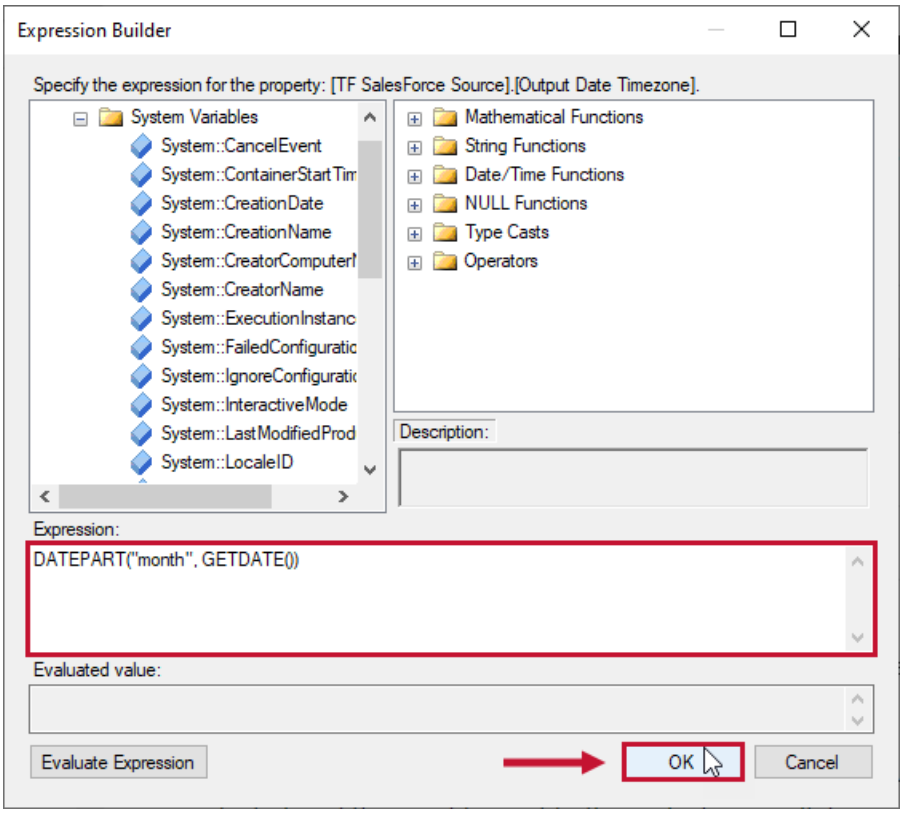


3. Select the Property where you want to apply a variable, and then select the ••• on the Expressions column to open the Expression Builder.



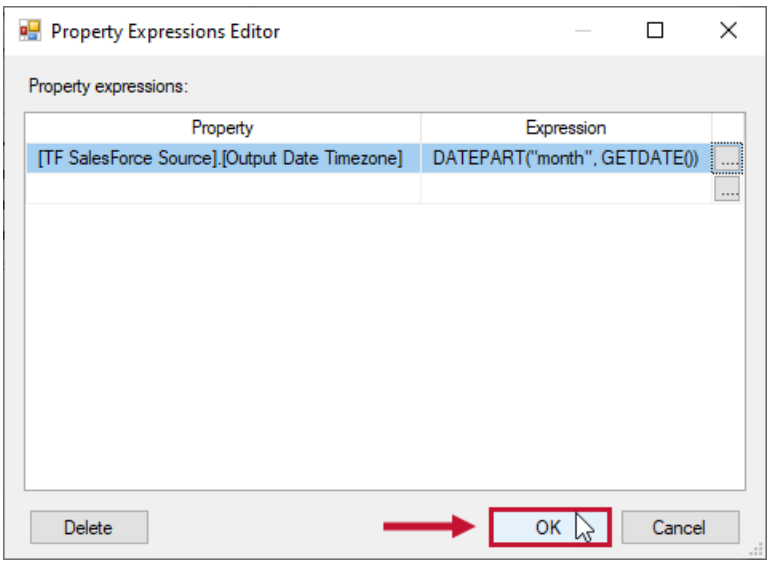
Note: Select the plus icon next to a folder to expand the folder and display the available options.

4. Enter your expression into the Expression: field and then select **Ok** to close the Expression Builder.



Note: Select Evaluate Expression to see a sample of your expression value.

5. Repeat steps 3 and 4 as desired. Select **Ok** to close the Property Expressions Editor.



Additional Information: For more information about Variables in Integrations Services, see the following Integration Services (SSIS) Variables MSFT article.

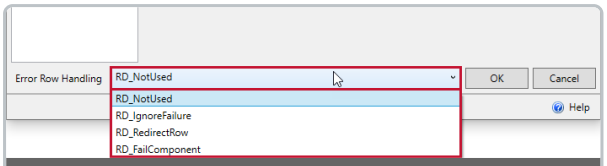
Task Factory Error Row Handling

Last Modified on 13 January 2022

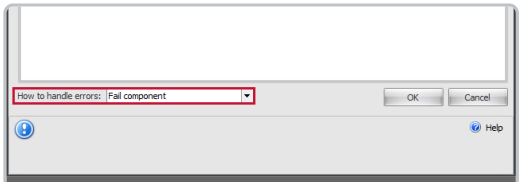
Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Using Error Row Handling

Many components allow for error row handling. These features can be found at the bottom of all Task Factory components that include this functionality.



Error Row Handling



How to handle errors

Note: Some components cannot use the **Not Used** option. This selection is omitted from their error handling options.

Error Row Handling Option	Definition
Not Used **RD_NotUsed**	Error handling is not used and no error constraint is available.
Ignore Failure **RD_IgnoreFailure**	All errors are ignored and the package continues to execute.
Redirect Row **RD_RedirectRow**	All errors are directed to an error output.
Fail Component **RD_FailComponent**	On error, the component fails execution and stops the package.

Task Factory Functions List A-C

Last Modified on 01 October 2020

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Abort

Description	Stops the dataflow and issues a specified error message to the dataflow log file.	
Syntax	ABORT(String_Message [, Condition])	
Returns	NULL	
Parameters		
Name	Optional	Description
String_Message	false	The message you want to display in the dataflow log file when the dataflow stops. The string can be any length and you can enter any valid transformation expression.
Condition	false	(Optional) The conditional expression in which the data flow should be aborted.
Examples		
Abort("Processing aborted"), Abort("Zero Length Detected In Name", Length(FirstName) = 0)		

Abs

Description	Returns the absolute value of a numeric value
Syntax	ABS([Numeric_Value])

Abs

Returns	Positive Numeric value. ABS returns the same datatype as the numeric value passed in as an argument, can be NULL.	
Parameters		
Name	Optional	Description
Numeric_Value	false	Passes the values for which you want to return the absolute values. You can enter any valid transformation expression.

Acos

Description	Returns the angle whose cosine is the specified number.	
Syntax	ACOS(Numeric_Value)	
Returns	Double value. NULL if a value passed to the function is NULL.	
Parameters		
Name	Optional	Description
Numeric_Value	false	Numeric datatype. A number representing a cosine, where number_value must be greater than or equal to -1, but less than or equal to 1. You can enter any valid task editor.
Examples		
Acos(0.5), Acos(varValue)		

AddToDate

AddToDate

Description	Returns a specified date with the specified number interval (integer) added to a specified datepart of that date. If you add or subtract value for specified datepart.	
Syntax	AddToDate(Date, Format, Amount)	
Returns	Returns modified date.	
Parameters		
Name	Optional	Description
Date	false	valid datetime value or expression
Format	false	Valid string format or expression for datepart you want to add/subtract. Supported format values are as below Year Format => Y, YYY, YY, YYYY Month Format => M, MM, MON, MONTH Day Format => D, DD, DDD, DY, DAY Hour Format => H, HH, HH12, HH24 Minute Format => MI, N Second Format => S, SS Milisecond Format => MS
Amount	false	Any valid integer value or expression specifying number of Year, Month, Day, Hour, Minute, Second or Milisecond. If you supply positive value then it will perform add else it will do subtract.
Examples		
AddToDate(ORDER_DATE, 'DD', 3) //Adds 3 days to ORDER_DATE and return new datetime AddToDate(ORDER_DATE, 'HH', 14) //Adds 14 hours to ORDER_DATE and return new datetime AddToDate(ORDER_DATE, 'MM', -5) //Subtract 5 months from ORDER_DATE and return new datetime AddToDate(ORDER_DATE, 'YYYY', 1) //Add one year to supplied ORDER_DATE and return new datetime		

AesDecrypt

AesDecrypt

Description	Returns decrypted data to string format using Advanced Encryption Standard (AES) algorithm with 128-bit encoding.	
Syntax	AesDecrypt(Value, Key)	
Returns	Returns a decrypted binary value that can be NULL if the input value is a null value.	
Parameters		
Name	Optional	Description
Value	false	The Binary datatype you wish to decrypt.
Key	false	The String of precision 16 characters or fewer that was originally used to encrypt the value. You can use mapping variables for the key.
Examples		
AesDecrypt(varEncryptedData, "key-value-1234"), AesDecrypt(EncryptedValueColumn, "key-value-1234"), AesDecrypt(FileReadAllText("C:\data\encrypted.txt"), "key-value-1234")		

AesEncrypt

Description	Returns data in encrypted format using Advanced Encryption Standard (AES) algorithm with 128-bit encoding.	
Syntax	AesEncrypt(Value, Key)	
Returns	Returns an encrypted binary value that can be NULL if the input is a null value.	
Parameters		
Name	Optional	Description
Value	false	The String value you wish to encrypt.

AesEncrypt

Key	false	The String of precision 16 characters or fewer that will be used to encrypt the value. You can use mapping variables for the key.

Examples

AesEncrypt("Encrypt this value", "key-value-1234"), AesEncrypt(PasswordColumn, "key-value-1234"), AesEncrypt(FileReadAllText("C:\data\unencrypted.txt"), "key-value-1234")

Asin

Description	Returns the angle whose sine is the specified number..
Syntax	Asin(Numeric_Value)
Returns	Double value. NULL if a value passed to the function is NULL.

Parameters

Name	Optional	Description
Numeric_Value	false	Numeric datatype. A number representing a sine, where number_value must be greater than or equal to -1, but less than or equal to 1. You can enter any valid task editor.

Examples

Acos(varValue)

Atan

Description	Returns the angle whose tangent is the specified number.

Atan		
Syntax	Atan(Numeric_Value)	
Returns	Numeric	
Parameters		
Name	Optional	Description
Numeric_Value	false	Passes the values for which you want to the Atan of
Examples		
Atan(55)		

Avg		
Description	Aggregation that calculates the average of all values. Can only be used as part of the Advanced Aggregation component	
Syntax	Avg(Value [, Condition])	
Returns	Double.	
Parameters		
Name	Optional	Description
Value	false	The value to add to the aggregation
Condition	false	The condition allows you to conditionally add values to the aggregation based on the expression. For instance, you could define the aggregate as Avg(SalesTotal, CustomerState="FL") and this would only add the value to the aggregation if the customers state was florida.
Examples		

Avg

Avg(SalesTotal), Avg(SalesTotal, CustomerState="FL")

BinToDecimal

Description	Converts a binary value to an decimal value.	
Syntax	BinToDecimal(Value)	
Returns	Decimal	
Parameters		
Name	Optional	Description
Value	false	The binary value to convert.
Examples		
BinToDecimal(10)		

BinToHex

Description	Converts a binary value to an hexadecimal value.	
Syntax	BinToHex(Value)	
Returns	Decimal	
Parameters		
Name	Optional	Description
Value	false	The binary value to convert.
Examples		

BinToHex

BinToHex(10)

BinToOct

Description	Converts a binary value to an octal value.	
Syntax	BinToOct(Value)	
Returns	Decimal	
Parameters		
Name	Optional	Description
Value	false	The binary value to convert.
Examples		
BinToOct(10)		

Ceiling

Description	Returns the smallest integer that is greater than or equal to the numeric value passed to this function.	
Syntax	Ceiling(Numeric_Value)	
Returns	Returns an integer if you pass a numeric value with declared precision between 0 and 28, otherwise it returns a Double. NULL values will return a NULL.	
Parameters		
Name	Optional	Description
Numeric_Value	false	Must be a numeric datatype. You can enter any valid transformation expression.

Choose

Description	Chooses a string from a list of strings based on a given position. You specify the position and the value. If the value matches the position, the Integration Service returns the value. You can either pass in comma delimeted static values or pass in a variable or column with delimeted values	
Syntax	Choose(Index, String1 [, String2, ..., StringN]), Choose(Index, Delimeted_String [, Delimeter_Value=","])	
Returns	The string that matches the position of the index value. NULL if no string matches the position of the index value.	
Parameters		
Name	Optional	Description
Index	false	Numeric datatype. Enter a number based on the position of the value you want to match.
String	false	Any character value.
Examples		
Choose(1, "SentryOne, 4001 Yancey Road, Charlotte, NC") - Will return "SentryOne" Choose(1, varPipeDelimetedAddress, "|")		

Chr

Description	CHR returns the ASCII character corresponding to the numeric value you pass to this function. ASCII values fall in the range 0 to 255. You can pass any integer to CHR, but only ASCII codes 32 to 126 are printable characters.	
Syntax	CHR(Numeric_Value)	
Returns	ASCII or Unicode character. A string containing one character. NULL if a value passed to the function is NULL.	
Parameters		
Name	Optional	Description

Chr		
Numeric_Value	false	Numeric datatype. The value you want to return as an ASCII or Unicode character. You can enter any valid transformation expression.
Examples		
Chr(65), Chr(varCharacterValue)		

ChrCode		
Description	ChrCode returns the numeric ASCII value of the first character of the string passed to the function. ASCII values fall in the range 0 to 255.	
Syntax	ChrCode (String)	
Returns	ASCII or Unicode character. A string containing one character. NULL if a value passed to the function is NULL.	
Parameters		
Name	Optional	Description
String	false	Character string. Passes the values you want to return as ASCII or Unicode values. You can enter any valid task editor.
Examples		
ChrCode("A"), ChrCode("?")		

Concat	
Description	Concatenates two strings.
Syntax	CONCAT(First_String, Second_String)

Concat		
Returns	String. NULL if both string values are NULL. Nulls If one of the strings is NULL, CONCAT ignores it and returns the other string. If both strings are NULL, CONCAT returns NULL.	
Parameters		
Name	Optional	Description
First_String	false	Any datatype except Binary. The first part of the string you want to concatenate. You can enter any valid task editor.
Second_String	false	Any datatype except Binary. The second part of the string you want to concatenate. You can enter any valid task editor.
Examples		
Concat("Sentry", " ", "One"), Concat(FirstNameColumn, " ", LastNameColumn)		

ConvertBase		
Description	Converts a integer base from one to another.	
Syntax	ConvertBase(Value, From_Base, To_Base)	
Returns	Integer.	
Parameters		
Name	Optional	Description
Value	false	The value to change the base of.
From_Base	false	The existing base of the integer. Valid values are 2, 8, 10, 16
To_Base	false	The new base of the integer. Valid values are 2, 8, 10, 16

ConvertBase

Examples
ConvertBase("1234", 10, 2)

Cos

Description	Returns the cosine of a numeric value (expressed in radians).	
Syntax	Cos(Numeric_Value)	
Returns	Double value. NULL if a value passed to the function is NULL.	
Parameters		
Name	Optional	Description
Numeric_Value	false	Numeric datatype. Numeric data expressed in radians (degrees multiplied by pi divided by 180). Passes the values for which you want to calculate a cosine. You can enter any valid task editor.
Examples		
Cos(10), Cos(varValue)		

Cosh

Description	Returns the hyperbolic cosine of a numeric value (expressed in radians).
Syntax	Cosh(Numeric_Value)
Returns	Double value. NULL if a value passed to the function is NULL.
Parameters	

Cosh		
Name	Optional	Description
Numeric_Value	false	Numeric datatype. Numeric data expressed in radians (degrees multiplied by pi divided by 180). Passes the values for which you want to calculate the hyperbolic cosine. You can enter any valid task editor.
Examples		
Cosh(10), Cosh(varValue)		

Count (*)		
Description	Aggregation that calculates the total number based on the grouping. Can only be used as part of the Advanced Aggregation component	
Syntax	Count(Value [, Condition])	
Returns	Integer.	
Parameters		
Name	Optional	Description
Value	false	The value to add to the aggregation
Condition	false	The condition allows you to conditionally add values to the aggregation based on the expression. For instance, you could define the aggregate as Count(*, CustomerState="FL") and this would only add the value to the aggregation if the customers state was florida.
Examples		

Count (*)

Count(*), Count(*, CustomerState="FL")

Count (column name)

Description	Aggregation that calculates the total number based on the grouping. Can only be used as part of the Advanced Aggregation component	
Syntax	Count(Column_Name [, Condition])	
Returns	Integer.	
Parameters		
Name	Optional	Description
Column_Name	false	The column to retrieve the value of to add to the aggregation
Condition	false	The condition allows you to conditionally add values to the aggregation based on the expression. For instance, you could define the aggregate as Count(CustomerID, CustomerState="FL") and this would only add the value to the aggregation if the customers state was florida.
Examples		
Count(CustomerID), Count(CustomerID, CustomerState="FL")		

Task Factory Functions List D-E

Last Modified on 01 October 2020

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

DateCompare

Description	Compares two dates against each other.	
Syntax	DateCompare(Date1, Date2)	
Returns	Integer. If Date1 is greater than Date2, 1 is returned. If Date1 equals Date2, 0 is returned. If Date1 is less than Date2, -1 is returned.	
Parameters		
Name	Optional	Description
Date1	false	The first date in the comparison.
Date2	false	The second date in the comparison.
Examples		
DateCompare("7/4/2012", "7/5/2012"), DateCompare(varStartDate, varEndDate), DateCompare(OrderDateColumn, StartDateColumn)		

DateDiff

Description	Returns the difference between two dates against each other.	
Syntax	DateDiff(Date1, Date2, Format)	
Returns	Integer. The difference based on datepart between the two days.	
Parameters		
Name	Optional	Description
Date1	false	The first date in the comparison.

DateDiff

Date2	false	The second date in the comparison.
Format	false	The datepart tells the DateDiff what part of the date to compare. Year = "Y", "YY", "YYY", "YYYY" Day = "D", "DD", "DDD", "DY", "DAY" Month = "M", "MM", "MON", "MONTH" Hour = "H", "HH", "HH12", "HH24" Minute = "MI" Second = "S", "SS"" Millisecond = "MS" Microsecond = ""US" Week Of Year = "W", "WW" Quarter = "Q", "QQ"
Examples		
DateDiff("7/4/2012", "7/5/2012", "d"), DateDiff(varStartDate, varEndDate, "M"), DateDiff(OrderDateColumn, StartDateColumn, "H")		

DatePart

Description	Returns the part of the date specified in the datepart parameter.	
Syntax	DatePart(Date, Date_Part)	
Returns	Object.	
Parameters		
Name	Optional	Description
Date	false	The date used to retrieve the datepart from.

DatePart

Date_Part	false	The datepart tells the DatePart what part of the date to retrieve. Year = "Y", "YY", "YYY", "YYYY" Day = "D", "DD", "DY", "DAY" Month = "M", "MM", "MON", "MONTH" Hour = "H", HH", "HH12", "HH24" Minute = "MI" Second = "SS"" Millisecond = "MS" Microsecond = ""US" Day Of Week (1-7) = "WD","W" Week Of Year = "WW" Day Of Year = "DDD", "YD" Quarter = "Q" Julian Date = "J"
Examples		
DatePart("7/4/2012", "d"), DatePart(varStartDate, "MM"), DatePart(OrderDateColumn, "HH")		

Day

Description	Returns the Day from the date specified.	
Syntax	Day(Date)	
Returns	returns integer value of 1 - 31	
Parameters		
Name	Optional	Description
Date	false	Date to retrieve day from
Examples		
Day("12/8/2011"), Day(varDate), Day(OrderDateColumn)		

Day_Of_Week

Description	Returns the Day of the week from the date specified.
Syntax	Day_Of_Week(date)

Day_Of_Week

Returns	returns integer value of 1 - 7	
Parameters		
Name	Optional	Description
Date	false	Date to retrieve day of week from
Examples		
Day_Of_Week("12/8/2011"), Day_Of_Week(varDate), Day_Of_Week(OrderDateColumn)		

DayName

Description	Returns the name of the day (Monday, Tuesday, etc) from the date specified.	
Syntax	DayName(Date, Return_Shortname)	
Returns	returns string. Name of the day	
Parameters		
Name	Optional	Description
Date	false	Date to retrieve day name from
Return_Shortname	false	Return the short name of the day (Mon, Tue, Wed)
Examples		
DayName("12/8/2011", false), DayName(varDate, True), DayName(OrderDateColumn, false)		

DayOfYear

Description	Returns the day of the year of the date specified.
Syntax	DayOfYear(Date)
Returns	returns integer value of 1 - 366
Parameters	

DayOfYear

Name	Optional	Description
Date	false	Date to retrieve day of year from
Examples		
DayOfYear("12/8/2011"), DayOfYear(varDate), DayOfYear(OrderDateColumn)		

DD_DELETE

Description	DD_DELETE is equivalent to the integer literal 2.

DD_INSERT

Description	DD_INSERT is equivalent to the integer literal 0.

DD_REJECT

Description	DD_REJECT is equivalent to the integer literal 3.

DD_UPDATE

Description	DD_INSERT is equivalent to the integer literal 1.

DecBase64

Description	Decodes a base 64 encoded value and returns a string with the binary data representation of the data. If you encode data using EncBase64, and you want to decode data using DecBase64, you must run the decoding dataflow using the same data movement mode. Otherwise, the output of the decoded data may differ from the original data.

DecBase64

Syntax	DecBase64(Value)	
Returns	Binary decoded value. NULL if the input is a null value. Return values differ if you run the dataflow in Unicode mode versus ASCII mode.	
Parameters		
Name	Optional	Description
Value	false	String datatype. Data that you want to decode.
Examples		
example goes here		

DecimalToBin

Description	Converts a decimal value to an binary value.	
Syntax	DecimalToBin(Value)	
Returns	String	
Parameters		
Name	Optional	Description
value	false	The decimal value to convert.
Examples		
DecimalToBin(10)		

DecimalToHex

Description	Converts a decimal value to an hexadecimal value.
Syntax	DecimalToHex(Value)
Returns	String
Parameters	

DecimalToHex

Name	Optional	Description
Value	false	The decimal value to convert.
Examples		
DecimalToHex(10)		

DecimalToOct

Description	Converts a decimal value to an octal value.	
Syntax	DecimalToHex(Value)	
Returns	Int64	
Parameters		
Name	Optional	Description
Value	false	The decimal value to convert.
Examples		
DecimalToOct(10)		

Decode

Description	Searches a string for a value you specify. If the function finds the value, it returns a result value, which you define. You can build an unlimited number of searches within a DECODE function.
Syntax	DECODE(Value, First_Search, First_Result [, Second_Search, Second_Result]...[, Default])
Returns	First_result if the search finds a matching value. Default value if the search does not find a matching value. NULL if you omit the default argument and the search does not find a matching value. Even if multiple conditions are met, the Integration Service returns the first matching result. If the data contains multibyte characters and the DECODE expression compares string data, the return value depends on the code page and data movement mode of the Integration Service.
Parameters	

Decode

Name	Optional	Description
Value	false	Any datatype except Binary. Passes the values you want to search. You can enter any valid task editor.
Search	false	Any value with the same datatype as the value argument. Passes the values for which you want to search. The search value must match the value argument. You cannot search for a portion of a value. Also, the search value is case sensitive. For example, if you want to search for the string 'Halogen Flashlight' in a particular port, you must enter 'Halogen Flashlight, not just 'Halogen'. If you enter 'Halogen', the search does not find a matching value. You can enter any valid task editor.
Result	false	Any datatype except Binary. The value you want to return if the search finds a matching value. You can enter any valid task editor.
Default	true	Any datatype except Binary. The value you want to return if the search does not find a matching value. You can enter any valid task editor.
Examples		
Decode(ProductCode, "R", "Rims", "S", "Silver Lining", "M", "Mats")		

DeleteFilesOlderThanNDays

Description	Deletes from from a directory that are N days old.
Syntax	DeleteFilesOlderThanNDays(Path, Pattern, Older_Than_Days, Recursive [, Continue_On_Failure])
Returns	Integer. Returns the number of files deleted
Parameters	

DeleteFilesOlderThanNDays

Name	Optional	Description
Path	false	The path where the file will be deleted from.
Pattern	false	The filter pattern for locating and deleting files. *.* = all files, *.xls= all xls files, 2012*.doc = any file that starts with 2012 and ends with .doc.
Older_Than_Days	false	The number of days that will be used to delete files older than.
Recursive	false	Delete files from subdirectories as well.
Continue_On_Failure	false	Continue deleting files even if one or more files fails to delete.
Examples		
DeleteFilesOlderThanNDays("C:\ssis\test", "*.xml", 14, false, true)		

DirCreate

Description	Creates a directory in the path supplied.	
Syntax	DirCreate(Path)	
Returns	Boolean. True if the Directory was created. False if it was not or path parameter is null	
Parameters		
Name	Optional	Description
Path	false	The path where the directory will be created.
Examples		
DirCreate("C:\ssis\test")		

DirDelete

DirDelete

Description	Removes a directory in the path supplied.	
Syntax	DirDelete(Path)	
Returns	Boolean. True if the Directory was removed. False if it was not or path parameter is null	
Parameters		
Name	Optional	Description
Path	false	The path where the directory will be removed from.
Examples		
DirDelete("C:\ssis\test")		

DirExists

Description	Check if the directory exists in the path supplied.	
Syntax	DirExists(Path)	
Returns	Boolean. True if the Directory exists. False if it does not or path parameter is null	
Parameters		
Name	Optional	Description
Path	false	The path where the directory is located.
Examples		
DirExists("C:\ssis\test")		

DirGetCreationTime

Description	Return the creation time of the directory.
Syntax	DirGetCreationTime(Path)

DirGetCreationTime

Returns	DateTime. If the path does not exist, the minimum date for your system is returned.	
Parameters		
Name	Optional	Description
Path	false	The path where the directory is located.
Examples		
DirGetCreationTime("C:\ssis\")		

DirGetLastAccessTime

Description	Return the time the directory was last accessed.	
Syntax	DirGetLastAccessTime(Path)	
Returns	DateTime. If the path does not exist, the minimum date for your system is returned.	
Parameters		
Name	Optional	Description
Path	false	The path where the directory is located.
Examples		
DirGetLastAccessTime("C:\ssis\")		

DirGetLastWriteTime

Description	Return the time the directory was last written to.
Syntax	DirGetLastWriteTime(Path)
Returns	DateTime. If the path does not exist, the minimum date for your system is returned.
Parameters	

DirGetLastWriteTime

Name	Optional	Description
Path	false	The path where the directory is located.
Examples		
DirGetLastWriteTime("C:\ssis\")		

DirGetRoot

Description	Returns the root directory of the path specified.	
Syntax	DirGetRoot(Path)	
Returns	String. Null if the path parameter is Null	
Parameters		
Name	Optional	Description
Path	false	The path where the directory is located.
Examples		
DirGetRoot("C:\ssis\test")		

DirMove

Description	Moves a directory in from one path to another.	
Syntax	DirMove(Path_Source, Path_Destination)	
Returns	Boolean. True if the Directory was moved. False if it was not or either the Path_Source or Path_Destination parameters are null	
Parameters		
Name	Optional	Description
Path_Source	false	The path where the directory will be moved from.

DirMove

Path_Destination	false	The path where the directory will be moved to.
Examples		
DirMove("C:\ssis\test", "C:\ssis\test1")		

DirSetCreationTime

Description	Set the time the directory was created.	
Syntax	DirSetCreationTime(Path, Date_Created)	
Returns	Boolean. True if the value is set. False if either parameter is null.	
Parameters		
Name	Optional	Description
Path	false	The path where the directory is located.
Date_Created	false	The date the directory was created.
Examples		
DirSetCreationTime("C:\ssis\", "7/6/2012 14:24")		

DirSetLastAccessTime

Description	Set the time the directory was last accessed.	
Syntax	DirSetLastAccessTime(Path, Date_Last_Access)	
Returns	Boolean. True if the value is set. False if either parameter is null.	
Parameters		
Name	Optional	Description
Path	false	The path where the directory is located.

DirSetLastAccessTime

Date_Last_Access	false	The date the directory was last accessed.
Examples		
DirSetLastAccessTime("C:\ssis\", "7/6/2012 14:24")		

DirSetLastWriteTime

Description	Set the time the directory was written to accessed.	
Syntax	DirSetLastWriteTime(Path, Date_Last_Written)	
Returns	Boolean. True if the value is set. False if either parameter is null.	
Parameters		
Name	Optional	Description
Path	false	The path where the directory is located.
Date_Last_Written	false	The date the directory was last written to.
Examples		
DirSetLastWriteTime("C:\ssis\", "7/6/2012 14:24")		

EncBase64

Description	Encodes data by converting binary data to string data using Multipurpose Internet Mail Extensions (MIME) encoding. Encode data when you want to store data in a database or file that does not allow binary data.	
Syntax	EncBase64(Value)	
Returns	Encoded value. NULL if the input is a null value.	
Parameters		
Name	Optional	Description

EncBase64

Value	false	Binary or String datatype. Data that you want to encode.
Examples		
example goes here		

EndsWith

Description	EndsWith determines whether a string ends with a character or string value	
Syntax	EndsWith (String_To_Search, Search_Value)	
Returns	Boolean. True if string_to_search ends with search_value. NULL if a value passed to the function is NULL.	
Parameters		
Name	Optional	Description
String_To_Search	false	Character string. The string to search.
Search_Value	false	Character string. The value to find at the end of String_To_Search
Examples		
EndsWith("400 College Dr", "Dr"), If EndsWith(AddressColumn, "Dr") Then ...)		

Error

Description	Causes the Integration Service to skip a row and issue an error message, which you define. The error message displays in the progress log.	
Syntax	ERROR(String)	
Returns	String.	
Parameters		
Name	Optional	Description

Error

String	false	String value. The message you want to display when the Integration Service skips a row based on the expression containing the ERROR function. The string can be any length.
Examples		
Error("Whoops! Something went wrong!")		

ExecuteSQL

Description	Executes supplied sql statement or stored procedure and returns single value or full resultset depending on column_name_or_index setting. By default it returns first column of first row in the resultset. If you execute DDL (e.g. CREATE TABLE...) or DML (UPDATE/DELETE...) statements which doesnt return anything then it will return NULL. You can use this function to call simple or parameterized SELECT queries or stored procs. You can also call DDL/DML statemets (e.g. UPDATE/DELETE/ALTER/INSERT/TRUNCATE)	
Syntax	ExecuteSQL(connection_manager, sql_statement [, is_stored_procedure] [, command_timeout] [, column_name_or_index] [, param1, param2 ... paramN])	
Returns	Returns single value or full resultset as ADO.net DataTable. If no result set or value found (e.g. UPDATE/DELETE Statement) then returns NULL	
Parameters		
Name	Optional	Description

ExecuteSQL		
Connection_Manager	false	The Connection Manager which will use to execute provided sql_statement. It has to be either OLEDB or ADO.net connection manager. Syntax for connection manager is @@[connectionname]. When you use OLEDB connection to execute queries then you cant define named parameters (e.g. @mypara). You have to use "?" for parameter (Example: select * from customer where customerid=?). When you use ADO.net connection type then you can use named parameters in your queries (e.g. select * from customer where customerid=@in_customer)
Sql_Statement	false	String value used representing the Stored Procedure that will be executed.
Is_Stored_Procedure	true	Boolean value indicating whether the Sql_Statement is a Stored Procedure or not. When this parameter is True then you dont have to include parameters as part of sql_statement. (Default=False)
Command_Timeout	true	Timeout in seconds for sql command. 0 means unlimited (Default=0)
Column_Name_OR_Index	true	If you have more than one column in output rowset and you want to return other than first column then specify column by name or Index (starting from 0). If you pass "-1" (in double quotes) then it will return full result set as ADO.net DataTable which you can store in object datatype variable and consume it later in script task or use it in ForEach Loop
		List of parameters for supplied sql statement. If sqlstatement/stored proc doesn't require paramets then ommit this argument. Parameters are passed using below functions. InParam(paraName

Parameters	ExecuteSQL true	[,DataType] [,Precision/Length] [,Scale] , Value) : This function defines Input parameter OutParam(paraName ,DataType ,Precision/Length , Scale , variableNameOrVariablePortName) : This function defines Output parameter Examples: ---------- OLEDB CONNECTION EXAMPLE ---------- -- Single Input and output Parameter for OLEDB connection (Use ? for parameter indicator. Nmaed parameters not supported with OLEDB). Returns value back to User Variable "var2" ExecuteSQL(@@[NorthwindOLEDB], "SELECT ?=COUNT(*) FROM ORDERS WHERE CUSTOMERID=?" ,FALSE ,OutParam("out1",DbType.Int32,0,0,"User -- See variable name is in Double quotes.. don't pass as @[var2] ,InParam("prod1","ALFKI")) ---------- ADO.NET CONNECTION EXAMPLE ---------- --Single Input and output Parameter for ADO.net connection (Use @ prefix for named parameters). Returns value back to Variable Column V_COUNT (only for DataFlow transforms) ExecuteSQL(@@[NorthwindOLEDB], "SELECT @MyCount=COUNT(*) FROM ORDERS WHERE CUSTOMERID=@pCustID " ,FALSE ,OutParam("MyCount",DbType.Int32,0,0, V_COUNT) ---See column name is not in DOUBLE QUOTES. This column must be variable column (Not output column) ,InParam("pCustID","ALFKI")) Here is list of supported datatypes which can be used with InParam or OutParam functions http://msdn.microsoft.com/en-us/library/system.data.dbtype.aspx AnsiString = 0, Binary = 1, Byte = 2, Boolean = 3, Currency = 4, Date = 5, DateTime = 6, Decimal = 7, Double = 8, Guid = 9, Int16 = 10, Int32 = 11, Int64 = 12, Object = 13, SByte = 14, Single = 15, String = 16, Time = 17, UInt16 = 18, UInt32 = 19, UInt64 = 20, VarNumeric = 21, AnsiStringFixedLength = 22, StringFixedLength = 23, Xml = 25, DateTime2 = 26, DateTimeOffset = 27

ExecuteSQL

Examples

========== Examples (Calling DML/DDL Statements e.g. CREATE/UPDATE/INSERT/DELETE): ========== --Call simple sql statement which doesn't contain any paramter ExecuteSQL(@@[NorthwindOLEDB] , "DELETE FROM Customer Where Country='USA'") --Call simple create table ExecuteSQL(@@[NorthwindOLEDB] , "CREATE TABLE MyTable(col1 int, col2 int)") ========== Examples (Returning data from stored proc or select query): ========== --Return first column of first row ExecuteSQL(@@[NorthwindOLEDB] , "SELECT * FROM Customer Where Country='USA'") --Return "Phone" column from 1st row of resultset [Use of Column Name] ExecuteSQL(@@[NorthwindOLEDB] , "SELECT * FROM Customer Where Country='USA'" ,False,0, "Phone") --Call stored proc and return "data" column from first row of sp_spaceused output ExecuteSQL(@@[NorthwindOLEDB] , "exec sp_spaceused 'dbo.Customers'" ,False,0, "data") --Return 5th column from 1st row of resultset [Use of ColumnIndex] ExecuteSQL(@@[NorthwindOLEDB] , "SELECT * FROM Customer Where Country='USA'" ,False,0, 5) --Return Full Datatable (Call from ExpressionTask. Bind Expression with variable with object datatype to store full DataTable) ExecuteSQL(@@[NorthwindOLEDB] , "SELECT * FROM Customer Where Country='USA'" ,False,0, "-1") --Timeout query if it takes more than 30 seconds ExecuteSQL(@@[NorthwindOLEDB] , "SELECT * FROM Customer Where Country='USA'" ,False,30) ========== Parameter Examples (OLEDB Connection): ========== --Single Input Parameter for OLEDB connection ExecuteSQL(@@[NorthwindOLEDB] , "SELECT * FROM Customer Where Country=? " ,False, InParam("para1",DbType.String,40,0, "USA")) --Single Input Parameter for OLEDB connection (Don't pass datatype info.. and let it guess)… If it can't guess it woll use String(4000) ExecuteSQL(@@[NorthwindOLEDB] , "SELECT * FROM Customer Where Country=? " ,False, InParam("para1", "USA")) --Single Input and output Parameter for OLEDB connection. Returns value back to User Variable "var2" ExecuteSQL(@@[NorthwindOLEDB], "SELECT ?=COUNT(*) FROM ORDERS WHERE CUSTOMERID=?" ,FALSE ,OutParam("out1",DbType.Int32,0,0,"User::var2") -- See variable name is in Double quotes.. don't pass as @[var2] ,InParam("prod1","ALFKI")) --Return value from stored procedure (RETURN VALUE always numeric and output from stored proc via RETURN statement) --Single output Parameter for OLEDB connection. Returns value back to User Variable "var2" ExecuteSQL(@@[NorthwindOLEDB], "exec ?=sp_MyStoredProc" ,FALSE ,OutParam("out1",DbType.Int32,0,0,"User::var2") -- See variable name is in Double quotes.. don't pass as @[var2].) --Single Input and output Parameter for OLEDB connection. Returns value back to Variable column V_COUNT (only for DataFlow transforms) ExecuteSQL(@@[NorthwindOLEDB], "SELECT @MyCount=COUNT(*) FROM ORDERS WHERE CUSTOMERID=@pCustID " ,FALSE ,OutParam("MyCount",DbType.Int32,0,0, V_COUNT) ---See port name is not in DOUBLE QUOTES ,InParam("pCustID","ALFKI")) ========== Parameter Examples (ADO.net Connection): ========== --Single Input Parameter for ADONET connection ExecuteSQL(@@[NorthwindADONET] , "SELECT * FROM Customer Where Country=@pCountry" ,False, InParam("pCountry ",DbType.String,40,0, "USA")) --Single Input and output Parameter for ADONET connection. Returns value back to User Variable "var2" ExecuteSQL(@@[NorthwindADONET], "SELECT @MyCount=COUNT(*) FROM ORDERS WHERE CUSTOMERID=@pCustID " ,FALSE ,OutParam("MyCount",DbType.Int32,0,0,"User::var2") ,InParam("pCustID","ALFKI")) --Single Input and output Parameter for ADONET connection. Returns value back to Variable column V_COUNT (only for DataFlow transforms) ExecuteSQL(@@[NorthwindADONET], "SELECT @MyCount=COUNT(*) FROM ORDERS WHERE CUSTOMERID=@pCustID " ,FALSE ,OutParam("MyCount",DbType.Int32,0,0, V_COUNT) ,InParam("pCustID","ALFKI"))

Exp

Description	Returns Euler's Number, e, raised to the specified power (Exponent), where e=2.71828183. For example, EXP(2) returns 7.38905609893065. You might use this function to analyze scientific and technical data rather than business data. EXP is the reciprocal of the LN function, which returns the natural logarithm of a numeric value.

Exp

Syntax	EXP(Exponent)	
Returns	Double value. NULL if a value passed as an argument to the function is NULL.	
Parameters		
Name	Optional	Description
Exponent	false	Numeric datatype. The value to which you want to raise e. The exponent in the equation e^value. You can enter any valid task editor.
Examples		
EXP(ExponentColumn)		

Task Factory Functions List F

Last Modified on 02 October 2020

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

FileAppendAllText

Description	Append text to the end of a file contents.	
Syntax	FileAppendAllText(Path, Text [, Encoding])	
Returns	Boolean. If the path does not exist, null is returned.	
Parameters		
Name	Optional	Description
Path	false	The path where the file is located.
Text	false	Text to append to file contents.
Encoding	true	Encoding to use to write to file. Choices are ASCII, UNICODE, UTF7, UTF8, UTF32 or Default.
Examples		
FileAppendAllText("C:\ssis\test.xml", "Wrox Professional SSIS 2012")		

FileChangeExtension

Description	Changes the file extension from one to another.
Syntax	FileChangeExtension(Path, New_Extension)
Returns	String. The path to the file with the new extension.

FileChangeExtension

Parameters		
Name	Optional	Description
Path	false	The path to the file.
New_Extension	false	The new extension of the file with or without the period. Pass in null to remove the extension.
Examples		
FileChangeExtension("C:\ssis\test.xml", "txt")		

FileCombinePath

Description	Combines the path of files together. It handles all of the checking for slashes and adds or removes them.	
Syntax	FileCombinePath(Path1, Path2)	
Returns	String. The combination of path1 and path2 with all slashes added or removed.	
Parameters		
Name	Optional	Description
Path1	false	The first part of the path.
Path2	false	The second part of the path.
Examples		
FileCombinePath("C:\ssis", "test.xml")		

FileCopy

FileCopy

Description	Copy a file from one location to another .	
Syntax	FileCopy(Source, Destination [, Overwrite])	
Returns	Boolean. If the copy succeeds. Null of either parameter is null or either the source file doesn't exist.	
Parameters		
Name	Optional	Description
Source	false	The source path file path.
Destination	false	The destination file path.
Overwrite	false	Overwrite the existing file?.
Examples		
FileCopy("C:\ssis\test.xml", "C:\ssis\processedfiles\test.xml")		

FileDelete

Description	Delete a file in the path supplied.	
Syntax	FileDelete(Path)	
Returns	Boolean. True if the file was removed. False if it was not or path parameter is null	
Parameters		
Name	Optional	Description
Path	false	The path where the file will be removed from.
Examples		
FileDelete("C:\ssis\test")		

FileExists

Description	Check if the file exists in the path supplied.	
Syntax	FileExists(Path)	
Returns	Boolean. True if the file exists. False if it does not or path parameter is null	
Parameters		
Name	Optional	Description
Path	false	The path where the file is located.
Examples		
FileExists("C:\ssis\test.xml")		

FileGetCreationTime

Description	Return the creation time of the file.	
Syntax	FileGetCreationTime(Path)	
Returns	DateTime. If the path does not exist, the minimum date for your system is returned.	
Parameters		
Name	Optional	Description
Path	false	The path where the file is located.
Examples		
FileGetCreationTime("C:\ssis\test.xml")		

FileGetDirectoryName

Description	Get the directory of a file path.	
Syntax	FileGetDirectoryName(Path)	
Returns	String	
Parameters		
Name	Optional	Description
Path	false	The path to the file.
Examples		
FileGetDirectoryName("C:\ssis\test.xml")		

FileGetExtension

Description	Get the extension of a file.	
Syntax	FileGetExtension(Path)	
Returns	String	
Parameters		
Name	Optional	Description
Path	false	The path to the file.
Examples		
FileGetExtension("C:\ssis\test.xml")		

FileGetFullPath

FileGetFullPath

Description	Get the full path to a file.	
Syntax	FileGetFullPath(Path)	
Returns	String	
Parameters		
Name	Optional	Description
Path	false	The path to the file.
Examples		
FileGetFullPath("C:\ssis\test.xml")		

FileGetLastAccessTime

Description	Return the time the file was last accessed.	
Syntax	FileGetLastAccessTime(Path)	
Returns	DateTime. If the path does not exist, the minimum date for your system is returned.	
Parameters		
Name	Optional	Description
Path	false	The path where the file is located.
Examples		
FileGetLastAccessTime("C:\ssis\test.xml")		

FileGetLastWriteTime

FileGetLastWriteTime

Description	Return the time the file was last written to.	
Syntax	FileGetLastWriteTime(Path)	
Returns	DateTime. If the path does not exist, the minimum date for your system is returned.	
Parameters		
Name	Optional	Description
Path	false	The path where the file is located.
Examples		
FileGetLastWriteTime("C:\ssis\test.xml")		

FileGetName

Description	Get the name of file from a file path.	
Syntax	FileGetName(Path)	
Returns	String	
Parameters		
Name	Optional	Description
Path	false	The path to the file.
Examples		
FileGetName("C:\ssis\test.xml")		

FileGetNameWithoutExtension

FileGetNameWithoutExtension

Description	Get the extension of file from a file path.	
Syntax	FileGetNameWithoutExtension(Path)	
Returns	String	
Parameters		
Name	Optional	Description
Path	false	The path to the file.
Examples		
FileGetNameWithoutExtension("C:\ssis\test.xml")		

FileGetRandomFileName

Description	Get a random file name without a path
Syntax	FileGetRandomFileName( )
Returns	String
Examples	
FileGetRandomFileName()	

FileGetTempFileName

Description	Get the name of a temporary file with a path to the temporary directory.
Syntax	FileGetTempFileName( )

FileGetTempFileName

Returns	String
Examples	
FileGetTempFileName()	

FileGetTempPath

Description	Get the path to the temporary files directory.
Syntax	FileGetTempPath( )
Returns	String
Examples	
FileGetTempPath()	

FileHasExtension

Description	Determine whether a file has an extension.	
Syntax	FileHasExtension(Path)	
Returns	Boolean	
Parameters		
Name	Optional	Description
Path	false	The path to the file.
Examples		
FileHasExtension("C:\ssis\test"), FileHasExtension("C:\ssis\test.xml")		

FileInUse

Description	Check if the file is in use for the path supplied.	
Syntax	FileInUse(Path)	
Returns	Boolean. True if the file is in use. False if it is not in use or path parameter is null	
Parameters		
Name	Optional	Description
Path	false	The path where the file is located.
Examples		
FileInUse("C:\ssis\test.xml")		

FileIsPathRooted

Description	Gets a value indicating whether the specified path string contains a root.	
Syntax	FileIsPathRooted(Path)	
Returns	Boolean	
Parameters		
Name	Optional	Description
path	false	The path to the file.
Examples		
FileIsPathRooted("C:\ssis\test"), FileIsPathRooted("test.xml")		

FileMove

FileMove

Description	Move a file from one location to another.	
Syntax	FileMove(Source, Destination)	
Returns	Boolean. If the move succeeds. Null of either parameter is null or either the source file doesn't exist.	
Parameters		
Name	Optional	Description
Source	false	The source path file path.
Destination	false	The destination file path.
Examples		
FileMove("C:\ssis\test.xml", "C:\ssis\processedfiles\test.xml")		

FileOrDirectoryGetPathRoot

Description	Get the root path to a file.	
Syntax	FileOrDirectoryGetPathRoot(Path)	
Returns	String	
Parameters		
Name	Optional	Description
Path	false	The path to the file.
Examples		
FileOrDirectoryGetPathRoot("C:\ssis\test.xml"), FileOrDirectoryGetPathRoot("C:\ssis\test\")		

FileReadAllBytes

Description	Return the bytes from a specified file.	
Syntax	FileReadAllBytes(Path)	
Returns	Byte Array. If the path does not exist, null is returned.	
Parameters		
Name	Optional	Description
Path	false	The path where the file is located.
Examples		
FileReadAllBytes("C:\ssis\test.xml")		

FileReadAllText

Description	Return the text from a specified file.	
Syntax	FileReadAllText(Path)	
Returns	String. If the path does not exist, null is returned.	
Parameters		
Name	Optional	Description
Path	false	The path where the file is located.
Examples		
FileReadAllText("C:\ssis\test.xml")		

FileReplace

FileReplace

Description	Copy a file from one location to another and create a backup of the destination .	
Syntax	FileReplace(Source, Destination, Backup_Destination)	
Returns	Boolean. If the copy succeeds. Null of either parameter is null or either the source file doesn't exist.	
Parameters		
Name	Optional	Description
Source	false	The source path file path.
Destination	false	The destination file path.
Backup_Destination	false	The path where the backup file will be created.
Examples		
FileReplace("C:\ssis\test.xml", "C:\ssis\processedfiles\test.xml", "C:\ssis\processedfiles\backup\test.xml")		

FileSetCreationTime

Description	Set the time the file was created.	
Syntax	FileSetCreationTime(Path, Date_Created)	
Returns	Boolean. True if the value is set. False if either parameter is null.	
Parameters		
Name	Optional	Description
Path	false	The path where the file is located.
Date_Created	false	The date the file was created.
Examples		

FileSetCreationTime

FileSetCreationTime("C:\ssis\test.xml", "7/6/2012 14:24")

FileSetLastAccessTime

Description	Set the time the file was last accessed.	
Syntax	FileSetLastAccessTime(Path, Date_Last_Access)	
Returns	Boolean. True if the value is set. False if either parameter is null.	
Parameters		
Name	Optional	Description
Path	false	The path where the file is located.
Date_Last_Access	false	The date the file was last accessed.
Examples		
FileSetLastAccessTime("C:\ssis\test.xml", "7/6/2012 14:24")		

FileSetLastWriteTime

Description	Set the time the file was written to accessed.	
Syntax	FileSetLastWriteTime(Path, Date_Last_Written)	
Returns	Boolean. True if the value is set. False if either parameter is null.	
Parameters		
Name	Optional	Description

FileSetLastWriteTime

Path	false	The path where the file is located.
Date_Last_Written	false	The date the file was last written to.
Examples		
FileSetLastWriteTime("C:\ssis\test.xml", "7/6/2012 14:24")		

FileWriteAllBytes

Description	Write bytes to the file contents.	
Syntax	FileWriteAllBytes(Path, Bytes [, Encoding])	
Returns	Boolean. If the path does not exist, null is returned.	
Parameters		
Name	Optional	Description
Path	false	The path where the file is located.
Bytes	false	Bytes to write to the file.
Encoding	true	Encoding to use to write to file. Choices are ASCII, UNICODE, UTF7, UTF8, UTF32 or Default.
Examples		
FileWriteAllBytes("C:\ssis\test.xml", varBytes, "UTF8")		

FileWriteAllText

FileWriteAllText

Description	Write text to the file contents.	
Syntax	FileWriteAllText(Path, Text [, Encoding])	
Returns	Boolean. If the path does not exist, null is returned.	
Parameters		
Name	Optional	Description
Path	false	The path where the file is located.
Text	false	Text to write to the file.
Encoding	true	Encoding to use to write to file. Choices are ASCII, UNICODE, UTF7, UTF8, UTF32 or Default.
Examples		
FileWriteAllText("C:\ssis\test.xml", "Wrox Professional SSIS 2012")		

First

Description	Aggregation that returns the first value of the group. Can only be used as part of the Advanced Aggregation component	
Syntax	First(Value [, Condition])	
Returns	Object.	
Parameters		
Name	Optional	Description
Value	false	The value to add to the aggregation

First		
Condition	false	The condition allows you to conditionally add values to the aggregation based on the expression. For instance, you could define the aggregate as First(SalesTotal, CustomerState="FL") and this would only add the value to the aggregation if the customers state was florida.
Examples		
First(SalesTotal), First(SalesTotal, CustomerState="FL")		

FirstDate		
Description	Returns the first day of a specified month.	
Syntax	FirstDate(), FirstDate(Date)	
Returns	returns date	
Parameters		
Name	Optional	Description
Date	false	Date to retrieve value from
Examples		
FirstDate(), FirstDate("12/8/2011"), FirstDate(varDate), FirstDate(OrderDateColumn)		

FirstDateOfNextMonth	
Description	Returns the first day of a next month.
Syntax	FirstDateOfNextMonth(), FirstDateOfNextMonth(Date)

FirstDateOfNextMonth		
Returns	returns date	
Parameters		
Name	Optional	Description
Date	false	Date to retrieve value from
Examples		
FirstDateOfNextMonth(), FirstDateOfNextMonth("12/8/2011"), FirstDateOfNextMonth(varDate), FirstDateOfNextMonth(OrderDateColumn)		

Floor		
Description	Returns the largest integer less than or equal to the numeric value you pass to this function. For example, if you pass 3.14 to FLOOR, the function returns 3. If you pass 3.98 to FLOOR, the function returns 3. Likewise, if you pass -3.17 to FLOOR, the function returns -4.	
Syntax	FLOOR(Numeric_Value)	
Returns	Integer if you pass a numeric value with declared precision between 0 and 28. Double if you pass a numeric value with declared precision greater than 28. NULL if a value passed to the function is NULL.	
Parameters		
Name	Optional	Description
Numeric_Value	false	Numeric datatype. You can enter any valid task editor as long as it evaluates to numeric data.
Examples		
example goes here		

FormatCurrency

Description	Returns an expression formatted as currency	
Syntax	FormatCurrency(Expression [, Digits_After_Decimal] [, Include_Leading_Digit] [, Use_Parens_For_Negative_Numbers] [, Group_Digits])	
Returns	String. Returns an expression formatted as a currency.	
Parameters		
Name	Optional	Description
Expression	false	Expression to be formatted.
Digits_After_Decimal	false	Numeric value indicating how many places are displayed to the right of the decimal. The default value is –1, which indicates that the computer's regional settings are used.
Include_Leading_Digit	false	Indicates whether a leading 0 is displayed for fractional values. Values are 0 for False, -1 for True
Use_Parens_For_Negative_Numbers	false	Indicates whether to place negative values within parentheses. Values are 0 for False, -1 for True
Group_Digits	false	Indicates whether or not numbers are grouped using the group delimiter specified in the locale settings. Values are 0 for False, -1 for True
Examples		
FormatCurrency("4589.932", 2)		

FormatDateTime

Description	Returns an expression formatted as a date

FormatDateTime

Syntax	FormatDateTime(Expression [, Date_Format])	
Returns	String. Returns an expression formatted as a date.	
Parameters		
Name	Optional	Description
Expression	false	Date expression to be formatted.
Date_Format	false	Numeric value that indicates the date/time format used. If omitted, DateFormat.GeneralDate is used. Values are General Date = 0 Long Date = 1 Short Date = 2 Long Time = 3 Short Time = 4
Examples		
FormatDateTime(GetDate(), 2)		

FormatNumber

Description	Returns an expression formatted as a number	
Syntax	FormatNumber(Expression [, Digits_After_Decimal] [, Include_Leading_Digit] [, Use_Parens_For_Negative_Numbers] [, Group_Digits])	
Returns	String. Returns an expression formatted as a number.	
Parameters		
Name	Optional	Description
Expression	false	Expression to be formatted.

FormatNumber		
Digits_After_Decimal	false	Numeric value indicating how many places are displayed to the right of the decimal. The default value is –1, which indicates that the computer's regional settings are used.
Include_Leading_Digit	false	Indicates whether a leading 0 is displayed for fractional values. Values are 0 for False, -1 for True
Use_Parens_For_Negative_Numbers	false	Indicates whether to place negative values within parentheses. Values are 0 for False, -1 for True
Group_Digits	false	Indicates whether or not numbers are grouped using the group delimiter specified in the locale settings. Values are 0 for False, -1 for True
Examples		
FormatNumber("4589.932", 2)		

FormatPercent		
Description	Returns an expression formatted as a percentage.	
Syntax	FormatPercent(Expression [, Digits_After_Decimal] [, Include_Leading_Digit] [, Use_Parens_For_Negative_Numbers] [, Group_Digits])	
Returns	String. Returns an expression formatted as a percentage.	
Parameters		
Name	Optional	Description
Expression	false	Expression to be formatted.

FormatPercent		
Digits_After_Decimal	false	Numeric value indicating how many places are displayed to the right of the decimal. The default value is –1, which indicates that the computer's regional settings are used.
Include_Leading_Digit	false	Indicates whether a leading 0 is displayed for fractional values. Values are 0 for False, -1 for True
Use_Parens_For_Negative_Numbers	false	Indicates whether to place negative values within parentheses. Values are 0 for False, -1 for True
Group_Digits	false	Indicates whether or not numbers are grouped using the group delimiter specified in the locale settings. Values are 0 for False, -1 for True
Examples		
FormatPercent("4589.932", 2)		

FV		
Description	Returns the future value of an investment, where you make periodic, constant payments and the investment earns a constant interest rate.	
Syntax	FV(Rate, Terms, Payment [, Present_Value] [, Type])	
Returns	Numeric.	
Parameters		
Name	Optional	Description

FV		
Rate	false	Numeric. Interest rate earned in each period. Expressed as a decimal number. Divide the percent rate by 100 to express it as a decimal number. Must be greater than or equal to 0.
Terms	false	Numeric. Number of periods or payments. Must be greater than 0.
Payment	false	Numeric. Payment amount due per period. Must be a negative number.
Present_Value	true	Numeric. Current value of the investment. If you omit this argument, FV uses 0.
Type	true	Integer. Timing of the payment. Enter 1 if payment is at the beginning of period. Enter 0 if payment is at the end of period. Default is 0. If you enter a value other than 0 or 1, the Integration Service treats the value as 1.
Examples		
FV(Ratevar, Termvar, Paymentvar), FV(3.25, 36, 125.50, 1200, 1)		

Task Factory Functions List G-H

Last Modified on 02 October 2020

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

GetChar

Description	GetChar returns the character at a specified index in a string.	
Syntax	GetChar(String_Value, Index)	
Returns	Single character. NULL if a value passed to the function is NULL.	
Parameters		
Name	Optional	Description
String_Value	false	Character string. The string value to return a character from.
Index	false	Character string. The index of the character you want returned from the String_Value parameter .
Examples		
GetChar("GetTheC", 7), If GetChar(AddressColumn, 1) = "A" Then ...		

GetConnectionProp

Description	Get a property from a connection manager.
Syntax	GetConnectionProp(Connection_Manager, Property_Name)
Returns	String.
Parameters	

GetConnectionProp

Name	Optional	Description
Connection_Manager	false	Connection manager to retrieve a property from. Must use the @@[connection_manager_name] syntax.
Property_Name	false	String value. The property to retrieve from the connection manager.
Examples		
GetConnectionProp(@@[localhost.AdventureWorks], "ServerName")		

GetDate

Description	Returns the current date and time.
Syntax	GetDate()
Returns	Date
Examples	
GetDate()	

GetDatePart

Description	Returns the part of the date specified in the datepart parameter.
Syntax	GetDatePart(Date, Date_Part)
Returns	Object.
Parameters	

GetDatePart

Name	Optional	Description
Date	false	The date used to retrieve the datepart from.
Date_Part	false	The datepart tells the DatePart what part of the date to retrieve. The datepart tells the DatePart what part of the date to retrieve. Year = "Y", "YY", "YYY", "YYYY" Day = "D", "DD", "DDD", "DY", "DAY" Month = "MM", "MON", "MONTH" Hour = "HH", "HH12", "HH24" Minute = "MI" Second = "SS"" Millisecond = "MS" Microsecond = ""US" Day Of Week (1-7) = "WD","W" Week Of Year = "WW" Day Of Year = "YD" Quarter = "Q" Julian Date = "J"
Examples		
GetDatePart("7/4/2012", "d"), GetDatePart(varStartDate, "MM"),GetDatePart(OrderDateColumn, "HH")		

GetDiskFreeSpace

Description	Returns the amount of free disk space in megabytes.	
Syntax	GetDiskFreeSpace(Drive_Letter)	
Returns	Integer. The amount of free disk space in megabytes	
Parameters		
Name	Optional	Description

GetDiskFreeSpace

Drive_Letter	false	The drive letter to determine disk space. Must include :\ with the drive letter (e.g. C:\).
Examples		
GetDiskFreeSpace("C:\")		

GetDiskSize

Description	Returns the amount of disk space for a drive in megabytes.	
Syntax	GetDiskSize(Drive_Letter)	
Returns	Integer. The amount of disk space in megabytes	
Parameters		
Name	Optional	Description
Drive_Letter	false	The drive letter to determine disk space. Must include :\ with the drive letter (e.g. C:\).
Examples		
GetDiskSize("C:\")		

GetUtcDate

Description	Returns the current utc date and time.
Syntax	GetUtcDate()
Returns	Date
Examples	

GetUtcDate

GetUtcDate()

Greatest

Description	Returns the greatest value from a list of input values. Use this function to return the greatest string, date, or number. By default, the match is case sensitive.	
Syntax	GREATEST(Case_Flag, Value1 [, Value2, ..., ValueN,])	
Returns	value1 if it is the greatest of the input values, value2 if it is the greatest of the input values, and so on. NULL if any of the arguments is null.	
Parameters		
Name	Optional	Description
Value	false	Any datatype except Binary. Datatype must be compatible with other values. Value you want to compare against other values. You must enter at least one value argument. If the value is numeric, and other input values are numeric, all values use the highest precision possible. For example, if some values are Integer datatype and others are Double datatype, the Integration Service converts the values to Double.
Case_Flag	false	Must be an integer. Determines whether the arguments in this function are case sensitive. You can enter any valid task editor. When CaseFlag is a number other than 0, the function is case sensitive. When CaseFlag is a null value or 0, the function is not case sensitive.
Examples		

Greatest

Greatest(false, varValues), Greatest(false, "test1", "test2", "test3")

GroupNum

Description	Returns the group of which this aggregate row belongs to. For instance, if there were 10 rows outputted from the Advanced Aggregate, then GroupNum would add 1-10 for each respective row
Syntax	GroupNum()
Returns	Integer
Examples	
GroupNum()	

GroupRowNum

Description	Returns the number of rows that were used in this aggregation group.
Syntax	GroupRowNum()
Returns	Integer
Examples	
GroupRowNum()	

Hex

Hex

Description	Convert a value to Hex value.	
Syntax	Hex(Value)	
Returns	String. Returns the Hex value of the integer.	
Parameters		
Name	Optional	Description
Value	false	The integer value to convert to Hex.
Examples		
Hex(18), Hex(varValue)		

HexToBin

Description	Converts a hexical value to a binary value.	
Syntax	HexToBin(Value)	
Returns	String	
Parameters		
Name	Optional	Description
Value	false	The hexadecimal value to convert.
Examples		
HexToBin(10)		

HexToDecimal

HexToDecimal

Description	Converts a hexical value to a decimal value.	
Syntax	HexToDecimal(Value)	
Returns	Int64	
Parameters		
Name	Optional	Description
Value	false	The hexadecimal value to convert.
Examples		
HexToDecimal(10)		

HexToOct

Description	Converts a hexical value to a octal value.	
Syntax	HexToOct(Value)	
Returns	String	
Parameters		
Name	Optional	Description
Value	false	The hexadecimal value to convert.
Examples		
HexToOct(10)		

Hour

Hour

Description	Returns the hour from the date specified.	
Syntax	Hour(Date)	
Returns	returns integer value of 1 - 24	
Parameters		
Name	Optional	Description
Date	false	Date to retrieve Hour from
Examples		
Hour("12/8/2011"), Hour(varDate), Hour(OrderDateColumn)		

Task Factory Functions List I-J

Last Modified on 02 October 2020

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

IfNull

Description	Tests for a null value and, if it exists, returns a default value.	
Syntax	IfNull(Value, defaultValue)	
Returns	defaultValue	
Parameters		
Name	Optional	Description
Value	false	The column/variable/string to test for NULL value(s).
defaultValue	false	The value returned when NULL is present
Examples		
IfNull(PhoneNumber, 'Unknown'), IfNull(IntRate, @[User::dailyRate])		

IfNullOrEmpty

Description	Tests for a null or empty value and, if it exists, returns a default value.	
Syntax	IfNullOrEmpty(Value, defaultValue)	
Returns	defaultValue	
Parameters		
Name	Optional	Description

IfNullOrEmpty		
Value	false	The column/variable/string to test for NULL or empty value(s).
defaultValue	false	The value returned when NULL is present
Examples		
IfNullOrEmpty(PhoneNumber, 'Unknown'), IfNullOrEmpty(IntRate, @[User::dailyRate])		

IIF		
Description	Returns one of two values you specify, based on the results of a condition.	
Syntax	IIF(Condition, Value1 , Value2)	
Returns	value1 if the condition is TRUE. value2 if the condition is FALSE.	
Parameters		
Name	Optional	Description
Condition	false	The condition you want to evaluate. You can enter any valid task editor that evaluates to TRUE or FALSE.
Value1	false	Any datatype except Binary. The value you want to return if the condition is TRUE. The return value is always the datatype specified by this argument. You can enter any valid task editor, including another IIF expression.
Value2	true	Any datatype except Binary. The value you want to return if the condition is FALSE. You can enter any valid task editor, including another IIF expression.

IIF

Examples

IIF(SALES > 100, EMP_NAME)

SALES	EMP_NAME	RETURN VALUE
150	John Smith	John Smith
50	Pierre Bleu	'' (empty string)
120	Sally Green	Sally Green
NULL	Greg Jones	'' (empty string)

IN

Description	Searches value in the given list. If value is found in the list then it will return True else False. You can perform case-sensetive or case-insensetive compare based on last (optional) argument.
Syntax	IN(Value_To_Find, value1 [, value2, value3..., valueN] [, caseFlag])
Returns	Boolean - True if the value is found, False is the value is not found

Parameters

Name	Optional	Description
Value_To_Find	false	The value you want to search for in the array.
List_Of_Values	false	List of values to be compared.

IN		
CaseFlag	true	(Optional) This flag specifies how to compare string data. If this flag is 0 then it will do case-insensetive compare. Anything other than 0 will be case-sensetive compare. By default it will do case-sensetive compare means "DEC" is not same as "Dec"
Examples		
In(MONTHNAME("12/12/2013",True), "JAN",FEB","DEC") --This should return False. Last argument not specified means case-sensetive compare. In(MONTHNAME("12/12/2013",True), "JAN",FEB","DEC",0) --This should return True. Last argument=0 means case-insensetive compare. In(MONTHNAME("12/12/2013",True), "JAN",FEB","DEC",1) --This should return False. Last argument=1 means case-insensetive compare.		

InitCap		
Description	Capitalizes the first letter in each word of a string and converts all other letters to lowercase.	
Syntax	InitCap(String)	
Returns	String. If the data contains multibyte characters, the return value depends on the code page and data movement mode of the Integration Service. NULL if a value passed to the function is NULL.	
Parameters		
Name	Optional	Description
String	false	Any datatype except Binary. You can enter any valid task editor.
Value1	false	para2 desc.
Examples		
InitCap("a fox runs over the river"), InitCap(FirstNameColumn), InitCap(FirstNameColumn + " " + LastNameColumn)		

InsertChars

Description	Inserts characters into a string at a defined index	
Syntax	InsertChars(Base_String, Index, String_To_Insert)	
Returns	String. NULL if a value passed to the function is NULL.	
Parameters		
Name	Optional	Description
Base_String	false	The base string in which characters will be inserted.
Index	false	The index of the Base_String where String_To_Insert will be inserted
String_To_Insert	false	The string to insert at the index.
Examples		
InsertChar("400 Dr", 4, "College ") will return "400 College Dr", InsertChar(FullNameColumn, InStr(FullName, " "), "-")		

InStr

Description	Returns the position of a character set in a string, counting from left to right.	
Syntax	INSTR(String_To_Search, Search_Value [, Start] [, Occurrence] [, Comparison_Type])	
Returns	Integer if the search is successful. Integer represents the position of the first character in the search_value, counting from left to right. 0 if the search is unsuccessful. NULL if a value passed to the function is NULL.	
Parameters		
Name	Optional	Description

InStr		
String_To_Search	false	The string must be a character string. Passes the value you want to evaluate. You can enter any valid transformation expression. The results of the expression must be a character string. If not, INSTR converts the value to a string before evaluating it.
Search_Value	false	Any value. The search value is case sensitive. The set of characters you want to search for. The Search_Value must match a part of the string. For example, if you write INSTR('Alfred Pope', 'Alfred Smith') the function returns 0. You can enter any valid transformation expression. If you want to search for a character string, enclose the characters you want to search for in single quotation marks, for example 'abc'.
Start	true	Must be an integer value. The position in the string where you want to start the search. You can enter any valid transformation expression. The default is 1, meaning that INSTR starts the search at the first character in the string. If the start position is 0, INSTR searches from the first character in the string. If the start position is a positive number, INSTR locates the start position by counting from the beginning of the string. If the start position is a negative number, INSTR locates the start position by counting from the end of the string. If you omit this argument, the function uses the default value of 1.

InStr		
Occurrence	true	A positive integer greater than 0. You can enter any valid transformation expression. If the search value appears more than once in the string, you can specify which occurrence you want to search for. For example, you would enter 2 to search for the second occurrence from the start position. If you omit this argument, the function uses the default value of 1, meaning that INSTR searches for the first occurrence of the search value. If you pass a decimal, the Integration Service rounds it to the nearest integer value. If you pass a negative integer or 0, the dataflow fails.
Comparison_Type	true	The string comparison type, either linguistic or binary, when the Integration Service runs in Unicode mode. When the Integration Service runs in ASCII mode, the comparison type is always binary. Linguistic comparisons take language-specific collation rules into account, while binary comparisons perform bitwise matching. For example, the German sharp s character matches the string "ss" in a linguistic comparison, but not in a binary comparison. Binary comparisons run faster than linguistic comparisons. Must be an integer value, either 0 or 1: - 0: INSTR performs a linguistic string comparison. - 1: INSTR performs a binary string comparison. Default is 0.
Examples		
INSTR("Is four in this string", "four"), InStr(ProductDescriptionColumn, "Blue"), InStr(ProductDescriptionColumn, "Blue", 0, 2, 1)		

IsDate

Description	Determines whether the value is a date.	
Syntax	IsDate(Value)	
Returns	Boolean	
Parameters		
Name	Optional	Description
Value	false	The value to check.
Examples		
IsDate("7/7/2012")		

IsEmpty

Description	Determines whether the value is empty.	
Syntax	IsEmpty(Value)	
Returns	Boolean	
Parameters		
Name	Optional	Description
Value	false	The value to check.
Examples		
IsEmpty(AddressColumn)		

IsNull

IsNull

Description	Determines whether the value is null.	
Syntax	IsNull(Value)	
Returns	Boolean	
Parameters		
Name	Optional	Description
Value	false	The value to check.
Examples		
IsNull(AddressColumn)		

IsNullOrEmpty

Description	Determines whether the value is null or empty.	
Syntax	IsNullOrEmpty(Value)	
Returns	Boolean	
Parameters		
Name	Optional	Description
Value	false	The value to check.
Examples		
IsNullOrEmpty(AddressColumn)		

IsNumber

IsNumber		
Description	Determines whether the value is a number.	
Syntax	IsNumber(Value)	
Returns	Boolean	
Parameters		
Name	Optional	Description
Value	false	The value to check.
Examples		
IsNumber("12345")		

IsSpaces		
Description	Determines whether the value is all spaces.	
Syntax	IsSpaces(Value)	
Returns	Boolean	
Parameters		
Name	Optional	Description
Value	false	The value to check.
Examples		
IsSpaces("test")		

Task Factory Functions List K-L

Last Modified on 02 October 2020

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Last

Description	Aggregation that returns the last value of the group. Can only be used as part of the Advanced Aggregation component	
Syntax	Last(Value [, Condition])	
Returns	Object.	
Parameters		
Name	Optional	Description
Value	false	The value to add to the aggregation
Condition	false	The condition allows you to conditionally add values to the aggregation based on the expression. For instance, you could define the aggregate as Last(SalesTotal, CustomerState="FL") and this would only add the value to the aggregation if the customers state was florida.
Examples		
Last(SalesTotal), Last(SalesTotal, CustomerState="FL")		

LastDate

Description	Returns the last day of a specified month.

LastDate

Syntax	LastDate(), LastDate(Date)	
Returns	returns date	
Parameters		
Name	Optional	Description
Date	false	Date to retrieve value from
Examples		
LastDate(), LastDate("12/8/2011"), LastDate(varDate), LastDate(OrderDateColumn)		

LastDateOfPrevMonth

Description	Returns the last day of the previous month.	
Syntax	LastDateOfPrevMonth(), LastDateOfPrevMonth(Date)	
Returns	returns date	
Parameters		
Name	Optional	Description
Date	false	Date to retrieve value from
Examples		
LastDateOfPrevMonth(), LastDateOfPrevMonth("12/8/2011"), LastDateOfPrevMonth(varDate), LastDateOfPrevMonth(OrderDateColumn)		

Least

Description	Returns the smallest value from a list of input values. By default, the match is not case sensitive.

Least

Syntax	LEAST(Case_Flag, Value1 [, Value2, ..., ValueN,])	
Returns	value1 if it is the smallest of the input values, value2 if it is the smallest of the input values, and so on. NULL if any of the arguments is null.	
Parameters		
Name	Optional	Description
CaseFlag	false	Determines whether the arguments in this function are case sensitive. You can enter any valid transformation expression. When CaseFlag is a number other than 0, the function is case sensitive. When CaseFlag is a null value or 0, the function is not case sensitive.
Value	false	Any datatype except Binary. Datatype must be compatible with other values. Value you want to compare against other values. You must enter at least one value argument. If the value is Numeric, and other input values are of other numeric datatypes, all values use the highest precision possible. For example, if some values are of the Integer datatype and others are of the Double datatype, the Integration Service converts the values to Double.
Examples		
Least(false, varValues), Least(false, "test1", "test2", "test3")		

Left

Left

Description	Returns a string containing a specified number of characters from the left side of a string..	
Syntax	Left(String, Number_Of_Characters)	
Returns	String. NULL if a value passed to the function is NULL.	
Parameters		
Name	Optional	Description
String	false	String datatype. String expression from which the leftmost characters are returned.
Number_Of_Characters	false	The number of characters to return from the string.
Examples		
Left("Stop Here. None of this will be there", 10), Left(FirstNameColumn, 20)		

Length

Description	Returns the number of characters in a string, including trailing blanks.	
Syntax	Length(String)	
Returns	Integer representing the length of the string. NULL if a value passed to the function is NULL.	
Parameters		
Name	Optional	Description
String	false	String datatype. The strings you want to evaluate. You can enter any valid transformation expression.

Length
Examples
Length("How Long Is This?"), Length(FirstNameColumn)

Log		
Description	Returns the logarithm of a numeric value. Most often, you use this function to analyze scientific data rather than business data.	
Syntax	LOG(Base, Exponent)	
Returns	Double value. NULL if a value passed to the function is NULL.	
Parameters		
Name	Optional	Description
Base	false	The base of the logarithm. Must be a positive numeric value other than 0 or 1. Any valid task editor that evaluates to a positive number other than 0 or 1.
Exponent	false	The exponent of the logarithm. Must be a positive numeric value greater than 0. Any valid task editor that evaluates to a positive number greater than 0.
Examples		
Log(BaseColumn, ExponentColumn)		

Log10	
Description	Returns the base 10 logarithm of a specified number.

Log10		
Syntax	Log10(Numeric_Value)	
Returns	Numeric	
Parameters		
Name	Optional	Description
Numeric_Value	false	Passes the values for which you want to the Atan of
Examples		
Log10(56)		

LogBufferRowToFile		
Description	Logs a row of data from a dataflow to a flat file.	
Syntax	LogBufferRowToFile(File_Path [, Delimeter] [, Quoted_String])	
Returns	Nothing	
Parameters		
Name	Optional	Description
File_Path	false	The path to the file to log the buffer row to.
Delimeter	optional	The file delimeter to be used to seperate the column values. Default is ","
Quoted_String	optional	Tells the engine whether to wrap string values in quotes or not. Default is false.
Examples		
LogBufferRowToFile("C:\temp\logfile.ext"), LogBufferRowToFile("c:\temp\logfile.txt", "|", True)]		

LogError

Description	Logs an error message to the ssis output log (progress tab in BIDS)	
Syntax	LogError(Message [, Condition])	
Returns	Nothing	
Parameters		
Name	Optional	Description
Message	false	The message to log.
Condition	true	The condition to evaluate to true before logging the message
Examples		
LogError("This is a error"), LogError("Values didn't match", value1!=value2)]		

LogInfo

Description	Logs an informational message to the ssis output log (progress tab in BIDS)	
Syntax	LogInfo(Message [, Condition])	
Returns	Nothing	
Parameters		
Name	Optional	Description
Message	false	The message to log.
Condition	true	The condition to evaluate to true before logging the message
Examples		

LogInfo

LogInfo("This is a message"), LogInfo("Values didn't match", value1!=value2)]

LogWarning

Description	Logs a warning message to the ssis output log (progress tab in BIDS)	
Syntax	LogWarning(Message [, Condition])	
Returns	Nothing	
Parameters		
Name	Optional	Description
Message	false	The message to log.
Condition	true	The condition to evaluate to true before logging the message
Examples		
LogWarning("This is a warning"), LogWarning("Values didn't match", value1!=value2)]		

LookupData

Description	LookupData will use an Advanced Lookup Cache Connection Manager to retrieve data from the cached connection manager. LookupData will return the value of the first output column defined on a cache connection manager. The args parameter array is used to pass in values that match the "Input Parameters" defined on a cache connection manager.
Syntax	LookupData(Cache_Connection_Manager, Arg1 [, Arg2, ..., ArgN])
Returns	returns the value of the output column if a match is found. Returns NULL if no match is found.

LookupData

Parameters

Name	Optional	Description
Cache_Connection_Manager	false	The Advanced Lookup Cache Connection Manager where the lookup will be performanced. Must use the @@[ConnectionManager] syntax.
Args	false	Args is a parameter array of values that directly correlate to the input parameters defined on a cache connection manager.

Examples

LookupData(@@[ProductCache], ProductID)] - In this example, the cache connection manager (@@[ProductCache]) is storing the products. ProductID is a column that is part of the current component being used (Advanced Derived Column, Advanced Aggregator, etc) and the value of ProductID is passed into LookupData from the Data Flow.
[LookupData(@@[ProductCache], ProductID, OrderDate)] - In this example, the cache connection manager (@@[ProductCache]) is storing the products. ProductID and OrderDate are columns that are part of the current component being used (Advanced Derived Column, Advanced Aggregator, etc) and the value of them is passed into LookupData from the Data Flow.

LookupDataByColumn

Description	LookupDataByColumn will use an Advanced Lookup Cache Connection Manager to retrieve data from the cached connection manager and return a specified. Unlike LookupData, which returns the value of the first output column defined on a cache connection manager, LookupDataByColumn can be used to grab a specific column. The args parameter array is used to pass in values that match the "Input Parameters" defined on a cache connection manager.
Syntax	LookupDataByColumn(Cache_Connection_Manager, Column_To_Retrieve, Arg1 [, Arg2, ..., ArgN])
Returns	returns the value of the output column if a match is found. Returns NULL if no match is found.

Parameters

LookupDataByColumn

Name	Optional	Description
Cache_Connection_Manager	false	The Advanced Lookup Cache Connection Manager where the lookup will be performanced. Must use the @@[ConnectionManager] syntax.
Column_To_Retrieve	false	The name of an column defined as "Output" on the Advanced Lookup Cache Connection Manager.
Args	false	Args is a parameter array of values that directly correlate to the input parameters defined on a cache connection manager.

Examples

LookupData(@@[ProductCache], "ProductName", ProductID)] - In this example, the cache connection manager (@@[ProductCache]) is storing the products. ProductID is a column that is part of the current component being used (Advanced Derived Column, Advanced Aggregator, etc) and the value of ProductID is passed into LookupData from the Data Flow. "ProductName" is the column being retrieved from the cache connection manager.
[LookupData(@@[ProductCache], "ProductCost", ProductID, OrderDate)] - In this example, the cache connection manager (@@[ProductCache]) is storing the products. ProductID and OrderDate are columns that are part of the current component being used (Advanced Derived Column, Advanced Aggregator, etc) and the value of them is passed into LookupData from the Data Flow. "ProductCost" is the column being retrieved from the cache connection manager.

Lower

Description	Converts uppercase string characters to lowercase.
Syntax	LOWER(String)
Returns	Lowercase character string. If the data contains multibyte characters, the return value depends on the code page and data movement mode of the Integration Service. NULL if a value in the selected port is NULL.

Lower

Parameters		
Name	Optional	Description
String	false	Any string value. The argument passes the string values that you want to return as lowercase. You can enter any valid transformation expression that evaluates to a string.
Examples		
Lower("Lower All THIS"), Lower(LastNameColumn)		

LPad

Description	Adds a set of blanks or characters to the beginning of a string to set the string to a specified length.
Syntax	LPad(First_String, Length [, Second_String])
Returns	String of the specified length. NULL if a value passed to the function is NULL or if length is a negative number.

Parameters		
Name	Optional	Description
First_String	false	Can be a character string. The strings you want to change. You can enter any valid task editor.
Length	false	Must be a positive integer literal. This argument specifies the length you want each string to be.

LPad		
Second_String	true	Can be any string value. The characters you want to append to the left-side of the first_string values. You can enter any valid task editor. You can enter a specific string literal. However, enclose the characters you want to add to the beginning of the string within single quotation marks, as in 'abc'. This argument is case sensitive. If you omit the second_string, the function pads the beginning of the first string with blanks.
Examples		
LPad("SentryOne", 30), LPad(CompanyNameColumn, 50)		

LTrim		
Description	Removes blanks or characters from the beginning of a string.	
Syntax	LTRIM(String [, Trim_Set])	
Returns	String. The string values with the specified characters in the trim_set argument removed. NULL if a value passed to the function is NULL. If the trim_set is NULL, the function returns NULL.	
Parameters		
Name	Optional	Description
String	false	Any string value. Passes the strings you want to modify. You can enter any valid task editor. Use operators to perform comparisons or concatenate strings before removing characters from the beginning of a string.

LTrim		
Trim_Set	true	Any string value. Passes the characters you want to remove from the beginning of the first string. You can enter any valid task editor. You can also enter a character string. However, you must enclose the characters you want to remove from the beginning of the string within single quotation marks, for example, 'abc'. If you omit the second string, the function removes any blanks from the beginning of the string. LTRIM is case sensitive. For example, if you want to remove the 'A' character from the string 'Alfredo', you would enter 'A', not 'a'.
Examples		
LTrim(" SentryOne"), LTrim(CompanyNameColumn), LTrim("----SentryOne", "-")		

Task Factory Functions List M-N

Last Modified on 02 October 2020

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

MakeDateTime

Description	Creates a date based on the parameters passed into the function.	
Syntax	MakeDateTime(Year, Month, Day [, Hour] [, Minute] [, Second] [, Millisecond])	
Returns	returns date	
Parameters		
Name	Optional	Description
Year	false	Year of the date being created
Month	false	Month of the date being created
Day	false	Day of the date being created
Hour	false	Hour of the date being created
Minute	false	Minute of the date being created
Second	false	Second of the date being created
Millisecond	false	Millisecond of the date being created
Examples		
MakeDateTime("2012", "07", "07"), MakeDateTime("2012", "07", "07", "12", "30", "45", "001")		

Max (Date)

Description	Aggregation that returns max date the group. Can only be used as part of the Advanced Aggregation component	
Syntax	Max(Value [, Condition])	
Returns	Date.	
Parameters		
Name	Optional	Description
Value	false	The value to add to the aggregation
Condition	false	The condition allows you to conditionally add values to the aggregation based on the expression. For instance, you could define the aggregate as Max(SalesDate, CustomerState="FL") and this would only add the value to the aggregation if the customers state was florida.
Examples		
Max(SalesDate), Max(SalesDate, CustomerState="FL")		

Max (Number)

Description	Aggregation that returns max number the group. Can only be used as part of the Advanced Aggregation component	
Syntax	Max(Value [, Condition])	
Returns	Integer.	
Parameters		
Name	Optional	Description

Max (Number)

Value	false	The value to add to the aggregation
Condition	false	The condition allows you to conditionally add values to the aggregation based on the expression. For instance, you could define the aggregate as Max(SalesTotal, CustomerState="FL") and this would only add the value to the aggregation if the customers state was florida.
Examples		
Max(SalesTotal), Max(SalesTotal, CustomerState="FL")		

Max (String)

Description	Aggregation that returns max string the group. The value is based on the ascii value of the string. Can only be used as part of the Advanced Aggregation component	
Syntax	Max(Value [, Condition])	
Returns	String	
Parameters		
Name	Optional	Description
Value	false	The value to add to the aggregation
Condition	false	The condition allows you to conditionally add values to the aggregation based on the expression. For instance, you could define the aggregate as Max(CustomerAlternateKey, CustomerState="FL") and this would only add the value to the aggregation if the customers state was florida.

Max (String)

Examples
Max(CustomerAlternateKey), Max(CustomerAlternateKey, CustomerState="FL")

MD5

Description	Calculates the checksum of the input value. The function uses Message-Digest algorithm 5 (MD5). MD5 is a one- way cryptographic hash function with a 128-bit hash value. You can conclude that input values are different when the checksums of the input values are different. Use MD5 to verify data integrity.	
Syntax	MD5(Value)	
Returns	Unique 32-character string of hexadecimal digits 0-9 and a-f. NULL if the input is a null value.	
Parameters		
Name	Optional	Description
Value	false	String or Binary datatype. Value for which you want to calculate checksum. The case of the input value affects the return value. For example, MD5(ssis) and MD5(SSIS) return different values.
Examples		
MD5("secure this")		

MessageBox

Description	Displays a MessageBox with a message (Only works in BIDS / SQL Server Data Tools). Used for debugging purposes during design time.
Syntax	MessageBox(Message [, Condition])

MessageBox

Returns	Nothing	
Parameters		
Name	Optional	Description
Message	false	The message to display.
Condition	true	The condition to evaluate to true before display the MessageBox
Examples		
MessageBox("This is a message"), MessageBox("Values didn't match", value1!=value2)]		

MilliSecond

Description	Returns the MilliSecond from the date specified.	
Syntax	MilliSecond(Date)	
Returns	returns integer value of 0 - 999	
Parameters		
Name	Optional	Description
Date	false	Date to retrieve MilliSecond from
Examples		
MilliSecond("12/8/2011"), MilliSecond(varDate), MilliSecond(OrderDateColumn)		

Min (Date)

Min (Date)

Description	Aggregation that returns Min date the group. Can only be used as part of the Advanced Aggregation component	
Syntax	Min(Value [, Condition])	
Returns	Date.	
Parameters		
Name	Optional	Description
Value	false	The value to add to the aggregation
Condition	false	The condition allows you to conditionally add values to the aggregation based on the expression. For instance, you could define the aggregate as Min(SalesDate, CustomerState="FL") and this would only add the value to the aggregation if the customers state was florida.
Examples		
Min(SalesDate), Min(SalesDate, CustomerState="FL")		

Min (Number)

Description	Aggregation that returns Min number the group. Can only be used as part of the Advanced Aggregation component	
Syntax	Min(Value [, Condition])	
Returns	Integer.	
Parameters		
Name	Optional	Description

Min (Number)		
Value	false	The value to add to the aggregation
Condition	false	The condition allows you to conditionally add values to the aggregation based on the expression. For instance, you could define the aggregate as Min(SalesTotal, CustomerState="FL") and this would only add the value to the aggregation if the customers state was florida.
Examples		
Min(SalesTotal), Min(SalesTotal, CustomerState="FL")		

Min (String)		
Description	Aggregation that returns Min string the group. The value is based on the ascii value of the string. Can only be used as part of the Advanced Aggregation component	
Syntax	Min(Value [, Condition])	
Returns	String	
Parameters		
Name	Optional	Description
Value	false	The value to add to the aggregation
Condition	false	The condition allows you to conditionally add values to the aggregation based on the expression. For instance, you could define the aggregate as Min(CustomerAlternateKey, CustomerState="FL") and this would only add the value to the aggregation if the customers state was florida.

Min (String)

Examples
Min(CustomerAlternateKey), Min(CustomerAlternateKey, CustomerState="FL")

Minute

Description	Returns the Minute from the date specified.	
Syntax	Minute(Date)	
Returns	returns integer value of 0 - 59	
Parameters		
Name	Optional	Description
Date	false	Date to retrieve Minute from
Examples		
Minute("12/8/2011"), Minute(varDate), Minute(OrderDateColumn)		

Mod

Description	Divides two numbers and returns the remainder.	
Syntax	Mod(Value1, Value2)	
Returns	Numeric	
Parameters		
Name	Optional	Description
Value1	false	The first number in the equation.

Mod		
Value2	false	The second number in the equation.
Examples		
Mod(10, 3)		

Month		
Description	Returns the Month from the date specified.	
Syntax	Month(Date)	
Returns	returns integer value of 1 - 12	
Parameters		
Name	Optional	Description
Date	false	Date to retrieve Month from
Examples		
Month("12/8/2011"), Month(varDate), Month(OrderDateColumn)		

MonthName		
Description	Returns the name of the Month (MonMonth, TuesMonth, etc) from the date specified.	
Syntax	MonthName(Date)	
Returns	returns string. Name of the Month	
Parameters		
Name	Optional	Description

MonthName

Date	false	Date to retrieve Month name from
return_shortname	false	Return the short name of the Month (Jan, Feb, Mar)
Examples		
MonthName("12/8/2011", false), MonthName(varDate, True), MonthName(OrderDateColumn, false)		

Now

Description	Returns the current date and time.
Syntax	Now()
Returns	Date
Examples	
Now()	

NullIf

Description	Compares two string values for equality and returns null if they match.	
Syntax	NullIf(String_Value1, String_Value2)	
Returns	Null if "String_Value1" or "String_Value2" is null. Returns value of "String_Value1" if "String_Value2" and "string_value2" do not match..	
Parameters		
Name	Optional	Description

NullIf		
String_Value1	false	Any string value. If the values do not match, this value is returned from the function.
String_Value2	false	Any string value.
Examples		
NullIf(CompanyNameColumn, ""), NullIf(AddressColumnLine2, "")		

NullIfEmpty		
Description	Checks a string value for an empty value. If the string value is empty a null value is returned.	
Syntax	NullIfEmpty(String_Value)	
Returns	Null if "String_Value" is empty. Returns value of "String_Value" if the value is not empty.	
Parameters		
Name	Optional	Description
String_Value	false	Any string value.
Examples		
NullIfEmpty(CompanyNameColumn), NullIfEmpty(varValue)		

Task Factory Functions List O-Q

Last Modified on 02 October 2020

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

OctToBin

Description	Converts a octal value to a binary value.	
Syntax	OctToBin(Value)	
Returns	String	
Parameters		
Name	Optional	Description
Value	false	The octal value to convert.
Examples		
OctToBin(10)		

OctToDecimal

Description	Converts a octal value to a binary value.	
Syntax	OctToDecimal(Value)	
Returns	String	
Parameters		
Name	Optional	Description
Value	false	The octal value to convert.

OctToDecimal

Examples
OctToDecimal(10)

OctToHex

Description	Converts a octal value to a binary value.	
Syntax	OctToHex(Value)	
Returns	String	
Parameters		
Name	Optional	Description
Value	false	The octal value to convert.
Examples		
OctToHex(10)		

Power

Description	Returns a specified number to a specified power.	
Syntax	Power(Value1, Value2)	
Returns	Numeric	
Parameters		
Name	Optional	Description
Value1	false	A double-precision floating-point number to be raised to a power.

Power		
Value2	false	A double-precision floating-point number that specifies a power.
Examples		
Power(10, 3)		

PV		
Description	Returns the present value of an investment, where you make periodic, constant payments and the investment earns a constant interest rate.	
Syntax	PV(Rate, Terms, Payment [, Future_Value] [, Type])	
Returns	Numeric.	
Parameters		
Name	Optional	Description
Rate	false	Numeric. Interest rate earned in each period. Expressed as a decimal number. Divide the percent rate by 100 to express it as a decimal number. Must be greater than or equal to 0.
Terms	false	Numeric. Number of periods or payments. Must be greater than 0.
Payment	false	Numeric. Payment amount due per period. Must be a negative number.
Future_Value	true	Numeric. Future value/balance of the investment. If you omit this argument, PV uses 0.

PV		
Type	true	Integer. Timing of the payment. Enter 1 if payment is at the beginning of period. Enter 0 if payment is at the end of period. Default is 0. If you enter a value other than 0 or 1, the Integration Service treats the value as 1.
Examples		
PV(Ratevar, Termvar, Paymentvar), PV(3.25, 36, 125.50, 5000, 1)		

Quarter		
Description	Returns the Quarter from the date specified.	
Syntax	Quarter(Date)	
Returns	returns integer value of 1 - 4	
Parameters		
Name	Optional	Description
Date	false	Date to retrieve Quarter from
Examples		
Quarter("12/8/2011"), Quarter(varDate), Quarter(OrderDateColumn)		

Task Factory Functions List R

Last Modified on 02 October 2020

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Rand

Description	Returns a random number between 0 and 1. This is useful for probability scenarios.	
Syntax	RAND(), RAND(Seed), RAND(Min, Max)	
Returns	Numeric. For the same seed, the Integration Service generates the same sequence of numbers.	
Parameters		
Name	Optional	Description
Seed	true	Double. Starting value for the Integration Service to generate the random number. Value must be a constant. If you do not enter a seed, the Integration Service uses the current system time to derive the numbers of seconds since January 1, 1971. It uses this value as the seed.
Min	true	Integer. Minimum value for the generate the random number. Value must be a constant.
Max	true	Integer. Maximum value for the generate the random number. Value must be a constant.
Examples		
Rand(), Rand(.5), Rand(1, 10)		

RegExtract

Description	Extract values out of a string based on a regular expression	
Syntax	RegExtract(Input_String, Pattern [, Instance_Of_Pattern_To_Extract])	
Returns	String.	
Parameters		
Name	Optional	Description
Input_String	false	String value. The string that will be manipulated by the regular expression replace.
Pattern	false	String value. Regular expression pattern.
Instance_Of_Pattern_To_Extract	true	Integer. If more than one instance of the pattern is found in input_string, the value returned will be based on the value provided.
Examples		
RegExtract("Get numbers 1234", "[0-9]+"), RegExtract("Get second set of numbers 1234 4567", "[0-9]+", 2)		

RegMatch

Description	Performs a regular expression match again a string	
Syntax	RegMatch(Input_String, Pattern [, Ignore_Case])	
Returns	Boolean. Null if input_string or pattern parameters are null.	
Parameters		
Name	Optional	Description
Input_String	false	String value. The string that will be searched and matched.

RegMatch

Pattern	false	String value. Regular expression pattern.
Ignore_Case	true	Boolean. Specifies whether the match will be case sensitive.
Examples		
RegMatch("123434-", "^[0-9]+$"), RegMatch("Sentry One", "^[a-zA-Z\s]+$")		

RegReplace

Description	Replace values in a string based on a regular expression	
Syntax	RegReplace(Input_String, Pattern, Replacement [, Number_Of_Replacements])	
Returns	String.	
Parameters		
Name	Optional	Description
Input_String	false	String value. The string that will be manipulated by the regular expression replace.
Pattern	false	String value. Regular expression pattern.
Replacement	false	String value. Replacement value if the pattern is found.
Number_Of_Replacements	true	Integer. The number of times the pattern should be replaced.
Examples		
RegReplace("Replace numbers 1234", "[0-9]+", ""), RegReplace("Replace first set of numbers 1234 4567", "[0-9]+", "", 1)		

RegToken

Description	Splits a string based on a regular expression.	
Syntax	RegToken(Input_String, Delimeter_Regex, Occurrence [, Ignore_Case])	
Returns	String. If the input_string or delimeter_regex is NULL, the function returns NULL.	
Parameters		
Name	Optional	Description
Input_String	false	A string value that will be split by the delimeter_regex parameter.
Delimeter_Regex	false	A regular expression to be used to split the value of Input_String.
Occurrence	false	The occurrence that will be returned after the string has been split.
Ignore_Case	true	Tells the regular expression engine whether or not to perform the split using case sensitivity.
Examples		
RegToken("Sentry~44~One~55~Inc", "~[0-9]+~", 2, false)		

RegTokenCount

Description	Returns the nksn in a string that contains tokens seperated by the regular expression.
Syntax	RegTokenCount(Input_String, Expression_Regex [, Ignore_Case])
Returns	Integer. If the input_string or expression_regex is NULL, the function returns NULL.
Parameters	

RegTokenCount

Name	Optional	Description
Input_String	false	A string value that will be searched by the expression_regex.
Expression_Regex	false	A regular expression to be located in the Input_String.
Ignore_Case	true	Tells the regular expression engine whether or not to perform the split using case sensitivity.
Examples		
RegTokenCount("Sentry~44~One~55~Inc", "~[0-9]+~", false)		

RemoveChars

Description	Removes a portion of a string starting at an index.	
Syntax	RemoveChars(String, Start [, Length])	
Returns	String. Empty string if you pass a negative or 0 length value. NULL if a value passed to the function is NULL.	
Parameters		
Name	Optional	Description
String	false	Must be a character string. Passes the string you want to remove characters from. You can enter any valid task editor. If you pass a numeric value, the function converts it to a character string.

RemoveChars		
Start	false	Must be an integer. The position in the string where you want to start removing characters. You can enter any valid task editor. If the start position is a positive number, RemoveChars locates the start position by counting from the beginning of the string. If the start position is a negative number, RemoveChars locates the start position by counting from the end of the string. If the start position is 0, RemoveChars searches from the first character in the string.
Length	true	Must be an integer greater than 0. The number of characters you want RemoveChars to return. You can enter any valid task editor. If you omit the length argument, RemoveChars returns all of the characters from the start position to the end of the string. If you pass a negative integer or 0, the function returns an empty string. If you pass a decimal, the function rounds it to the nearest integer value.
Examples		
RemoveChars("This will be cut off here. This won't be here", 27), RemoveChars("This will be cut off here. This will be here too.", 27, 4))		

ReplaceChr	
Description	Replaces characters in a string with a single character or no character. REPLACECHR searches the input string for the characters you specify and replaces all occurrences of all characters with the new character you specify.
Syntax	REPLACECHR(CaseFlag, InputString, OldCharSet, NewChar)

ReplaceChr

Returns	String. Empty string if REPLACECHR removes all characters in InputString. NULL if InputString is NULL. InputString if OldCharSet is NULL or empty.

Parameters

Name	Optional	Description
CaseFlag	false	Must be an integer. Determines whether the arguments in this function are case sensitive. You can enter any valid task editor. When CaseFlag is a number other than 0, the function is case sensitive. When CaseFlag is a null value or 0, the function is not case sensitive.
InputString	false	Must be a character string. Passes the string you want to search. You can enter any valid task editor. If you pass a numeric value, the function converts it to a character string. If InputString is NULL, REPLACECHR returns NULL.
OldCharSet	false	Must be a character string. The characters you want to replace. You can enter one or more characters. You can enter any valid task editor. You can also enter a text literal enclosed within single quotation marks, for example, 'abc'. If you pass a numeric value, the function converts it to a character string. If OldCharSet is NULL or empty, REPLACECHR returns InputString.

ReplaceChr		
NewChar	false	Must be a character string. You can enter one character, an empty string, or NULL. You can enter any valid task editor. If NewChar is NULL or empty, REPLACECHR removes all occurrences of all characters in OldCharSet in InputString. If NewChar contains more than one character, REPLACECHR uses the first character to replace OldCharSet.
Examples		
REPLACECHR(false, "I can replace this i with t", "i", "t"), REPLACECHR(True, CompanyNameColumn, "_", " ")		

ReplaceStr		
Description	Replaces characters in a string with a single character, multiple characters, or no character. REPLACESTR searches the input string for all strings you specify and replaces them with the new string you specify.	
Syntax	ReplaceStr (CaseFlag, InputString, OldString1 [, OldString2, ..., OldStringN], NewString)	
Returns	String. Empty string if REPLACESTR removes all characters in InputString. NULL if InputString is NULL. InputString if all OldString arguments are NULL or empty.	
Parameters		
Name	Optional	Description
CaseFlag	false	Must be an integer. Determines whether the arguments in this function are case sensitive. You can enter any valid task editor. When CaseFlag is a number other than 0, the function is case sensitive. When CaseFlag is a null value or 0, the function is not case sensitive.

ReplaceStr		
InputString	false	Must be a character string. Passes the strings you want to search. You can enter any valid task editor. If you pass a numeric value, the function converts it to a character string. If InputString is NULL, REPLACESTR returns NULL.
OldString	false	Must be a character string. The string you want to replace. You must enter at least one OldString argument. You can enter one or more characters per OldString argument. You can enter any valid task editor. You can also enter a text literal enclosed within single quotation marks, for example, 'abc'. If you pass a numeric value, the function converts it to a character string. When REPLACESTR contains multiple OldString arguments, and one or more OldString arguments is NULL or empty, REPLACESTR ignores the OldString argument. When all OldString arguments are NULL or empty, REPLACESTR returns InputString. The function replaces the characters in the OldString arguments in the order they appear in the function. For example, if you enter multiple OldString arguments, the first OldString argument has precedence over the second OldString argument, and the second OldString argument has precedence over the third OldString argument. When REPLACESTR replaces a string, it places the cursor after the replaced characters in InputString before searching for the next match.

ReplaceStr

NewString	false	Must be a character string. You can enter one character, an empty string, or NULL. You can enter any valid task editor. If NewChar is NULL or empty, REPLACECHR removes all occurrences of all characters in OldCharSet in InputString. If NewChar contains more than one character, REPLACECHR uses the first character to replace OldCharSet.
Examples		
REPLACESTR(false, "I can replace this with that", "this", "that"), REPLACESTR(True, CompanyAddressColumn, "Street", "St")		

Reverse

Description	Reverses the input string.	
Syntax	Reverse(String)	
Returns	String. Reverse of the input value.	
Parameters		
Name	Optional	Description
String	false	Any character value. Value you want to reverse.
Examples		
Reverse("This will be backwards")		

Right

Right

Description	Returns a string containing a specified number of characters from the Right side of a string..	
Syntax	Right(String, Number_Of_Characters)	
Returns	String. NULL if a value passed to the function is NULL.	
Parameters		
Name	Optional	Description
String	false	String datatype. String expression from which the rightmost characters are returned.
Number_Of_Characters	false	The number of characters to return from the string.
Examples		
Right("Only these here", 10), Right(FirstNameColumn, 20)		

Round(DateTime)

Description	Rounds upto specified part of supplied datetime.	
Syntax	Round(Value [, Format])	
Returns	Date	
Parameters		
Name	Optional	Description
Value	false	A date value to be rounded.

Round(DateTime)

Format	true	Date part format which needs to be rounded. See below list for possible format specifier Year = Y, YY, YYY, YYYY Month = M, MM, MON, MONTH Day = D, DD, DDD, DY, DAY Hour = H, HH, HH12, HH24 Minute = MI Second = S,SS Rules: ====== When format=YY and Month >=7 then Year is increased by one and all other parts after year portion is reset When format=MM and Day >=16 then Month is increased by one and all other parts after month portion is reset When format=DD and Hour >=12 then Day is increased by one and all other parts after day portion is reset When format=HH and Minute >=30 then Hour is increased by one and all other parts after hour portion is reset When format=MI and Second >=30 then Minute is increased by one and all other parts after minute portion is reset When format=SS and Milliseconds >=500 then Second is increased by one and all other parts after second portion is reset

Examples

ROUND(TO_DATE('4/16/1998 8:24:19'), 'YY') ==> "1/1/1998" ROUND(TO_DATE('7/16/1998 8:24:19'), 'YY') ==> "1/1/1999" ROUND(TO_DATE('4/15/1998 8:24:19'), 'MM') ==> "4/1/1998" ROUND(TO_DATE('05/22/1998 10:15:29'), 'MM') ==> "6/1/1998" ROUND(TO_DATE('06/13/1998 2:30:45'), 'DD') ==> "6/13/1998" ROUND(TO_DATE('06/13/1998 22:30:45'), 'DD') ==> "6/14/1998" ROUND(TO_DATE('04/01/1998 11:29:35'), 'HH') ==> "04/01/1998 11:00:00" ROUND(TO_DATE('04/01/1998 13:39:00'), 'HH') ==> "04/01/1998 14:00:00" ROUND(TO_DATE('05/22/1998 10:15:29'), 'MI') ==> "05/22/1998 10:15:00" ROUND(TO_DATE('05/22/1998 10:15:30'), 'MI') ==> "05/22/1998 10:16:00" ROUND(TO_DATE('05/22/1998 10:15:29.499'), 'SS') ==> "05/22/1998 10:15:29.000" ROUND(TO_DATE('05/22/1998 10:15:29.500'), 'SS') ==> "05/22/1998 10:15:30.000"

Round(Number)

Round(Number)

Description	Rounds a decimal value to a specified number of fractional digits.	
Syntax	Round(Value [, Decimal_Points])	
Returns	Decimal	
Parameters		
Name	Optional	Description
Value	false	A decimal number to be rounded.
Decimal_Points	true	The number of decimal places in the return value. If you pass -ve argument then digits before decimal point is truncated. If you ommit this argument then number is rounded to nearest integer value
Examples		
ROUND(12.9938,3) ==> 12.994 ROUND(-18.8679, 3) ==> -18.868 ROUND(12.9937, 3) ==> 12.994 ROUND(13252.0, -2) ==> 13200.0 ROUND(1432.99, -2) ==> 1400.00 ROUND(12.99,0.7) ==> 13.0 ROUND(56.34, 0.7) ==> 56.3 ROUND(11.99) ==> 12.0 ROUND(-14.99) ==> -15.0		

RowNum

Description	Returns the current row number of the buffer from the source.
Syntax	RowNum()
Returns	Numeric (Int64)
Examples	
RowNum()	

Rpad

Description	Converts a string to a specified length by adding blanks or characters to the end of the string.	
Syntax	Rpad(First_String, Length [, Second_String])	
Returns	String of the specified length. NULL if a value passed to the function is NULL or if length is a negative number.	
Parameters		
Name	Optional	Description
First_String	false	Can be a character string. The strings you want to change. You can enter any valid task editor.
Length	false	Must be a positive integer literal. This argument specifies the length you want each string to be.
Second_String	true	Can be any string value. The characters you want to append to the right-side of the first_string values. You can enter any valid task editor. You can enter a specific string literal. However, enclose the characters you want to add to the beginning of the string within single quotation marks, as in 'abc'. This argument is case sensitive. If you omit the second_string, the function pads the beginning of the first string with blanks.
Examples		
Rpad("SentryOne", 30), Rpad(CompanyNameColumn, 50)		

RTrim

Description	Removes blanks or characters from the end of a string.

RTrim

Syntax	RTRIM(String [, Trim_Set])	
Returns	String. The string values with the specified characters in the trim_set argument removed. NULL if a value passed to the function is NULL.	
Parameters		
Name	Optional	Description
String	false	Any string value. Passes the strings you want to modify. You can enter any valid task editor. Use operators to perform comparisons or concatenate strings before removing characters from the beginning of a string.
Trim_Set	true	Any string value. Passes the characters you want to remove from the end of the string. You can also enter a text literal. However, you must enclose the characters you want to remove from the end of the string within single quotation marks, for example, 'abc'. If you omit the second string, the function removes blanks from the end of the first string. RTRIM is case sensitive.
Examples		
RTrim("SentryOne "), RTrim(CompanyNameColumn), RTrim("SentryOne----", "-")		

Task Factory Functions List S

Last Modified on 02 October 2020

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Second

Description	Returns the Second from the date specified.	
Syntax	Second(Date)	
Returns	returns integer value of 0 - 59	
Parameters		
Name	Optional	Description
Date	false	Date to retrieve Second from
Examples		
Second("12/8/2011"), Second(varDate), Second(OrderDateColumn)		

SendMail

Description	Allows the user to send an SMTP Message.	
Syntax	SendMail(Send_To_Emails, From_Email [, CC] [, Bcc] [, Subject] [, Body] [, Is_Body_HTML] [, Smtp_Server] [, Smtp_Port], [, Username] [, Password] [, Attachments] [, Priority])	
Parameters		
Name	Optional	Description

SendMail		
Send_To_Emails	false	A string that represents the E-mail Address(es) the user wishes the E-mail to be sent to. Multiple E-mails can be specified by seperating them with a semicolon (;).
From_Email	false	A string representing the E-mail Address the user wishes the E-mail to be sent from.
CC	true	Optional. A string representing any E-mail Address(es) the user wishes to carbon-copy. Multiple E-mails can be specified by seperating them with a semicolon (;).
Bcc	true	Optional. A string representing any E-mail Address(es) the user wishes to blind carbon-copy. Multiple E-mails can be specified by seperating them with a semicolon (;).
Subject	true	Optional. A string representing the subject for the Email.
Body	true	Optional. A string representing the Body for the Email
Is_Body_HTML	true	Optional. A boolean value representing whether the Body is formated in HTML.
Smtp_Server	true	Optional. A string value representing the name of the Server you wish to use to send the E-mail.
Smtp_Port	true	Optional. An integer value representing the Port required to send the E-mail through.
Username	true	Optional. A string value representing the username of the credentials required by the Smtp_Server to send the E-mail.

SendMail

Password	true	Optional. A string value representing the password for the Username required by the Smtp_Server to send the E-mail.
Attachments	true	Optional. A string value representing any Attachement(s) to include on the E-mail. Multiple attachments can be specified by seperating them with a Vertical Bar (|).
Priority	true	Optional. A string value representing the Priority level of the E-mail. The Priority can be "High", "Low", or "Normal".

Sequence

Description	Returns a unique number throughout your dataflow. You can use up to 5 different sequences at a time by calling SEQUENCE, SEQUENCE_2, SEQUENCE_3, SEQUENCE_4 or SEQUENCE_5	
Syntax	Sequence(), Sequence_2, Sequence(Reference_Count, Start_Value, Increment, Max_Value, Cycle_Value)	
Returns	Numeric (Int64)	
Parameters		
Name	Optional	Description
Reference_Count	true	A variable that will store the value of the current sequence.
Start_Value	true	Integer. The staring point of the sequence value.

Sequence

Increment	true	Integer. The value in which the sequence will be incremented.
Max_Value	true	Long. The maximum value of the sequence before the value is reset or an error is thrown based on the cycle parameter
Cycle_Value	true	Boolean. If cycle is set to true and the maximum value (max_value) is hit, the sequence number will be reset to the start_value parameter. If cycle is set to false, when the max_value is hit, an error will be thrown
Examples		
Sequence(), Sequence_5, Sequence(varRefValue, 1, 1, 100000, false)		

SetConnectionProp

Description	Set a property on the connection manager.	
Syntax	SetConnectionProp(Connection_Manager, Property_Name, Property_Value)	
Returns	String.	
Parameters		
Name	Optional	Description
Connection_Manager	false	Connection manager to set a property on. Must use the @@[connection_manager_name] syntax.
Property_Name	false	String value. The property to set on the connection manager.

SetConnectionProp		
Property_Value	false	String value. The property value to set on the connection manager.
Examples		
SetConnectionProp(@@[localhost.AdventureWorks], "ServerName", "localhost\sql2008")		

SetDatePart		
Description	Set any part of supplied date/time to user defined value.	
Syntax	SetDatePart(Date, Format, Value)	
Returns	Date.	
Parameters		
Name	Optional	Description
Date	false	The value to add to the aggregation
Format	false	Date part format specifier (See below). Year = "Y", "YY", "YYY", "YYYY" Day = "D", "DD", "DDD", "DY", "DAY" Month = "M", "MM", "MON", "MONTH" Hour = "H", "HH", "HH12", "HH24" Minute = "MI" Second = "S", "SS"" Millisecond = "MS"
Value	false	new value for specified date part.
Examples		
SetDatePart(Now(),"D", 1)		

SetVariable

Description	Sets the current value of a mapping variable to a value you specify. Returns the specified value. The SETVARIABLE function executes only if a row is marked as insert or update. SETVARIABLE ignores all other row types and the current value remains unchanged.	
Syntax	SETVARIABLE(@[Namespace::VariableName], Value)	
Returns	Current value of the variable. When value is NULL, the Integration Service returns the current value of $$Variable.	
Parameters		
Name	Optional	Description
VariableName	false	Namespace and Name of the mapping variable you want to set. Use with mapping variables with Max/Min aggregation type.
Value	false	The value you want to set the current value of the variable to. You can enter any valid task editor that evaluates to a datatype compatible with the datatype of the variable.
Examples		
SetVariable(@[User::FirstNameVar], "Chris"), SetVariable(@[User::FirstNameVar], FirstNameColumn)		

Sign

Description	Returns the value indicating the sign of a number.	
Syntax	Sign(Numeric_Value)	
Returns	Numeric	
Parameters		
Name	Optional	Description

Sign		
Numeric_Value	false	The value to return the sign of
Examples		
Sign(55)		

Sin		
Description	Returns the sine of the specified angle.	
Syntax	Sin(Numeric_Value)	
Returns	Numeric	
Parameters		
Name	Optional	Description
Numeric_Value	false	Passes the values for which you want to the sine of
Examples		
Sin(55)		

Sinh		
Description	Returns the hyperbolic tangent of the specified angle.	
Syntax	Sinh(Numeric_Value)	
Returns	Numeric	
Parameters		
Name	Optional	Description

Sinh		
Numeric_Value	false	Passes the values for which you want to the sine of
Examples		
Sinh(55)		

Sleep		
Description	Pause the processing of a package for x milliseconds.	
Syntax	Sleep(Milliseconds_To_Sleep)	
Returns	NULL	
Parameters		
Name	Optional	Description
Milliseconds_To_Sleep	false	The number of milliseconds to pause processing.
Examples		
Sleep(6000), Sleep(varNumberOfSecondsToSleep)		

SoundEx	
Description	Encodes a string value into a four-character string. SOUNDEX works for characters in the English alphabet (A-Z). It uses the first character of the input string as the first character in the return value and encodes the remaining three unique consonants as numbers.
Syntax	SOUNDEX(String)
Returns	String. NULL if one of the following conditions is true: ¨ If value passed to the function is NULL. ¨ No character in string is a letter of the English alphabet. ¨ string is empty.

SoundEx

Parameters		
Name	Optional	Description
String	false	Character string. Passes the string value you want to encode. You can enter any valid transformation expression.
Examples		
SoundEx("Sentr"), SoundEx(varSoundEx)		

Space

Description	Returns a string containing a specified number of spaces.	
Syntax	Space(Number_Of_Spaces)	
Returns	String. NULL if a value passed to the function is NULL.	
Parameters		
Name	Optional	Description
Number_Of_Spaces	false	The number of spaces to create.
Examples		
Space(10), Space(varNumberOfSpaces)		

Sqrt

Description	Returns the square root of a specified number.
Syntax	Sqrt(Numeric_Value)

Sqrt

Returns	Numeric	
Parameters		
Name	Optional	Description
Numeric_Value	false	Passes the values for which you want to the square root of
Examples		
Sqrt(55)		

StartsWith

Description	StartsWith determines whether a string starts with a character or string value	
Syntax	StartsWith (String_To_Search, Search_Value)	
Returns	Boolean. True if string_to_search ends with search_value. NULL if a value passed to the function is NULL.	
Parameters		
Name	Optional	Description
string	false	Character string. The string to search.
string	false	Character string. The value to find at the start of String_To_Search
Examples		
StartsWith("400 College Dr", "Dr"), If StartsWith(AddressColumn, "400") Then ...)		

SubStr

SubStr		
Description	Returns a portion of a string. SUBSTR counts all characters, including blanks, starting at the beginning of the string.	
Syntax	SubStr(String, Start [, Length])	
Returns	String. Empty string if you pass a negative or 0 length value. NULL if a value passed to the function is NULL.	
Parameters		
Name	Optional	Description
String	false	Must be a character string. Passes the string you want to extract a string from. You can enter any valid task editor. If you pass a numeric value, the function converts it to a character string.
Start	false	Must be an integer. The position in the string where you want to start counting. You can enter any valid task editor. If the start position is a positive number, SUBSTR locates the start position by counting from the beginning of the string. If the start position is a negative number, SUBSTR locates the start position by counting from the end of the string. If the start position is 0, SUBSTR searches from the first character in the string.
Length	true	Must be an integer greater than 0. The number of characters you want SUBSTR to return. You can enter any valid task editor. If you omit the length argument, SUBSTR returns all of the characters from the start position to the end of the string. If you pass a negative integer or 0, the function returns an empty string. If you pass a decimal, the function rounds it to the nearest integer value.

SubStr

Examples
Substr("This will be cut off here. This won't be here", 0, 27)

Sum

Description	Aggregation that calculates the total of all values. Can only be used as part of the Advanced Aggregation component	
Syntax	Sum(Value [, Condition])	
Returns	Double.	
Parameters		
Name	Optional	Description
Value	false	The value to add to the aggregation
Condition	false	The condition allows you to conditionally add values to the aggregation based on the expression. For instance, you could define the aggregate as Sum(SalesTotal, CustomerState="FL") and this would only add the value to the aggregation if the customers state was florida.

Sum
Examples
Sum(SalesTotal), Sum(SalesTotal, CustomerState="FL")

SysTimeStamp		
Description	Returns the current date and time of the node hosting the Integration Service with precision to the nanosecond	
Syntax	SysTimeStamp([Format])	
Returns	Date.	
Parameters		
Name	Optional	Description
Format	true	Optional. Sets the precision of the System Time. If Format is not specified, it defaults to nanoseconds. Valid format specifiers Year = "Y", "YY", "YYY", "YYYY" Day = "D", "DD", "DDD", "DY", "DAY" Month = "M", "MM", "MON", "MONTH" Hour = "H", "HH", "HH12", "HH24" Minute = "MI" Second = "S", "SS"" Millisecond = "MS"

Task Factory Functions List T-Z

Last Modified on 02 October 2020

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Tan

Description	Returns the tangent of the specified angle.	
Syntax	Tan(Numeric_Value)	
Returns	Numeric	
Parameters		
Name	Optional	Description
Numeric_Value	false	Passes the values for which you want to the tangent of
Examples		
Tan(55)		

Tanh

Description	Returns the hyperbolic tangent of the specified angle.	
Syntax	Tanh(Numeric_Value)	
Returns	Numeric	
Parameters		
Name	Optional	Description
Numeric_Value	false	Passes the values for which you want to the tangent of

Tanh
Examples
Tanh(55)

ToBigint		
Description	Converts a value to a big int (Int64)	
Syntax	ToBigint(Value)	
Returns	Int64 value	
Parameters		
Name	Optional	Description
Value	false	The value to convert.
Examples		
ToBigint(32), To_Big_Int(89090989)		

ToChar		
Description	Converts a value to a string with optional formatting for date values	
Syntax	ToChar(Value [, Format])	
Returns	String value	
Parameters		
Name	Optional	Description
Value	false	The value to convert.

ToChar

Format	false	The formatting that will applied when the date value is converted. This is the equivalent of calling DateTime.ToString().
Examples		
ToChar(32000), ToChar(Now(), "yyyy/MM/dd")		

ToDate

Description	Converts a string value to a date with optional formatting	
Syntax	ToDate(Value [, Format])	
Returns	String value	
Parameters		
Name	Optional	Description
Value	false	The value to convert.
Format	false	The formatting that will applied when the date value is converted. This is the equivalent of calling DateTime.ToString().
Examples		
ToDate("07/07/2012"), ToDate(Now(), "yyyy/MM/dd")		

Today

Description	Returns the current date without the time.

Today

Syntax	Today()
Returns	Date
Examples	
Today()	

ToDecimal

Description	Converts a value to a decimal value.	
Syntax	ToDecimal(Value [, Scale])	
Returns	Decimal	
Parameters		
Name	Optional	Description
Value	false	The value to convert.
Scale	false	The scale used to convert the decimal value.
Examples		
ToDecimal("1234.78"), ToDecimal("45876.765", 2)		

ToFloat

Description	Converts a value to a float value.
Syntax	ToFloat(Value)
Returns	Float

ToFloat

Parameters		
Name	Optional	Description
Value	false	The value to convert.
Examples		
ToFloat("1234.78")		

ToInteger

Description	Converts a value to an integer value.
Syntax	ToInteger(Value)
Returns	Integer. Any decimal values will be rounded.

Parameters		
Name	Optional	Description
Value	false	The value to convert.
Examples		
ToInteger("1234.78")		

Token

Description	Splits a string based on a delimeter.
Syntax	Token(Input_String, Delimeter, occurrence)
Returns	String. If the input_string or delimeter is NULL, the function returns NULL.

Token

Parameters		
Name	Optional	Description
Input_String	false	A string value that will be searched.
Delimeter	false	A delimeter to be used to split the value of Input_String.
Occurrence	false	The occurrence that will be returned after the string has been split.
Examples		
Token("400 College Dr|Middleburg|FL"|", 2) returns "Middleburg"		

TokenCount

Description	Gets the number of times a token occurs in a string.	
Syntax	TokenCount(Input_String, Token)	
Returns	Integer. If the input_string or token is NULL, the function returns NULL.	
Parameters		
Name	Optional	Description
Input_String	false	A string value that will be searched by the token value.
Token	false	A string expression to be located in Input_String.
Examples		
TokenCount("Sentry~44~One~55~Inc", "~")		

Trim

Description	Removes blanks or characters from the beginning and end of a string.	
Syntax	Trim(String [, Trim_Set])	
Returns	String. The string values with the specified characters in the trim_set argument removed. NULL if a value passed to the function is NULL. If the trim_set is NULL, the function returns NULL.	
Parameters		
Name	Optional	Description
String	false	Any string value. Passes the strings you want to modify. You can enter any valid task editor. Use operators to perform comparisons or concatenate strings before removing characters from the beginning of a string.
Trim_Set	true	Any string value. Passes the characters you want to remove from the beginning of the first string. You can enter any valid task editor. You can also enter a character string. However, you must enclose the characters you want to remove from the beginning of the string within single quotation marks, for example, 'abc'. If you omit the second string, the function removes any blanks from the beginning and end of the string. Trim is case sensitive. For example, if you want to remove the 'A' character from the string 'Alfredo', you would enter 'A', not 'a'.
Examples		
Trim(" SentryOne "), Trim(CompanyNameColumn), Trim("----SentryOne", "-")		

Trunc(Date)

Description	Truncates date to a specific year, month, day, hour, minute, second, millisecond, or microsecond.	
Syntax	Trunc(Date [, Format])	
Returns	Date.	
Parameters		
Name	Optional	Description
Date	false	The value to add to the aggregation
Format	true	Anything after this part will be reset to default value. e.g. if you pass HH then anything after Hour (e.g. Minute, Second, Milliseconds) will be reset to 0. Valid format specifiers Year = "Y", "YY", "YYY", "YYYY" Day = "D", "DD", "DDD", "DY", "DAY" Month = "M", "MM", "MON", "MONTH" Hour = "H", "HH", "HH12", "HH24" Minute = "MI" Second = "S", "SS"" Millisecond = "MS" Microsecond = "US"
value	false	new value for specified date part.
Examples		
SetDatePart(Now(),"D", 1)		

Trunc(Number)

Description	Truncates specified decimal places from numeric value.

Trunc(Number)

Syntax	Trunc(Number [, Precision])	
Returns	Number.	
Parameters		
Name	Optional	Description
Number	false	The number you want to truncate
Precision	true	Number of decimal places to keep. If you pass 0 then all digits after decimal point will be truncated. If you pass -ve precision number then it will truncate digits before decimal point.
Examples		
Trunc(567.7789,2) --> 567.7700 Trunc(567.7789) --> 567.0000 Trunc(567.7789,-2) --> 500.0000		

TryEval

Description	Returns defaultvalue if supplied expression fails to execute. This function help to continue execution on error and it behaves like Try..Catch construct provided in other programming languages such as VB.net/C#.	
Syntax	TryEval(expression, defaultvalue [, logerrors])	
Returns	Returns result of main expression if succueed else returns defaultvalue.	
Parameters		
Name	Optional	Description
expression	false	Any valid expression (without compiler error). E.g. MyCol/0 is valid expression but produce "Divide By Zero" error at runtime.

TryEval		
defaultvalue	false	Any valid expression or value which can be returned when main expression fails.
logerrors	true	Boolean value indicating whether error message should be logged or not. If this argument is True (which is default setting) then error messages will appear in you ssis package execution log as Information icon [e.g. <> some error occurred].
Examples		
TryEval(TO_CHAR("1")/TO_CHAR("0") , -999) -- Returns -999 becuase TO_CHAR("1")/TO_CHAR("0") fails with DivideByZero error TryEval(TO_DATE("55/12/1231","MM/DD/YYYY"), GETDATE()) -- Returns today's date if date parsing fails		

Upper		
Description	Converts lowercase string characters to uppercase.	
Syntax	UPPER(String)	
Returns	Uppercase string. If the data contains multibyte characters, the return value depends on the code page and data movement mode of the Integration Service. NULL if a value passed to the function is NULL.	
Parameters		
Name	Optional	Description
String	false	String datatype. Passes the values you want to change to uppercase text. You can enter any valid task editor.
Examples		
Upper("this will be uppercase now"), IIF(Upper(FirstNameColumn) = "CHRIS", 1, 0)		

WeekOfYear

Description	Returns the week number of the year of the date specified.	
Syntax	WeekOfYear(Date)	
Returns	returns integer value of 1 - 52	
Parameters		
Name	Optional	Description
Date	false	Date to retrieve week of year from
Examples		
WeekOfYear("12/8/2011"), WeekOfYear(varDate), WeekOfYear(OrderDateColumn)		

Year

Description	Returns the Year from the date specified.	
Syntax	Year(Date)	
Returns	returns integer value of year	
Parameters		
Name	Optional	Description
Date	false	Date to retrieve Year from
Examples		
Year("12/8/2011"),Year(varDate), Year(OrderDateColumn)		

Task Factory Advanced Lookup Cache

Last Modified on 20 January 2022

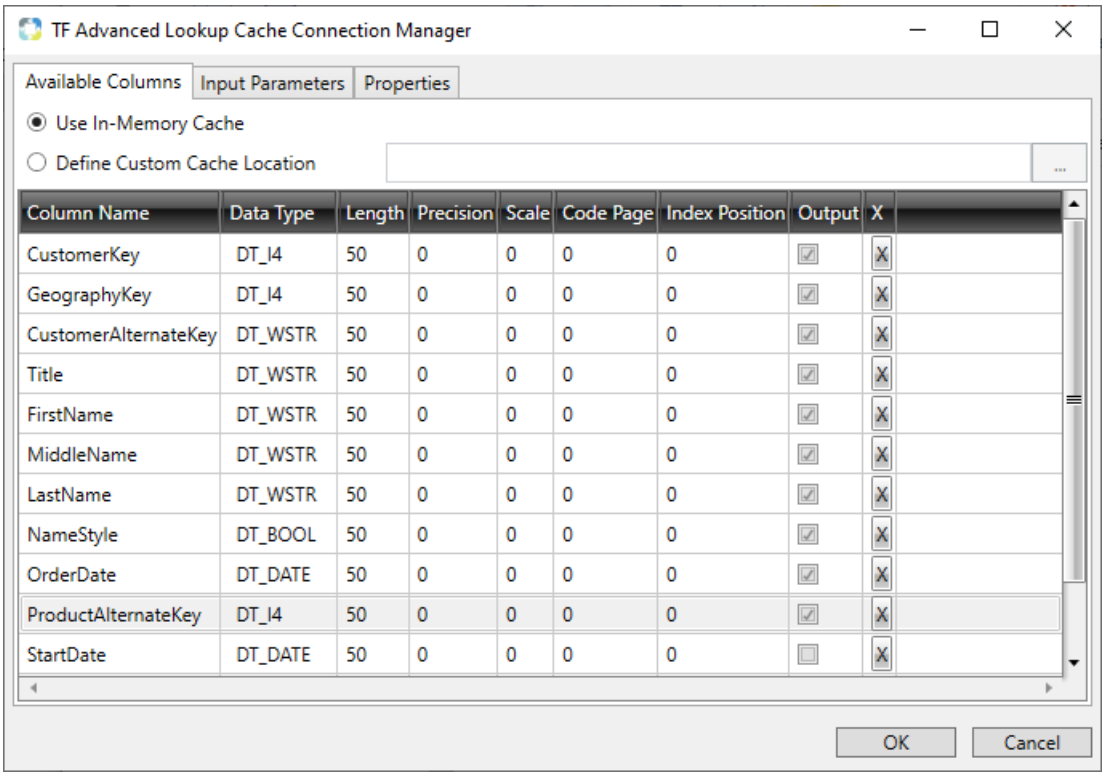
> Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Advanced Lookup Cache Connection Manager

The **Advanced Lookup Cache Connection Manager** is used to setup the cache database and configure how data is retrieved from it.

Available Columns Tab

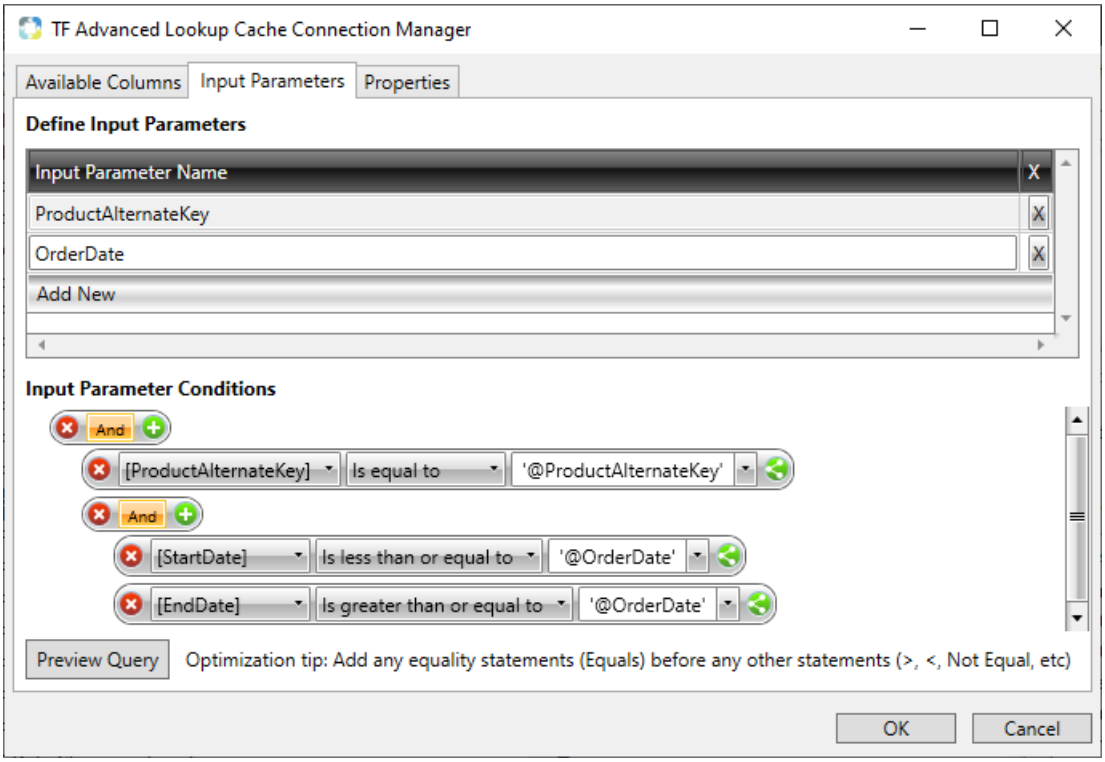
The available columns tab allows you to define columns that are part of the cache, whether or not the column is output when a lookup is performed, and what type of cache database to use.



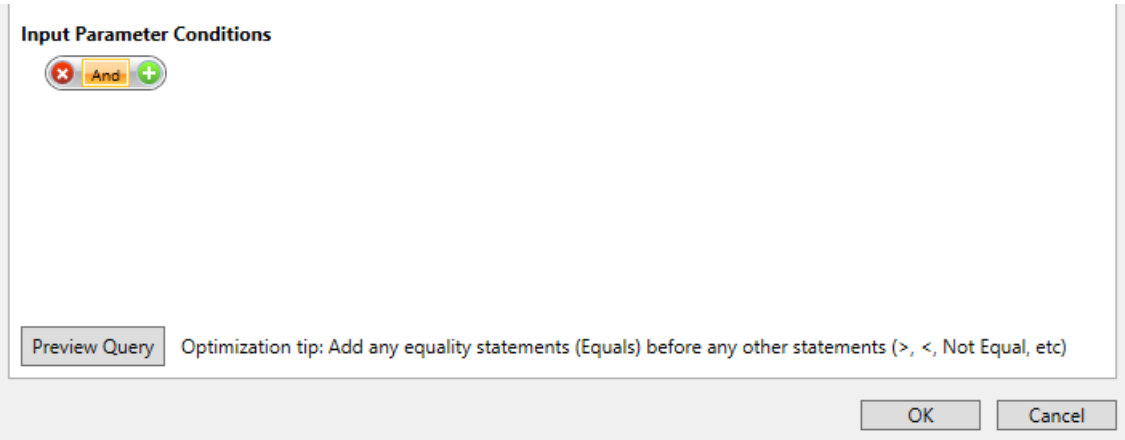
Option	Description
Use In-Memory Cache	This tells the connection manager whether to use in-memory cache or on-disk cache when creating the cache database.

Option	Description
Define Customer Cache Location	By default, the cache connection manager is created in a temporary file location defined by SSIS. You can choose to store the database in a custom location by checking this box and entering/selecting a path to the file. **Note:** The file should be defined with an .s3db extension.
Output Columns	The **Output column** tells the connection manager to return this value when a lookup is performed. The output columns can be seen on the **Advanced Lookup Transform UI** after a cache connection manager is selected.

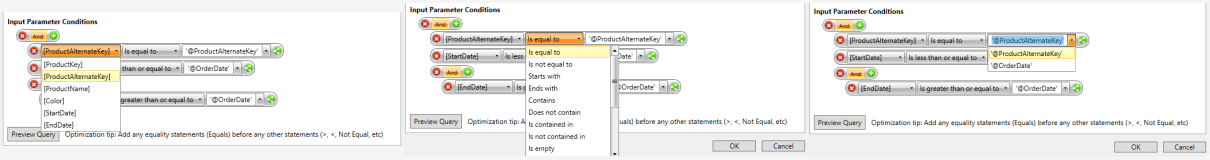
Input Parameters Tab



Option	Description
Input Parameter	Used by the **Advanced Lookup Transform** to define the lookup values to find rows in the cache database. They are mapped from **Input columns** in the **Advanced Lookup Transform**.
Input Parameter Condition	This is where you can easily create conditions for queries. By selecting the word **And**, you can select between an **And** and **Or** conditions. Selecting the green adds more conditions and selecting the red removes them.

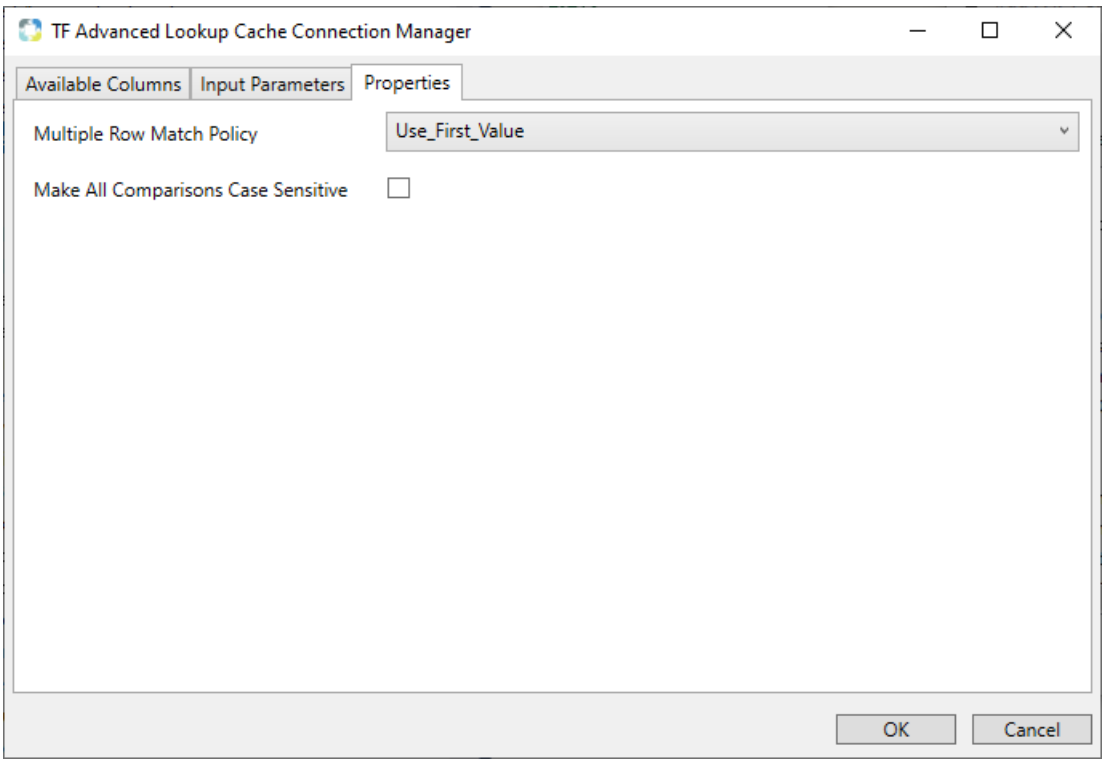


Note: The first condition requirement is the table key is equal to the input parameter.



Select the green **New Group Icon** at the end of the condition to create a new **And/Or** group.

Properties Tab

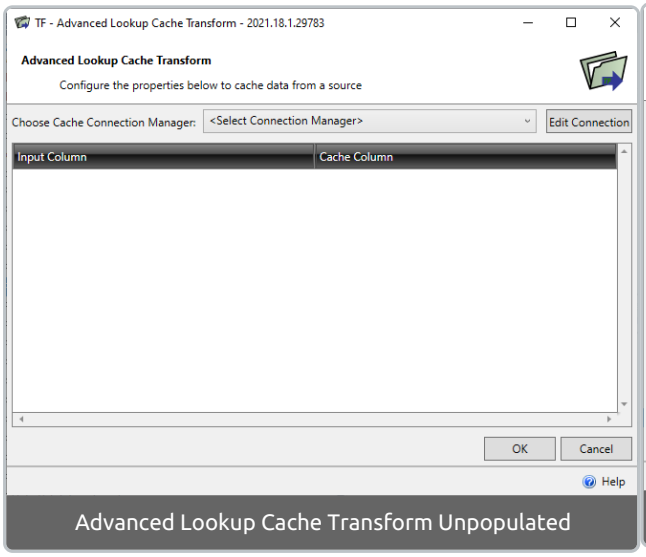


Option	Description

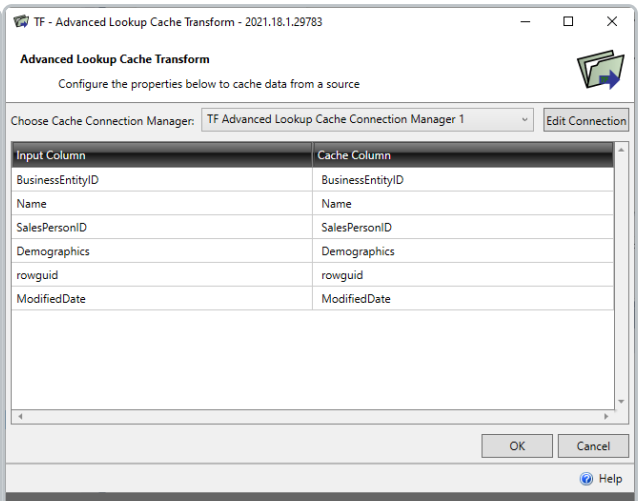
Option	Description
Multiple Row Match Policy	This option lets you define what should happen if multiple rows are found during a lookup.
Make All Comparisons Case Sensitive	Using this option makes all string comparisons performed case sensitive (e.g. SolarWinds does not equal solarwinds).

Advanced Lookup Cache Transform

Transform Icon	Transform Description
	The **Advanced Lookup Cache Transform** is used to store rows from a source into an in-memory or on-disk cache. See Advanced Lookup Cache Connection Manager to learn more about setting up the connection manager.



Advanced Lookup Cache Transform Unpopulated

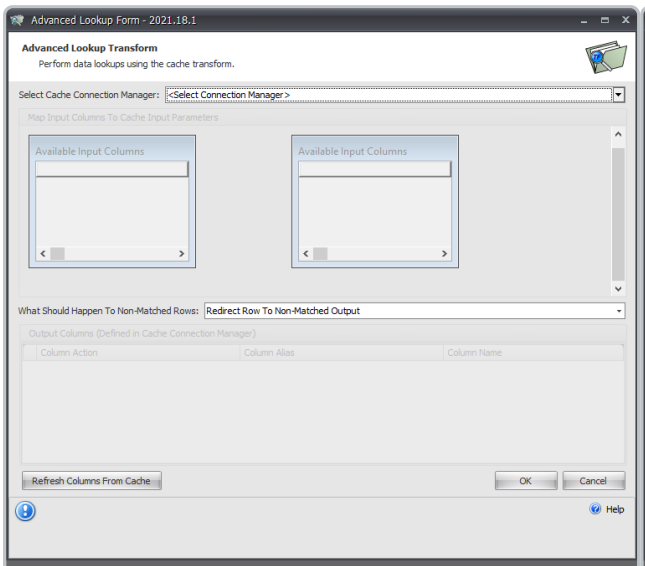


Advanced Lookup Cache Transform Populated

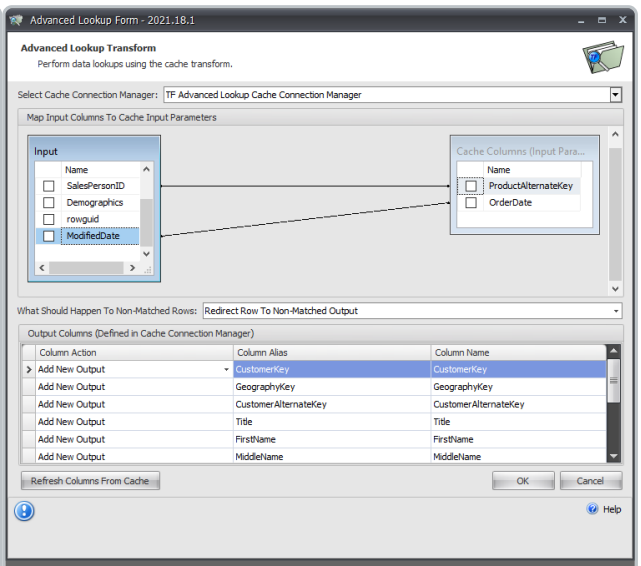
Option	Description
Choose Cache Connection Manager	You can select an existing **Advanced Lookup Cache Connection Manager** or choose to create a new one.
Map Cache Columns	After a **Cache Connection Manager** has been selected, the input columns and columns from the cache are automatically mapped if the names match. Otherwise you need to manually map the columns.

Advanced Lookup Transform

Transform Icon	Transform Description
	The **Advanced Lookup Transform** allows you to store a temporary cache of rows and perform lookups on that cache to pull that data into a data flow. There are two modes for the **Advanced Lookup Transform** : • Connected • Disconnected See the Advanced Lookup Cache Connection Manager to learn more about setting up the connection manager.



Advanced Lookup Transform Unpopulated



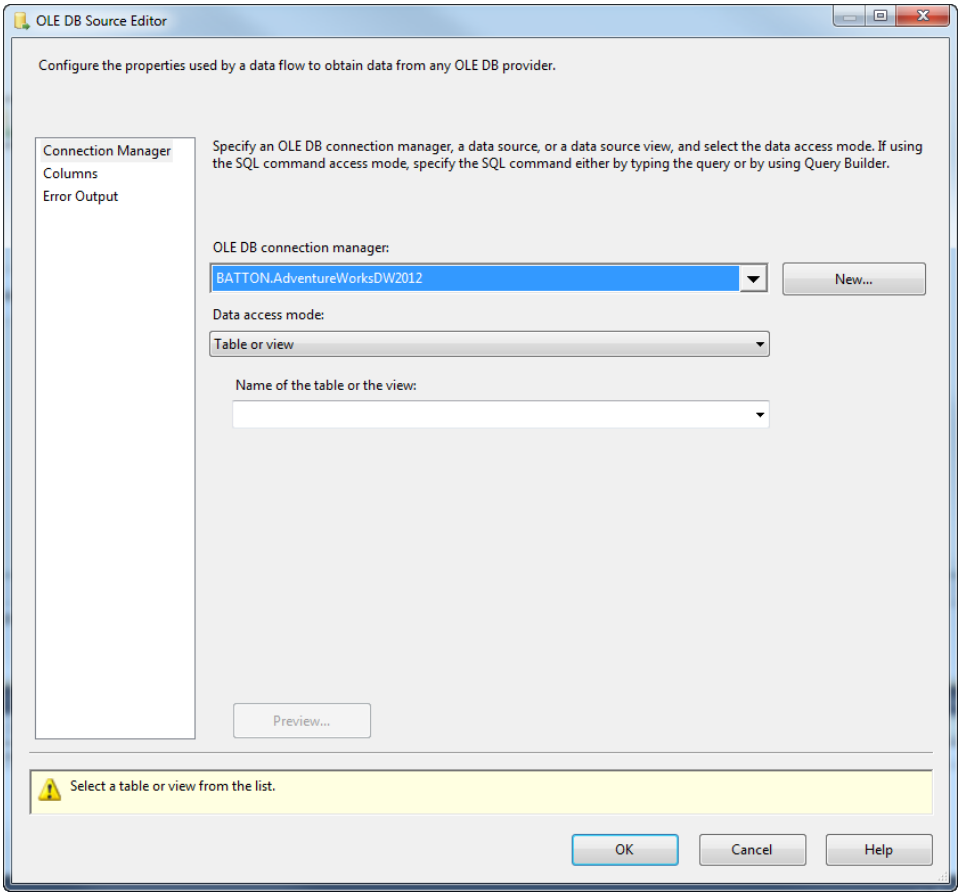
Advanced Lookup Transform Populated

Option	Description
Select Cache Connection Manager	Select an **Advanced Lookup Cache Connection Manager**. If you have not created one yet, you will need to create one. Once you have one selected, it automatically populates the available columns.
Map Input Columns To Cache Input Parameters	This is where you map input columns to the input parameters defined in the **Advanced Lookup Cache Connection Manager**.
What Should Happen To Non-Matched Rows	This tells the component what should happen when a non-matched row is found in the lookup. • **Fail Component** - This option fails the component when a non-matched row is found. • **Redirect Rows To Non-Matched Output** - This option redirects the rows to the non-matched output. • **Output Null Values To Output** - This option outputs NULL values for each of the output columns defined in the advanced lookup.
Output Columns (Defined in Cache Connection Manager)	This section lists all of the output columns defined in the **Advanced Lookup Cache Connection Manager**. You have a few choices on how to output the values of the output columns based on the column action. • **Add New Output** - This option creates a new output column based on the column alias • **Replace Input Column** - This option allows you to overwrite the current value of the selected input column with the value of the output column from the lookup cache.
Refresh Columns From Cache Button	This button refreshes the columns from the cache connection manager.

Setting up an Advanced Lookup Transform

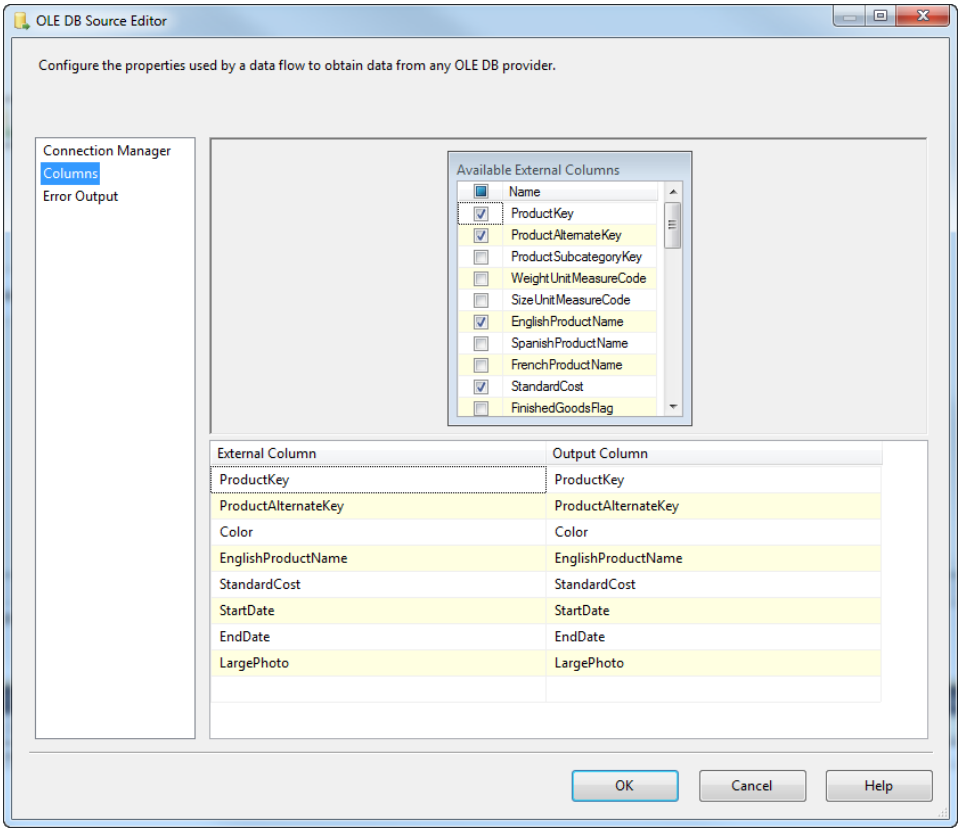
Complete the following steps to setup an **Advanced Lookup Transform**:

1. Create a data flow task, and then add an **OLE DB Source**. In this example we are using an OLE DB source connected to AdventureWorksDW. (download here).

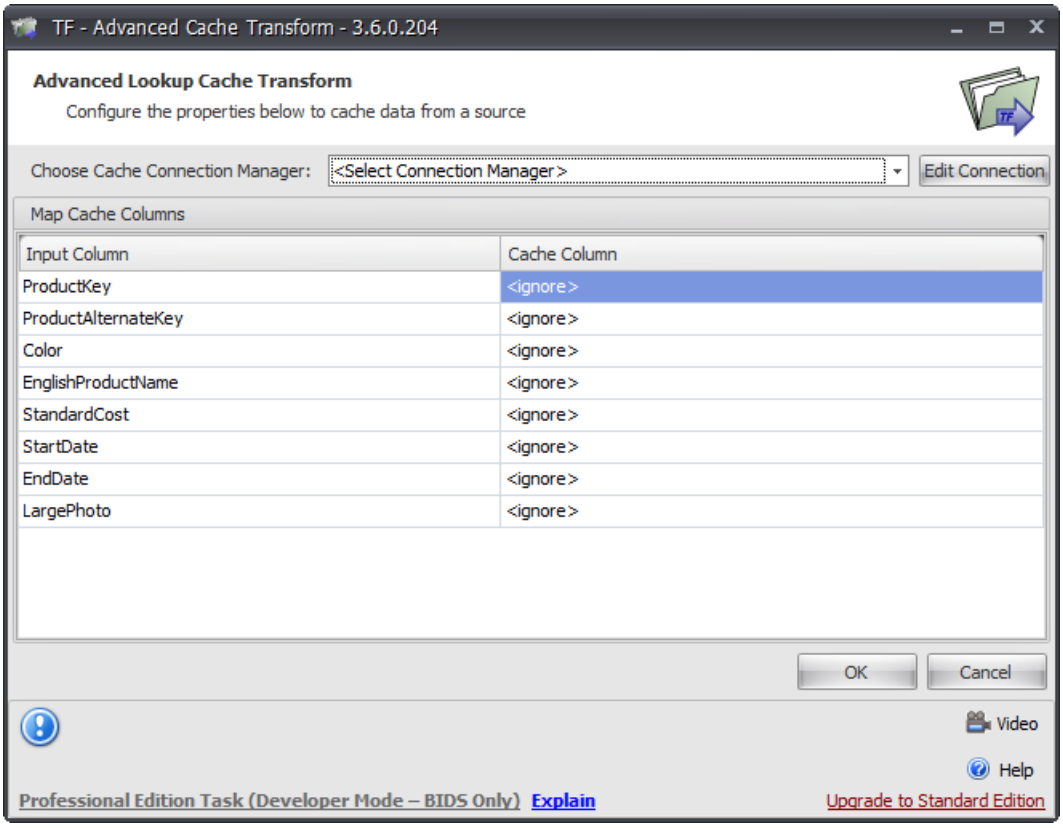


2. For this example select only the following columns from the columns tab.

- ProductKey
- ProductAlternateKey
- Color
- EnglishProductName
- StandardCost
- StartDate
- EndDate
- LargePhoto

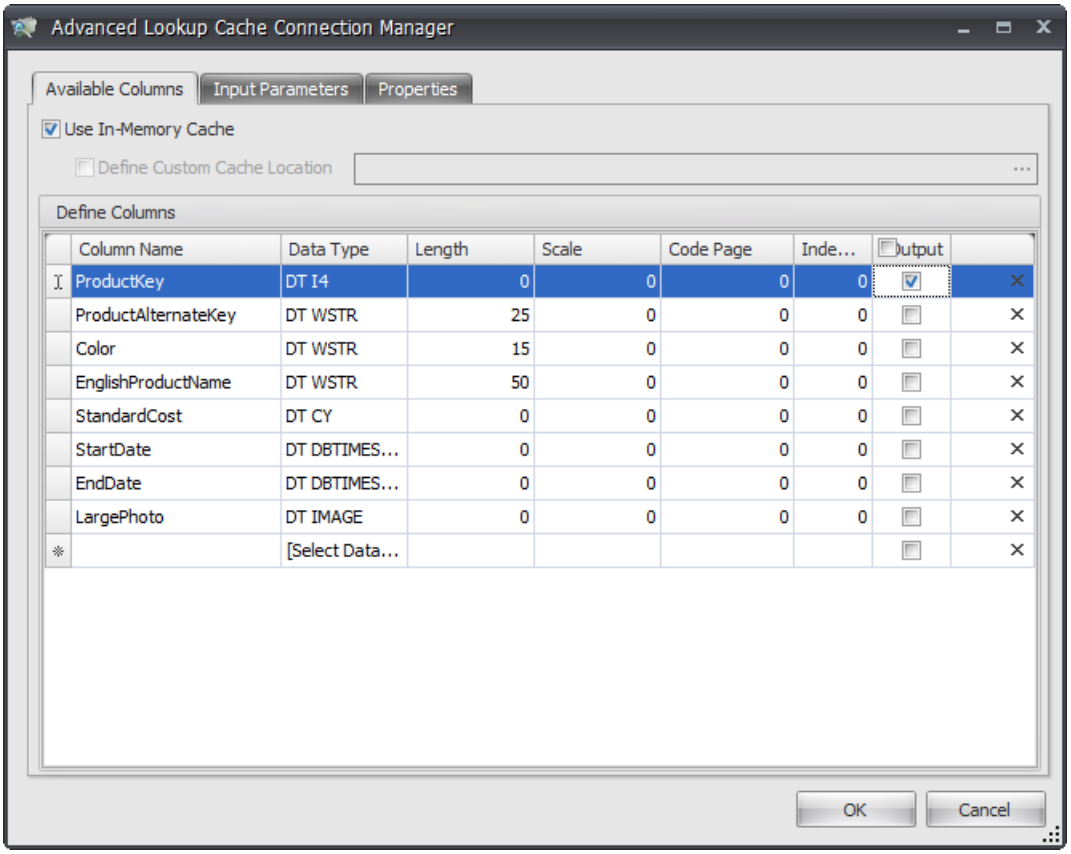


3. Now, create a lookup cache using the **Advanced Lookup Cache Transform**. Drag an instance of **TF Advanced Lookup Cache Transform** to the designer window. Attach the OLE DB Source Output to the **TF Advanced Lookup Cache Transform**. Open the transform by double clicking it, which opens the edit window.

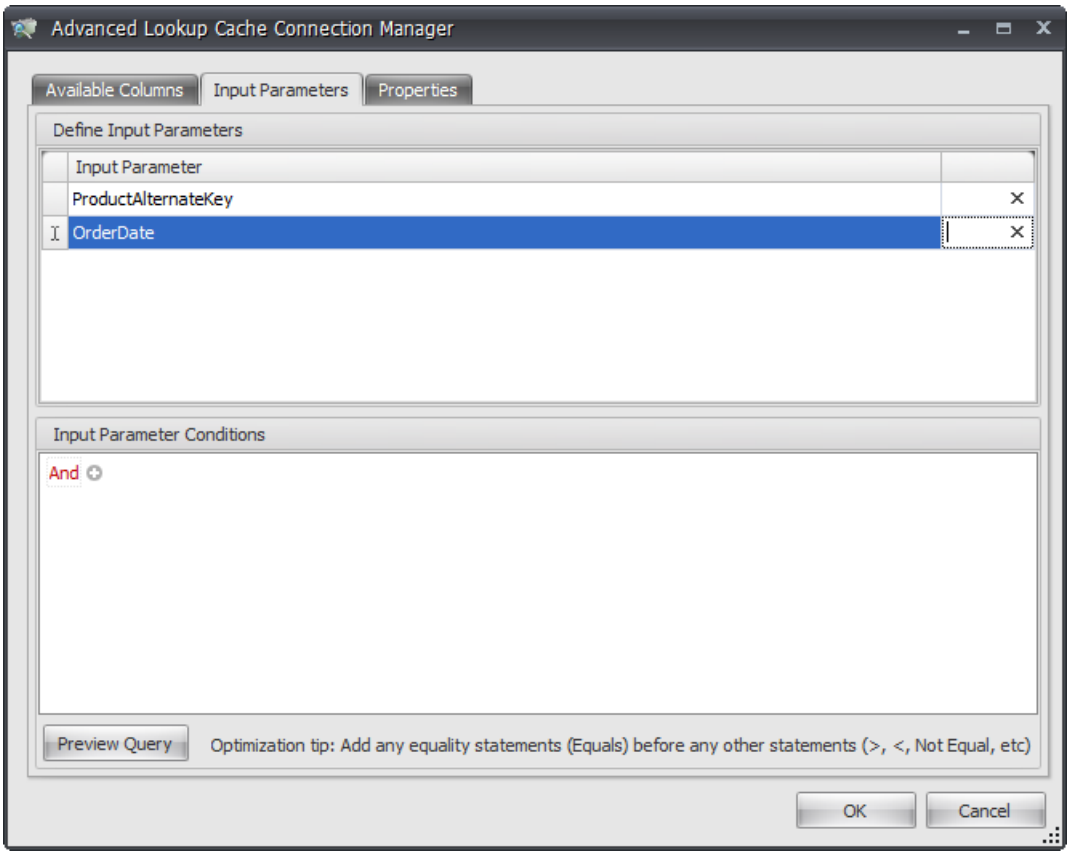


4. Select the **Choose Cache Connection Manager** drop down list and then choose **Create New Lookup Cache Connection...**

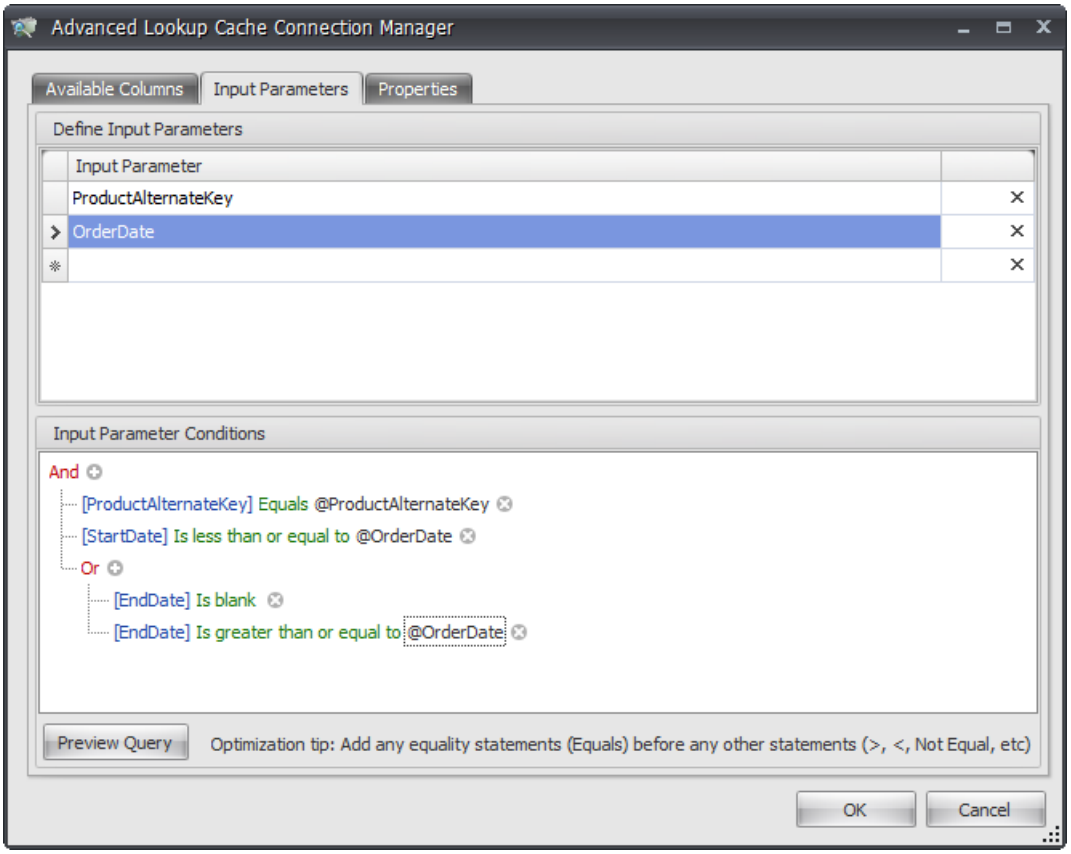
5. Setup the **Advanced Lookup Cache Connection Manager**. Then deselect all output except for **Product Key**.



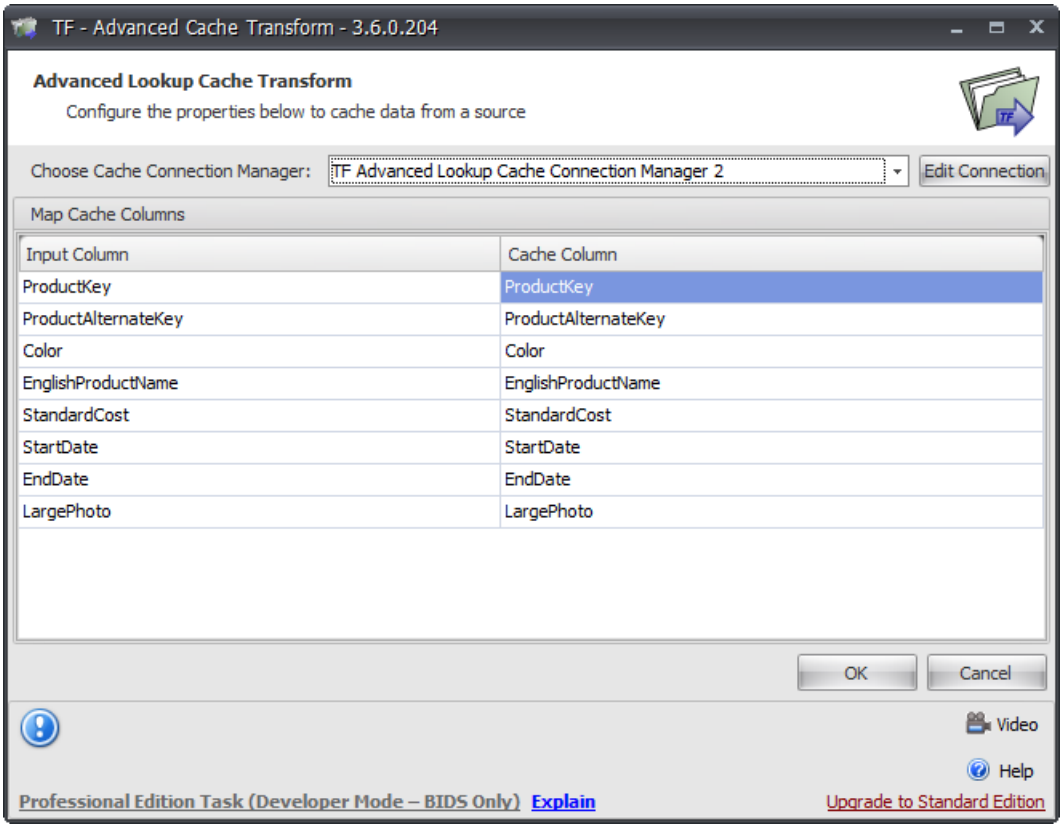
6. Next, select the **Input Parameters** tab, and then create two input parameters named **ProductAlternateKey** and **OrderDate**.



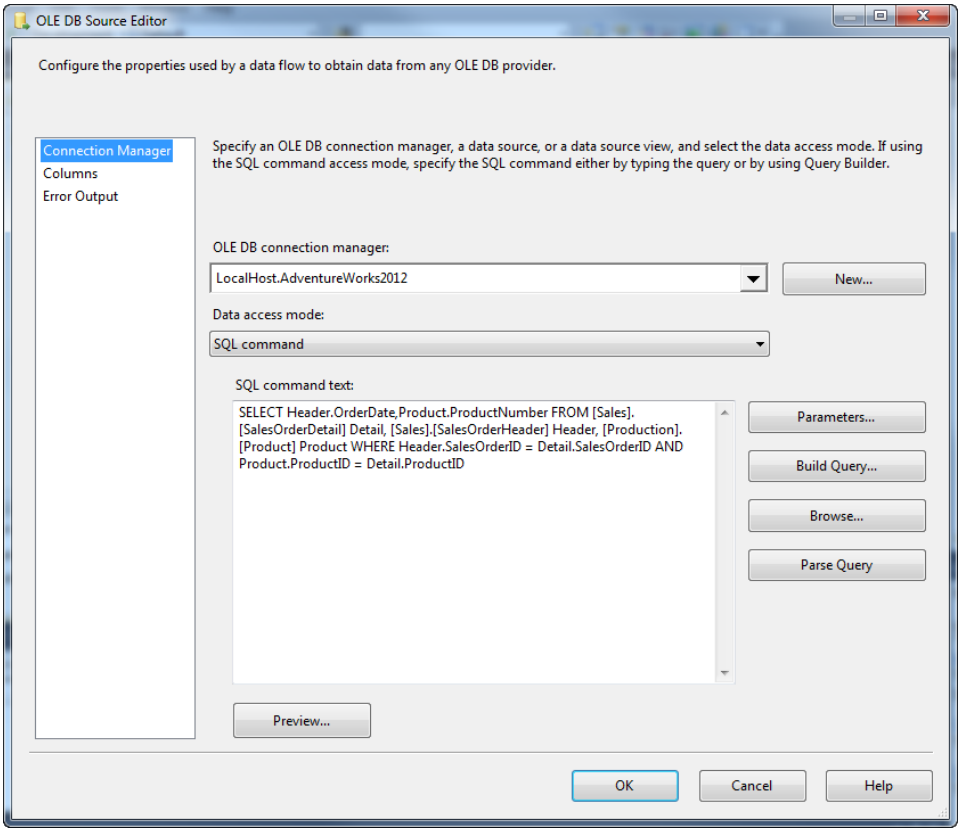
7. Then, create the following conditions: Note that **Is Blank** = **Is Null**.



8. Select **OK** and then your window should look like this:

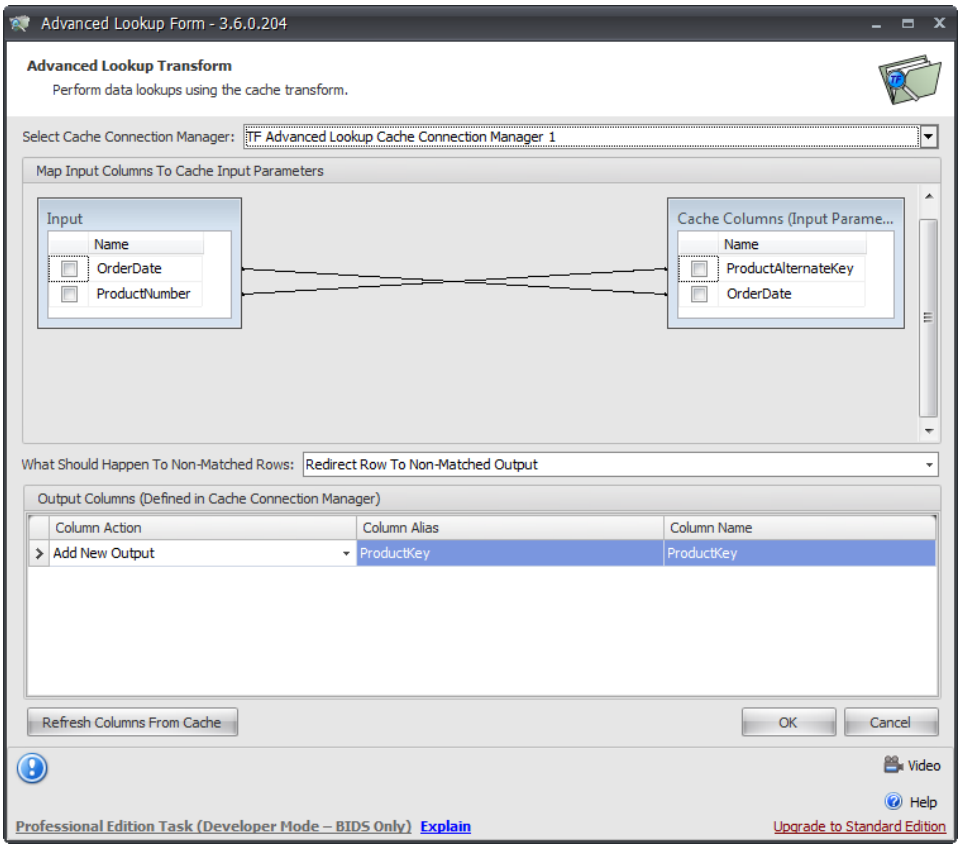


9. Add a new data flow task and a new OLE DB source, for this example we are connecting to AdventureWorks and creating a custom SQL query.

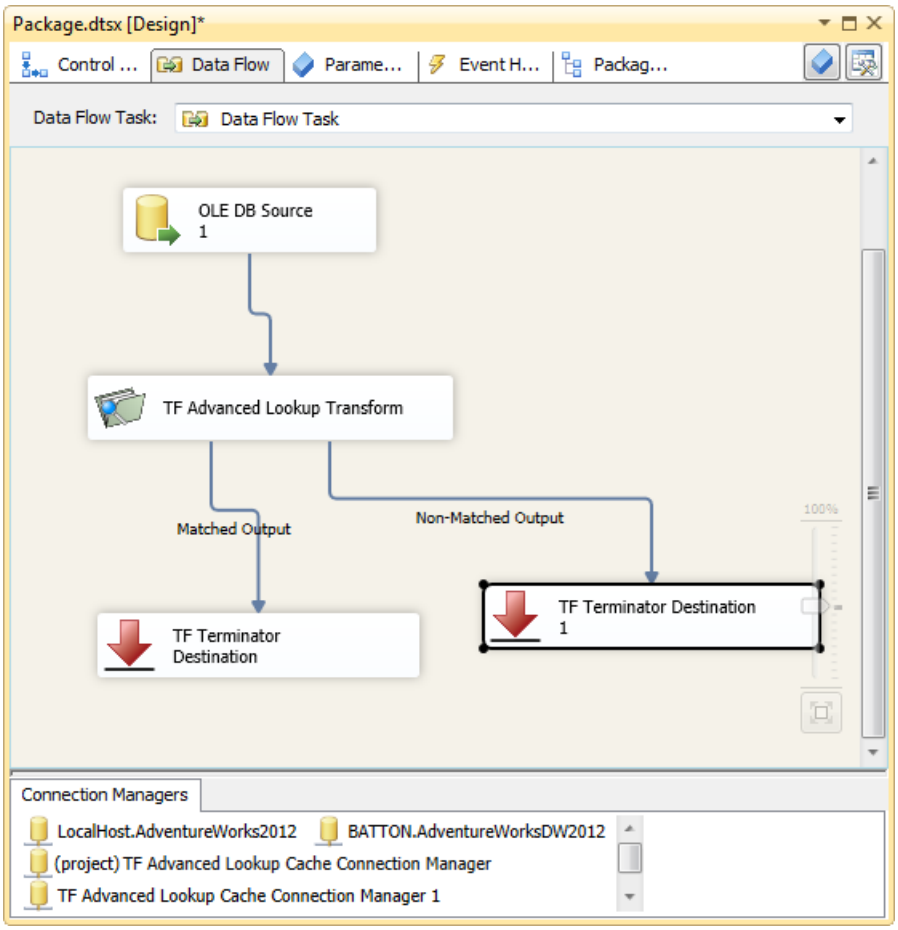


10. Add a new **TF Advanced Lookup Transform** to the designer, drag the output from the **OLE DB source** to the **TF Advanced Lookup Transform**. Double click the transform to open it. Once the **editor** window is open, choose the previously created Cache Connection Manager.

11. In the **Map Input Columns to Cache Input Parameters** window, connect **OrderDate** from the **Input column** to the **OrderDate** from the **Cache Column** by dragging one to the other. Do the same for the remaining field, and then select **OK**.



12. Drag your Matched and Non-Matched outputs to your desired destination. For this example we used TF Terminator Destinations for both.



Disconnected Lookups

Disconnected Lookups allow you to use a cache to perform lookups without an **Advanced Lookup Cache Transform** in your data flow.

Functions

There are two functions that can be used to execute disconnected lookups:

Function	Description

Function	Description
LookupData(connectionManager As AdvancedLookupCacheConnectionManager, ParamArray args As Object)	Used to retrieve the FIRST column setup as an output column in the **Advanced Lookup Cache Manager** defined in the connectionManager parameter. Parameters: • **connectionManager** - The name of the Advanced Lookup Cache Connection Manager. • **args** - Parameter array of input parameters defined in the **Advanced Lookup Cache Connection Manager**. Args matches the number of input parameters setup in the Advanced Lookup Cache Connection manager defined in the connectionManager argument.
LookupDataByColumn(connectionManager As AdvancedLookupCacheConnectionManager, outputColumnName As String, ParamArray args As Object	Used to retrieve the value of the column defined in the **outputColumnName** parameter from the Advanced Lookup Cache Connection manager defined in the connectionManager parameter. Parameters: • **connectionManager** - The name of the Advanced Lookup Cache Connection Manager • **outputColumnName** - The name of the column to retrieve from the output. The column defined here must be setup as an output column in the Advanced Lookup Cache Connection Manager. • **args** - Parameter array of input parameters defined in the **Advanced Lookup Cache Connection Manager**. Args matches the number of input parameters setup in the Advanced Lookup Cache Manager defined in the connectionManager argument.

Task Factory Amazon

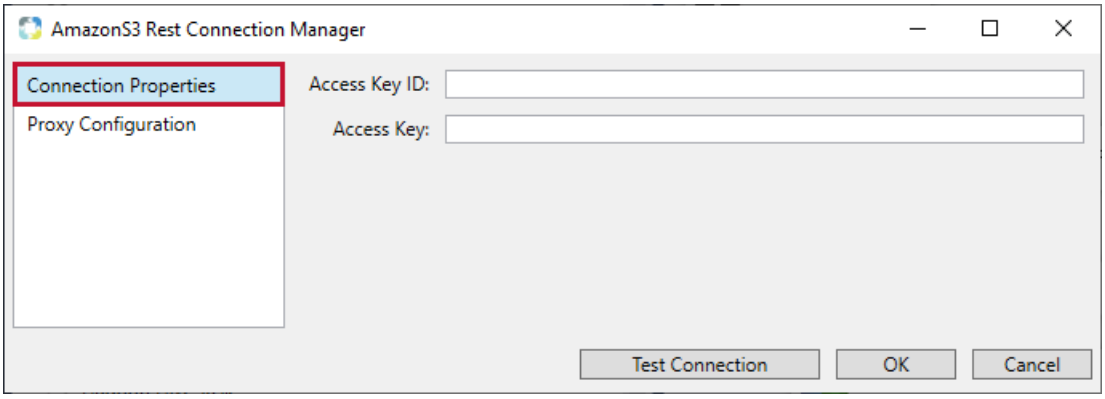
Last Modified on 20 January 2022

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

AmazonS3 Rest Connection Manager

Connection Properties

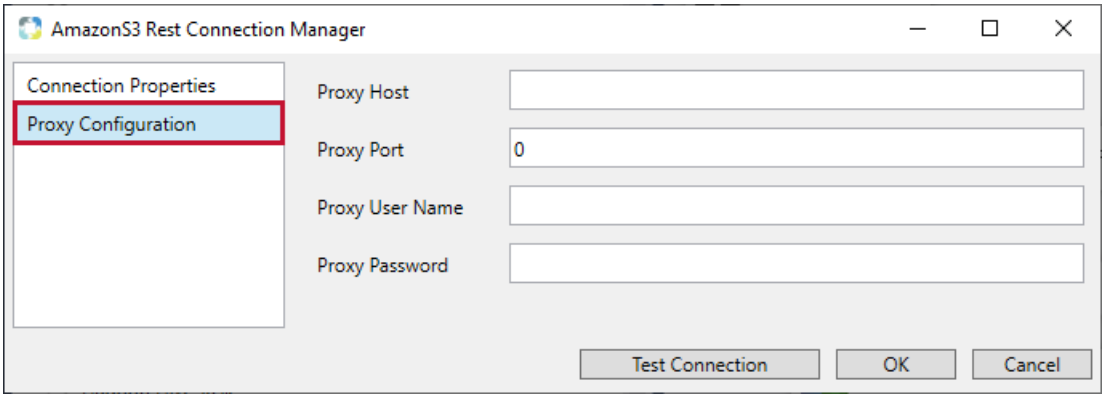
Used with the AmazonS3 Rest Source and AmazonS3 Rest Destination.



Option	Description
Access Key ID	The Access Key ID provided by Amazon Web Services to connect to the desired container.
Access Key	The Access Key provided by Amazon Web Services.

Note: Test Connection requires the IAM user to have *ListAllMyBuckets* permissions. To find the available permissions for your IAM user, open the AWS Console, select the user and look under **Permission Policies**. You may need to add/edit the policy to add *ListAllMyBuckets* permissions.

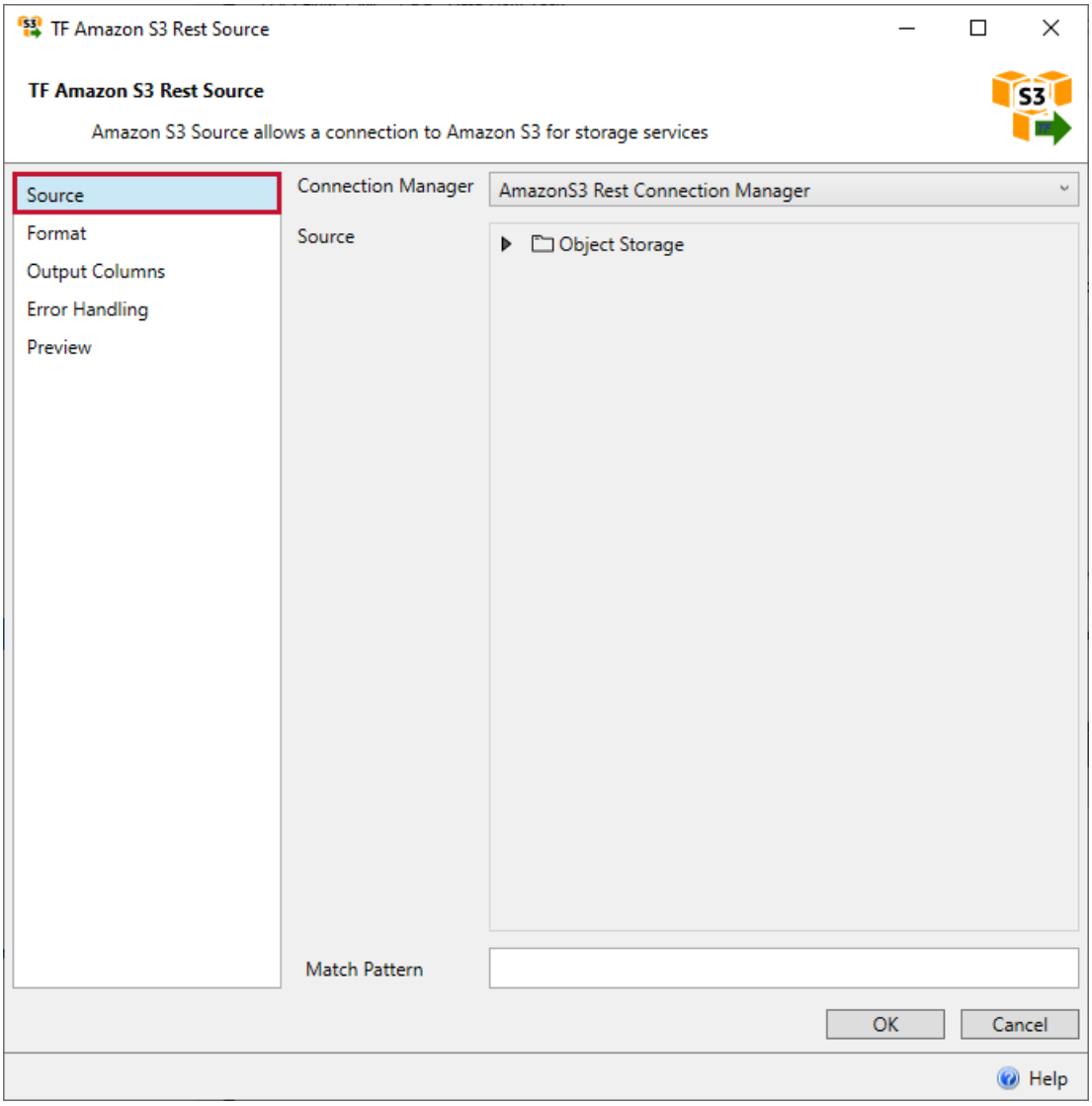
Proxy Configuration



Option	Description
Proxy Host	Identifies the proxy address.
Proxy Port	Identifies the port used by the proxy.
Proxy User Name	Enter the user name for proxies that require user authentication.
Proxy Password	Enter the password for proxies that require user authentication.

AmazonS3 Rest Source

Source



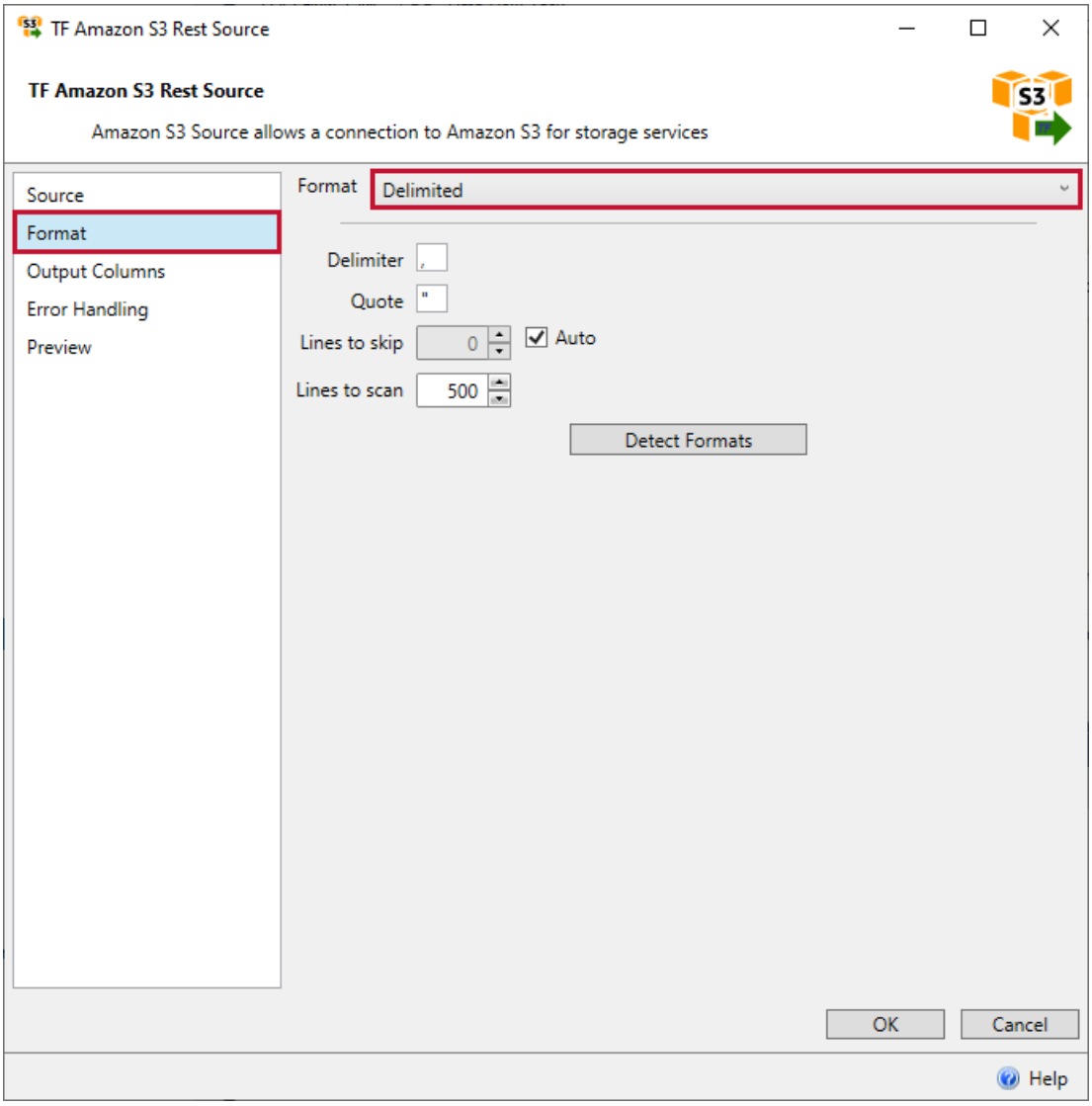
Begin by creating a connection manager that connects to the service's storage container.

Format

The following formats are available:

Delimited Format

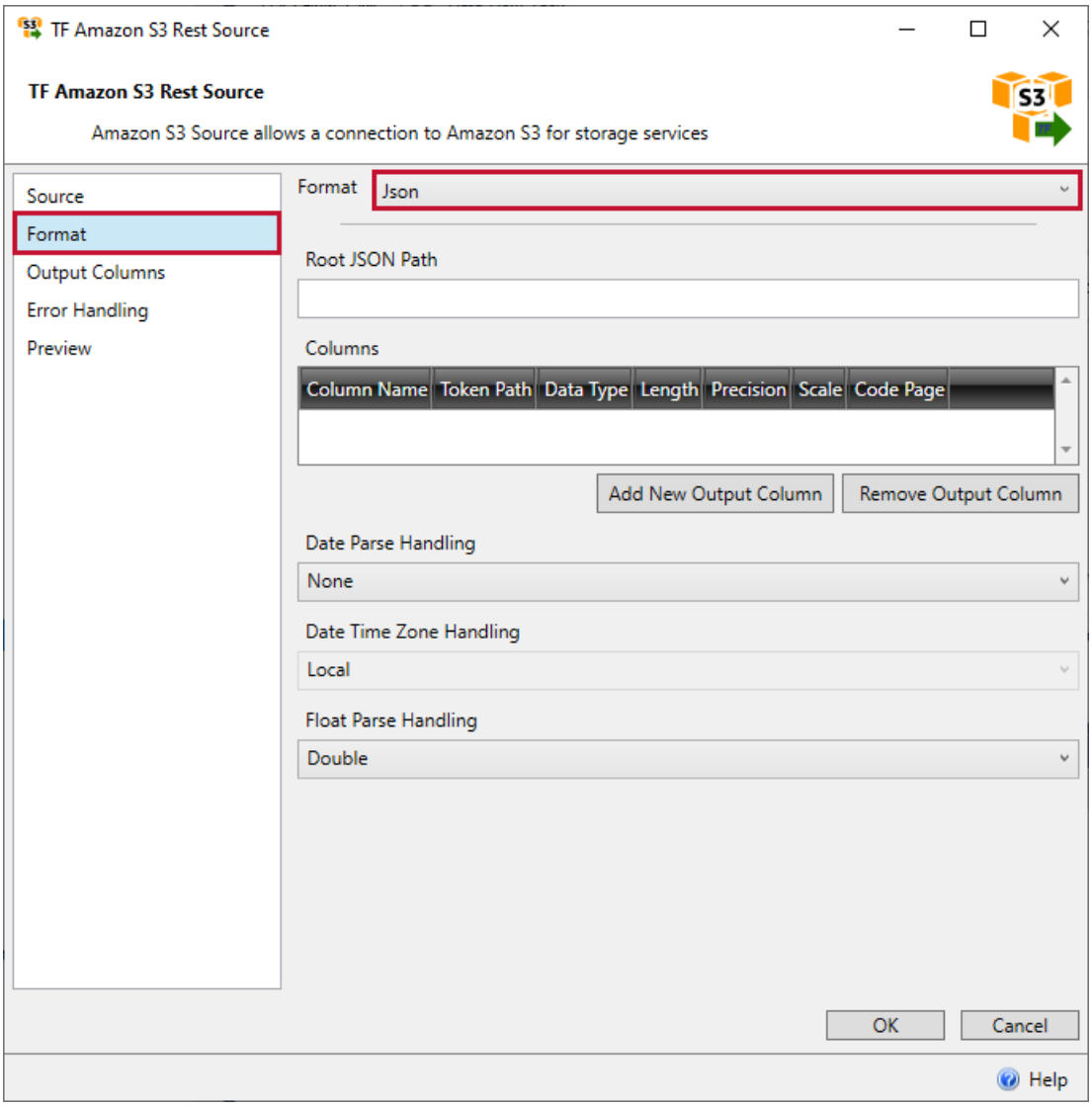
Users can manually configure the delimiter, quote identifiers, the lines to skip, and lines to scan. Select **Detect Formats** to configure these fields automatically.



Option	Description
Delimiter	Identifies the character that separates columns. The default value is the comma (,).
Quote	Identifies the character used to indicate a value (if present).
Lines To Skip	Users can designate the number of lines to skip at the beginning of the delimited file.
Auto	Automatically detects which lines to skip.
Lines To Scan	Determines the number of lines to scan when detecting the file.
Detect Formats	Select this button to begin scanning the chosen file.

Detected Delimited View - After the format is detected, a new view appears that allows users to configure or change different properties of the file.

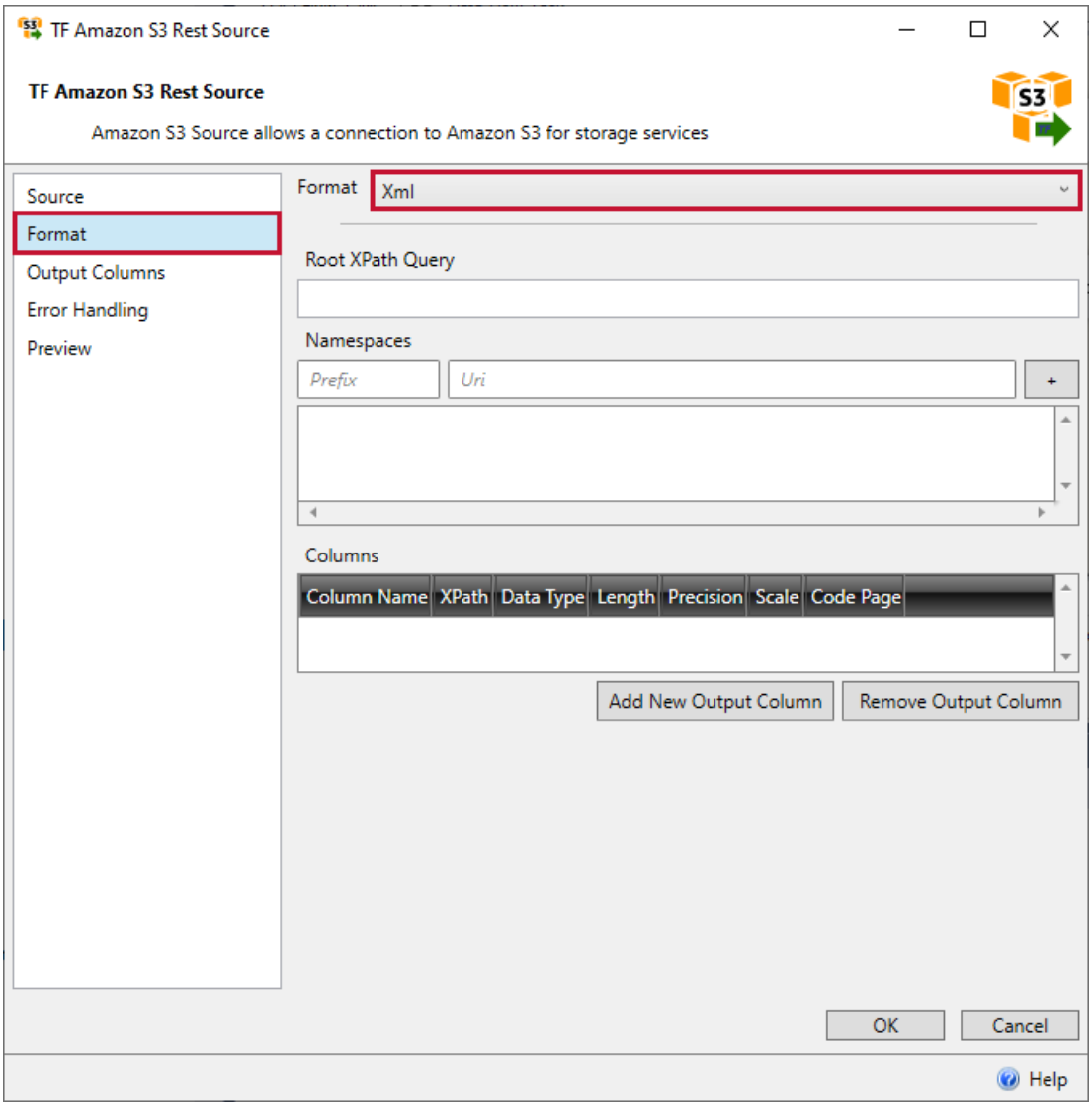
Json Format



Option	Description
Root Json Path	JSON queries can return multiple levels, therefore, this field Identifies the root to be used.
Output Columns	In this window, users can add and remove columns, define their name, token path (defined within the raw data), data type, length, precision, scale, and code page.
Date Parse Handling	Specifies how date formatted strings are parsed when reading JSON text. • **None** - Strings are not parsed as a date type and are read as strings. • **DateTime** - Date formatted strings are parsed as DateTime. • **DateTime Offset** - Date formatted strings are parsed as DateTimeOffset.

Option	Description
Date Time Zone Handling	Specifies how to treat time values converted between string and DateTime. • **Local** - Treats the time as local (UTC times convert to local times.) • **Utc** - Treats the time as UTC. This converts local time to UTC. • **Unspecified** - Treats the time as local time if a DateTime is being converted to a string. If a string is being converted to DateTime, convert to a local time if a time zone is specified. • **RoundtripKind** - Time zone is preserved when converting.
Float Parse Handling	Specifies how floating point numbers are parsed. • **Double** - Floating point numbers are parsed as a double datatype. • **Decimal** - Floating point numbers are parsed as a decimal datatype.

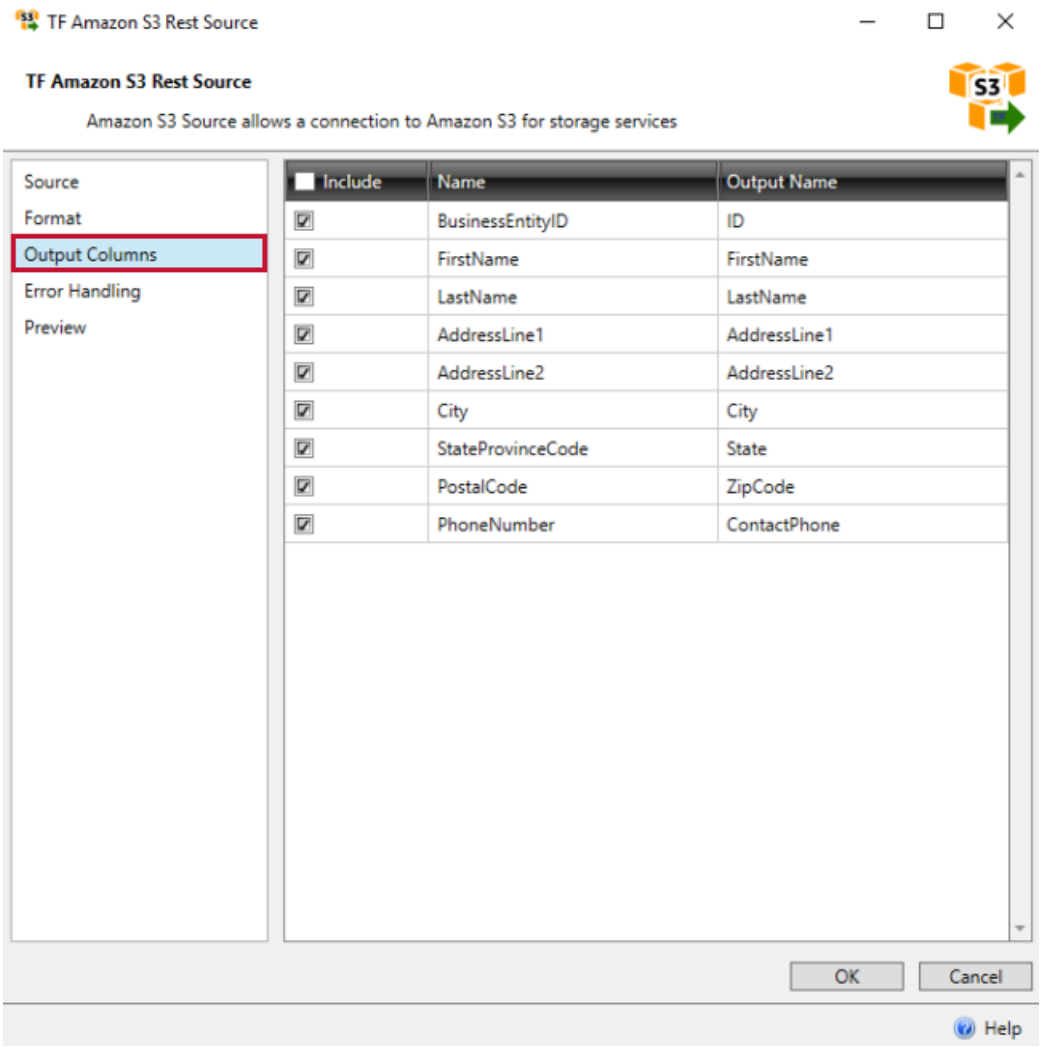
XML Format



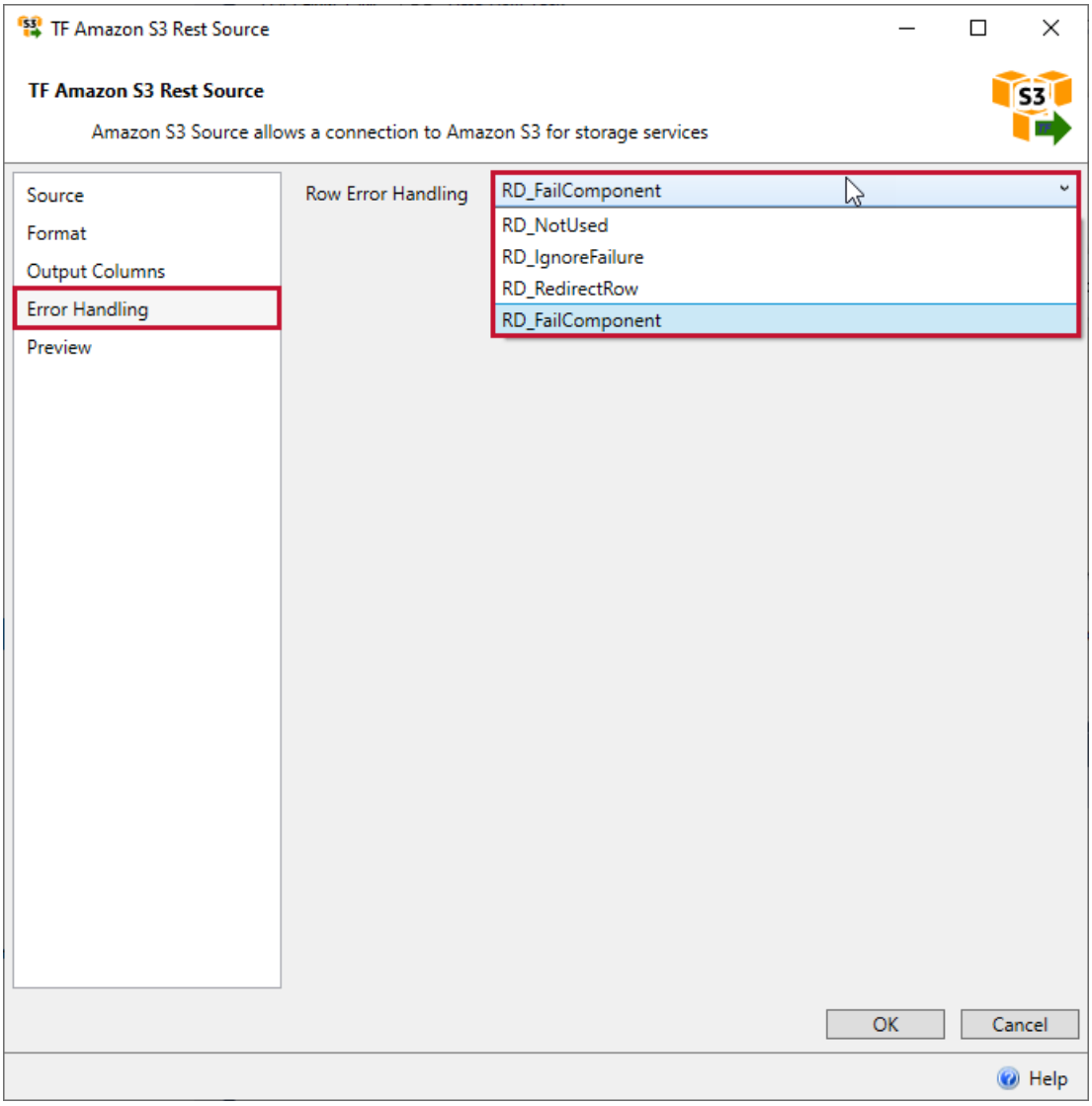
Option	Description
Root XPath Query	Xml queries can return multiple levels, therefore, this field Identifies the root to be used.
Namespaces:	• **Prefix** - Identifies the prefix used in an XML namespace. Example: In the element **** the prefix is **url**. • **Uri** - The url or address identified in the namespace.
Columns	Users can create, remove, and configure the name, index (zero-based), data type, length, precision, and scale of the columns being extracted from the file.

Output Columns

Users can select which columns to include/exclude in the output. Additionally, users can rename the columns by selecting into its corresponding output name.



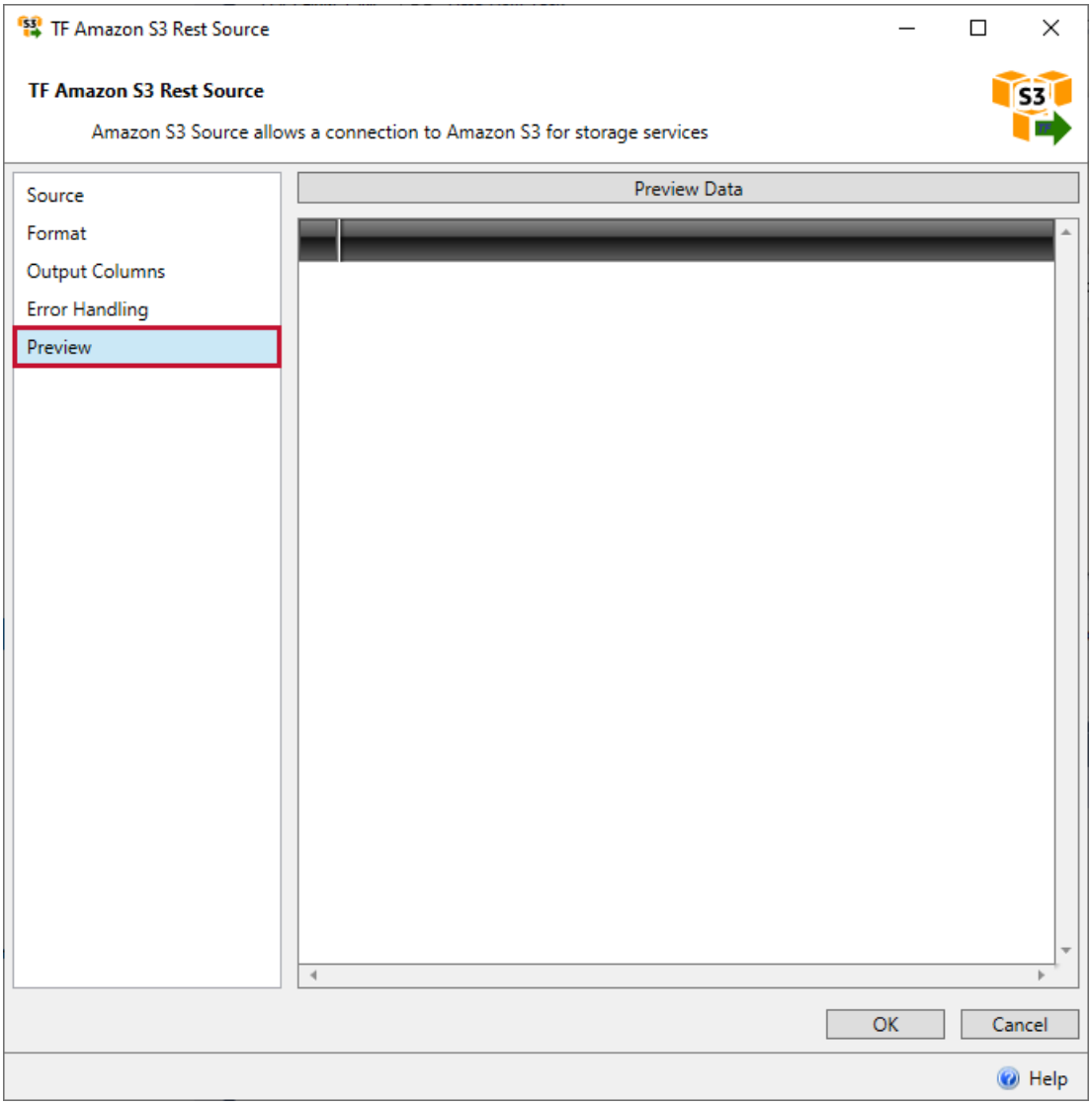
Error Handling



Option	Description
Not Used	Error handling is not used and no error constraint is available.
Ignore Failure	All errors are ignored and the package continues to execute.
Redirect Row	All error rows are directed to an error output.
Fail Component	(Default selection) On error, the component fails and the package execution stops.

Preview

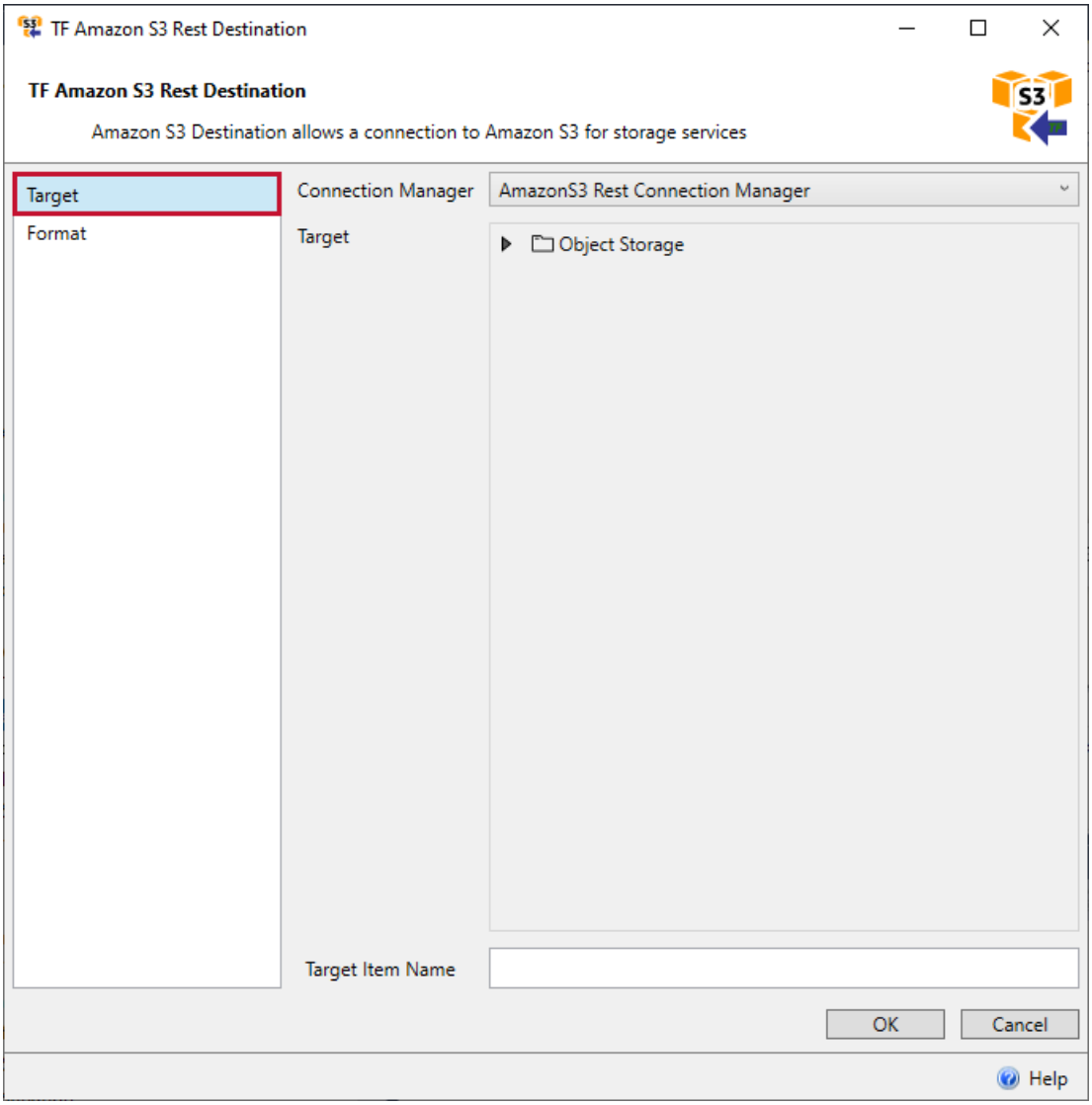
Select **Preview Data** after configuring the previous tabs to view an output sample.



AmazonS3 Rest Destination

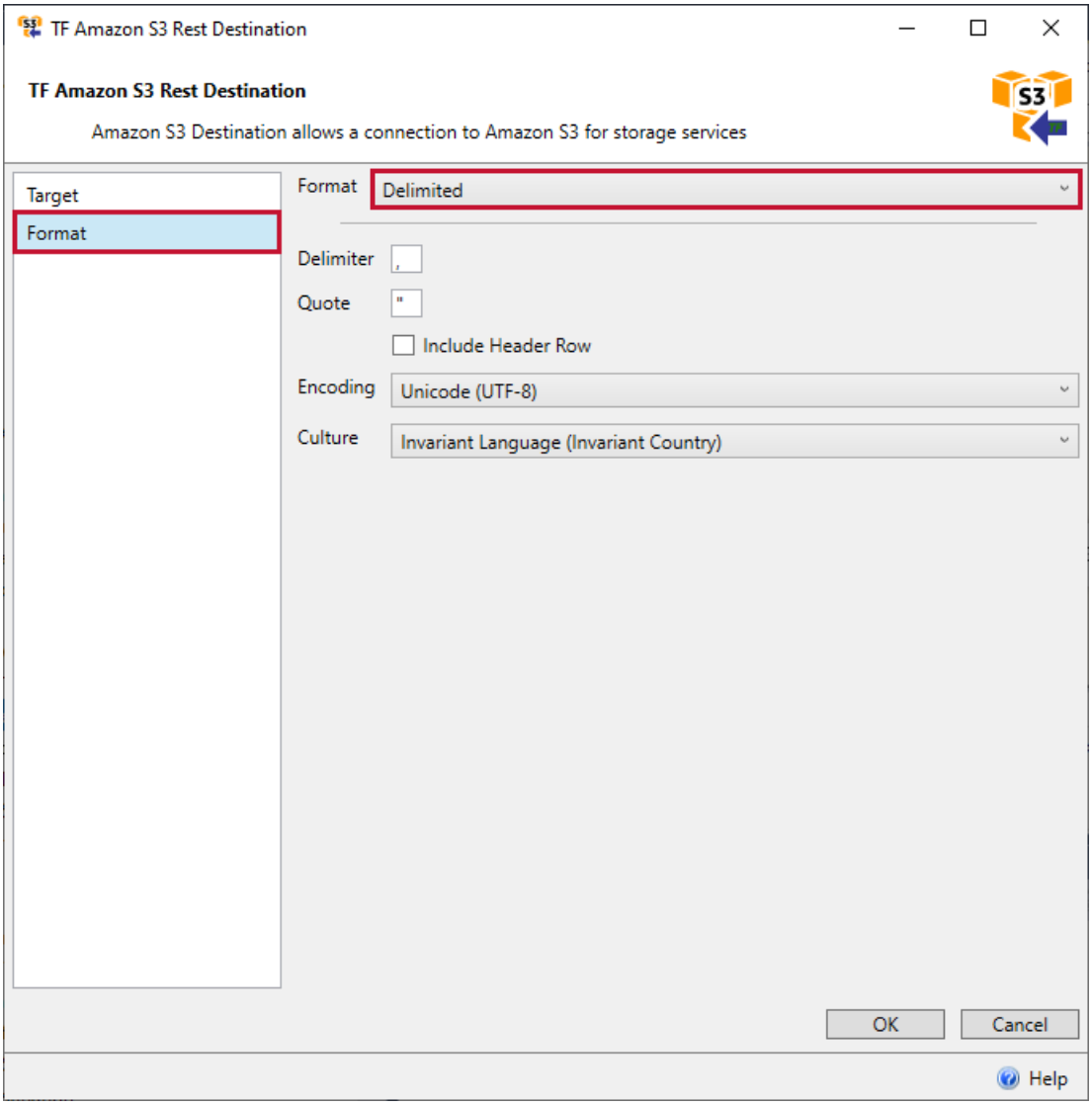
Target

Begin by creating a connection manager that connects to an Amazon Storage container. After a connection manager is created, the source window populates with files and folders. Select the desired file to continue configuration.



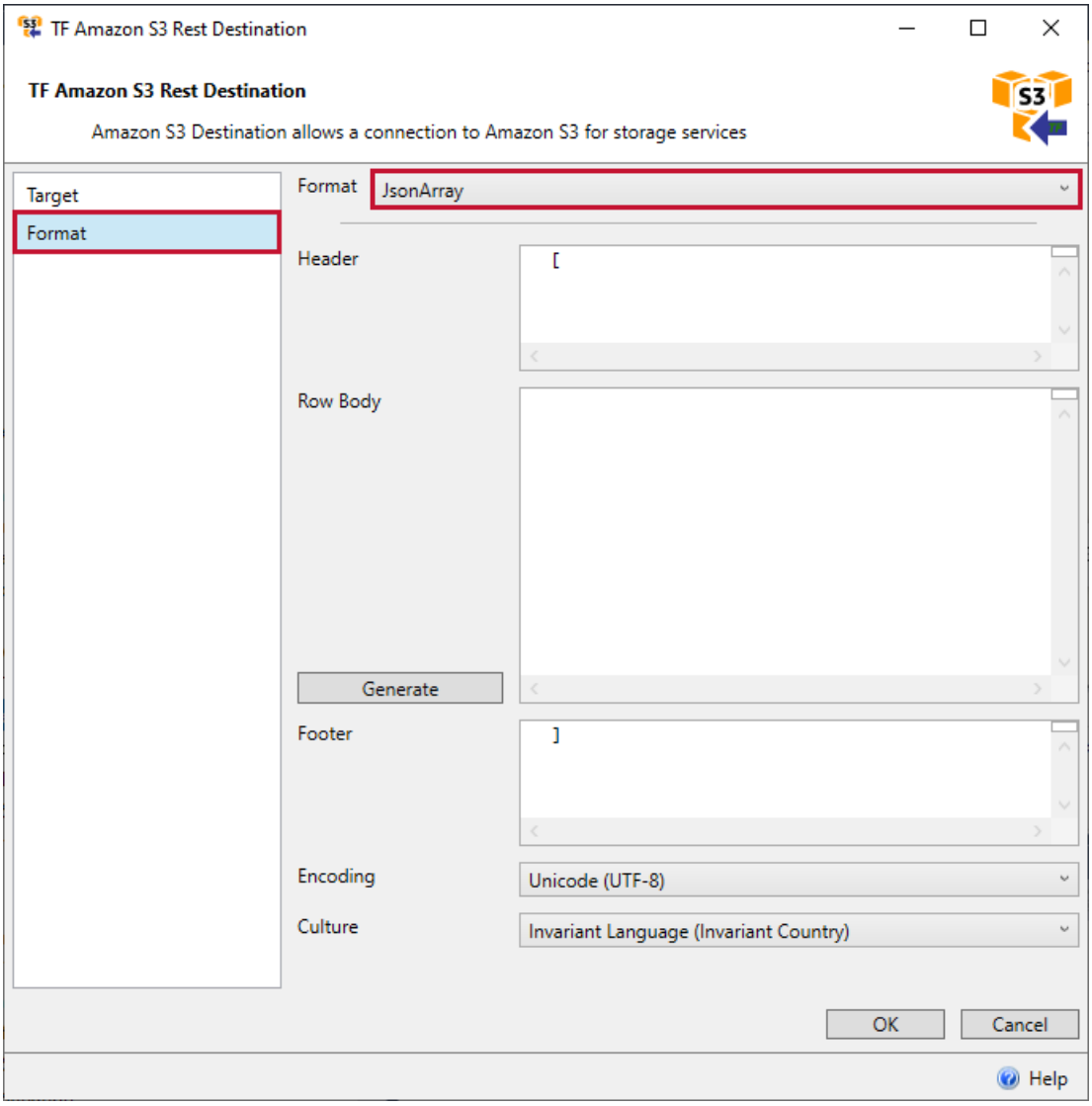
Option	Description
Target Item Name	Defines the name and extension of the file you would like to create when the component is executed. For example: **MyNewXmlList.xml**

Delimited Format



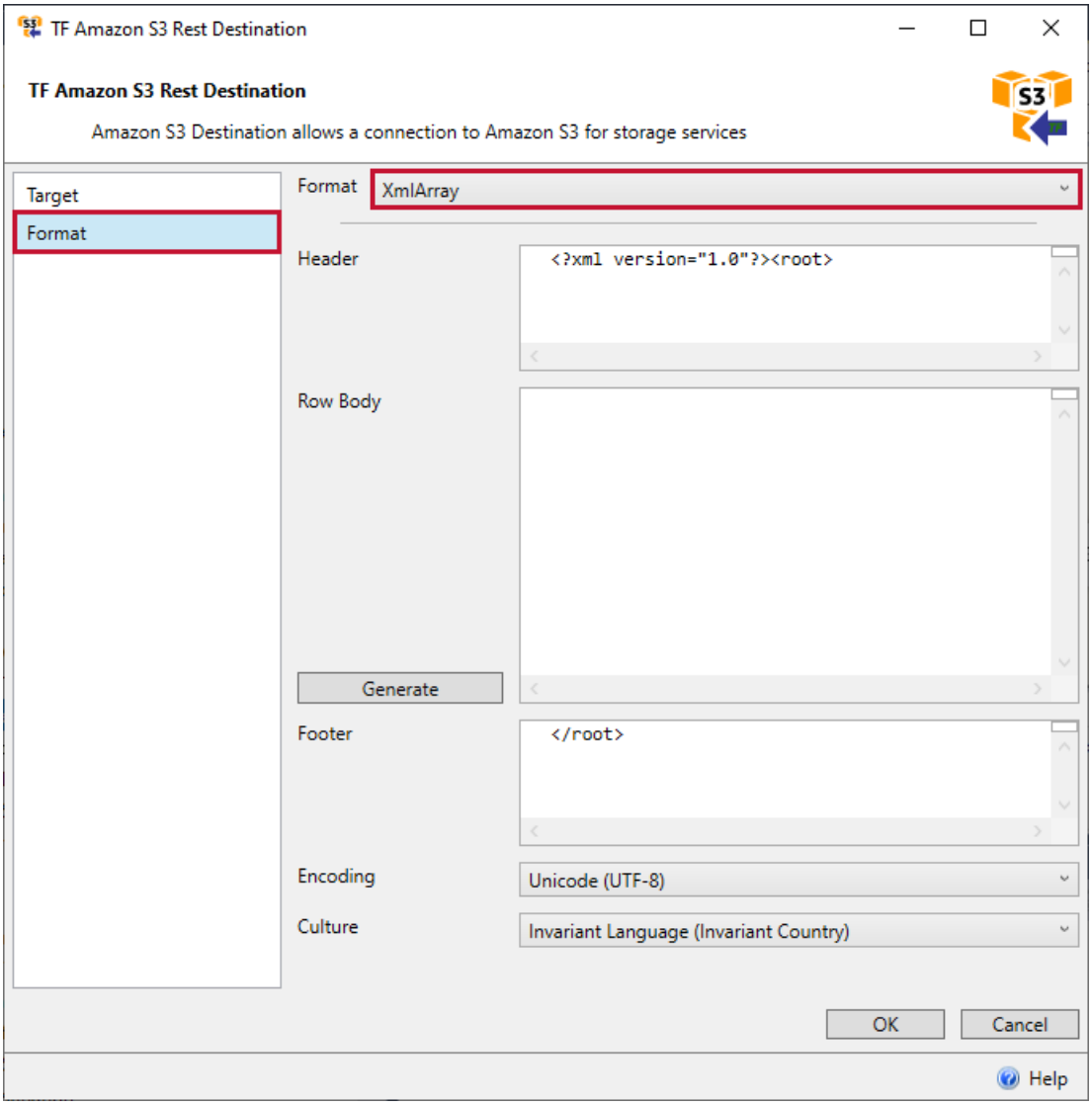
Option	Description
Delimiter	Identifies the character that separates text.
Quote	Identifies the character (single or double quote) that indicates value.
Include Header Row	Writes the column headers at the beginning of the document.
Encoding	Selects which encoding is used to create the document.
Culture	Used to identify the document's language culture.

Json Array Format



Option	Description
Generate	Select to automatically generate the Row Body metadata. Users can also add custom header and footer data.
Encoding	Select which encoding is used to create the document.
Culture	Used to identify the document's language culture.

XML Array Format



Option	Description
Generate	Select to automatically generate the Row Body metadata. (Users can also add custom header and footer data.)
Encoding	Select which encoding is used to create the document.
Culture	Used to identify the document's language culture.

Task Factory Azure

Last Modified on 03 March 2022

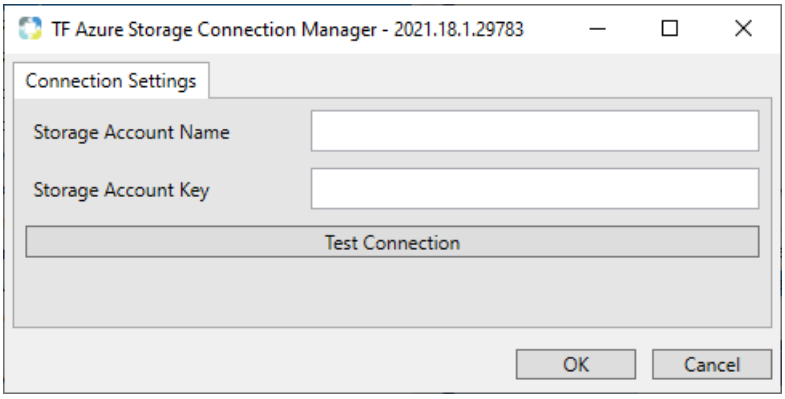
Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Note: Task Factory components can be used with Azure databases. As of the 2018.2.3 release, Task Factory can also be used with Azure Data Factory.

Note: Azure Storage connection manager is available for SQL versions 2012 and higher.

Azure Storage Connection Manager

The **Azure Storage Connection Manager** is used to connect to an Azure Machine Learning blob storage.

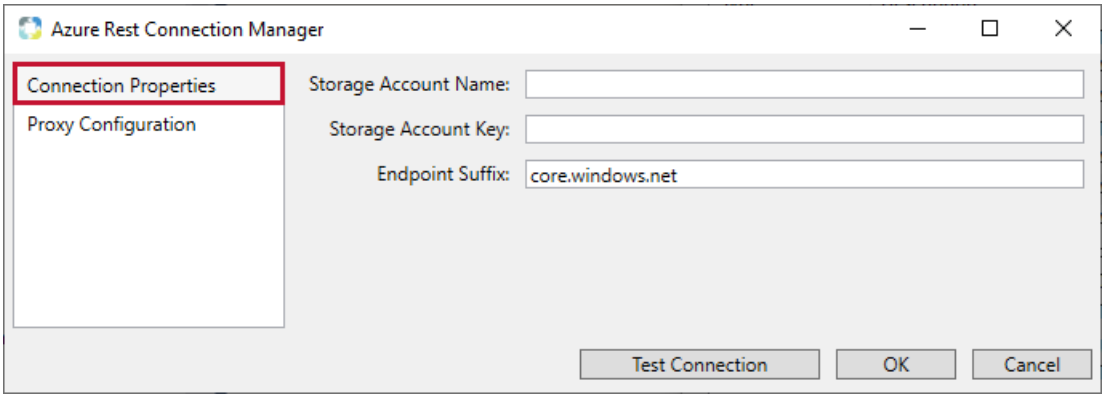


Option	Description
Storage Account Name	The username used to login to an Azure ML account.
Storage Account Key	The key generated within the Azure ML account.

Azure Rest Connection Manager

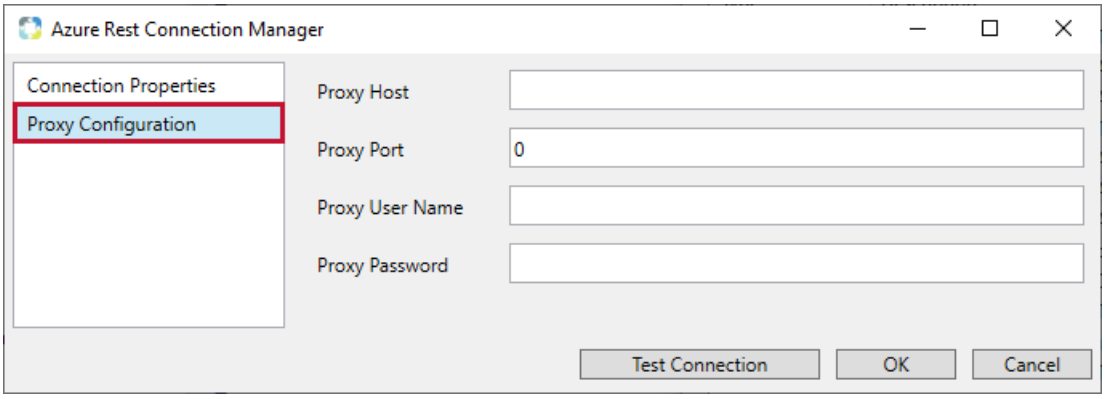
Connection Properties

Used with the Azure Rest Source and Azure Rest Destination.



Option	Description
Storage Account Name	Storage account name provided by Azure to connect to the desired container.
Storage Account Key	The account key provided by Azure.
Endpoint Suffix	Endpoint suffix provided by Azure. Default: core.windows.net

Proxy Configuration

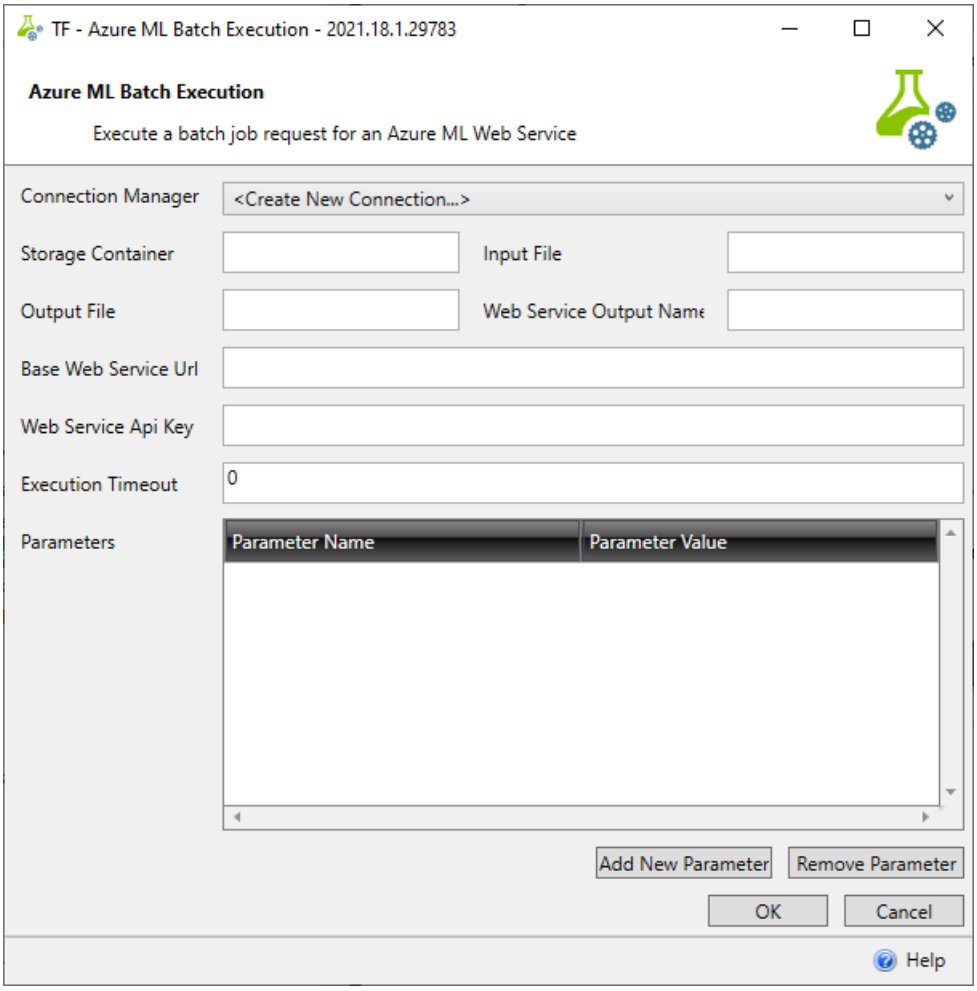


Option	Description
Proxy Host	Identifies the proxy address.
Proxy Port	Identifies the port used by the proxy.
Proxy User Name	Enter the user name for proxies that require user authentication.
Proxy Password	Enter the password for proxies that require user authentication.

Note: Azure ML Batch Execution is available for SQL versions 2012 and higher.

Azure ML Batch Execution Task

Task Icon	Task Description
	The **Azure ML Batch Execution Task** is used to execute a batch job for an Azure ML web service. See the Azure Storage Connection Manager to learn more about setting up the connection manager.

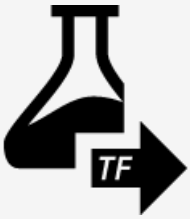


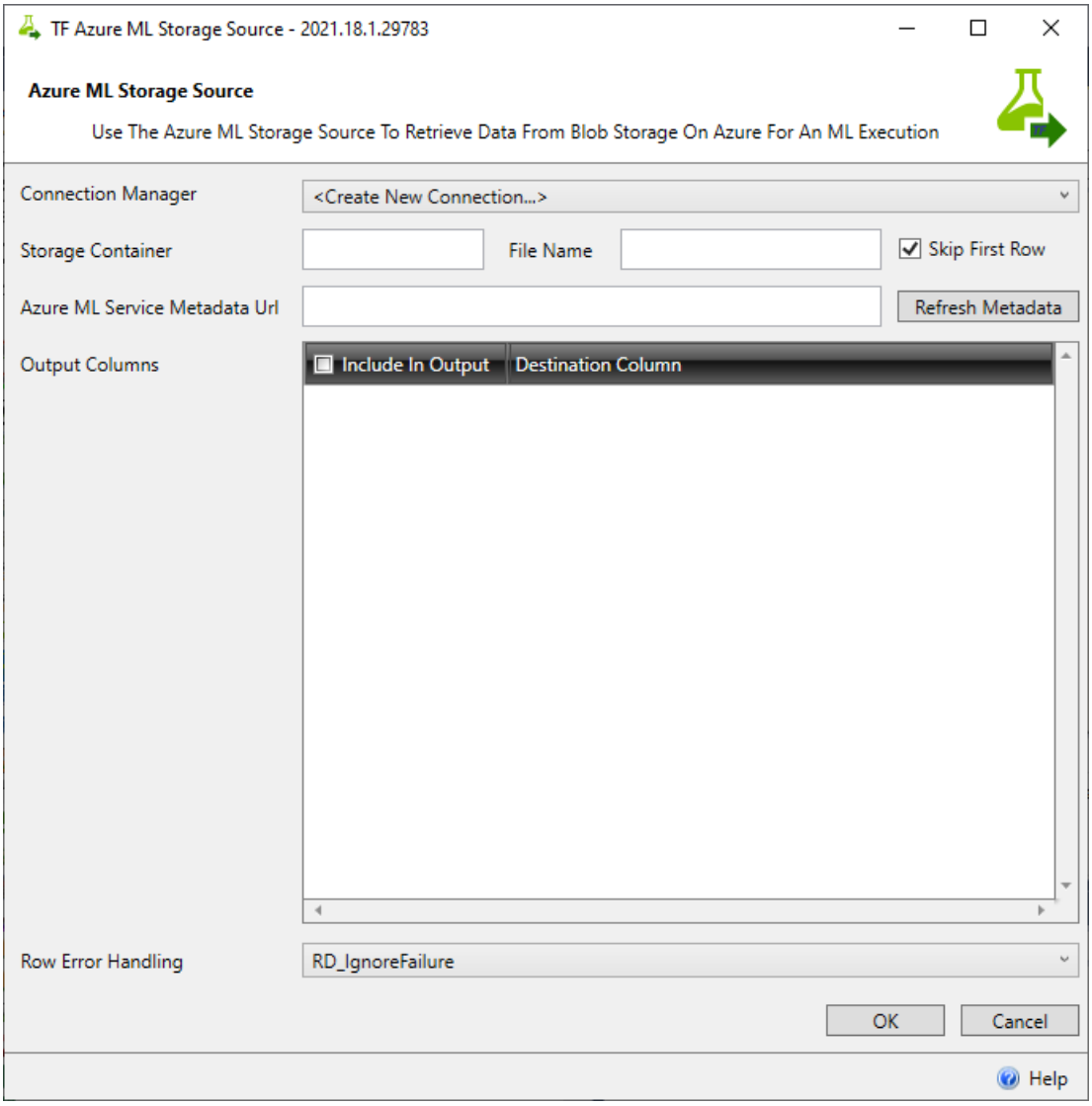
Function	Description
Storage Container	Identifies the container within Azure storage where the input and output files are stored.
Input File	The name of the source file located in the Azure storage container.
Output File	The name of the destination file located in the Azure storage container. **Note:** This does not need to exist.

Function	Description
Web Service Output Name	The name of the web service output defined in the Azure ML experiment web service.
Base Web Service Url	The request Url defined in the batch execution api documentation for your web service. To obtain this Url, users need to do the following in Azure ML Studio: • Select the web service group. • Select the Batch Execution URL in the default endpoint. • Select the Request URL until the end of the word **jobs** and copy.
Execution Timeout	The amount of time in seconds before the component sends a cancel command.
Parameters	If users have a web service with parameters, this is where to define them.

Note: Azure ML is available for SQL versions 2012 and higher.

Azure ML Storage Source

Task Icon	Task Description
	The **Azure ML Storage Source** retrieves data from blob storage on Azure for an ML web service. See the Azure Storage Connection Manager to learn more about setting up the connection manager.

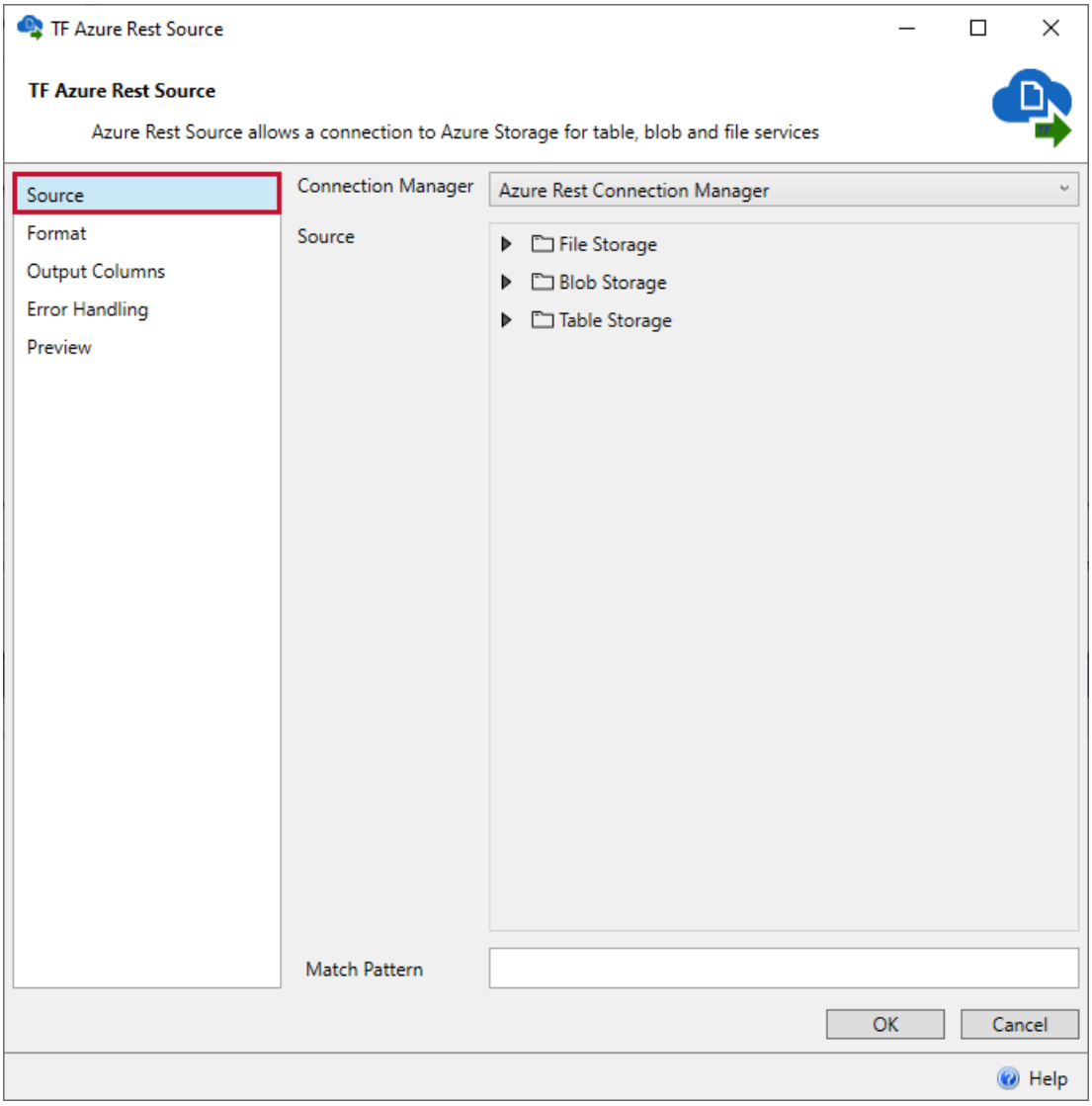


Function	Description
Storage Container	Identifies the container within Azure storage where the data will be retrieved.
File Name	The name of the .csv file located in the Azure storage container.
Skip First Row	This option is selected by default and used if the first row of data contains headers.

Function	Description
Azure ML Service Metadata URI	The base web service URI with the bold parameters added to the base URL (e.g.**https://ussouthcentral.services.azureml.net/odata/workspaces/3b47fa82350146aaa16a80d10ed01ba7/services/4171be6bcb584f20afa396ab8590196c/$metadata**.) To obtain the URI, users need to do the following in Azure ML Studio: • Select the web service group. • Select the Batch Execution URI in the default endpoint. • Select the Request URI up to the word **jobs** and copy (see screenshot below): Submit (but not start) a Batch Execution job Request Method Request URI POST https://ussouthcentral.services.azureml.net/workspaces/3b47fa82350146aaa16a80d10ed01ba7/services/4171be6bcb584f20afa396ab8590196c/jobs?api-version-2.0 **Do not include the highlighted section of the URL.**
Refresh Metadata	Selecting the button updates the **Destination Column** window. By default, all columns are included in the output.
Row Error Handling	**Additional Information:** See the Task Factory Error Row Handling article for more information about this functionality.

Azure Rest Source

Source



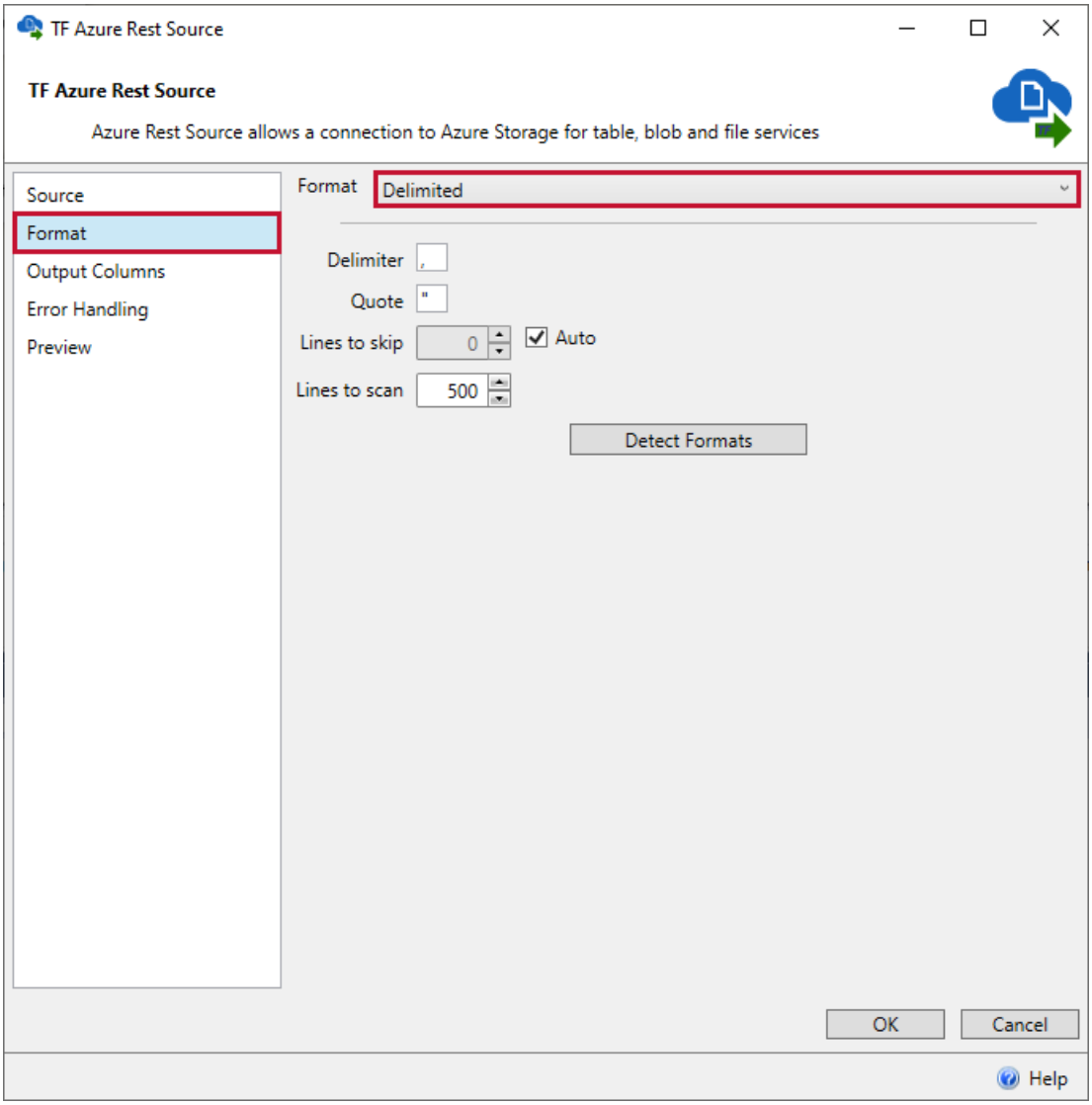
Begin by creating a connection manager that connects to the service's storage container.

Format

The following formats are available:

Delimited Format

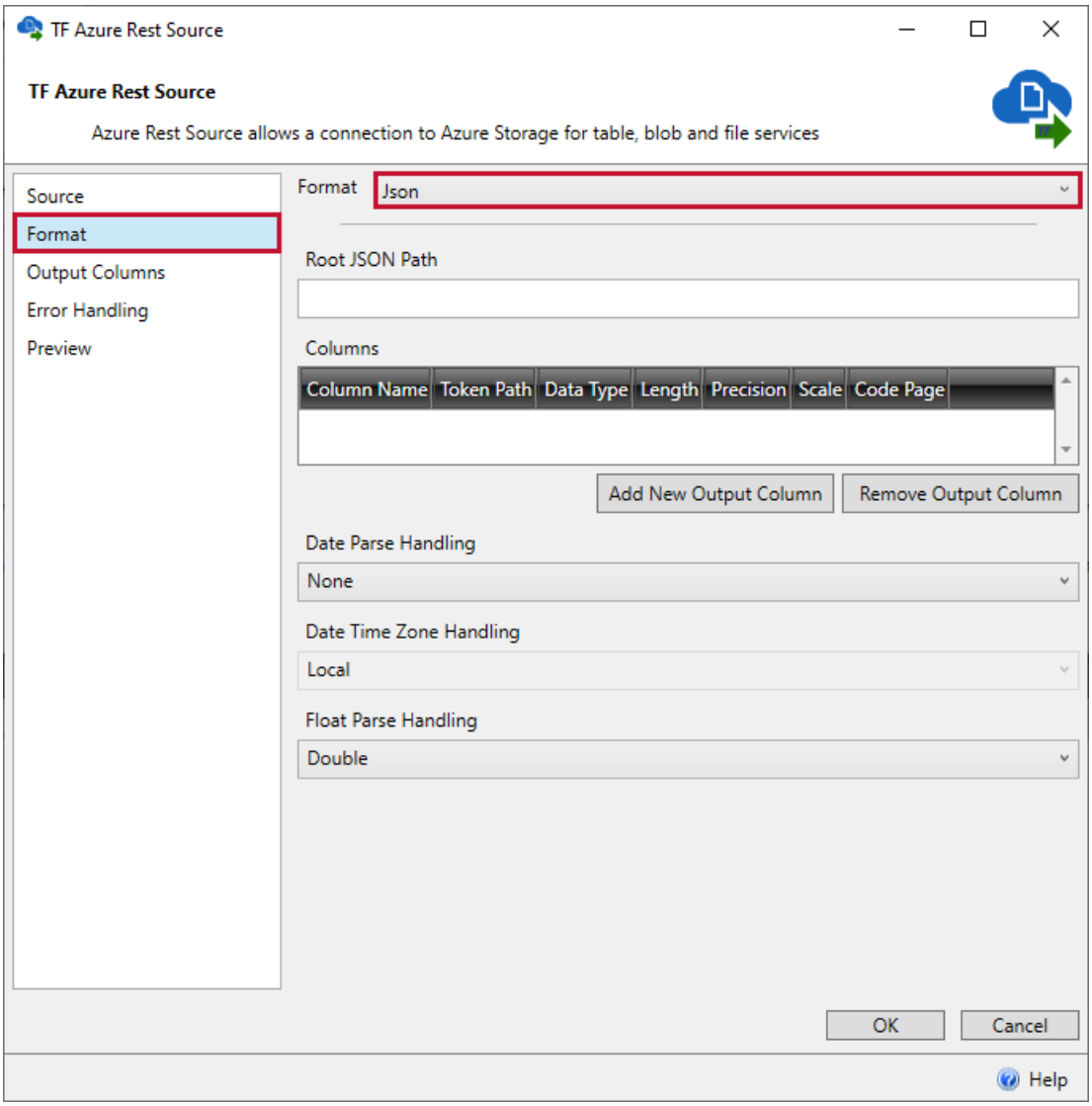
Users can manually configure the delimiter, quote identifiers, the lines to skip, and lines to scan. Select **Detect Formats** to configure these fields automatically.



Option	Description
Delimiter	Identifies the character that separates columns. The default value is the comma (**,**).
Quote	Identifies the character used to indicate a value (if present).
Lines To Skip	Users can designate the number of lines to skip at the beginning of the delimited file.
Auto	Automatically detects which lines to skip.
Lines To Scan	Determines the number of lines to scan when detecting the file.
Detect Formats	Select this button to begin scanning the chosen file.

Detected Delimited View - After the format is detected, a new view appears that allows users to configure or change different properties of the file.

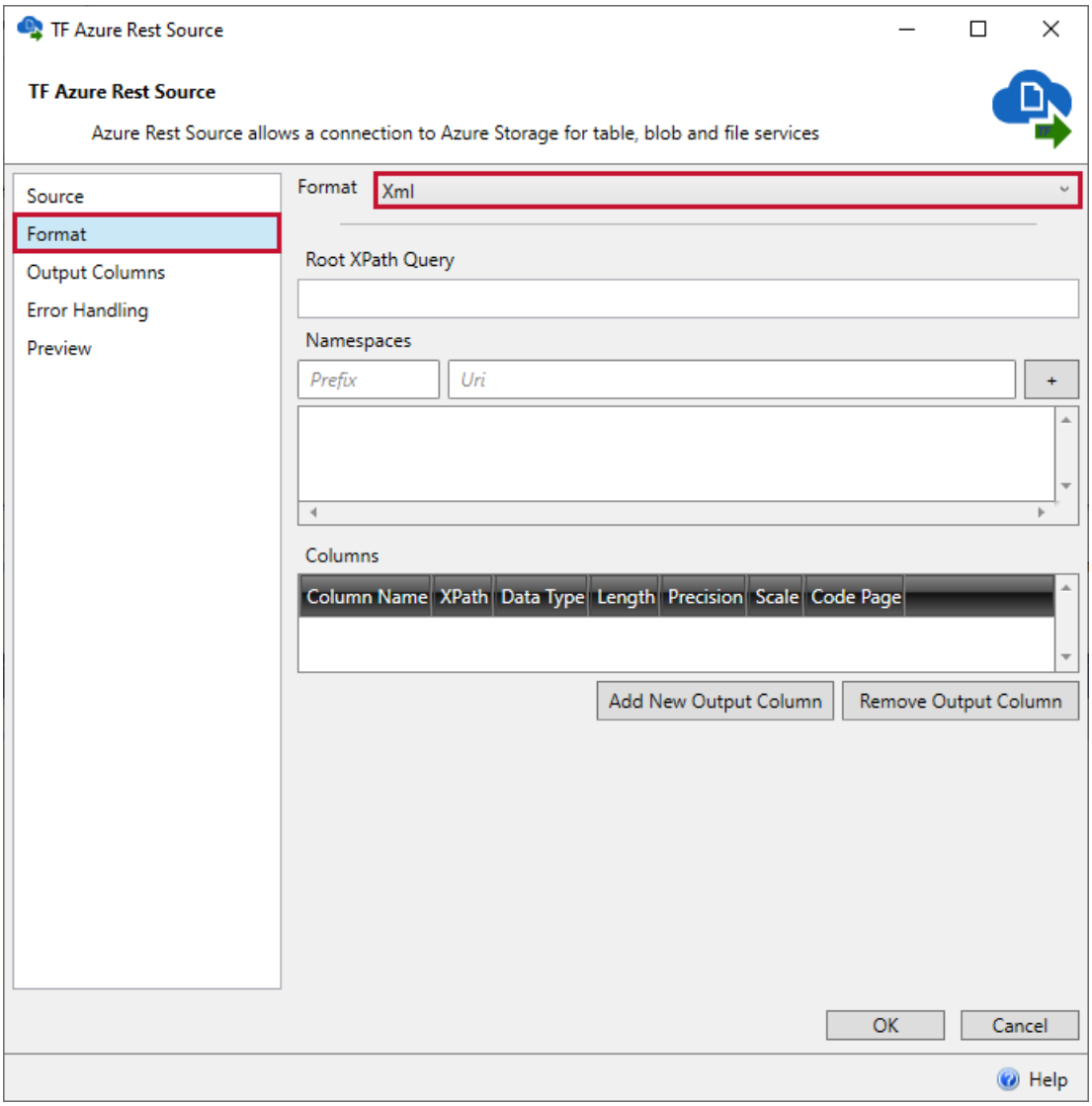
JSON Format



Option	Description
Root Json Path	JSON queries can return multiple levels, therefore, this field Identifies the root to be used.
Output Columns	In this window, users can add and remove columns, define their name, token path (defined within the raw data), data type, length, precision, scale, and code page.
Date Parse Handling	Specifies how date formatted strings are parsed when reading JSON text. • **None** - Strings are not parsed as a date type and are read as strings. • **DateTime** - Date formatted strings are parsed as DateTime. • **DateTime Offset** - Date formatted strings are parsed as DateTimeOffset.

Option	Description
Date Time Zone Handling	Specifies how to treat time values converted between string and DateTime. • **Local** - Treats the time as local (UTC times convert to local times.) • **UTC** - Treats the time as UTC. This converts local time to UTC. • **Unspecified** - Treats the time as local time if a DateTime is being converted to a string. If a string is being converted to DateTime, convert to a local time if a time zone is specified. • **RoundtripKind** - Time zone is preserved when converting.
Float Parse Handling	Specifies how floating point numbers are parsed. • **Double** - Floating point numbers are parsed as a double datatype. • **Decimal** - Floating point numbers are parsed as a decimal datatype.

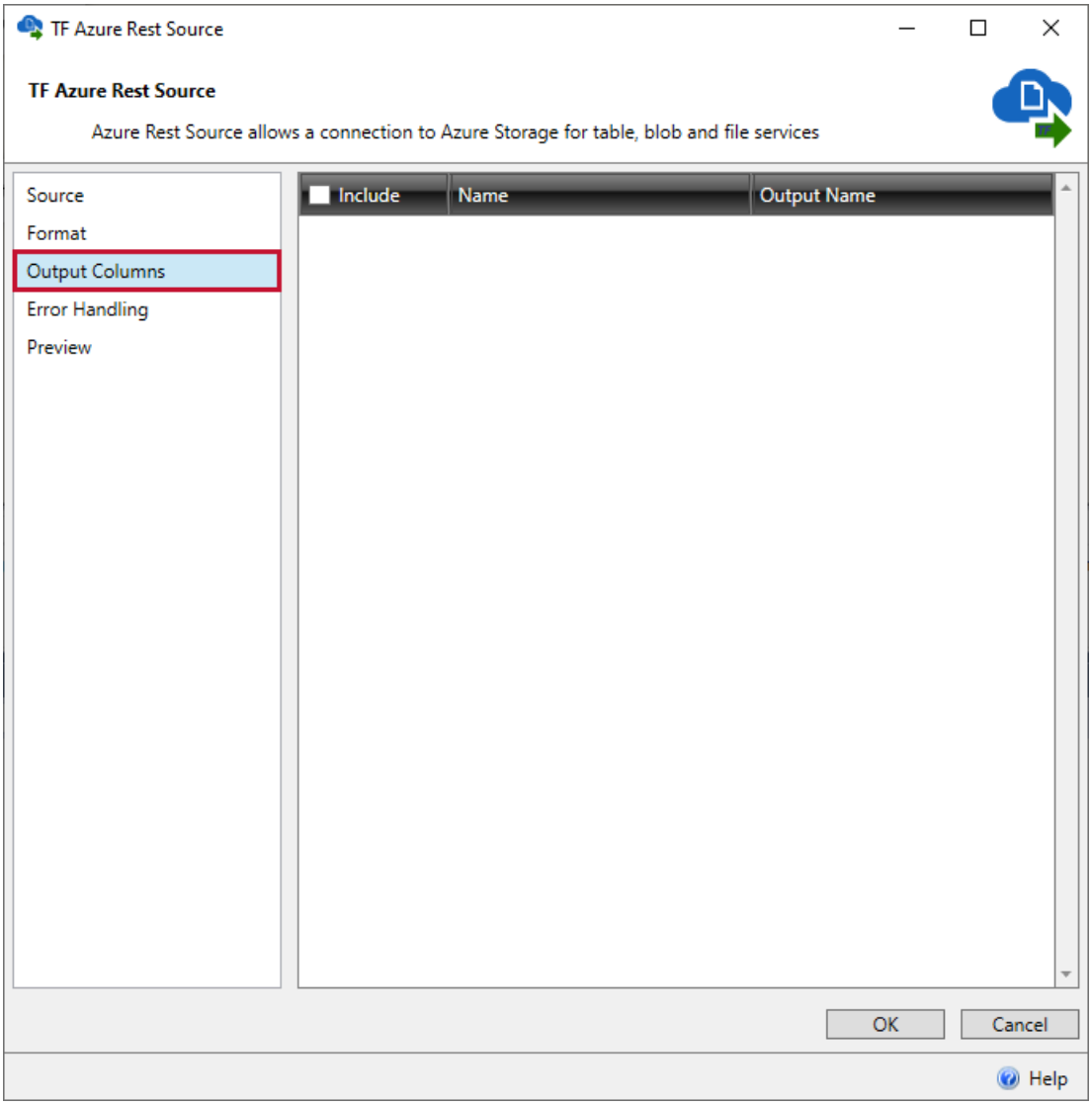
XML Format



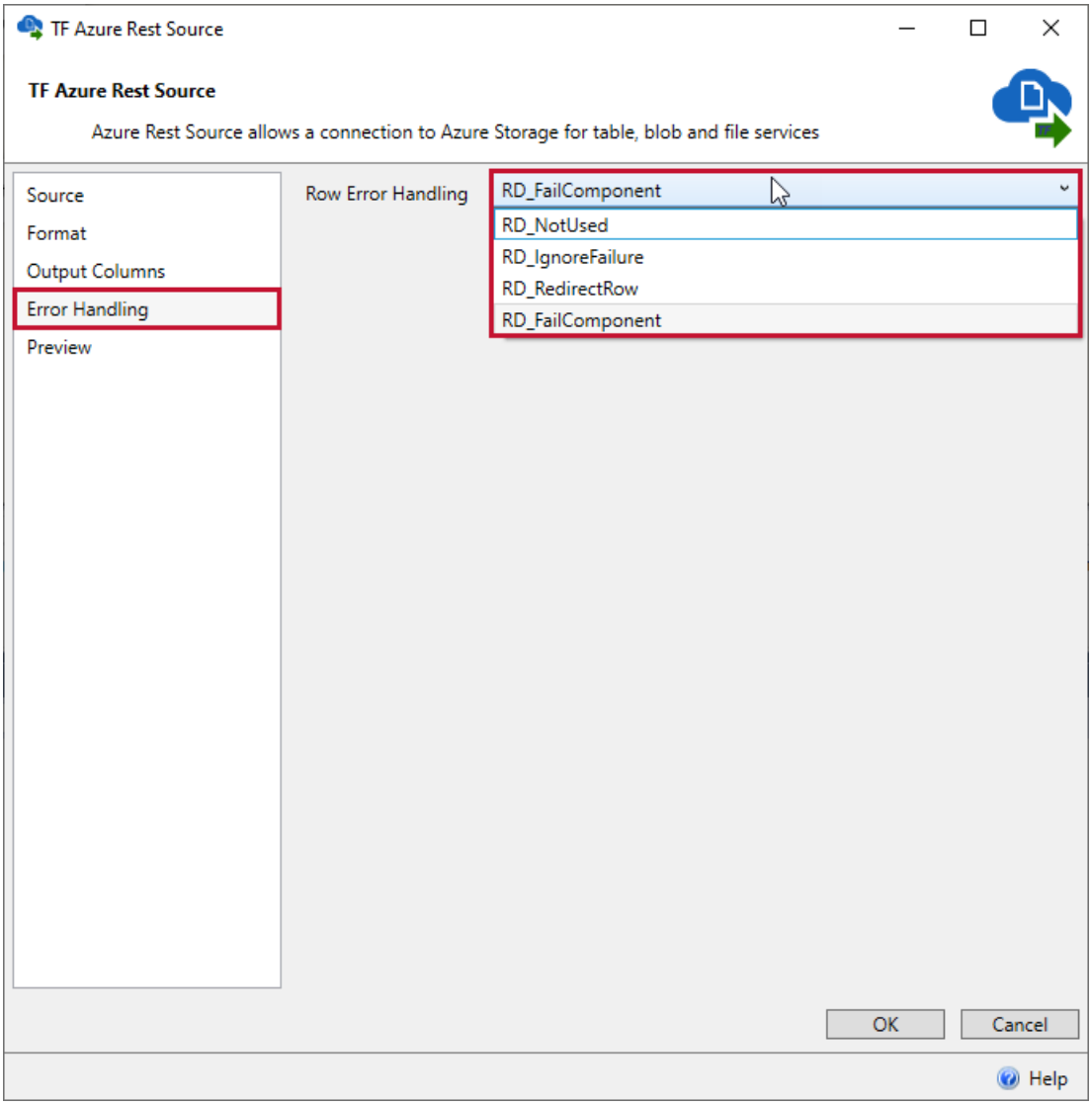
Option	Description
Root XPath Query	XML queries can return multiple levels, therefore, this field Identifies the root to be used.
Namespaces:	• **Prefix** - Identifies the prefix used in an xml namespace. Example: In the element **\> the prefix is url.** • **Uri - The url or address identified in the namespace.**
Columns	Users can create, remove, and configure the name, index (zero-based), data type, length, precision, and scale of the columns being extracted from the file.

Output Columns

Users can select which columns to include/exclude in the output. Additionally, users can rename the columns by selecting into its corresponding output name.



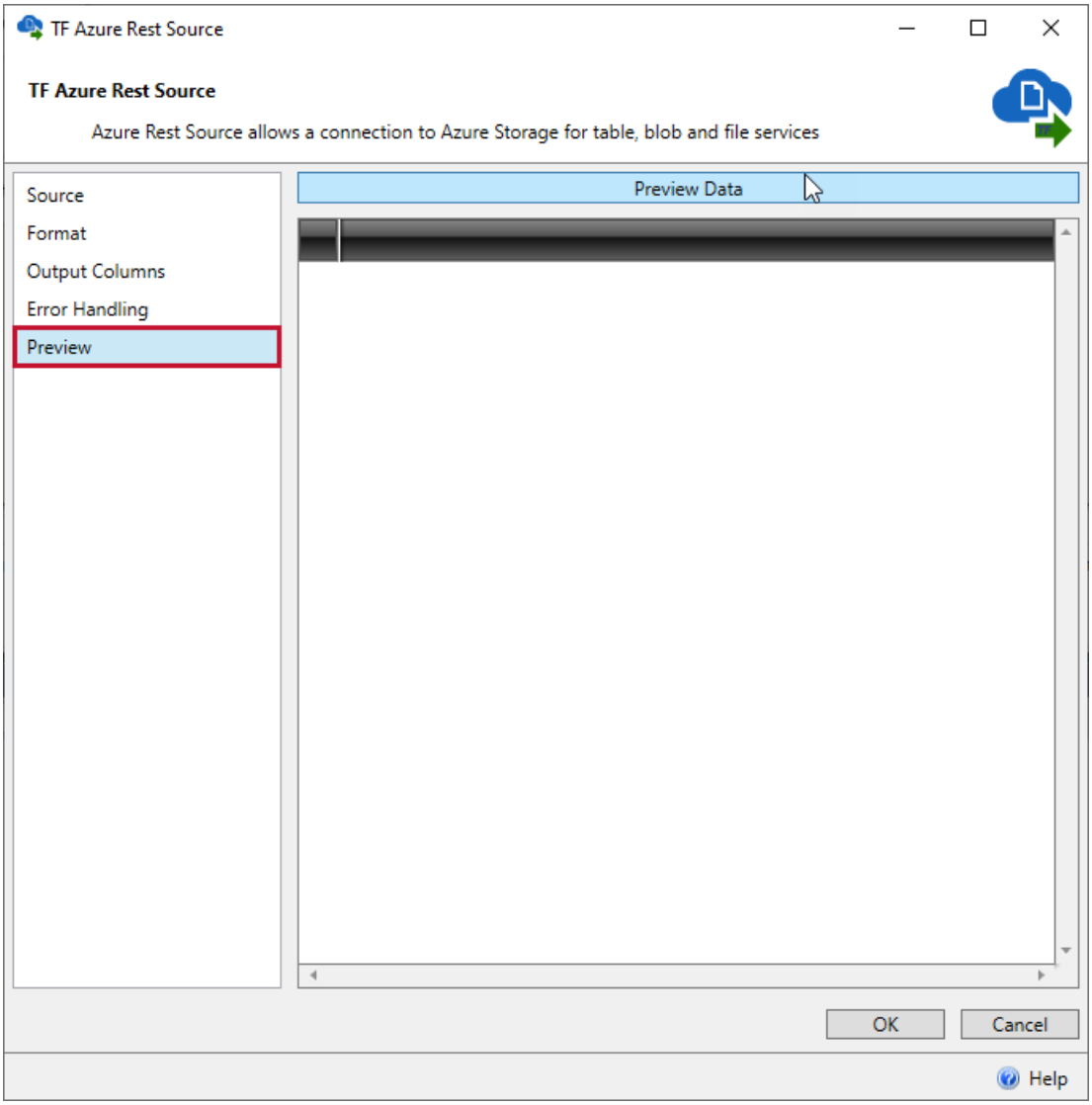
Error Handling



Option	Description
Not Used	Error handling is not used and no error constraint is available.
Ignore Failure	All errors are ignored and the package continues to execute.
Redirect Row	All error rows are directed to an error output.
Fail Component	(Default selection) On error, the component fails and the package execution stops.

Preview

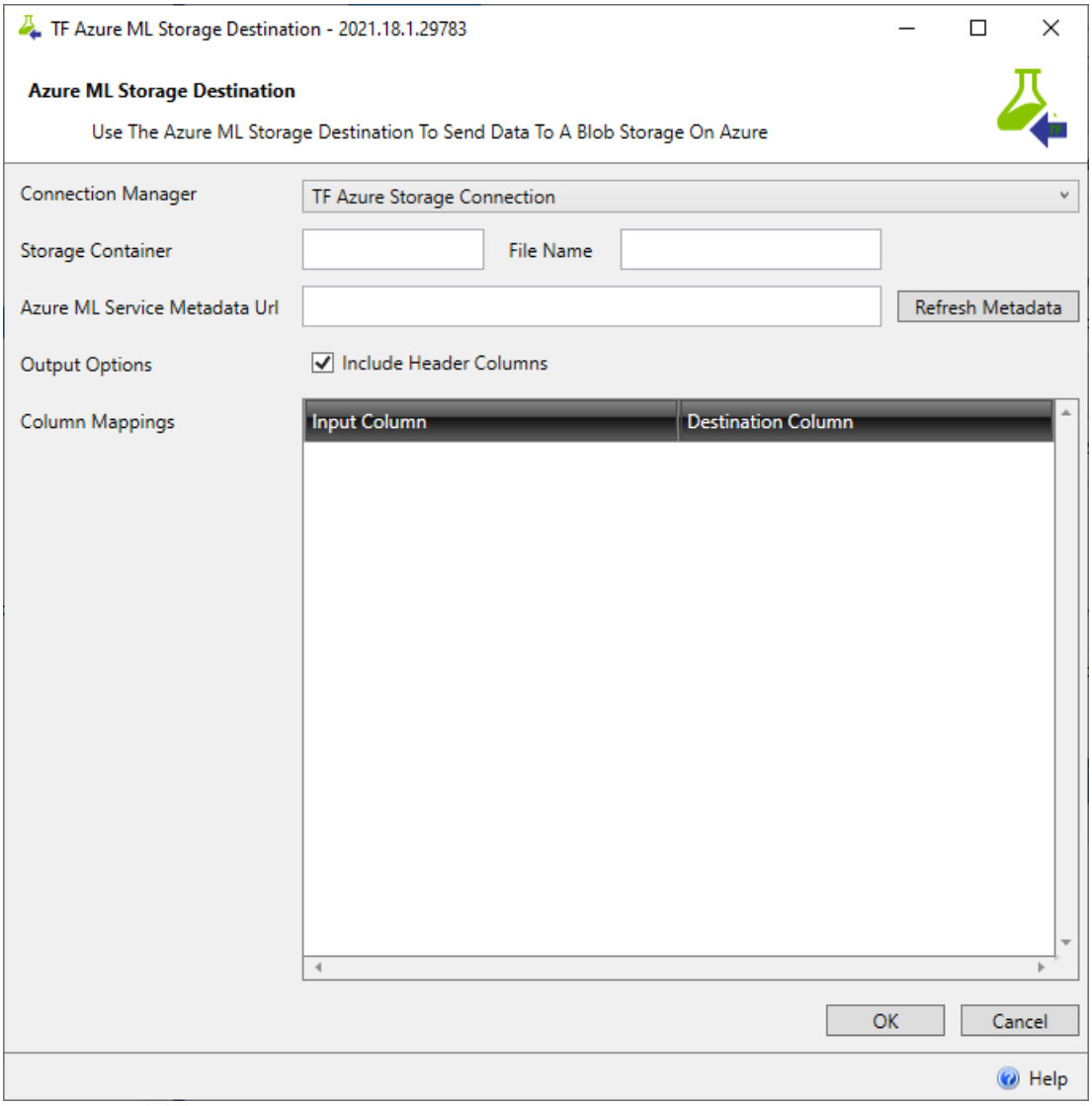
Select **Preview Data** after configuring the previous tabs to view an output sample.



Note: Azure ML is available for SQL versions 2012 and higher.

Azure ML Destination

Destination Icon	Destination Description
	The **Azure ML Destination** is used to send data to a blob storage on Azure storage to be consumed by an Azure ML web service. This is the input file of the Azure ML batch execution task. See the Azure Storage Connection Manager to learn more about setting up the connection manager.



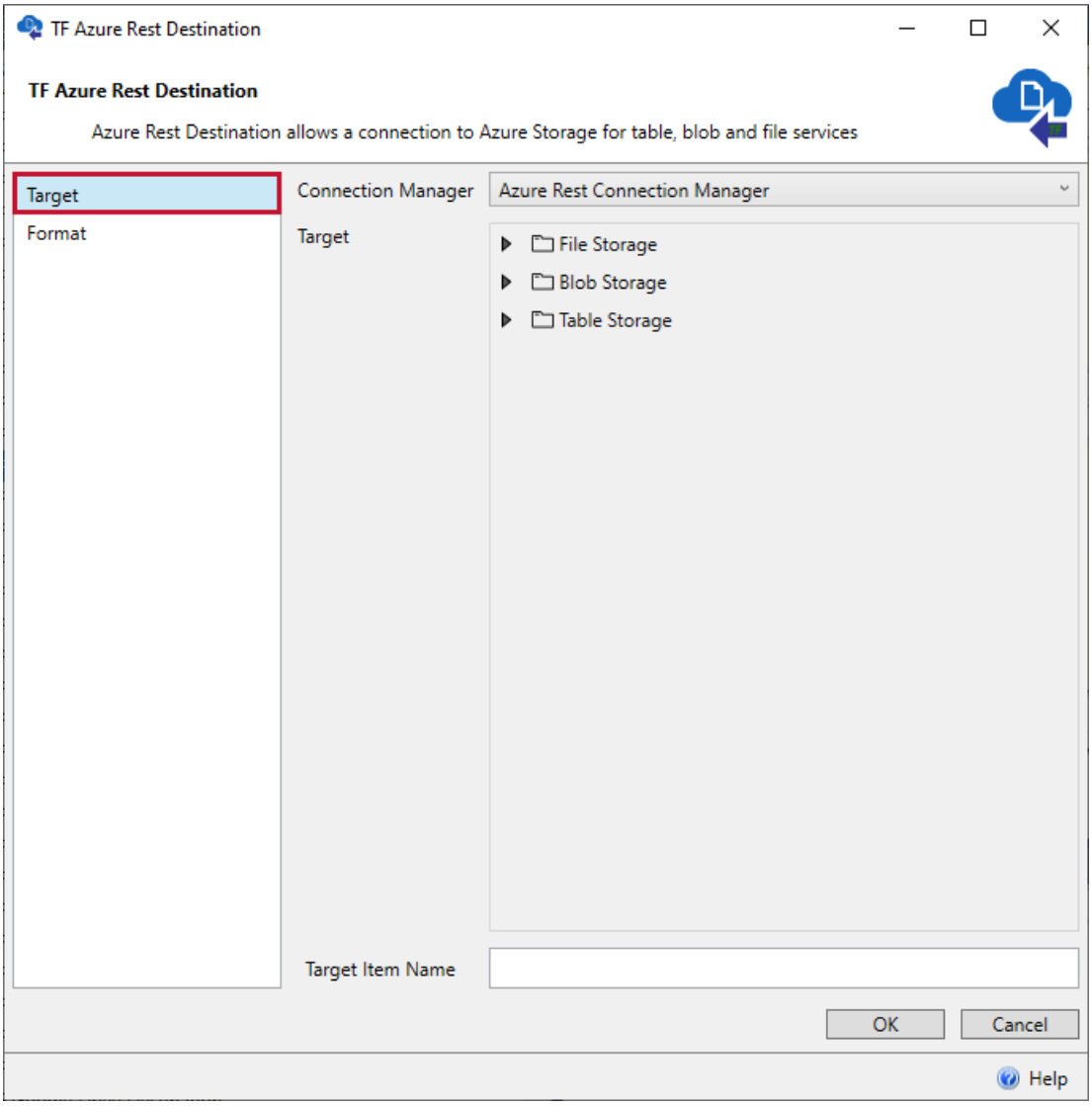
Function	Description
Storage Container	Identifies the container within Azure storage where the data is stored.
File Name	The name of the file in the Azure storage container.

Function	Description
Azure ML Service Metadata Url	The base web service URL with the bold parameters added to the base URL (e.g. **https://ussouthcentral.services.azureml.net/odata/workspaces/3b47fa82350146aaa16a80d10ed01ba7/services/4171be6bcb584f20afa396ab8590196c/$metadata**.) To obtain the Url, users need to do the following in Azure ML Studio: • Select the web service group. • Select the Batch Execution URL in the default endpoint. • Select the Request Uri up to the word "**jobs**" and copy (see screenshot below): Submit (but not start) a Batch Execution job Request Method Request URI POST https://ussouthcentral.services.azureml.net/workspaces/3b47fa82350146aaa16a80d10ed01ba7/services/4171be6bcb584f20afa396ab8590196c/jobs?api-version-2.0 **Do not include the highlighted section of the URL.**
Output Options	This selection includes header columns in the output by default.
Refresh Metadata	Selecting this button updates the **Input** and **Destination** columns window (input and destination columns with the same names are auto-mapped.)

Azure Rest Destination

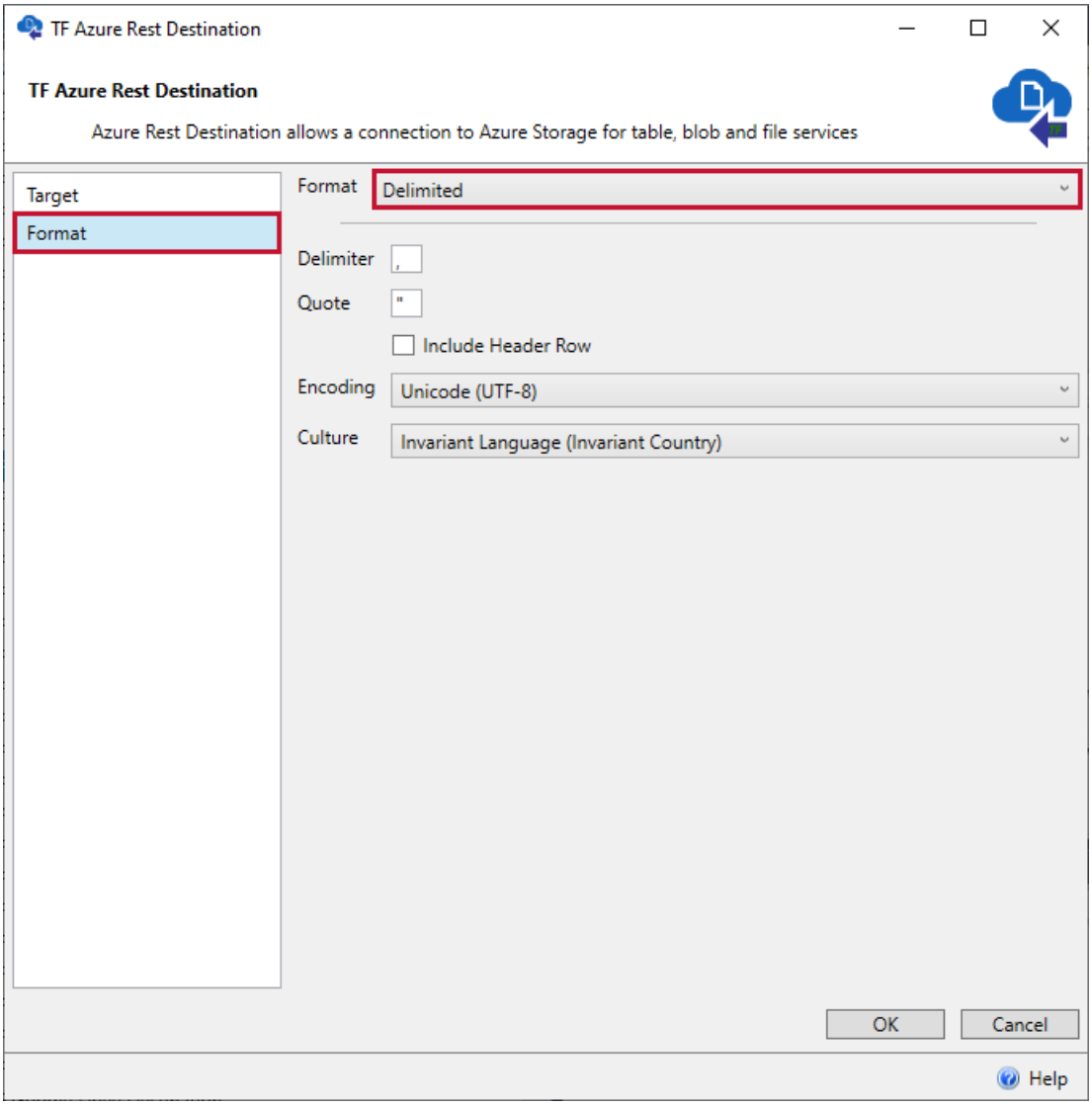
Target

Begin by creating a connection manager that connects to an Azure Storage container. After a connection manager is created, the source window populates with files and folders. Select the desired file to continue configuration.



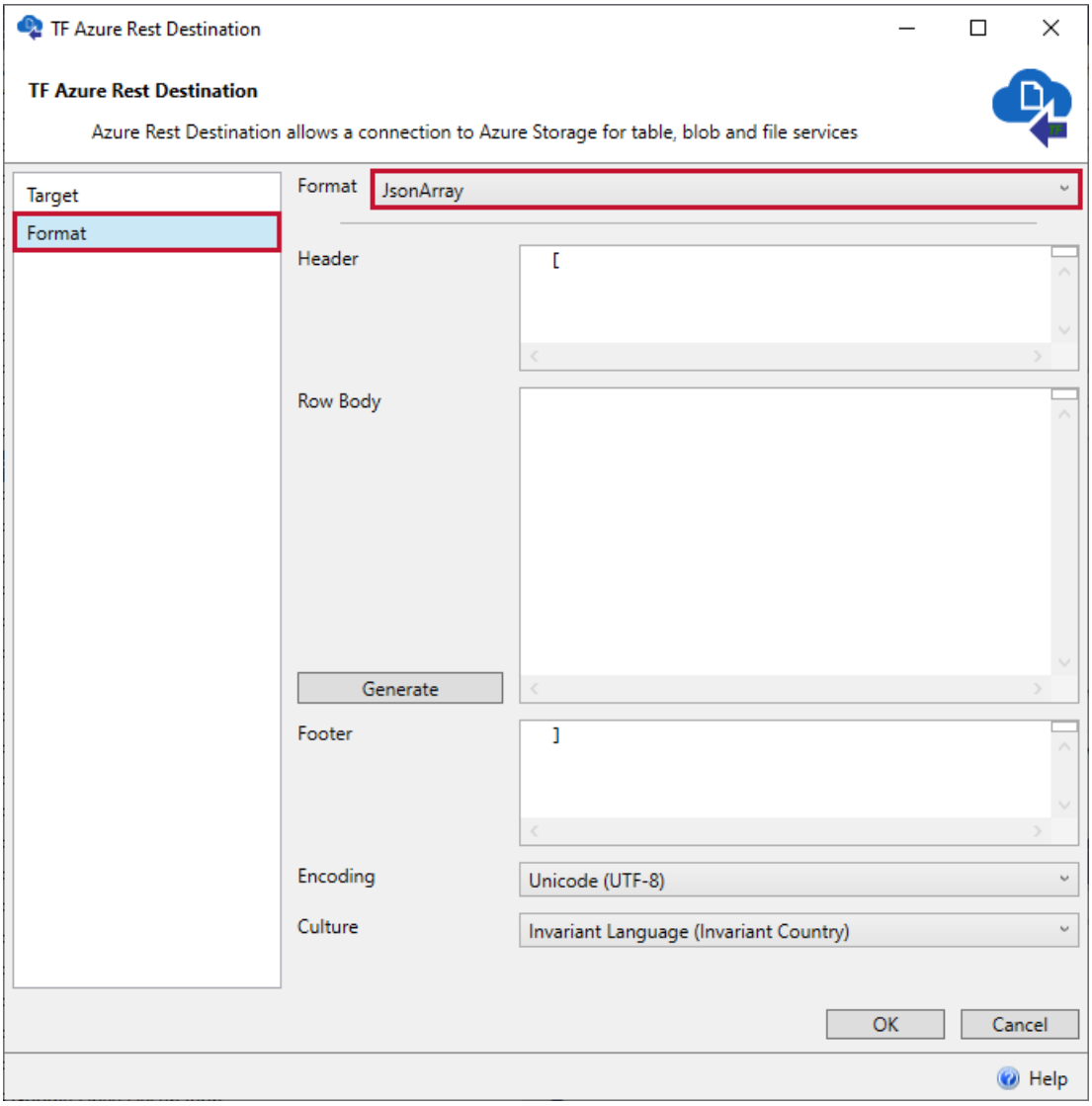
Option	Description
Target Item Name	Defines the name and extension of the file you would like to create when the component is executed. (For example: **MyNewXmlList.xml**)

Delimited Format



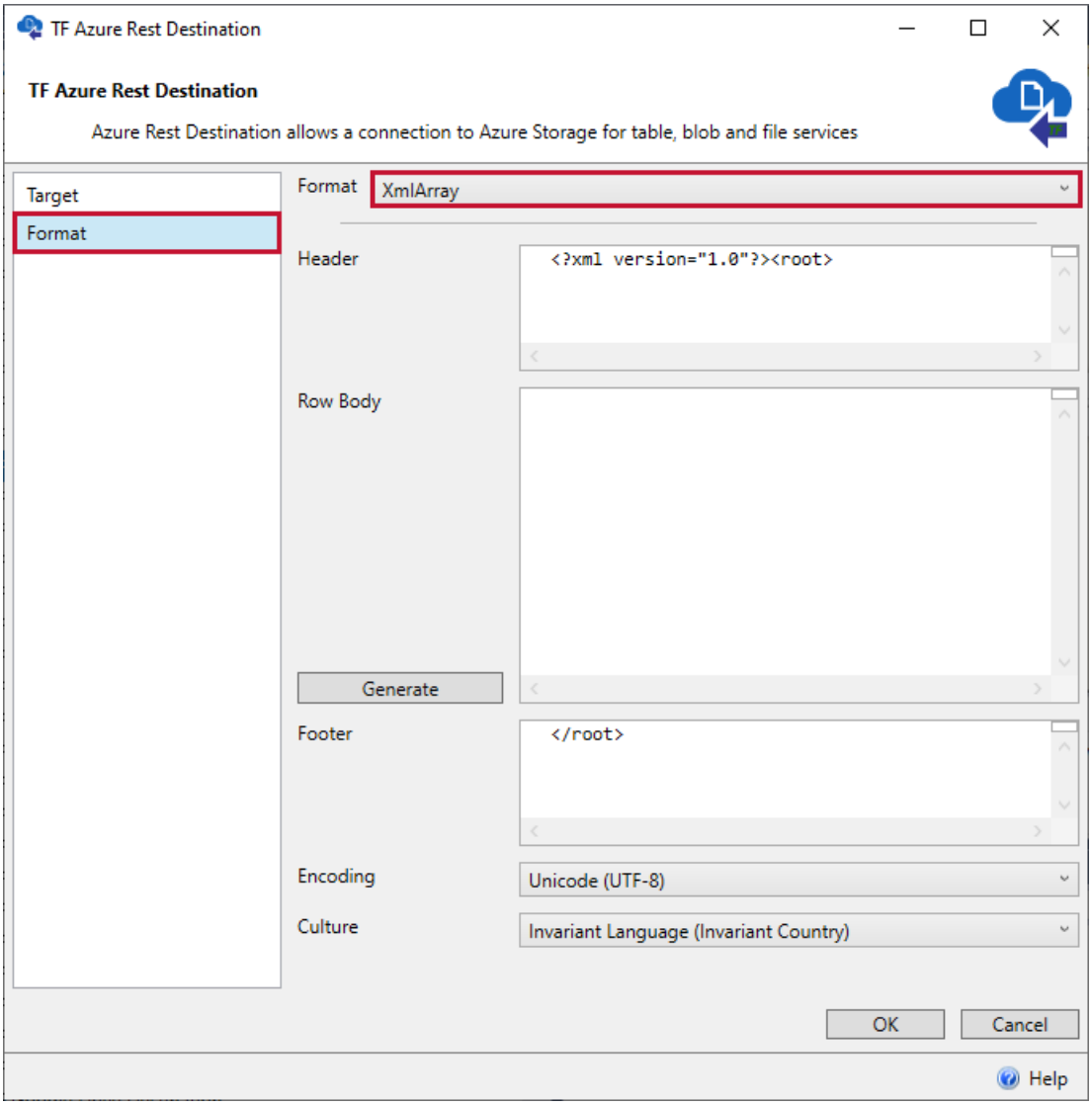
Option	Description
Delimiter	Identifies the character that separates text.
Quote	Identifies the character (single or double quote) that indicates value.
Include Header Row	Writes the column headers at the beginning of the document.
Encoding	Selects which encoding is used to create the document.
Culture	Used to identify the document's language culture.

Json Array Format



Option	Description
Generate	Select to automatically generate the Row Body metadata. Users can also add custom header and footer data.
Encoding	Select which encoding is used to create the document.
Culture	Used to identify the document's language culture.

XML Array Format



Option	Description
Generate	Select to automatically generate the Row Body metadata. Users can also add custom header and footer data.
Encoding	Select which encoding is used to create the document.
Culture	Used to identify the document's language culture.

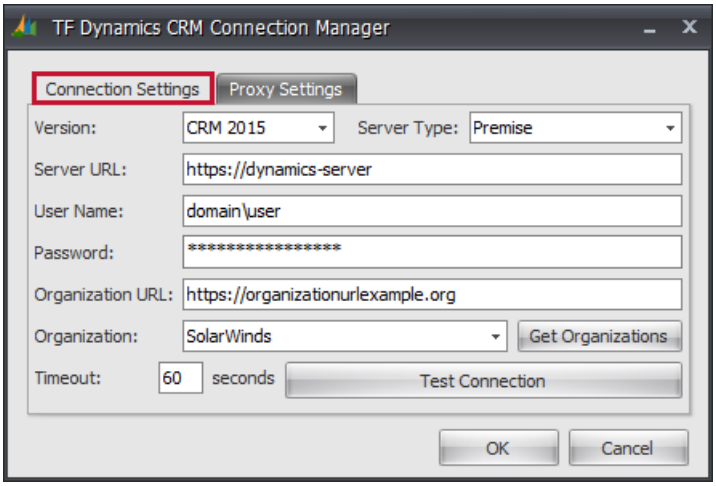
Task Factory Dynamics

Last Modified on 23 August 2022

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Dynamics CRM Connection Manager

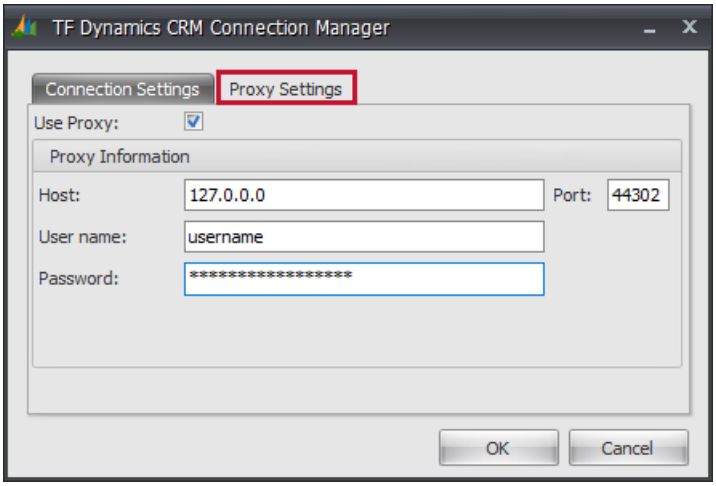
Connection Settings



Option	Description
Version	There are five versions of Dynamics CRM that can be utilized with the Dynamics CRM Connection Manager: • CRM 4.0 • CRM 2011 • CRM 2013 • CRM 2015 • CRM 2016

Option	Description
Server Type	Based on the version selected, the server type changes to reflect the available server types supported. **Available CRM 4.0 Server Types** : • **Premise** - This is a Dynamics server that is on site. • **Live** - An online account with Dynamics Live. When using this server type you should leave the default URL of **crm.dynamics.com** or **CRM4.dynamics.com**. • **Hosted** - A Server instance hosted off site with a third party. **Available CRM 2011 and 2013 - 2016 Server Types**: • **Premise** - This is a Dynamics server that is on site. • **Live** - An online account with Dynamics Live. When using this server type you should leave the default URL of **crm.dynamics.com** or **CRM4.dynamics.com**. • **Hosted** - A Server instance hosted off site with a third party. • **Office 365** - An online account with Dynamics Office 365. When using this server the URL should be left as **disco.crm.dynamics.com**.
Server URL	URL or IP address used to connect to the Dynamics CRM server.
User Name	Account used to sign into Dynamics.
Password	Password for Dynamics account.
Organization	The organization is a required field that must match the name of the organization setup in the dynamics crm server. If you do not know the name of the organization, you can select the **Get Organizations** to get a list of organizations from the server.
Organization URL	The url for the selected organization.
Timeout	The number of seconds before the connection manager times out trying to connect to the Dynamics CRM instance.
Test Connection	Select the Test Connection button to verify your connection credentials.

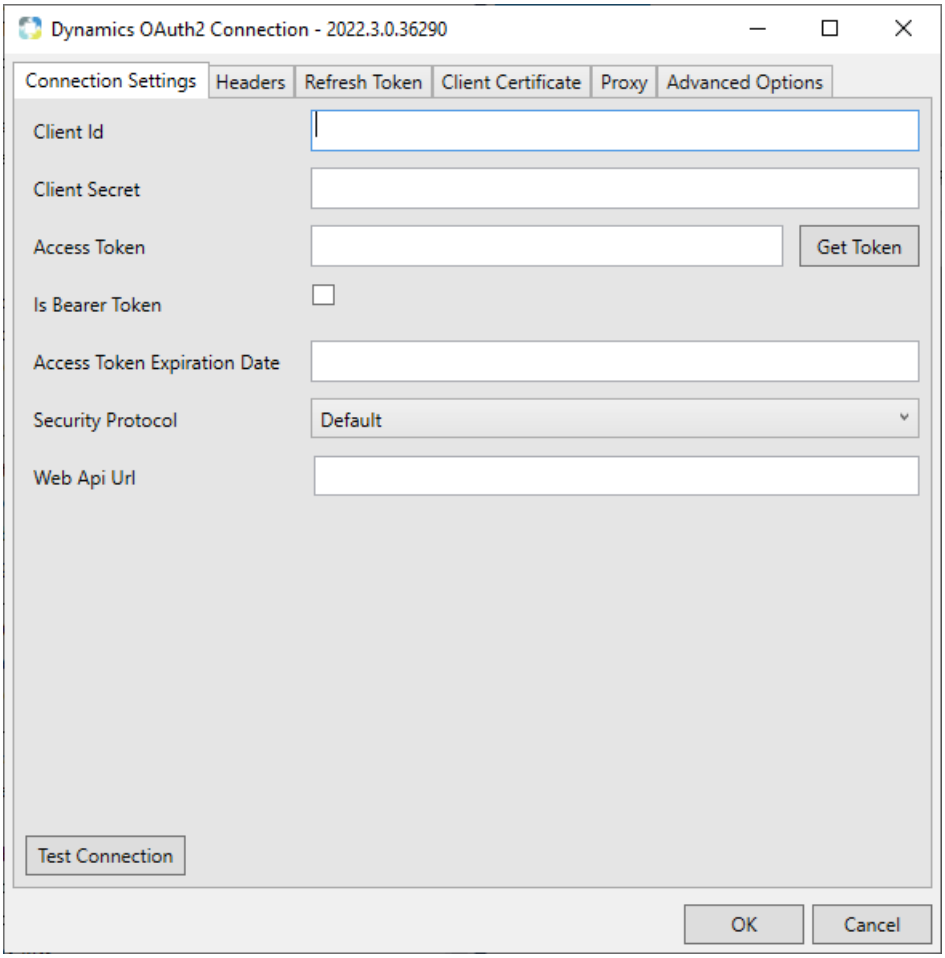
Proxy Settings



Option	Description
Host	Enter the Host's ip address.
Port	Enter the Host's port.
User name	Enter the Host's username.
Password	Enter the Host's password.

Dynamics OAuth2 Connection Manager

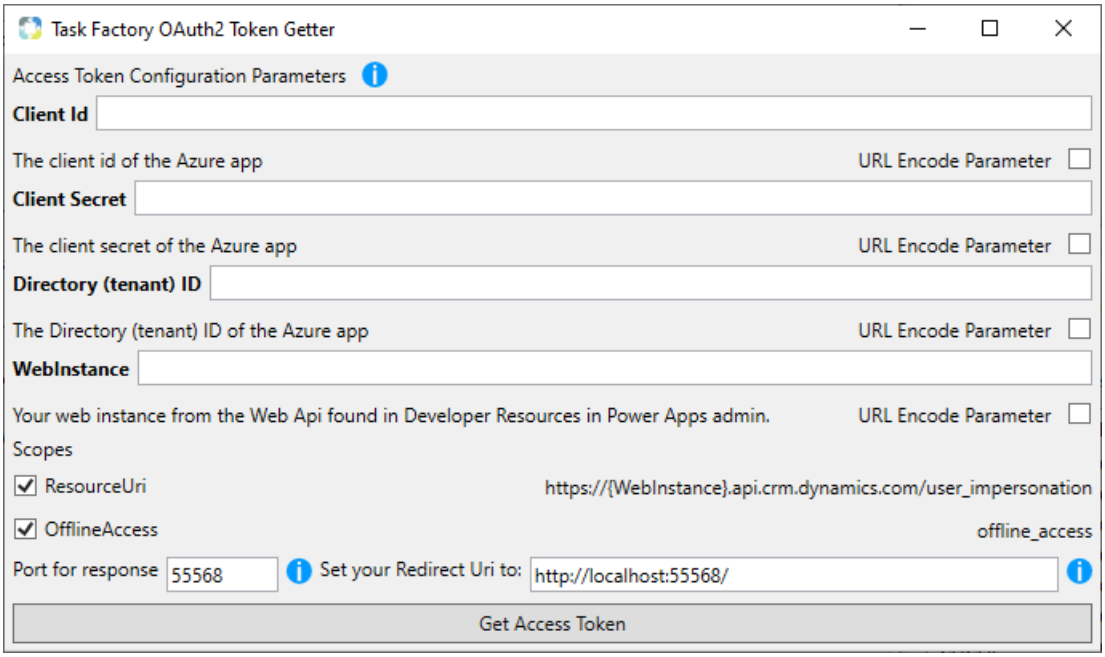
Connection Settings



Option	Description
Client id	The Client id for your Dynamics OAuth2 connection.
Client Secret	The Client secret for your Dynamics OAuth2 connection.
Access Token	Select Get Token to open the Token Getter window.
Is Bearer Token	Select this option if the Access Token is a bearer token.
Access Token Expiration Date	The expiration date of the access token. **Note:** This setting is optional.

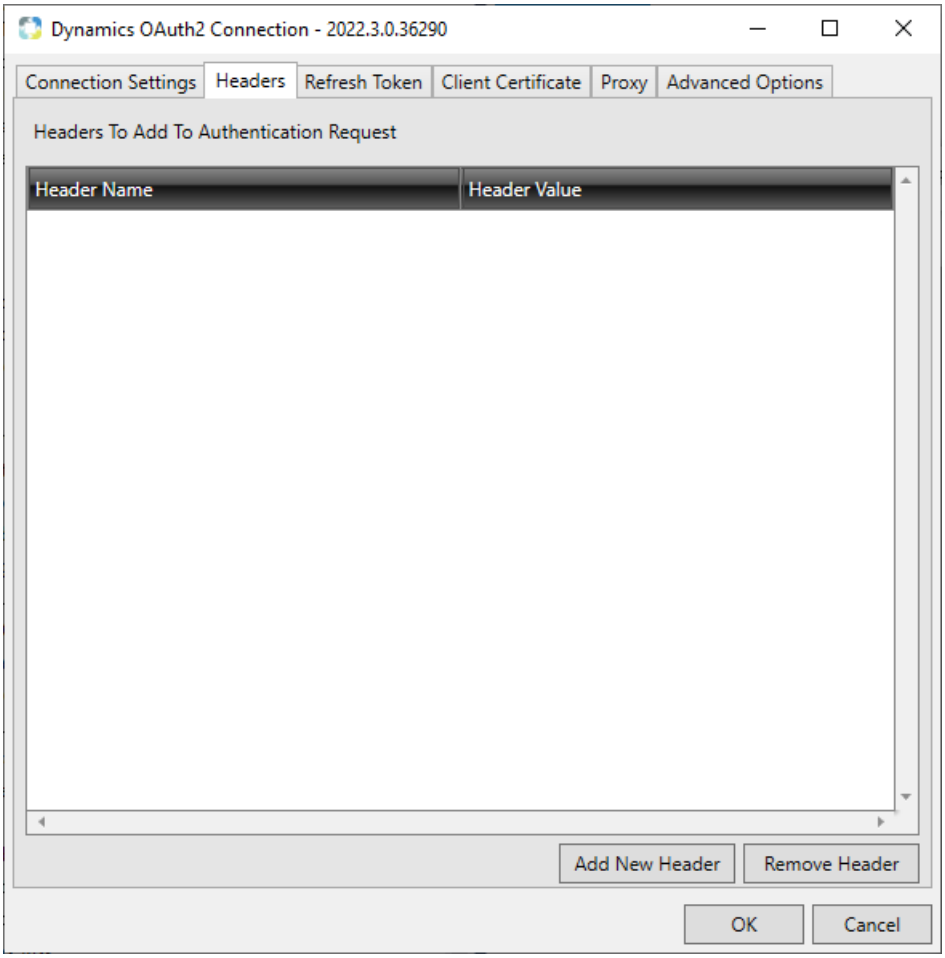
<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>Security Protocol</td><td>The Security Protocol used by the API. The Default option uses the TLS version used by your .NET version by default. See the following chart for more information:
<table>
<tr><th>.NET Version</th><th>TLS Support Information</th></tr>
<tr><td>.NET 4.6 and above</td><td>Supports TLS 1.2 by default.</td></tr>
<tr><td>.NET 4.5</td><td>TLS 1.2 is supported, but you need to opt-in to use it. Execute the following code before making a connection to a secured resource to make TLS 1.2 the default :
ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12</td></tr>
<tr><td>.NET 4.0</td><td>TLS 1.2 is not supported. If you have .NET 4.5 (or above) installed on the system, you can opt in for TLS 1.2 even if your application framework doesn’t support it. SecurityProtocolType in .NET 4.0 doesn’t have an entry for TLS1.2, so you need to use a numerical representation of this enum value:
ServicePointManager.SecurityProtocol = (SecurityProtocolType)3072;</td></tr>
<tr><td>.NET 3.5 or below</td><td>TLS 1.2 is not supported, and there is no workaround. Upgrade your application to a more recent version of the framework.</td></tr>
</table></td></tr>
<tr><td>Web Api Url</td><td>The URL for the Web API connection.</td></tr>
</table>

Token Getter



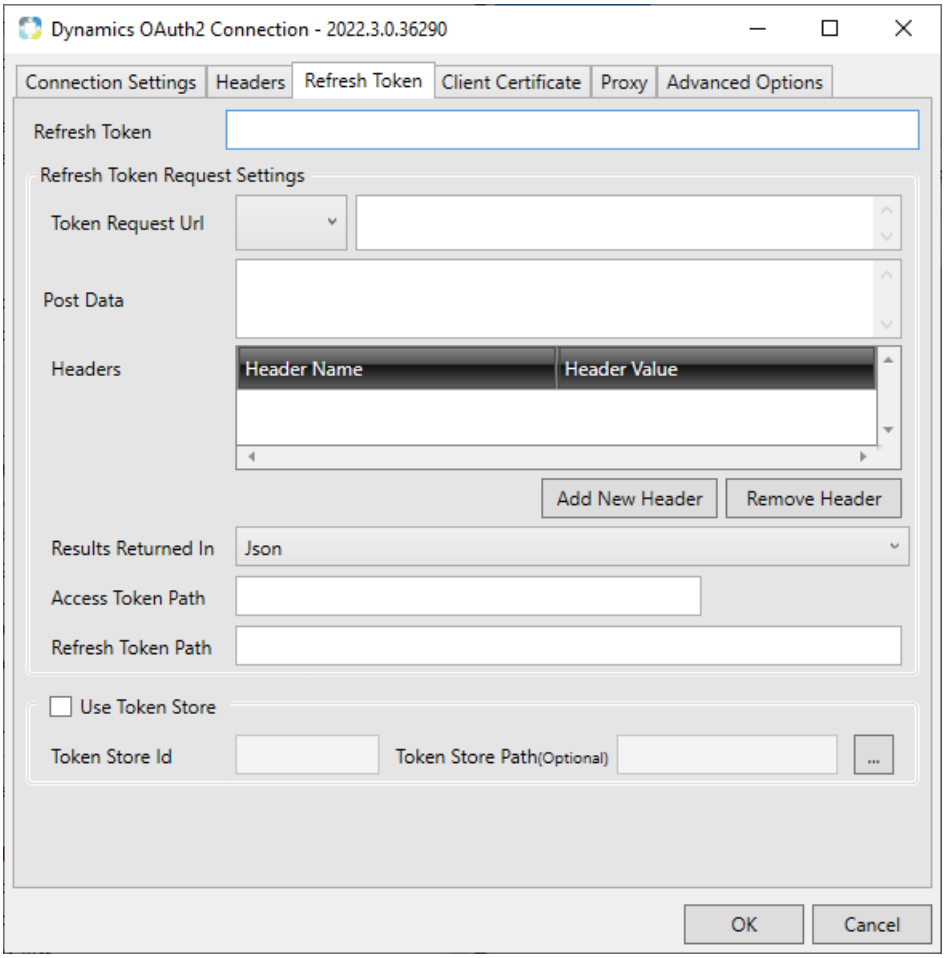
Option	Description
Client Id	This field should be auto-completed from the previous window with your Client Id.
Client Secret	This field should be auto-completed from the previous window with your Client Secret.
Directory (tenant) ID	The Directory (tenant) ID of the Azure app.
WebInstance	Your web instance from the Web API found in Developer Resources in Power Apps Admin.
ResourceUri	Select this option to enable the resource uri option.
OfflineAccess	Select this option to enable the offline access option
Port for response	The port on your local machine where you want to send the token request response.
Set your Redirect Uri to:	Use this as your Redirect Uri in your OAuth2 App settings.
Get Access Token	Opens a browser window to login to your Microsoft account and grant access.

Headers



You can create header names and values by entering the information in their corresponding windows.

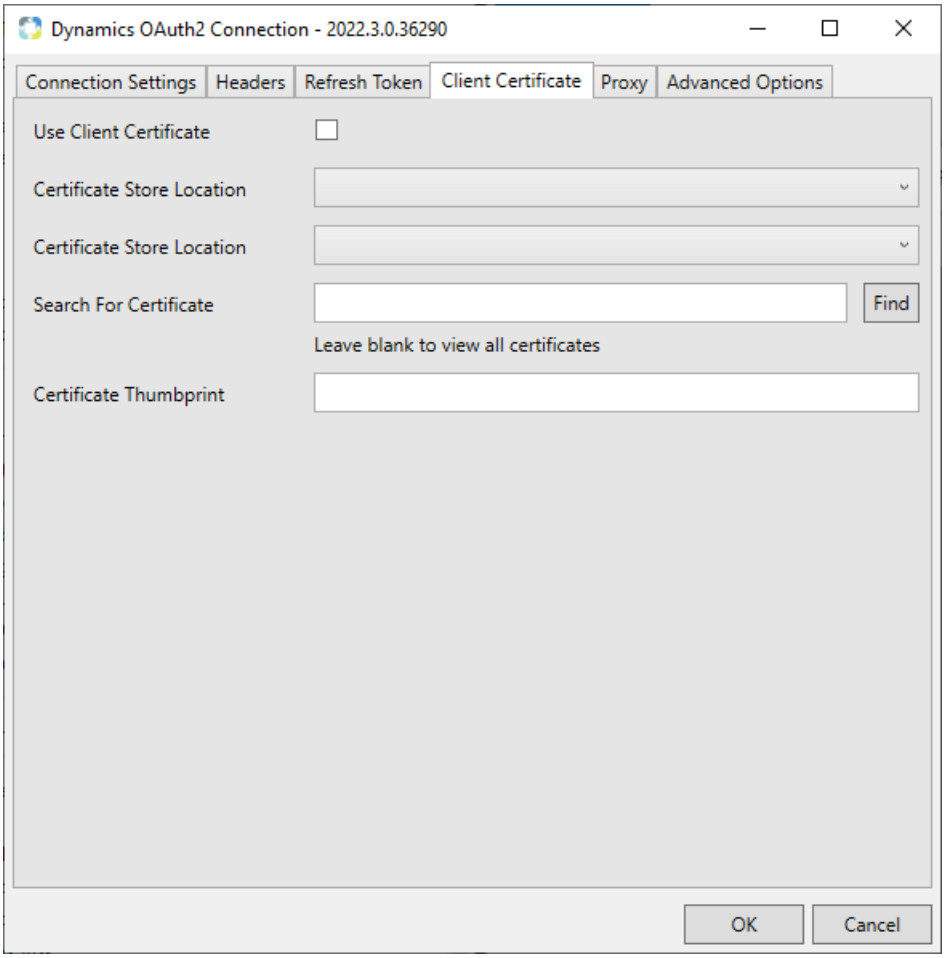
Refresh Token



Option	Description
Refresh Token	Authentication information provided to you at the application's developer site. **Note:** This should match the API Key from the Connection Settings window.
Token Request Url	The URL that returns a refresh token. Example: **https://api.citrixonline.com/oauth/access_token**
Headers	Allows you to add headers within the refresh token. Select **Add New Header** to add a header. Select **Remove header** to remove a header.
Results Returned In	Specifies how results are returned from the application (JSON, XML, or String.)
Access Token Path	The token path for the access token. Example: **access_token**
Refresh Token Path	The token path for the refresh token. Example: **refresh_token**

Option	Description
Group Index	Identifies the index position to return (0 based). **Note:** This option is available when you select **String** for the **Results Returned In** option.
Use Token Store	This option creates a file on your machine that saves the access token, and refresh token in an encrypted format.
Token Store Id	Identifies the Token Store Id.
Token Store Path (Optional)	Identifies the Token Store path.
Test Get Refresh Token	Select this option to test the retrieval of the Refresh Token.

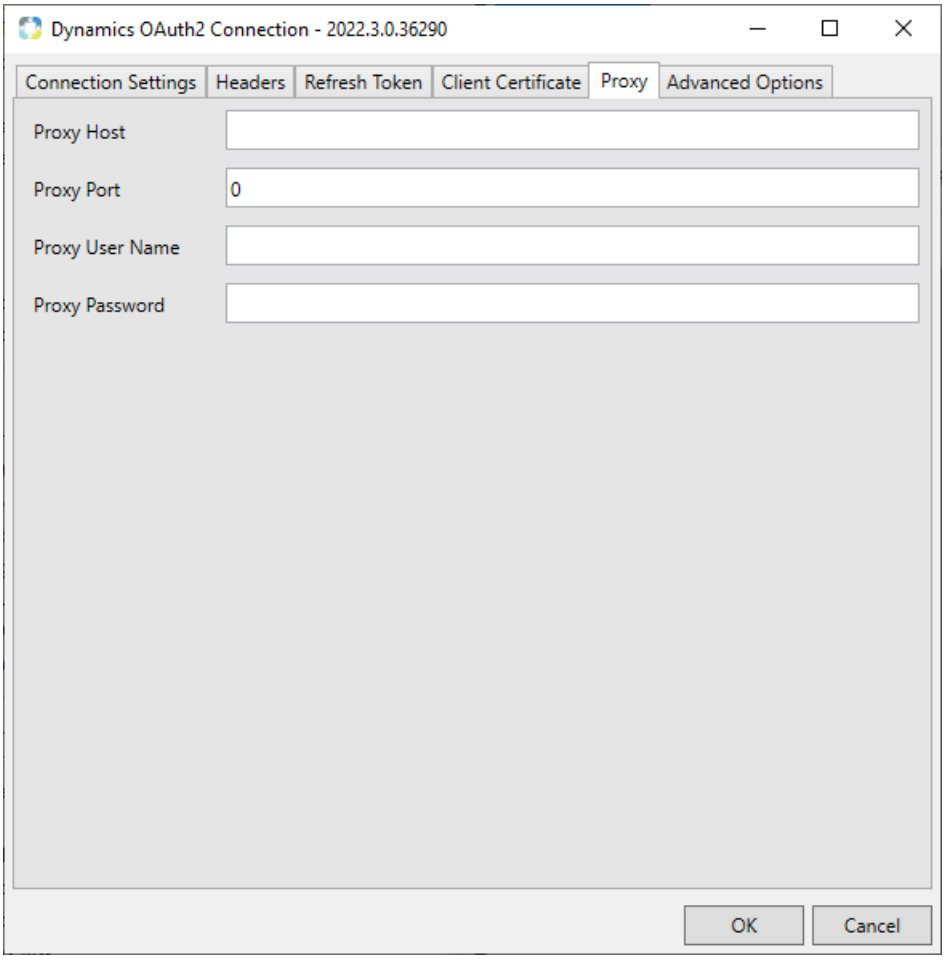
Client Certificate



Option	Description
Use Client Certificate	Select this option if you want to use the client certificate.

Option	Description
Certificate Store Location	The client store location.
Search For Certificate	Enter a certificate that you want to search for and then select **Find**. **Note:** Leave this section blank and select **Find** to view all certificates.
Certificate Thumbprint	The identifier thumbprint of the selected certificate.

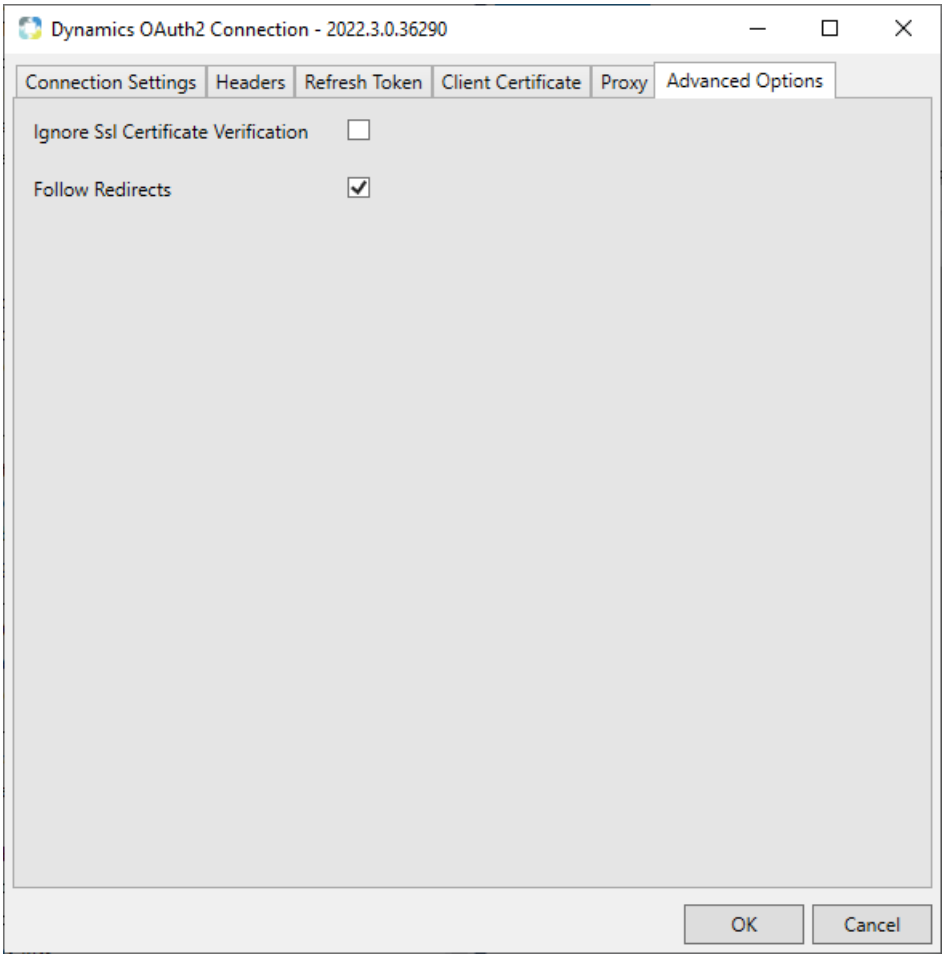
Proxy



Option	Description
Proxy Host	When connecting to an endpoint using a proxy, you should enter the proxy URL in this field.
Proxy Port	The port number that corresponds to the URL proxy host.
User Name	The username needed to authenticate to the proxy.

Option	Description
Password	The password that allows you to authenticate to the proxy.

Advanced Options



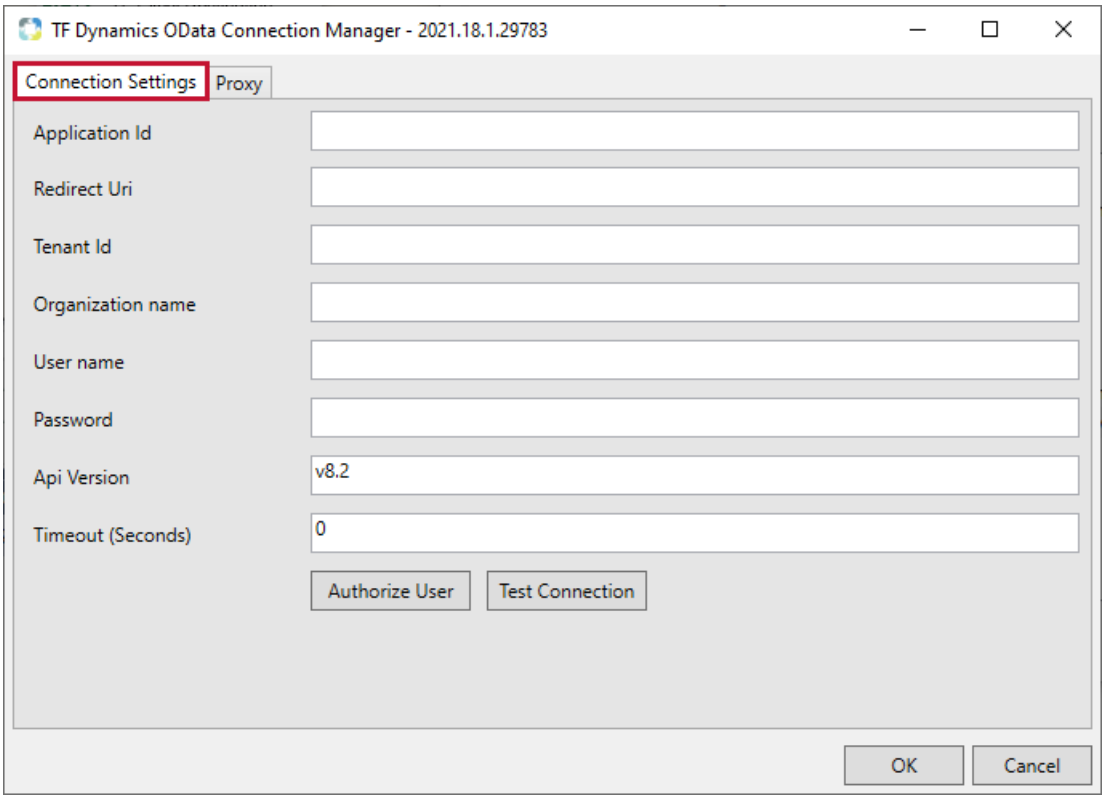
Option	Description
Ignore Ssl Certificate Verification	Selecting this option ignores the SSL Certificate notification from the API during the connection. **Note:** This option applies to APIs that require you to have an SSL Certificate. Select this option if the SSL Certificate is not needed.
Follow Redirects	Selecting this option implements a 301 redirect on configured endpoints. For example, this option would automatically send your request to an HTTPS redirect once the request has reached the HTTP version of the site.

Dynamics CRM OData Connection Manager

Note: Azure Storage connection manager is available for SQL versions 2012 and higher. Currently,

the OData can only target Dynamics CRM instances that use Office 365.

Connection Settings Tab



Additional Information: The following configurable items are located within the user's Azure portal.

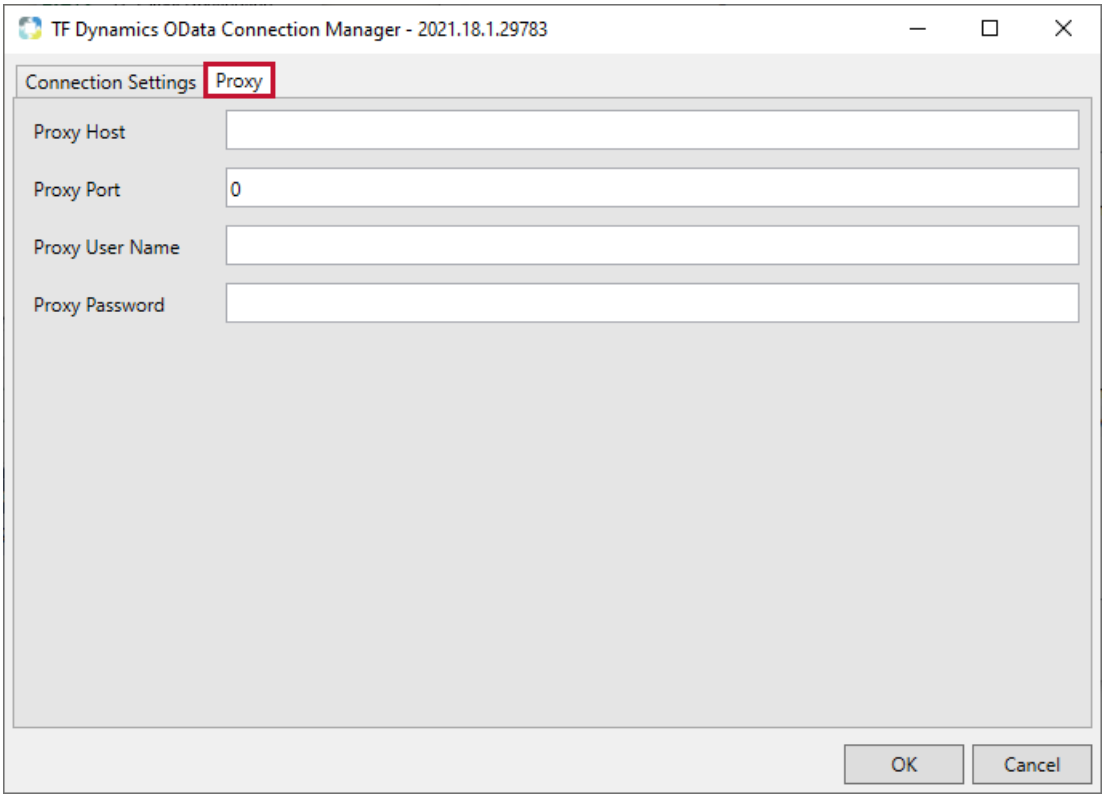
Note: A Dynamics app registration needs to be completed to access these items.

Function	Description
Application Id	Enter the application Id located in the user's **Azure Active Directory** > **App Registrations** >(NameOfApp).
Redirect URI	Enter the application's redirect URI configured in the user's **Azure Active Directory** > **App Registrations** > (NameOfApp) > **Redirect URI's.**

Function	Description
Tenant Id	Enter the application's tenant Id located at http://login.windows.net/YOUR_DOMAIN/.well-known/openid-configuration. Alternatively, users can open https://portal.azure.com/#blade/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/Properties and navigate to **Azure Active Directory** > **Properties**. **Note:** The Directory Id is also the Tenant Id.
Organization name	Enter your organization's name located at https://YOUR_ORGANIZATION.crm.dynamics.com.
Username	Enter the username for the user logging into the Dynamics CRM instance.
Password	Enter the password for the user logging into the Dynamics CRM instance.
Api Version	Enter the Api version. The default value is v8.2.
Timeout (Seconds)	The number of seconds before the connection manager times out trying to connect to the Dynamics CRM instance. **Note:** The default value is zero which is infinite.

Authorize User - Before using the connection manager, users must setup their app in the azure portal. After the connection manager details have been entered, users need to authorize their user account to interact with dynamics by selecting **Authorize User**. This opens a miniature browser window that automatically navigates to the webpage where the credentials must be entered.

Proxy Tab

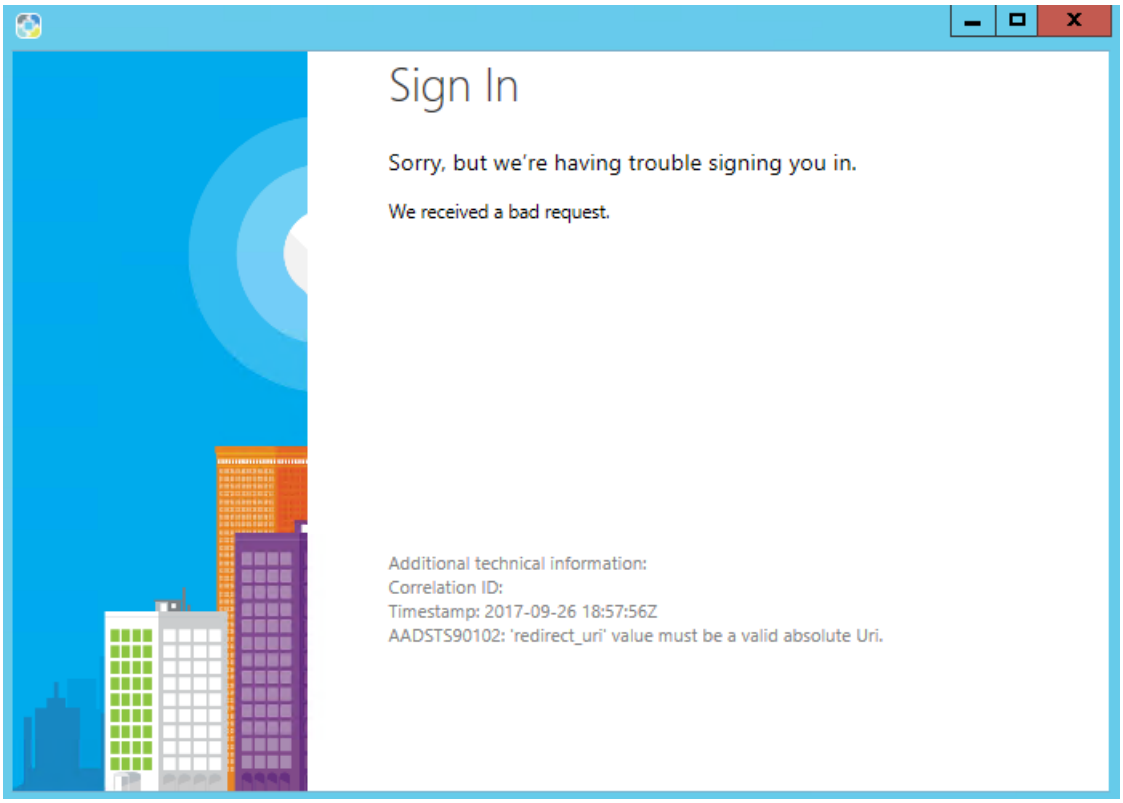


Note: Proxy use is enabled when a value is added to any of the Proxy fields.

Function	Description
Proxy Host	Enter the proxy host's IP address.
Proxy Port	Enter the proxy host's port number.
Proxy User Name	Enter the proxy host's username credential if required. If it's not required, leave blank.
Proxy Password	Enter the proxy host's password if required. If it's not required, leave blank.
Auth Scheme	Select the authentication scheme used by the host proxy. Users can select from the following options: • Basic • Digest • Proprietary • None • Ntlm • Negotiate

Troubleshooting Tips

Users who are not authorized encounter the following:



Note: Currently, the ODATA is missing the following functions and actions for some organization service messages:

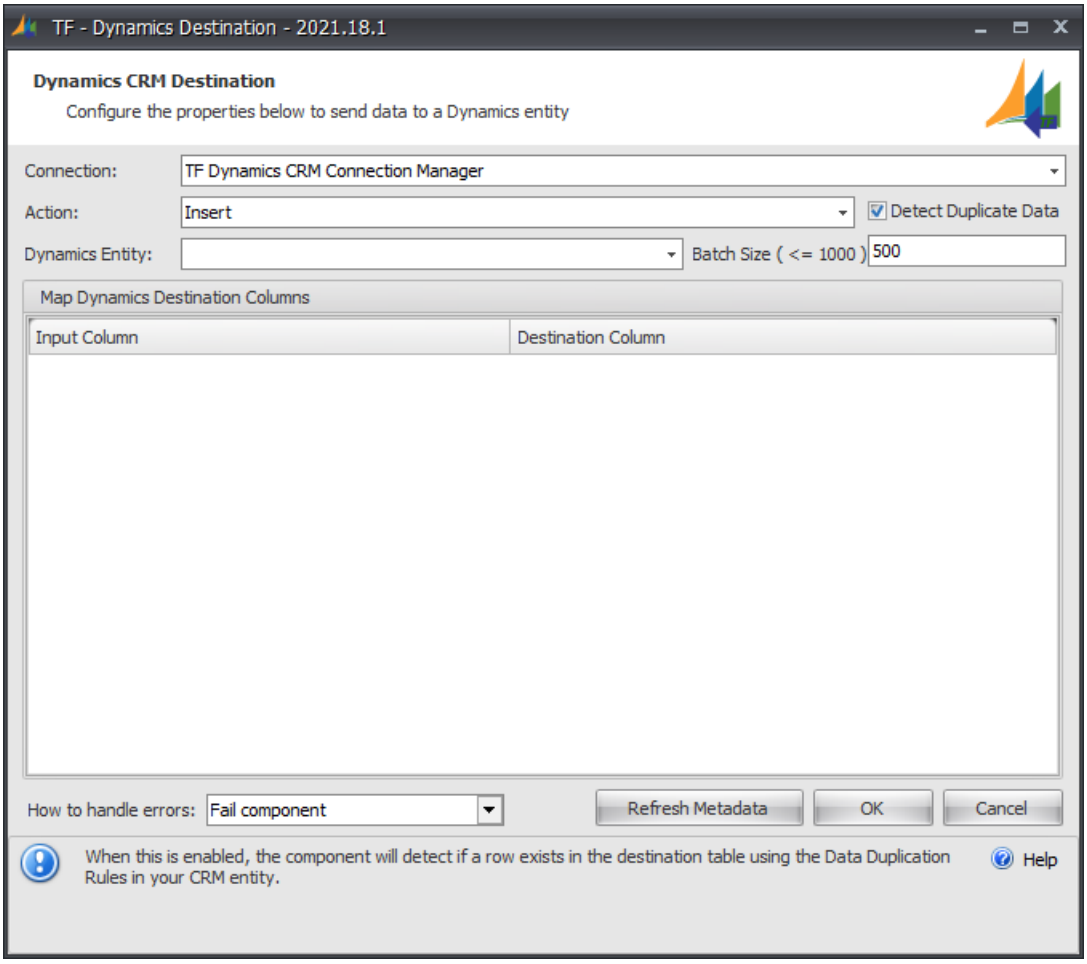
- GrantAccessRequest
- ModifyAccessRequest
- ReactivateEntityKeyRequest
- RemoveMemberListRequest
- RemoveItemCampaignRequest
- RemoveItemCampaignActivityRequest

Additional Information: For more information, see the Microsoft Dynamics 365 Web API Limitations MSDN article.

Dynamics CRM Destination

Note: Task Factory currently supports the following versions of Dynamics CRM: 4.0, 2011, 2013, 2015, 2016.

Destination Icon	Destination Description
	The Dynamics CRM Destination is used for writing data to a table in a Dynamics CRM server. See the Dynamics CRM Connection Manager or Dynamics OData Connection Manager to learn more about setting up the connection managers.



Function	Description
Connection	Using the drop down menu, create or select a connection manager for the Dynamics CRM server.
Action	Choose the desired action : • **Insert** - Insert new rows into the Dynamics Entity. • **Update** - Update rows that already exist in the Dynamics Entity. • **Delete** - Delete existing rows from the Dynamics Entity. • **Upsert** - Upsert is used to insert new rows into the Dynamics Entity, or update rows that already exist.
Detect Duplicate Rows	When this option is enabled, the destination checks for duplicate rows based on the data duplication rules on the CRM Server. The rows are handled according to the **How to handle errors** setting.
Dynamics Entity	This field displays all of the available tables in the Dynamics server.
Map Dynamics Destination Columns	Sets up desired mappings.

Upsert Modes

The Upsert mode of Dynamics has two modes, Primary Key Column Mapped, and Primary Key Column Unmapped.

Primary Key Column Mapped

If the primary key from entity is mapped to a column from the source, the destination looks for NULL values in the source data. If the source data for the primary key is NULL, then an insert executes. If the primary key is NOT NULL then an update executes. For an Update to work the primary key column value from the source MUST be the primary key value from the Dynamics entity. For instance, the Account entity primary key column is **AccountID**. This column must be mapped to be in **Primary Key Mapped** mode.

If **Detect Duplicate Rows** is checked an insert is attempted. However, if a **Duplicate Data** rule on the dynamics server detects a duplicate row, the data in the current row that would have been inserted updates instead.

Example

To give you an example of this let's assume your Account entity on Dynamics has a Duplicate Data rule on the Name column. We have an account named **Sentry One** in the Dynamics Account entity. If a row with the account name of **Sentry One** from your source is sent to the server as an insert (because the primary key value is null) then the Dynamics Server rejects the insert but the Dynamics Destination component then takes the current row from the source and updates the row in the Dynamics Account entity. If this behavior is NOT DESIRED then **Detect Duplicate Rows** should be turned off.

If **Detect Duplicate Rows** is unchecked and duplicate data is detected on the server the row is rejected and an error is thrown.

Primary Key Column Not Mapped

In this mode, the primary key is not mapped and only the **Duplicate Data** rules on the Dynamic server are used. For each row from the source an insert is attempted on the server.

Due to the **Duplicate Data rule**, if **Detect Duplicate Rows** is checked and a duplicate row is detected on the Dynamics server, the detected duplicate row on the dynamics server updates.

Example

To give you an example of this let's assume your Account entity on Dynamics has a Duplicate Data rule on the **Name** column. We have an account named **Sentry One** in the Dynamics Account entity. If a row with the account name of **Sentry One** from your source is sent to the server and a duplicate row is detected on the Dynamics Server the row is rejected as an insert and treated as update.

If **Detect Duplicate Rows** is unchecked and a duplicate row is detected on the Dynamics server based on the Duplicate Data rule then an error is thrown.

Additional Information: See the Task Factory Error Row Handling article for more information about this functionality.

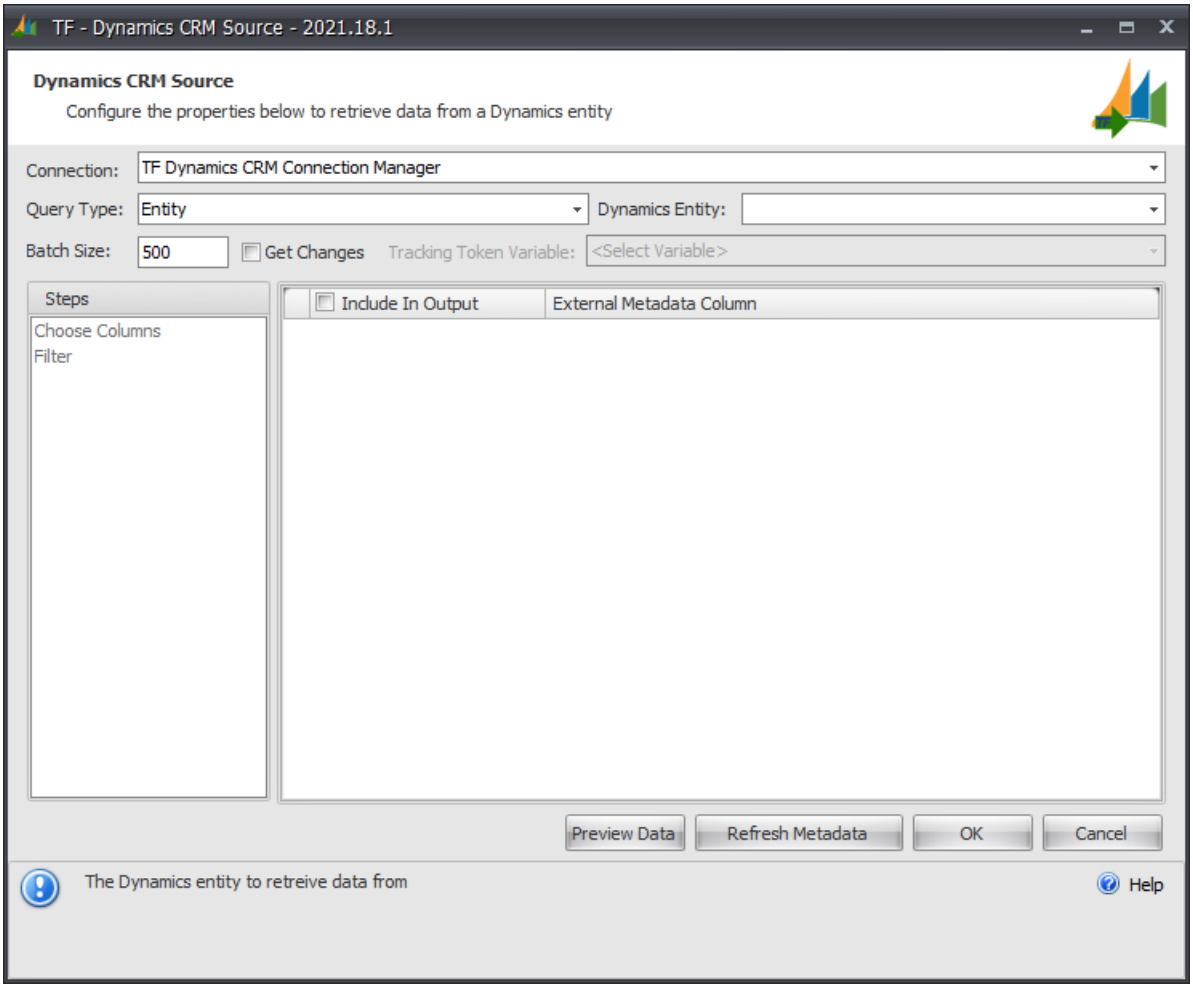
Dynamics CRM Source

Source Icon	Source Description
	The Dynamics CRM Source reads data from the entities in a Microsoft Dynamics Server. See the Dynamics CRM Connection Manager or the Dynamics OData Connection Manager to learn more about setting up the connection manager.

Task Factory currently supports the following versions of Dynamics CRM:

- 4.0
- 2015
- 2011
- 2016
- 2013

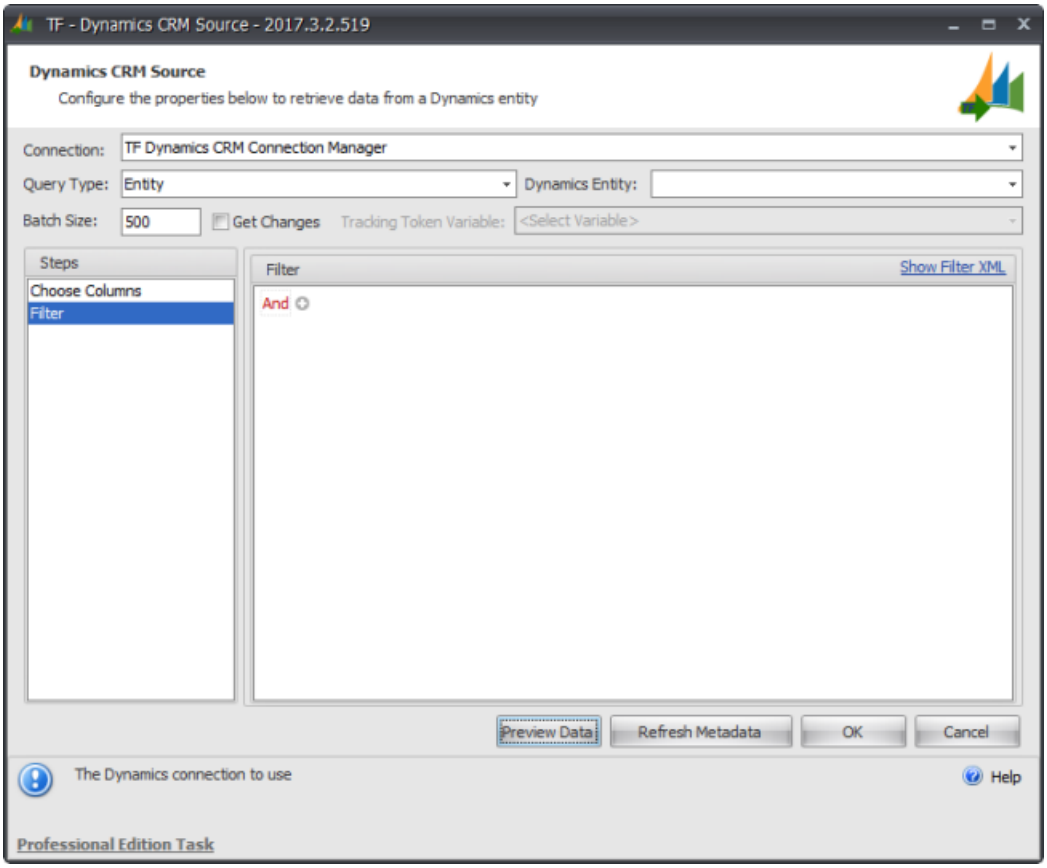
Choose Columns Tab



Function	Description
Connection	Create or select a connection manage for the Dynamics CRM server.
Query Type	Determines where to look for the Dynamics Entity: • **Entity** - Selects the "standard" entities in Dynamics such as Accounts, Contacts, Opportunities, etc. Views are selected in the Dynamics Entity drop-down. • **Personal View** - Allows users to select custom views. Views are selected in the Dynamics Entity drop-down. • **System View** - Allows users to select all non-custom views within the Dynamics CRM system.
Dynamics Entity	Displays a list of entities form which the source can read.
Batch Size	The Batch Size field determines the number of rows that are retrieved during a single round trip to the Dynamics server. **Note:** The default value is 500.

Function	Description
Get Changes	Toggle for users to track changes/updates for synchronization. (Filters cannot be used when selecting this option.) For more information, see Change Tracking . **Note:** Get Changes can only be used when the Entity Query Type is selected.
Tracking Token Variable	Identifies the variable that stores the (required) change. **Note:** If you're using Dynamics with change tracking, the **Tracking Token Variable** saves the Tracking token that returns from dynamics. If you don't have change tracking enabled, the subsequent calls to dynamics will return the same rows without the changes.
Choose Columns	Once an entity is selected, the available columns display. Columns can be ignored by unchecking the corresponding box in the **Include in Output** column.

Filters Tab



Function	Description
Filter	Filters can be set so that users can customize the data that is returned.
Preview Data	Opens a new window that displays the data based on the entity and conditions (if configured.)

Function	Description
Refresh Metadata	Updates the component with the most current metadata.

To add a filter, complete the following steps:

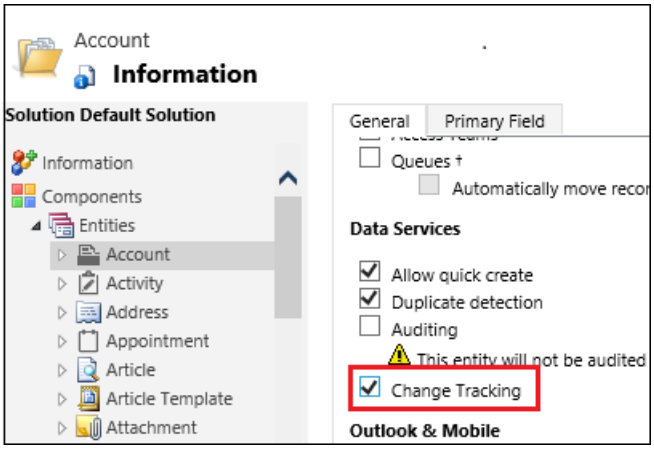
1. Select the add icon **(+).**
2. Choose the column to be filtered.
3. Select the parameter (e.g. Begins with, Greater than, Less than, etc.).
4. Define the value in the box. (Variable replacements can be used with the following syntax:**<@Namespace::MyVariableName>** . For example: **<@User::Variable>**).
5. Select the word **And** if **Or** filter needs to be added.

Dynamics CRM Source Change Tracking

The Dynamics CRM Source allows organizations to synchronize their data with external data sources. Users can track when a change occurs by selecting the **Get Changes** toggle. Before using this feature, users must first enable the feature.

To enable the feature, complete the following steps:

1. Go to the entity to be tracked.
2. Go to **Customizations** > **Customize the System**.
3. Select an entity, and under **Data Services**, select the **Change Tracking** check box.



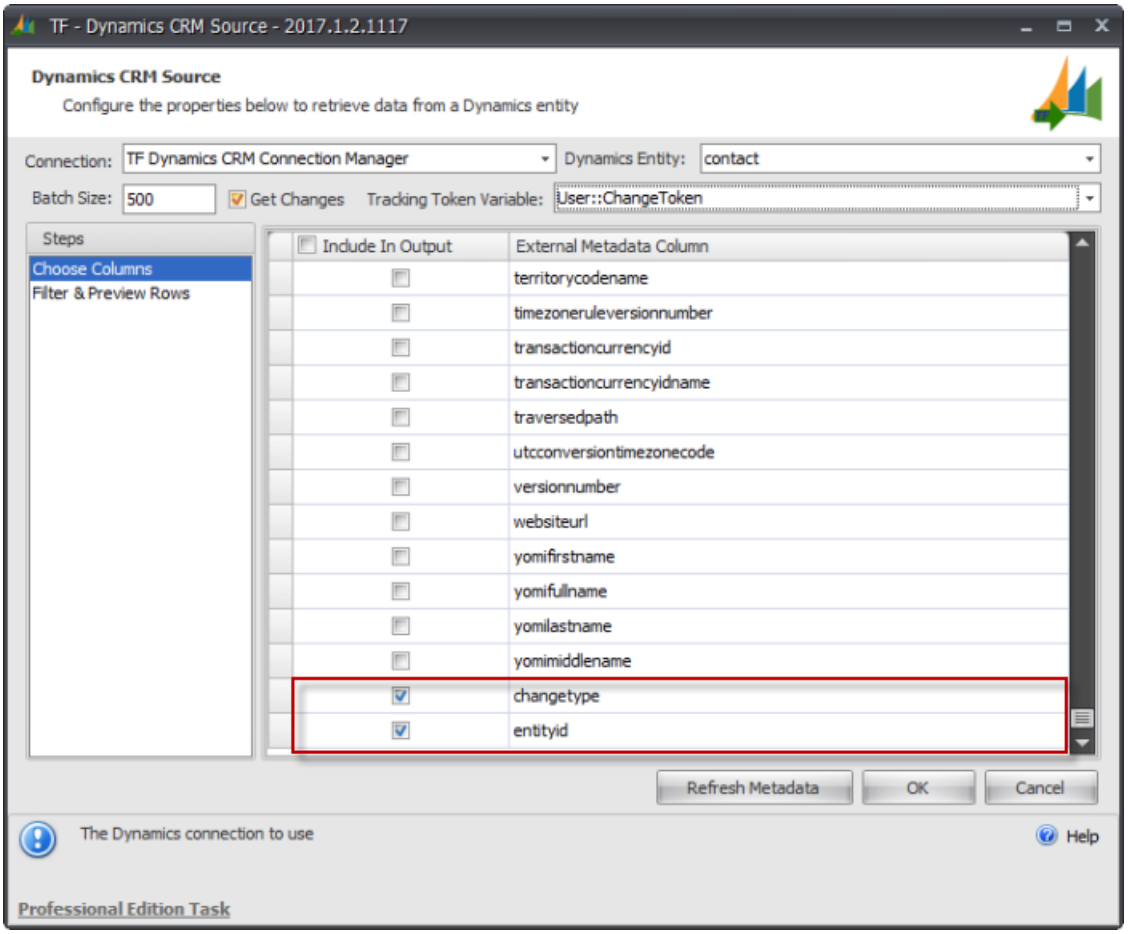
Note: Before using this feature, please note the following:

- **Change Tracking** is an entity-level setting that must be enabled before using. This must be enabled for each entity the users wish to track changes.
- A change token is also required for each individual entity. (For example: the Accounts entity token is different than the Contacts token.)

- Lookup field properties are not returned.
- The data is returned in their current status. The pre-change status is not returned.
- Users accounts must be granted the organization level read privileges for each tracked entity.

Additional Information: For more information on enabling your Dynamics CRM instance, please see the Use change tracking to synchronize data with external systems MSDN article.

Note: The **changetype** and **entityid** columns are added to the end of the output when the Get Changes option is selected.



Task Factory Email

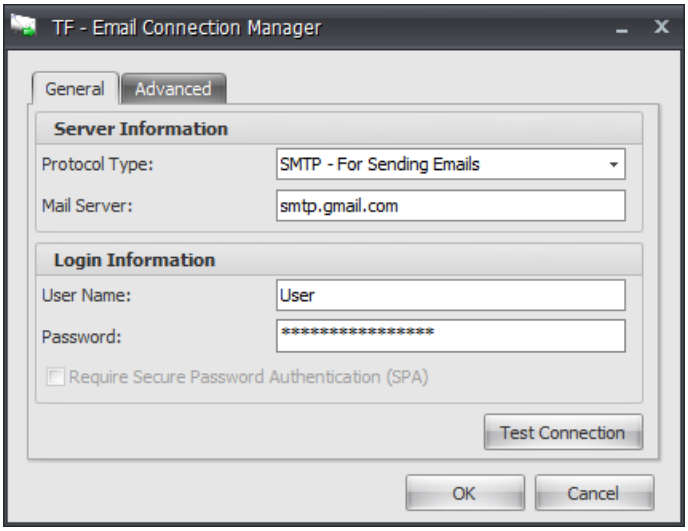
Last Modified on 27 January 2022

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Email Connection Manager

The Task Factory Email Connection Manager is used with Advanced Email and SMS Task, Email Source and Email Source Delete or Move Message Task.

General Tab



Server Information

Option	Description
Protocol Type	The protocol used to read or send email messages. There are three protocols available: • **POP3** - Read only • **IMAP** - Read Only • **SMTP** - Send Only
Mail Server	The address of the mail server.

Login Information

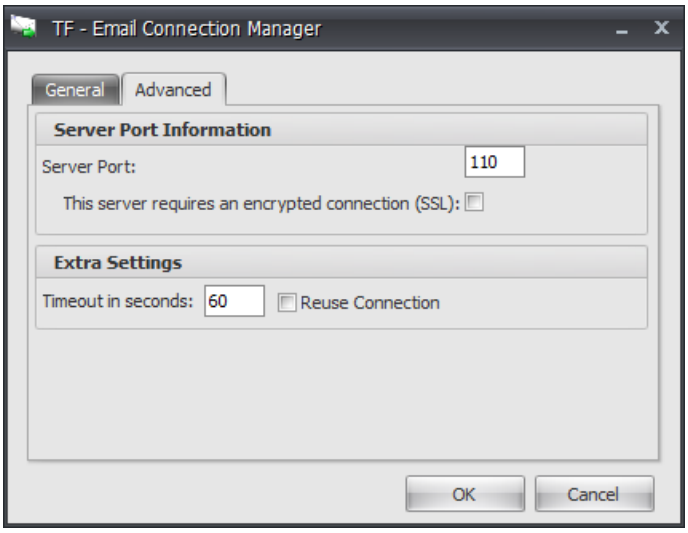
Option	Description

Option	Description
User Name	The user name for the email account. To use a domain account, enter the domain name and login (e.g. DOMAIN\User1)
Password	The password for the email account.
Requires Secure Password Authentication (SPA)	Some servers require SPA to be able to access the account.

Advanced tab

The **Advanced tab** changes depending on which Protocol Type is selected in the **General** Tab.

Pop3



Server Port Information

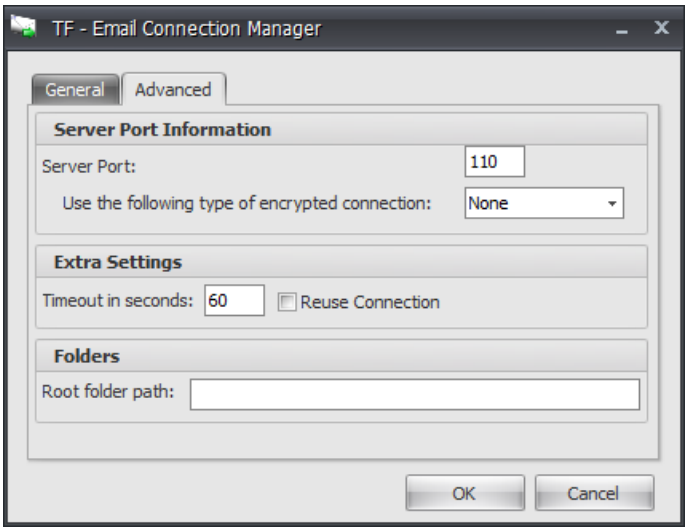
Option	Description
Server Port	Port used to connect to the mail server specified in the general tab.
This server requires an encrypted connection (SSL)	Some servers require a secure connection to access the POP3 mailbox.

Extra Settings

Option	Description

Option	Description
Timeout in seconds	The number of seconds before the connection manager will timeout connecting to the mail server.
Reuse Connection	Option to maintain the original connection. **Note:** This option should be selected when the Advanced Email and SMS Task is used within the loop containers.

IMAP



Server Port Information

Option	Description
Server Port	Port used to connect to the mail server specified in the general tab.
Use the following type of encrypted connection	Some servers require a secure connection to access the IMAP mailbox. There are three available encryption types: • None (default) • SSL • TLS

Extra Settings

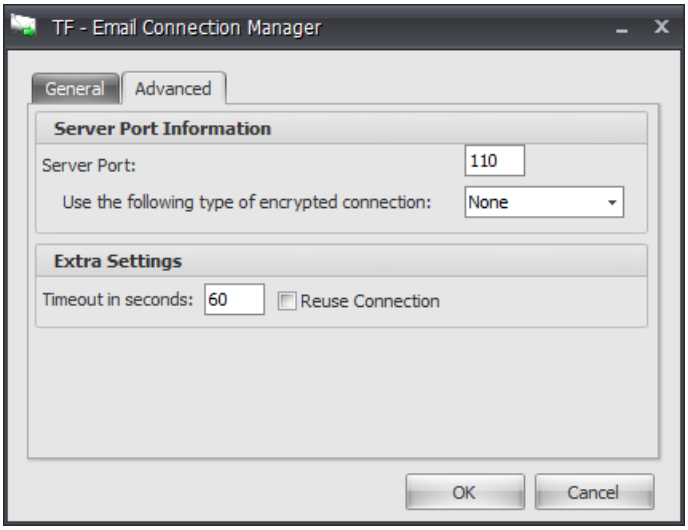
Option	Description
Timeout in seconds	The number of seconds before the connection manager will timeout connecting to the mail server.

Option	Description
Reuse Connection	Option to maintain the original connection.

Folders

Option	Description
Root Folder Path	The folder on the IMAP server that contains the messages you want to download.

SMTP



Server Port Information

Option	Description
Server Port	Port used to connect to the mail server specified in the general tab
Use the following type of encrypted connection	Some servers require a secure connection to access the IMAP mailbox. There are three available encryption types: • None (default) • SSL • TLS

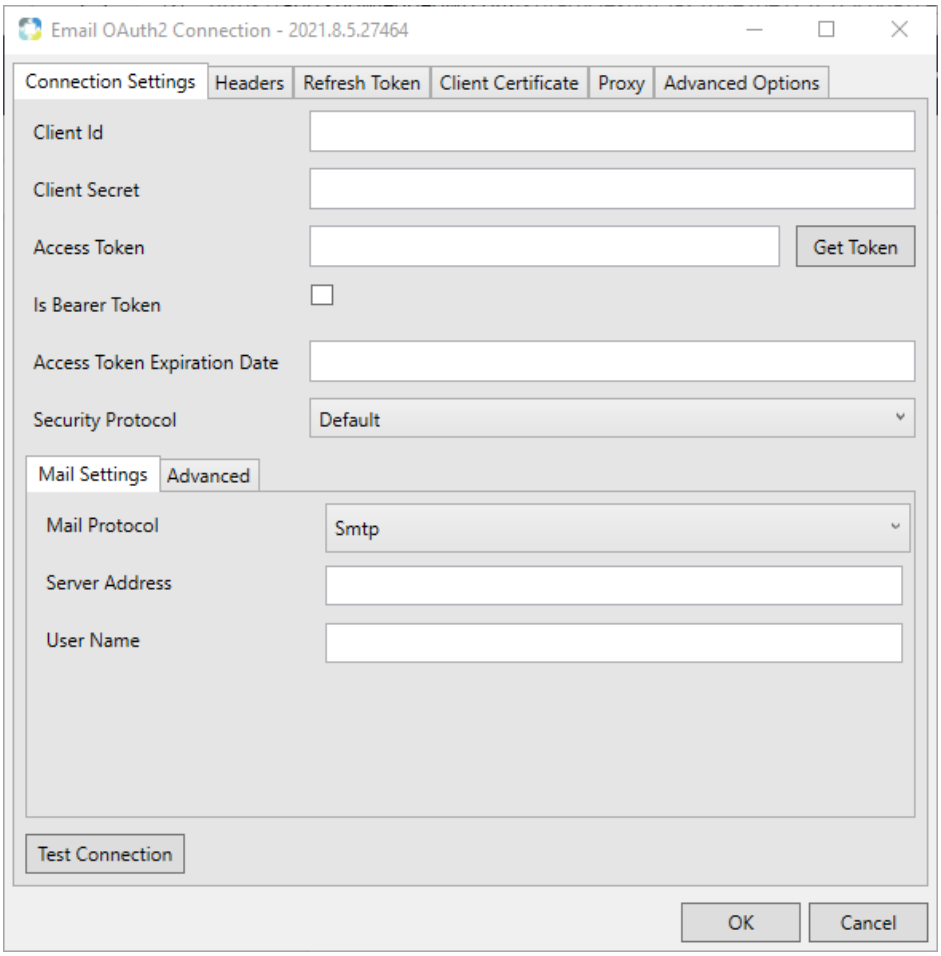
Server Timeout

Option	Description
Timeout in seconds	The number of seconds before the connection manager will timeout connecting to the mail server.
Reuse Connection	Option to maintain the original connection.

Email OAuth2 Connection Manager

The Task Factory Email OAuth2 Connection Manager is used with Advanced Email and SMS Task, Email Source and Email Source Delete or Move Message Task.

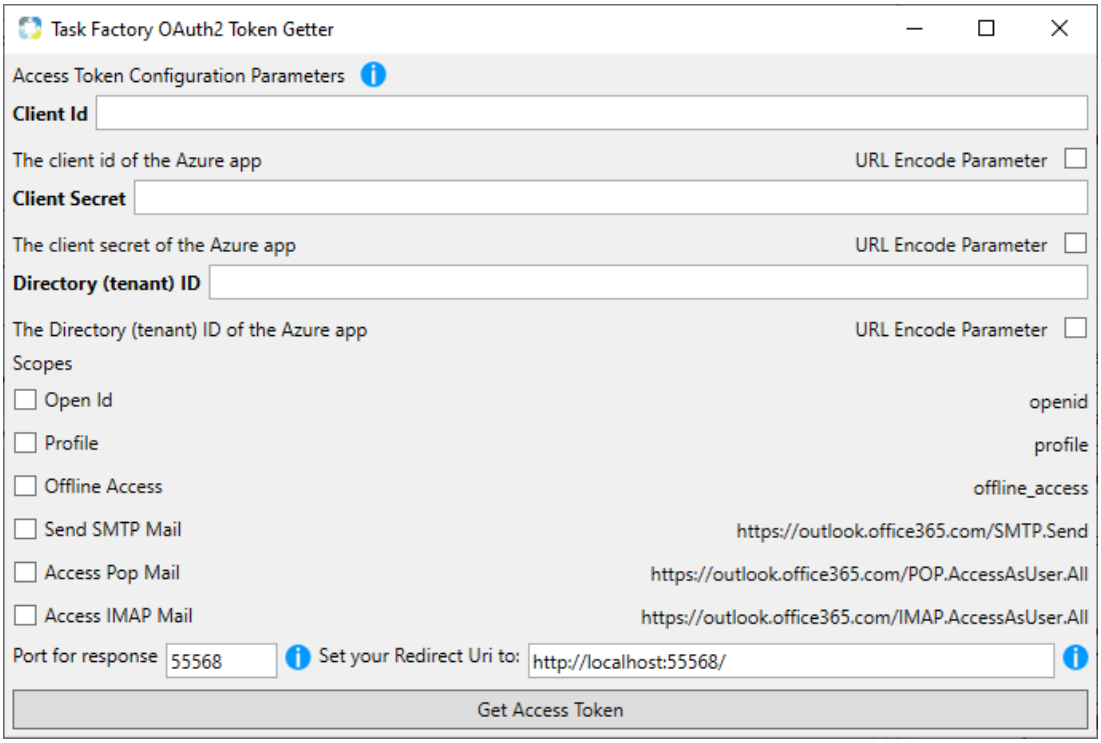
Connection Settings



Option	Description
Client Id	The Client Id of the Azure App that is found in your App's overview page on Office 365. **Additional Information:** For information about configuring an app to access a web API see the following MSDN article.

Option	Description
Client Secret	The Client Secret of the Azure App that is found in your App's overview page on Office 365.
Access Token	Select **Get Token** to open the Token Getter window.
Is Bearer Token	Select this option if the Access Token is a bearer token.
Access Token Expiration Date	The expiration date of the access token. **Note:** This setting is optional.
Security Protocol	The Security Protocol used by the API. The **Default** option uses the TLS version used by your .NET version by default. The following options are available: • Ssl3 • Tls • Tls11 • Tls12
Test Connection	Select **Test Connection** to test your entered connection credentials.

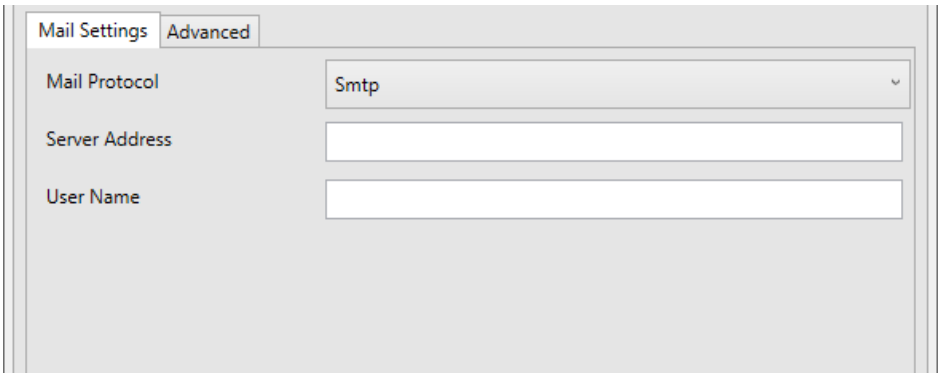
Email OAuth2 Token Getter



Option	Description
Client Id	The Client Id of the Azure App. **Note:** This field automatically populates with your entry on the Connection Settings tab.

Option	Description
Client Secret	The Client Secret of the Azure App. **Note:** This field automatically populates with your entry on the Connection Settings tab.
Directory (tenant) ID	The directory tenant id of the Azure app that is found in your App's overview page on Office 365.
Scopes	Select the scope checkbox(es) that are applicable: **Open Id** — openid **Profile** — profile **Offline Access** — offline_access **Send SMTP Mail** — https://outlook.office.365.com/SMTP.send **Access Pop Mail** — https://outlook.office365.com/POP.AccessAsUser.All **Access IMAP Mail** — https://outlook.office365.com/IMAP.AccessAsUser.All
Port for response	The port where the response for the token request will be sent to on your local machine.
Set your Redirect Uri to:	Use this as your Redirect uri in your OAuth2 App settings.

Mail Settings Tab



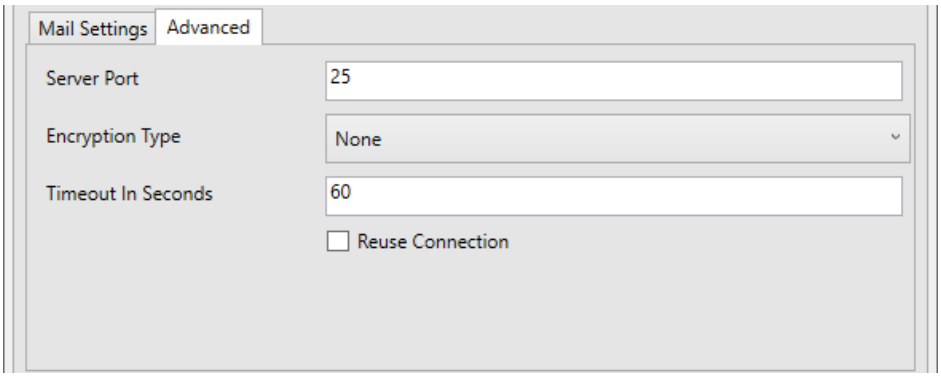
Option	Description

Option	Description

Mail Protocol	• **Protocol Type** - The protocol used to read or send email messages.
	• There are three protocols available:
	  ◦ **POP3** - Read only
	  ◦ **IMAP** - Read Only
	  ◦ **SMTP** - Send Only

Server Address	The server address of your email server.
User Name	The username associated to your email server.

Advanced Tab

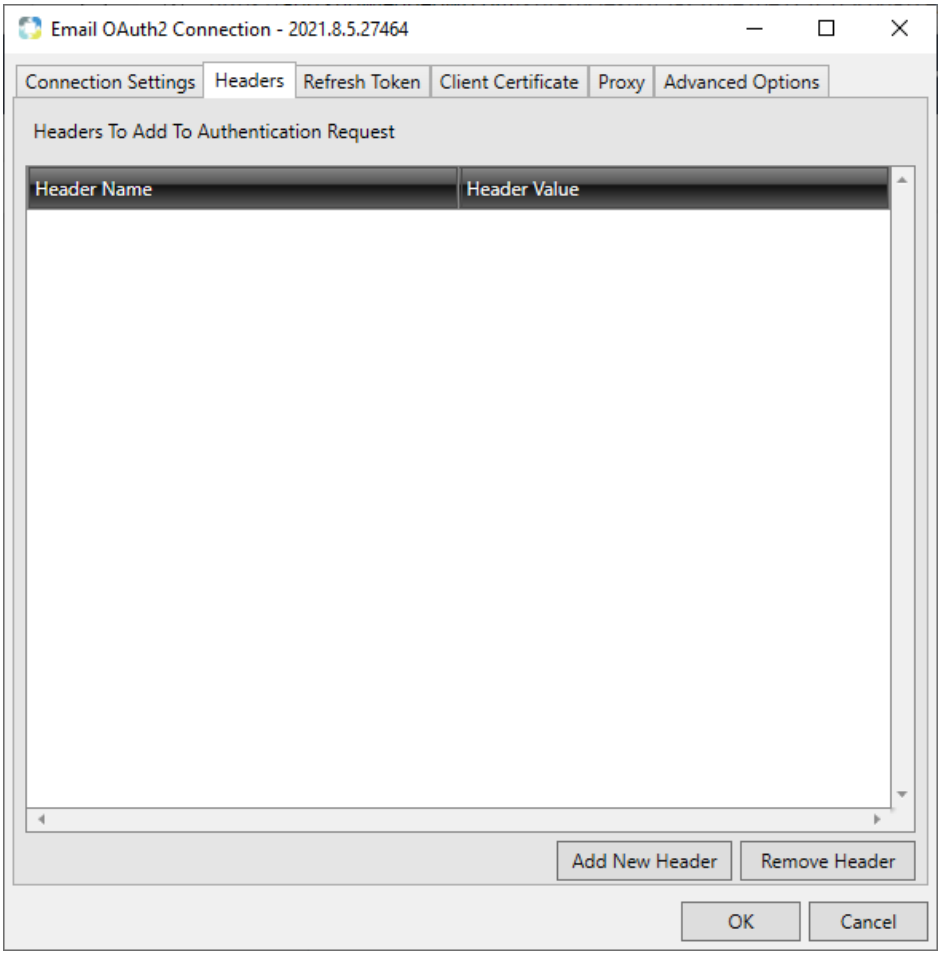


Option	Description
Server Port	Port used to connect to the client specified in Connection Settings.

Encryption Type	The encryption method used by your email server.
	• There are three available encryption types:
	  ◦ None (default)
	  ◦ SSL
	  ◦ TLS

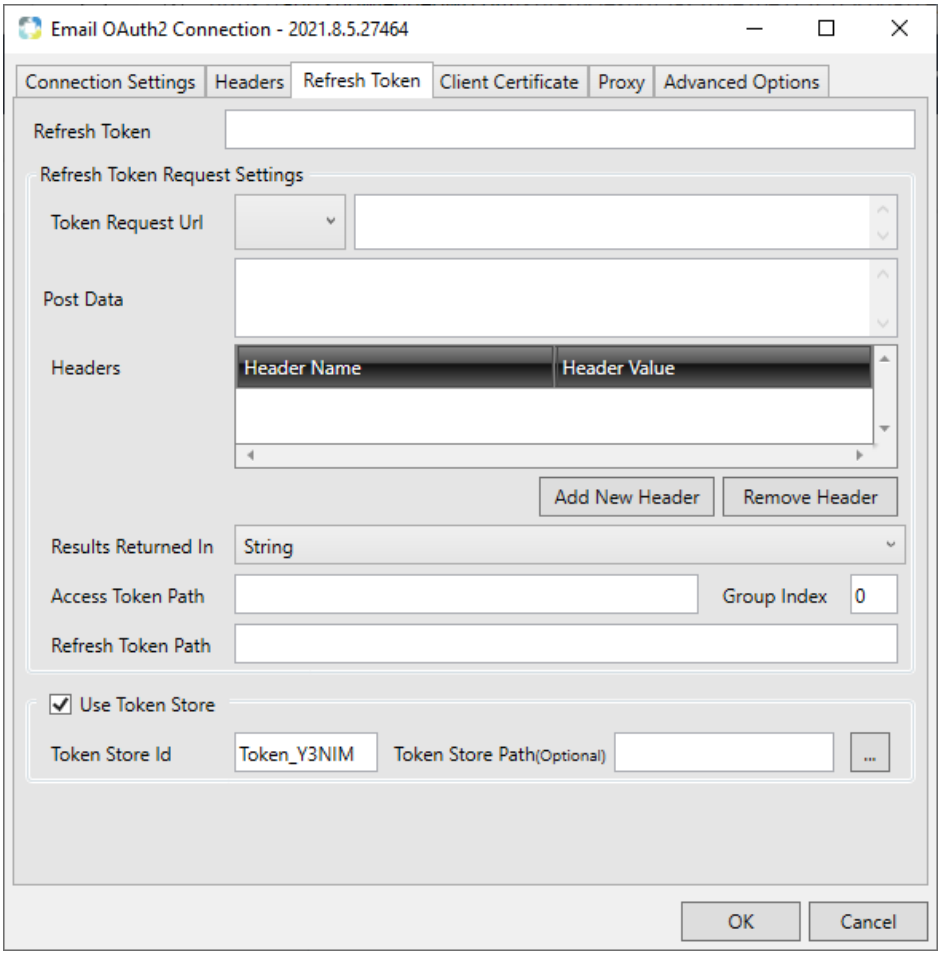
Timeout in Seconds	The number of seconds before the connection manager will timeout connecting to the mail server.
Reuse Connection	Select this checkbox to reuse the input connection.

Headers



You can create header names and values by entering the information in their corresponding windows. Select **Add New Header** to add a header. Select **Remove header** to remove a header.

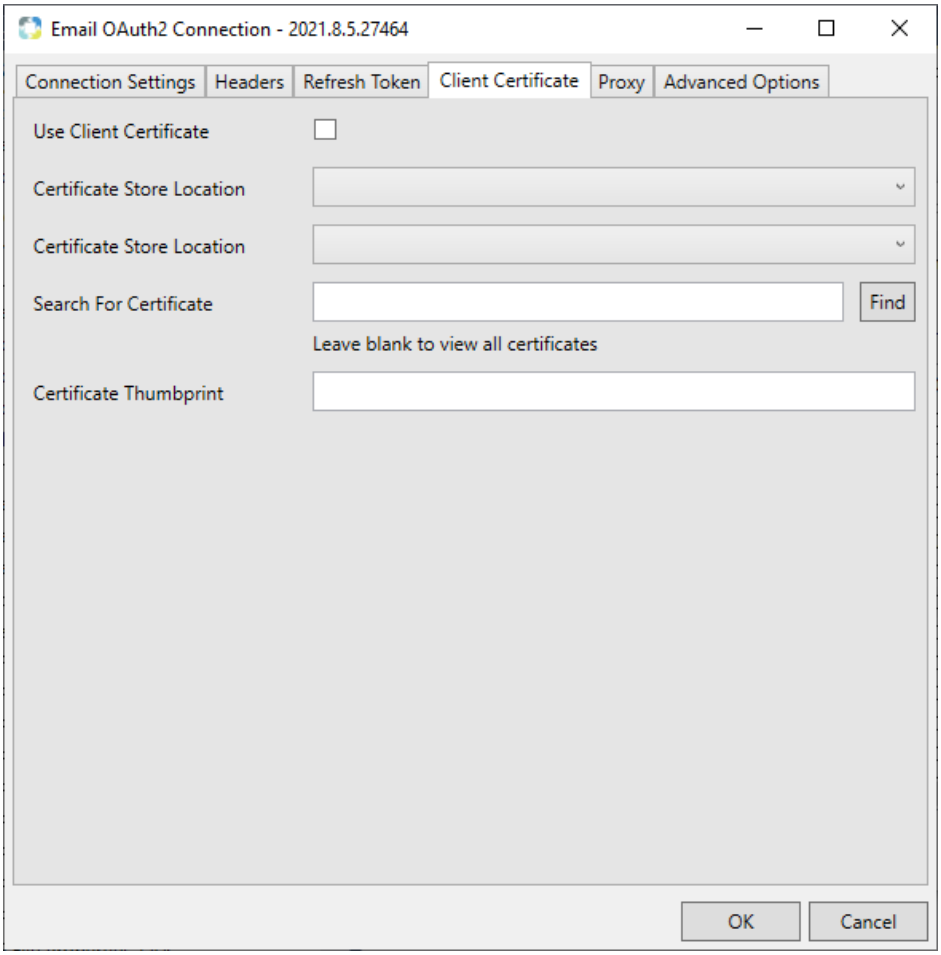
Refresh Token



Option	Description
Refresh Token	Authentication information provided to you at the application's developer site. **Note:** This should match the API Key from the Connection Settings window.
Token Request Url	The URL that returns a refresh token. Example: **https://api.citrixonline.com/oauth/access_token**
Headers	Allows you to add headers within the refresh token. Select **Add New Header** to add a header. Select **Remove header** to remove a header.
Results Returned In	Specifies how results are returned from the application (JSON, XML, or String.)
Access Token Path	The token path for the access token. Example: **access_token**
Refresh Token Path	The token path for the refresh token. Example: **refresh_token**
Group Index	Identifies the index position to return (0 based). **Note:** This option only appears if you selected String for the Results Returned In option.

Option	Description
Use Token Store	This option creates a file on your machine that saves the access token, and refresh token in an encrypted format.
Token Store Id	Identifies the Token Store Id.
Token Store Path (Optional)	Identifies the Token Store path.

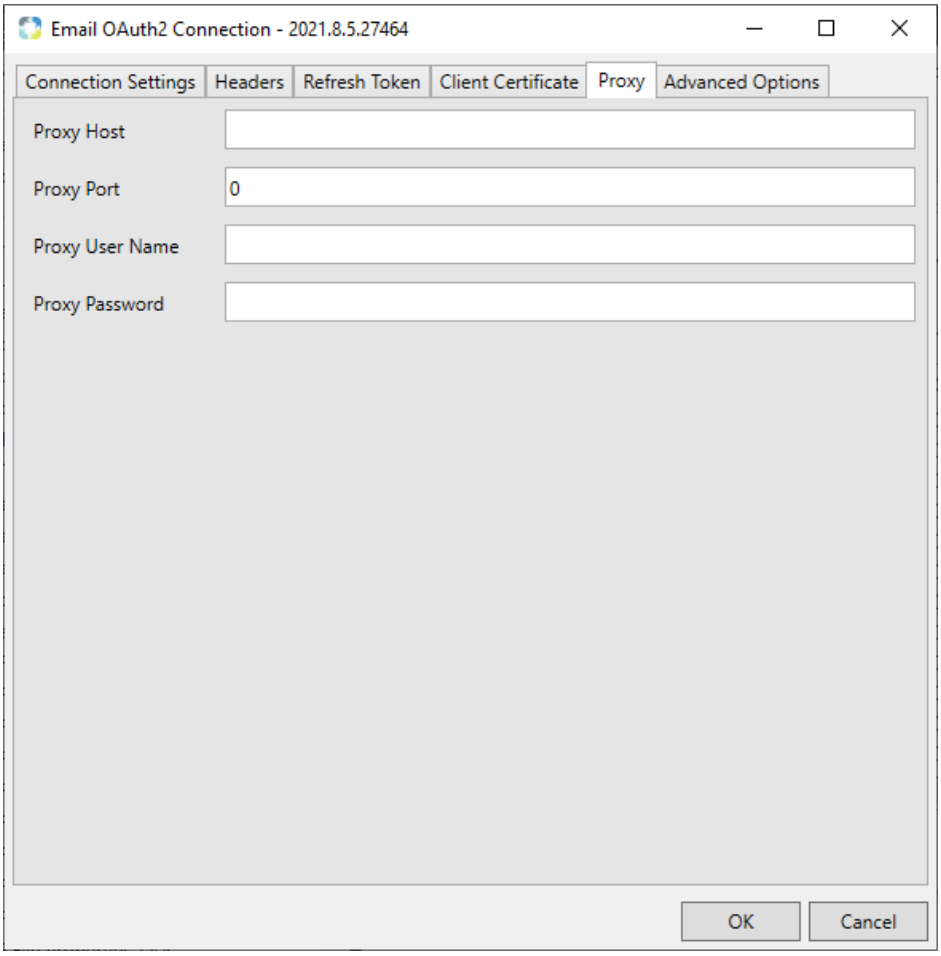
Client Certificate



Option	Description
Use Client Certificate	Select this checkbox to use the client certificate.
Certificate Store Location	Select the certificate store location: • Current User • Local Machine • PFX File (Stored on disk)

Option	Description
Certificate Store Location	Select the certificate store location: • AddressBook • AuthRoot • CertificateAuthority • Disallowed • My • Root • TrustedPeople • TrustedPublisher
Search For Certificate	Enter a specific certificate to search for and select Find. Leave this field blank to view all certificates.
Certificate Thumbprint	Enter the certificate thumbprint.

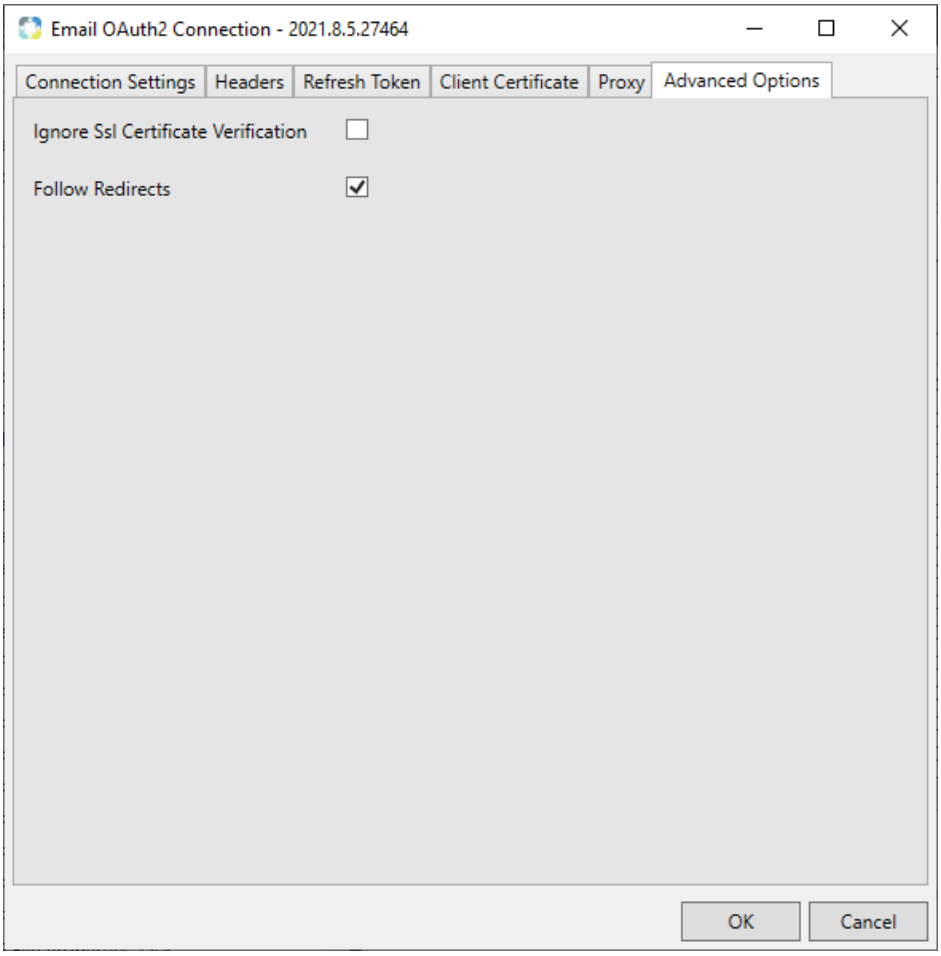
Proxy



Option	Description
Proxy Host	When connecting to an endpoint using a proxy, you should enter the proxy URL in this field.

Option	Description
Proxy Port	The port number that corresponds to the URL proxy host.
Proxy User Name	The username needed to authenticate to the proxy.
Proxy Password	The password that allows you to authenticate to the proxy.

Advanced Options



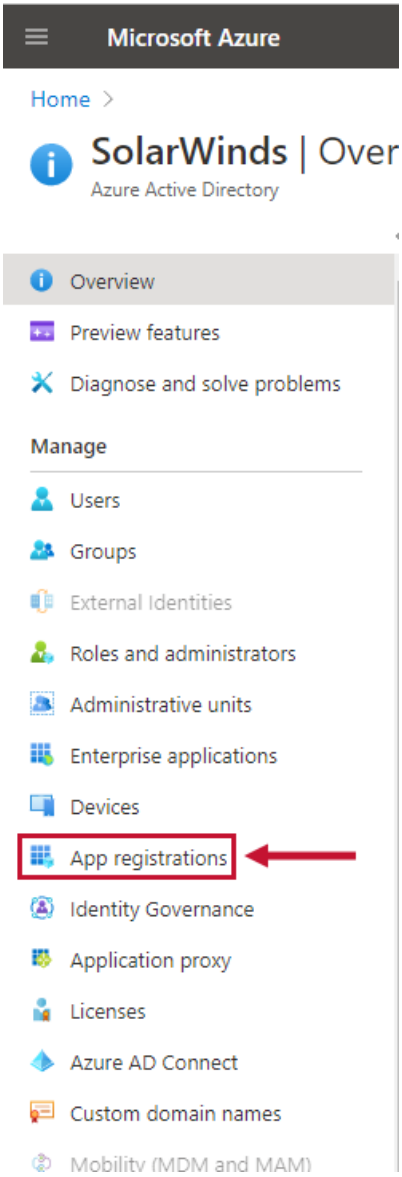
Option	Description
Ignore Ssl Certificate Verification	Selecting this option ignores the SSL Certificate notification from the API during the connection. **Note:** This option applies to APIs that require you to have an SSL Certificate. Select this option if the SSL Certificate is not needed.
Follow Redirects	Selecting this option implements a 301 redirect on configured endpoints. For example, this option would automatically send your request to an HTTPS redirect once the request has reached the HTTP version of the site.

Configuring your Azure App for Email OAuth2 Connection Manager

Before connecting to your Azure App with the Email OAuth2 Connection Manager, you need to configure your settings in Azure Active Directory. Complete the following steps:

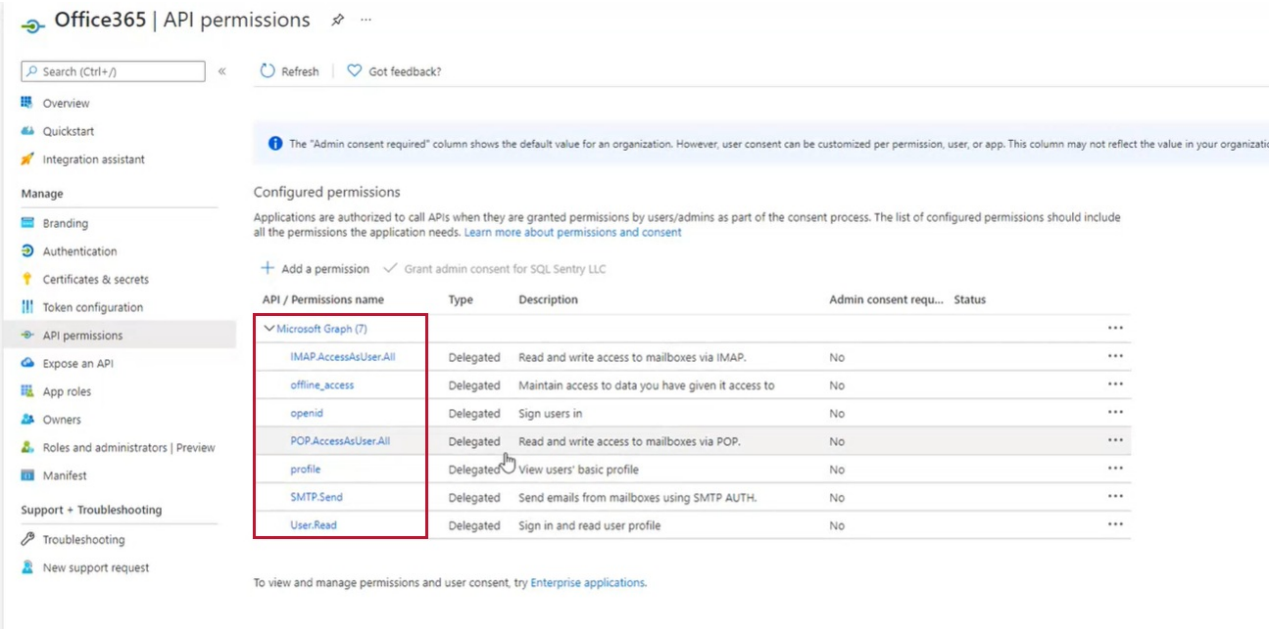
> **Additional Information:** If you haven't configured an App to connect to your web API for the Email OAuth2 Connection Manager, see the following MSDN article for more information about configuring an app.

1. Login to Azure Active Directory, and go to **App registrations**.



2. Select the Azure App you want to use for your connection.

3. Select **Manage** > **API Permissions** to open your App's permissions. Enable the appropriate scopes for your connection. In this example we have enabled the following: **IMAp.AcessAsUser.All**, **offline_access**, **openid**, **POP.AccessAsUser.All**, **profile**, **SMTP.Send**, and **User.Read**.

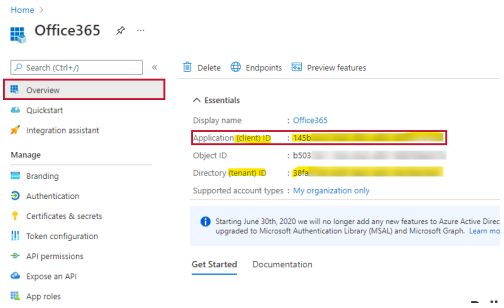


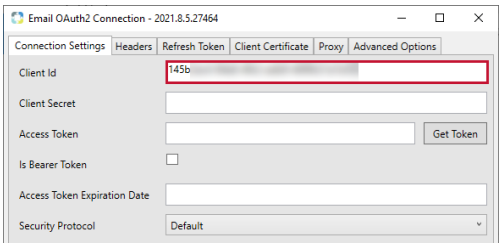
Configuring Email OAuth2 Connection Manager Example

After configuring your App and permissions in Azure Active Directory, you can configure your Email OAuth2 Connection Manager. Complete the following steps:

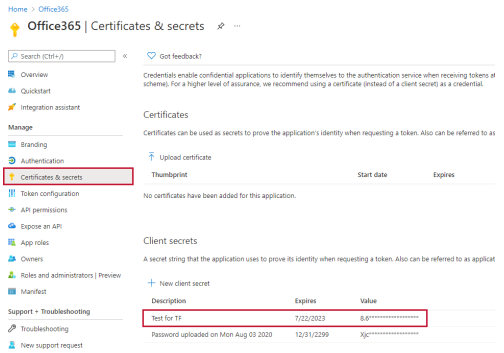
1. Add the Email OAuth2 Connection Manager to your package.

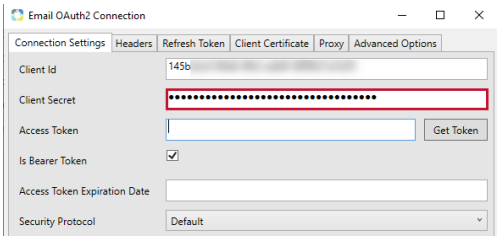
2. Copy the **Application (client) ID** from your Office365 App Overview page in the **Client Id field**.



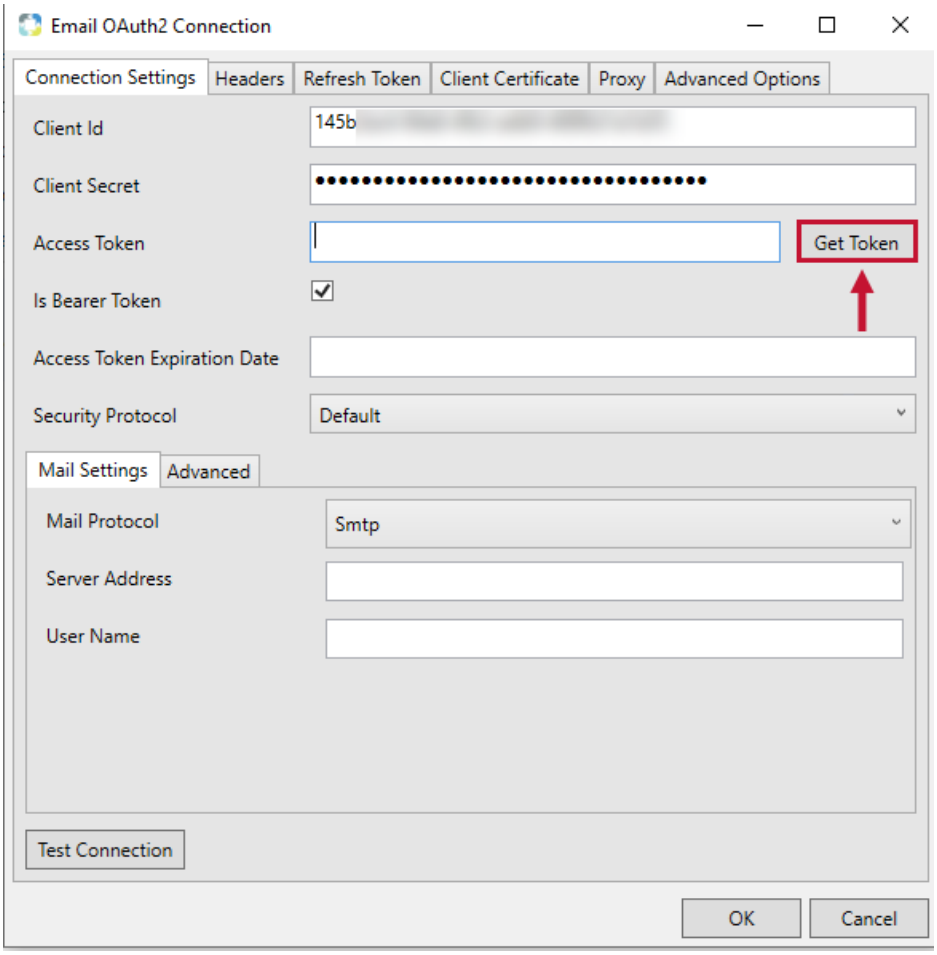


3. Copy your **client secret** from your Office365 Certificates & secrets page in the **Client Secret field**.

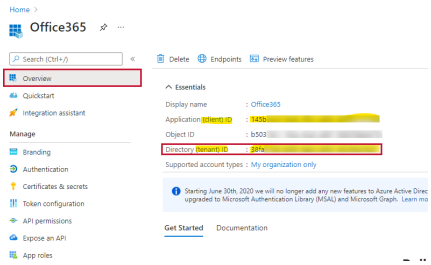




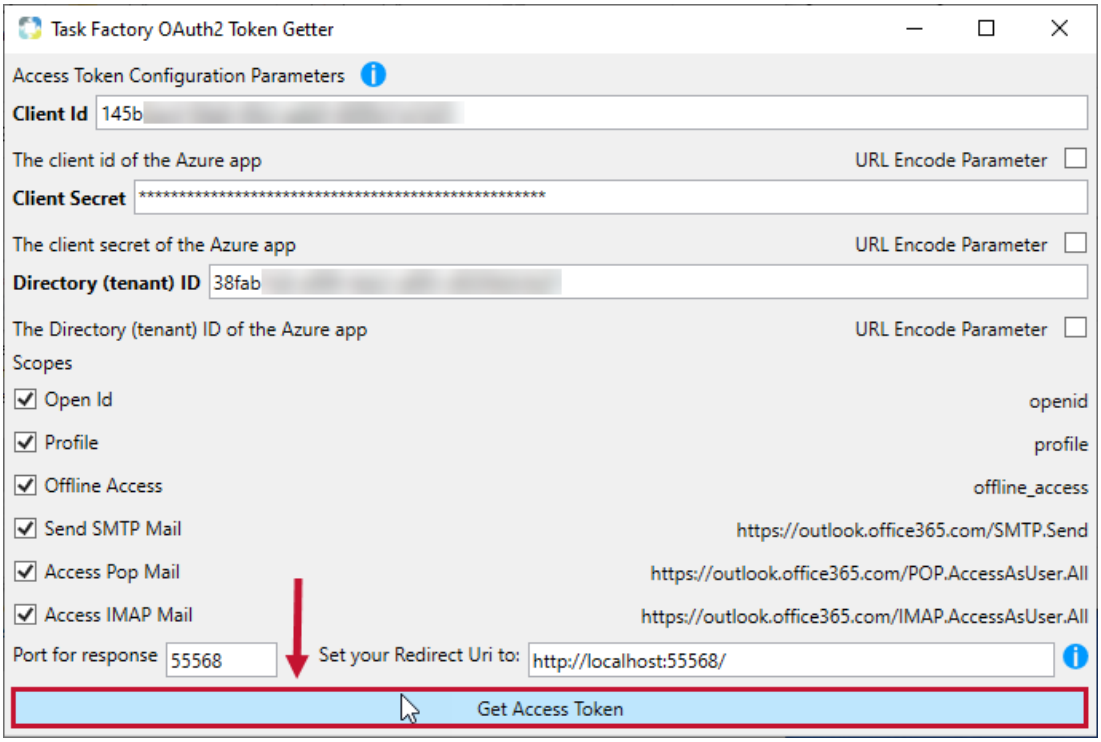
4. Select **Get Token** to open the Token Getter.



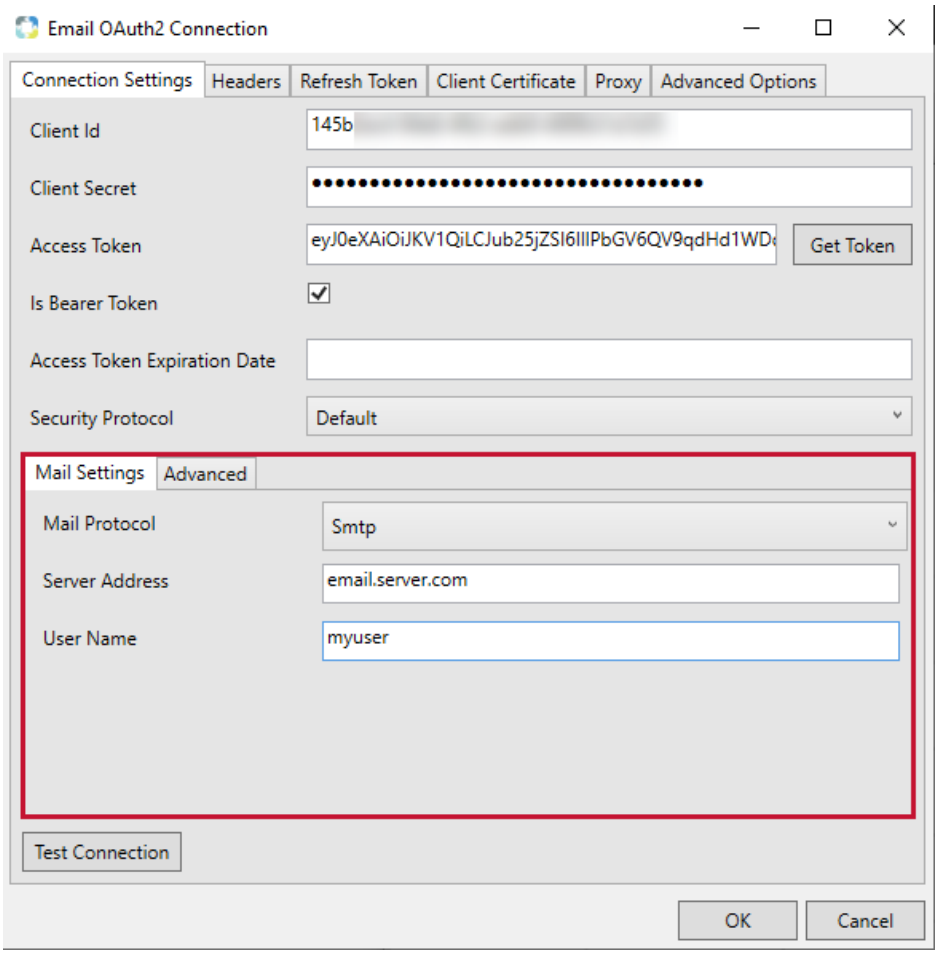
5. Copy the **Directory (tenant) ID** from your Office365 App Overview page in the **Directory (tenant) ID field**.



6. Select the checkbox(es) for the desired connection scopes, and then select **Get Access Token**.



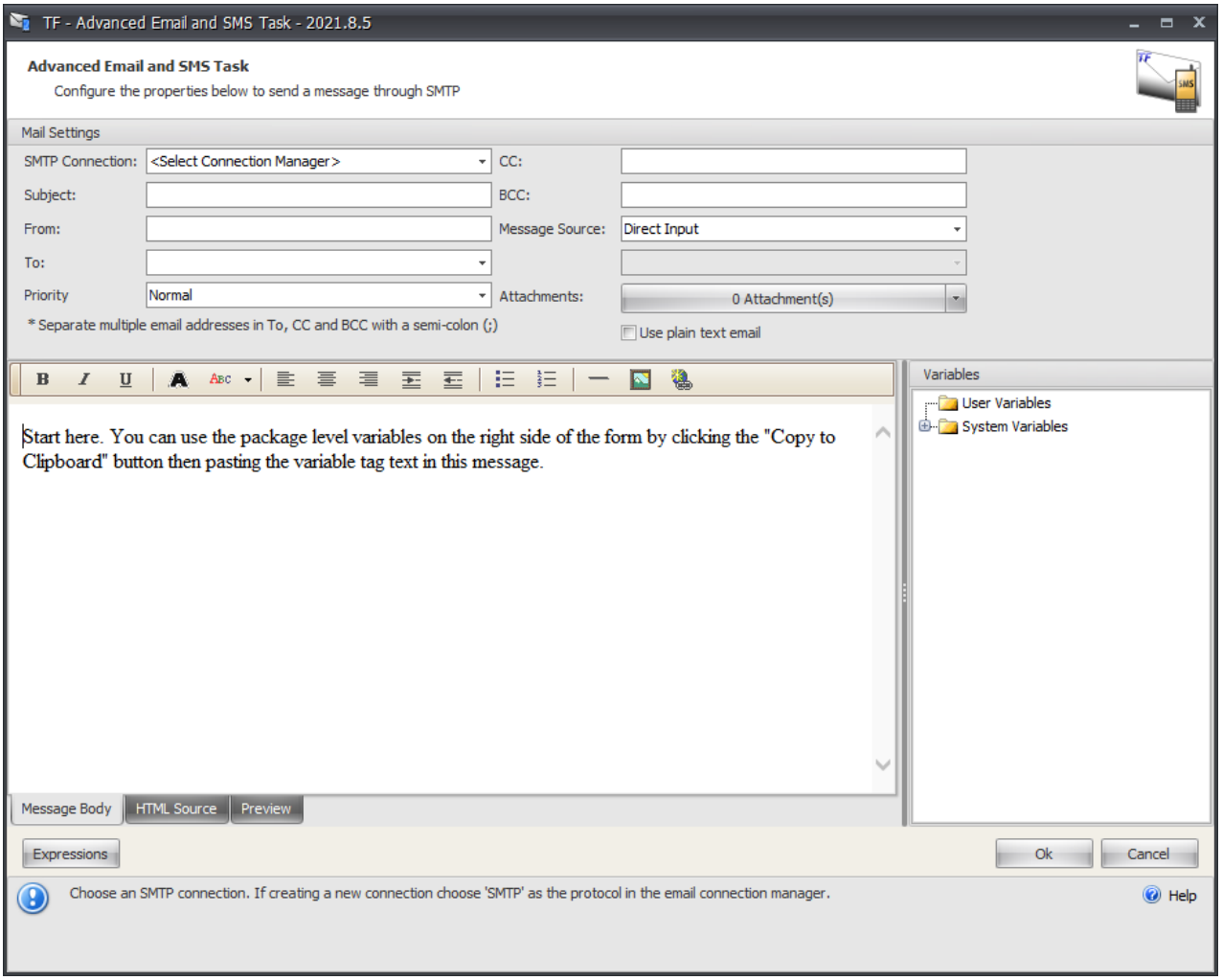
7. Enter your App's mail settings in the Mail Settings tab, and then select **Ok** to complete your connection.



Advanced Email and SMS Task

Task Icon	Task Description
	The **Advanced Email and SMS Task** is used to send multiple Email or SMS messages to multiple users at the same time.

Mail Settings



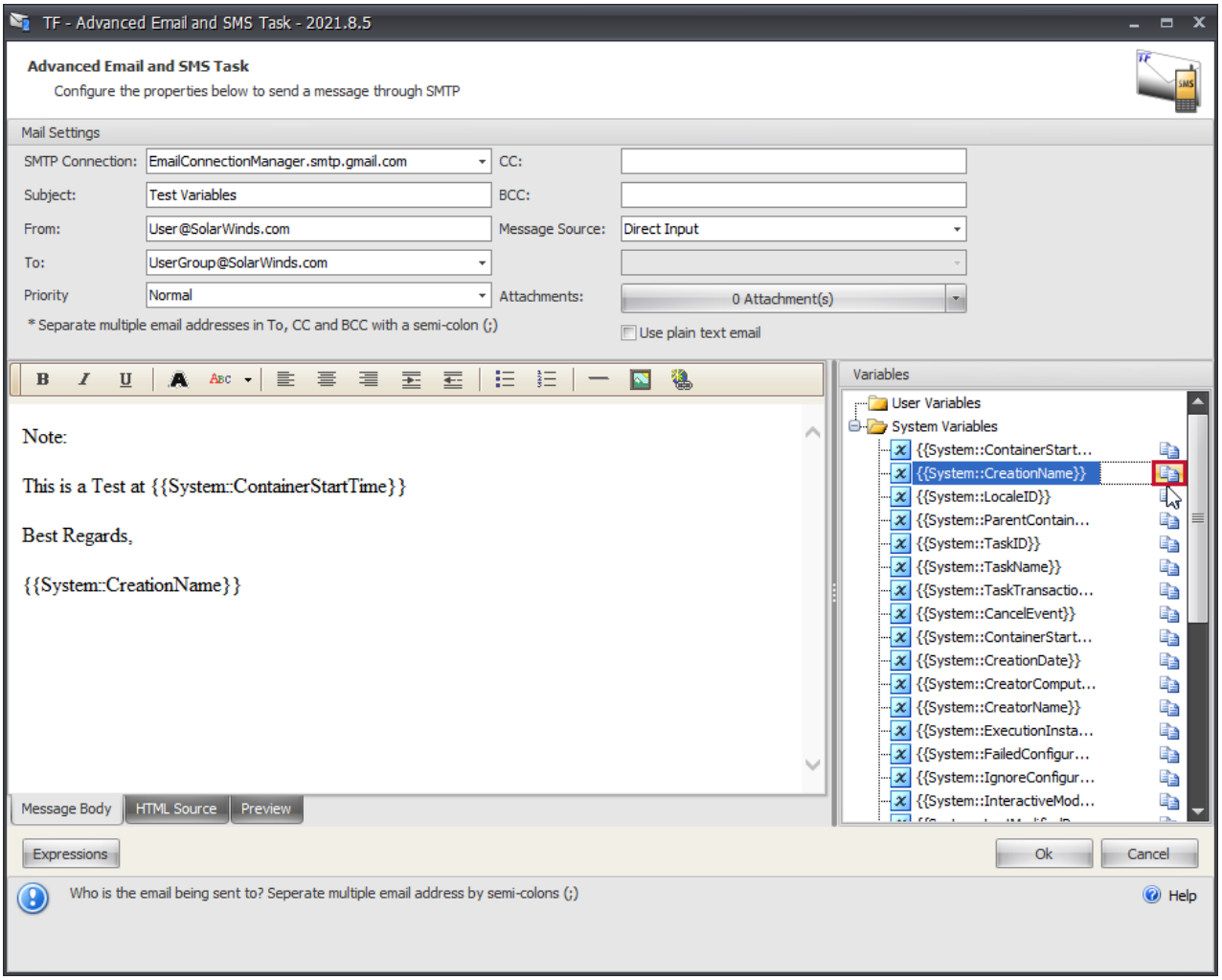
Option	Description
SMTP Connection	The email connection manager used to send the message.
Subject	The subject of the email being sent.
From	The email address being used to send the email.
To	The email addresses or SMS addresses the email will be sent to.
Priority	The message's level of importance.
CC	The carbon copy email addresses.
BCC	The blind carbon copy email address.

Option	Description
Message Source	The source of the message being sent. • **Direct Input** - Uses the message created in the HTML or Plain Text editor in the UI. • **File** - Uses the file in the **File Connection Manager** selected. You will be able to edit the file content within the UI. When you select the **OK** button the file contents are replaced with the contents within the HTML editor or Plain Text editor. • **Variable** - Uses the text contained within the variable selected. The variable text can be edited within the UI. When the **OK** button is selected the variable contents are replaced with the contents within the HTML editor or the Plain Text editor.
Attachments	Attachments sent with the email. When using variable expressions to send more than one file, use a pipe delimiter or | to separate each file's path. (Example: C:\users\me\File1.txt|C:\users\me\File2.txt)
Use Plain Text Email	Selecting this checkbox will remove the HTML editor from the UI and add a Plain Text Editor.

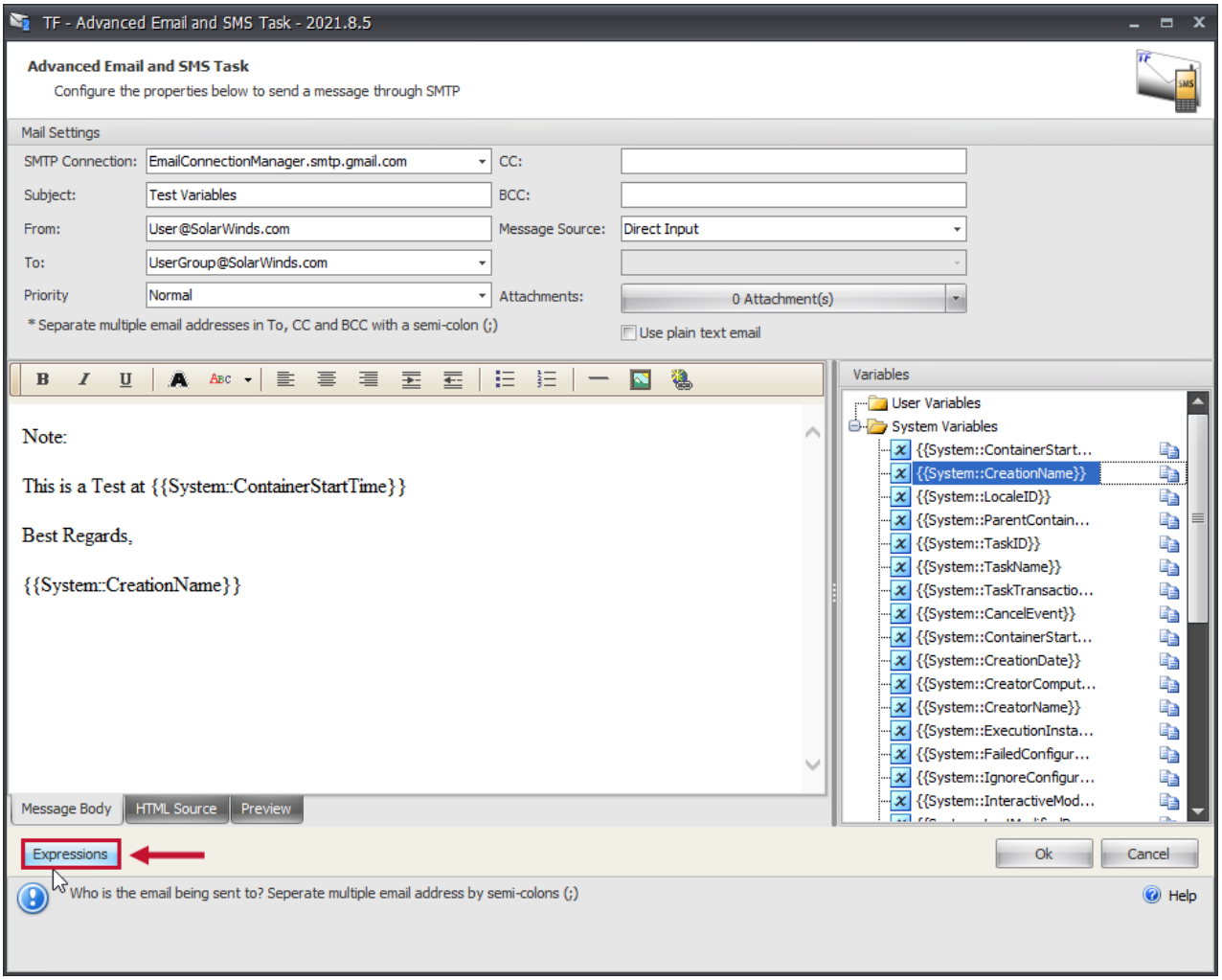
Variable Window

The package level user variables and system variables can be used within the message being sent. The placeholder is replaced with the value of the variables.

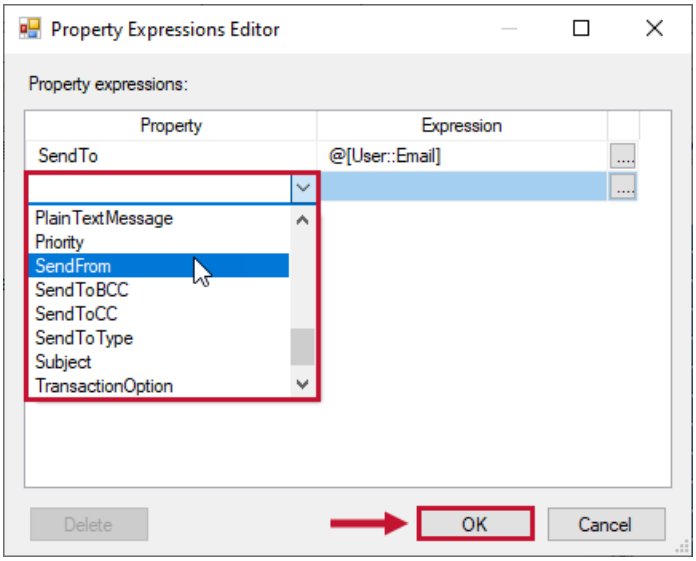
You can add a variable by selecting to the right of the variable name and then pasting it into the message body.



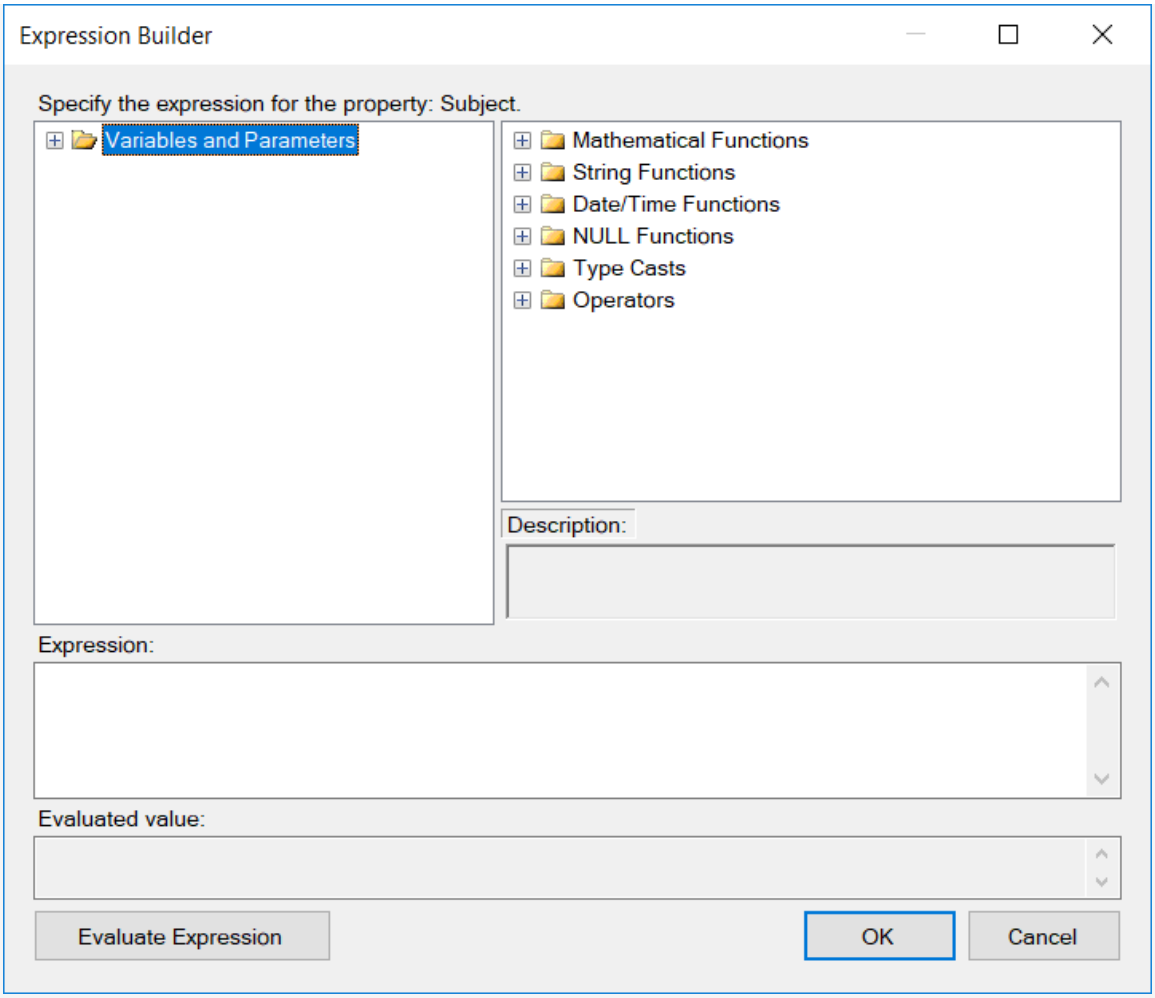
To add variables to any of the **Mail Settings** options, select the **Expressions** button to open the **Property Expression Editor**.



Select the desired property from the drop-down list, and then enter the expression. Select **OK** to save the expression(s).

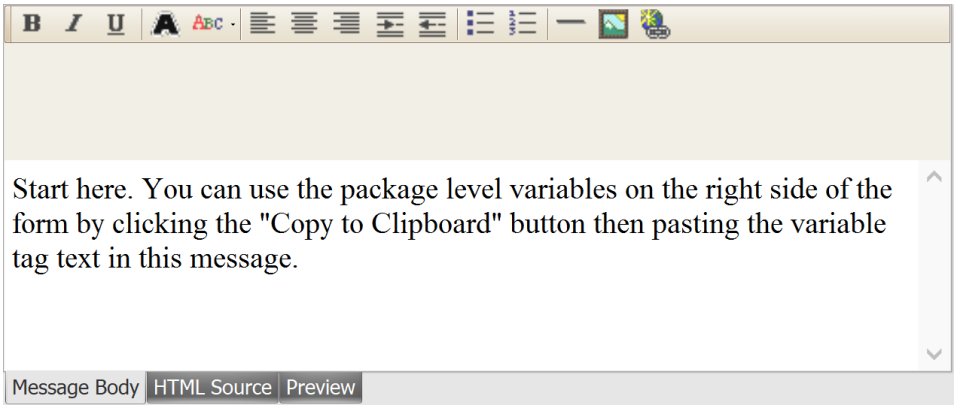


Note: Select the ellipsis to open the **Expression Builder** and build more complex expressions.



Additional Information: See the Expressions article for more details on using this functionality.

Message Body Tab



HTML Source Tab

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD>
<META content="text/html; charset=unicode" http-equiv=Content-Type>
<META name=GENERATOR content="MSHTML 11.00.10570.1001"></HEAD>
<BODY>
<P>Start here. You can use the package level variables on the right side of the
form by clicking the "Copy to Clipboard" button then pasting the variable tag
text in this message.</P></BODY></HTML>
```

Message Body | HTML Source | Preview

Use the HTML editor to edit the body of an HTML email.

Preview Tab

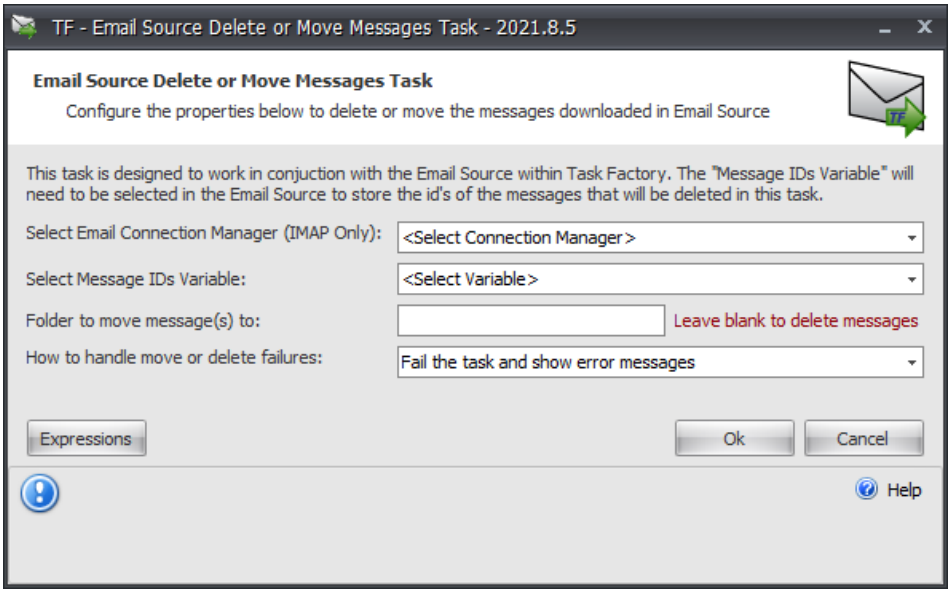
Start here. You can use the package level variables on the right side of the form by clicking the "Copy to Clipboard" button then pasting the variable tag text in this message.

Message Body | HTML Source | Preview

Use the preview tab to see what your email or SMS looks like.

Email Source Delete or Move Messages Task

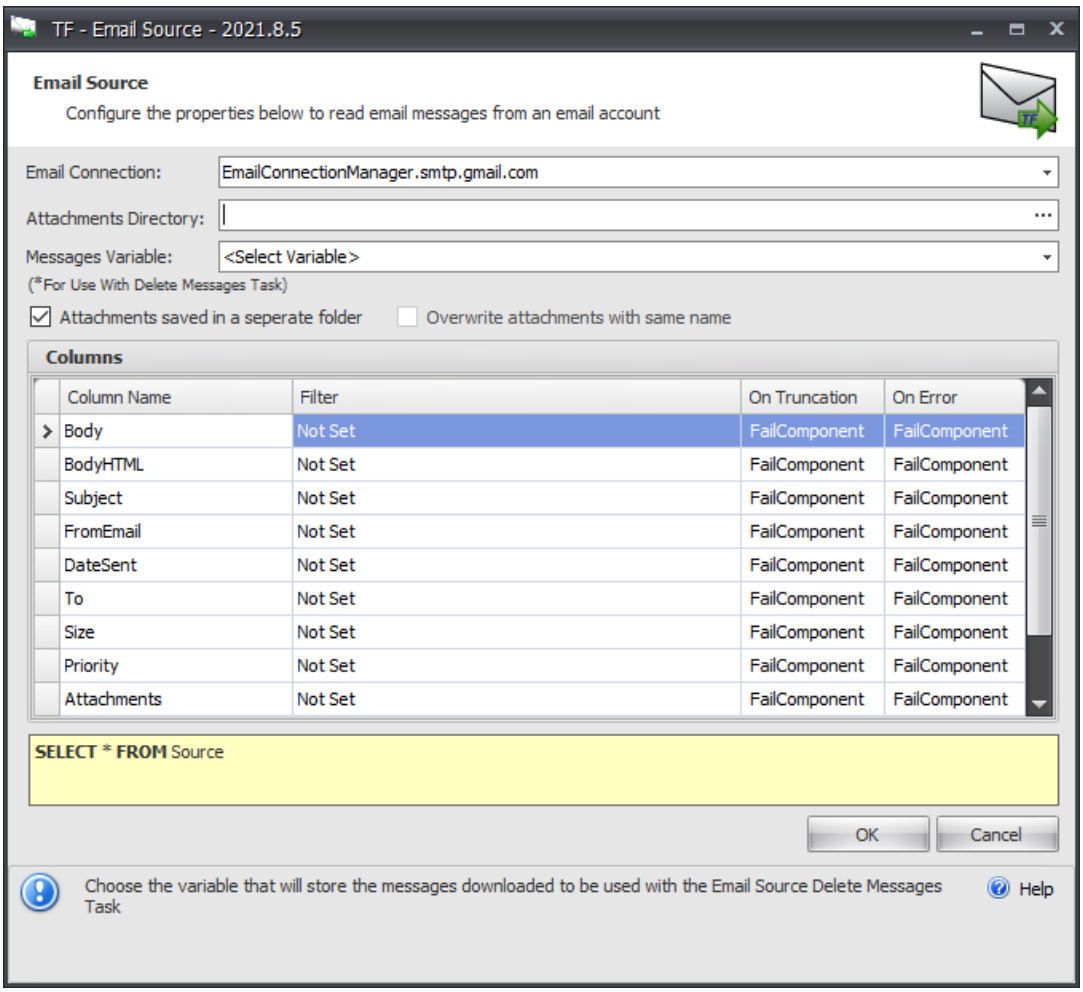
Task Icon	Task Description
	The **Email Source Delete or Move Messages Task** allows the user to move or delete messages downloaded using the Email Source. See the Email Connection Manager to learn more about setting up the connection manager.



Option	Description
Select Email Connection Manager (IMAP Only)	Select or create a connection manager.
Select Message IDs Variable	Choose the variable for the message ids.
Folder To move messages to	Choose a folder to move messages to or leave blank to delete them.

Email Source

Source Icon	Source Description
	The Email Source allows you to setup the properties to read messages from a POP3/IMAP mailbox. You can setup filtering for the data coming in from the mailbox to prevent unwanted messages or to target specific messages. See the Email Connection Manager to learn more about setting up the connection manager.



Option	Description
Email Connection	Choose an existing Email Connection Manager or select **Create New Connection**.
Attachments Directory	The attachments directory is where any attachments for email messages are stored. Download all attachments into a single selected directory. The location of each attachment is stored in the **Attachments** column of the Output.
Messages Variable	The messages variable is an object variable that can be used by the **Delete or Move Messages Task** which allows you to delete messages retrieved in the email source.

Columns

Column Name - All email source adapters include the following fields to be used by your package.

Column Name	Description
Body	The plain text version of the email.
BodyHTML	The HTML version of the email.
Subject	Subject of the email.
FromEmail	Email address only of the person who sent the email.
DateSent	The date the email was sent.
To	Email address of the account the message was sent to.
Size	The size in bytes of the message.
Priority	The priority of the message.
Attachments	The attachments of a message.
RelatedItems	The items embedded into a message. Usually images in message footers.

Filter - The Email Source allows you to filter the data coming in from the source mailbox. There are different filters allowed for different data types.

Data type	Usable Filters
String Columns	Allow Contains, Equals, Ends With and Start With filters.

Data type	Usable Filters
Integer Columns	Allows Greater Than, Less Than, and Equal filters.
Data Columns	Allow After and Before filters.

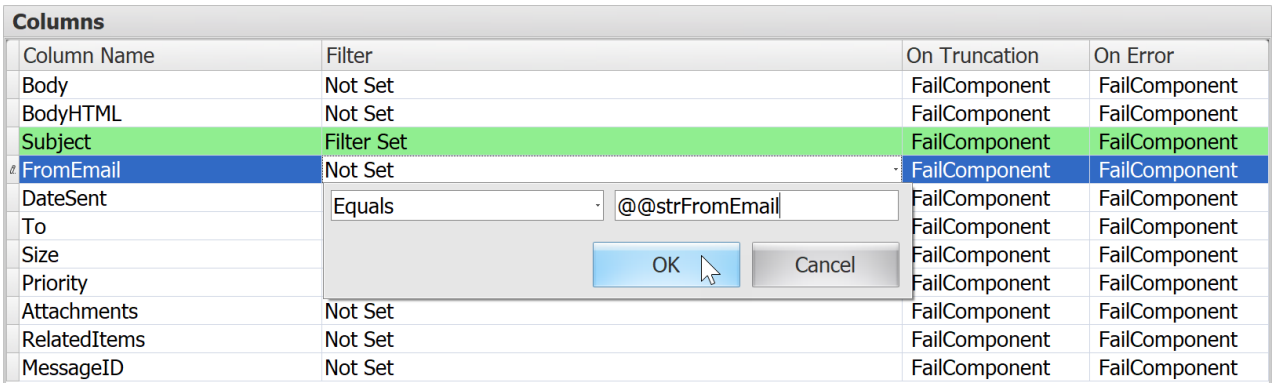
Condition Textbox - Here you can use variables to use with filtering.

Column Filtering

The Email Source allows you to filter the data coming from the source mailbox using simple expressions. There are different filters allowed for each data types:

Data Type	Usable Filters
String Columns	Allows for the usage of the Contains, Equals, Ends With, and Starts With filters.
Integer Columns	Allows for the usage of the Greater Than, Less Than, and Equals filters.
Date Columns	Allows for the usage of the After and Before filters.

Additionally, filters allow for the use of variables within the filter's condition textbox. This can be done by appending two ampersats (@@) to the front of the variable name. For instance, the variable named **strFromEmail** could be applied to the **FromEmail** column in the following manner:



After the filters are set, each row that contains a filter turns green and displays a pseudo-select statement that allows for a quick view of all of the filters that are set within the Email Source.

Examples

How To Connect To An Office 365 Shared Mailbox

See the support article How To Connect To An Office 365 Shared Mailbox Using TF – Email Source for an example of how to read messages from an Office 365 shared mailbox using the Email Source.

Passing a Variable for the "To" line of the Advanced Email/SMS Task

See the support article Task Factory - Passing a Variable for the "TO" line of the Advanced Email/SMS Task for an example of how to pass a variable using the Advanced Email and SMS task.

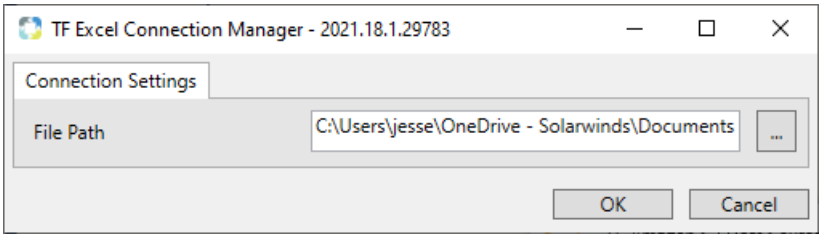
Task Factory Excel

Last Modified on 03 February 2022

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Excel Connection Manager

This connection manager is used by the **Excel Source** and **Excel Destination** components. Both components are located in the data flow. To select a file, select the **ellipsis** > **navigate to** and then select the Excel file.



Excel Power Refresh Task

Important:

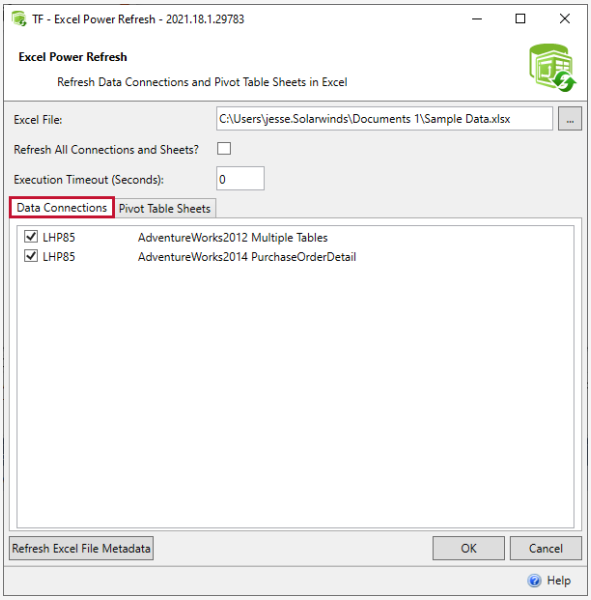
- Microsoft Office must be installed on the environment where the package is located.
- Excel Power Refresh is available for SQL versions 2012 and higher.

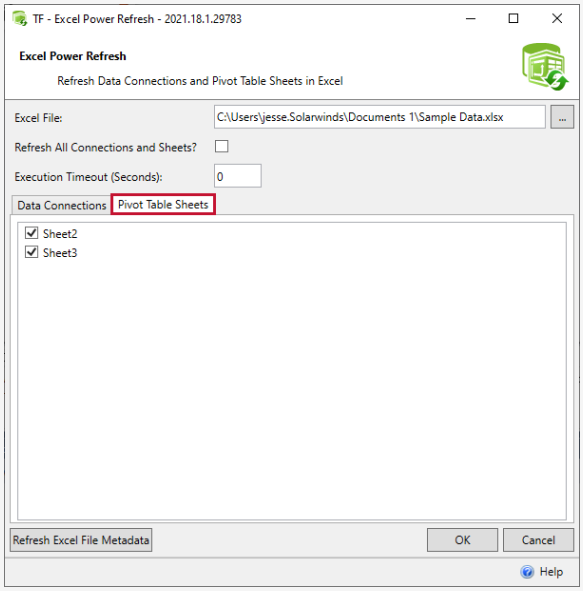
Task Icon	Task Description

Task Icon	Task Description
	The **Excel Power Refresh Task** is a control flow component that refreshes an Excel file with Power Query connections and Power Pivot tables.

Option	Description
Excel File	Defines the path to the Excel file to be refreshed. **Note:** **.xlsx** and **.xlsm** files are supported.
Refresh All Connections and Sheets?	Toggle that selects all Power Query Connections and Pivot tables within the Excel file.
Execution Timeout (Seconds)	Defines the allotted time for the refresh to execute in seconds. **Note:** The timeout cannot be used for connections that prompt users for their credentials.

Power Query Connections	Power Pivot Sheets
Select individual connections to be refreshed in the selected Excel file.	Select individual Power Pivot tables to be refreshed in the selected Excel file.



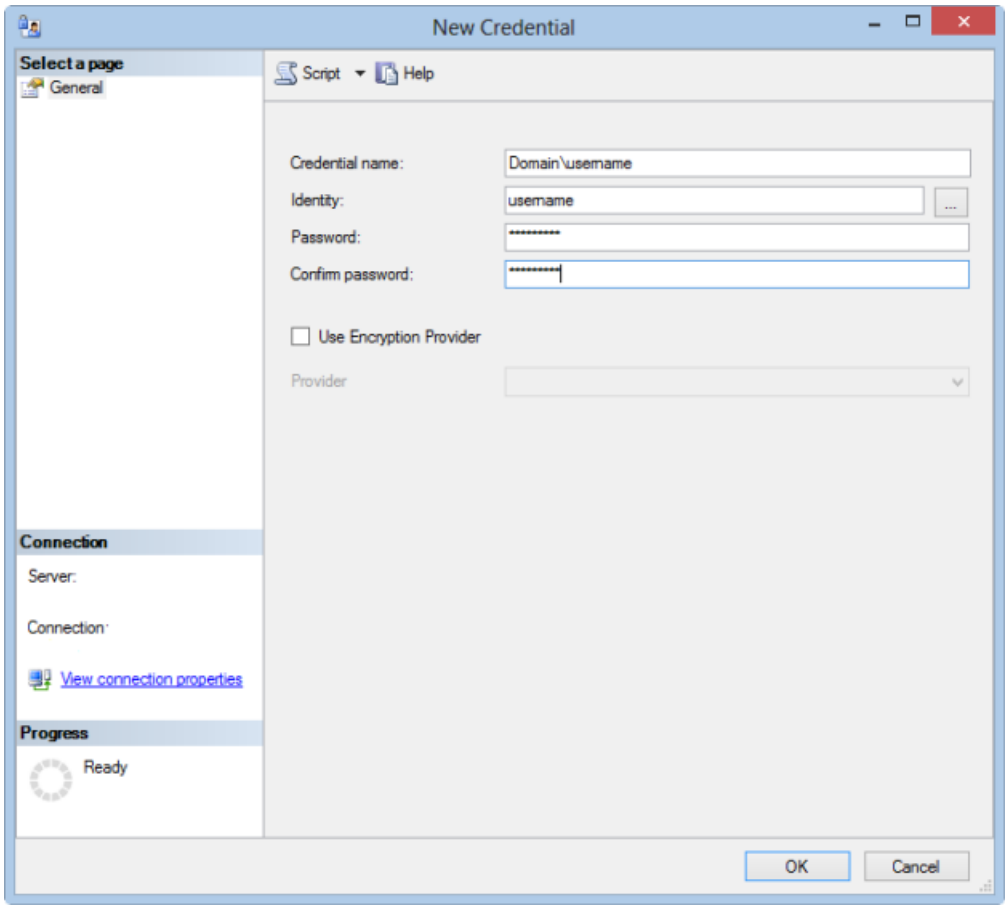


Configuring Excel Power Refresh for SQL Agent Jobs

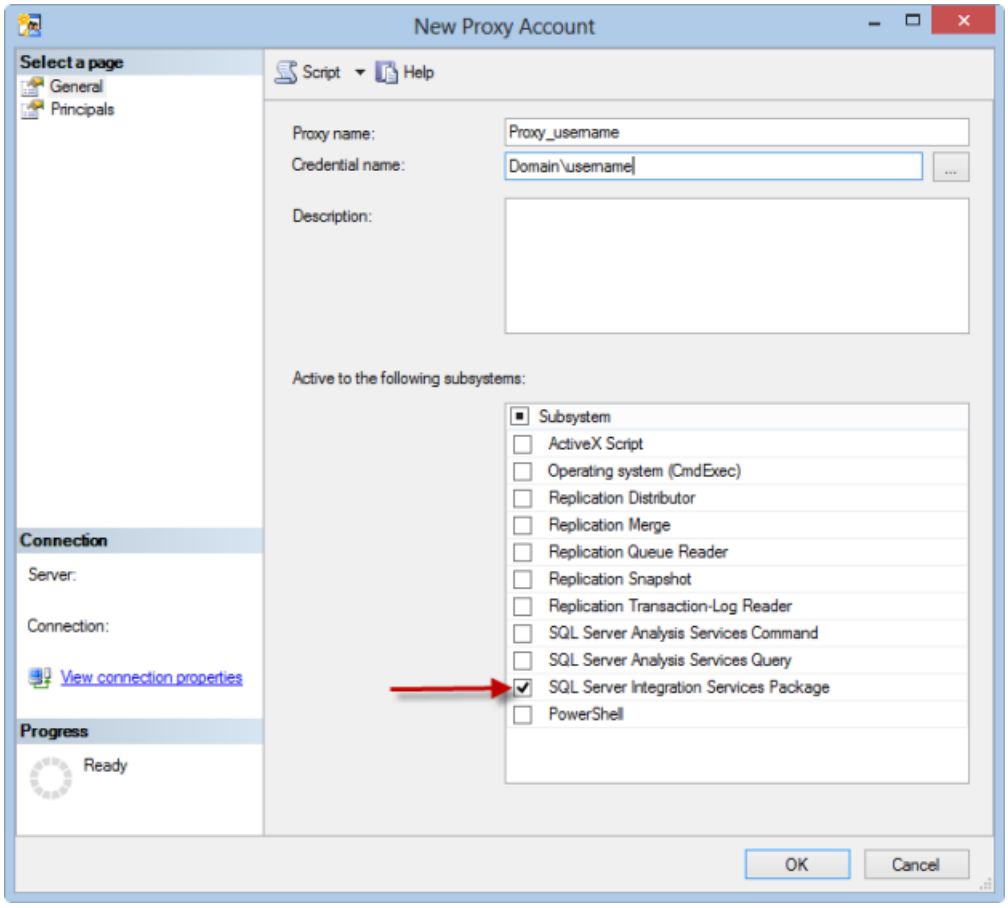
Excel Power Refresh within a SQL Agent Job must be done by using a proxy account. Please use the following

steps prior to configuring your SQL Agent Job:

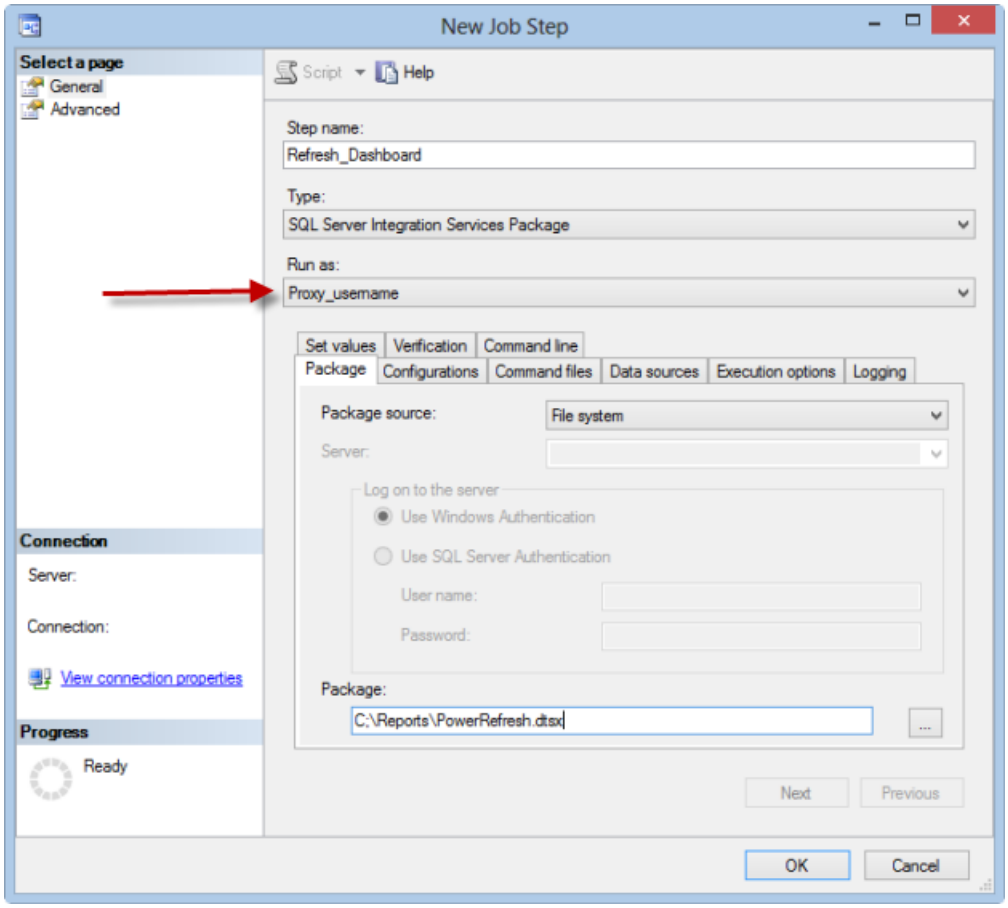
1. Create a credential based on your login (found within the Security folder in SSMS):



2. Create a proxy account based on the previously created credential. Select the SSIS Package Integration Services Package subsystem:



3. Set the job step to run as the proxy account:



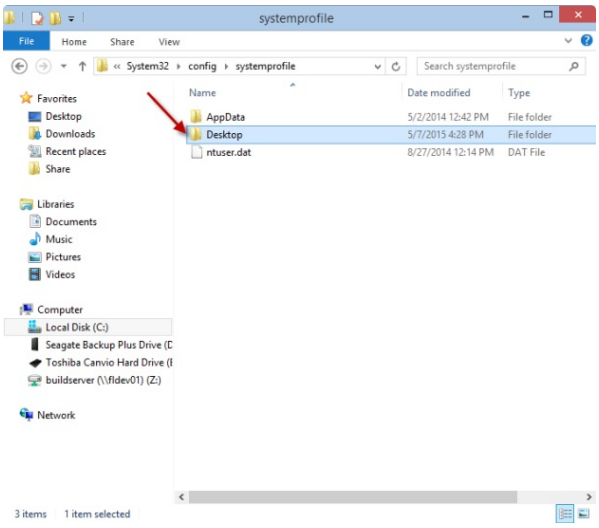
4. Start the job.

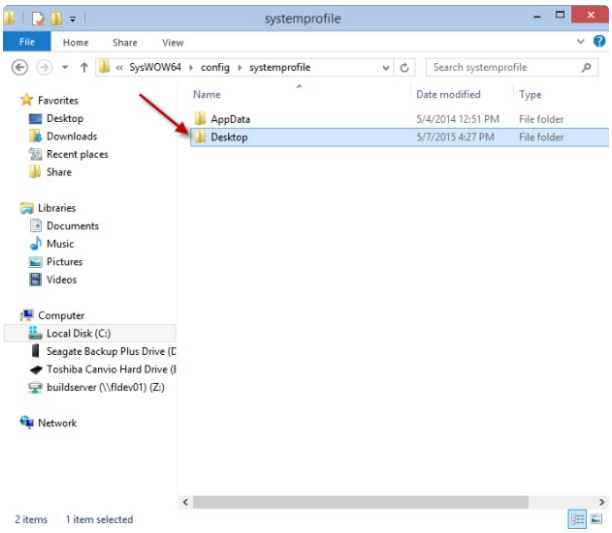
5. If you receive the following error:

```
Code: 0x00000000    Source: TF Excel Power Refresh Task ExecuteTask Failed:    Description: Microsoft Excel cannot access the file .xlsx'. There are several possible reasons:    ? The file name or path does not exist.  ? The file is being u sed by another program.  ? The workbook you are trying to save has the same name as a currently open workbook.  E nd Error  DTExec: The package execution returned DTSER_FAILURE (1).  Started:  1:28:39 PM  Finished: 1:28:44 PM  El apsed:  4.735 seconds.  The package execution failed.  The step failed.
```

This issue can be resolved by adding a Desktop folder to the following directories:

- C:\Windows\SysWOW64\config\systemprofile\
- C:\Windows\System32\config\systemprofile\





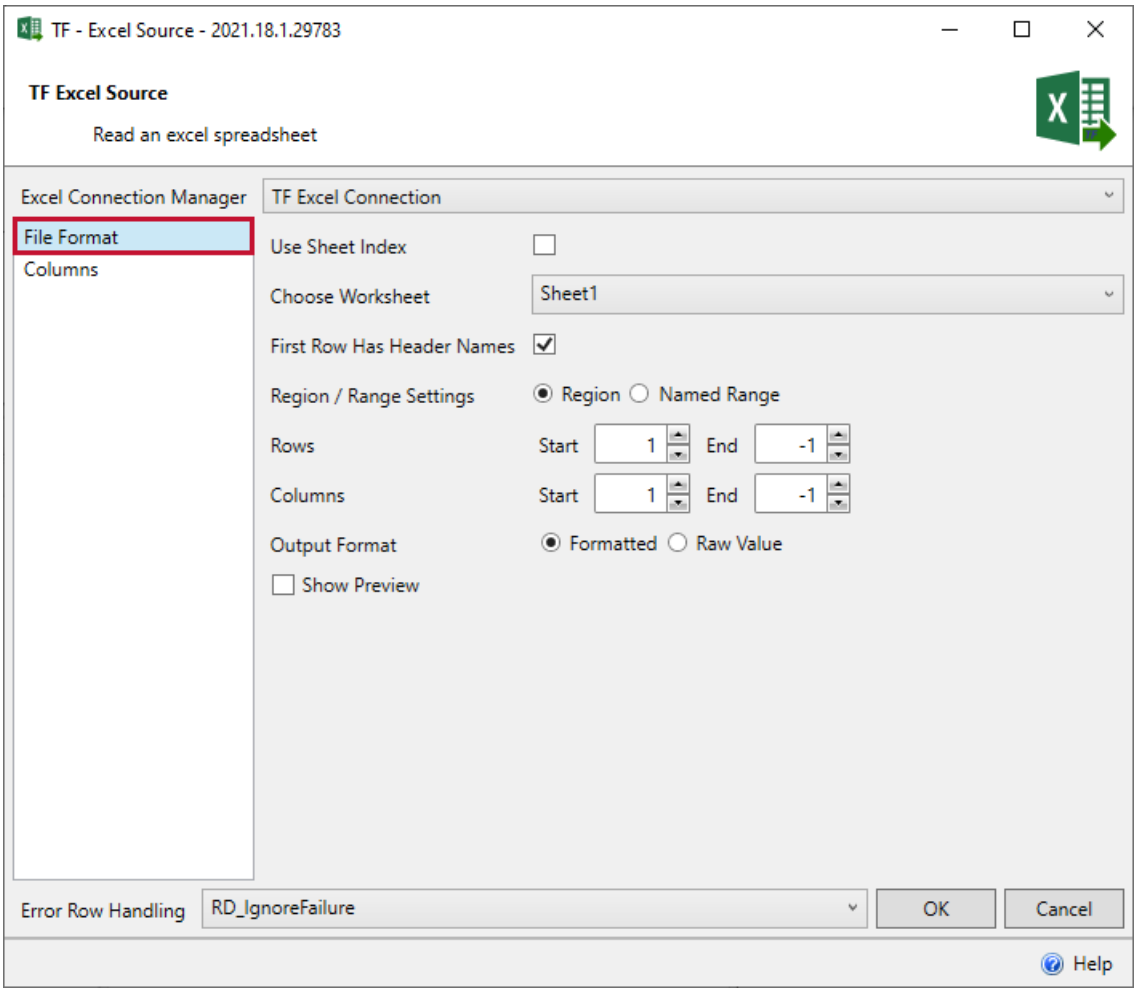
6. Restart the job.

Excel Source

Important: Excel Source is available for SQL versions 2012 and higher

Source Icon	Source Description
	The **Excel Source** connects to an existing Excel file and extracts its data.

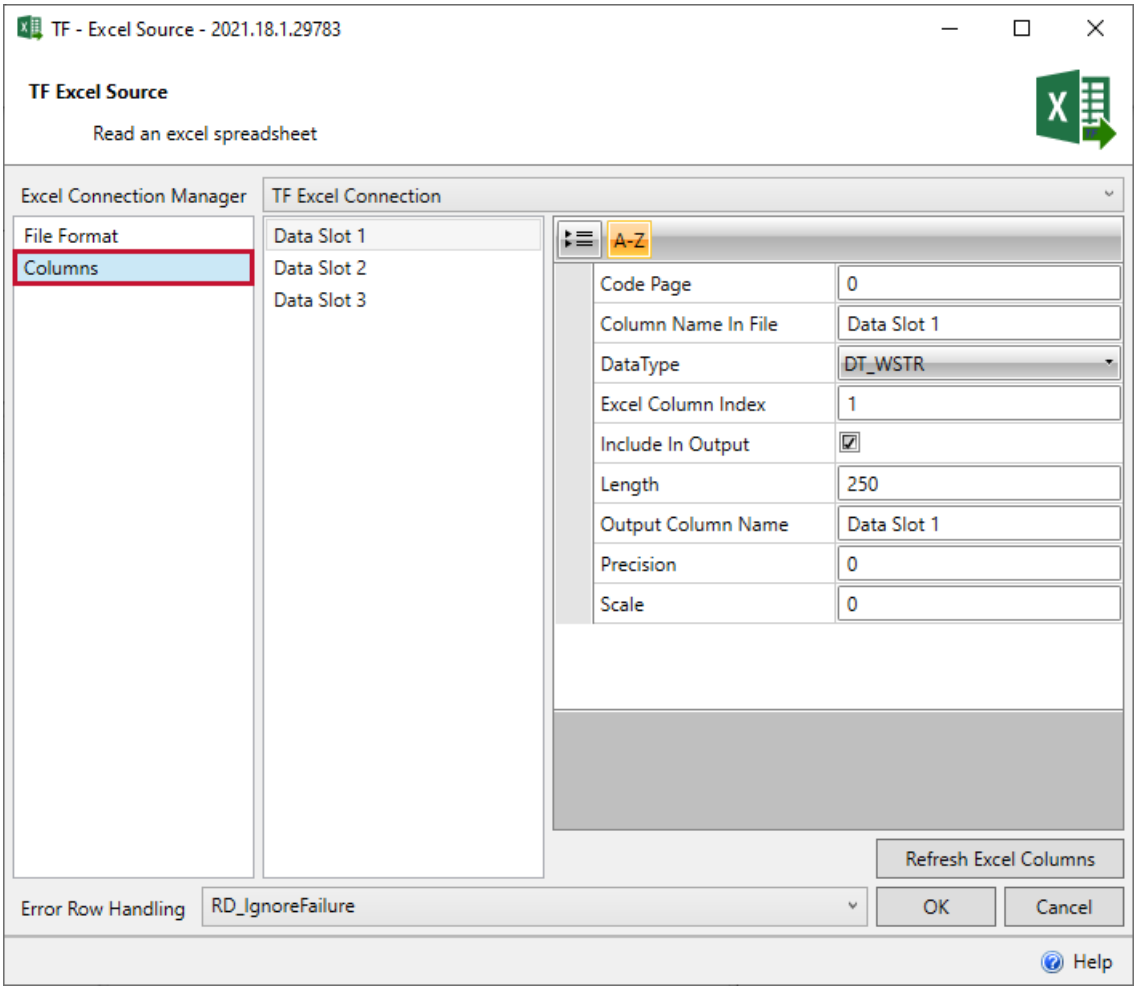
File Format



Option	Description
Excel Connection Manager	Create a new or select an existing connection to an Excel worksheet.

Option	Description
Choose Worksheet	For Excel files that use many worksheets, you can choose which one to extract data from. **Note:** You can configure the Excel source component to use sheet names that are dynamically set by a variable with an Expression in your Data Flow Task.
First Row Has Header Names	This option is selected if the first row contains headers. If the first row contains data, remove this option and configure the columns in the **Columns** tab.
Region / Range Settings	Choose from the following: • **Region** - Use the Row and Column values below to define the data range. • **Named Range** - Select from named ranges created in an Excel file. When Named Range is selected, a dropdown box appears. Choose from the available named ranges within the document.
Rows	Determines the starting and ending row. **-1** denotes using all available rows. **Note:** When the first row contains headers, it counts as the first row when starting your range at row **1**.
Columns	Determines the number of columns in the output. **-1** denotes using all available columns.
Output Format	• **Formatted** - Select this option if the Excel file's data is formatted and you want to keep it in the output. • **Raw Value** - Ignores formatted data and the original raw values are output.
Show Preview	Selecting this option enables and displays the preview based on the component's configuration.

Columns



Option	Description
Code Page	Identifies character encoding.
Column Name In File	Identifies the name of the column coming from the Excel file.
DataType	All data types default to **DT_WSTR** but can be edited for output to be integers, boolean, strings, etc.
Excel Column Index	Defines the numeric order of the column.
Include In Output	Selected by default, determines whether or not the column is included in the output.
Length	Defines the number of characters for string and wstring data types.
Output Column Name	Defines the name of the column in the output. Column names can be edited here.
Precision	Defines the number of digits for numeric data types.
Scale	Defines the number of digits after the decimal for numeric data types.

Option	Description
Refresh Excel Columns	Refreshes the column information if the Excel document is changed while the component is open.

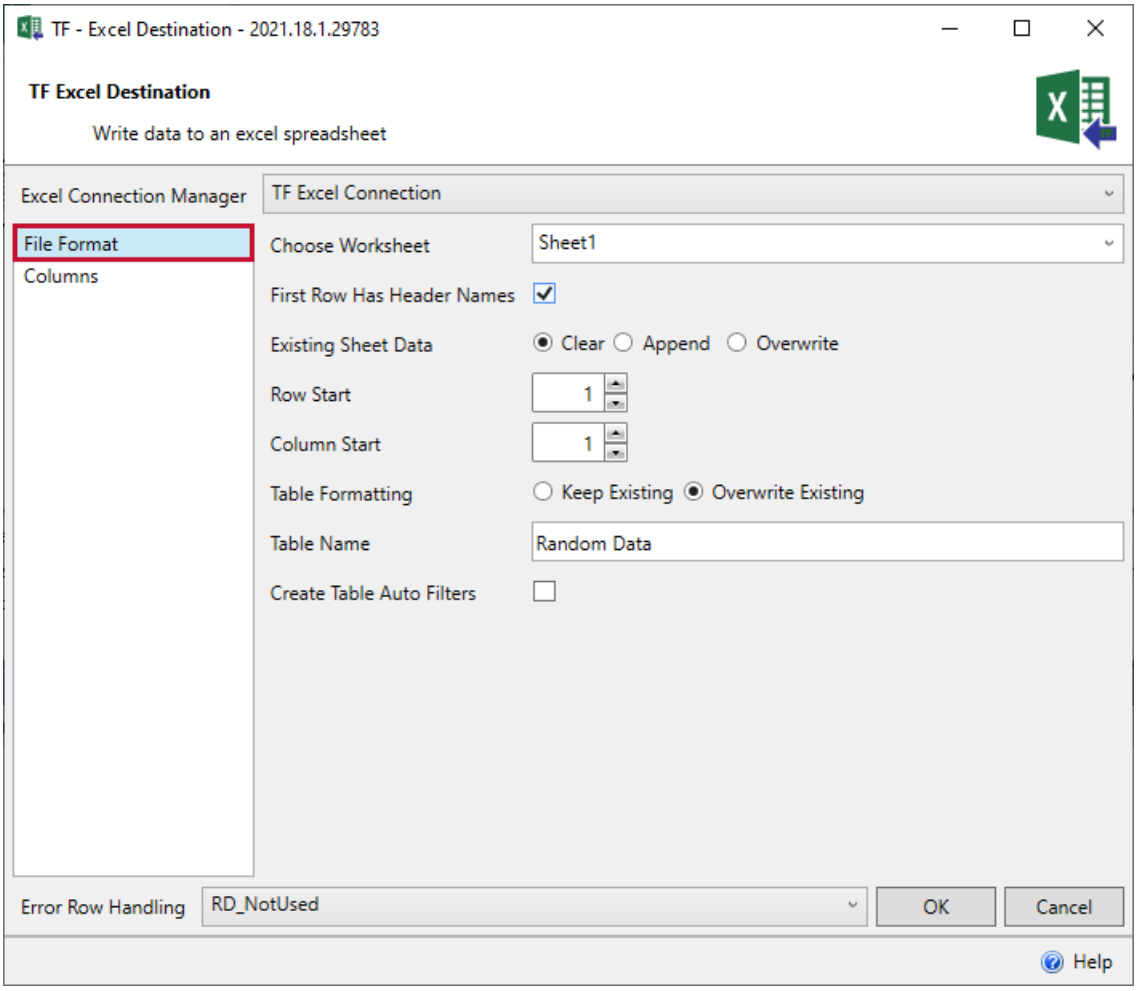
Additional Information: See the Task Factory Error Row Handling article for more information about this functionality.

Excel Destination

Important: Excel Destination is available for SQL versions 2012 and higher.

Destination Icon	Destination Description
	The **Excel Destination** connects to a new or existing Excel file and writes data to it.

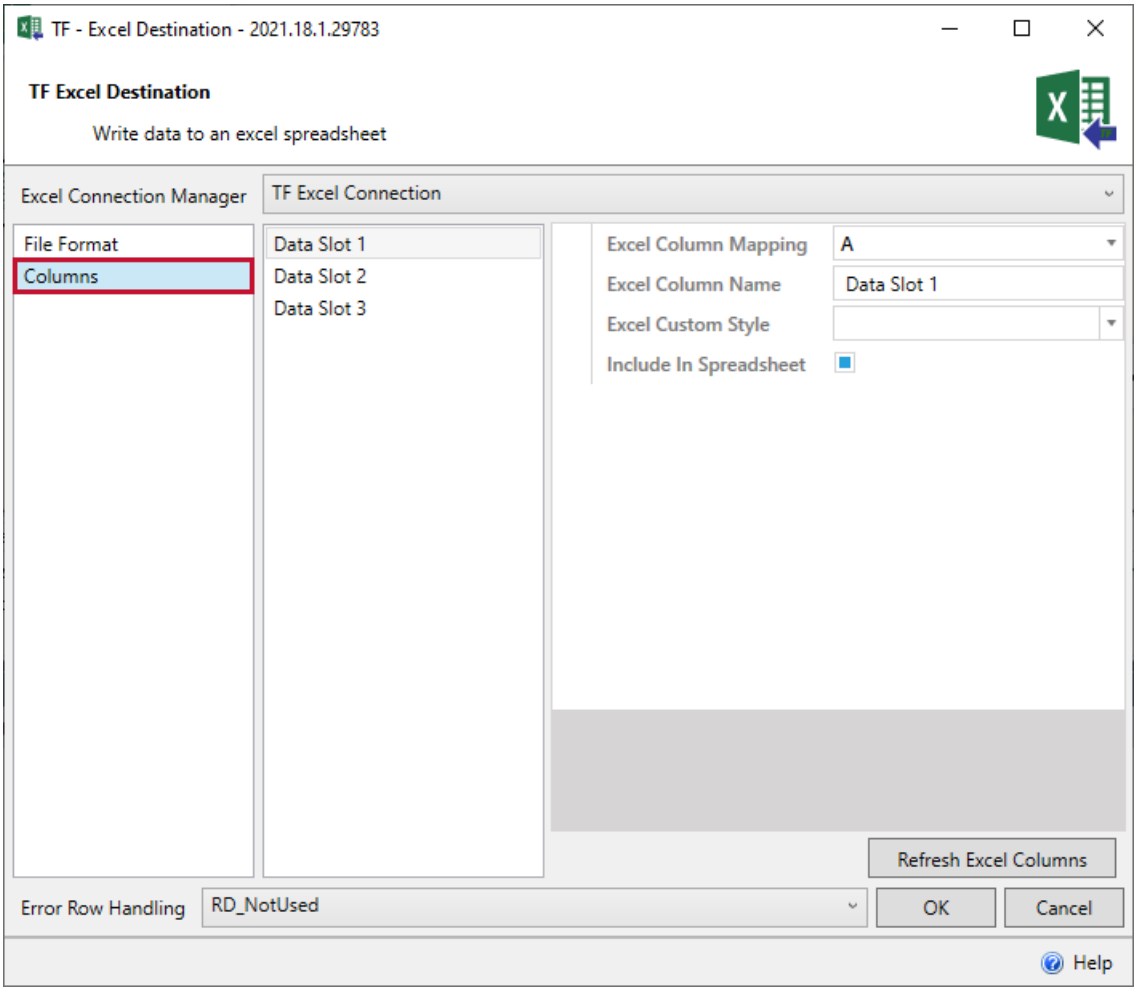
File Format



Option	Description
Excel Connection Manager	Create a new or select an existing connection to an Excel worksheet. For more information, see the Excel Connection Manager.
Choose Worksheet	Select which worksheet to load data to.
Existing Sheet Data	Choose from the following: • **Clear** - Clears all existing and residual data before writing to the spreadsheet. • **Append** - Adds data after pre-existing data in the spreadsheet. Previous data will remain in the file/worksheet. • **Overwrite** - Overwrites the existing data but keeps residual data. (Example: If 100 rows are inserted into a spreadsheet with 200 rows, rows 1-100 will be overwritten and rows 101-200 will remain.)
Row Start	Determines the row number to begin writing.
Column Start	Determines the column number to begin writing.
Create Table	Select this option to add table formatting.

Option	Description
Table Name	Defines the name of the table.
Create Table Auto Filters	Select to create auto filters.

Columns



Option	Description
Excel Column Mapping	Maps the column to the excel file's cell.
Excel Columns	Edits or renames a column that is written to the Excel file.
Include In Spreadsheet	This option is selected by default. Clear this option to prevent the column from being output to the Excel file.
Style	Columns can be formatted using styles within Excel. Please see the Excel Style page for more information.

Option	Description
Refresh Excel Columns	Refreshes the available columns that can be selected in the Excel Column Mapping property using data from the destination spreadsheet. If no columns currently exist, you will be shown the letter reference of the excel column (e.g. A, B, C, etc.)

Additional Information: See the Task Factory Error Row Handling article for more information about this functionality.

Excel Destination Column Styles

The most common and useful implementations of Excel format codes are:

Format Code	Description
General	General number format.
#	Digit placeholder that represents optional digits and does not display extra zeros.
0	Digit placeholder that displays insignificant zeros.
?	Digit placeholder that leaves a space for insignificant zeros but doesn't display them.
@	Text placeholder.
. (period)	Decimal point.
, (comma)	Thousands separator. A comma that follows a digit placeholder scales the number by a thousand.
****	Displays the character that follows it.
" "	Display any text enclosed in double quotes.
%	Multiplies the numbers entered in a cell by 100 and displays the percentage sign.
/	Represents decimal numbers as fractions.
E	Scientific notation format.

Format Code	Description
_ (underscore)	Skips the width of the next character. It's commonly used in combination with parentheses to add left and right indents, _(and _) respectively.
* (asterisk)	Repeats the character that follows it until the width of the cell is filled. It's often used in combination with the space character to change alignment.
[]	Creates conditional formats.

The following characters can be included in Excel custom format codes without the use of backslash or quotation marks:

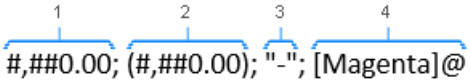
Symbol	Description
+ and -	Plus and minus signs
( )	Left and right parenthesis
:	Colon
^	Caret
'	Apostrophe
{ }	Curly brackets
< >	Less-than and greater than signs
=	Equal sign
/	Forward slash
!	Exclamation point
&	Ampersand
~	Tilde
	Space character

Excel Number Format

Excel number format consists of 4 sections of code separated by semicolons in the following order:

POSITIVE; NEGATIVE; ZERO; TEXT

Example:



In this example, positive numbers display two decimal places and a **thousands separator**. Negative numbers appear the same as positive numbers but enclosed in parenthesis. Zeros are replaced as dashes, and the text displays in the magenta color.

Digit and Text Placeholders

Code	Description	Example
0	Digit placeholder that displays zeros.	#.00 - displays two decimal places. A cell that contains 1.2 displays as 1.20.
#	Digit placeholder that represents optional digits (does not display extra zeros.)	#.## - displays up to two decimal places. 1.2 displays as 1.2 1.236 displays as 1.24
?	Digit placeholder that leaves a space for insignificant zeros on either side of the decimal point but doesn't display them. (Often used to align numbers in a column by decimal point.)	#.??? - displays a maximum of three decimal places and aligns numbers in the column by decimal point.
@	Text placeholder.	0.00; -0.00; 0; [Green]@ - applies the green font color to text values.

Rounding Numbers

Microsoft Excel separates thousands by commas if a comma is enclosed by any digit placeholders - pound sign (#), question mark (?) or zero (0). If no digit placeholder follows a comma, it scales the number by thousand, two consecutive commas scale the number by million, and so on.

Example:

General	#,###	#,	#.00,	#.00,,
50	50		.05	.00

General	#,###	#,	#.00,	#.00,,
500	500	1	.50	.00
5000	5,000	5	5.00	.01
55500	55,500	56	55.50	.06
555500	555,500	556	555.50	.56

Text and Spacing Formatting

To display both text and numbers in a cell, do the following:

To add a single character, precede that character with a backslash (\).

To add a text string, enclose it in double quotation marks (" ").

Example: Numbers that are rounded by thousands and millions - add \K and \M to the format codes, respectively:

To display thousands: #.00,\K

To display millions: #.00,,\M

General	#.00, \K	#,###.0, \K	#.00,, \M	#,###.0000,, \M
50	.05 K	.1 K	.00 M	.0001 M
500	.50 K	.5 K	.00 M	.0005 M
55500	55.50 K	55.5 K	.06 M	.0555 M
5555000	5555.00 K	5,555.0 K	5.56 M	5.5550 M
5555000000	5555000 K	5,555,000.0 K	5555.00 M	5,555.0000 M

Temperatures

- #"°F" displays Fahrenheit (Example: 85°F).
- #"°C" displays Celsius (Example: 23°C).

Display Leading Zeros

- Adds the number zero (0) as a placeholder to indicate the number of leading zeros.
- Example: Adding 00000 to the style displays the number 55 as 00055.

Percentages

- Adds the percent sign (%) to the style format box.
- Example: #% displays the number 0.08 as 8%.

Fractions

- For decimal numbers to appear as fractions, add the forward slash (/) to the style format box.
- Example: #/# displays the number .5 as 1/2.

Scientific Notation

- To display numbers as scientific notation, add the capital letter E to the style format box.
- Example: 00E+00 - displays 1,500,500 as 1.50E+06.

Indents

- Some prefer a cell's contents not position against the cell's border. To add an indent, use the underscore (_).
- _(adds indent from the left border.
- _) adds indent from the right border.

Font Color

- Adding the color surrounded by square brackets ([ ]) will format the font to the specified color. (: the color code must be the first item in the section. Also, only the following colors can be used - Black, Green, White, Blue, Magenta, Yellow, Cyan, and Red.)
- Example: [Blue] $#,##0.00 displays the number 25.3 as $25.30.

Repeat Characters

To repeat a specific character so that it fills the column width, use the asterisk (*) before the character to be repeated.

Example: *- makes the cell with the number 555 appear as |----------555| .

Alignment

- To align numbers left in a cell, type an asterisk and a space after the number code. Example: "#,###* " (Exclude the double quotes when adding the format.)
- Custom Number Formats Based on Conditions.
- To apply conditions that must be met for a custom style, enclose the condition within square brackets.
- Example: [Red] [<100] ; [Blue] [>=100] displays numbers as 49 and 184.

Custom Date

Format	Description	Example (January 1, 2018)

Format	Description	Example (January 1, 2018)
m	Month number without a leading zero	1
mm	Month number with a leading zero	01
mmm	Month name in short form	Jan
mmmm	Month name in full form	January
mmmmm	Month as the first letter	J
d	Day number without a leading zero	1
dd	Day number with a leading zero	01
ddd	Day of the week in short form	Mon
dddd	Day of the week in full form	Monday
yy	Last 2 digits of the year	18
yyyy	All 4 digits of the year	2018

Custom Time

Format	Description	Example (January 1, 2018)
h	Hours without a leading zero	0-23
hh	Hours with a leading zero	00-23
m	Minutes without a leading zero	0-59
mm	Minutes with a leading zero	00-59
s	Seconds without a leading zero	0-59
ss	Seconds with a leading zero	00-59
AM/PM	Periods of the day	AM or PM

Additional Information: For more Excel Style information, please refer to this blog post here.

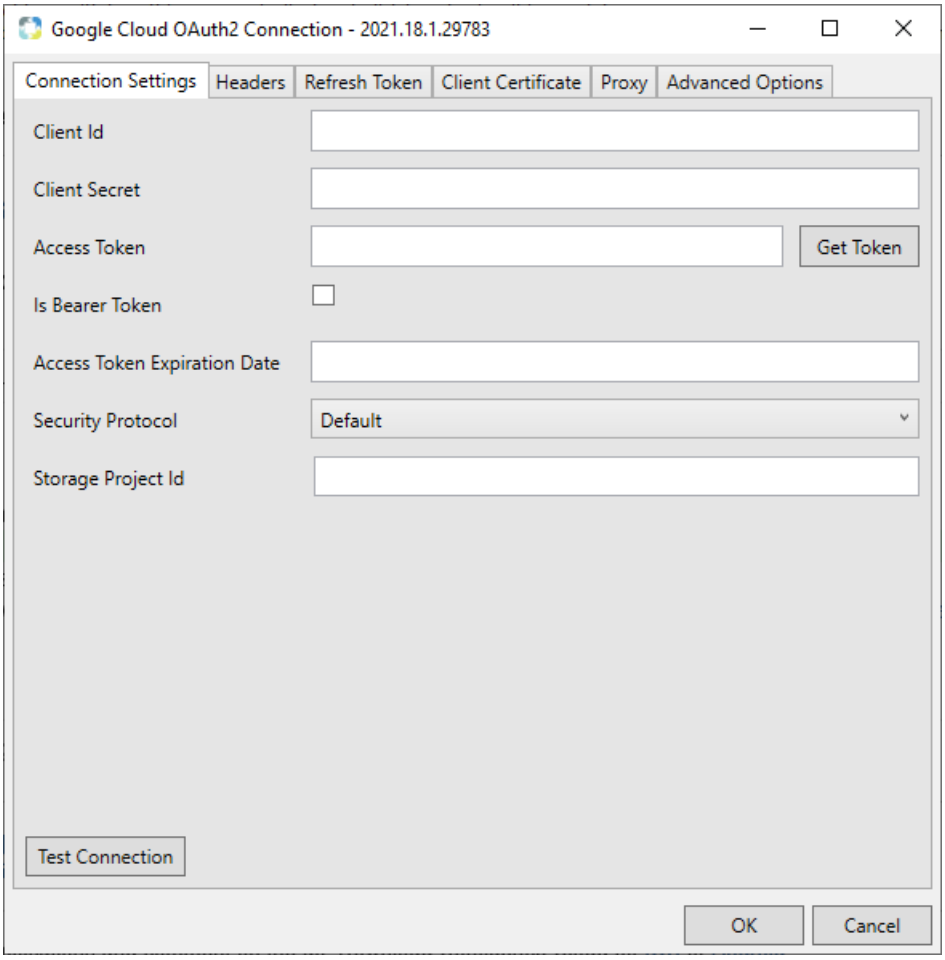
Task Factory Google

Last Modified on 03 February 2022

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Google Cloud OAuth2 Connection Manager

This connection manager is used with the **Google Cloud Source** and **Google Cloud Rest Destination**. Before connecting, users must obtain a **Client ID**, **Client Secret**, a **Storage Project Id**, and a **Redirect URL** from Google's APIs & Services Credentials dashboard.



Connection Settings

Option	Description
Client Id	The Client ID is found in Google's APIs & Services - Credentials console.
Client Secret	The Client Secret is found in Google's APIs & Services - Credentials console.

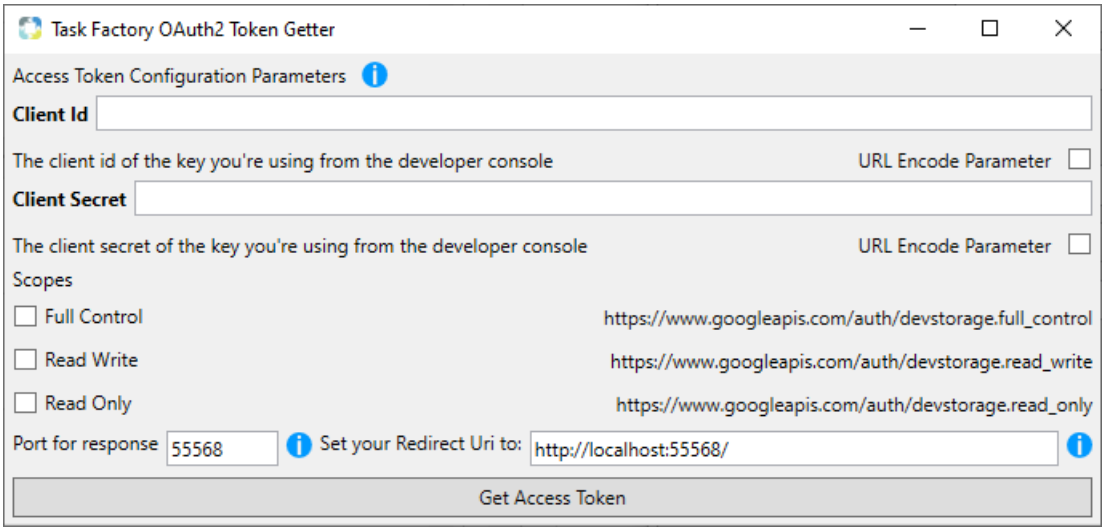
<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>Access Token</td><td>Select Get Token to open the Token Getter window.</td></tr>
<tr><td>Is Bearer Token?</td><td>Select this option if the Access Token is a bearer token.</td></tr>
<tr><td>Access Token Expiration Date</td><td>The expiration date of the access token.

ℹ Note: This settings is optional.</td></tr>
<tr><td>Security Protocol</td><td>The Security Protocol used by the API. The Default option uses the TLS version used by your .NET version by default. See the following chart for more information:
<table>
<tr><th>.NET Version</th><th>TLS Support Information</th></tr>
<tr><td>.NET 4.6 and above</td><td>Supports TLS 1.2 by default.</td></tr>
<tr><td>.NET 4.5</td><td>TLS 1.2 is supported, but you need to opt-in to use it. Execute the following code before making a connection to a secured resource to make TLS 1.2 the default:

• ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12</td></tr>
<tr><td>.NET 4.0</td><td>TLS 1.2 is not supported. If you have .NET 4.5 (or above) installed on the system, you can opt in for TLS 1.2 even if your application framework doesn't support it. SecurityProtocolType in .NET 4.0 doesn't have an entry for TLS1.2, so you need to use a numerical representation of this enum value:

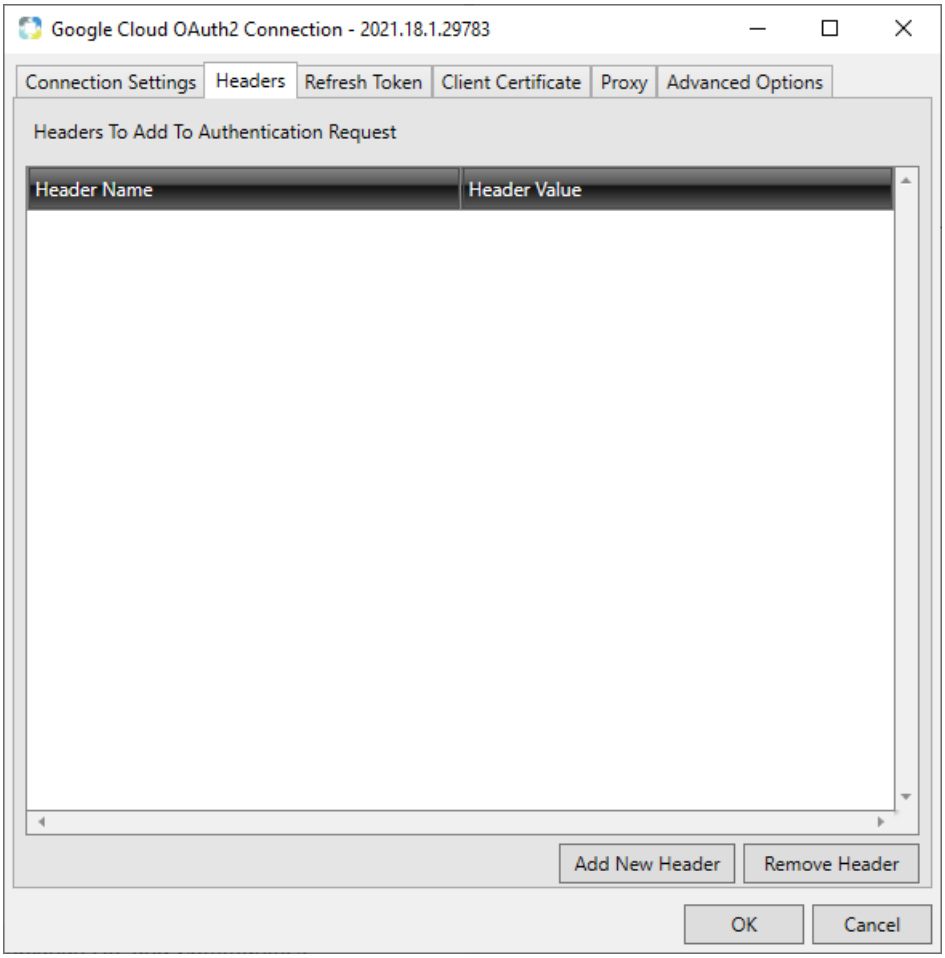
• ServicePointManager.SecurityProtocol = (SecurityProtocolType)3072;</td></tr>
<tr><td>.NET 3.5 or below</td><td>TLS 1.2 is not supported, and there is no workaround. Upgrade your application to a more recent version of the framework.</td></tr>
</table></td></tr>
<tr><td>Storage Project Id</td><td>The storage project Id for your connection.</td></tr>
</table>

Token Getter



Option	Description
Client Id	This field should be auto-completed from the previous window with your Client ID.
Client Secret	This field should be auto-completed from the previous window with your Client Secret.
Scopes	Selects which permission to be used in the call: • **Full Control** • **Read Write** • **Read Only** **Additional Information:** For more information about Scopes, see Google's Cloud Storage Authentication.
Port for response	The port on your local machine where you want to send the token request response.
Set your Redirect Uri to	Use this as your Redirect Uri in your OAuth2 App settings.
Get Access Token	Opens a browser window to login to your Google account and grant access.

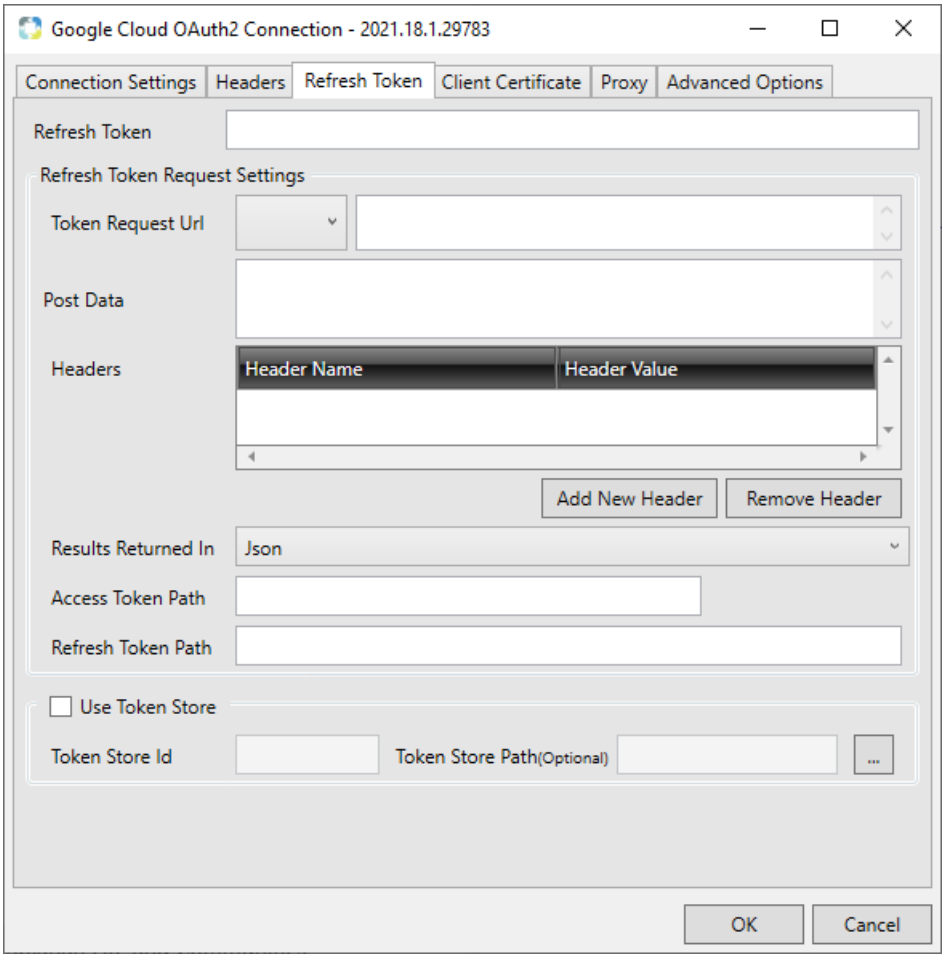
Headers



You can create header names and values by entering the information in their corresponding windows.

Option	Description
Add New Header	Select to add a new header name and value. Once added, you can select within the column and configure the name and the value.
Remove Header	Remove the selected header to be included in the component's output.

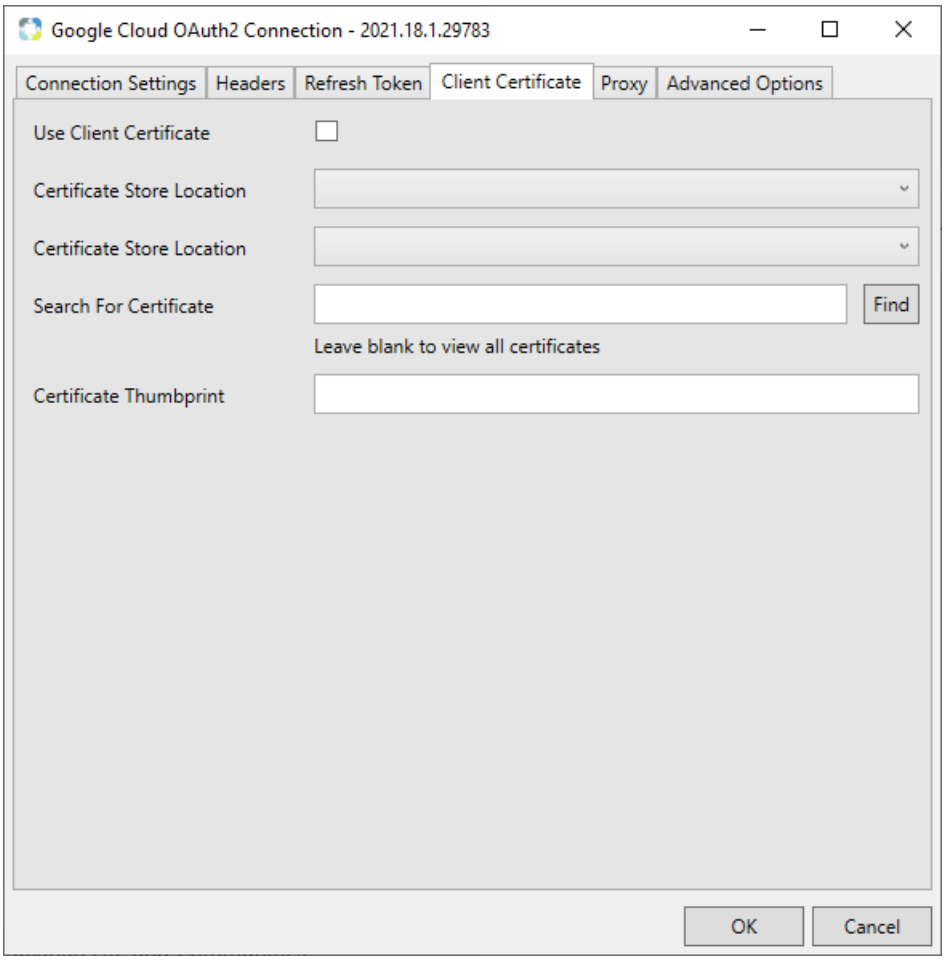
Refresh Token



Option	Description
Refresh Token	Authentication information provided to you at the application's developer site. **Note:** This should match the API Key from the Connection Settings window.
Token Request Url	The URL that returns a refresh token. Example: **https://api.citrixonline.com/oauth/access_token**
Headers	Allows you to add headers within the refresh token. Select **Add New Header** to add a header. Select **Remove header** to remove a header.
Results Returned In	Specifies how results are returned from the application (JSON, XML, or String.)
Access Token Path	The token path for the access token. Example: **access_token**
Refresh Token Path	The token path for the refresh token. Example: **refresh_token**
Use Token Store	This option creates a file on your machine that saves the access token, and refresh token in an encrypted format.

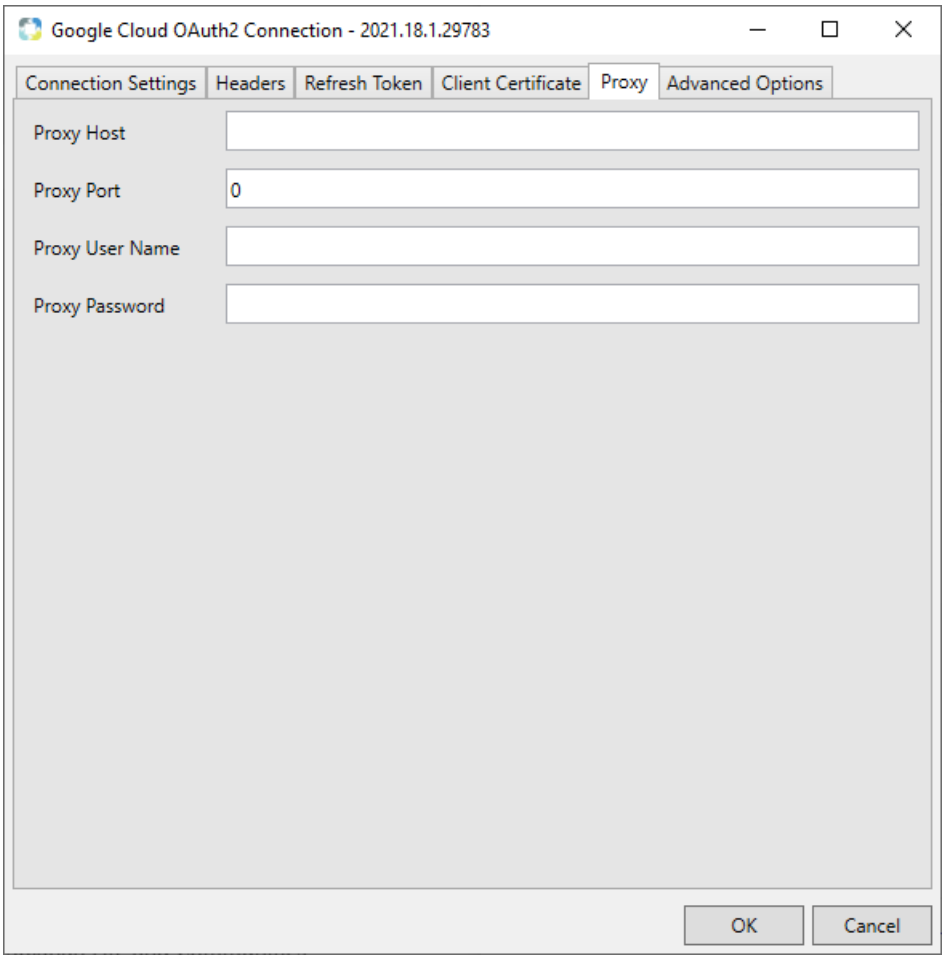
Option	Description
Token Store Id	Identifies the Token Store Id.
Token Store Path (Optional)	Identifies the Token Store path.

Client Certificate



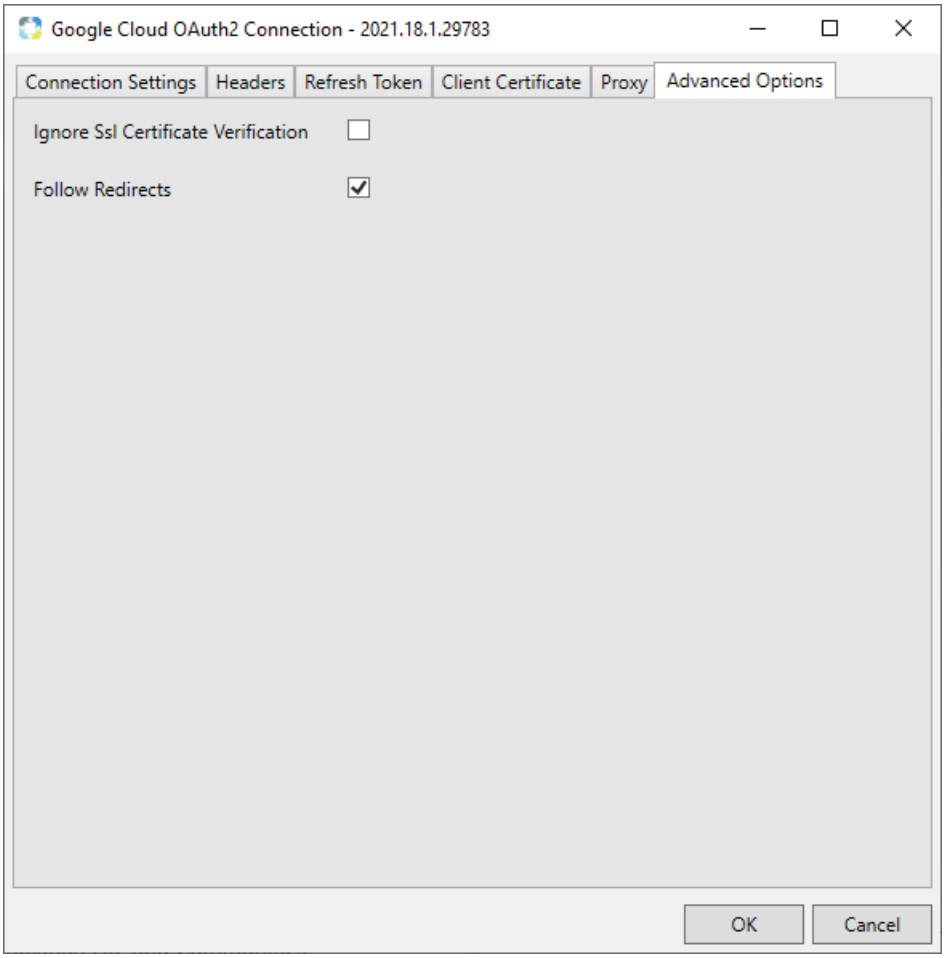
Option	Description
Use Client Certificate	Select this option if you want to use the client certificate.
Certificate Store Location	The client store location.
Search For Certificate	Enter a certificate that you want to search for and then select **Find**. **Note:** Leave this section blank and select **Find** to view all certificates.
Certificate Thumbprint	The identifier thumbprint of the selected certificate.

Proxy



Option	Description
Proxy Host	When connecting to an endpoint using a proxy, you should enter the proxy URL in this field.
Proxy Port	The port number that corresponds to the URL proxy host.
Proxy User Name	The username needed to authenticate to the proxy.
Proxy Password	The password that allows you to authenticate to the proxy.

Advanced Options



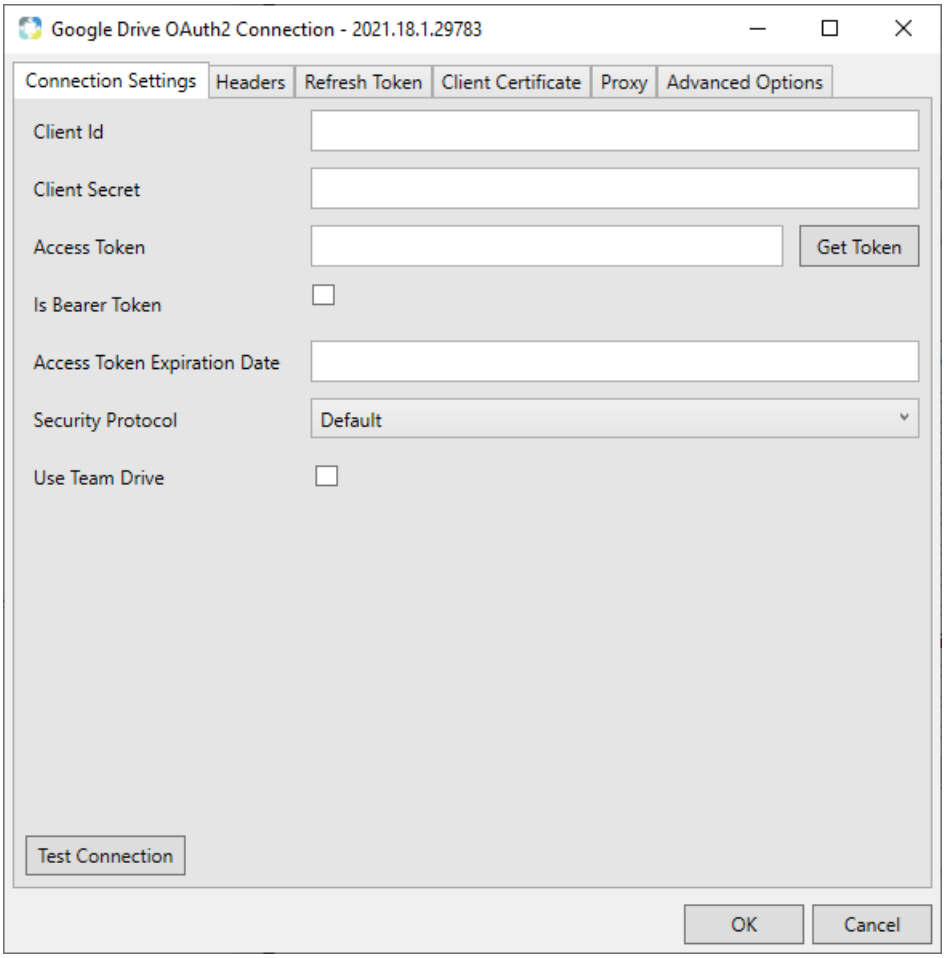
Option	Description
Ignore Ssl Certificate Verification	Selecting this option ignores the SSL Certificate notification from the API during the connection. **Note:** This option applies to APIs that require you to have an SSL Certificate. Select this option if the SSL Certificate is not needed.
Follow Redirects	Selecting this option implements a 301 redirect on configured endpoints. For example, this option would automatically send your request to an HTTPS redirect once the request has reached the http version of the site.

Google Drive Oauth2 Connection Manager

Used with the Google Drive Source and Google Drive Destination. Before connecting, users must obtain a **Client ID**, **Client Secret**, and **Redirect URL** from Google's APIs & Services Credentials dashboard.

Connection Settings

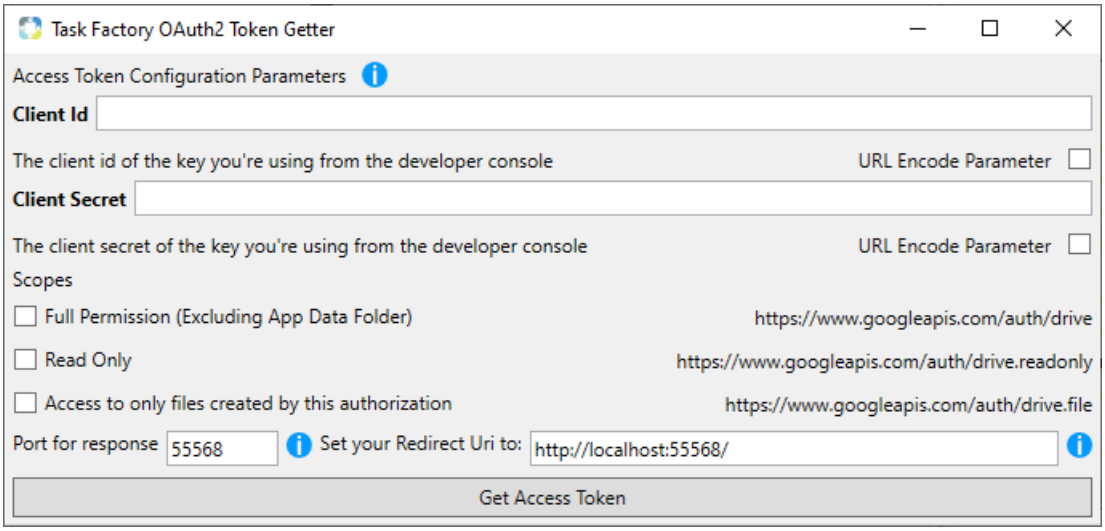
To connect to Google Drive, complete the following:



Option	Description
Client Id	The **Client ID** is found in Google's APIs & Services - Credentials console.
Client Secret	The **Client Secret** is found in Google's APIs & Services - Credentials console.
Access Token	Select **Get Token** to open the **Token Getter** window.
Is Bearer Token?	Select this option if the Access Token is a bearer token.
Access Token Expiration Date	The expiration date of the access token. **Note:** This settings is optional.

<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>Security Protocol</td><td>The Security Protocol used by the API. The Default option uses the TLS version used by your .NET version by default. See the following chart for more information:
<table>
<tr><th>.NET Version</th><th>TLS Support Information</th></tr>
<tr><td>.NET 4.6 and above</td><td>Supports TLS 1.2 by default.</td></tr>
<tr><td>.NET 4.5</td><td>TLS 1.2 is supported, but you need to opt-in to use it. Execute the following code before making a connection to a secured resource to make TLS 1.2 the default:
ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12</td></tr>
<tr><td>.NET 4.0</td><td>TLS 1.2 is not supported. If you have .NET 4.5 (or above) installed on the system, you can opt in for TLS 1.2 even if your application framework doesn’t support it. SecurityProtocolType in .NET 4.0 doesn’t have an entry for TLS1.2, so you need to use a numerical representation of this enum value:
ServicePointManager.SecurityProtocol = (SecurityProtocolType)3072;</td></tr>
<tr><td>.NET 3.5 or below</td><td>TLS 1.2 is not supported, and there is no workaround. Upgrade your application to a more recent version of the framework.</td></tr>
</table></td></tr>
<tr><td>Use Team Drive</td><td>Select this option if your application uses a Team Drive to share documents with other users.</td></tr>
</table>

Token Getter

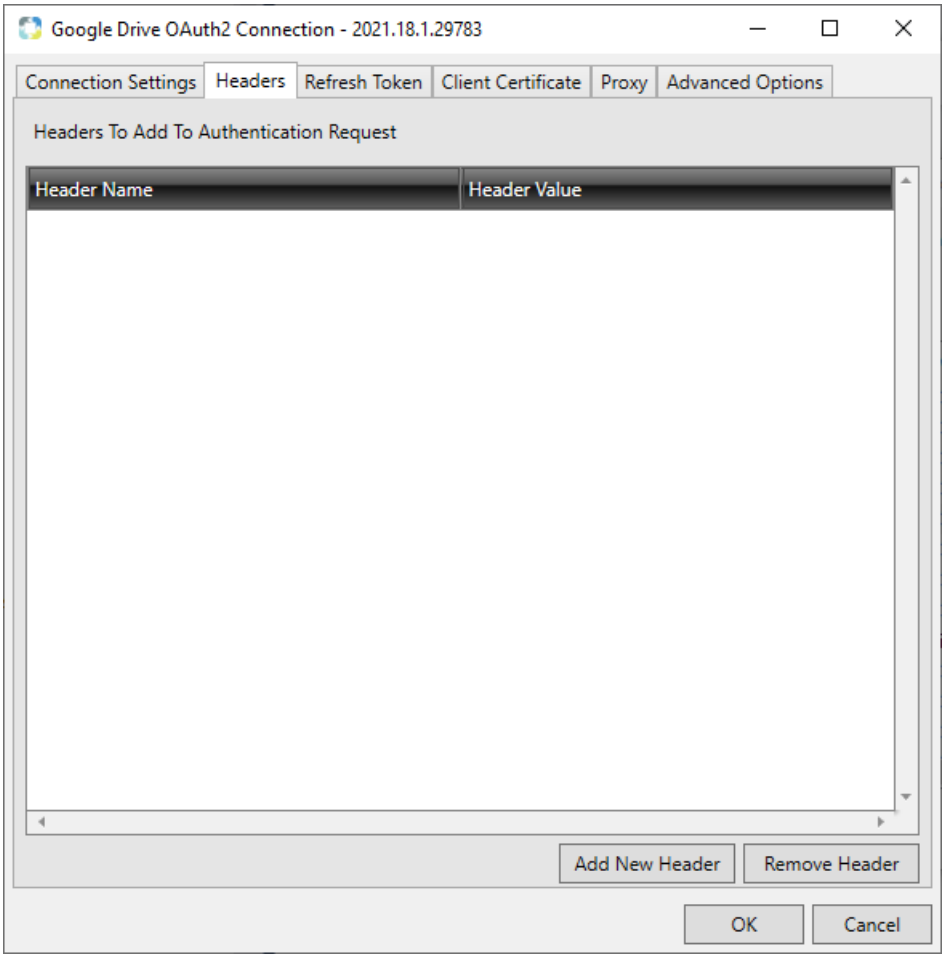


Option	Description
Client Id	This field should be auto-completed from the previous window with your Client ID.
Client Secret	This field should be auto-completed from the previous window with your Client Secret.
Scopes	Selects which permission to be used in the call: • **Full Control** • **Read Write** • **Read Only** **Additional Information:** For more information about Scopes, see Google's Cloud Storage Authentication.
Port for response	The port on your local machine where you want to send the token request response.
Set your Redirect Uri to	Use this as your Redirect Uri in your OAuth2 App settings.
Get Access Token	Opens a browser window to login to your Google account and grant access.

The Token Getter window automatically closes and the rest of the connection manager automatically completes. A browser window opens to your google account. Log in to your Google Drive application's account and select **Allow** to grant access.

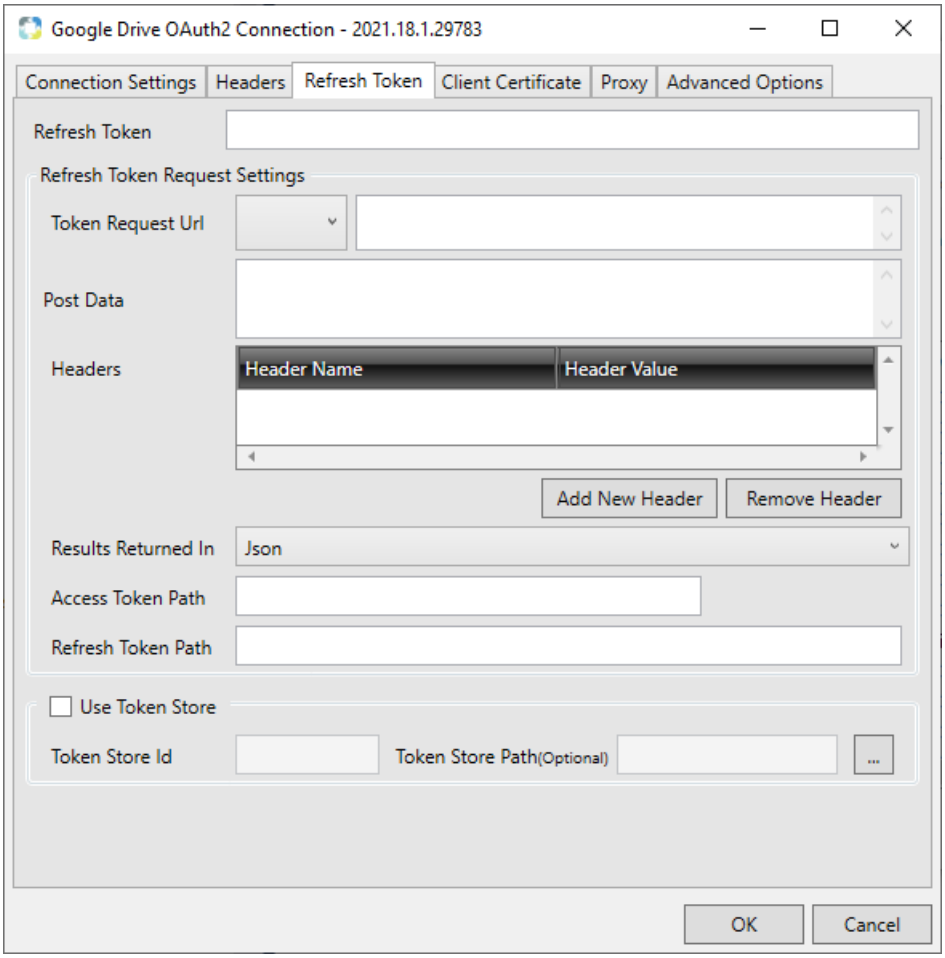
Headers

You can create header names and values by entering the information in their corresponding windows.



Option	Description
Add New Header	Select to add a new header name and value. Once added, you can select within the column and configure the name and the value.
Remove Header	Remove the selected header to be included in the component's output.

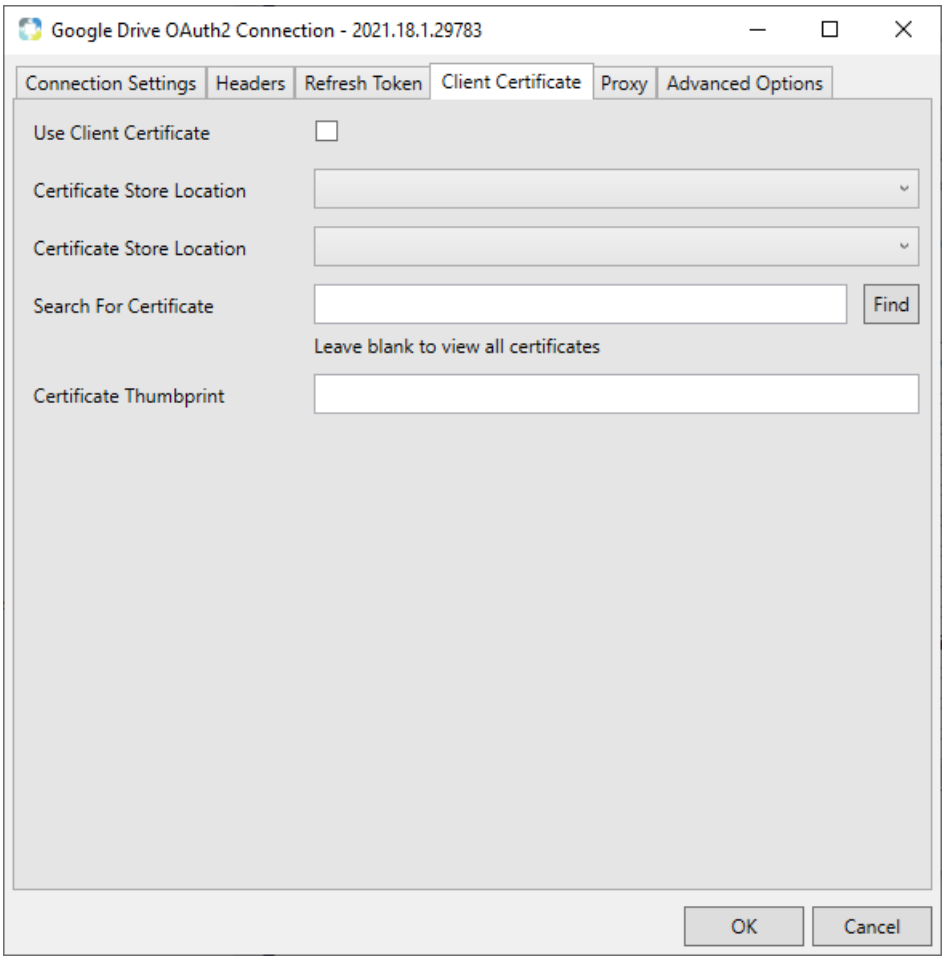
Refresh Token



Option	Description
Refresh Token	Authentication information provided to you at the application's developer site. **Note:** This should match the API Key from the Connection Settings window.
Token Request Url	The URL that returns a refresh token. Example: **https://api.citrixonline.com/oauth/access_token**
Headers	Allows you to add headers within the refresh token. Select **Add New Header** to add a header. Select **Remove header** to remove a header.
Results Returned In	Specifies how results are returned from the application (JSON, XML, or String.)
Access Token Path	The token path for the access token. Example: **access_token**
Refresh Token Path	The token path for the refresh token. Example: **refresh_token**
Use Token Store	This option creates a file on your machine that saves the access token, and refresh token in an encrypted format.

Option	Description
Token Store Id	Identifies the Token Store Id.
Token Store Path (Optional)	Identifies the Token Store path.

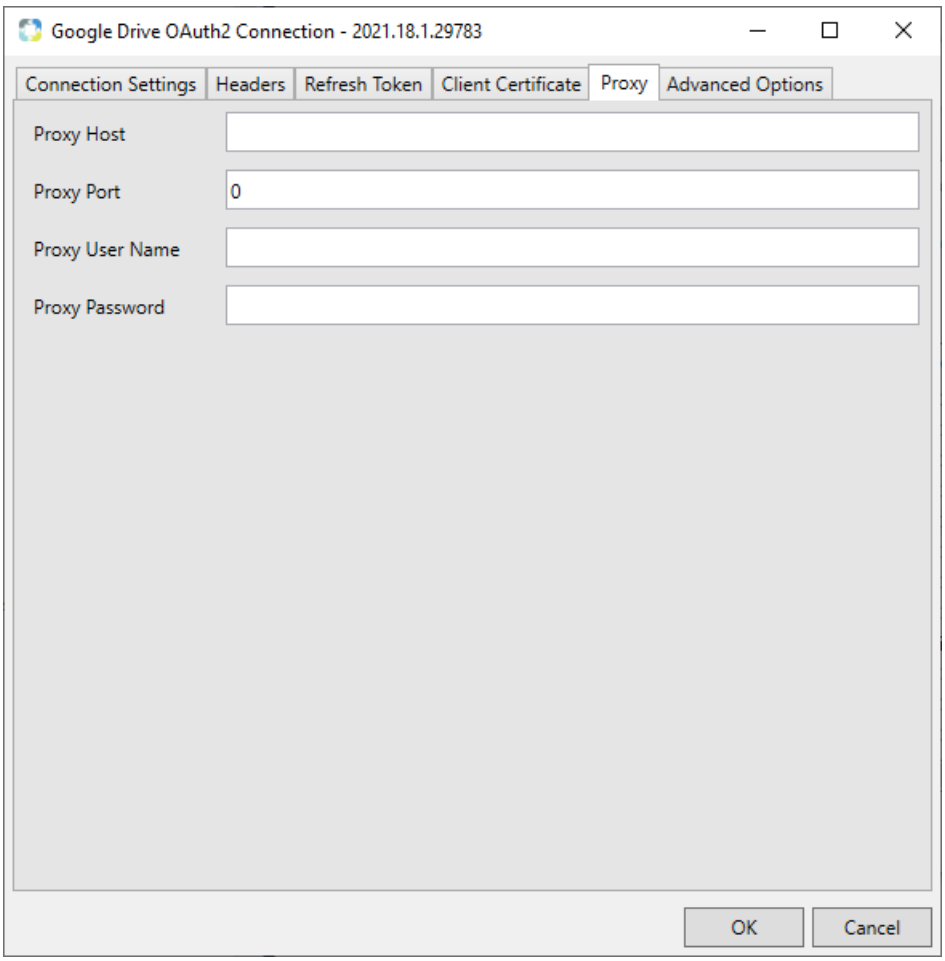
Client Certificate



Option	Description
Use Client Certificate	Select this option if you want to use the client certificate.
Certificate Store Location	The client store location.
Search For Certificate	Enter a certificate that you want to search for and then select **Find**. **Note:** Leave this section blank and select **Find** to view all certificates.
Certificate Thumbprint	The identifier thumbprint of the selected certificate.

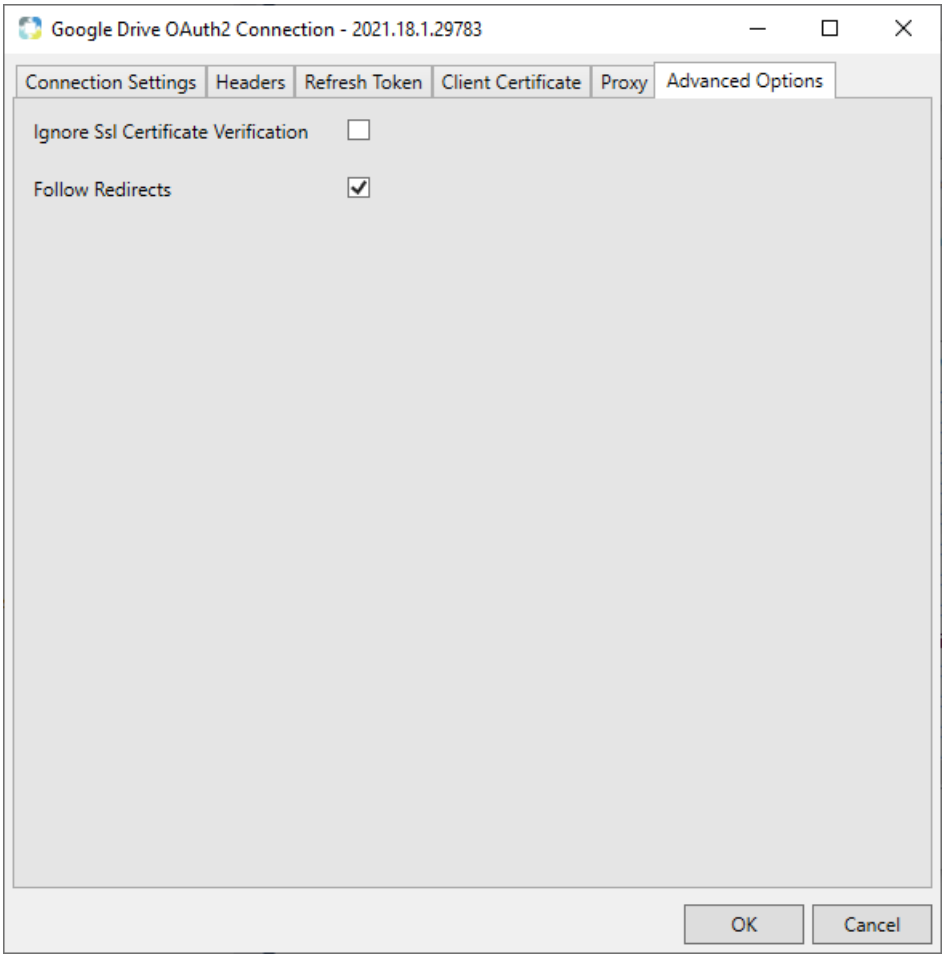
Proxy Configuration

Select the Proxy tab to connect to your Google Drive via a proxy connection.



Option	Description
Proxy Host	Identifies the proxy address.
Proxy Port	Identifies the port used by the proxy.
Proxy User Name	Enter the user name for proxies that require user authentication.
Proxy Password	Enter the password for proxies that require user authentication.

Advanced Options

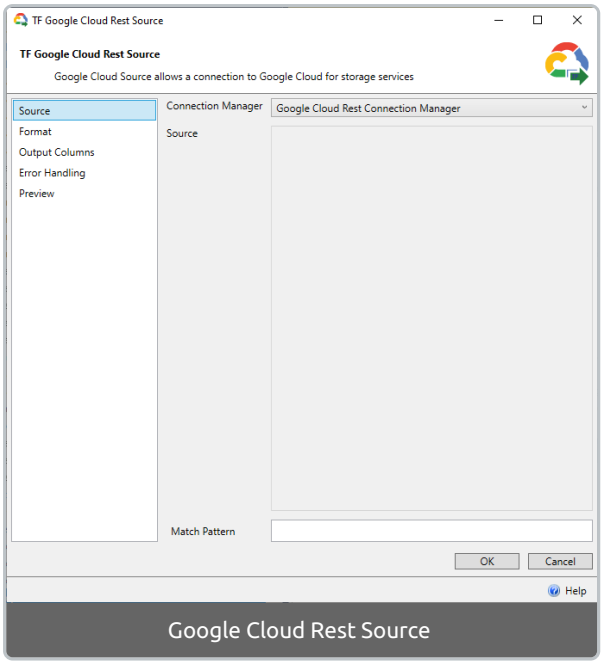


Option	Description
Ignore Ssl Certificate Verification	Selecting this option ignores the SSL Certificate notification from the API during the connection. **Note:** This option applies to APIs that require you to have an SSL Certificate. Select this option if the SSL Certificate is not needed.
Follow Redirects	Selecting this option implements a 301 redirect on configured endpoints. For example, this option would automatically send your request to an HTTPS redirect once the request has reached the http version of the site.

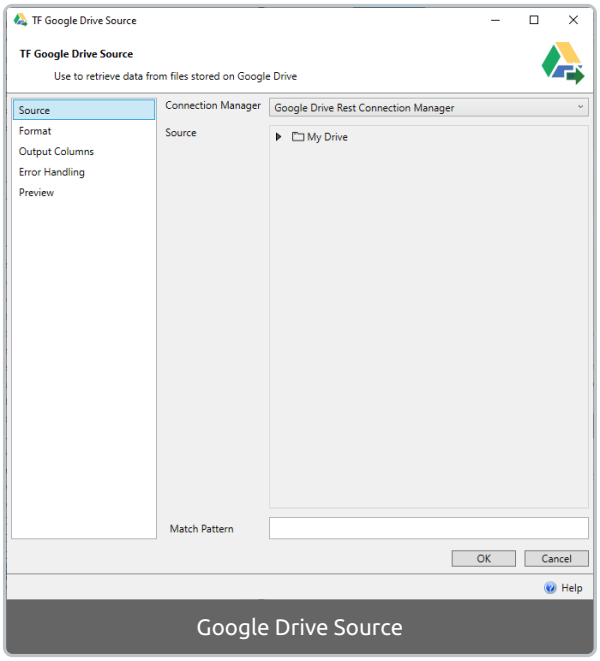
Google Cloud Rest Source and Google Drive Rest Source

Source

Begin by creating a connection manager that connects to the service's storage container.



Google Cloud Rest Source



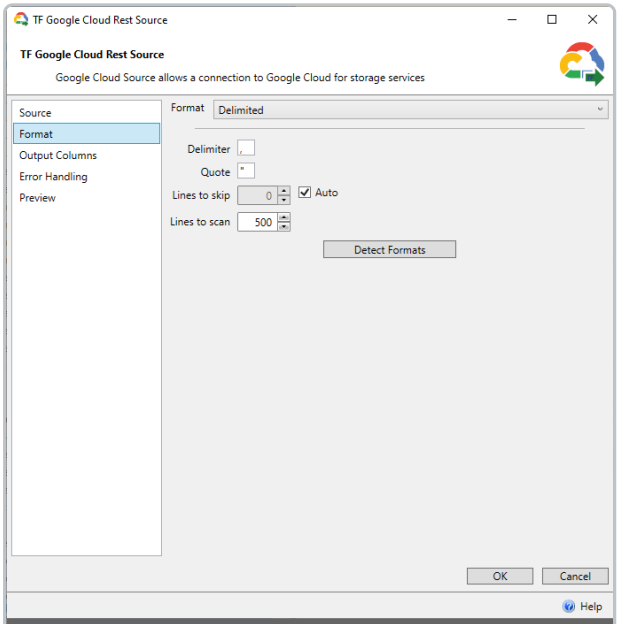
Google Drive Source

Format

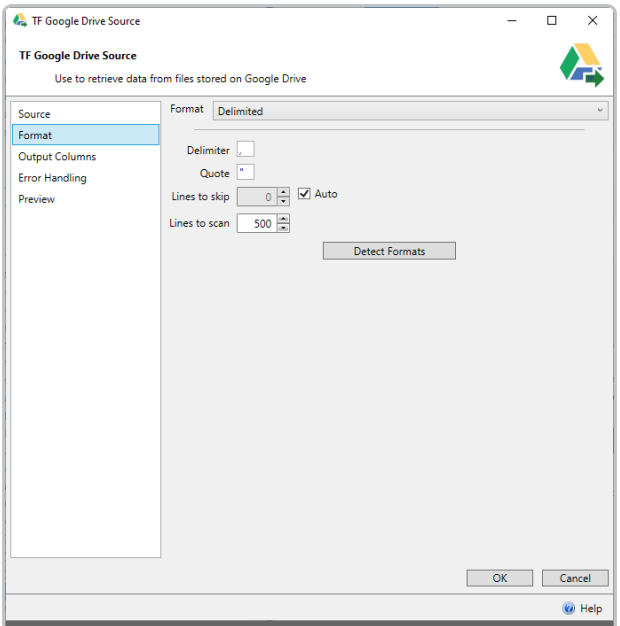
The following formats are available:

Delimited Format

Users can manually configure the delimiter , quote identifiers, the lines to skip, and lines to scan. Select **Detect Formats** to configure these fields automatically.



Google Cloud Rest Source Format



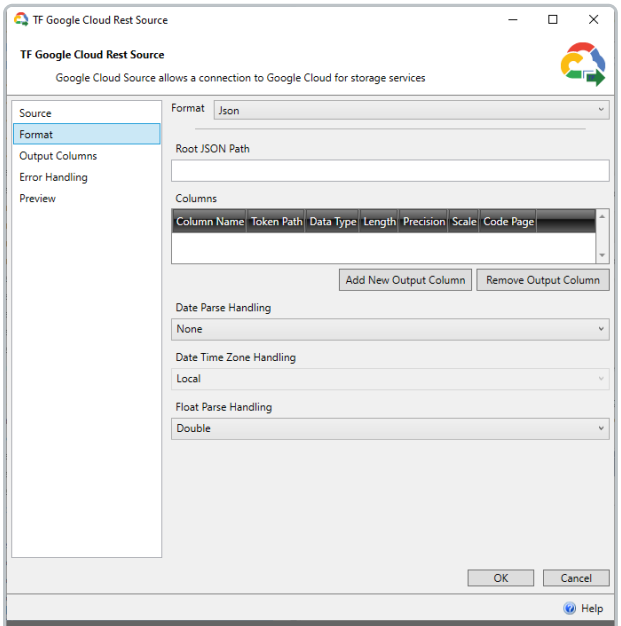
Google Drive Source Format

Option	Description
Delimiter	Identifies the character that separates columns. The default value is the comma (,).
Quote	Identifies the character used to indicate a value (if present).
Lines To Skip	Users can designate the number of lines to skip at the beginning of the delimited file.
Auto	Automatically detects which lines to skip.
Lines To Scan	Determines the number of lines to scan when detecting the file.

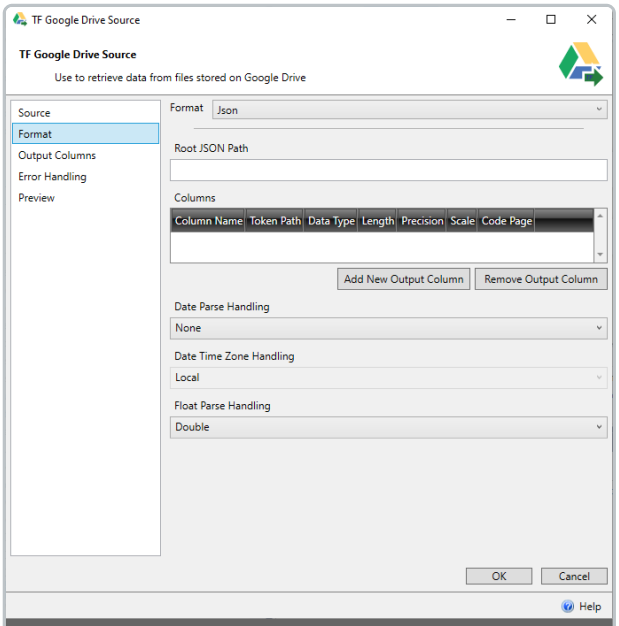
Option	Description
Detect Formats	Select this button to begin scanning the chosen file.

Detected Delimited View - After the format is detected, a new view appears that allows users to configure or change different properties of the file.

Json Format



Google Cloud Rest Source Format Json

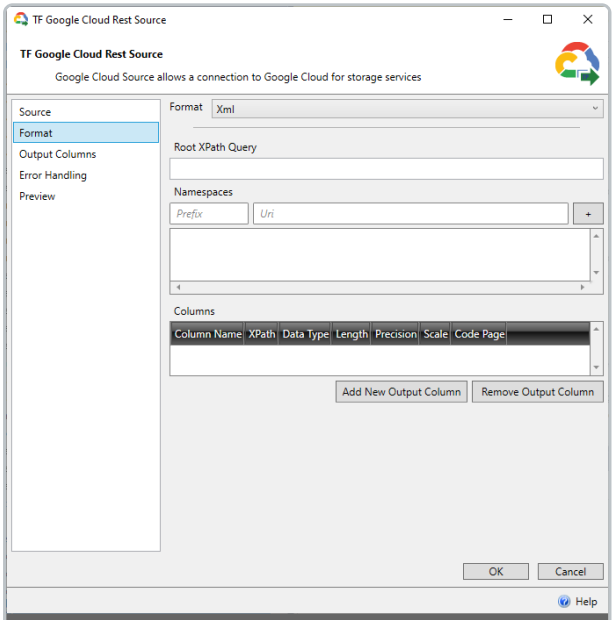


Google Drive Source Format Json

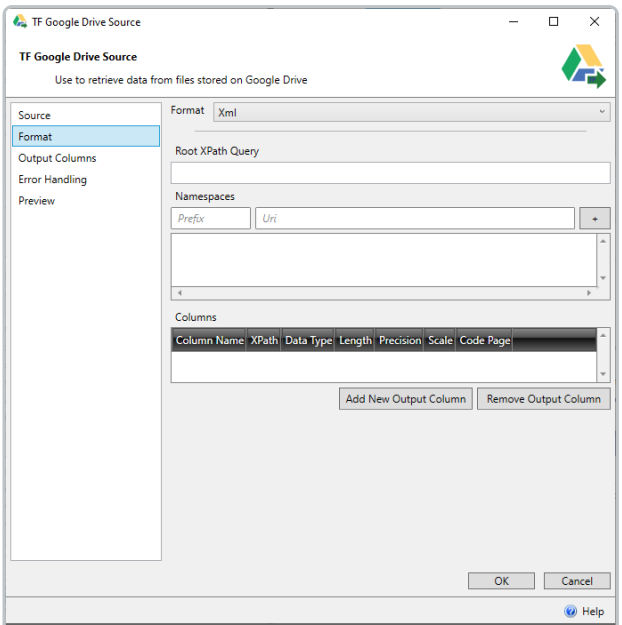
Option	Description
Root Json Path	JSON queries can return multiple levels, therefore, this field Identifies the root to be used.
Output Columns	In this window, users can add and remove columns, define their name, token path (defined within the raw data), data type, length, precision, scale, and code page.
Date Parse Handling	Specifies how date formatted strings are parsed when reading JSON text. • **None** - Strings are not parsed as a date type and are read as strings. • **DateTime** - Date formatted strings are parsed as DateTime. • **DateTime Offset** - Date formatted strings are parsed as DateTimeOffset.

Option	Description
Date Time Zone Handling	Specifies how to treat time values converted between string and DateTime. • **Local** - Treats the time as local (UTC times will be converted to local times.) • **Utc** - Treats the time as UTC. This will convert local time to UTC. • **Unspecified** - Treats the time as local time if a DateTime is being converted to a string. If a string is being converted to DateTime, convert to a local time if a time zone is specified. • **RoundtripKind** - Time zone is preserved when converting.
Float Parse Handling	Specifies how floating point numbers are parsed. • **Double** - Floating point numbers are parsed as a double datatype. • **Decimal** - Floating point numbers are parsed as a decimal datatype.

XML Format



Google Cloud Rest Source Format Xml



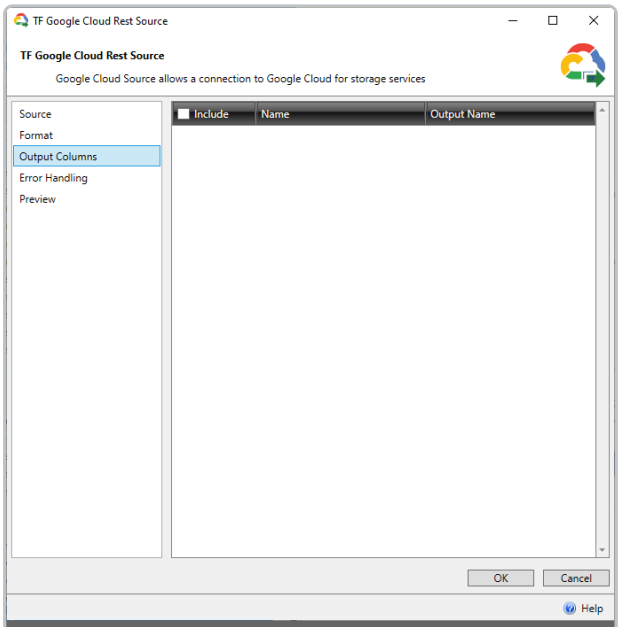
Google Drive Source Format Xml

Option	Description
Root XPath Query	XML queries can return multiple levels, therefore, this field Identifies the root to be used.
Namespaces:	• **Prefix** - Identifies the prefix used in an xml namespace. Example: In the element \ the prefix is url. • **Uri** - The url or address identified in the namespace.

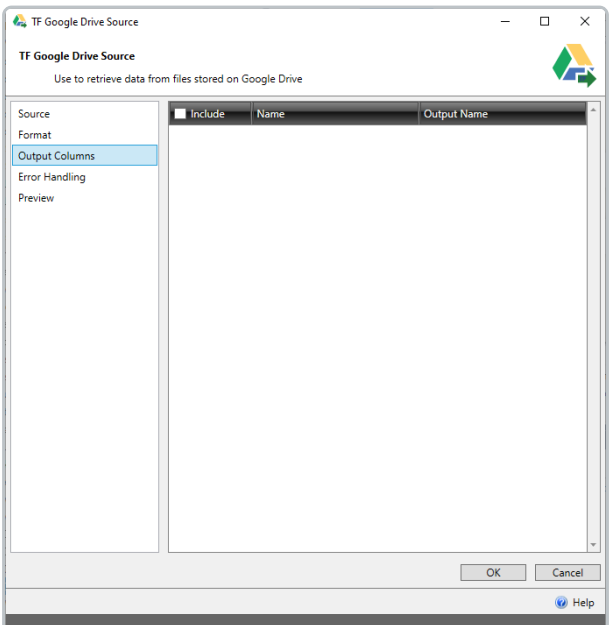
Option	Description
Columns	Users can create, remove, and configure the name, index (zero-based), data type, length, precision, and scale of the columns being extracted from the file.

Output Columns

Users can select which columns to include/exclude in the output. Additionally, users can rename the columns by selecting into its corresponding output name.

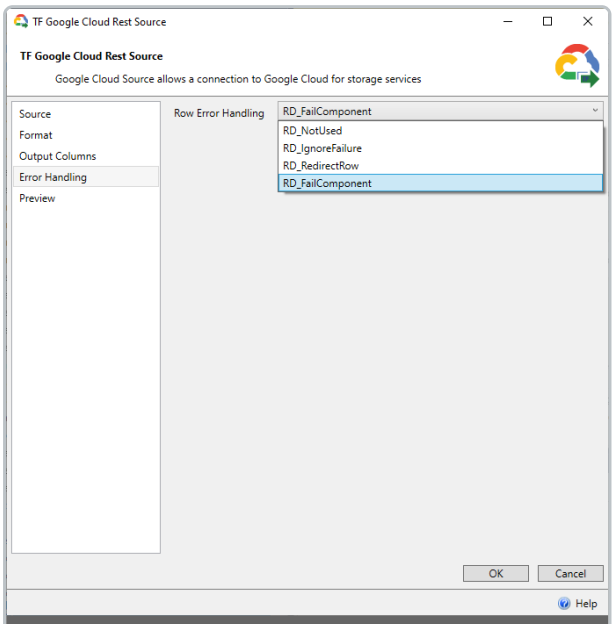


Google Cloud Rest Source Output Columns

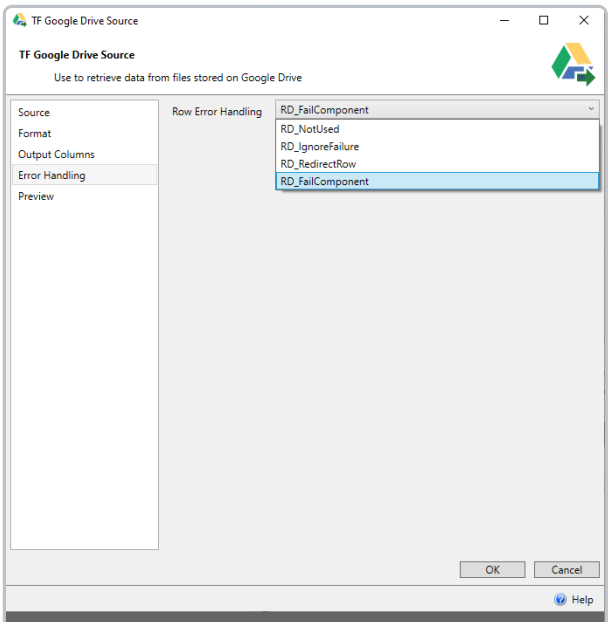


Google Drive Source Output Columns

Error Handling



Google Cloud Rest Source Error Handling

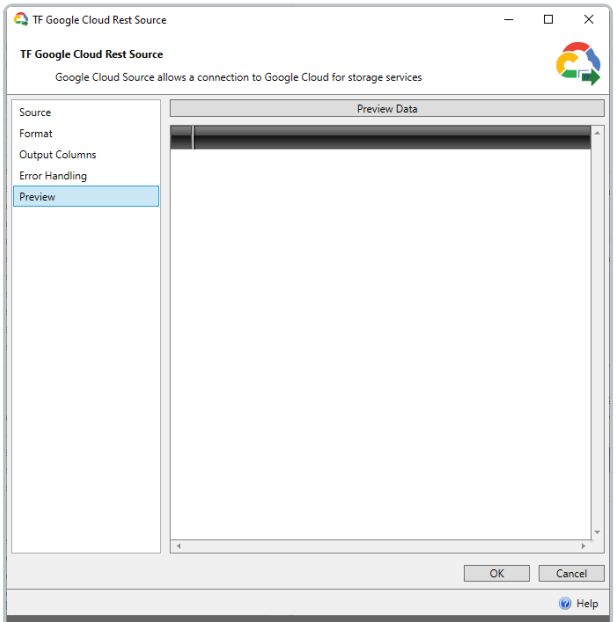


Google Drive Source Error Handling

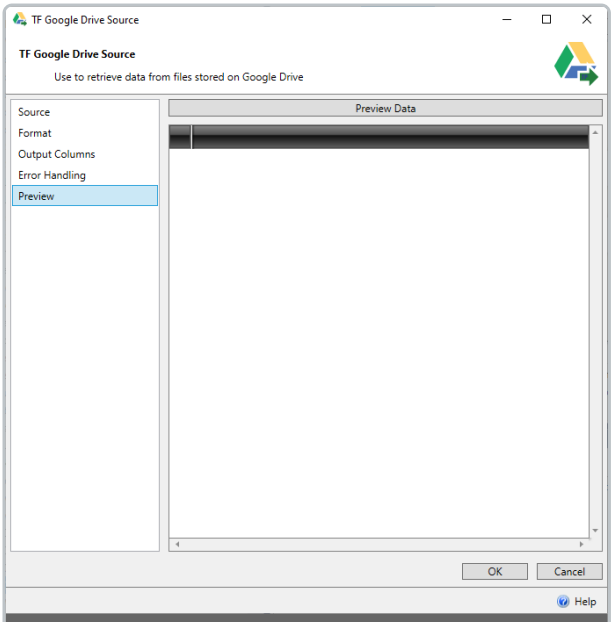
Option	Description
Not Used	Error handling is not used and no error constraint is available.
Ignore Failure	All errors are ignored and the package continues to execute.
Redirect Row	All error rows are directed to an error output.
Fail Component	(Default selection) On error, the component fails and the package execution stops.

Preview

Select **Preview Data** after configuring the previous tabs to view an output sample.



Google Cloud Rest Source Preview

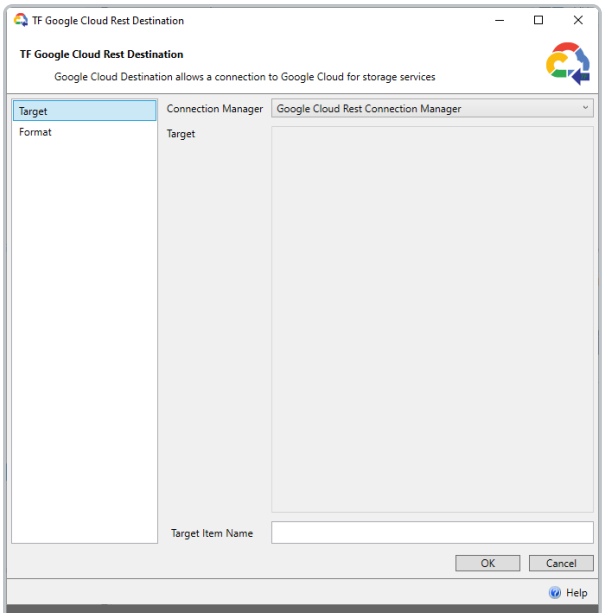


Google Drive Source Preview

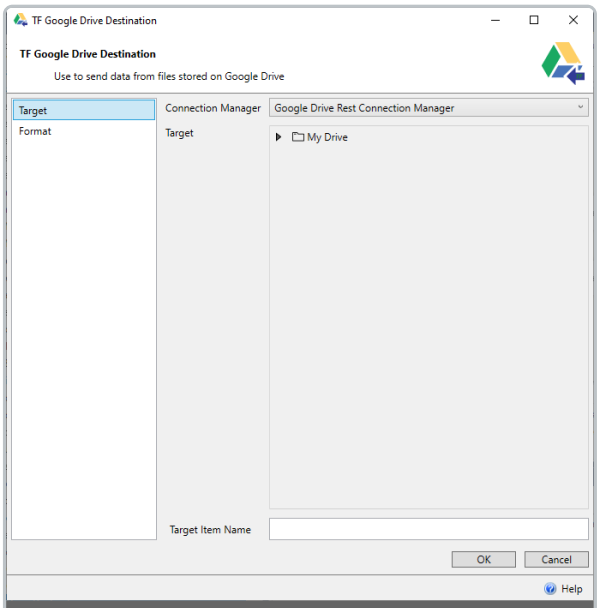
Google Cloud Destination and Google Drive Destination

Target

Begin by creating a connection manager that connects to a Google Storage container. After a connection manager is created, the source window populates with files and folders. Select the desired file to continue configuration.



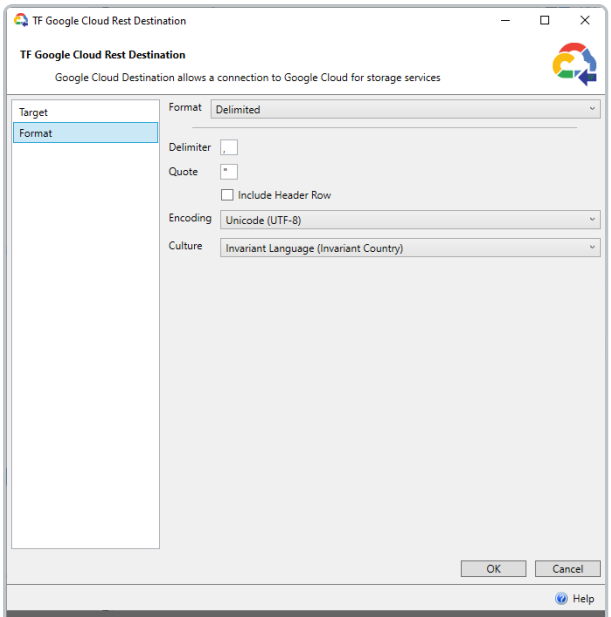
Google Cloud Rest Destination Target



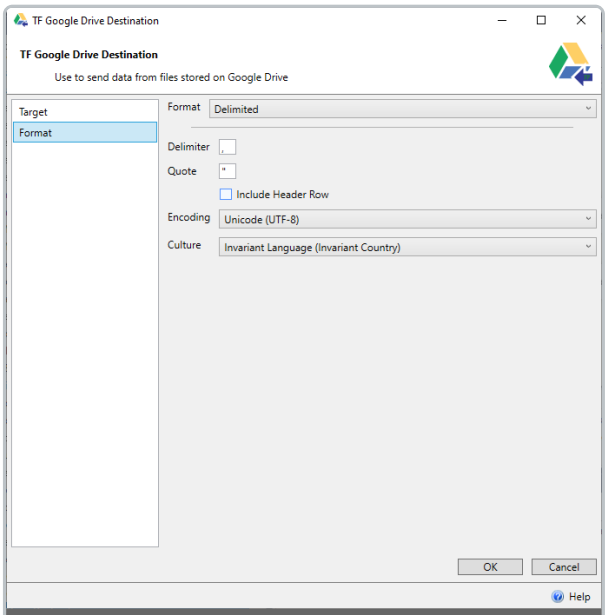
Google Drive Destination Target

Option	Description
Target Item Name	Defines the name and extension of the file you would like to create when the component is executed. (For example: **MyNewXmlList.xml**)

Delimited Format



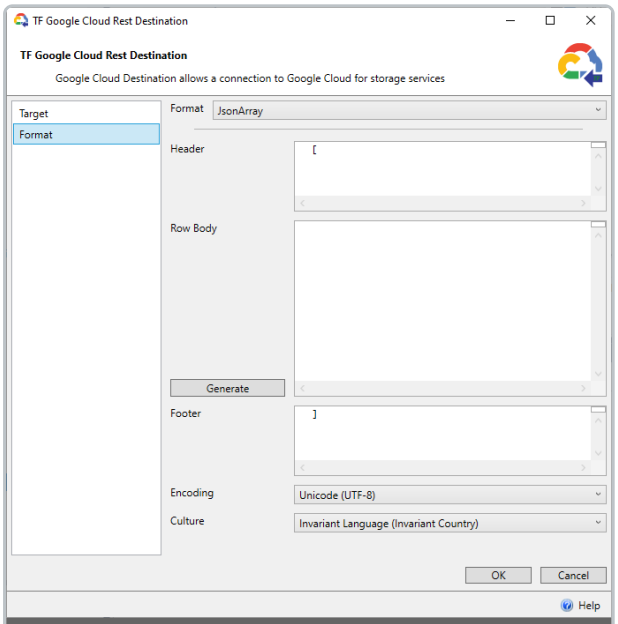
Google Cloud Rest Destination Format



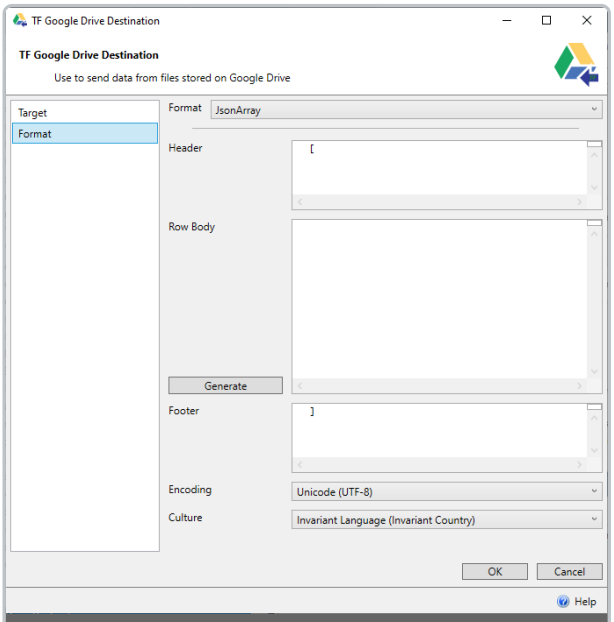
Google Drive Rest Destination Format

Option	Description
Delimiter	Identifies the character that separates text.
Quote	Identifies the character (single or double quote) that indicates value.
Include Header Row	Writes the column headers at the beginning of the document.
Encoding	Selects which encoding is used to create the document.
Culture	Used to identify the document's language culture.

Json Array Format



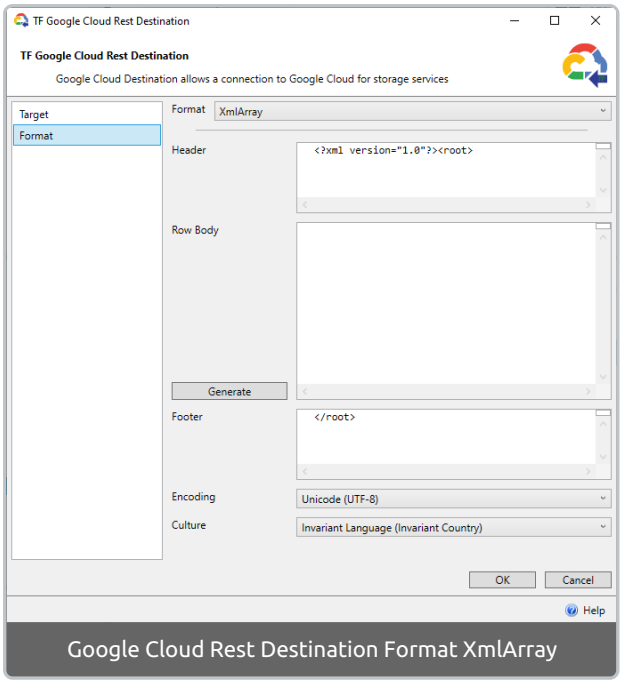
Google Cloud Rest Destinaton Fomrat JsonArray



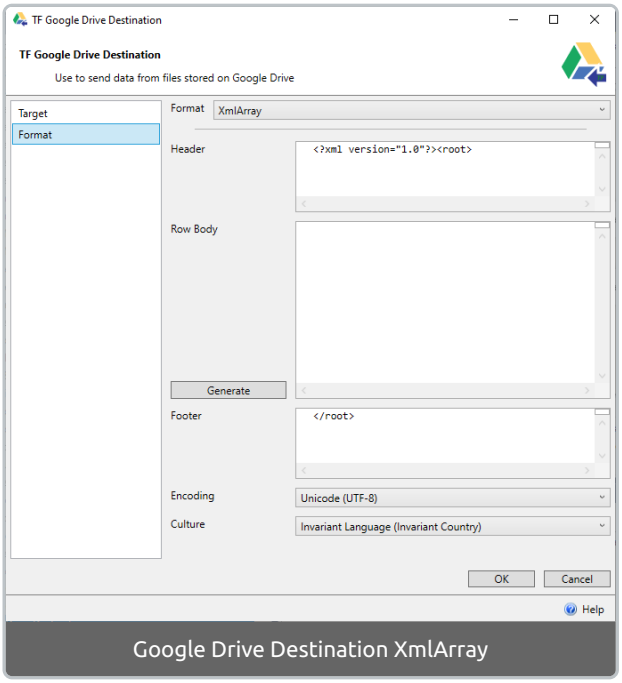
Google Drive Rest Destination Format JsonArray

Option	Description
Generate	Select to automatically generate the Row Body metadata. (Users can also add custom header and footer data.)
Encoding	Select which encoding is used to create the document.
Culture	Used to identify the document's language culture.

XML Array Format



Google Cloud Rest Destination Format XmlArray



Google Drive Destination XmlArray

Option	Description
Generate	Select to automatically generate the Row Body metadata. Users can also add custom header and footer data.
Encoding	Select which encoding is used to create the document.
Culture	Used to identify the document's language culture.

Task Factory Hadoop WebHDFS

Last Modified on 02 February 2022

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

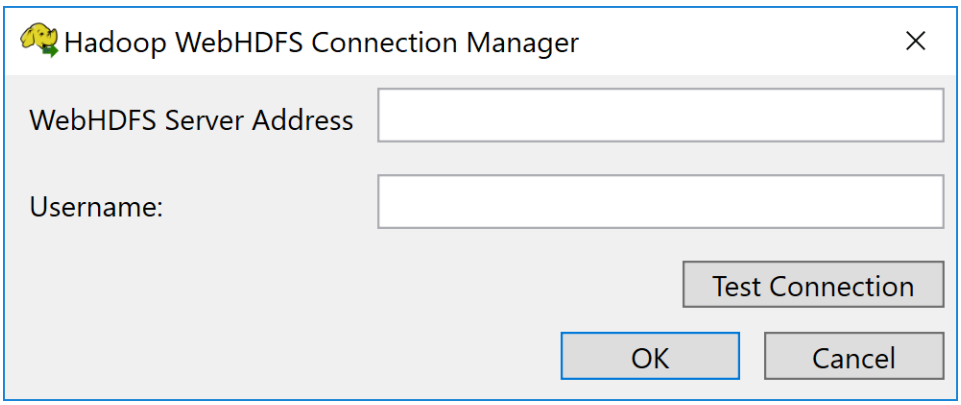
Unsupported: The Hadoop component was deprecated in version 2019.4.1 and is no longer supported.

Important: Users who are able to successfully test their connection yet receive an **unable to connect** error at runtime, please direct your attention to the following help document as you may need to update your local hosts file.

Hadoop WebHDF connection manager is available for SQL versions 2012 and higher.

Hadoop WebHDFS Connection Manager

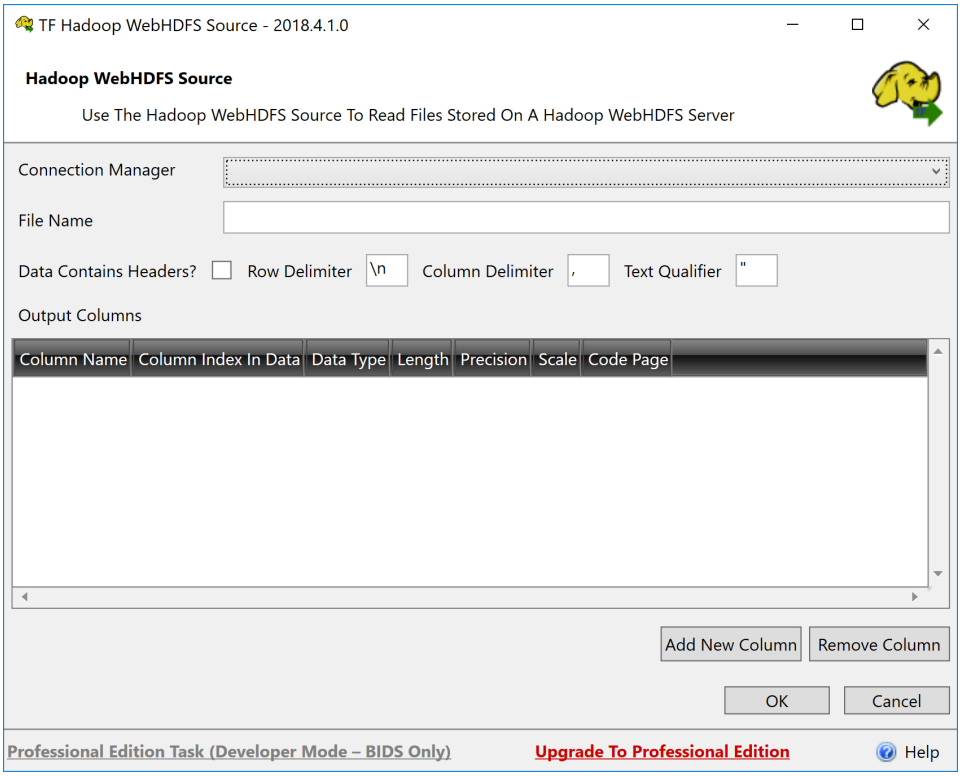
Used with Hadoop WebHDFS Source.



Option	Description
WebHDFS Server Address	The fully qualified web address and port number where the HDFS (Hadoop Distributed File System) is located (example: http://192.168.1.10:50070).
Username	The username with permission to access HDFS files.

Hadoop WebHDFS Source

Source Icon	Source Description
	The **Hadoop WebHDFS Source** is used to stream large files stored in the HDFS of a Hadoop server which can be converted into rows of data within SSIS. Currently, the Hadoop WebHDFS Source only supports text and CSV files. See the Hadoop WebHDFS Connection Manager to learn more about setting up the connection manager.



Option	Description
File Name	The filename (if in the root directory) or path to the files stored within HDFS (example: FolderName/DataFile.txt).
Options	• **Data Contains Headers?** - Similar to the native Flat File Source, this selection identifies the first row as containing column headers. • **Row Delimiter** - Identifies a character or carriage return (\n) to signify a new row. • **Column Delimiter** - Identifies the character used to separate values for the different columns such as a comma. • **Text Qualifier** - Identifies the character used to wrap values such as quotation marks.
Output Columns	Users can create, remove, and configure the name, index (zero-based), data type, length, precision, and scale of the columns being extracted from the text file.

Task Factory Marketo

Last Modified on 02 February 2022

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Unsupported: The Marketo component was deprecated in version 2019.4.1 and is no longer supported.

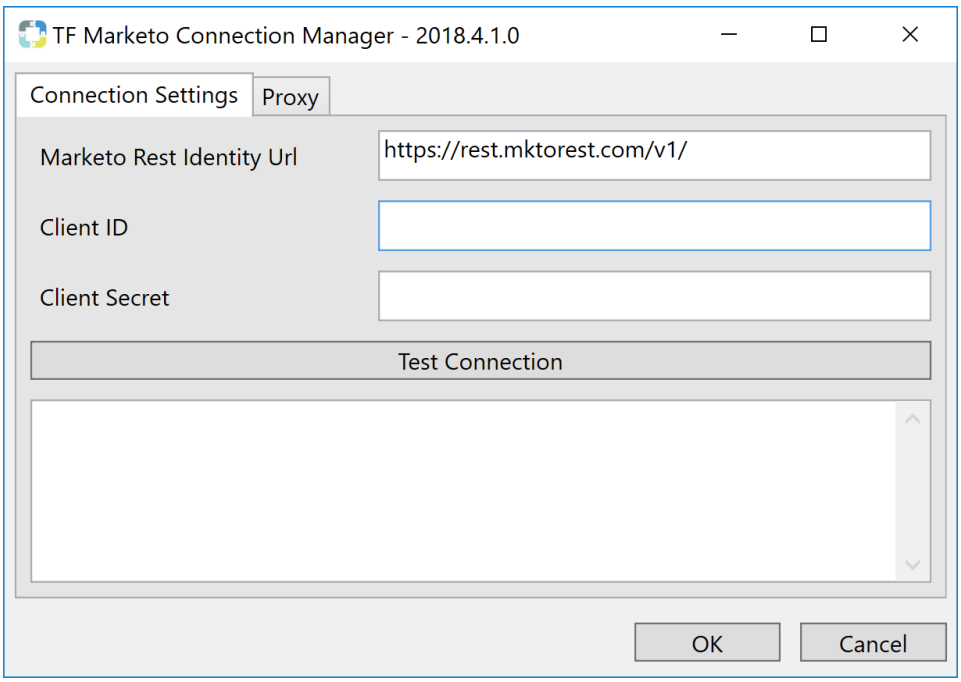
Important: Users need to visit Marketo's developer website to create an application before using the Marketo components. Once created, Marketo assigns the Client ID and Client Secret information needed to connect to their API.

Note: Marketo is available for SQL Server versions 2012 and higher.

Marketo Connection Manager

Used with the **Marketo Source** and **Marketo Destination** data flow components.

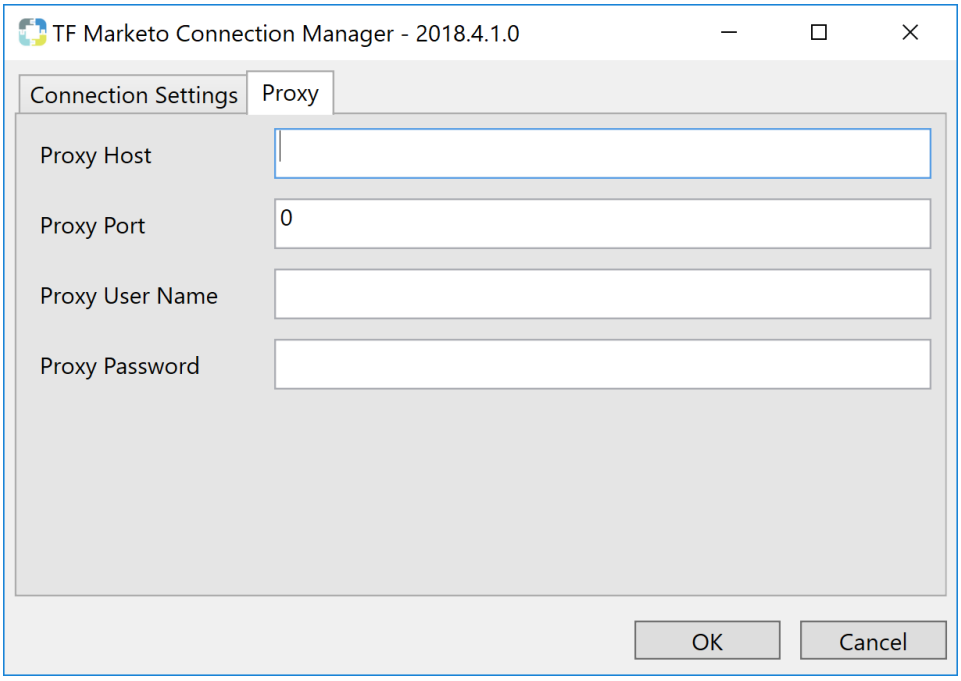
Connection Settings Tab



Option	Description

Option	Description
Marketo Rest Identity Url	In this field, enter the URL of the web service you want to connect to.
Client ID	The application ID assigned by Marketo.
Client Secret	The secret password assigned by Marketo.
Test Connection	Users can test their connection once all fields have been entered. In the pane below, connection and access token information displays.

Proxy Tab

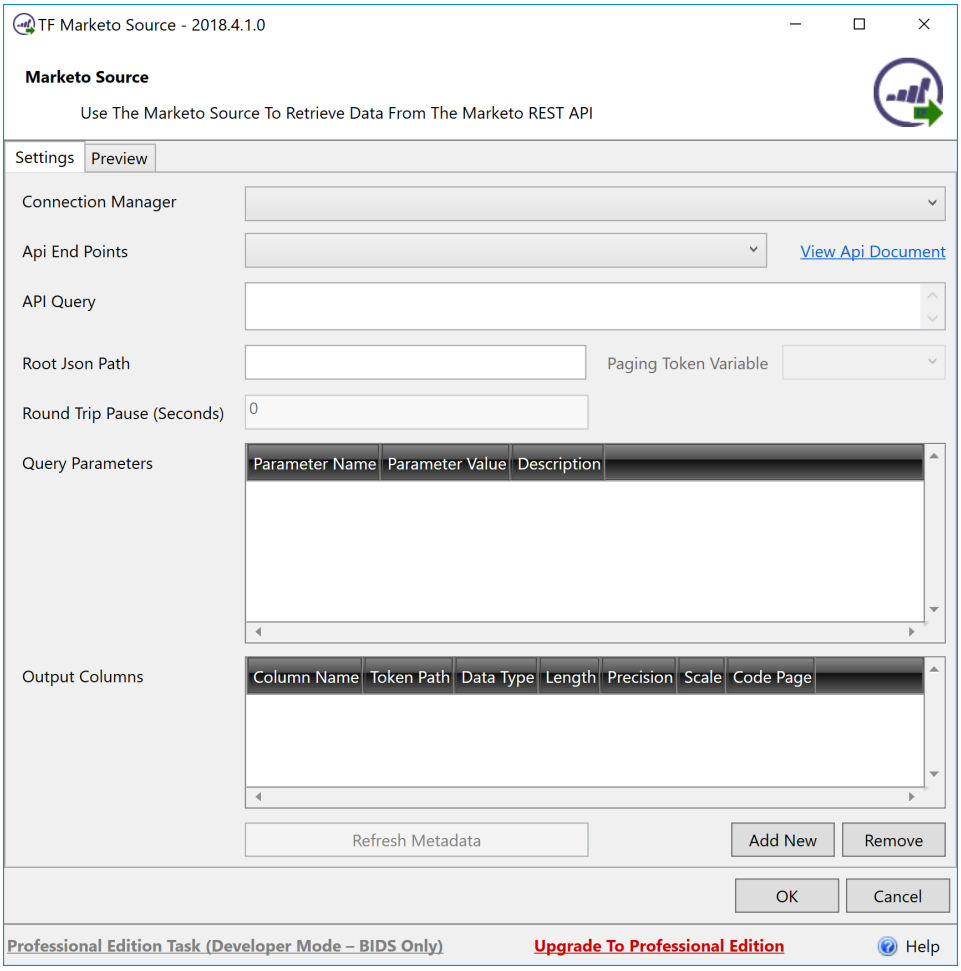


Option	Description
Proxy Host	The IP address or URL used when utilizing a proxy connection.
Proxy Port	The port number that corresponds to the URL proxy host.
User Name	The username needed to authenticate to the proxy.
Password	The password needed to authenticate to the proxy.

Marketo Source

Source Icon	Source Description
	The **Marketo Source** is a data flow component used to retrieve leads, opportunities, and other objects for users that employ the Marketo automated marketing software. See the Marketo Connection Manager to learn more about setting up this component's connection manager.

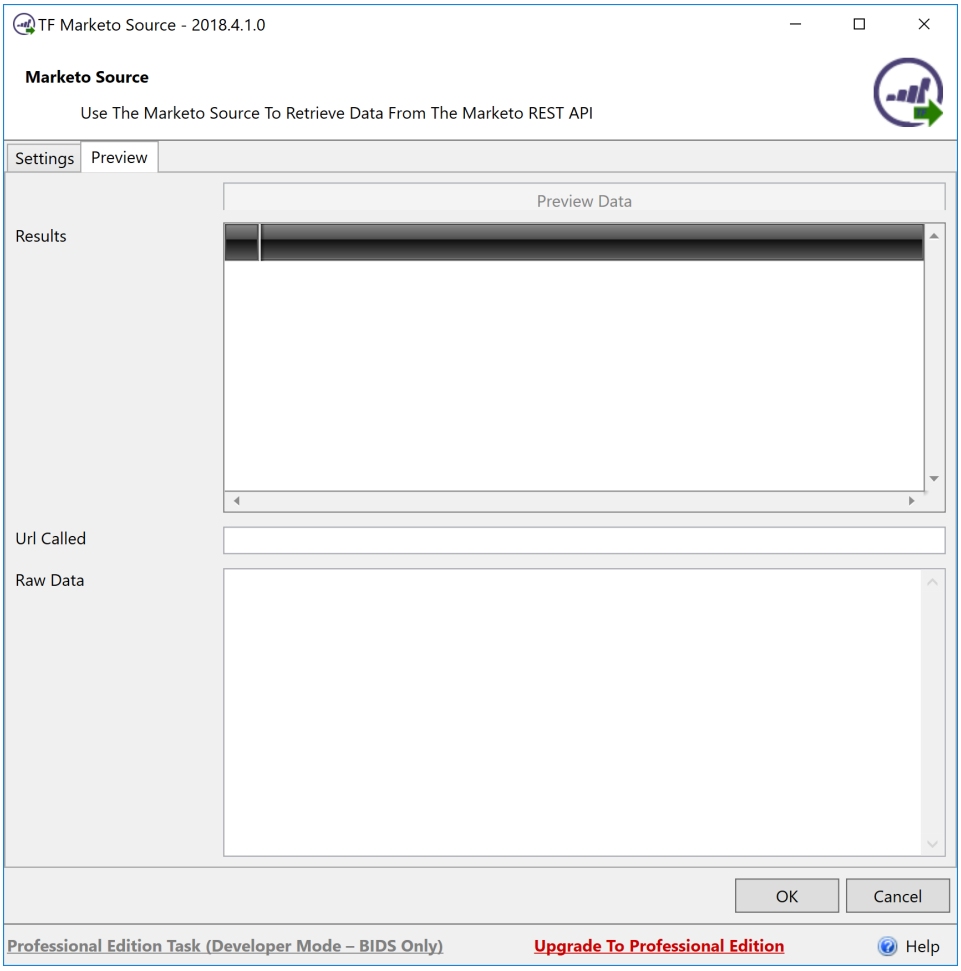
Marketo Source Settings Tab



Option	Description
Connection Manager	In this field, users can create a new connection manager or select a pre-existing one.
Api End Points	In this field, users select the Api they wish to connect to. Selections include but are not limited to Leads - Get Activity Types, Leads, Get Lead Activities, and a custom setting that allows users to configure the component to endpoints not included (selecting the View Api Documents link can assist users with configuration information for all available endpoints.) Selecting an endpoint auto-populates that field's parameters and commonly used Output Columns.

Option	Description
API Query	This field defines the query called to Marketo's web API. Although several end points have been pre-configured, users can type in or paste custom calls that are not included with the component.
Root Json Path	Marketo returns raw data in JSON format, therefore, users need to define the root path when creating custom queries or retrieving information from different levels.
Pause Between Round Trips (Seconds)	The number of seconds between calls to Marketo.
Query Parameters	Optional and required parameters to be configured by the user to filter or set the results being returned. Parameter values can be set dynamically by entering the variable name. Example: <@User::leadID>
Output Columns	In this window, users can add and remove columns and define their name, token path (defined within the raw data), data type, length, precision, scale, and code page.

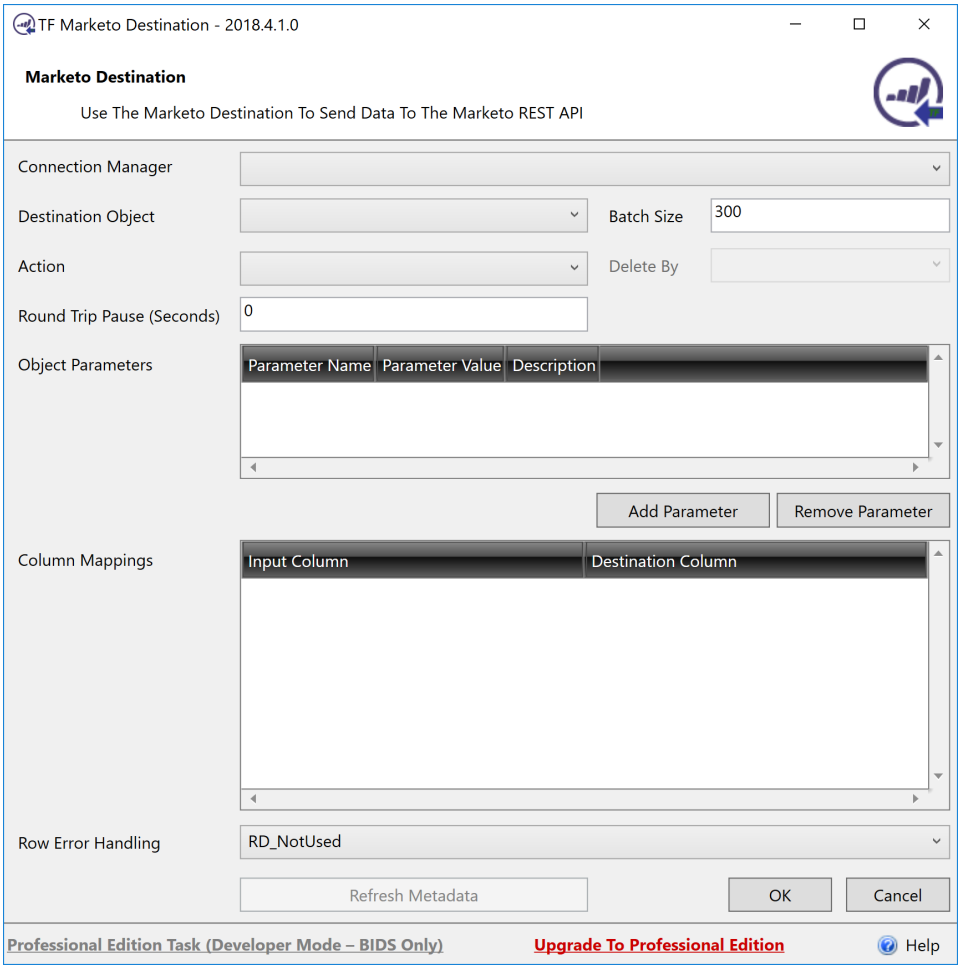
Marketo Source Preview Tab



Option	Description
Preview Data	Selecting this button returns the results and raw data based on the user-defined settings.
Results	In this window, users can see a preview of the first rows of data that output based on the configuration.
Raw Data	This window displays JSON data in raw form based on the configuration in the **Settings** tab. Developers can use this to help determine root and token paths and other fields that can be used.

Marketo Destination

Destination Icon	Destination Description
	The **Marketo Destination** is a data flow component that connects to the Marketo automated marketing software and allows users to insert, update, delete, and duplicate data such as Leads and Opportunities. See the Marketo Connection Manager to learn more about setting up this component's connection manager.



Option	Description
Connection Manager	In this field, users can create a new connection manager or select a pre-existing one.
Destination Object	In this field, users can select between objects (such as Leads and Opportunities) within their Marketo account.
Batch Size	In this field, users can adjust the batch size to help optimize performance. **Note:** 300 is the default and maximum value allowed by the Marketo Api.
Action	This object selects which action the component performs. Actions include Create (insert), Create or Update (default, upsert), Update Only, Delete, and Duplicate.
Pause Between Round Trips (Seconds)	The number of seconds between calls to the application.
Add Parameter	Some calls require parameters during processing. These parameters can be added with this button. Examples: the dedupeFields parameter is needed when deduping leads..
Remove Parameter	Deletes the selected item in the Object Parameters list.
Delete By	This field identifies which key to use when performing a delete action.
Object Parameters	In this window, users can configure optional and required parameters determined by the object and action being performed. Parameter values can be set dynamically by entering the variable name. Example: <@User::leadID> **dedupeField** - Determines how the dedupe occurs on the entity (eg. through an ID column or the dedupe fields.
Column Mappings	In this window, users map the input and destination columns. **Note:** Like-named columns are auto mapped once an object and action is defined.

Additional Information: See the Task Factory Error Row Handling article for more information about this functionality.

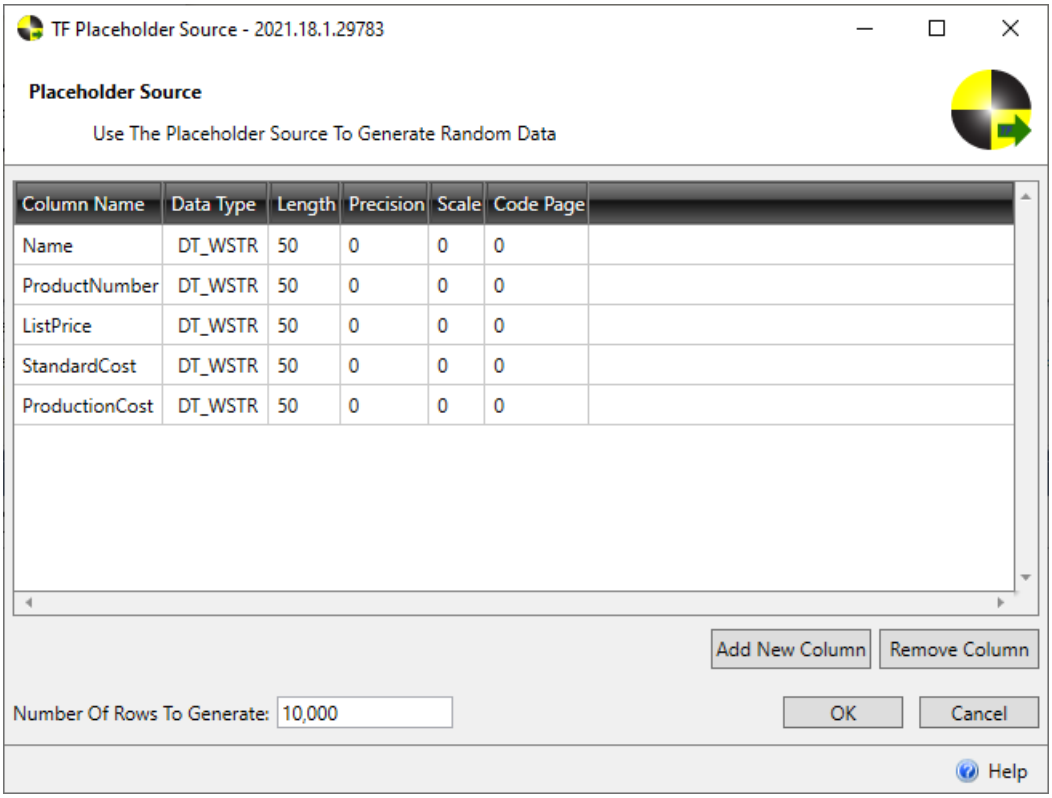
Task Factory Placeholder

Last Modified on 10 February 2022

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

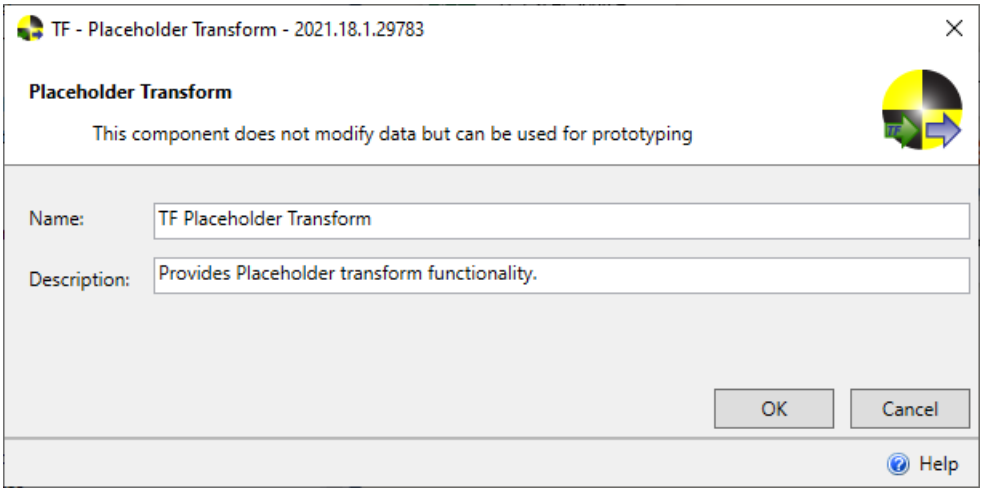
Placeholder Source

Source Icon	Source Description
	The **Placeholder Source** serves two purposes as part of Task Factory: 1. Use as a Data Generator The **Placeholder Source** can be used to generate random data based on the columns defined in the Placeholder Source. In this option, you define the columns and the number of rows you would like generated randomly by setting the **Number of Rows to Generate** option. 2. Use as the stat of a Data Flow Nugget The **Placeholder Source** is used to mark the beginning of a reusable Data Flow Nugget. When used as the beginning of a Data Flow Nugget the number of rows to generate option is ignored.



Placeholder Transform

Transform Icon	Transform Description
	The **Placeholder Transform** is a component developers can use at design time to represent a transform to be used in the future. If executed, data simply passes through unaltered.



The only configurable aspects of the component are the **Name** and **Description** to assist you in recalling what should replace the Placeholder Transformation in the future.

Task Factory REST

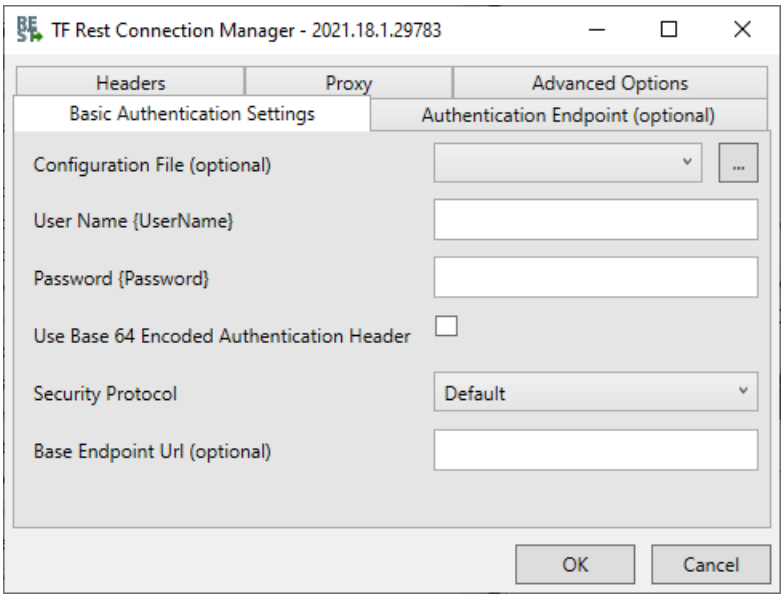
Last Modified on 23 August 2022

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

REST Connection Manager (Basic)

Used with the Rest Source and Rest Destination data flow components, and the Rest Task control flow component.

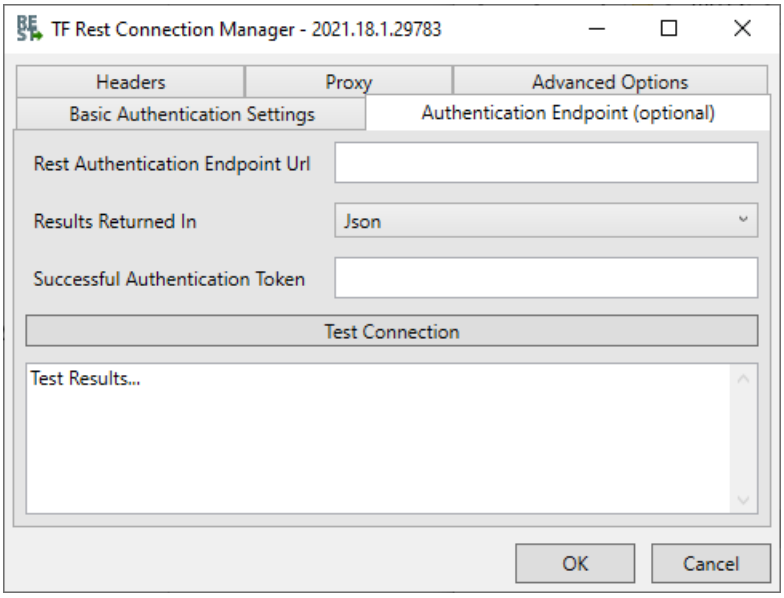
Basic Authentication Settings



Option	Description
Configuration File	The file path to the configuration file. The configuration file can be used to auto-populate the Rest Source and Destination's endpoint URLs. Select the drop-down to view all config files currently in the repository. The repository's location is **C:\Program Files (x86)\Pragmatic Works\Task Factory\OAuth2ConfigFiles**. You can add new config files to this location which appear in the dropdown. **Note:** For config files that exist in a different location, select the ellipsis to launch a file explorer window to navigate to the file.
Skip Authentication	Option for endpoints that do not require authentication. **Note:** Open the component properties to configure this option.
User Name	The username needed to authenticate to the endpoint URL.
Password	The password that allows you to authenticate to the endpoint URL.
Use Base 64 Encoded Authentication Header	Select when Base 64 Encoded Authentication is used by an endpoint. This option takes your username and password delimited with a colon (:), and Base 64 encodes them. This encoded value is then sent as a header. **Note:** If you've configured your API with Postman, this option is configured automatically.

<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>Security Protocol</td><td>The Security Protocol used by the API. The Default option uses the TLS version used by your .NET version by default. See the following chart for more information:
<table>
<tr><th>.NET Version</th><th>TLS Support Information</th></tr>
<tr><td>.NET 4.6 and above</td><td>Supports TLS 1.2 by default.</td></tr>
<tr><td>.NET 4.5</td><td>TLS 1.2 is supported, but you need to opt-in to use it. Execute the following code before making a connection to a secured resource to make TLS 1.2 the default :
ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12</td></tr>
<tr><td>.NET 4.0</td><td>TLS 1.2 is not supported. If you have .NET 4.5 (or above) installed on the system, you can opt in for TLS 1.2 even if your application framework doesn't support it. SecurityProtocolType in .NET 4.0 doesn't have an entry for TLS1.2, so you need to use a numerical representation of this enum value:
ServicePointManager.SecurityProtocol = (SecurityProtocolType)3072;</td></tr>
<tr><td>.NET 3.5 or below</td><td>TLS 1.2 is not supported, and there is no workaround. Upgrade your application to a more recent version of the framework.</td></tr>
</table></td></tr>
<tr><td>Base Endpoint URL</td><td>The endpoint url for your REST connection.
Note: This setting is optional.</td></tr>
</table>

Authentication Endpoint



Important: If you aren't getting an access token from your API, you don't need to configure this tab.

Note: Some endpoints may require a successful authentication token every time they are accessed. You can dynamically set this token using a variable by applying the following format:
Example:**<@User::SuccessToken>**

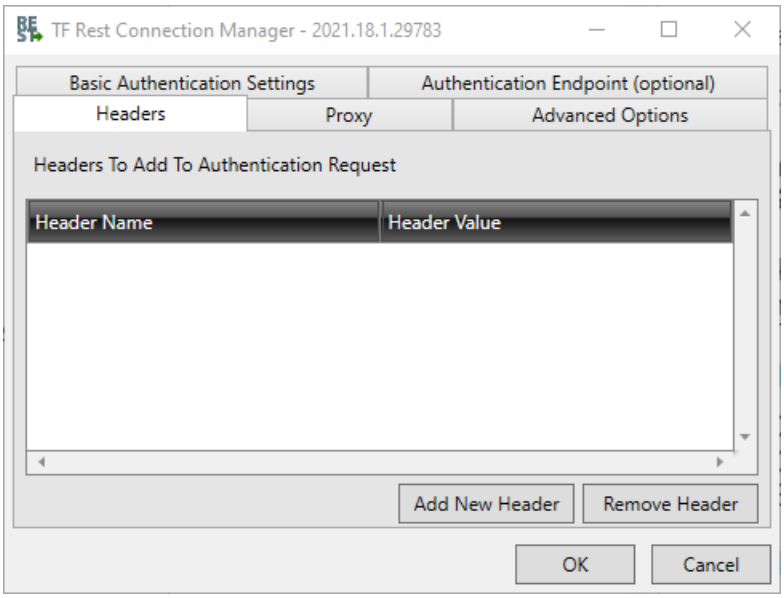
For information about setting a dynamic token as a bearer token for your Rest Source, see the following support article.

Option	Description
Rest Authentication Endpoint Url	The endpoint that returns an access token. For example, In this field, you should enter the URL of the website/ web service that you want to connect to. **Note:** Your token endpoint will typically end in token, for example **google.com/api/token**
Results Returned In	The Rest Source allows you to connect to endpoints with JSON and XML formats. This selection determines which format the component uses.

Option	Description

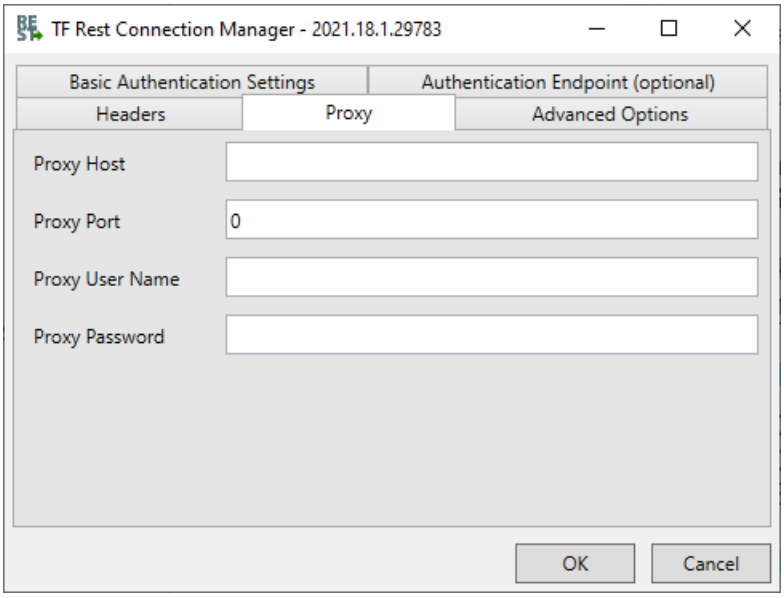
Successful Authentication Token	The token path of the token that you want to get. For example, if your response looks like :
	`{`
	`"access_token": "123",`
	`"refresh_token":"456"`
	`}`
	
	Your successful Authentication will be **access_token**.

Headers



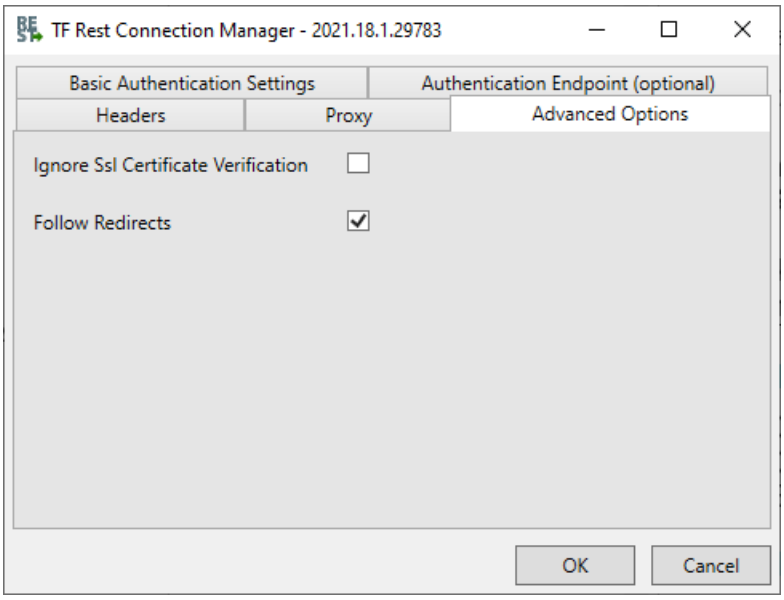
Option	Description
Add New Header	Select to add a new header name and value. Once added, you can select within the column and configure the name and the value.
Remove Header	Removes the selected header to be included in the component's output.

Proxy



Option	Description
Proxy Host	When connecting to an endpoint using a proxy, you should enter the proxy URL in this field.
Proxy Port	The port number that corresponds to the URL proxy host.
User Name	The username needed to authenticate to the proxy.
Password	The password that allows you to authenticate to the proxy.

Advanced Options



Option	Description

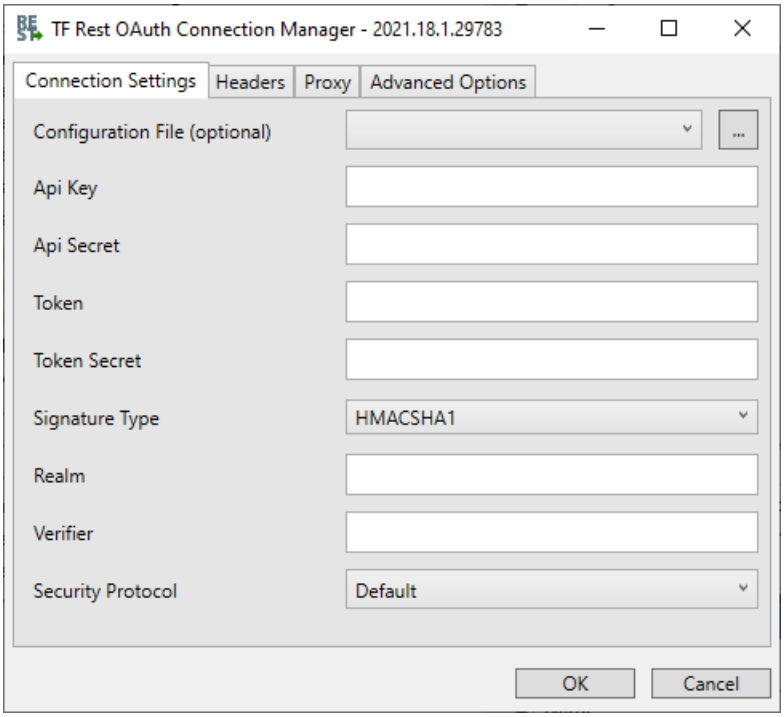
Option	Description
Ignore Ssl Certificate Verification	Selecting this option ignores the SSL Certificate notification from the API during the connection. **Note:** This option applies to APIs that require you to have an SSL Certificate. Select this option if the SSL Certificate is not used.
Follow Redirects	Selecting this option implements a 301 redirect on configured endpoints. For example, this option would automatically send your request to an HTTPS redirect once the request has reached the http version of the site.

Important: You need to register a new app with the service being used to obtain API Keys and Secrets. This can be done by visiting the service's developer website.

REST OAuth Connection Manager

Used with the Rest Source and Rest Destination data flow components, and the Rest Task control flow component.

Connection Settings

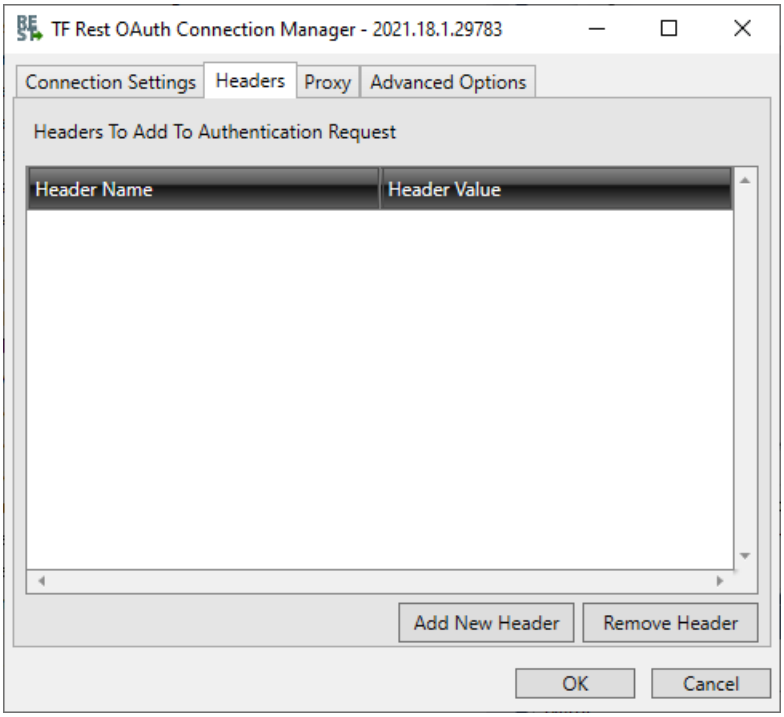


Option	Description

Option	Description
Configuration File	The file path to the configuration file. The configuration file can be used to auto-populate the Rest Source and Destination's endpoint URL's. Select the down arrow to view all config files currently in the repository. The repository's location is **C:\Program Files (x86)\Pragmatic Works\Task Factory\OAuth2ConfigFiles**. You can add new config files to this location which appear in the dropdown. **Note:** For config files that exist in a different location, select the ellipsis to launch a file explorer window to navigate to the file.
Api Key, API Secret, Token, and Token Secret	Authentication information provided to you at the application's developer site. **Note:** The Token field can be left empty for API's that do not require one.
Signature Type	Identifies the hash algorithm used in the vendor's authentication process.
Realm and Verifier	Optional properties supplied by the vendor that the connection is authenticating with.

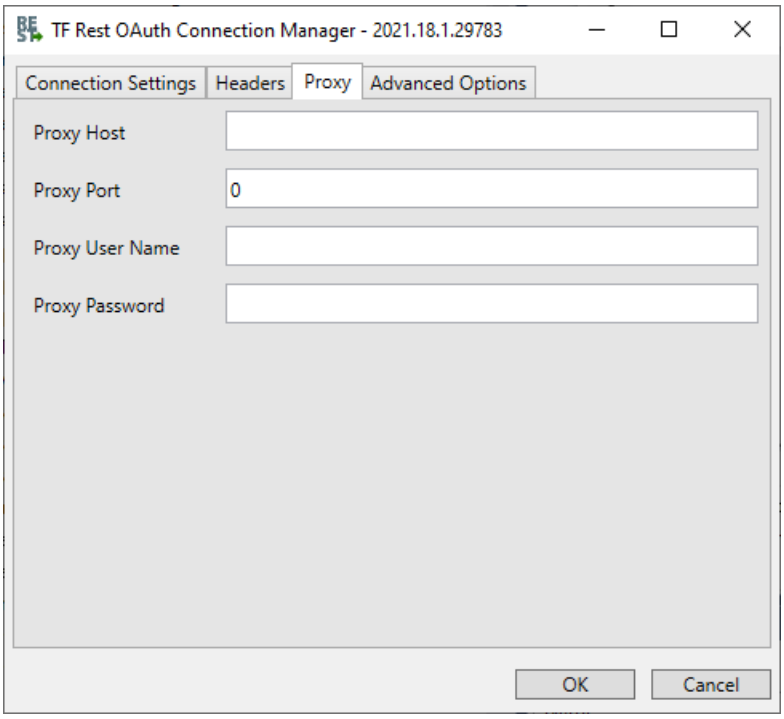
<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>Security Protocol</td><td>The Security Protocol used by the API. The Default option uses the TLS version used by your .NET version by default. See the following chart for more information:
<table>
<tr><th>.NET Version</th><th>TLS Support Information</th></tr>
<tr><td>.NET 4.6 and above</td><td>Supports TLS 1.2 by default.</td></tr>
<tr><td>.NET 4.5</td><td>TLS 1.2 is supported, but you need to opt-in to use it. Execute the following code before making a connection to a secured resource to make TLS 1.2 the default:
ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12</td></tr>
<tr><td>.NET 4.0</td><td>TLS 1.2 is not supported. If you have .NET 4.5 (or above) installed on the system, you can opt in for TLS 1.2 even if your application framework doesn't support it. SecurityProtocolType in .NET 4.0 doesn't have an entry for TLS1.2, so you need to use a numerical representation of this enum value:
ServicePointManager.SecurityProtocol = (SecurityProtocolType)3072;</td></tr>
<tr><td>.NET 3.5 or below</td><td>TLS 1.2 is not supported, and there is no workaround. Upgrade your application to a more recent version of the framework.</td></tr>
</table></td></tr>
</table>

Headers



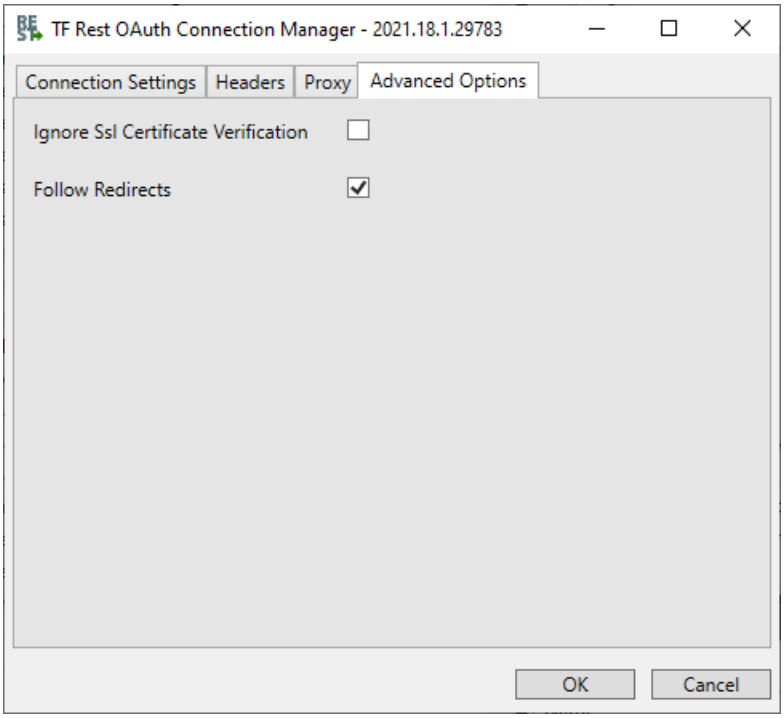
Option	Description
Add New Header	Select to add a new header name and value. Once added, you can select within the column and configure the name and the value.
Remove Header	Remove the selected header to be included in the component's output.

Proxy



Option	Description
Proxy Host	When connecting to an endpoint using a proxy, you should enter the proxy URL in this field.
Proxy Port	The port number that corresponds to the URL proxy host.
User Name	The username needed to authenticate to the proxy.
Password	The password that allows you to authenticate to the proxy.

Advanced Options



Option	Description
Ignore Ssl Certificate Verification	Selecting this option ignores the SSL Certificate notification from the API during the connection. **Note:** This option applies to APIs that require you to have an SSL Certificate. Select this option if the SSL Certificate is not needed.
Follow Redirects	Selecting this option implements a 301 redirect on configured endpoints. For example, this option would automatically send your request to an HTTPS redirect once the request has reached the HTTP version of the site.

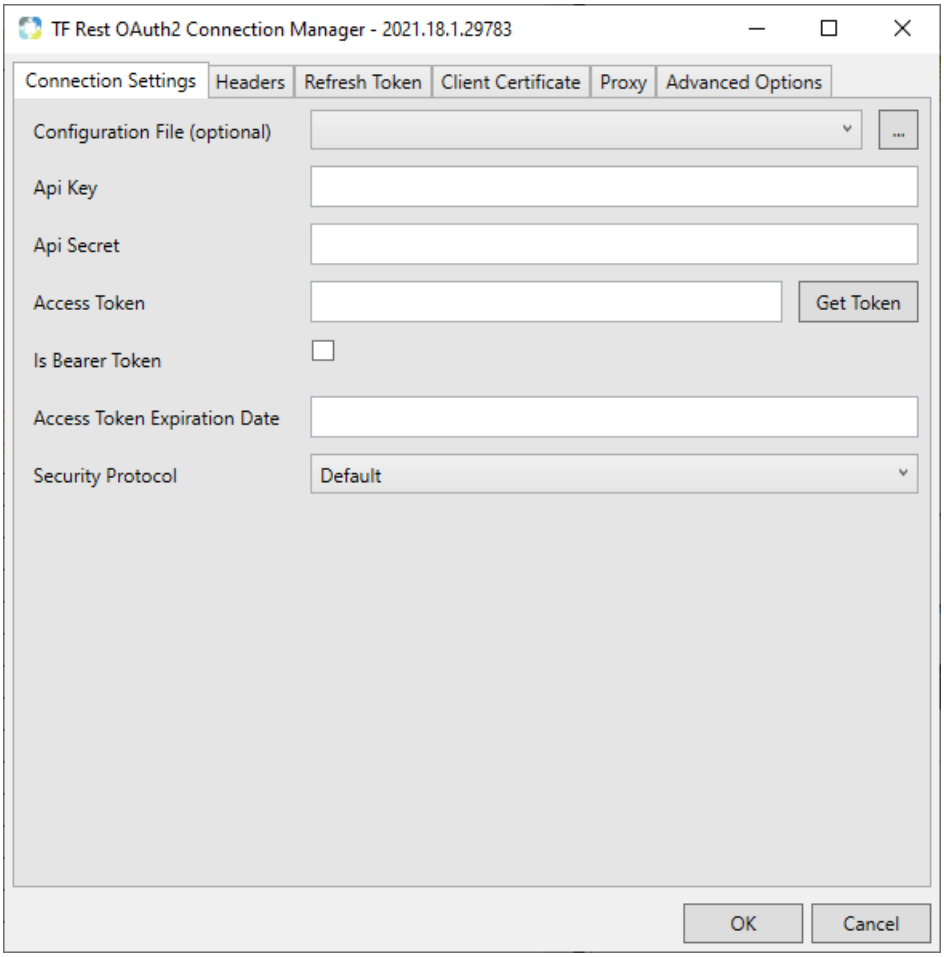
Important: You need to register a new app with the service being used to obtain API Keys and Secrets. This can be done by visiting the service's developer website. For more information about creating

and using OAuth2 connections, see OAuth2 Simplified.

REST OAuth2 Connection Manager

Used with the Rest Source and Rest Destination data flow components, and the Rest Task control flow component.

Connection Settings



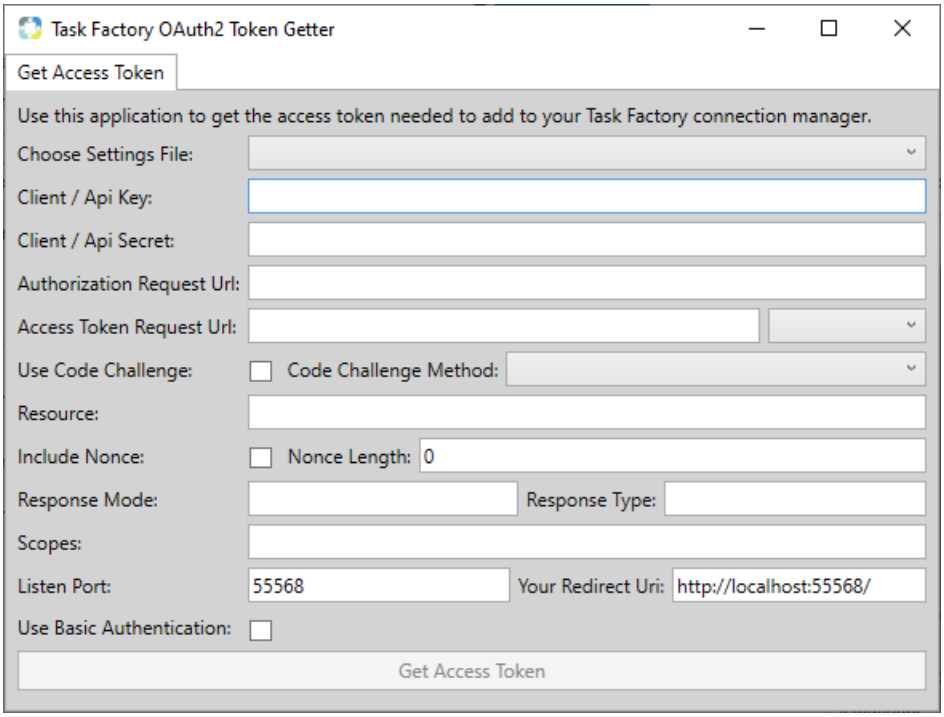
Option	Description
Configuration File	You can create and use configuration files to setup the OAuth2 connections. Select the down arrow to view all config files currently in the repository. The repository's location is **C:\%PROGRAMFILES(X86)%\SentryOne\Task Factory\OAuth2ConfigFiles**. You can add new config files to this location which appear in the drop-down. **Note:** For config files that exist in a different location, select the ellipsis to launch a file explorer window to navigate to the file. **Additional Information:** For a tutorial explaining configuration files for the Rest OAuth2 connection manager, see the following video.
Api Key and Api Secret	Authentication information provided to users at the application's developer site.
Access Token	To obtain this information if you do not already have it, select the **Get Token** button and complete the information requested in the Token Getter user interface.

<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>Is Bearer Token</td><td>Select this option if the Access Token is a bearer token.</td></tr>
<tr><td>Access Token Expiration Date</td><td>The expiration date of the access token.

ℹ Note: This settings is optional.</td></tr>
<tr><td>Security Protocol</td><td>The Security Protocol used by the API. The Default option uses the TLS version used by your .NET version by default. See the following chart for more information :
<table>
<tr><th>.NET Version</th><th>TLS Support Information</th></tr>
<tr><td>.NET 4.6 and above</td><td>Supports TLS 1.2 by default.</td></tr>
<tr><td>.NET 4.5</td><td>TLS 1.2 is supported, but you need to opt-in to use it. Execute the following code before making a connection to a secured resource to make TLS 1.2 the default :

• ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12</td></tr>
<tr><td>.NET 4.0</td><td>TLS 1.2 is not supported. If you have .NET 4.5 (or above) installed on the system, you can opt in for TLS 1.2 even if your application framework doesn't support it. SecurityProtocolType in .NET 4.0 doesn't have an entry for TLS1.2, so you need to use a numerical representation of this enum value:
• ServicePointManager.SecurityProtocol = (SecurityProtocolType)3072;</td></tr>
<tr><td>.NET 3.5 or below</td><td>TLS 1.2 is not supported, and there is no workaround. Upgrade your application to a more recent version of the framework.</td></tr>
</table></td></tr>
</table>

Token Getter (opened when Get Token button is used.)



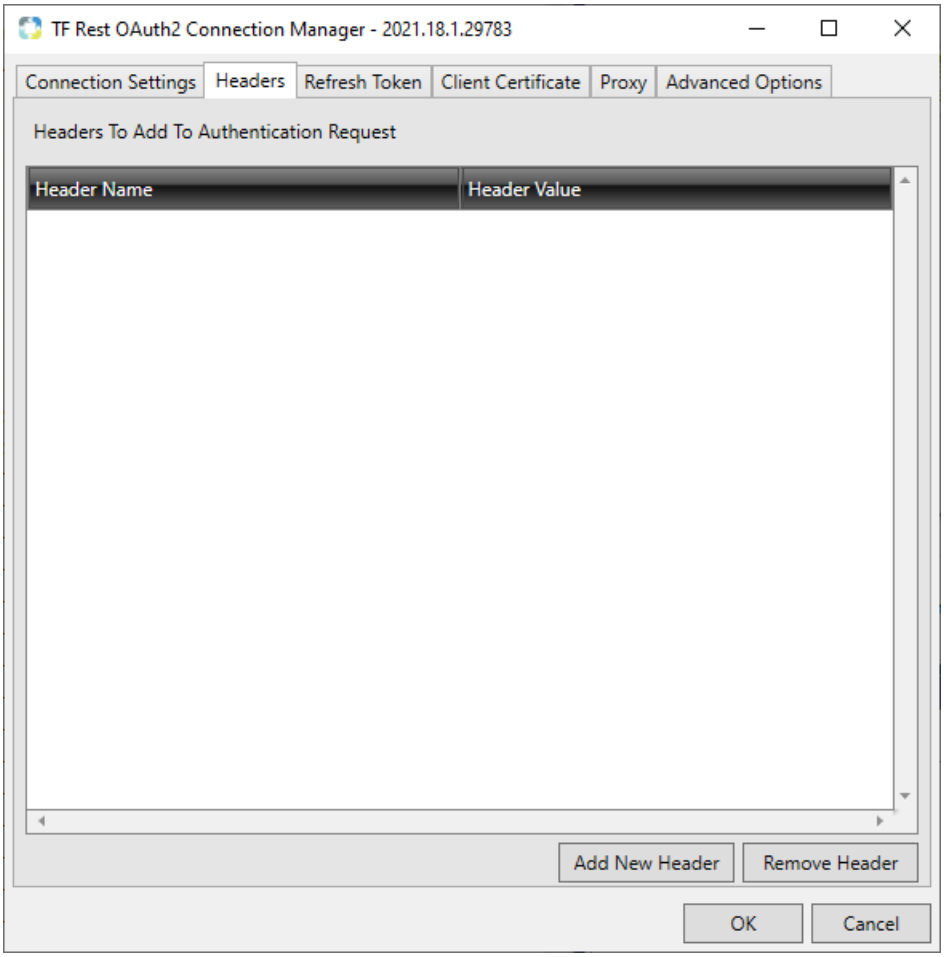
Note: Please see your service's online documentation to assist in retrieving the information below.

Option	Description
Choose Settings File	You can select pre-configured settings for several popular services including, but not limited to: Google, GoToMeeting, and PayPal Sandbox. **Note:** You will use **.oauth** files for this option, which can be found in the following directory **C:\%PROGRAMFILES(X86)%\SentryOne\Task Factory\OAuth2ConfigFiles**
Client / Api Key	Authentication information provided to you at the application's developer site. **Note:** This should match the API Key from the Connection Settings window.
Client / Api Secret	Authentication information provided to you at the application's developer site. **Note:** This should match the API Secret from the Connection Settings window.
Authorization Request Url	The URL used to request authorization from the service. Example: **https://api.citrixonline.com/oauth/authorize?client_id={ApiKey})**

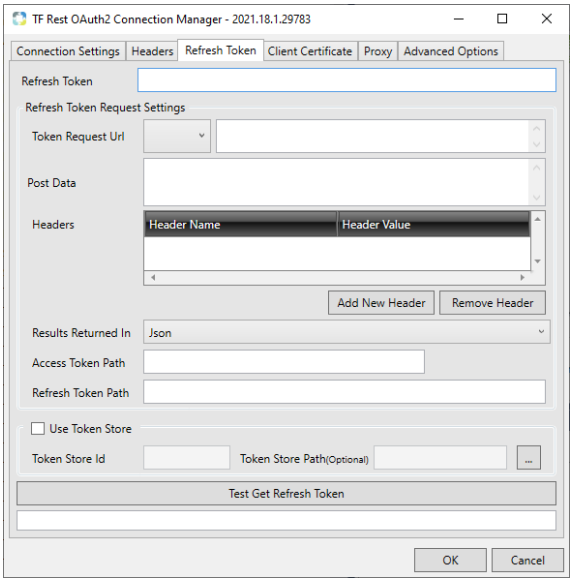
Option	Description
Access Token Request Url	The Url used to request an access token from the service. Example: **https://api.citrixonline.com/oauth/access_token**
Use Code Challenge	Select to use a challenge code.
Code Challenge Method	Select the available challenge code method: **Plain** or **S256**.
Resource	The server that will handle the authenticated request.
Include Nonce	Select to include nonce.
Nonce Length	Enter the desired nonce length.
Response Mode	The format of the request that's required for the call.
Response Type	The type of the request that's required for the call.
Scopes	The list of permissions that you want to use in the call.
Listen Port	The port on your local machine where you want to send the token request response.
Your Redirect Uri	Use this as your Redirect Uri in your OAuth2 App settings.
Use Basic Authentication	Select to use Basic Authentication.
Get Access Token	Selecting this button opens the (configured) service's online authentication screen in the window below. To complete the process, enter the credentials of the account used to access the service. If successful, the Token Getter window closes and you return to the Connection Settings window. **Note:** Bad requests should redirect to a URL configured either in the application or the access token request URL. You can also copy the URL to help troubleshoot invalid requests.

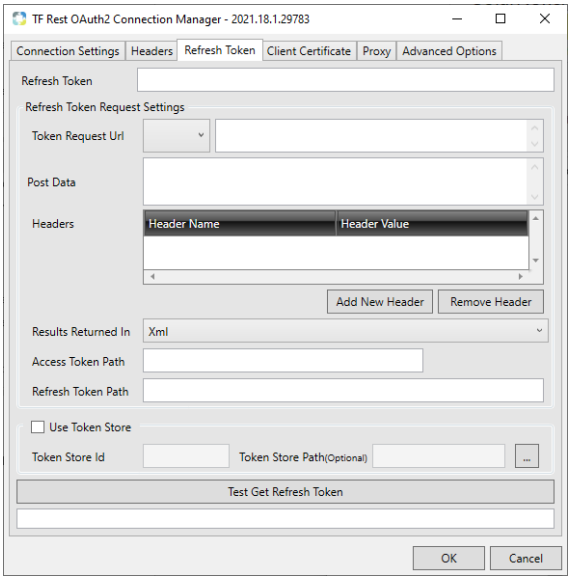
Headers

You can create header names and values by entering the information in their corresponding windows.



Refresh Token (Results Returned In JSON / XML)



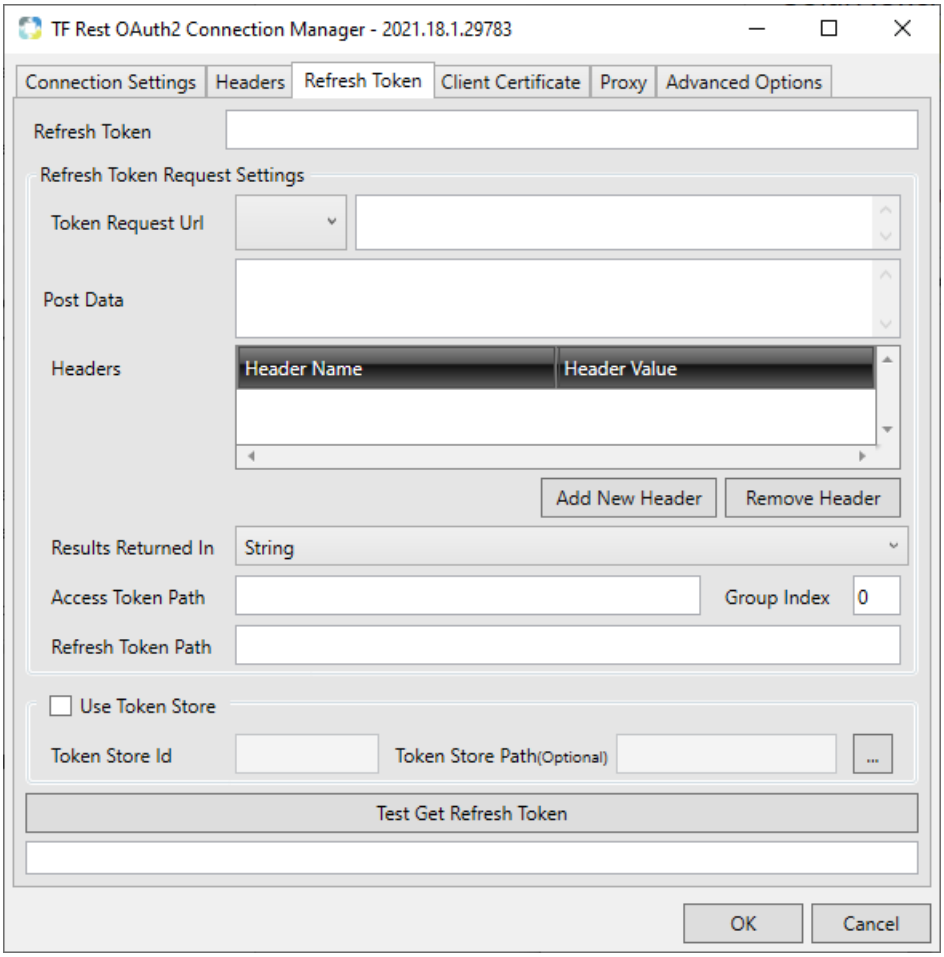


> **Note:** When you get an access token manually from the get token button or you've copied it from your service, it only does this once which could cause potential failures in your package. You need to have a refresh token setup to get subsequent tokens in your package.

Additional Information: For information about OAuth 2.0 grants, see the following guide.

Option	Description
Refresh Token	Authentication information provided to you at the application's developer site. **Note:** This should match the API Key from the Connection Settings window.
Token Request Url	The URL that returns a refresh token. Example: **https://api.citrixonline.com/oauth/access_token**
Headers	Allows you to add headers within the refresh token. Select **Add New Header** to add a header. Select **Remove header** to remove a header.
Results Returned In	Specifies how results are returned from the application (JSON, XML, or String.)
Access Token Path	The token path for the access token. Example: **access_token**
Refresh Token Path	The token path for the refresh token. Example: **refresh_token**
Use Token Store	This option creates a file on your machine that saves the access token, and refresh token in an encrypted format.
Token Store Id	Identifies the Token Store Id.
Token Store Path (Optional)	Identifies the Token Store path.
Test Get Refresh Token	Select this option to test the retrieval of the Refresh Token.

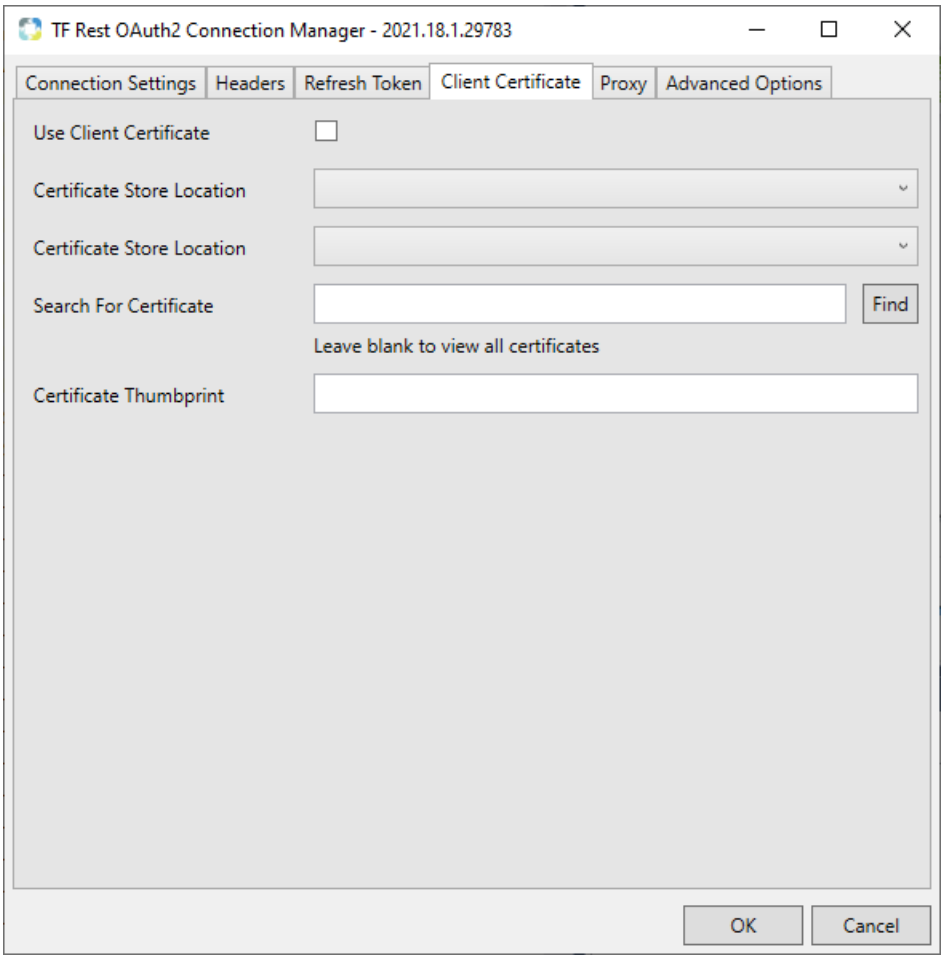
Refresh Token (Results Returned In String)



Option	Description
Refresh Token	Authentication information provided to you at the application's developer site. **Note:** This should match the API Key from the Connection Settings window.
Token Request Url	The URL that returns a refresh token. Example: **https://api.citrixonline.com/oauth/access_token**
Headers	Allows you to add headers within the refresh token. Select **Add New Header** to add a header. Select **Remove header** to remove a header.
Results Returned In	Specifies how results are returned from the application (JSON or XML.)
Access Token Path	The token path for the access token. Example: **access_token**
Refresh Token Path	The token path for the refresh token. Example: **refresh_token**
Group Index	Identifies the index position to return (0 based).
Use Token Store	This option creates a file on your machine that saves the access token, and refresh token in an encrypted format.

Option	Description
Token Store Id	Identifies the Token Store Id.
Token Store Path (Optional)	Identifies the Token Store path.
Test Get Refresh Token	Select this option to test the retrieval of the Refresh Token.

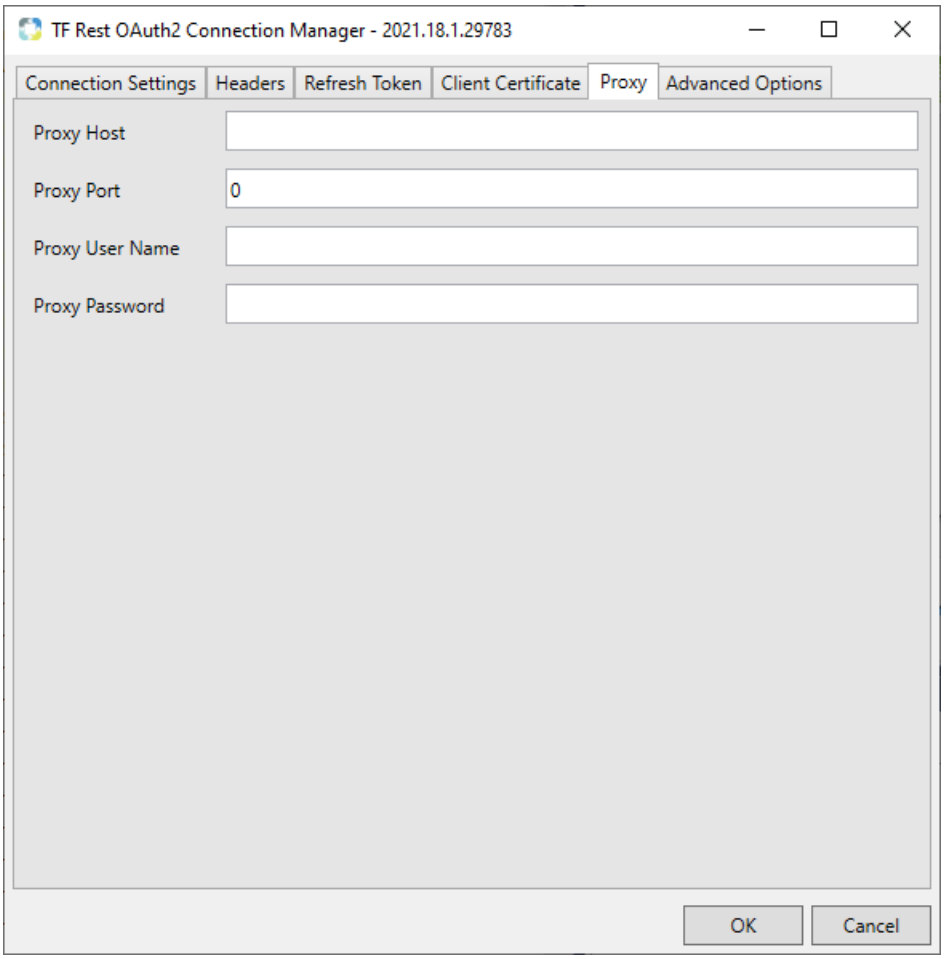
Client Certificate



Option	Description
Use Client Certificate	Select this option if you want to use the client certificate.
Certificate Store Location	The client store location.
Search For Certificate	Enter a certificate that you want to search for and then select **Find**. **Note:** Leave this section blank and select **Find** to view all certificates.

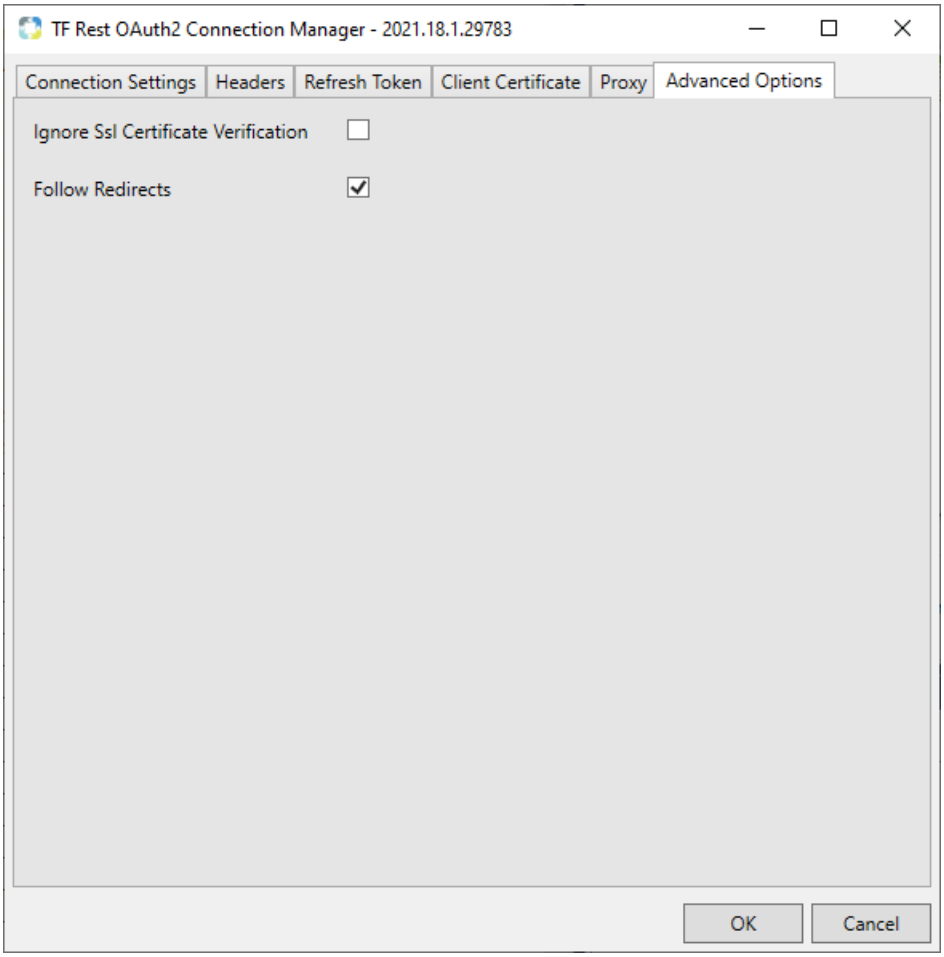
Option	Description
Certificate Thumbprint	The identifier thumbprint of the selected certificate.

Proxy



Option	Description
Proxy Host	When connecting to an endpoint using a proxy, you should enter the proxy URL in this field.
Proxy Port	The port number that corresponds to the URL proxy host.
User Name	The username needed to authenticate to the proxy.
Password	The password that allows you to authenticate to the proxy.

Advanced Options



Option	Description
Ignore Ssl Certificate Verification	Selecting this option ignores the SSL Certificate notification from the API during the connection. **Note:** This option applies to APIs that require you to have an SSL Certificate. Select this option if the SSL Certificate is not needed.
Follow Redirects	Selecting this option implements a 301 redirect on configured endpoints. For example, this option would automatically send your request to an HTTPS redirect once the request has reached the HTTP version of the site.

Configuration Files

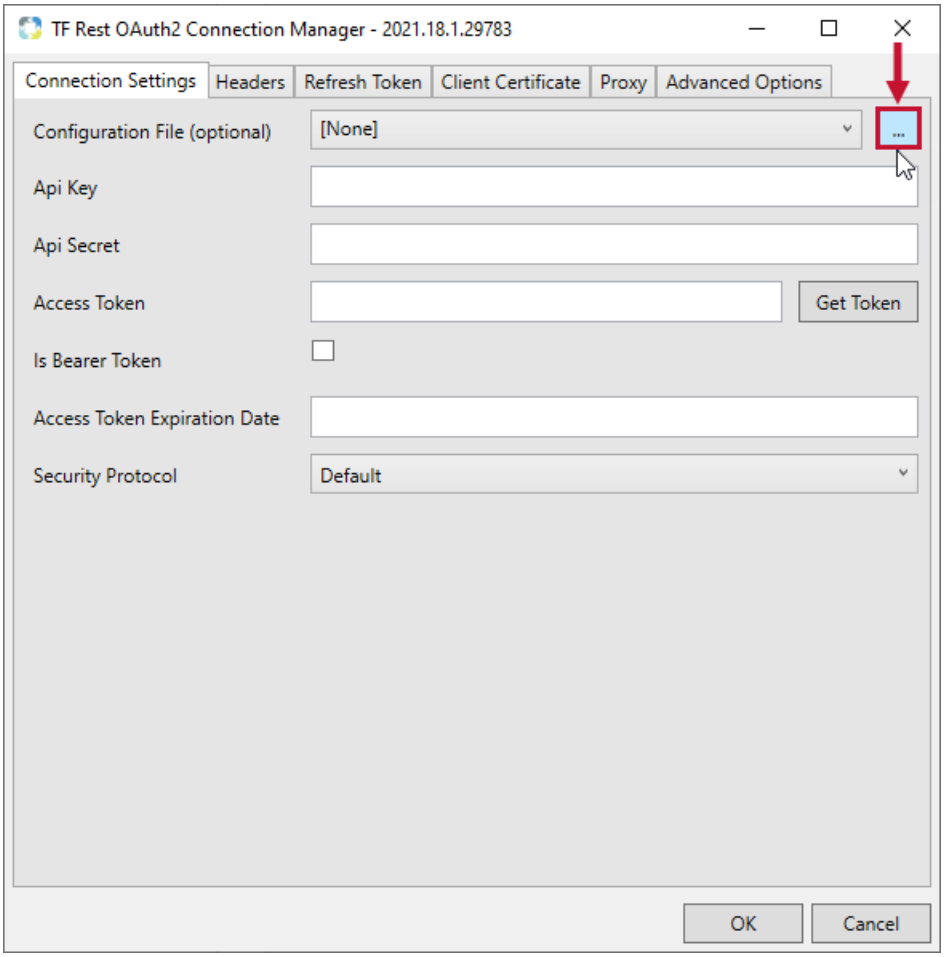
Additional Information: Customize the configuration files to fit your needs. For a full list of the configuration files for Task Factory Rest connection managers, see the Task Factory GitHub.

You can choose to create or select a configuration file to help you configure your OAuth2 connection manager. Select a link to download the corresponding config file.

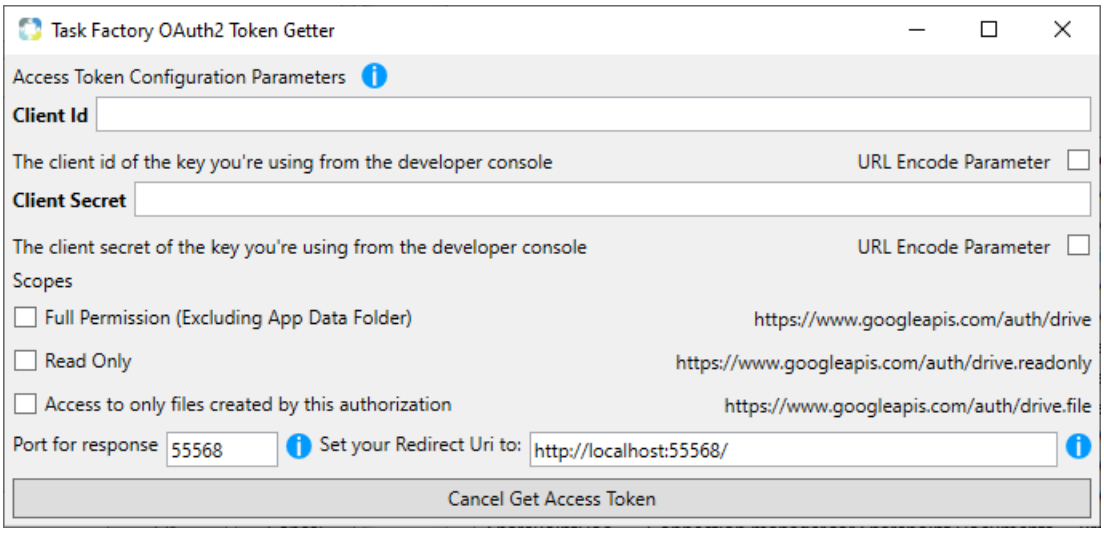
- Azure
- Dynamics CRM
- Facebook
- Fitbit
- Google Analytics
- HubSpot
- Jive
- SalesForce
- Visual Studio Online
- Yelp
- Youtube

Select a configuration file to configure your OAuth2 Connection Manager by completing the following steps:

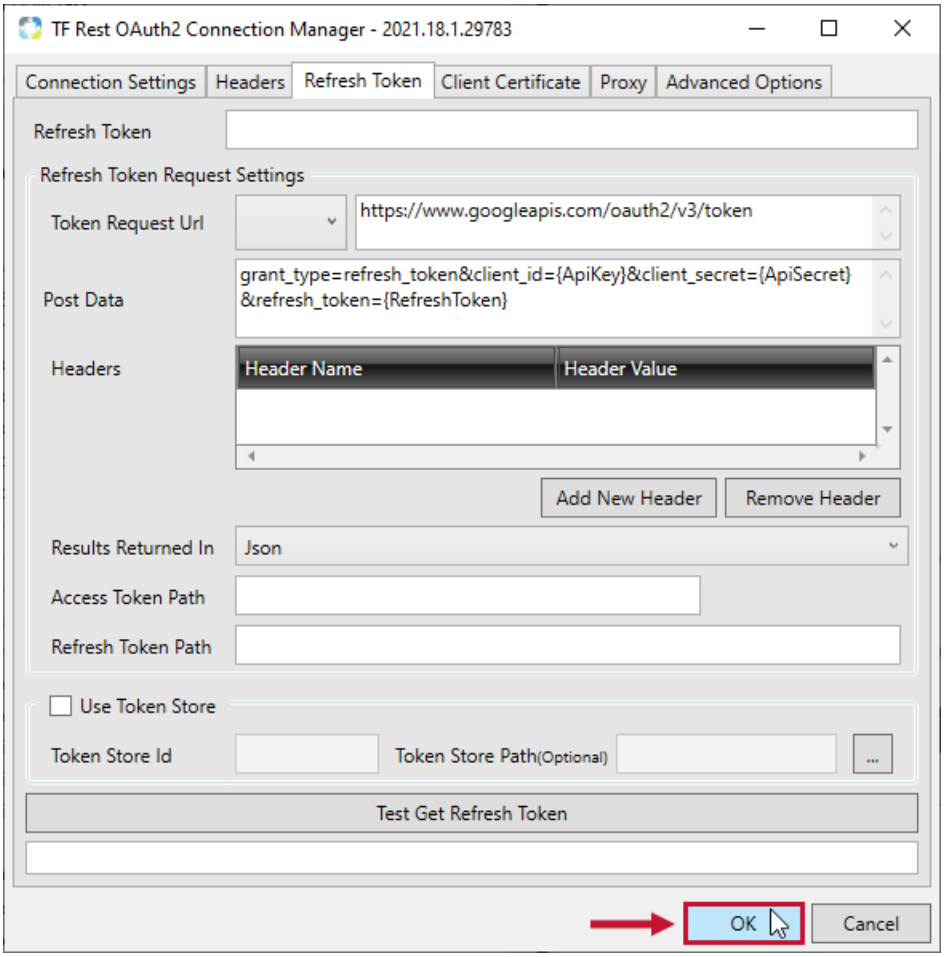
1. Select the **ellipsis ···** to open the file explorer. Navigate to the .config file and select **OK** to open the OAuth2 Token Getter. **Note:** You can also select an applicable Configuration File from the drop-down menu.



2. Enter your access token configuration parameters in the OAuth2 Token Getter. Select **Get Access Token**, to open a sign-in window, and enter the application's credentials. ℹ **Note:** The Task Factory OAuth2 Token Getter will require different Access Token Configuration Parameters based on your configuration file selection. This example displays the Google Drive OAuth2 Token Getter requirements.



3. After permission is granted, the **OAuth2 Token Getter** window closes and the **Refresh Token** tab populates with the required information. Below is an example of Google Analytics. (The actual Refresh Token has been deleted for security.) Select **OK** to close the Connection Manager.

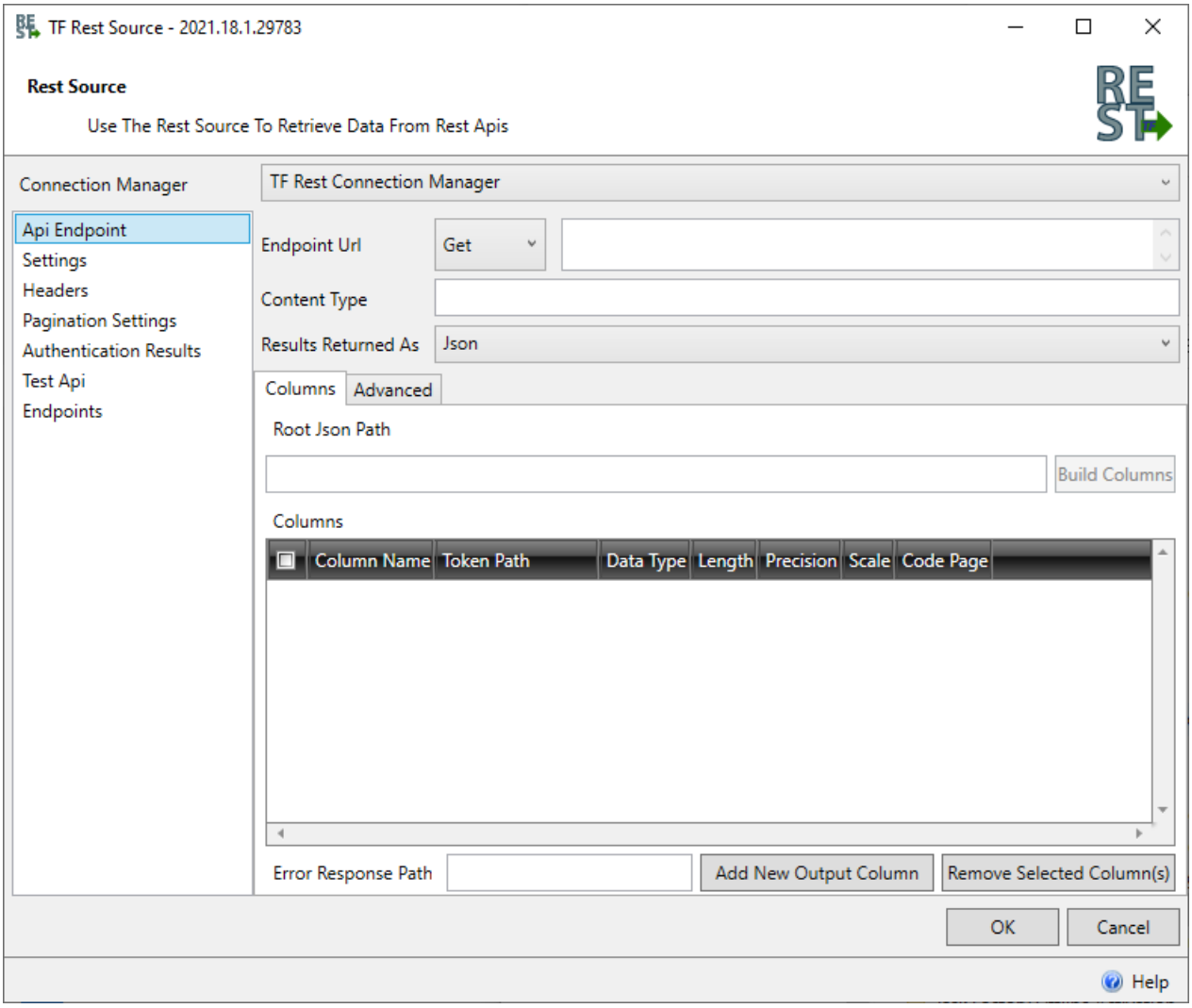


REST Source

Source Icon	Source Description

Source Icon	Source Description
	The **REST Source** allows you to connect to a web service that utilizes a REST API and extracts data in XML or JSON format. See the Rest Source Connection Manager, Rest Source OAuth Connection Manager, and the Rest Source OAuth2 Connection Manager to learn more about configuring the component's connection manager.

API Endpoint / Output Columns



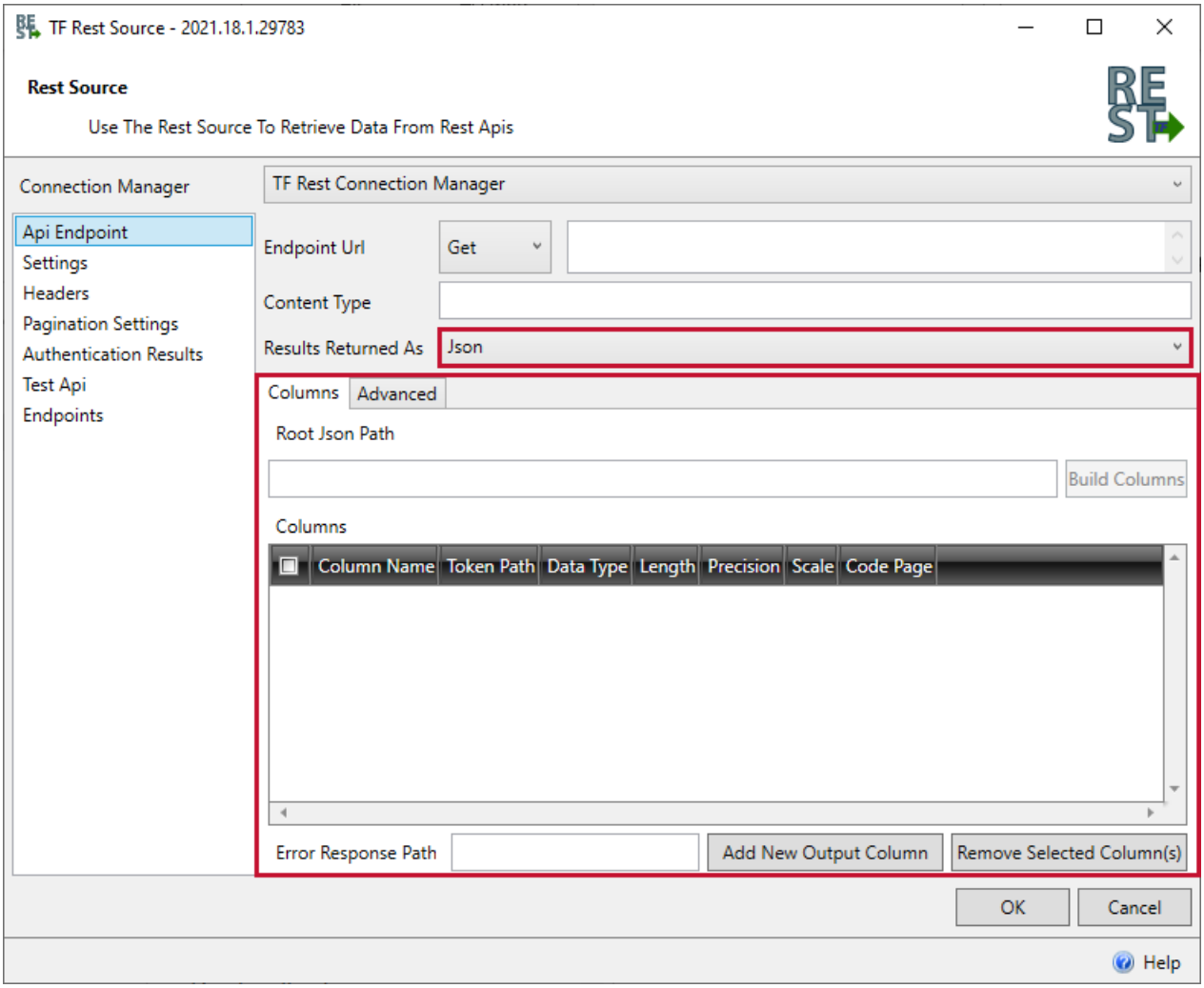
Option	Description

Option	Description
Endpoint URL	In this field, you set the URL of the web service. You should also choose whether you're making a call to receive information (**Get**), or making a request to the service (**Post**). To add a user variable as a parameter to the API Endpoint URL, use the following syntax: **<@Namespace::VariableName>** or **<@Namespace::ParameterName>** **Note:** The angle brackets (**<>**) and the commercial at (**@**) are required in the syntax. Examples • Most variables created in an SSIS package will be a part of the **User** namespace: **<@User::VariableName>** • System variables are accessible using this syntax: ***<@System::StartTime>*** • Both project and package-level parameters are accessible from their respective namespaces: ***<@$Package::ParameterName>*** ***<@$Project::ParameterName>*** Variable or Parameter Name Examples The last part of the syntax will be the variable or parameter name. The syntax is case-sensitive to the name of the variable or parameter. For example, to replace the **API Endpoint URL** with a variable or parameter named **str_URL**: • If the syntax should be replaced by a user created variable, the syntax should be: **<@User::str_URL>** • If the replacement should come from a user created package parameter, the syntax should be: **<@$Package::str_URL>** • If the replacement should come from a user created project parameter, the syntax should be: **<@$Project::str_URL>**

Option	Description
Post/Put/Delete/Patch Data	You set the URL of the web service and choose the method making the call: • **Post** - (Default) Request that the destination URI perform an action with the provided data. • **Get** - Retrieves data from the destination • **Put** - Stores data at a URI and can be used to create a new entity or update an existing one. • **Delete** - Requests that data be removed • **Patch** - Requests an update to the specified fields of an entity at a URI. **Note:** See your application's API page to determine the appropriate action **Additional Information:** For more information about these Rest options, see the following guide.
Content Type	Identifies the type of content that is requested from the API.
Results Returned As	Selects the format for the results: XML, String, or JSON.

JSON, String, and XML Properties

JSON Properties Configuration



Option	Description
Root JSON Path	JSON queries can return multiple levels, therefore, this field Identifies the root to be used.
Output Columns	In this window, you can add and remove columns, define their name, token path (defined within the raw data), data type, length, precision, scale, and code page.

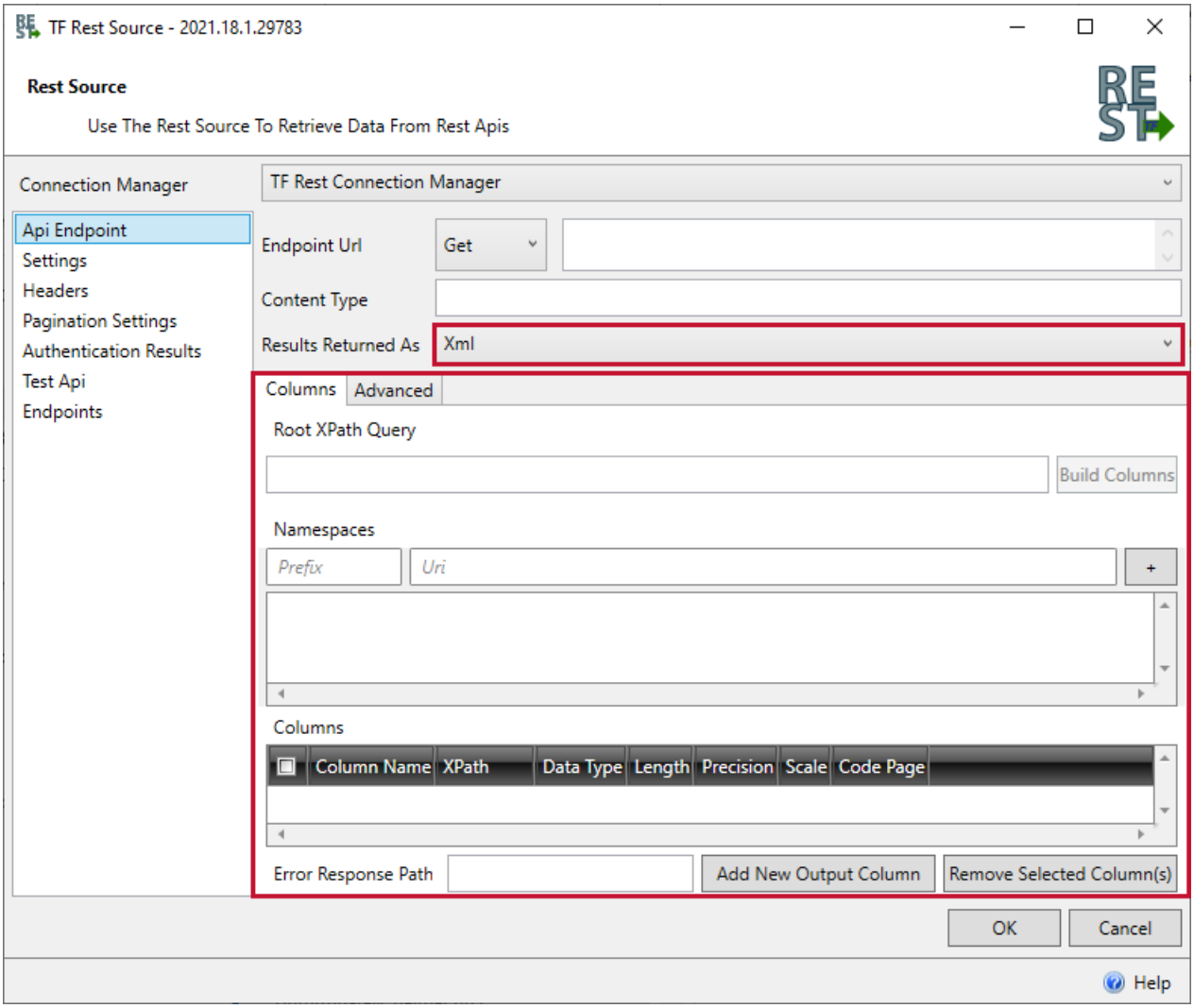
Using Wildcards (*)

The Rest Source allows for wildcards when arrays are returned. Example: Using the JSON Results below, the Root JSON Path would be results[0].appInventory[*]. This information tells the component to use the array found at position **0**. Because some child arrays do not have an object name, a wildcard or ***** can be used to return all child array objects.

```
{
  "results": [
    {
      "appInventory": [
        {
          "name": "Sample name 1",
          "identifier": "123",
        },
        {
          "name": "Sample name 2",
          "identifier": "456",
        },
        {
          "name": "Sample name 3",
          "identifier": "789",
        }
                ]
        }
    ]
}
```

Important: Columns that use DT_STR and DT_WSTR return empty string values when the data returns no value. Columns that contain other data types convert the empty string values to NULL.

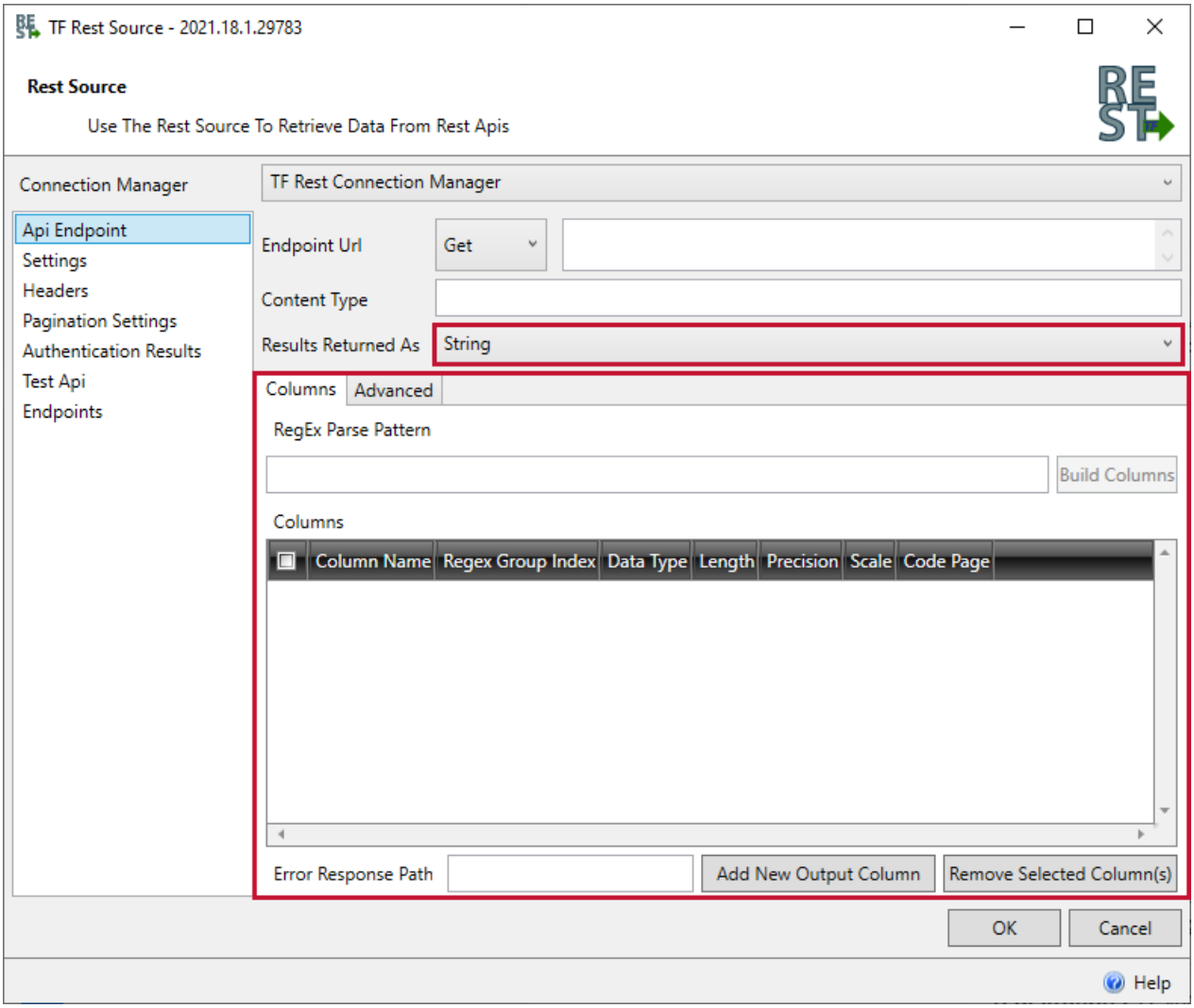
XML Properties Configuration



Important: Columns that use DT_STR and DT_WSTR return empty string values when the data returns no value. Columns that contain other data types convert the empty string values to NULL.

Option	Description
Root XPath Query	XML and JSON queries can return multiple levels, therefore, this field Identifies the root to be used.
Output Columns	In this window, you can add and remove columns, define their name, token path (defined within the raw data), data type, length, precision, scale, and code page.

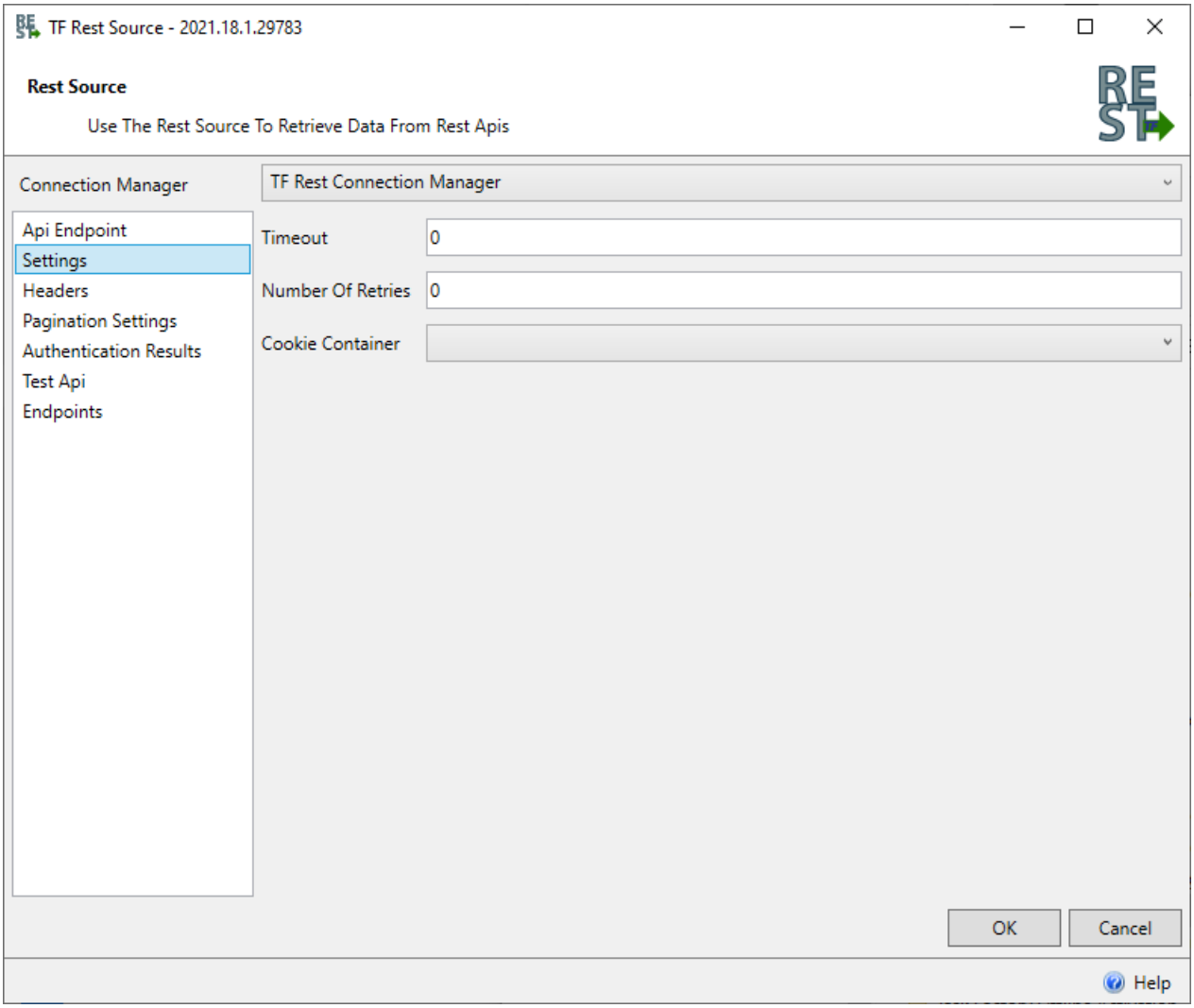
String Configuration Properties



Important: Columns that use DT_STR and DT_WSTR return empty string values when the data returns no value. Columns that contain other data types convert the empty string values to NULL.

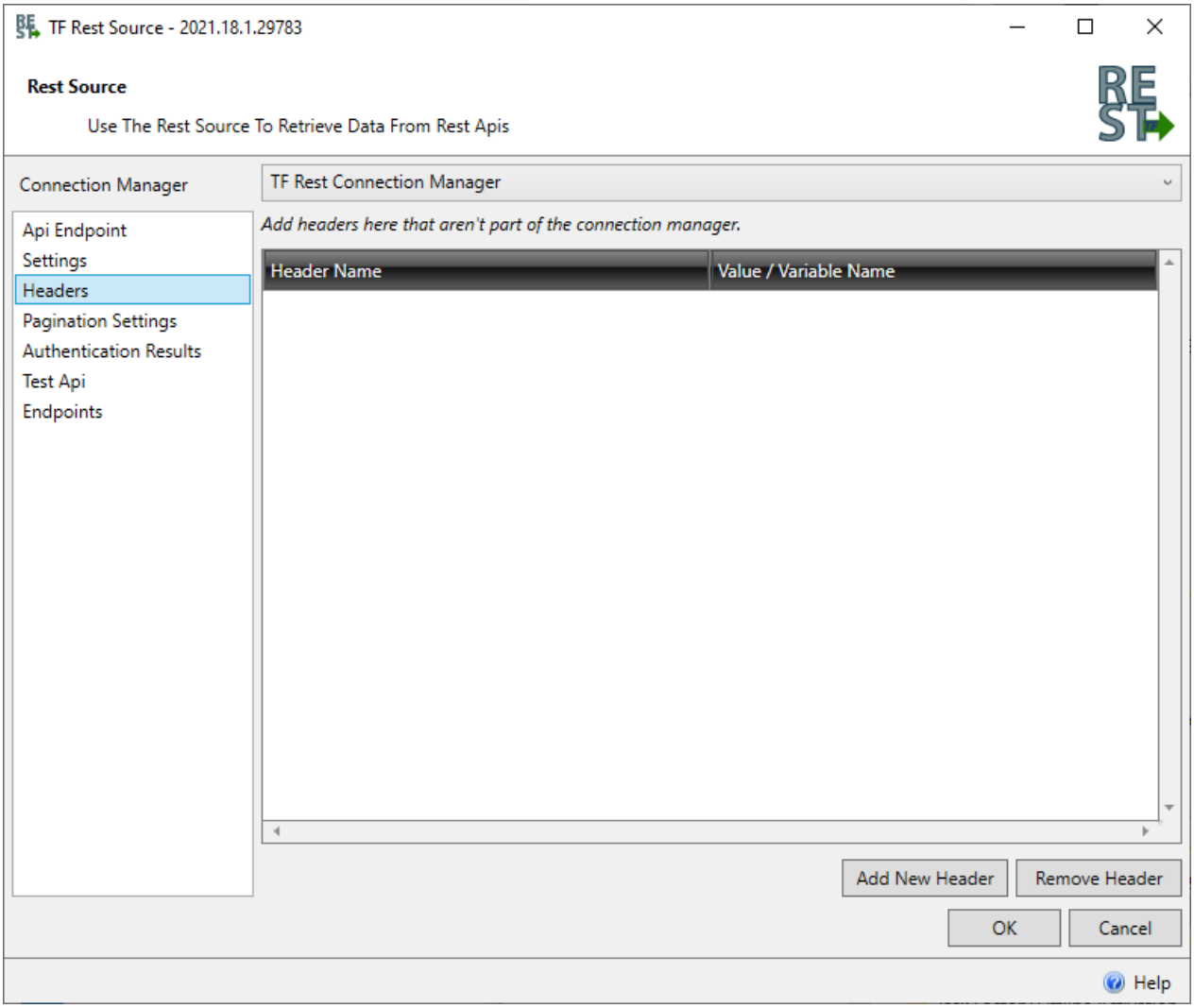
Option	Description
RegEx Parse Pattern	The regular expression used to parse the returned JSON. Leaving this field blank returns all data from the **0** index and only one row with this information returns.
Output Columns	In this window, you can add and remove columns, define their name, token path (defined within the raw data), data type, length, precision, scale, and code page.

Settings



Option	Description
Timeout	Sets the amount of time (in seconds) the component should fail if there is no response from the web service. **Note:** The default value for this option is 0, meaning the connection will not timeout.
Number of Retries	Indicates the number of times to retry connecting to the API before the component fails. **Note:** The default value for this option is 0, meaning the connection attempt will not be reattempted.
Cookie Container	Used to identify the variable that stores a returned cookie response. The variable must be of object datatype. **Note:** If your using cookie authentication, link you'll need to setup a data flow to capture the cookie with a request in the rest source. In a subsequent dataflow, the cookie will populate into the object variable, and it can be set with the object variable selected.

Headers (Rest Source)



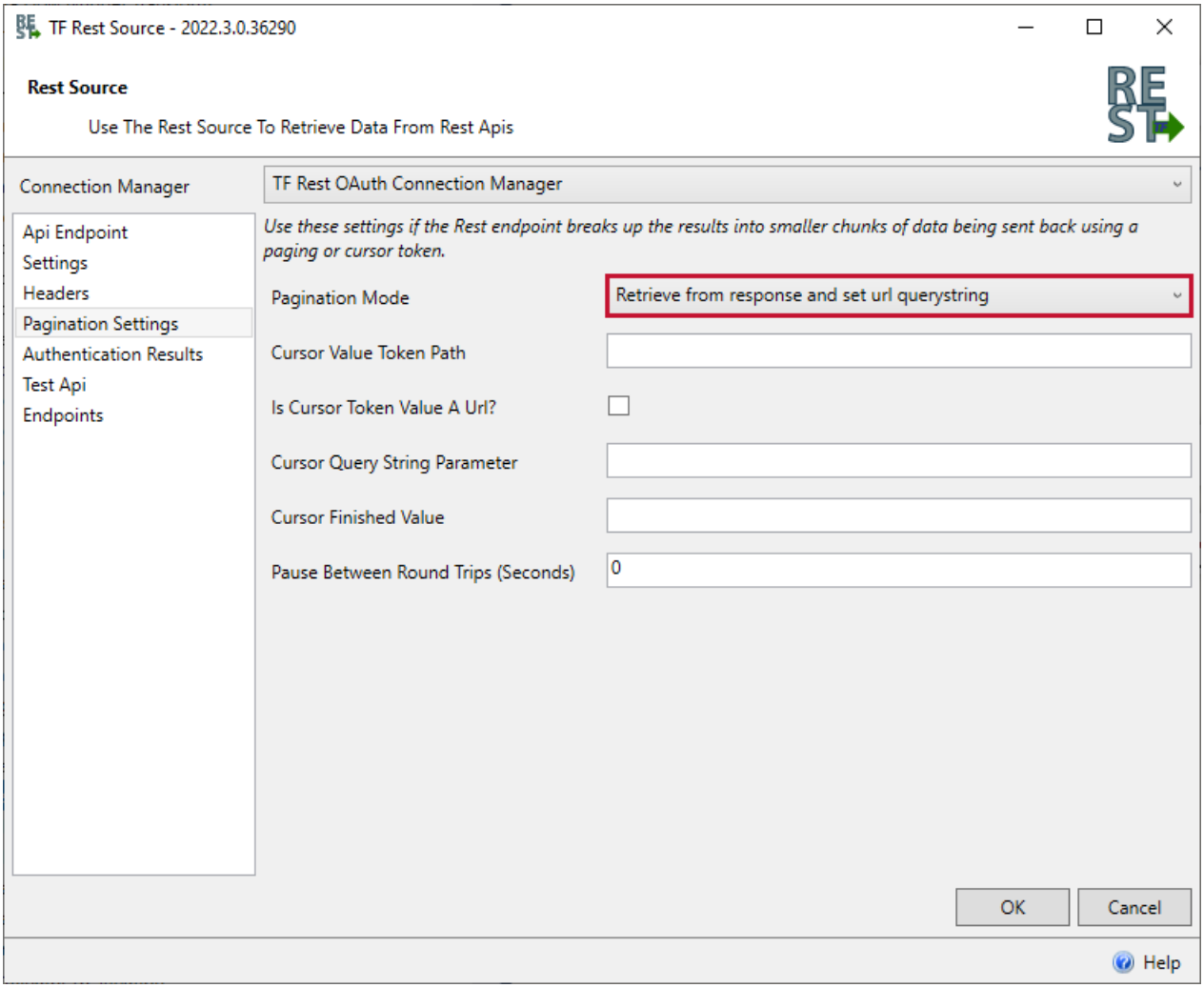
Headers are simply information about the type of data returned. You can add and remove headers by selecting the corresponding buttons. Once a header is added, select in its **Name** and **Variable Name** fields to configure them. Headers Values can be replaced by variables using the following syntax:

- < @User::NameOfVariable >
- < User::NameOfVariable >

Pagination Settings

Pagination Mode Menu Options

Retrieve from response and set URL querystring



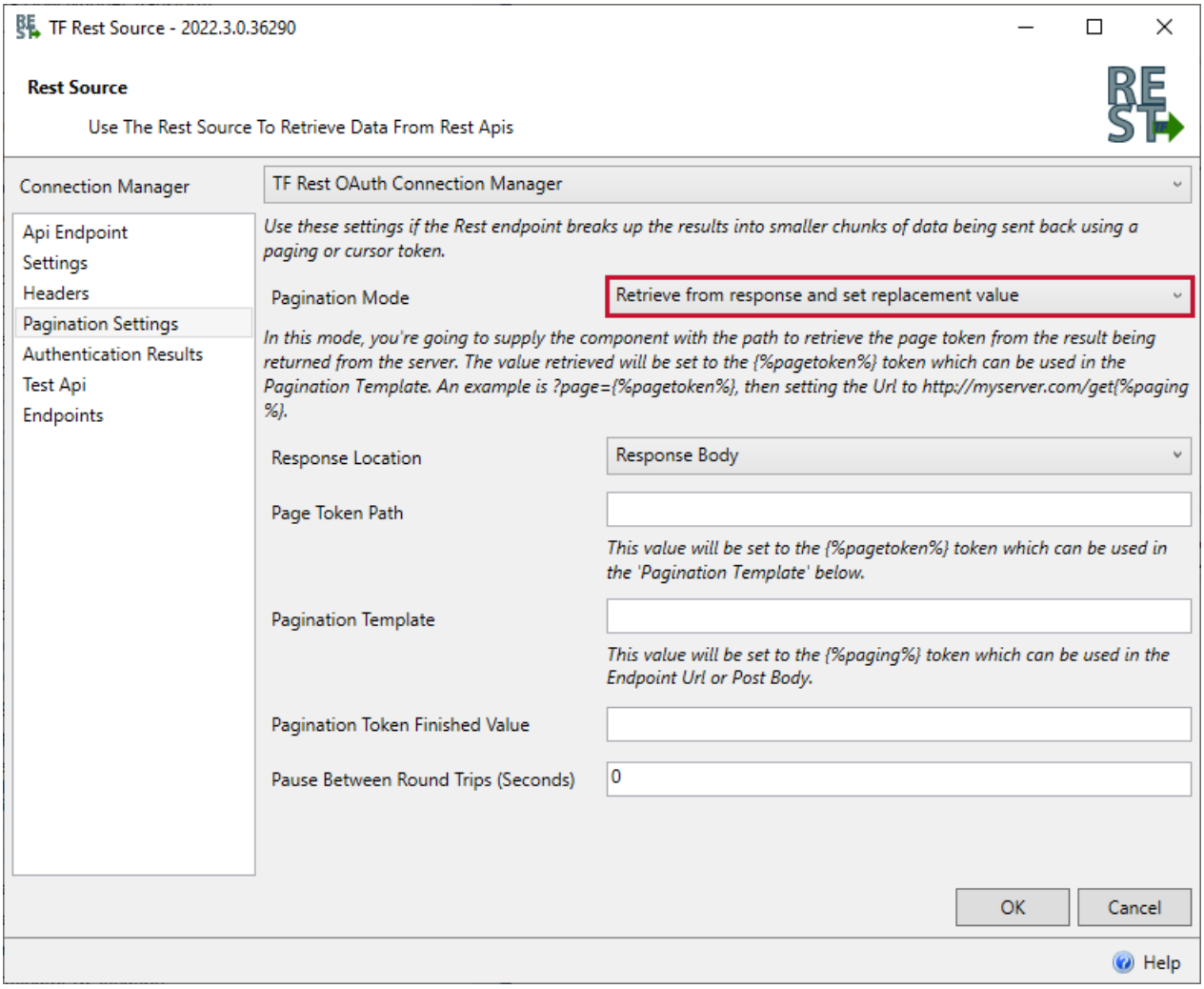
Option	Description

Cursor Value Token Path	The token path that returns the page number, or the place where the set of results is located (page number, url, page token, etc.) for example:
	Codeblock
	{
	"pagingToken":"12345",
	"results":{"name":"Sentry One"}
	}
	The token path in this instance is **pagingToken**.

Is Cursor Token Value A Url	Select this option if the cursor token value is a Url.

Option	Description
Cursor Query String Parameter	The query string parameter is appended to each round trip (after the first) to the API Endpoint URL. Example: Creating the parameter pagingToken appends it to the end of the Url such as the following: **http://webservice/endpoint?pagingToken=12345.**)
Cursor Finished Value	The value that the token path defined at cursor value token path that displays when paging has finished. For example, if paging token 12345 ends with a 0, you would enter 0 in this spot. **Note:** This can be left blank if the endpoint omits the results token when it's finished.
Pause Between Round Trips (Seconds)	The number of seconds between calls to the application.

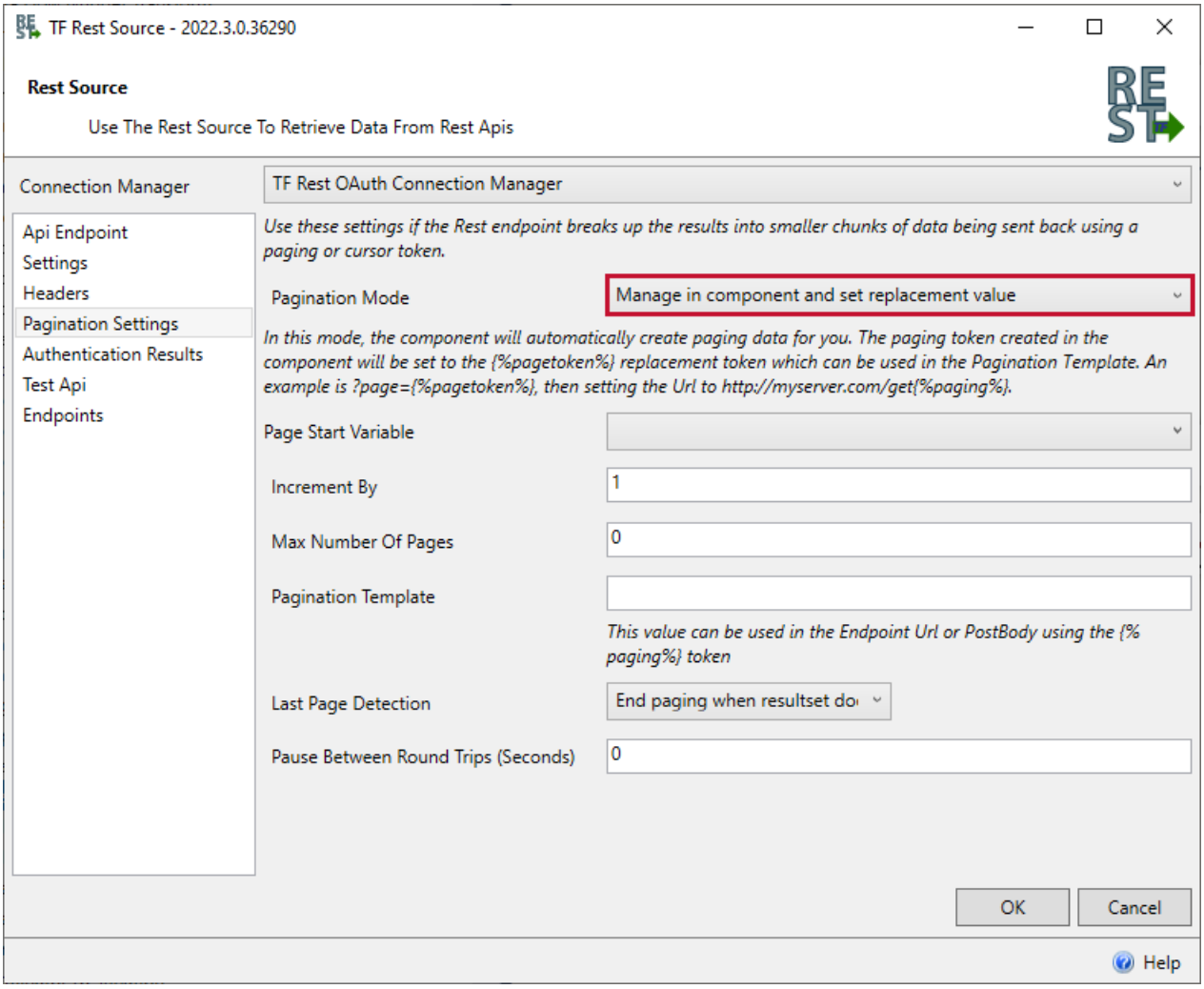
Retrieve from response and set replacement value



Option	Description

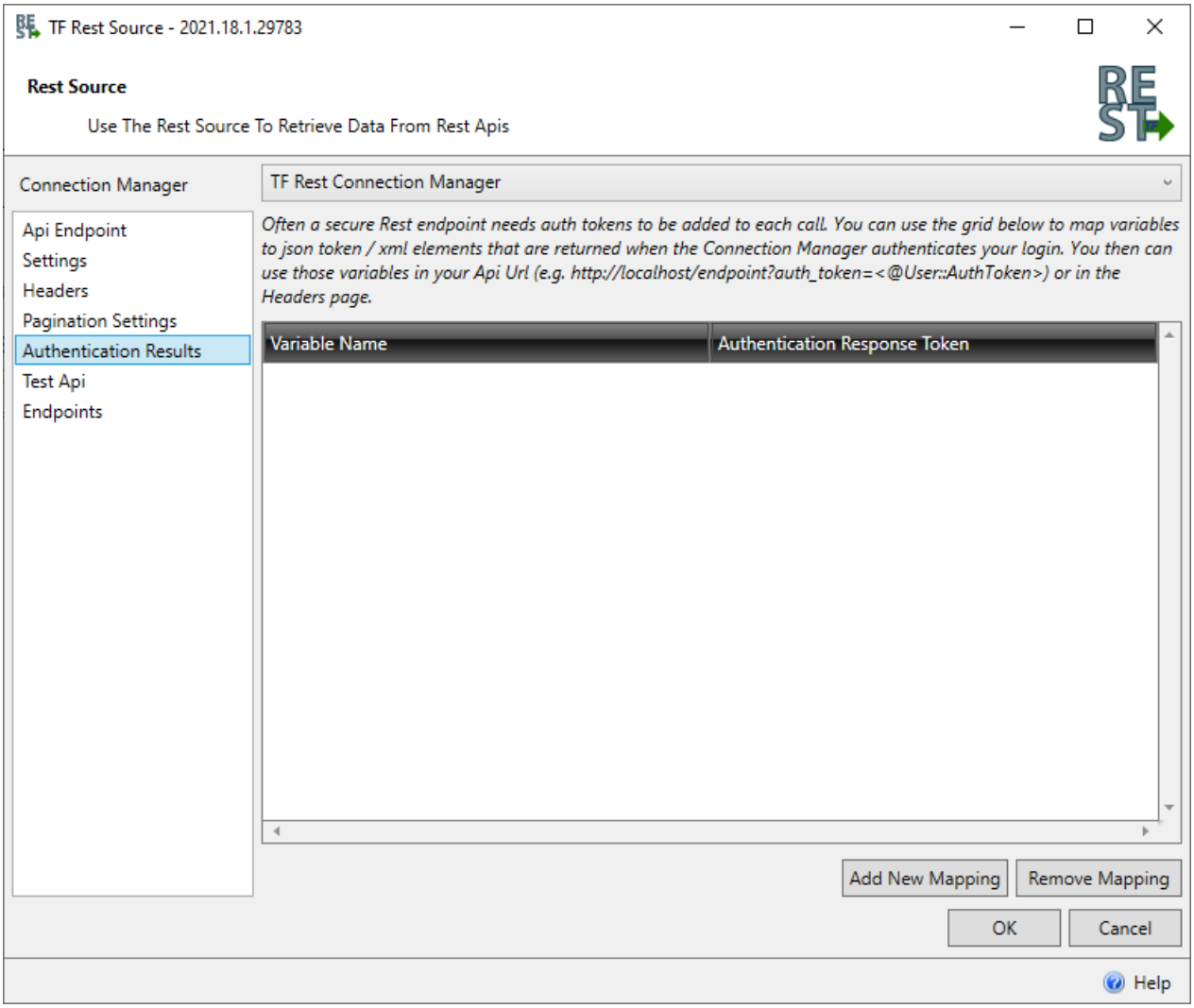
Option	Description
Response Location	Select whether the response location will be **Response Body** or **Response Header**.
Page Token Path	The token path for the paging token. Example: **pagingToken**
Pagination Template	Sets the template that can be replaced in the Endpoint Url or Post Body. Example: **?value={%pageToken%}**
Pagination Token Finished Value	Identifies the final value. **Note:** This value varies by your API.
Pause Between Round Trips (Seconds)	The number of seconds between calls to the application.

Manage in component and set replacement value



Option	Description
Page Start Variable	The record number where you want to start. ℹ **Note:** Generally, this value is **1**.
Increment By	The number of items per page.
Max Number Of Pages	The total number of pages. ℹ **Note:** This value isn't dynamic, so **0** can't be entered as a value.
Pagination Template	The value appended to the call and stored in the **{%paging%}** variable which is added to the Endpoint.
Last Page Detection	Identifies the method the endpoint uses : • End paging when status code is • End paging when response contains error message • End paging when resultset doesn't contain any rows
Pause Between Round Trips (Seconds)	The number of seconds between calls to the application.

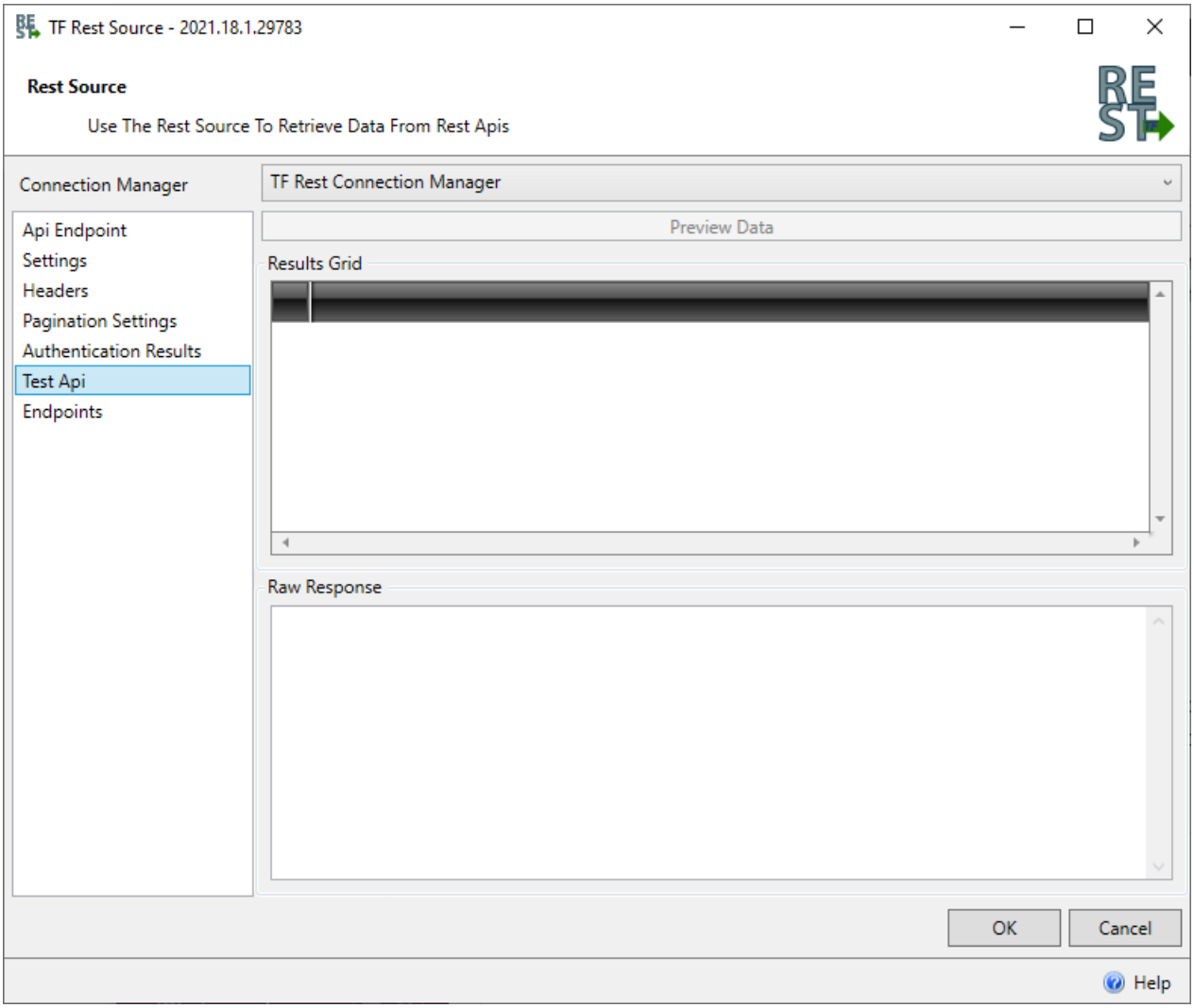
Authentication Results (Rest Source)



If you need a bearer token sent every call to the Rest API, you'll set the bearer token up here. Create a variable to dynamically set the authentication response token by selecting the **Add New Mapping** button.

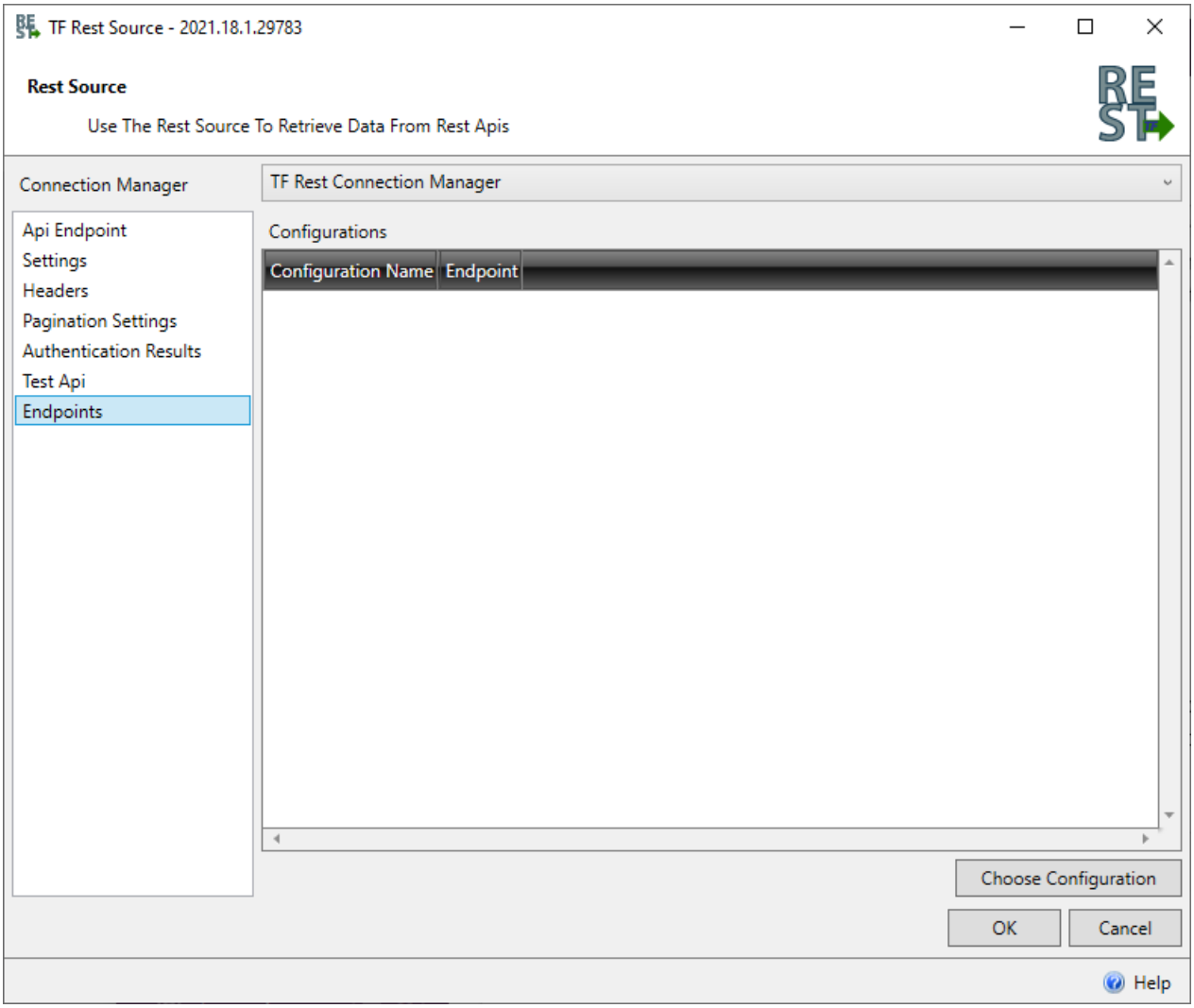
Additional Information: For more information about setting up a bearer token, see the following support article.

Test API (Rest Source)



Once properly configured, you can select the **Preview Data** button to see an output sample as well as return raw XML/JSON should you need additional information to help configure the component further.

Endpoints (Rest Source)

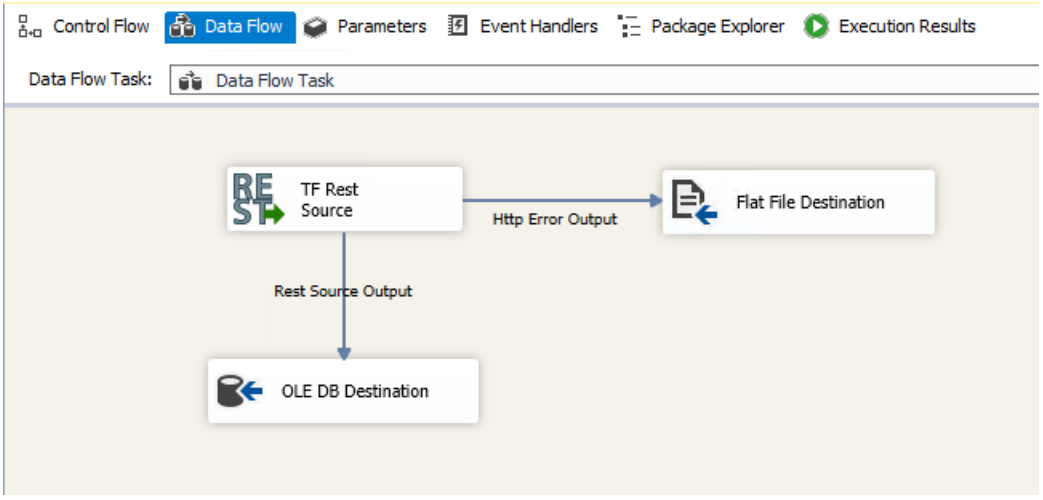


Option	Description
Choose Configuration	The endpoints that are selectable for this section are configured in your config file. **Note:** The Google analytics config file provides a good example of these parameters. You can access this config file at the following file path: **C:\%Program Files (x86)%\SentryOne\Task Factory\OAuth2ConfigFiles** **Additional Information:** For a tutorial explaining configuration files for the Rest OAuth2 connection manager, see the following video.

HTTP Error Output

The rest source can capture error responses with the **Http Error Output**. You are asked to choose between the Rest Source Output and Http Error Output when connecting downstream components.

Note: An execution error causes a package to fail unless the Http Error Output is attached.

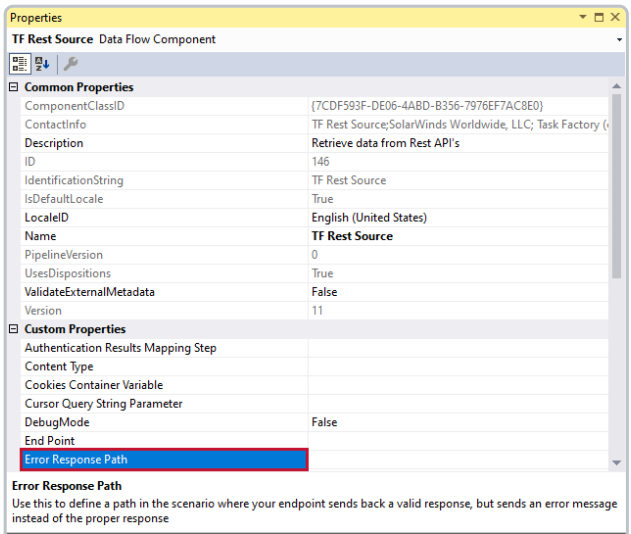


Error Response Path Property

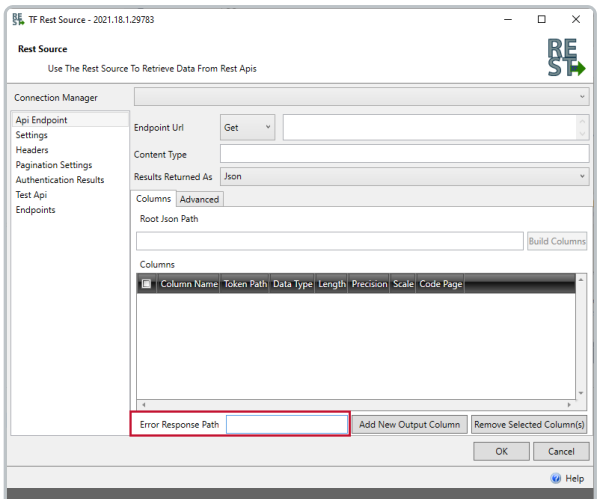
The rest source depends on failure HTTP status codes. Codes such as 401, 404, and 500 are used to detect if an error has occurred during execution. If an API doesn't return failure HTTP status codes when an error occurs, you can tell the component what to look for. This allows the component to fail with an error.

For example: a user makes a call to an API and it returns a valid status code of 200. If the JSON response is **{ "response" : { "error":"Invalid auth token" } }**, you can set the Error Response Path to **response.error**. The component detects the error response and throws an exception that can be handled. At this point, the component throws an exception and does not send the error to an error output.

This only applies to endpoints that return an HTTP status code of 200 through 208, instead of using a failure code such as 401 or 404.



Rest Source Error Response Path in properties

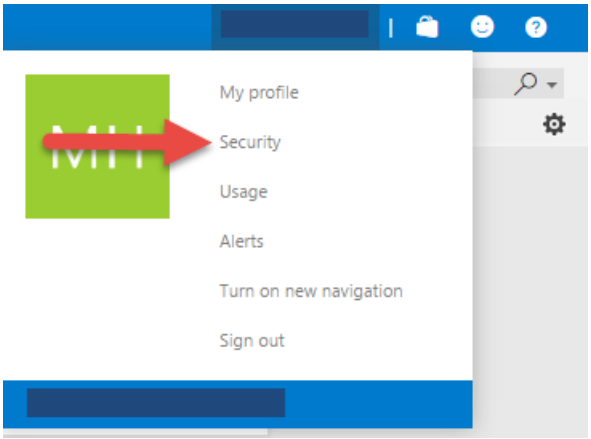


Rest Source Error Response Path in component

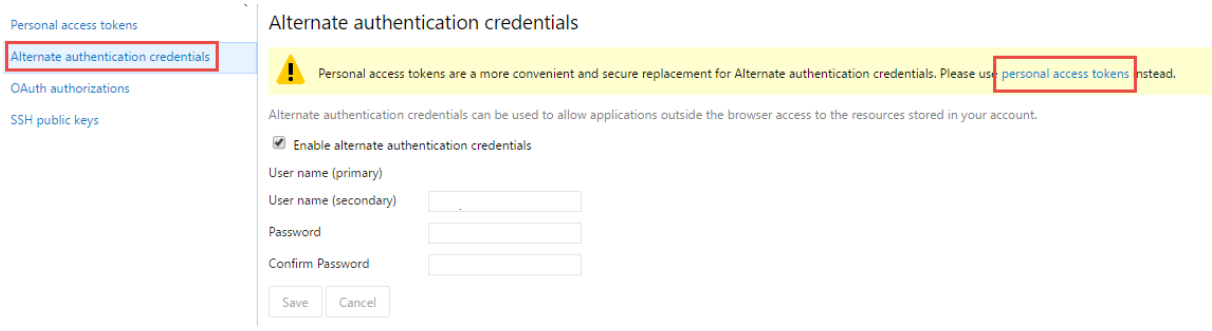
Configuring Rest Source for Visual Studio Team System (VSTS)

Configure Rest Source for Visual Studio Team System by completing the following steps:

1. Before configuring the Rest Source, you need to set up alternative credentials in VSTS. This can be done by accessing the Security Settings for the account.

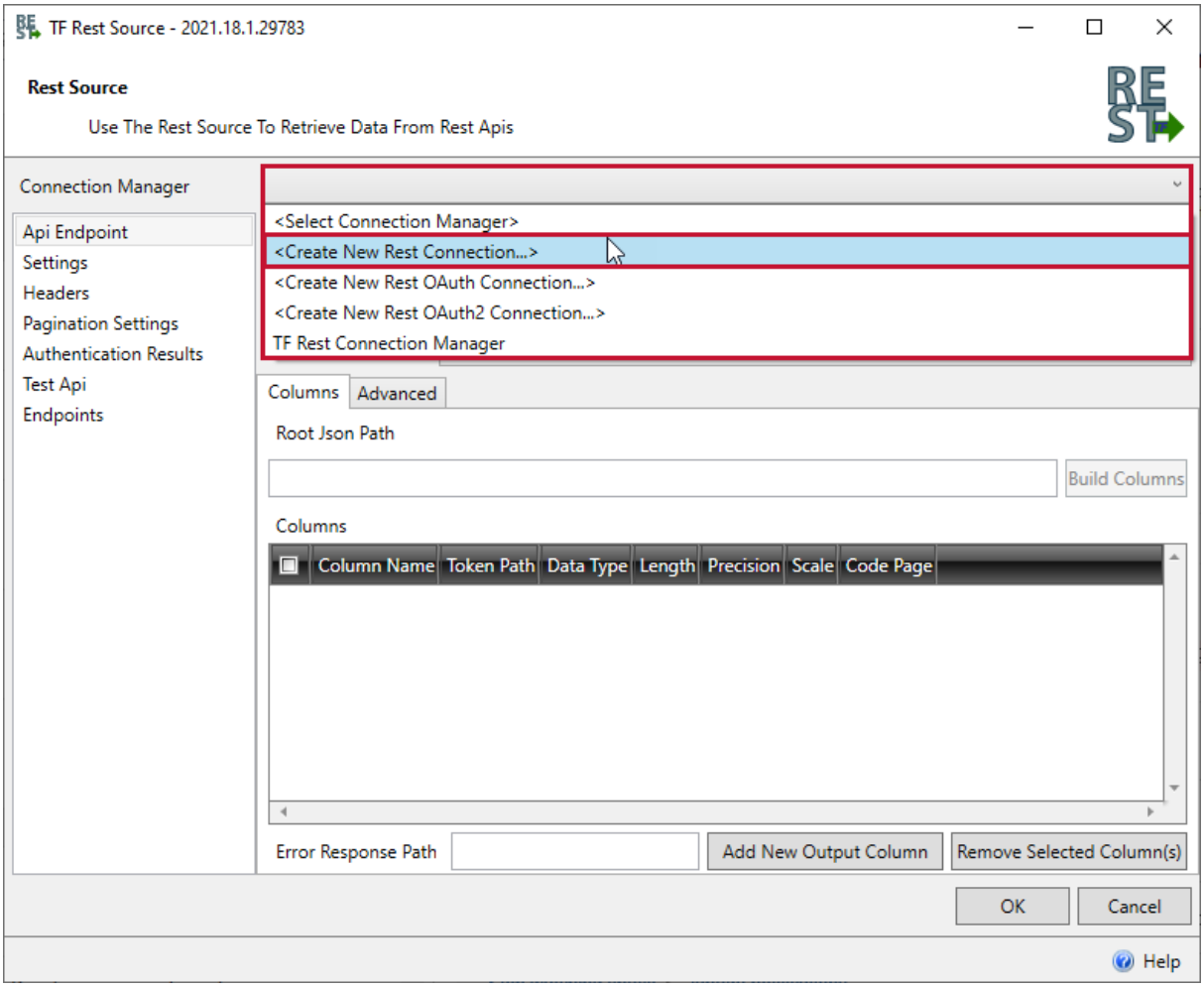


2. Select **Alternate authentication credentials**, and then set up the username and password:

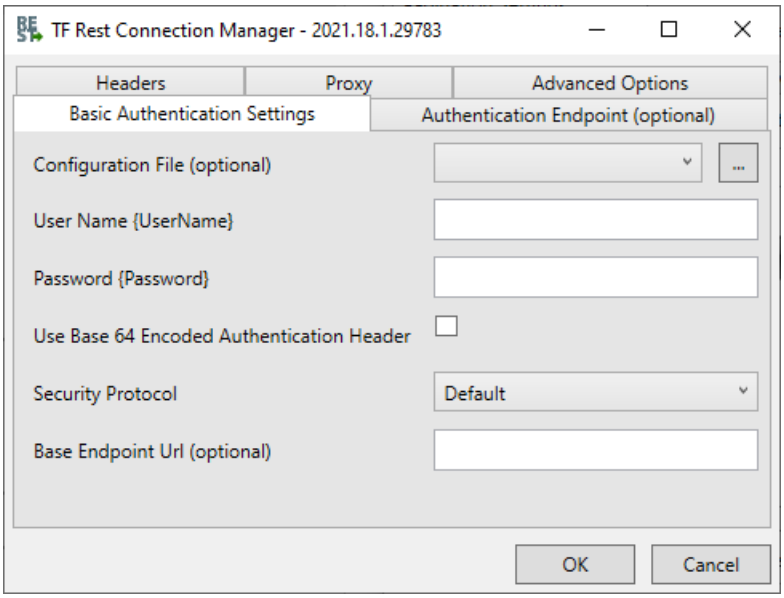


Alternatively, the **personal access tokens** (highlighted above) can be used as the password in the REST Source.

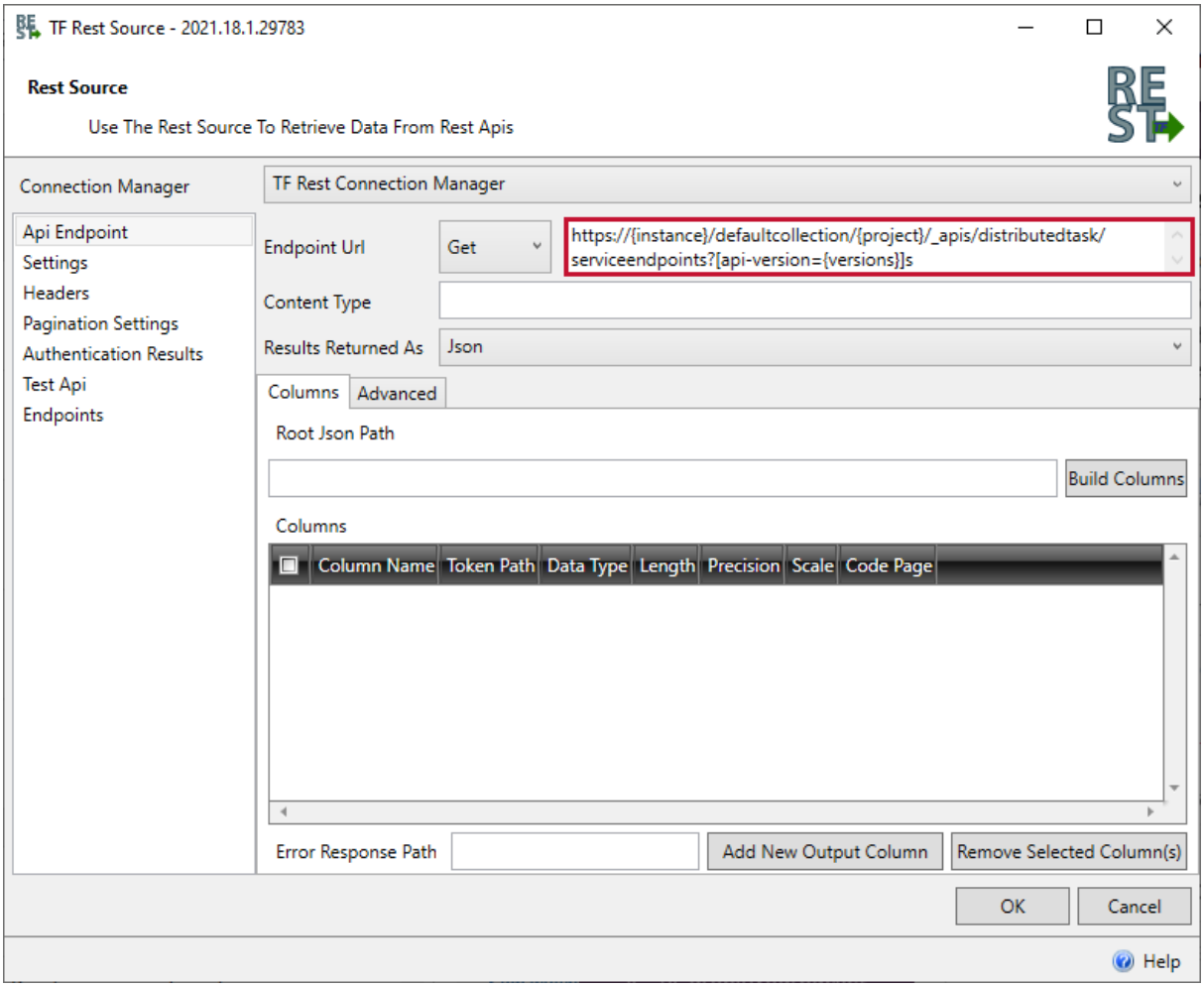
3. Open the Rest Source component.
4. Create a new Basic Authentication connection manager:



5. Enter the username and password in the connection manager.



6. Close the connection manager and configure the Endpoint Url. Please consult VSTS's API documentation for endpoint information.

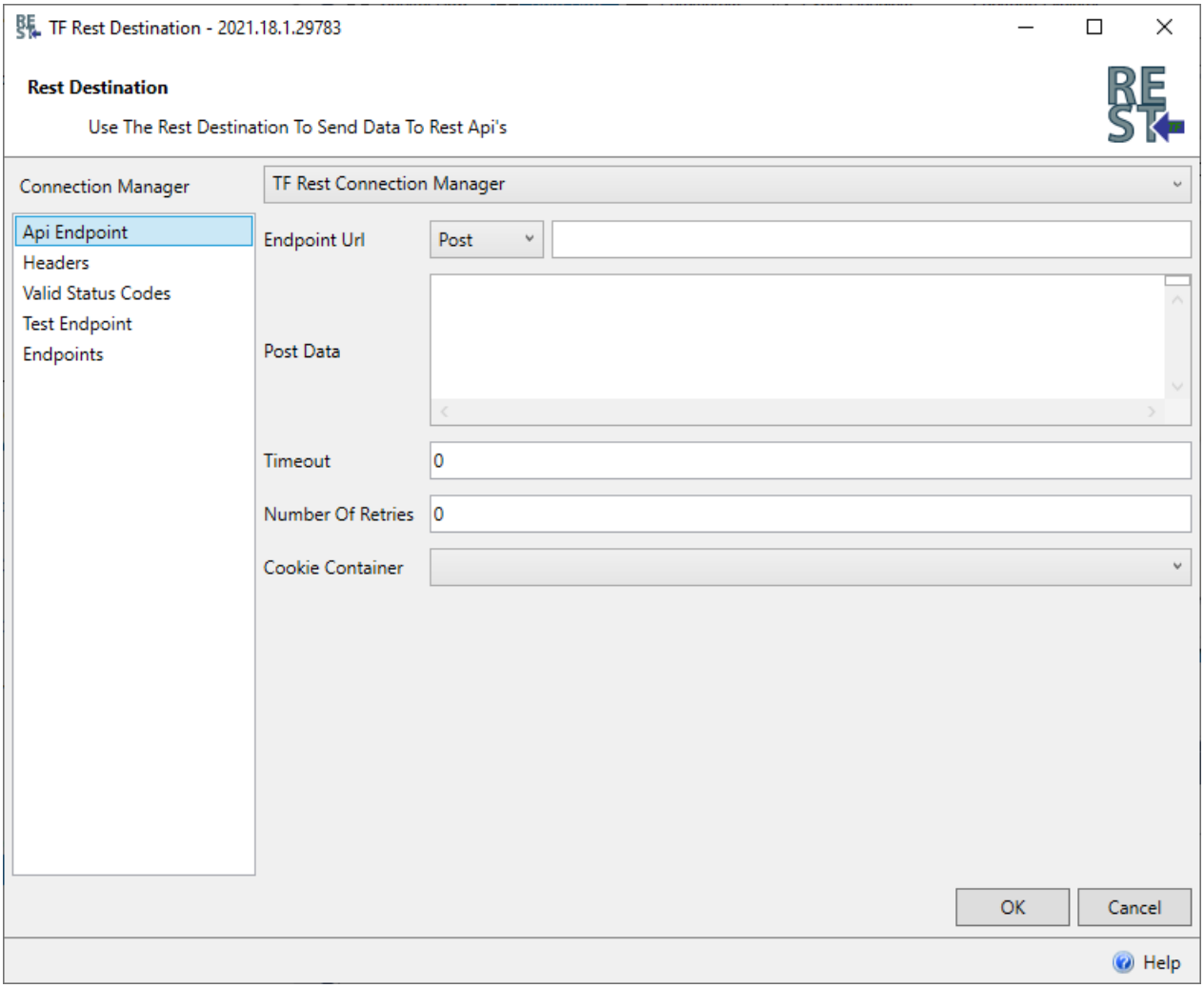


REST Destination

Destination Icon	Destination Description

Destination Icon	Destination Description
	REST Destination enables you to connect to a web service that utilizes a REST API and send data to it. See the Rest Source Connection Manager, Rest Source OAuth Connection Manager, and the Rest Source OAuth2 Connection Manager to learn more about configuring the component's connection manager.

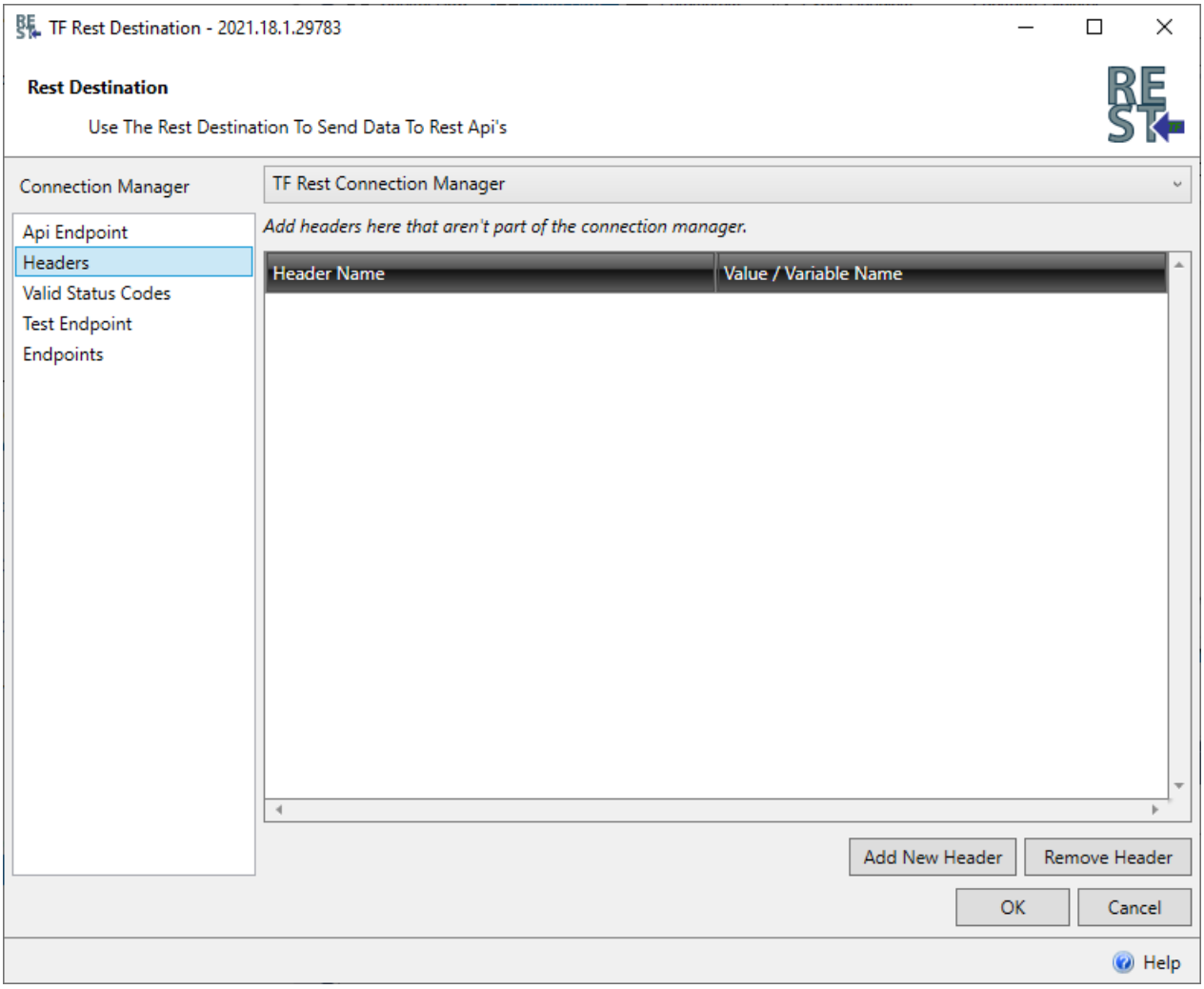
API Endpoint



Option	Description

Option	Description
Endpoint URL	You set the URL of the web service and choose the method making the call: • **Post** - (Default) Request that the destination URI perform an action with the provided data. • **Get** - Retrieves data from the destination • **Put** - Stores data at a URI and can be used to create a new entity or update an existing one. • **Delete** - Requests that data be removed • **Patch** - Requests an update to the specified fields of an entity at a URI. ⚠ **Important:** The Endpoint URL and Post Data uses auto-complete for source columns and variables. Source columns are surrounded in curly braces **{{ColumnName}}**. Variables use the following format: **<@User::MyVariable>**.
Post Data	The user defined data used in the post request.
Timeout	Sets the number of seconds the component should fail if there is no response from the web service.
Number of Retries	Indicates the number of times to retry connecting to the API before the component fails.
Cookie Container	Used to identify the variable that stores a returned cookie response. The variable must be of object datatype.

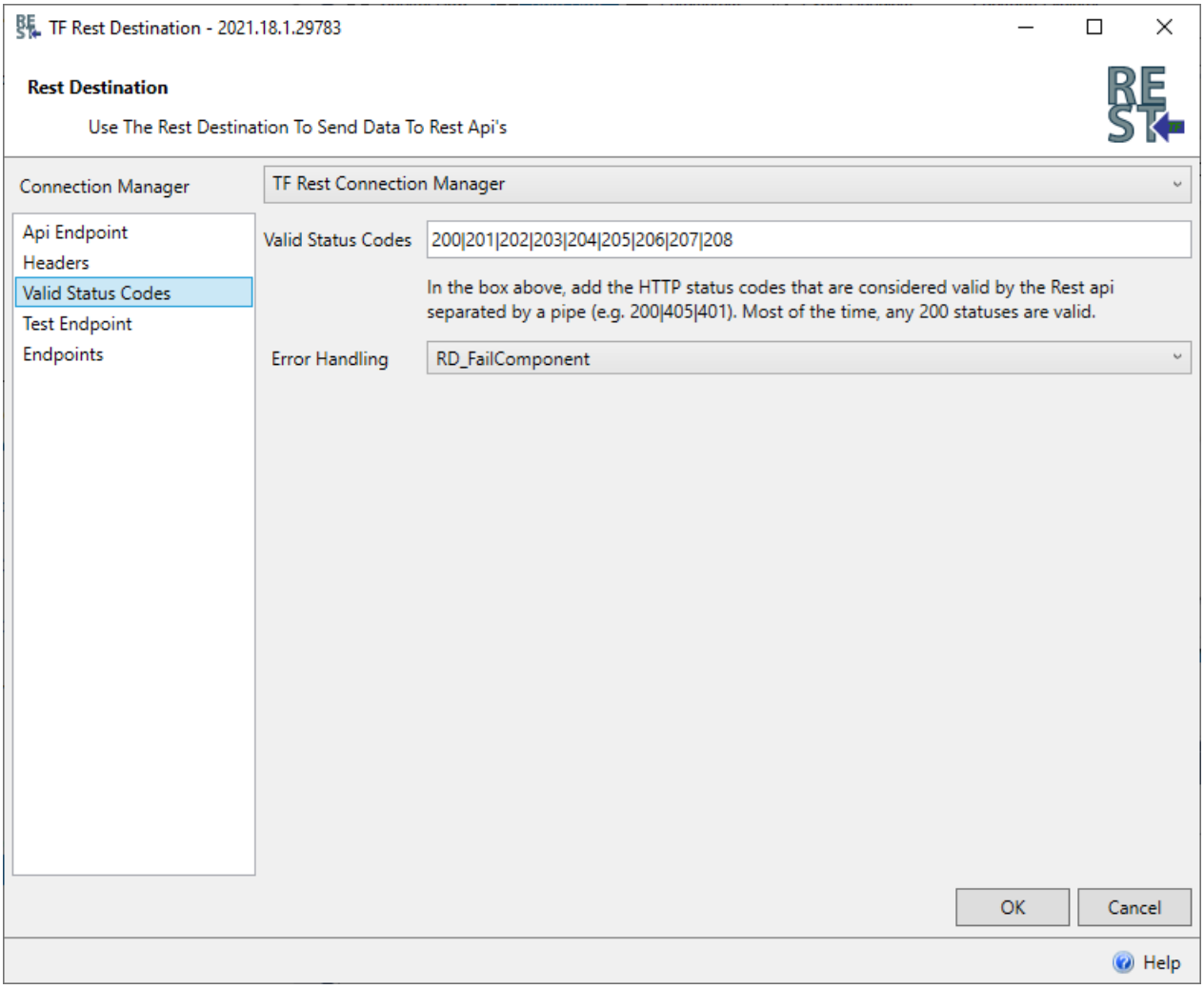
Headers (Rest Destination)



Headers are information about the type of data returned. You can add and remove headers by selecting the corresponding buttons. Select the header's **Name** and **Variable Name fields** to configure them after they are added.

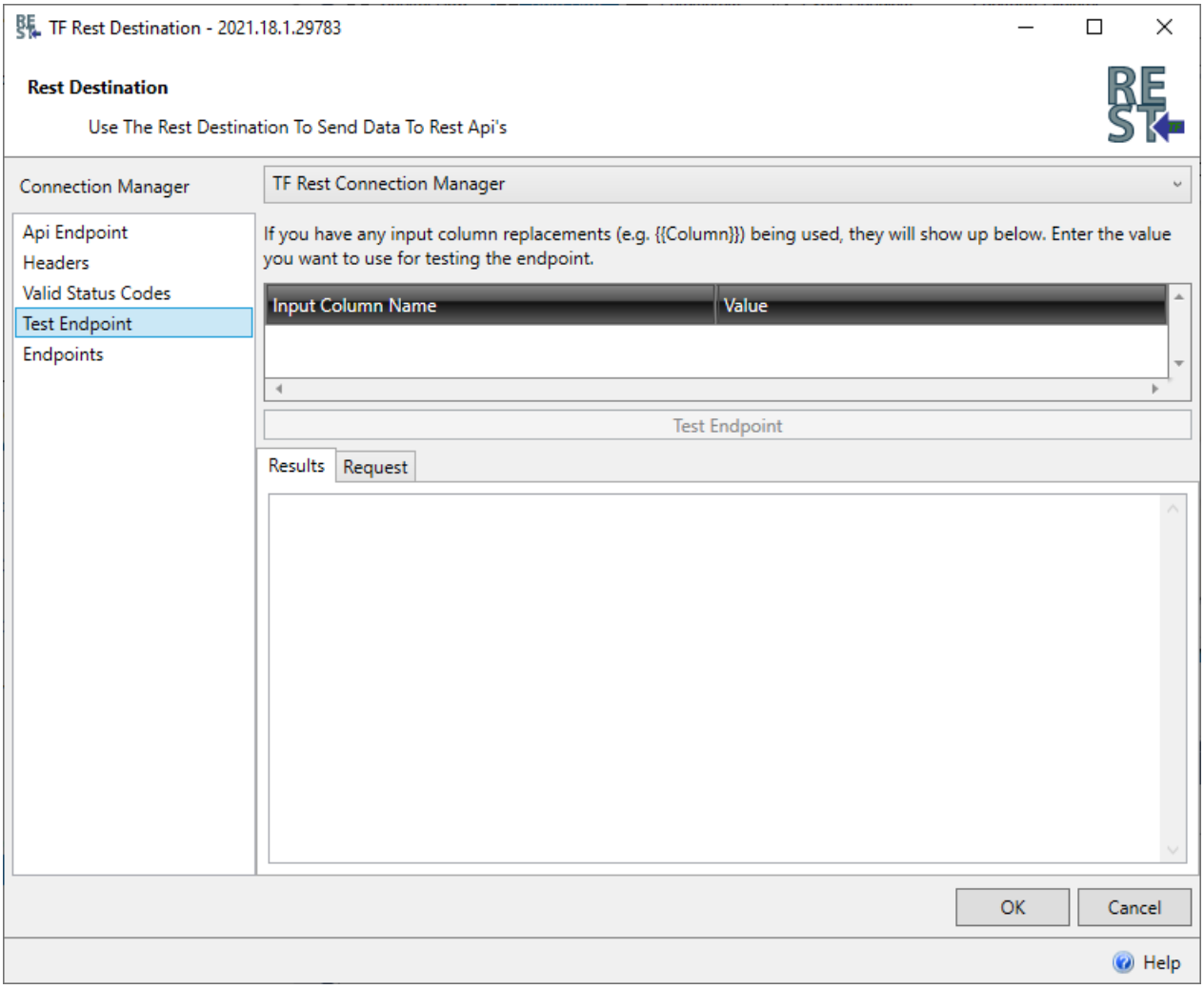
> **Note:** Source columns and variables can be used as header values. Columns are surrounded in curly braces **{{ColumnName}}**. Variables use the following format: **<@User::MyVariable>**.

Valid Status Codes



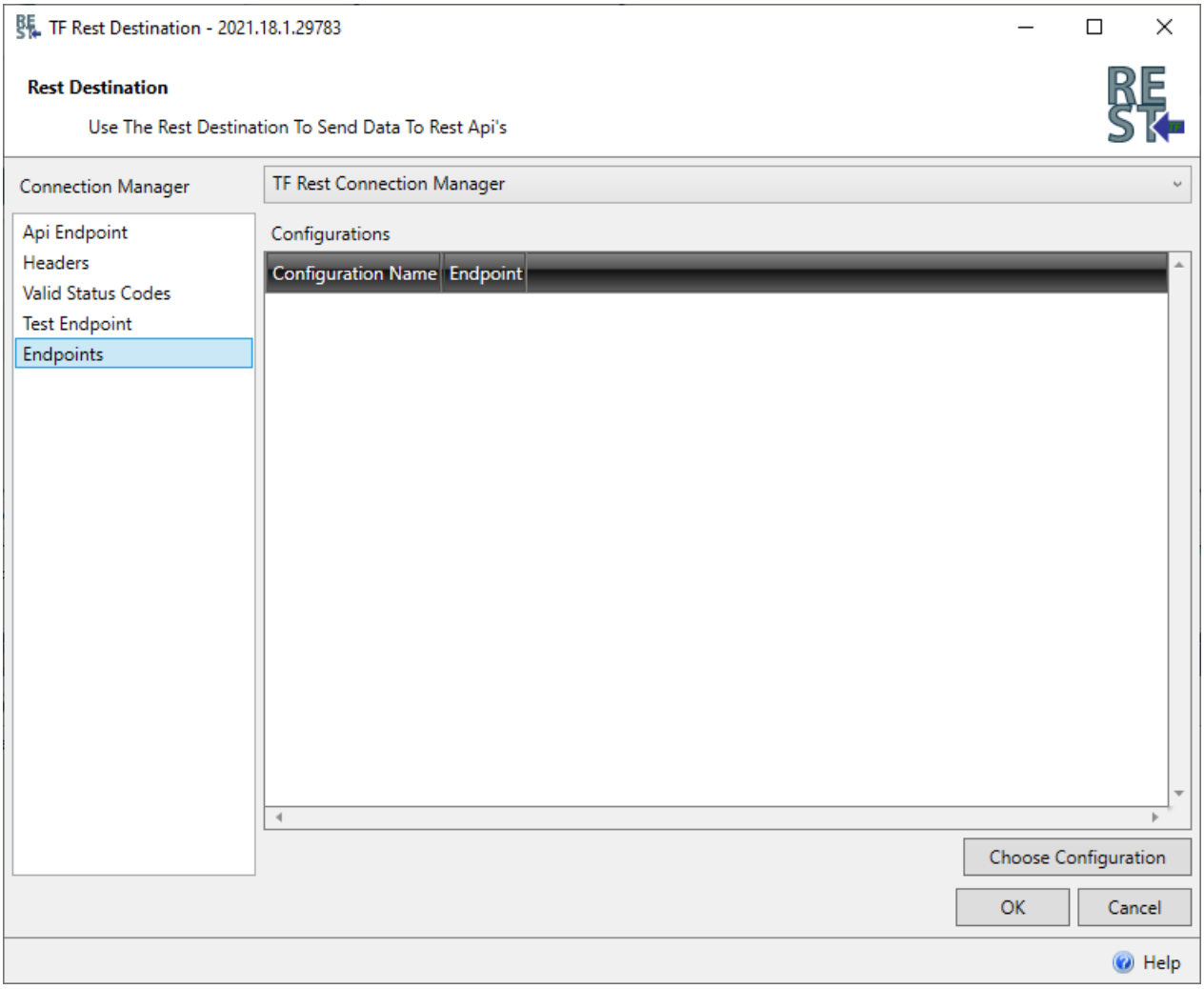
Option	Description
Valid Status Codes	You can add the code to report when a valid request is made. When several status codes are used, they should be separated with the **|** ("pipe") character. **Note:** In most cases, any 200 statuses are valid.
Error Handling	• **Not Used** - Error handling is not used and no error constraint is available. • **Ignore Failure** - All errors are ignored and the package continues to execute. • **Redirect Row** - All errors are directed to an error output. • **Fail Component** - (Default) On error, the component fails execution and stops the package.

Test Endpoint (Rest Destination)



Option	Description
Test Endpoint button	The **Test Endpoint** button produces a sample output based on the user's configuration. Values can be substituted by entering them in the table above the **Test Endpoint** button. **Note:** This is a real call to the API with the data provided. Input Column Names cannot be changed.
Request tab	The **Results** tab displays the data returned in the request. The **Request** tab displays the actual request made to the endpoint.

Endpoints (Rest Destination)



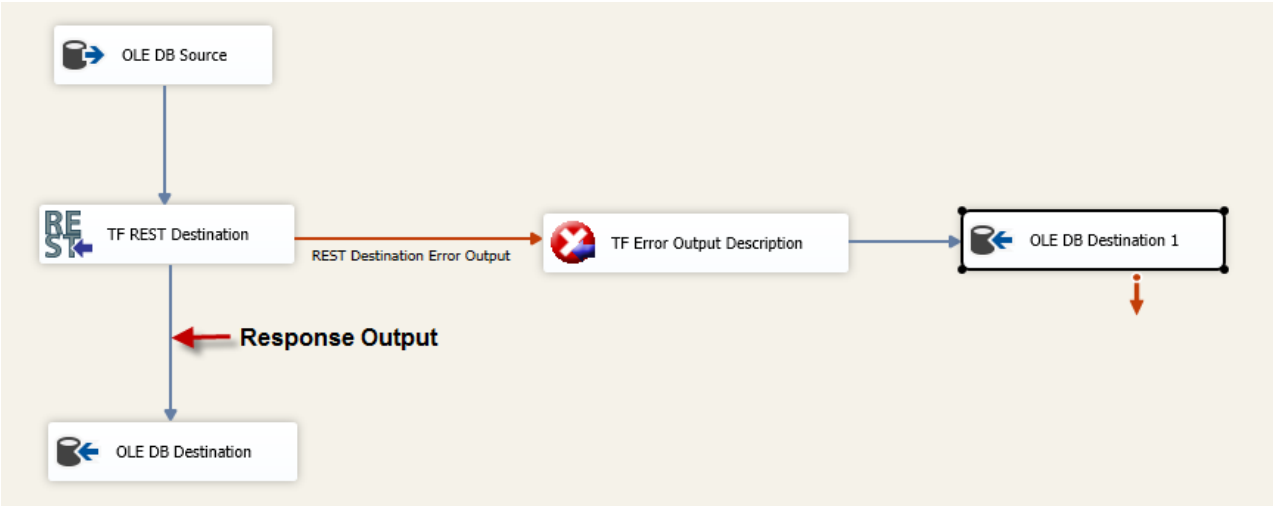
Option	Description
Choose Configuration	This view displays the configuration name and endpoint value. This is generated from the configuration file chosen in an OAuth2 connection manager. Parameters are configured by selecting the endpoint and selecting the **Choose Configuration** button. (This opens a parameter window.)

Sending Files to a Rest Destination

You can send files with the Rest Destination. The column must use **DT_BYTES or DT_IMAGE** data types. Additionally, only the column that contains the file can be included in the POST body.

Rest Destination Response Output

The Rest Destination uses the **successful** output to capture responses from the service.

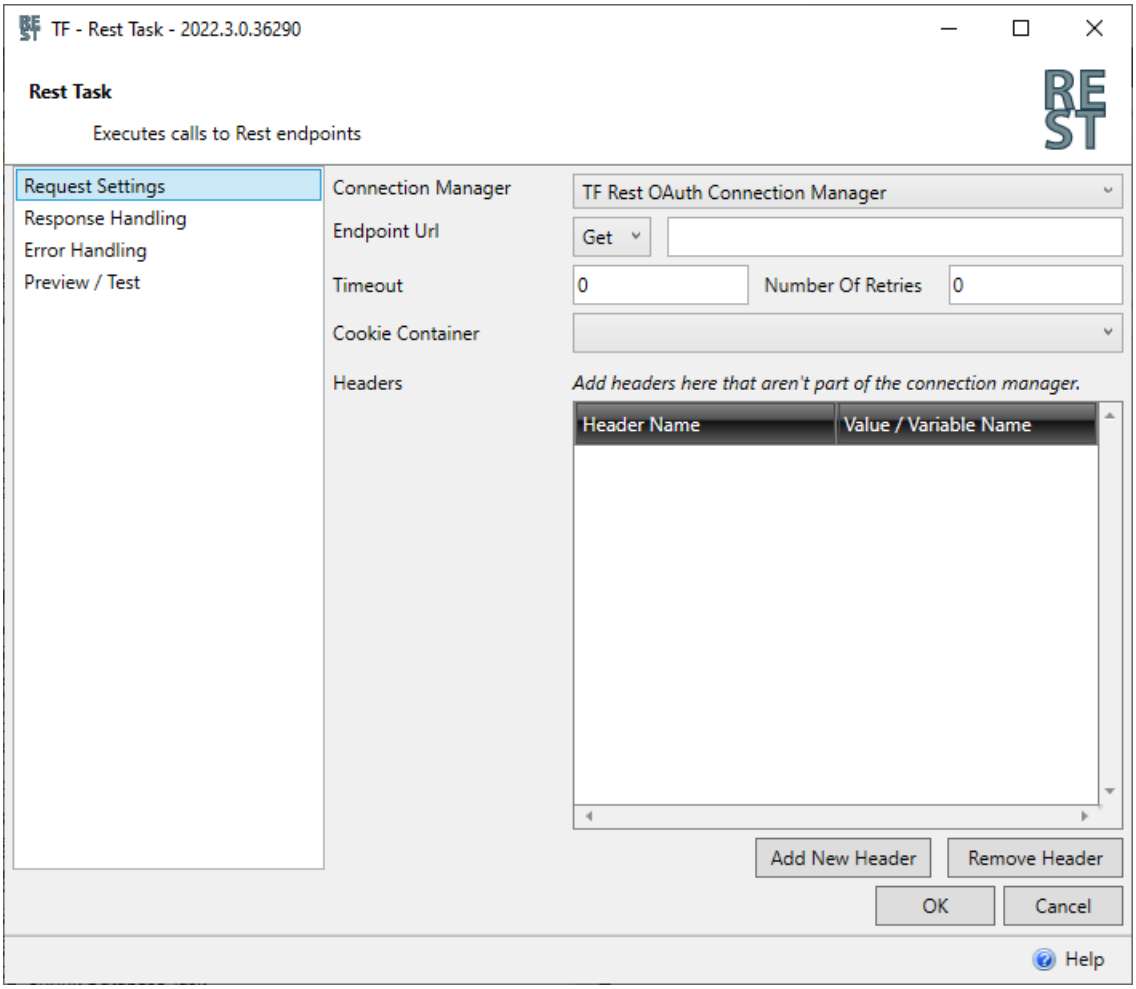


Output responses are output as NTEXT data types. To view the response in a data viewer, you need to use SSIS's native Data Conversion component.

REST Task

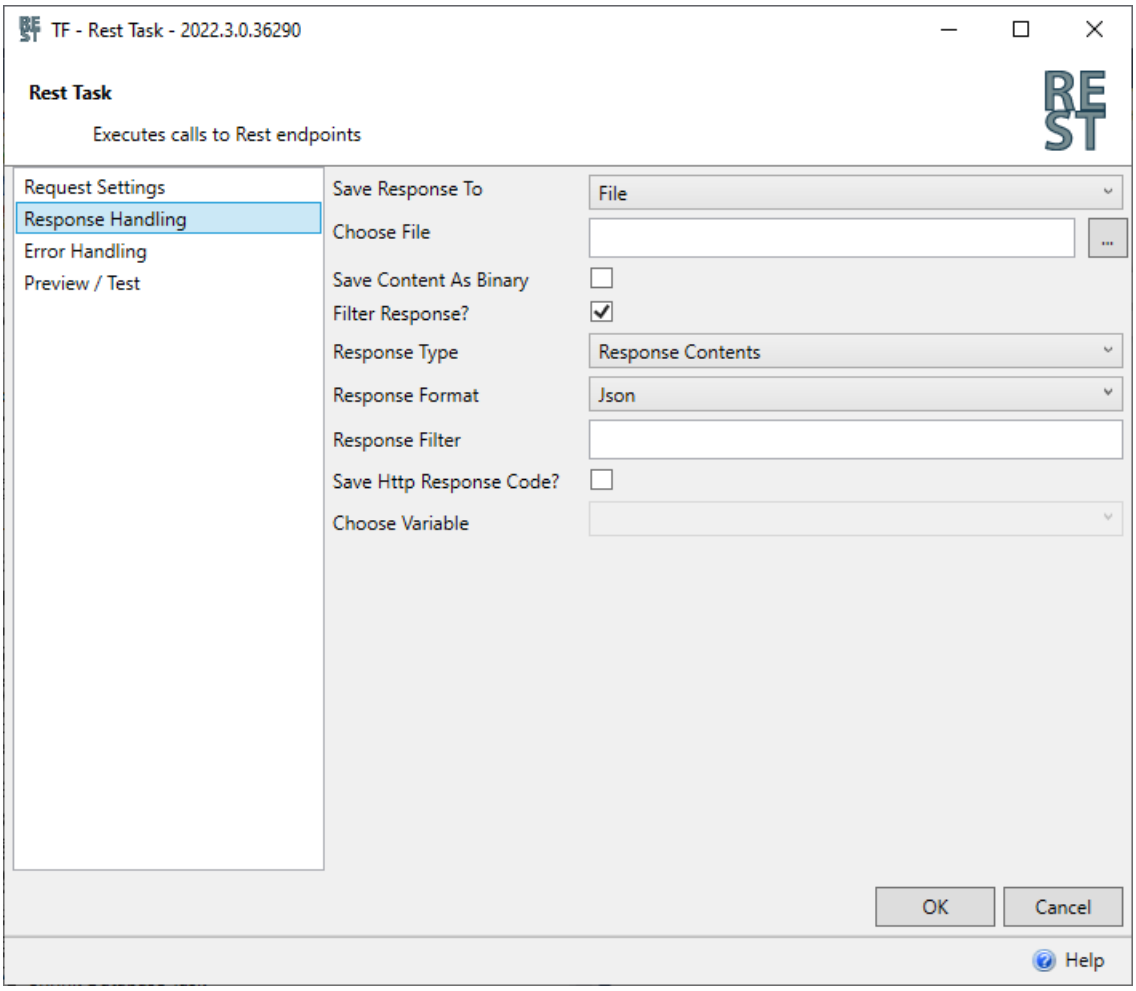
Task Factory Icon	Description
	REST Task allows you to execute calls to a REST endpoint. See the Rest Source Basic Connection Manager, Rest Source OAuth Connection Manager, and the Rest Source OAuth2 Connection Manager to learn more about configuring the component's connection manager.

Request Settings Tab



Option	Description
API Endpoint URL	In this field, you set the URL of the web service. You should also choose whether you're making a call to receive information (**Get**) or making a request to the service (**Post**.)
Get/Post Data	Specifies whether the action is a get or post request. **Note:** See your application's API page to determine the appropriate action.
Timeout	Sets the amount of time (in seconds) the component should fail if there is no response from the web service.
Number of Retries	Indicates the number of times to retry connecting to the API before the component fails.
Headers	Headers are simply information about the type of data returned.
Add New Header button / **Remove Header** button	You can add and remove headers by selecting the corresponding buttons. Once a header is added, select in its corresponding Name and Value/Variable Name fields to configure. **Note:** Header Values can be replaced by variables using the following syntax: **User::NameOfVariable**.

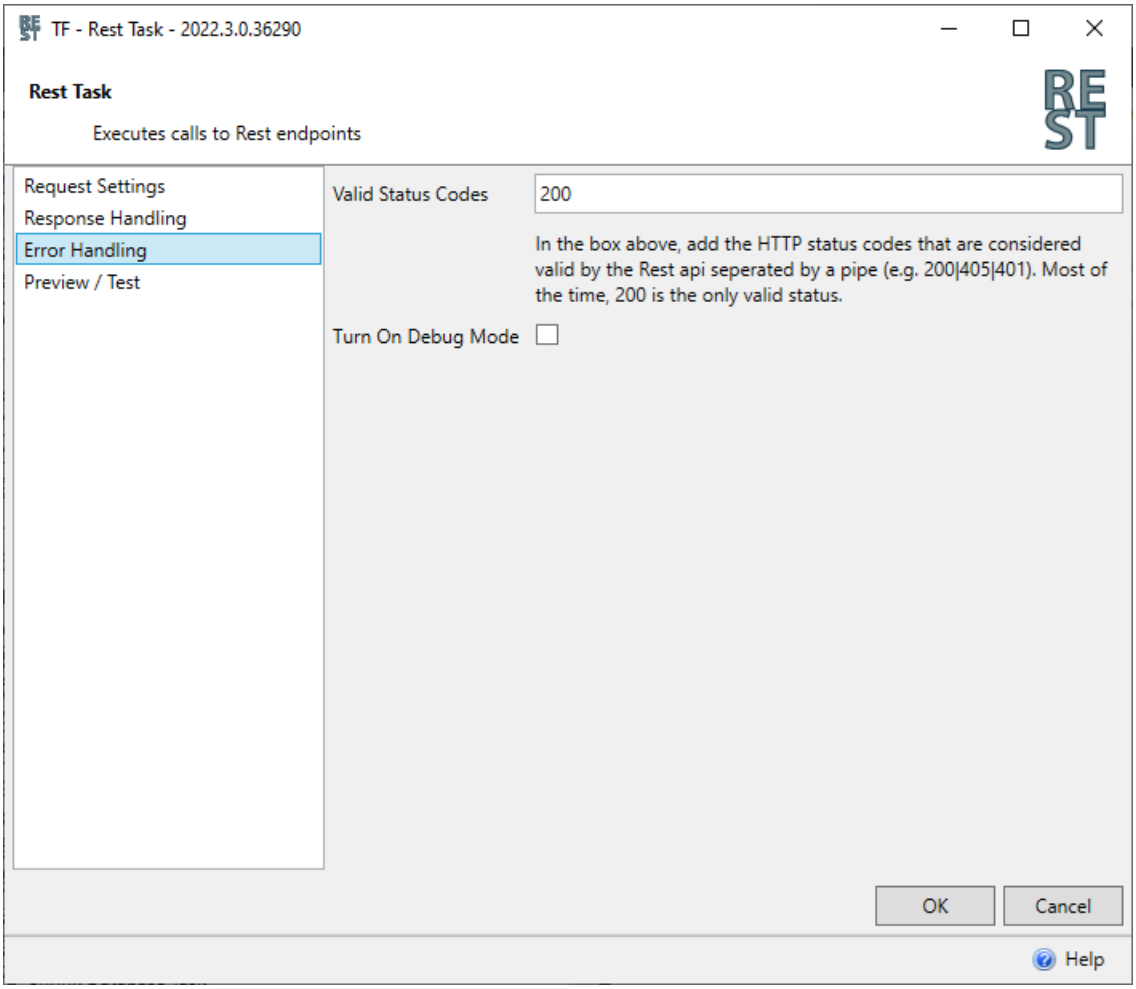
Response Handling Tab



Option	Description
Save Response To	Identifies the variable that populates with the response value.
Choose File	Select the file where you want to save the response. **Note:** This option enables when **File** is selected for the Save Response To option.
Save Content as Binary	Select this option to save the content as a binary. **Note:** This option enables when **File** is selected for the Save Response To option.
Filter Response	Select this option to filter responses.
Response Type	Select the response type you are filtering, **Response Contents** or **Response Headers**. **Note:** This option enables when **Filter Response** is selected.

Option	Description
Response Format	Select JSON, XML, or String. **Note:** This option enables when **Filter Response** is selected.
Response Filter	Add the words separated by commas to return only the items listed here.
Save Http Response Code	Select this option if an HTTP response code is returned.
Choose Variable	Identifies the variable that populates with the HTTP response code. **Note:** This option enables when **Save Http Response Code** is selected.

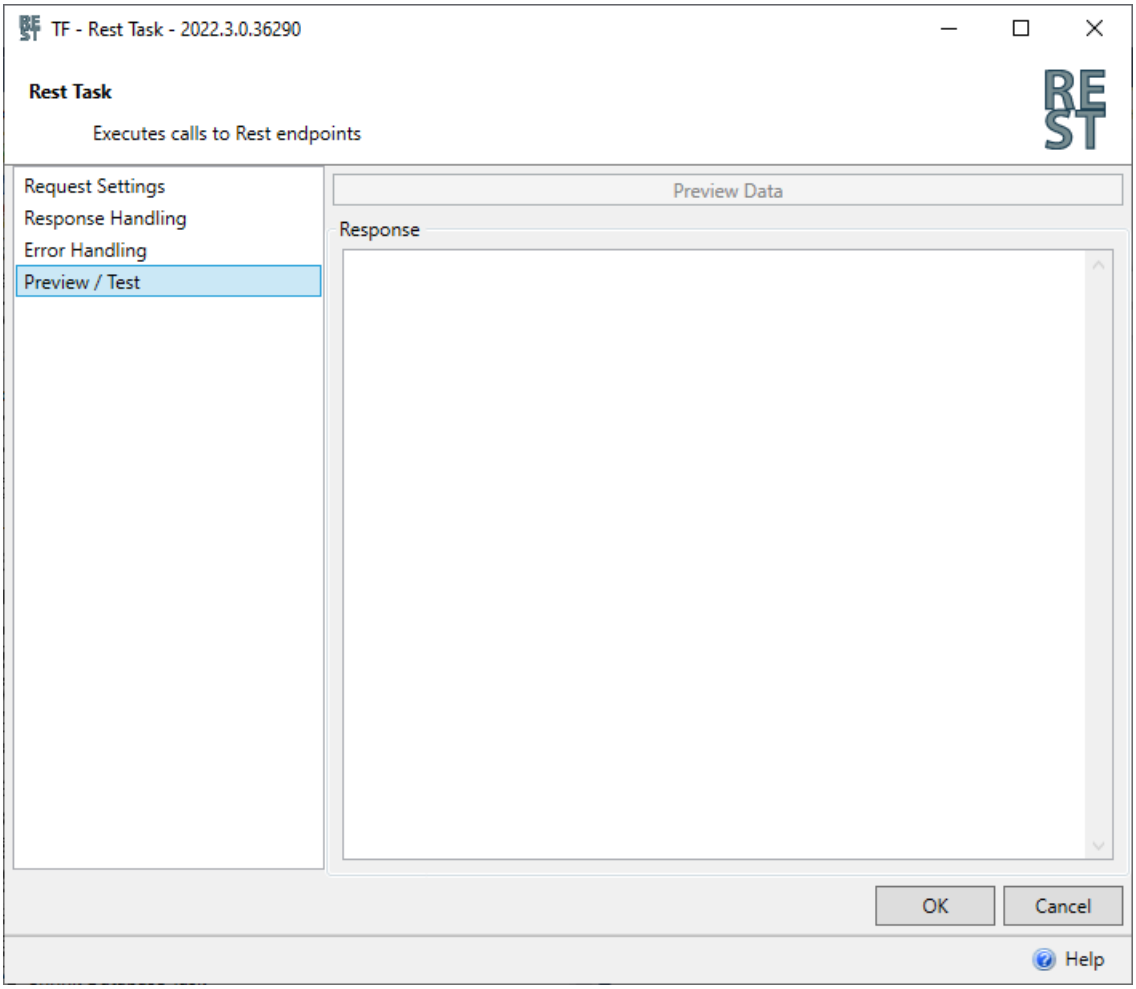
Error Handling (Rest Task)



Option	Description

Option	Description
Valid Status Code	Add the valid HTTP status codes returned by the REST API. Multiple codes can be added by separating them with the pipe (**|**) character. **Note:** Most of the time, 200 is the only valid status code.
Turn On Debug Mode	Package executes in debug mode.

Preview / Test



Once properly configured, select the **Preview Data** to view an output sample. The preview also returns raw XML/JSON/string data that can assist in configuring the component further.

REST Examples

Connecting to Google Analytics Example

The following is an example for configuring a Rest OAuth2 Connection Manager, and Rest Source to your Google Analytics API.

API Info

1. Log into your Google developer account tied to analytics, and create an App or agent that communicates with Google API.
2. Enable the Google Analytics Reporting API by going to **library**, and selecting **Google Analytics Reporting Api**. Select **enable** to activate the API.
3. Create an OAuth2 Client ID by selecting **Credentials** > **Create Credentials** > **OAuth Client ID**. Select the application type, in this example we are using a **Web Application.** Provide a name for the credentials, and then enter a redirect URI. Select Create to finalize the creation of the OAuth Client ID. The entered URI needs to be an HTTP or HTTPS URL that will respond whenever the API reaches out to it.
4. Return to **Credentials** to display the newly created Client ID under the **OAuth2 Client IDs** section. Select the client name to obtain the **Client ID**, **Client Secret**, and **Redirect URI** from the API. The **Client ID**, **Client Secret**, and **Redirect URI** are essential to establishing the Rest OAuth2 Connection Manager, and Rest Source.

Connection Manager

1. Create a new, or open an existing Visual Studio SSIS project and then create a new **Rest OAuth2 Connection Manager**. Google Analytics uses a Rest OAuth2 Connection Manager.
2. In the **Rest OAuth2 Connection Manager** window, select the **Google Analytics** Configuration file to open the **Token Getter** Window.
3. Enter the Redirect URI, the Client ID, and the Client Secret into the appropriate fields in the Token Getter. Select **Get Access** to establish access. Select **Allow Access** to the API to continue. If you are connecting to the API for the first time, you may need to log into the API to proceed.

> ℹ **Note:** Providing the above information into the **Token Getter** auto-populates many fields within the Connection Manager, including the **Refresh Token** tab.

Rest Source

1. Consult the applicable API reference guide for information about Endpoints.
2. Create a new **Rest Source**. Right click the **Rest Source** and select **edit** to configure the Rest Source.
3. Connect to the OAuth2 Connection Manager by selecting the Connection Manager from the **Connection Manager drop-down** list. Some fields within the Rest Source auto-populate based on the information provided to the connection manger in the configuration file.
4. In the **Api Endpoint** tab, enter the applicable Endpoint URL into the appropriate field, and then select the action you want the rest source to take.
5. Select the **Test API** tab to test the data that the API returns. Select **Preview Data** to display a Raw Response from the API.
6. Select the **JSON Properties** tab to configure the **Root JSON** path. Enter the applicable Root JSON Path,

and then select **OK** to complete the Rest Source.

> ℹ **Note:** For more information about the processes involved with this example, see the video tutorials below.

REST Samples

The following REST samples are available in the taskfactory-samples repository on GitHub:

- **1_SetupLocalRestDirectory.dtsx** - *Use to set up the REST Samples*
- **RestTask_SaveResponseToVariable.dtsx** - *REST Task configured to save response to a variable*
- **RestTask_FilteredResponse.dtsx** - *REST Task configured to filter the response of a request and save to a variable*
- **RestSource_BasicPullData.dtsx.dtsx** - *REST Source configured to pull data from an endpoint*
- **RestSource_CustomHeaders.dtsx** - *REST Source configured to use custom headers*
- **RestSource_DotNotation.dtsx** - *REST Source configured to use dot notation in a JSON token*
- **RestSource_VariableReplacements.dtsx** - *REST Source configured to use variable replacements in the Endpoint URL, PostBody, and Headers*
- **RestDestination_BasicPostData.dtsx** - *REST Destination configured to post data to an endpoint*

Tutorials

REST Tutorials

How to Connect to Yelp using the Task Factory REST Source

Tutorial walks through using the Task Factory REST Source to connect to Yelp to pull down data pertaining to local businesses.

How to make a Multipart call in the Rest Task

Some APIs (such as the public API Census GeoCoder from the United States Census Bureau), require you to make Multipart form-data calls. This example shows how to do that with the REST Task.

How to use the Unpack Data Transform with JSON

If you've connected to a REST Source, but are unable to parse more than one level of the JSON, this tutorial will help you through that.

How to use the Unpack Data Transform with XML

If you've connected to a REST Source, but are unable to parse more than one level of the XML, this tutorial will help you through that.

Using the Rest Task to get an Access Token and use it as a Bearer Token

Using the Rest Connection Manager to get an access token to use as a Bearer Token

Many APIs will need you to get an **Access Token** every call and use it as a **Bearer Token**. These examples shows you how to do that using the REST Task.

Video Tutorials

Using Rest Source with Salesforce

Using Rest Source to Connect to Google Analytics

Rest Source Training Video Title	Video

Rest Source Training Video Title	Video
API Endpoint	
Properties	
Headers	

Rest Source Training Video Title	Video
Pagination Settings	
Authentication Results	

Rest Source Training Video Title	Video
Test API	

Rest Destination Training Video Title	Video
API Endpoint	

Rest Destination Training Video Title	Video
Headers	
Valid Status Codes	

Rest Destination Training Video Title	Video
Test Endpoint	

Rest Task Training Video Title	Video
Request Settings (Rest Task Video)	

Rest Task Training Video Title	Video
Response Handling (Rest Task Video)	
Error Handling (Rest Task Video)	

Task Factory SalesForce.com

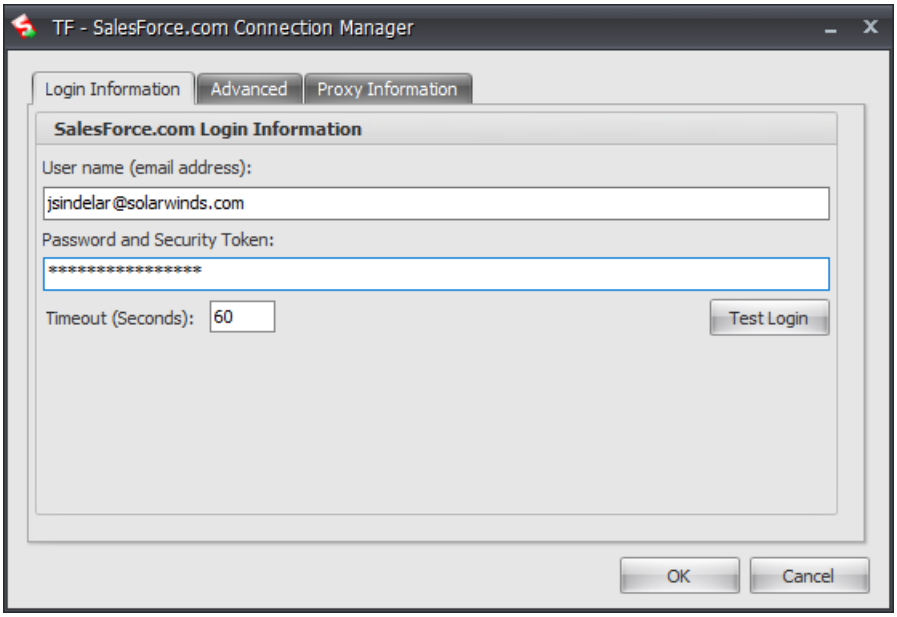
Last Modified on 17 February 2022

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

SalesForce.com Connection Manager

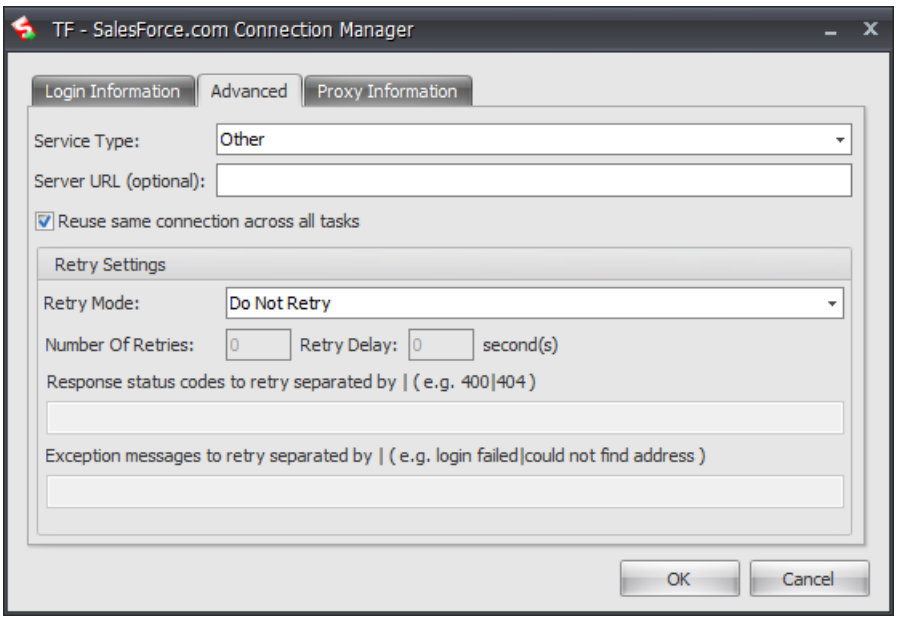
This connection manager is used with **SalesForce.com Source** and **SalesForce.com Destination**.

Login Information Tab



Option	Description
User Name (email address)	This is your log in email address for SalesForce.com.
Password and Security Token	If you do not have your security token you must request it from SalesForce.com. You can do this by logging into your SalesForce account on their site, selecting the Setup link, and then selecting to **Reset your security token**. ⚠**Important:** Enter the password and security token without a delimiter, in the following format: • **PasswordSecuritytoken**
Timeout (seconds)	Seconds before the connection times out trying to connect.

Advanced Tab

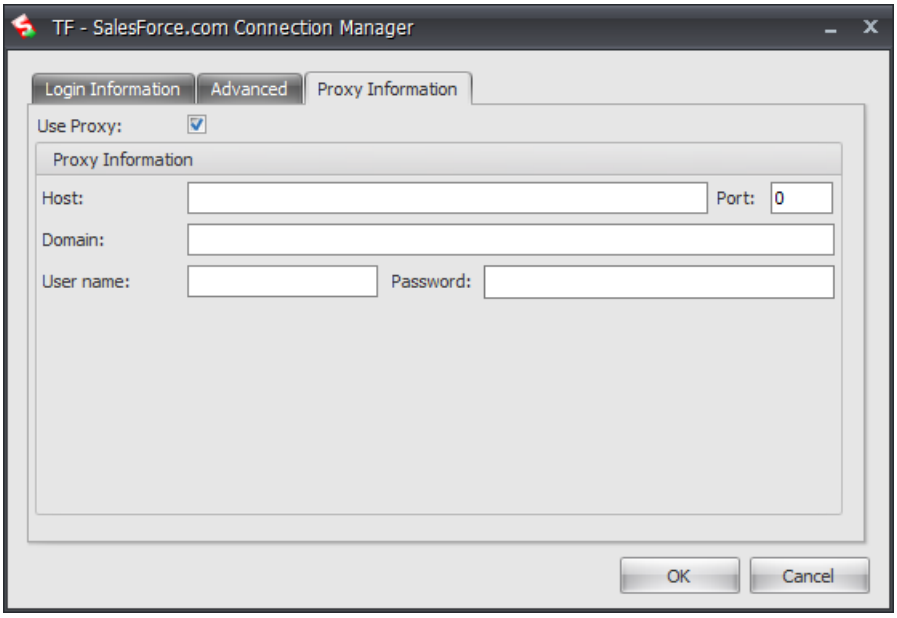


Option	Description
Server URL (optional)	If you are using a SalesForce.com sandbox account the Server URL text box needs to be changed to point to a sandbox web service URL.
Reuse same connection across all tasks	This option allows you to use the same connection across any SalesForce task that is part of your package, that uses this connection manager.
Number of retry attempts	Identifies the number of attempts to reconnect to Salesforce before the component fails. This option is useful when pulling large datasets and executions fail due to timing out.

Retry Settings

Option	Description
Retry Mode	Do Not Retry - The component does not attempt to reconnect. Retry For Any Errors - The component attempts to reconnect for any error. Retry for Http Status codes - The component attempts to reconnect for Http Status codes that you enter in the **Response status codes to retry separated by** field. Retry for Exception Message - The component attempts to reconnect for exception messages that you enter in the **Exception messages to retry separated by** field.
Number of Retries	Set the number of attempts to reconnect to Salesforce before the component fails. This option is useful when pulling large datasets and executions fail due to timing out. **Note:** This option becomes available when you select a Retry Mode other than **Do Not Retry**.
Retry Delay (in seconds)	Set the time in seconds that the component waits before trying to connect to Salesforce. **Note:** This option becomes available when you select a Retry Mode other than **Do Not Retry**.
Response status codes to retry separated by	Enter the response status codes that if encountered, the component tries to reconnect. **Note:** Enter multiple Response status codes separated by a **|** (bar). This option becomes available when you select **Retry for Http Status codes**.
Exception messages to retry separated by	Enter the exception messages that if encountered, the component tries to reconnect. **Note:** Enter multiple Exception messages seperated by a **|** (bar). This option becomes available when you select **Retry for Exception Messages**.

Proxy Information Tab



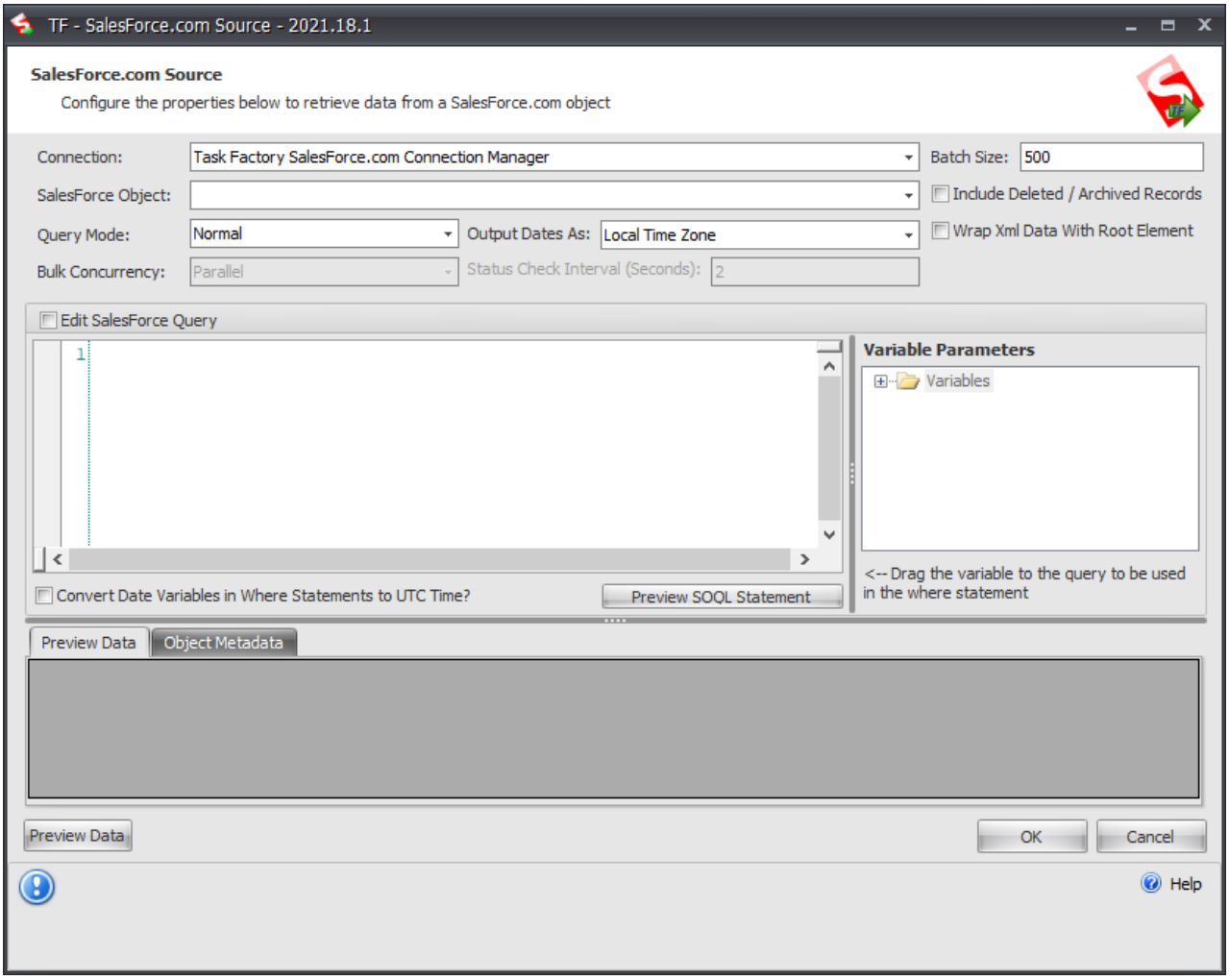
Note: Auth Scheme is only available on machines that are running SQL 2008 only.

Option	Description
Use Proxy	If you wish to make your connection through a proxy you first select **Use Proxy**, and then fill out the appropriate information below in the Proxy Information pane.

SalesForce.Com Source

Source Icon	Source Description

Source Icon	Source Description
	The **SalesForce.com source** is used to retrieve data from SalesForce.com objects within an SSIS package. The data is retrieved using a SQL statement built by the SalesForce.com source. See the SalesForce.com Connection Manager to learn more about setting up the connection manager.

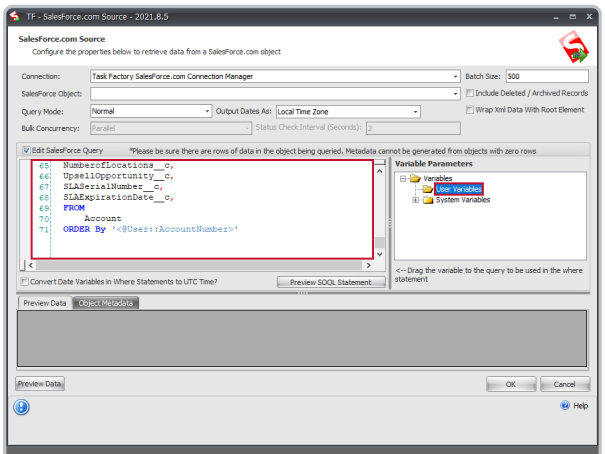


Note: SalesForce Source 2019.1.1 and above support nested queries.

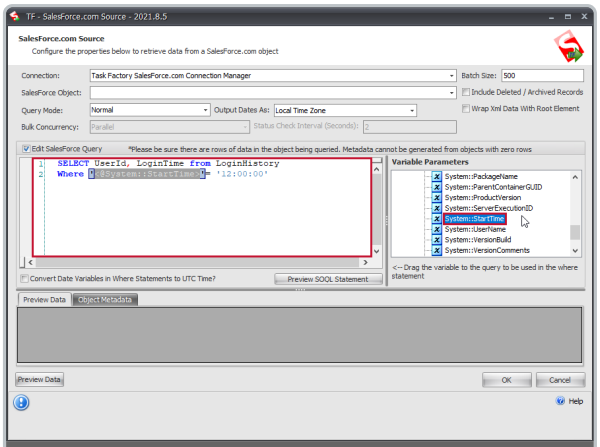
Option	Description
Connection	Here you select a connection manager that already exists or you may choose to create one.

Option	Description
SalesForce Object	The name of the SalesForce.com object that is used to retrieve data from your SalesForce.com account. Choose your object from the drop down list after connecting to SalesForce.com. Once your object is selected, a query is created in the text box below.
Batch Size	Determines the number of rows that are retrieved during a single round trip to the SalesForce.com server.
Include Deleted / Archived Records	Selecting this option returns deleted and archived records. The **IsDeleted** column in the output allows you to determine whether the row is deleted.
Query Mode	• **Normal** - Uses normal processing which can use up to 200 batch size. Normal mode is a SOAP request. • **BulkQuery** - Uses bulk processing that can achieve up to 10,000 batch size. BulkQuery mode is webrequest with a file stream.
Output Date As:	Using **BulkQuery** mode, you can choose between the **Local Time Zone** or **UTC Time Zone** options.
Bulk Concurrency	Using the **BulkQuery** mode, you can choose between parallel (based on the batch size) and serial (one record at a time) modes.
Status Check Interval (seconds)	The amount of time in seconds before a status check occurs. **Note:** The **Status Check Interval** is two seconds by default.
Edit SalesForce Query	Selecting this makes the auto filled query editable. **Note:** : Variable replacements can be used within queries. The syntax used is . For variable replacements that require the elimination of tics (''), use the syntax . For example: SELECT FROM Opportunity
SalesForce Query Window	Once your object is selected, this window fills with a query.
Variable Parameters	Variables here can be dragged into the query window to edit or create new queries.

Variable Parameter Examples



User Variable example using syntax



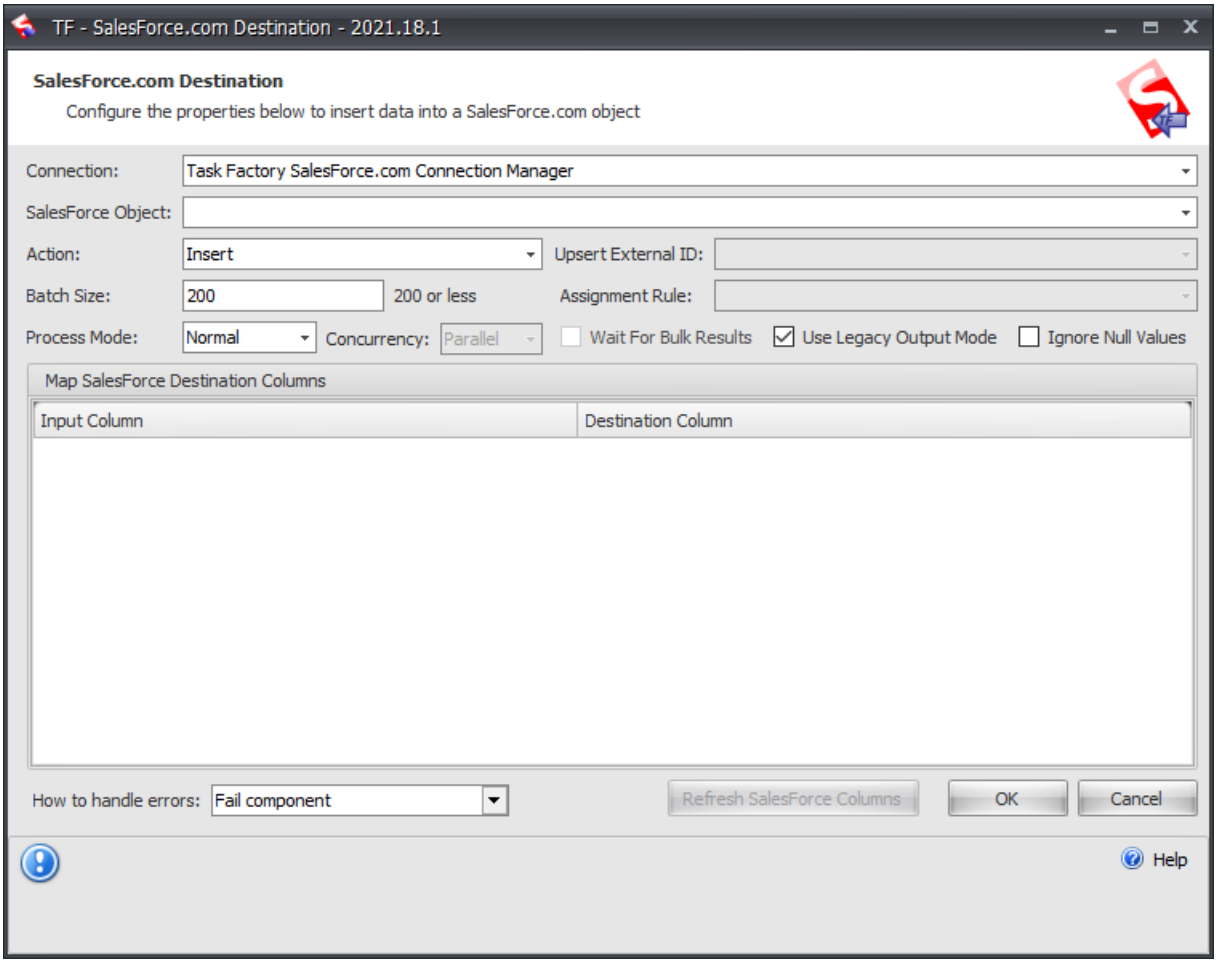
System Variable dragged from Variable Parameters example

Additional Videos

Export Data from Salesforce Source In Under 2 Minutes

SalesForce.com Destination

Destination Icon	Destination Description
	The **SalesForce.com Destination** is used to send data to a SalesForce.com object. See the SalesForce.com Connection Manager to learn more about setting up the connection manager.



Option	Description
Connection	Here you select an existing SalesForce.com connection, or create a new one.
SalesForce Object	Once selecting the connection, you then select and object where the data is inserted within your SalesForce.com account.

Option	Description

Action	• **Insert**: Inserts data into the SalesForce object.
	• **Upsert**: Inserts data into the SalesForce object if the data does not exist or update the data if it already exists.
	  ◦ Upsert requires that an external ID be added to your SalesForce object in your SalesForce.com account.
	• **Update**: Updates your data in the SalesForce object based on the ID column from the SalesForce object.
	  ◦ The ID from the SalesForce object must exist in the local source data. To the ID column data, the TF Sales Source Adapter must be used to retrieve the data or the ID needs to be retrieved and updated using the output from the SalesForce Destination.
	• **Delete**: Deletes data from the SalesForce object by sending it to the Recycle bin.
	  ◦ The ID column of the SalesForce object is used to delete the data.
	• **Merge**: Merges three or more SalesForce objects of the same type. You must supply the master record ID and at least one ID for the Merged IDs.
	  ◦ You can supply a maximum of two IDs for this action.
	• **HardDelete**: Bypasses **Delete** and permanently deletes data from the SalesForce object.
	  ◦ The ID column of the SalesForce object is used to delete the data.
	Warning: Using **HardDelete** permanently deletes data instead of sending the data to the Recycle bin.
	• **Undelete**: Restores previously deleted data from the SalesForce object by removing the data from the Recycle bin.

Upsert External ID	If you selected the Upsert action you must select your External ID here.
Batch Size	You can configure a custom batch size (200 or less).
Assignment Rule	Select an assignment rule created for their object(s). Rules can be ignored by selecting [No Assignment Rule].

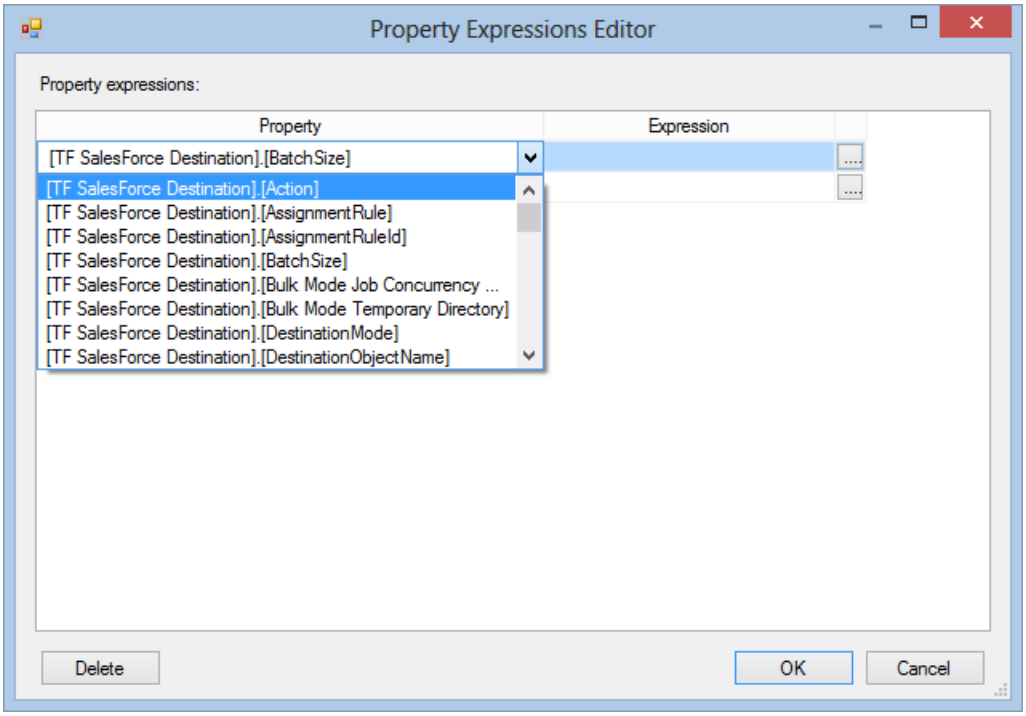
Option	Description
Process Mode	• **Normal**: Creates a CSV file using normal processing that can use up to 200 batch size. • **Bulk**: Creates a CSV file using bulk processing that can achieve up to 10,000 batch size. Bulk mode uses more network traffic but less IO because no compression occurs. • **BulkZip**: Also uses bulk processing but compresses the bulk CSV file before sending. BulkZip uses more IO for compression but less network traffic. • **Bulk XML**: Creates an XML file using bulk processing that can achieve up to 10,000 batch size. • **Bulk XML zip**: Also uses bulk processing but compresses the bulk XML file before sending. Bulk XML uses more IO for compression but less network traffic.
Concurrency	Using bulk processing mode, you can choose between parallel (based on the batch size) and serial (one record at a time) modes.
Map SalesForce Destination Columns	Once you have an object selected, the columns here will be filled. Any column names that match are automatically mapped. To map more columns, select on the Input column that you wish to be mapped to the destination column.
How to handle errors	There are three options to handle errors: • **Fail Component**: The component fails upon the first data error that is thrown. • **Redirect row to error output**: The rows of data that failed are sent to the error output which then can be used to handle the data errors. • **Ignore Failure**: The failures are ignored but reported in the Execution Results log.
Refresh Salesforce Columns	Selecting this button allows the component to refresh the metadata and update any changes made in Salesforce after the destination was opened.
Wait For Bulk Results	This selection can be used along with the bulk processing mode. When selected, it sends a batch of rows to a job and waits for the results of the operation (insert, update, upsert, delete) and then outputs the results to the success output with the ID's generated by Salesforce when updated or created. If this option is not selected, the component does not wait for the results and processes the execution as quickly as possible, and the Salesforce ID is not properly returned.
Use Legacy Output Mode	In legacy versions of Task Factory, you could update the unique ID's created by Salesforce.com by connecting a destination component to the Salesforce Destination's error output. This legacy output also returned errors (when error handling was enabled) which could be confusing when sifting through multiple updates mixed with errors. The current version now defaults with the option turned off thus directing the unique ID to the success output. This change allows you to separate the data returned from Salesforce.com from the error output.

Option	Description
Ignore Null Values	Selecting this option allows the component to ignore any field that has null values.

Configuring the Salesforce Destination using Expressions

Users who wish to use expressions to configure different properties of the Salesforce Destination should follow these steps:

1. Return to the package's control flow.
2. Right-click the Data Flow that contains the Salesforce Destination and select **properties**.
3. Navigate to the Expressions property and select the **ellipsis (...)** .
4. Select the property in the dropdown menu to be configured.



Additional Information: See the Task Factory Error Row Handling article for more information about this functionality.

Task Factory Secure FTP

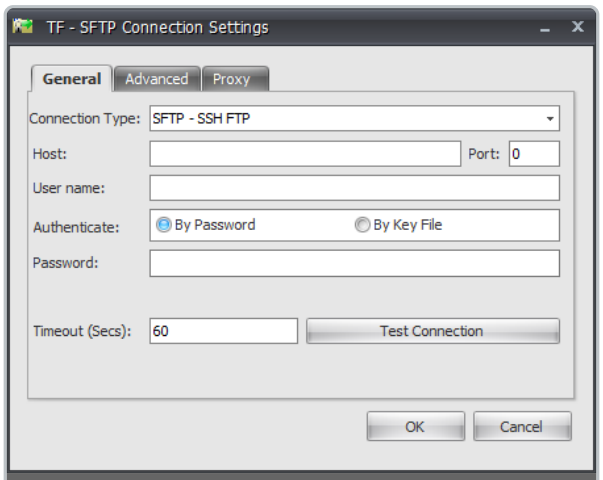
Last Modified on 03 March 2022

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

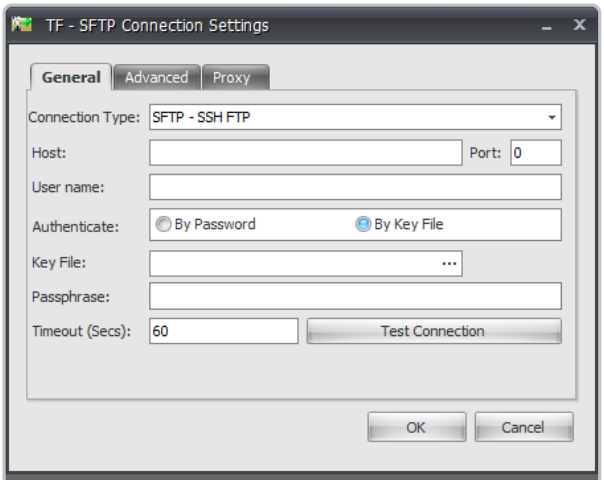
Secure FTP Connection Manager

The **SFTP** connection manager is used to create a connection manager with an SFTP site. It is used with the **Secure FTP** task.

General Tab



General - By Password



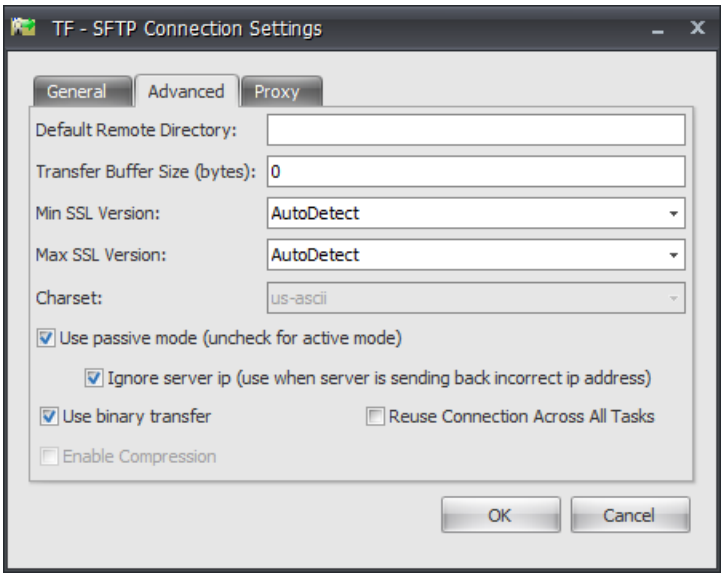
General - By Key File

Option	Description

Option	Description
Connection Type	There are five connection types that are supported: • SFTP - SSH FTP (**Note:** This supports SSH2) • FTPS - FTP over implicit TLS or SSL • FTPES - FTP over explicit TLS or SSL • FTP **Important:** The **SFTP - SSH FTP**, **FTPS - FTP over implicit TLS or SSL**, **FTPES - FTP over explicit TLS or SSL**, and **FTP** connection types use the Chilkat connection service. **Additional Information:** For more information about the **Chilkat** connection service, see Chilkat. • SFTP - SSH FTP Legacy **Important:** The **SSH FTP Legacy** connection type uses the **edtFTPnet/PRO** connection services. Use this connection type if you are using a version of Task Factory from 2017 or older. **Additional Information:** For more information about the **edtFTPnet/PRO** connection service, see edtFTPnet/PRO.
Host	The IP Address or URL of the FTP server.
Port	The port used for the FTP server.
User name	The user name used to login to the FTP server.
Authenticate	Select how you authenticate the connection. • By Password • By Key File
Password	This option is available if **By Password** is selected in the **Authenticate** section.
Key File	Available if **By Key File** is selected in the **Authenticate** section. You can use the file browser to locate the file.
Passphrase	Available if **By Key File** is selected in the **Authenticate** section.

Option	Description
Timeout (secs)	The number of seconds before the application stops trying to make a connection if it cannot connect.
Test Connection button	Use this button to test whether a connection can be made using the settings in the **General** and **Advanced** tab.

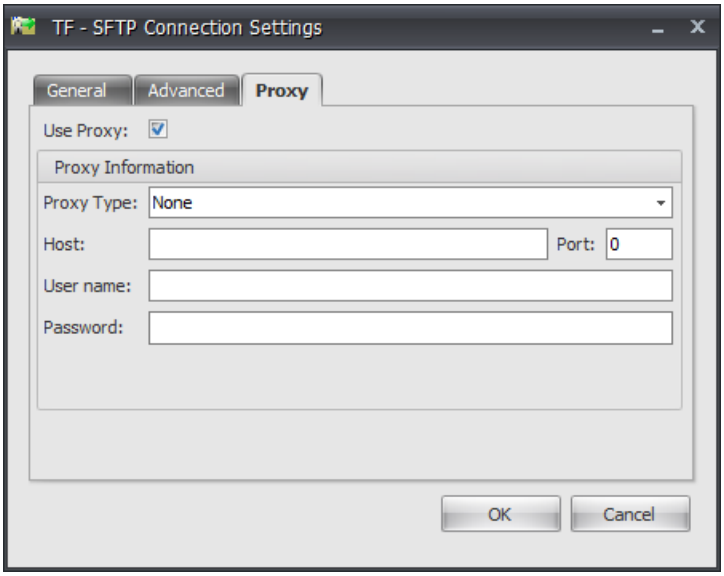
Advanced Tab



Options	Description
Default Remote Directory	The default remote directory tells the connection to change directories on the FTP server for each connection made to the server. You must use backslashes to define the directory.
Transfer Buffer Size (bytes)	Specify the buffer size.
SSL Version	Select the SSL version: • AutoDetect • SSL3 • TLS1 • TLS1 1 • TLS1 2
Use binary transfer	Selecting this option switches the mode of transfer from ASCII to Binary. Select this option if you are primarily working with non-text file. **Note: Use binary transfer** is only applicable to legacy connections.
Use passive mode	Deselecting this option switches to active mode.

Options	Description
Ignore server IP	Use when the server is sending back incorrect IP addresses.
Reuse Connection Across All Tasks	Select this to make the connection reusable for all tasks.
Enable Compression	Select this option to enable compression for the SFTP- SSH FTP, or SFTP - SSH FTP Legacy options.

Proxy Tab



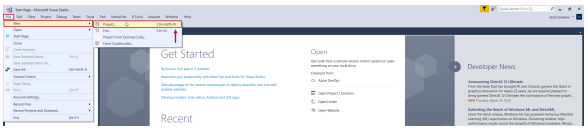
Option	Description
Use Proxy	Selecting this allows the use of a proxy, to use the proxy you must fill in the information below the check box.

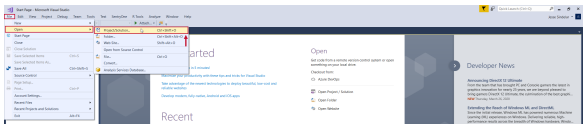
Configuring the SFTP Connection Manager

To configure the Task Factory **SFTP** connection manager in your Visual Studio Package, complete the following steps:

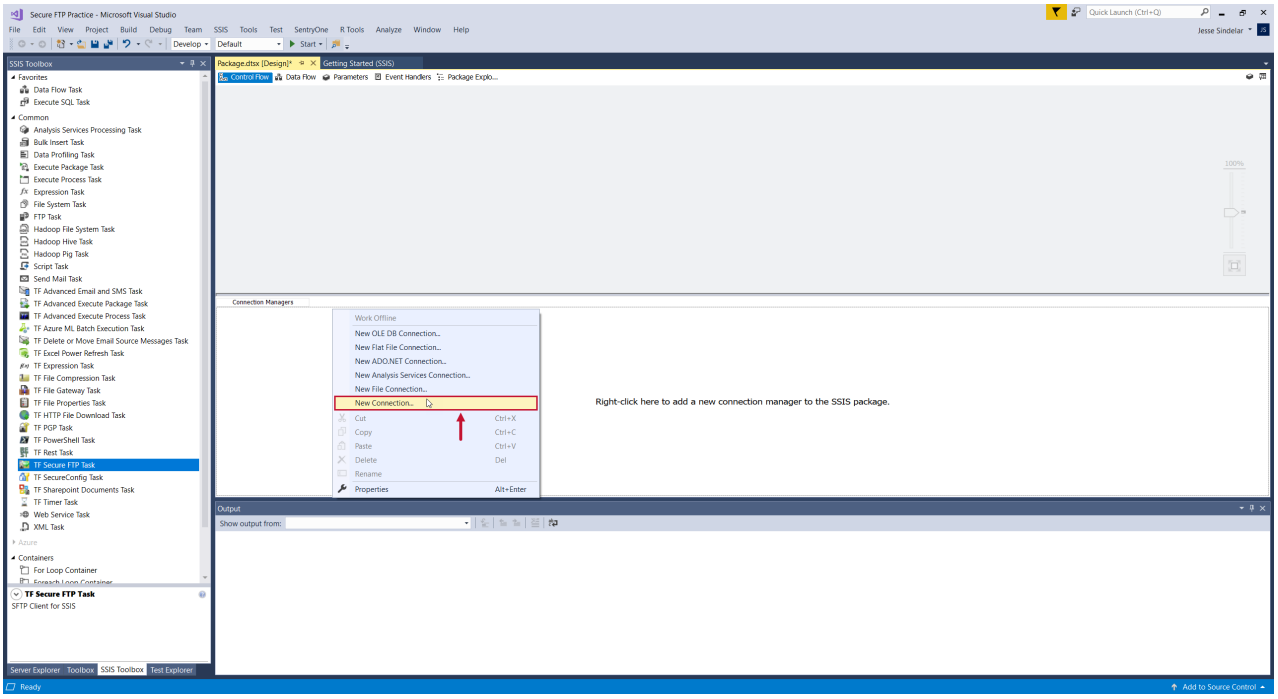
> **Note:** This example uses the free demo SFTP servers that can be found on Free Public SFTP Servers.

1. Create a new, or open an existing, Visual Studio Integration Services project.

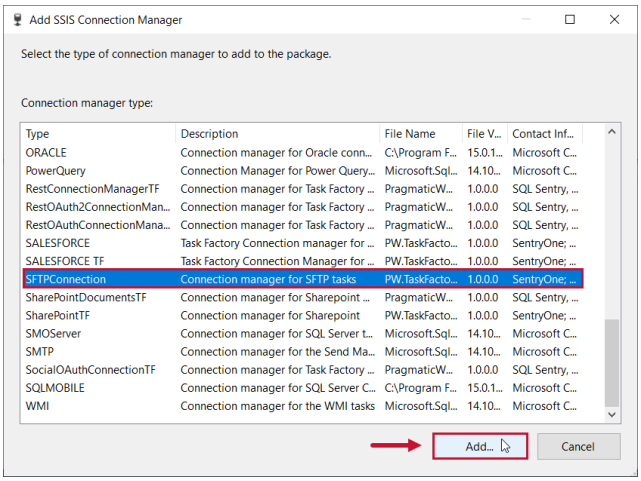




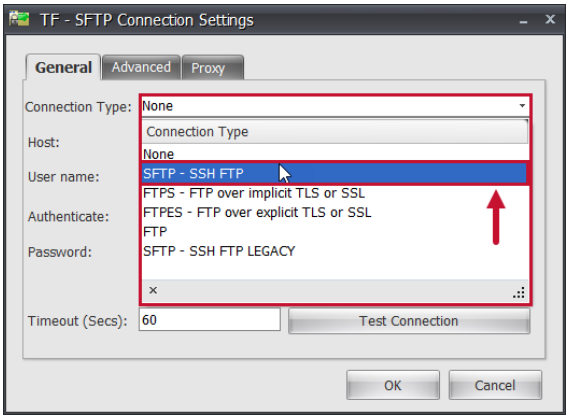
2. Right Click the Connection Managers pane, and then select **New Connection** to open the **Add SSIS Connection Manager** window.

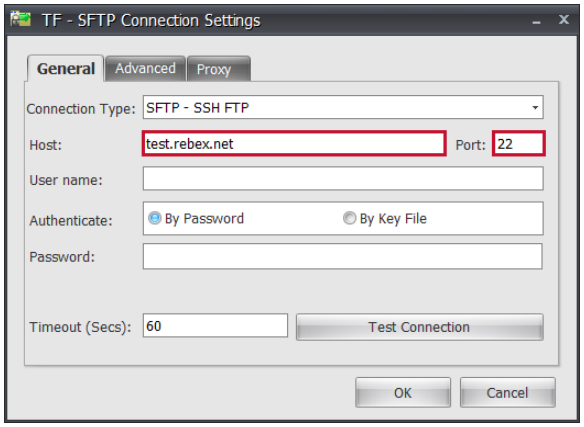


3. Select the **Task Factory SFTP Connection**, then select **Add** to open the **TF-SFTP Connection Settings** configuration window.

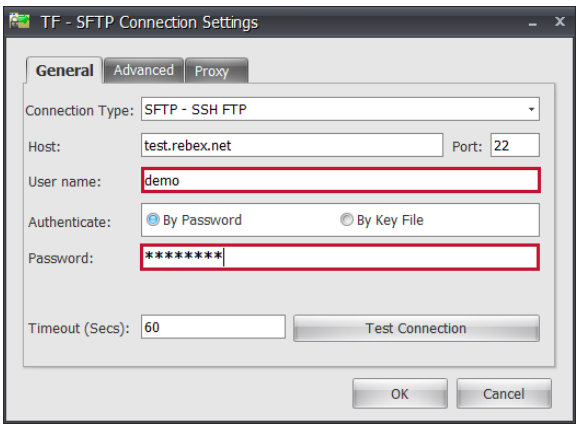


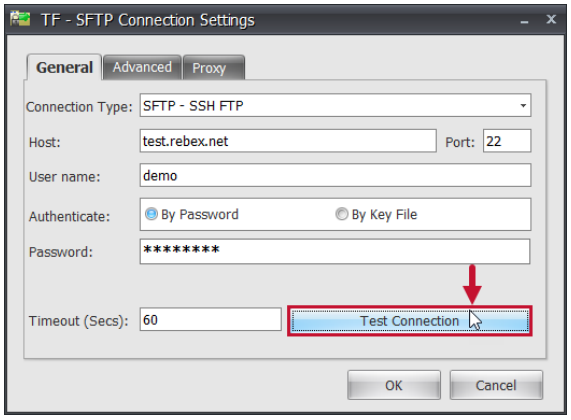
4. Select your **Connection Type** from the drop down list. Enter your connection's **Host** address and **port** number in the appropriate fields.



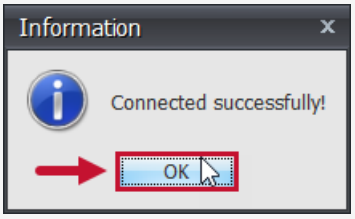


5. Enter the **User name** and **Password** associated with your connection, then select **Test Connection** to ensure that you can connect successfully to your SFTP server.

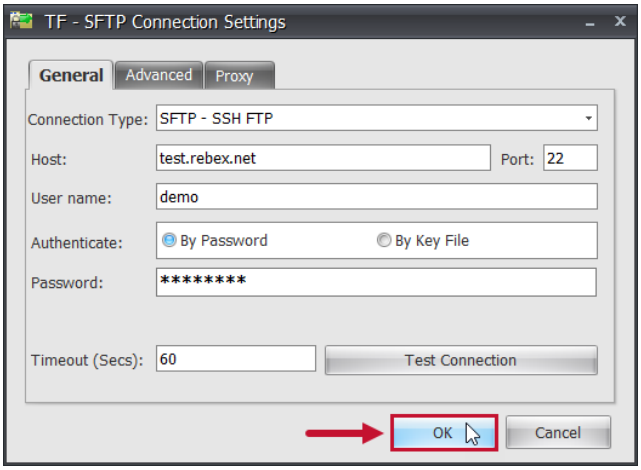




Note: A **Connected Successfully** message prompt displays if your connection was successful. Select **OK** to close the prompt.



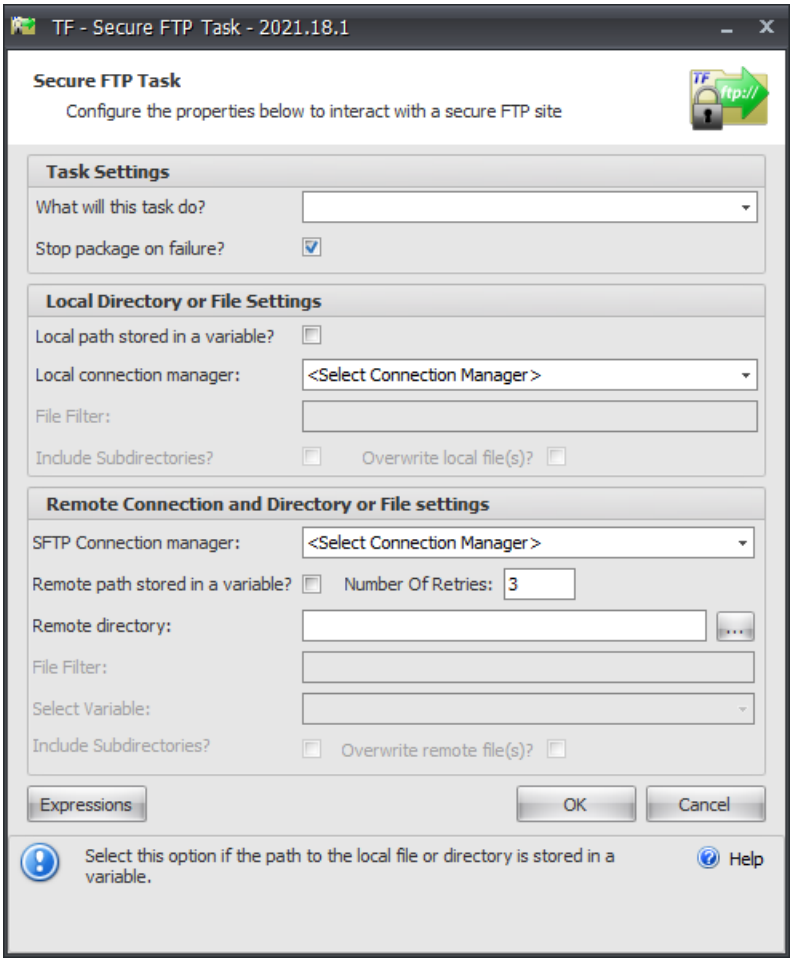
6. Select **OK** to save your TF SFTP connection manager.



Note: For more information about **Advanced Settings**, or using a **Proxy** for your connection, see the **Advanced Settings**, and **Proxy** sections of this article.

Secure FTP Task

Task Icon	Task Description
	The **Secure FTP Task** interacts with a secure FTP server. See the **Secure FTP Connection Manager** tab to learn more about setting up the connection manager.



Task Settings

Option	Description
	Choose what the task does: **Option** — **Description** **Upload file to server** — Uploads a selected file to your desired server connection. **Download file from server** — Downloads a selected file from your desired server connection. For a detailed example on configuring the **Download file from sever** action in the **Secure FTP** task to download a list of files from an SFTP server, see the following support article: SFTP Task Download list of files from SFTP Server.

Option	Description
What will this task do?	

Option	Description
Create directory on server	Creates a directory on the desired server connection.
Delete directory on server	Deletes the directory on the desired server connection. **Note:** You must delete all the files in the selected directory before that directory can be deleted with the Secure FTP Task.
Delete file on server	Deletes a selected file on the desired server connection.
Upload directory to server	Uploads a selected directory to the desired server connection.
Download directory from server	Downloads the selected directory from the desired server connection.
Check if files exists on server	Searches for a selected file on the desired server connection.
Get list of files	Retrieves a list of files with metadata from the desired server connection. **Additional Information:** You can configure the **Get list of files** action in the **Secure FTP** task to retrieve a desired file type from a specified directory. For a detailed example on configuring the **Get list of files** action, see the following support article: Secure FTP Task - Get List of Files with Metadata.
Rename file	Renames the selected file on the desired server connection. **Important:** When configuring the component to rename a file, the **New File Name** must include the path. For example, renaming a file in the documents folder is configured **/Documents/RenamedFile.ext**.

Option	Description
	<table><tr><th>Option</th><th>Description</th></tr><tr><td>**Delete directory files on server**</td><td>Deletes the directory files from a selected directory on the desired server connection.</td></tr></table>
Stop package on failure?	This fails the package if the selected action fails to complete successfully. **Note:** This option is selected by default.

Local Directory or File Settings

Option	Description
Local path store in a variable?	Choose if the path is in a variable, selecting this changes the **Local Connection Manager** drop down to **Select Variable**.
Local connection manager	Available if **Local path stored in a variable** is not selected. Choose or create a connection manager.
File Filter	Use for **Upload directory to server** task. By using the ***** symbol you can specify different filters. For example to only upload XML files you would use ***.xml** .

Option	Description
Include Subdirectories	Choose whether or not to include sub-directories for this task.
Overwrite local file(s)?	This allows the task to overwrite the files if they already exist.

Remote Connection and Directory or File Settings

⚠**Important:** When configuring the component to rename a file, the **New File Name** must include the path. For example, renaming a file in the documents folder is configured **/Documents/RenamedFile.ext**.

Option	Description
SFTP Connection Manager	Select an existing or create a new SFTP connection manager.
Remote path stored in a variable?	Selecting this option changes the **Directory on server to upload file:** drop down context to
Directory on server to upload file	Depending on what action is selected, this option is used to define the directory or file to be used on the remote server to complete the selected action.
File Filter	This option is used to define a filter when using the **Download directory from server** option.
New name of file on server	Enabled when **Upload file to server** option is selected. Users can create or rename the destination file. Best practice is to include the file's extension. Example: Source file *UploadThisFile.txt* can be renamed to the destination server as *SameFileWithDifferentName.txt*
Select Result Variable	To use this action you must select a variable to contain the result of the file check. Once the SFTP Task completes, the selected variable contains the result.
Include Subdirectories	Choose whether or not to include subdirectories for this task.
Overwrite remote file(s)?	This allows the task to overwrite the files if they already exist.

Expressions

Additional Information: See the Expressions article for details on using the **Expressions** button.

SFTP Using a Connection String

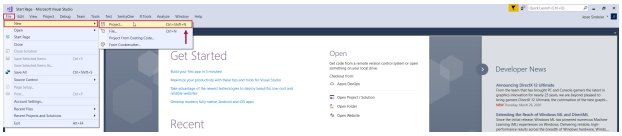
This example shows you how to put the SFTP connection string in a variable, so that in the properties of the connection manager, you can set the expression of the connection string to take the created variable.

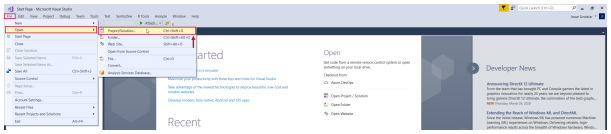
Configuring a Secure FTP Task

> **Note:** You need to configure an SFTP connection manager for your Secure FTP Task. For more information about configuring an SFTP Connection Manger, go to the **Connection Manager** tab.

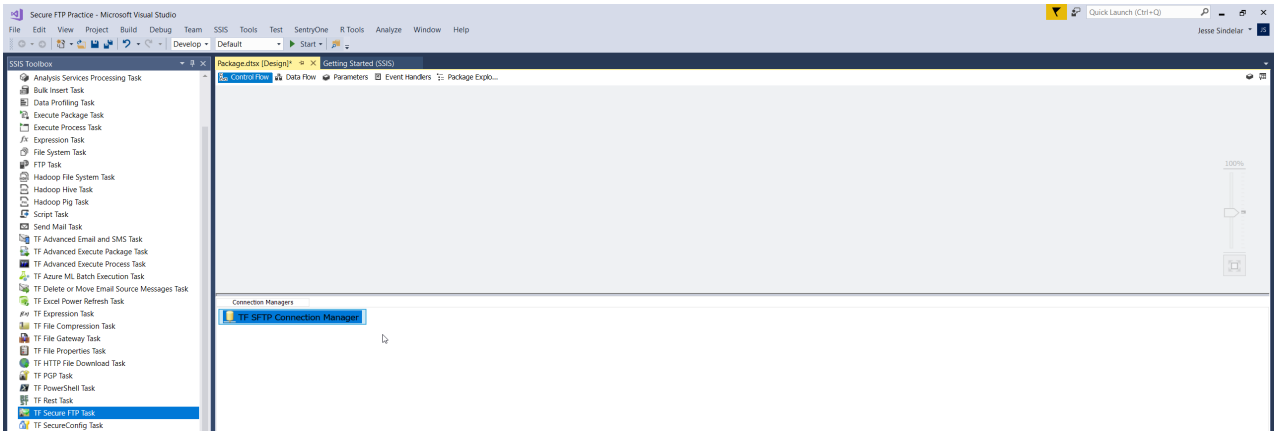
To configure a **Secure FTP Task** in your BIDS package, complete the following steps:

1. Create a new, or open an existing, Visual Studio Integration Services project.



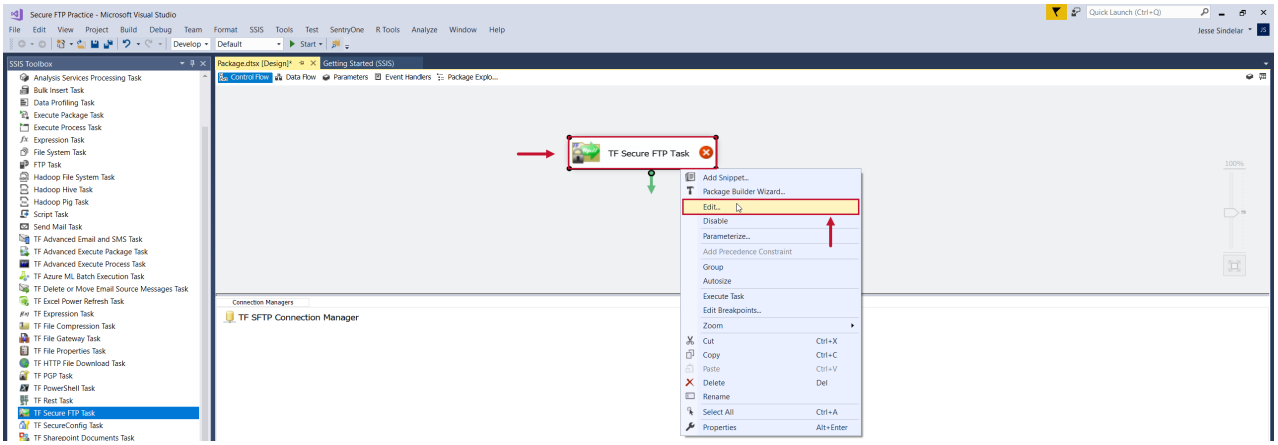


2. Configure your **TF SFTP Connection Manager**.

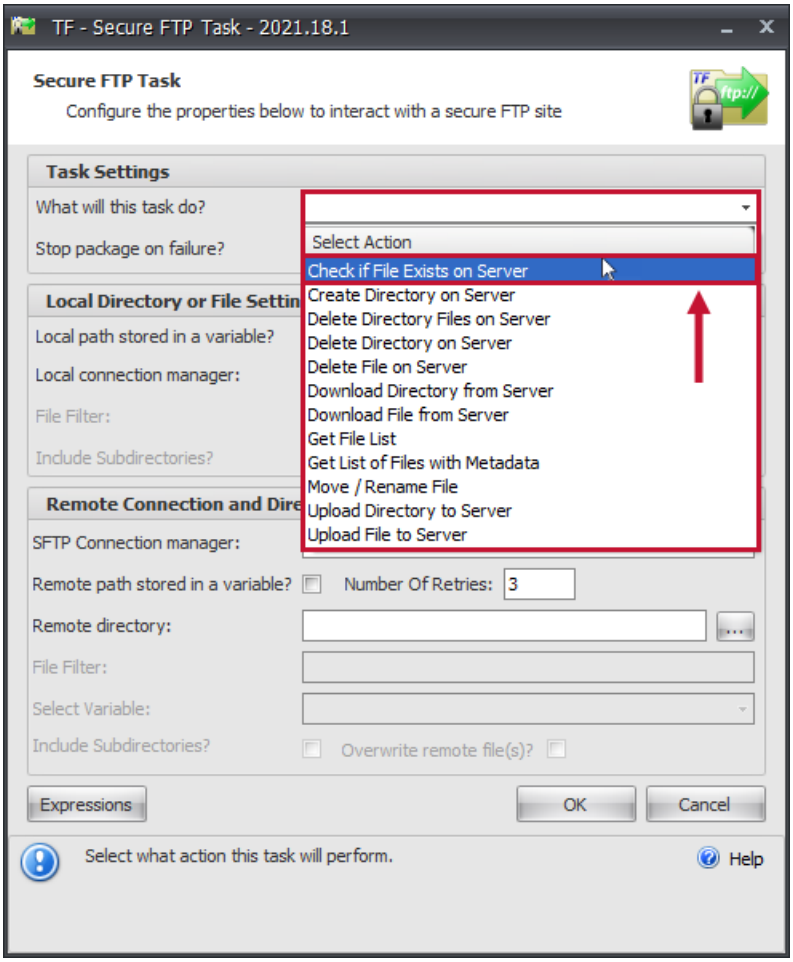


> **Note:** For information about configuring an **SFTP Connection Manager**, see the **Connection Manager** section.

3. Drag the **TF Secure FTP Task** component into your package Control Flow. Select the task, then select **Edit** to begin configuring the task properties.



4. Select what you want the task to do from the **What will this task do** drop-down list.

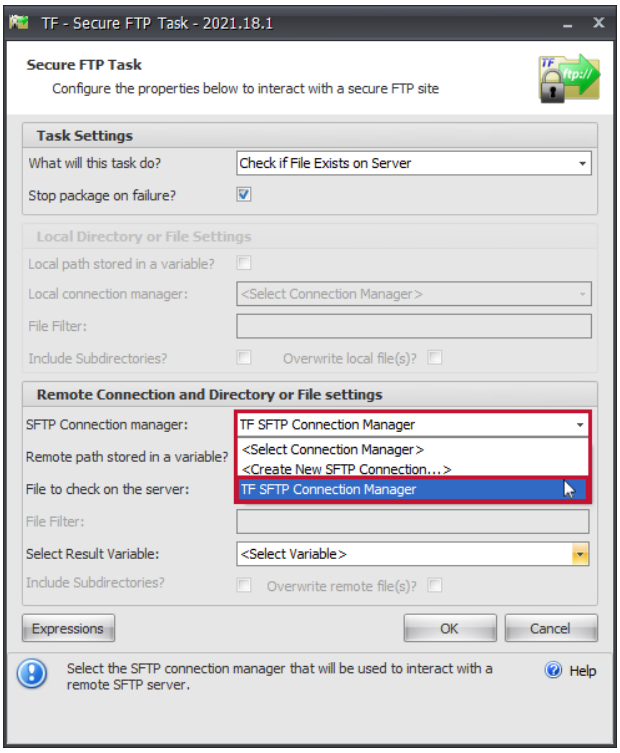


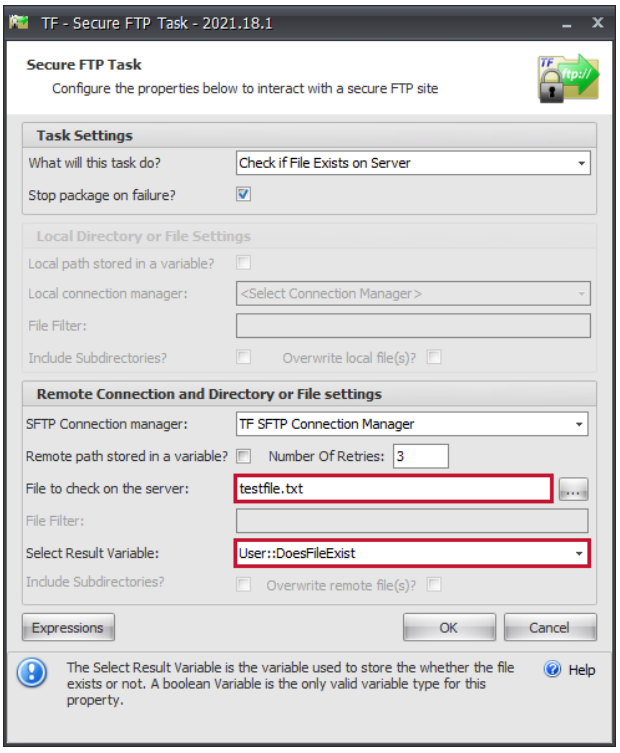
> **Note:** For a detailed list of actions this task can take, see **Task Settings** table in above.

> **Important:** Your selection will change the configurable task options.

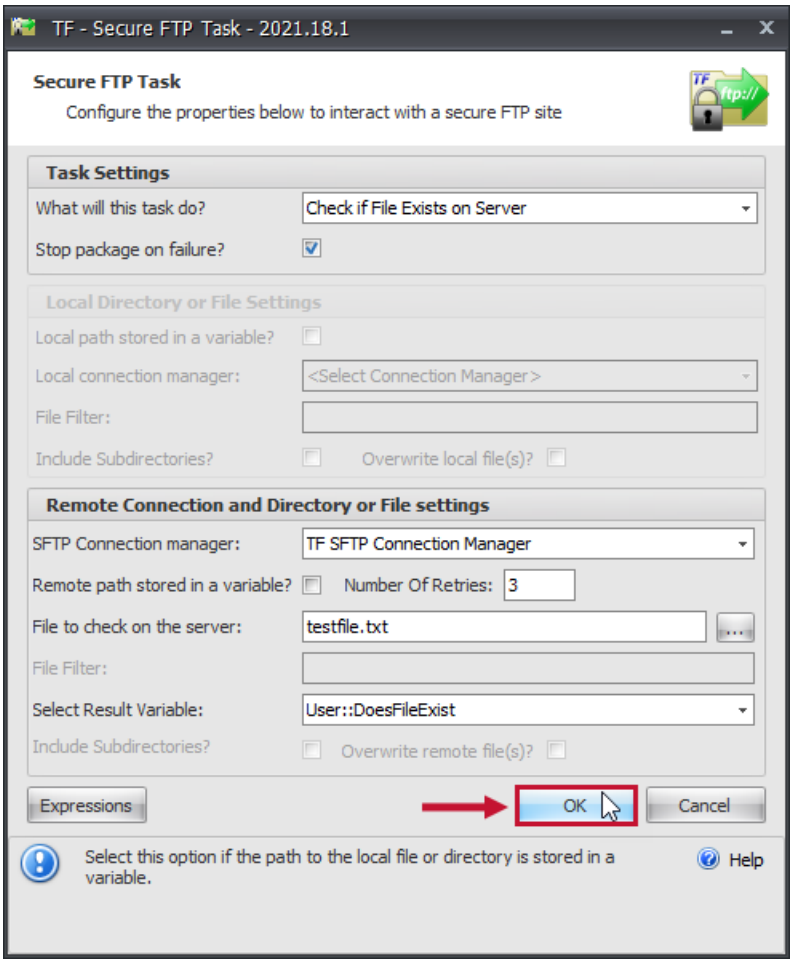
5. Select the connection manager for the task from the applicable drop-down list, and then enter your

specifications for the remaining applicable settings.





6. Select **OK** to finish configuring your Task Factory **Secure FTP Task**.



> **Note:** For more specific Task Factory **Secure FTP Task** tutorials, see the **Tutorials** section.

Samples

See the taskfactory-samples repository on GitHub for Secure FTP Task samples:

- **1_SetupLocalSFTPDirectory.dtsx** - *Use to set up the SFTP Samples*
- **SecureFTP_CheckIfFileExists.dtsx** - *SFTP Task configured to check if a file exists on the server*
- **SecureFTP_DownloadDirectory.dtsx** - *SFTP Task configured to download a directory from the server*
- **SecureFTP_DownloadFileFromServer.dtsx** - *SFTP Task configured to download a file from the server*
- **SecureFTP_RenameFileOnServer.dtsx** - *SFTP Task configured to rename a file on the server*
- **SecureFTP_UploadDirectory.dtsx** - *SFTP Task configured to upload a local directory to the server*
- **SecureFTP_UploadFileToServer.dtsx** - *SFTP Task configured to upload a local file to the server*

Tutorials

STFP Task Examples

Download list of files from SFTP Server

Tutorial walks through how to download a list of files from an SFTP server using the **Task Factory Secure FTP Task**. The initial setup of this process is covered in the KBA article titled Task Factory – SFTP – Get List of Files with Metadata. It is recommended that you review that article before continuing with this example.

Enumerating a ForEachLoop Container for Multiple SFTP Logins

Tutorial for if you want to store **SFTP Login** credentials outside of the SSIS package or want to use multiple credentials to download data from multiple Salesforce accounts.

Get a List of Files with Metadata

Tutorial shows how to use the **Get List of Files with Metadata** action in the **TF Secure FTP Task**. The tutorial covers how to write the metadata to a SQL table.

Video Tutorials

Configure an SFTP Task In Under One Minute

Read Metadata From an SFTP Task Using a Script Task

SFTP Upload Using A File Filter

SFTP Get Files With Metadata

SFTP Download Files Using File Filter

Troubleshooting

Troubleshooting Checklist

Warning: You must use the same version of Task Factory across all your environments (dev, production, qa, etc.) to ensure optimal performance of Task Factory components.

Important: If your organization has changed any of the ciphers, or algorithms within your packages, you may need to upgrade your version of Task Factory. If you aren't fully committed to an upgrade, or want to test an upgrade first, see Testing the upgrade.

Additional Information: If you are encountering any issues with the Secure FTP Connection Manager, or Secure FTP Task, you can begin Creating a log file to help identify the problems you're having.

Troubleshooting Examples

SFTP Task Could Not Authenticate

[ExecuteTask Failed:] Information: Could not authenticate user [Username]

SFTP Task Connection Attempt Failed

[ExecuteTask Failed:] Error: A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond 129.130.13.3:49088

SFTP Task File Doesn’t Exist on the Server

[ExecuteTask Failed:] Error: File /(SFTPFilePath) doesn't exist on the server.

SFTP Task No Error

Issue in which the **TF SFTP Task** is not performing its designated function (i.e. **Download File from Server**, **Upload File to server**, etc.), but it’s showing as a success despite there being no sign the task actually performed the function requested.

Task Factory SharePoint

Last Modified on 24 March 2022

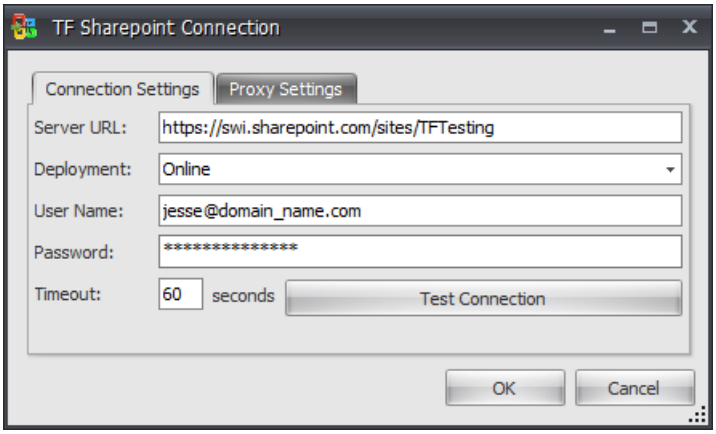
Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Note: Task Factory supports SharePoint versions 2010 through 2019.

SharePoint Connection Manager

The **SharePoint Connection Manager** is used to set up a connection to a SharePoint server. Used with SharePoint Source and SharePoint Destination. The SharePoint Documents Task requires the similar SharePoint Documentation Connection Manager.

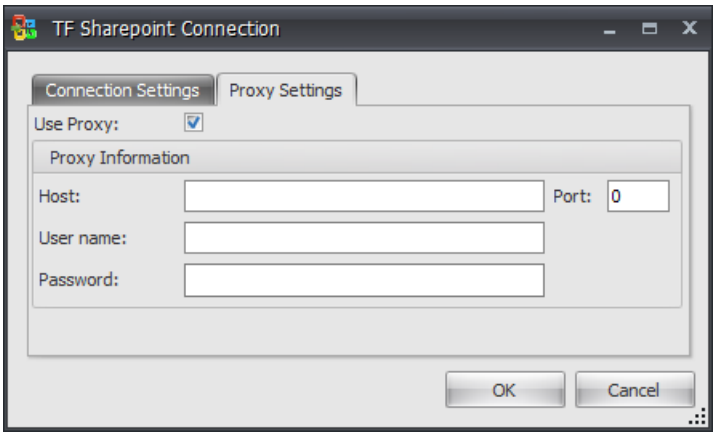
Connection Settings



Option	Description
Server URL	URL of the SharePoint server.
Deployment	You have three options for the deployment: • Premise • Online • Claims Based Authentication
User Name	The user name of the SharePoint server.
Password	Password for the server.

Option	Description
Timeout	Time in seconds before the connection times out if the connection fails.

Proxy Settings

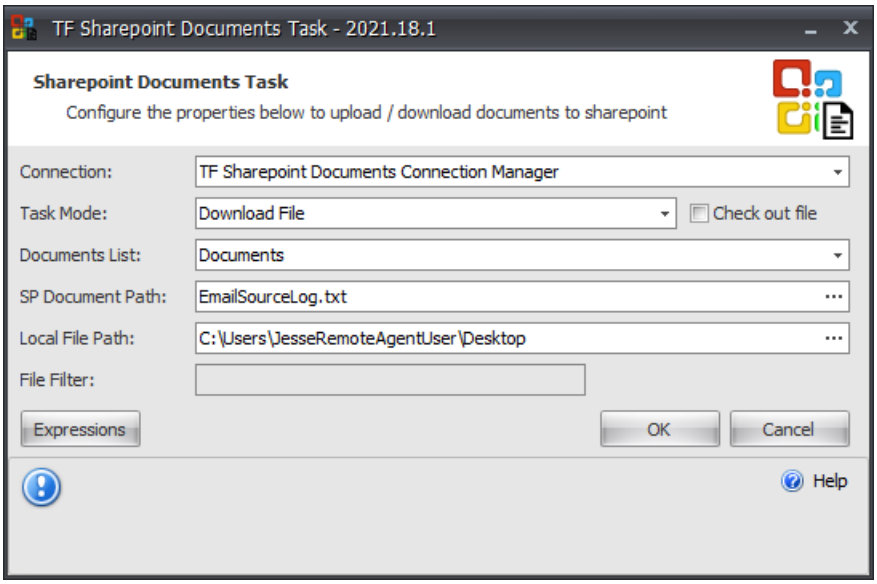


Option	Description
Use Proxy	Selecting this allows you to connect through a proxy. **Note:** If checked you must fill in the proxy information located below the check box.

SharePoint Documents Task

Task Icon	Task Description

Task Icon	Task Description
	The **SharePoint Documents Task** gives you the ability to upload and download documents from a SharePoint documents list in SSIS. See the SharePoint Connection Manager to learn more about setting up the connection manager.

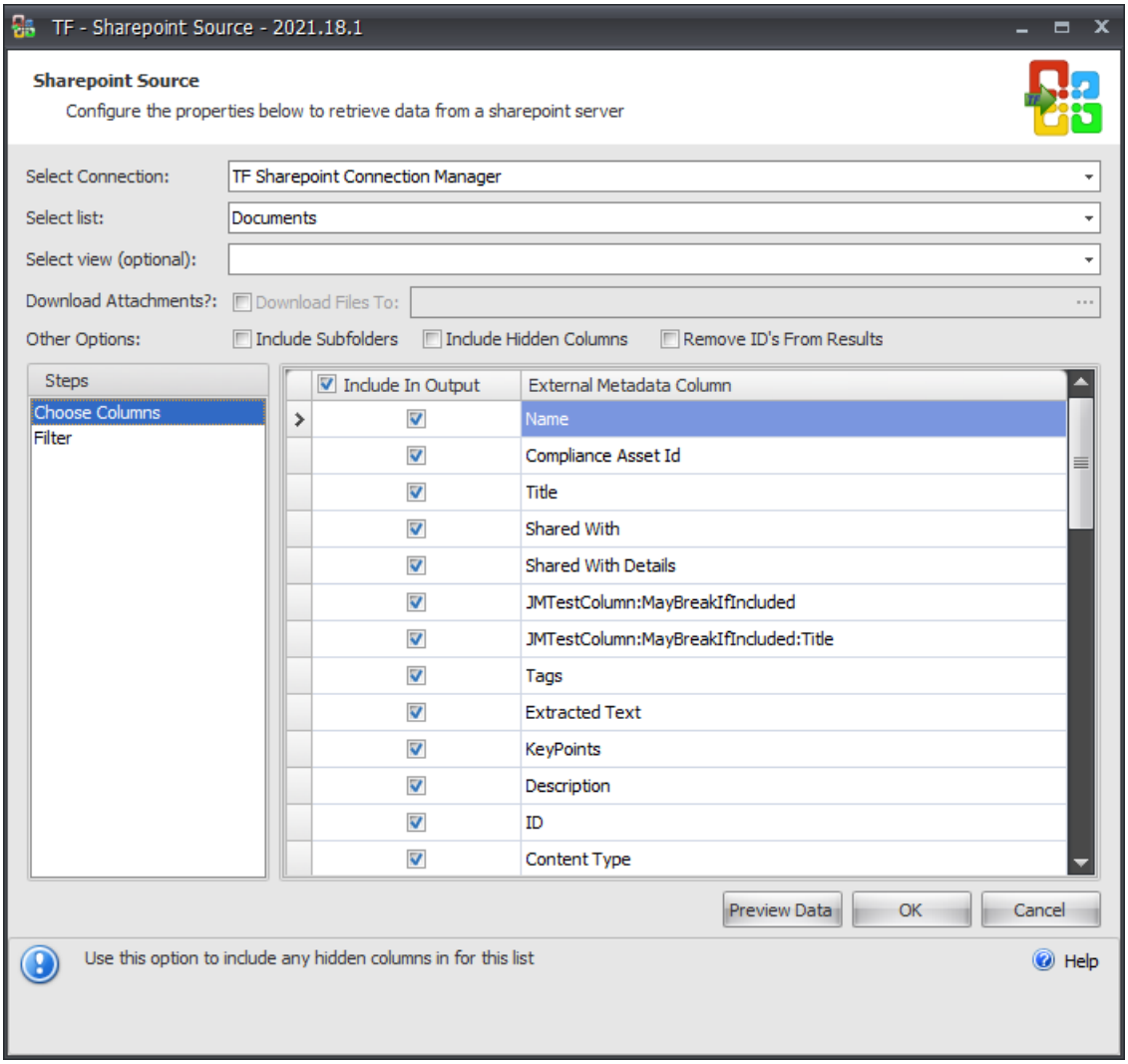


Option	Description
Connection	Select an existing or create a new SharePoint Documents Connection Manager.
Task Mode	There are four modes in which you can use the SharePoint Documents Task: • **Upload File** - Upload a single file to the documents list. • **Download File** - Download a single file to the documents list. • **Upload Directory** - Upload a directory of files from the local machine to a SharePoint documents lists. • **Download Directory** - Download a list of documents from a SharePoint documents list.
Check out / Check in file	When selected, the file(s) are marked as **Checked Out** when downloaded from or **Checked In** when uploaded to Sharepoint to the account executing the package.
Documents List	The SharePoint list that is used in this task. Must be a documents list.
SP Document Path	This is the document subfolder in which to upload / download documents from, or the full path in which to download a file from. Selecting the ellipsis **(...)** opens the **SharePoint Documents Browser**.
Local File Path / Local Directory	The local directory or path in which to download or upload files from or to.

SharePoint Source

Option	Description

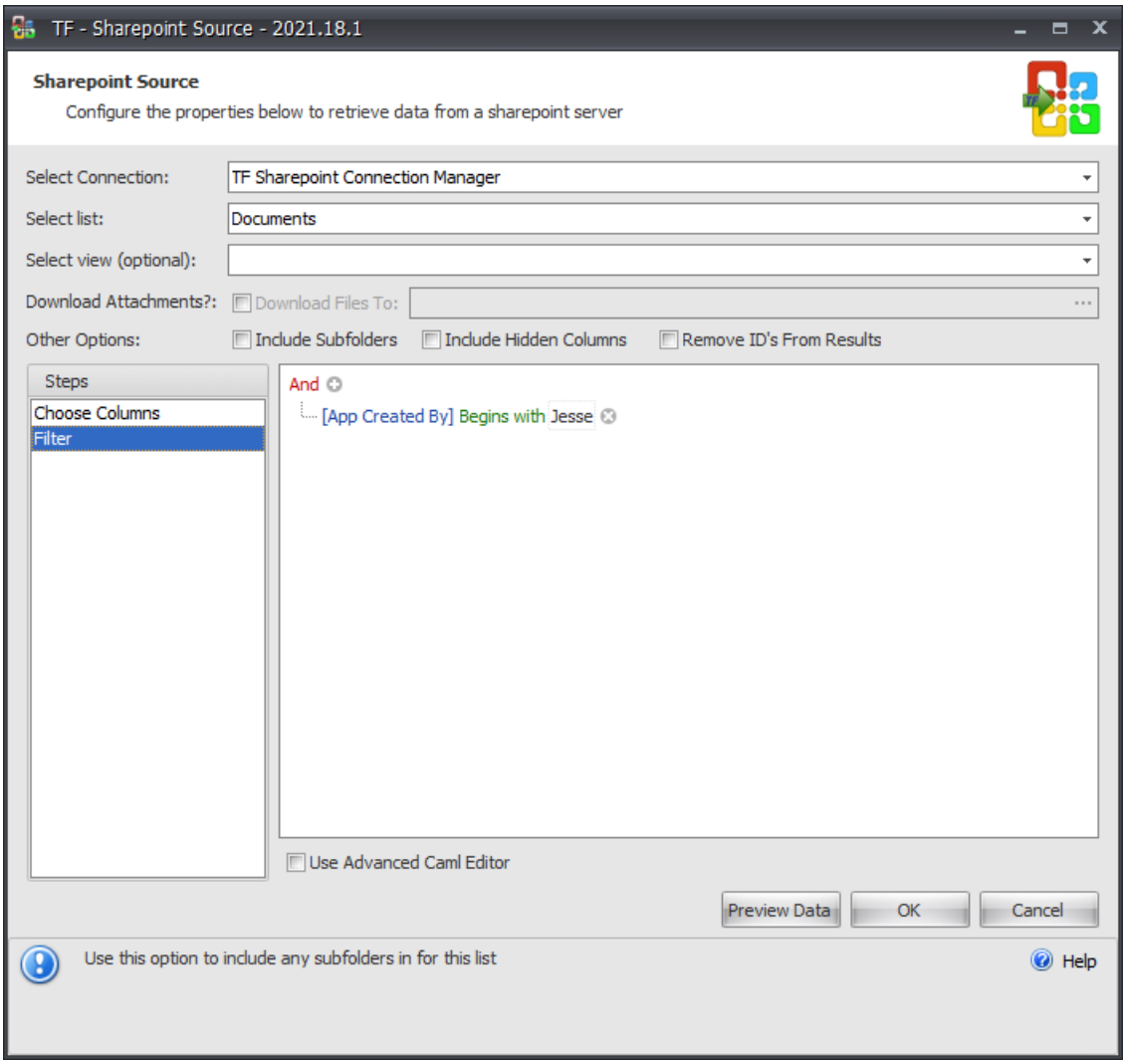
Source Icon	Source Description
	The SharePoint Source is used to retrieve data from a SharePoint object within an SSIS package. See the SharePoint Connection Manager to learn more about setting up the connection manager.



Option	Description
Select Connection	Select an existing SharePoint Connection Manager or choose to create a new one.
Select List	Once your connection manager is set, choose a list from your SharePoint server.
Select View (optional)	Choose the view to use for the chosen list.
Download Attachments?	Enabling this option allows users to download files as attachments within lists. Select the ellipsis in its corresponding window to define the destination path for the file.
Include Subfolders	Choosing this option shows the subfolders that are part of a documents list in the result set.
Include Hidden Columns	Choosing this option retrieves data from all of the columns of a list / view.

Option	Description
Remove ID's from Results	Choosing this option removes identifier data from columns that contain both the identifier and text data of a column.
Steps	• **Choose Columns** - This tab is used to choose which columns from the source are part of the output from the SharePoint Source. • **Filter & Preview Rows** - The filter control allows the user to create custom filters on the data returned from the server.

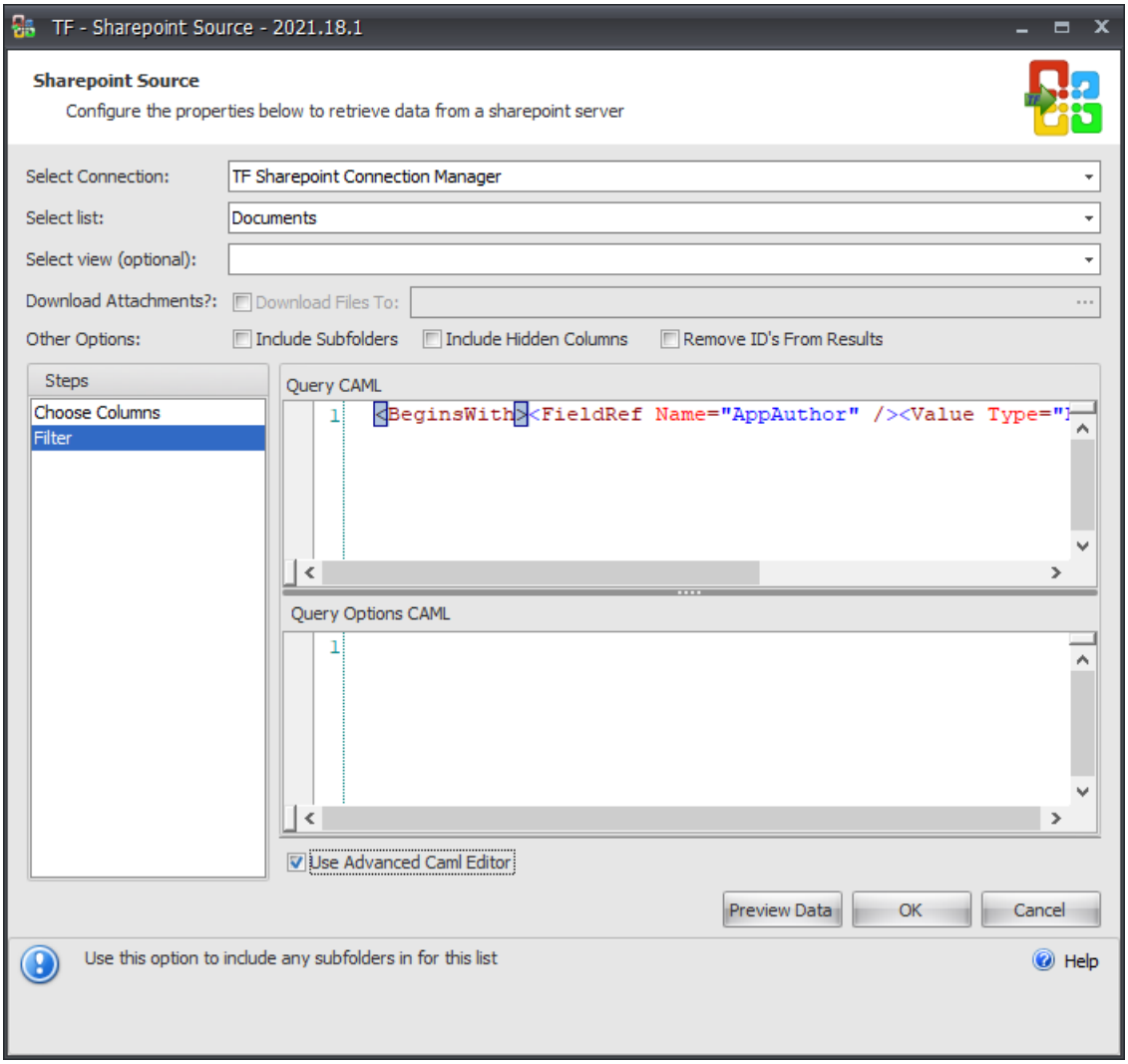
Filter



Users can begin configuring filters by selecting the **+** beside the word **And**. Column filters are configured by selecting the blue hyperlink column name (**[Start Time]**). Operators are configured by selecting the green operator selector (for example, **Is greater than** or **equal to**).

Use Advanced Caml Editor

Selecting the **Use Advanced Caml Editor** option enables the Query CAML and Query Options CAML windows.

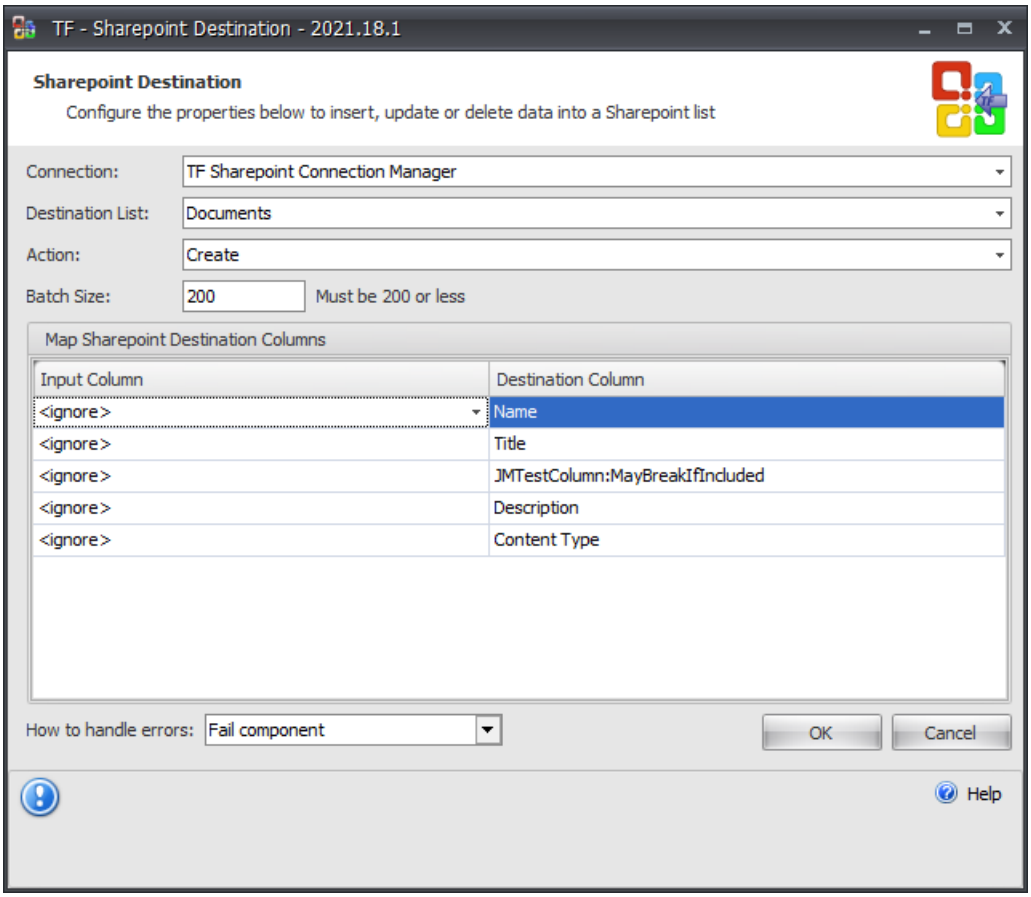


Option	Description
Query CAML	The Sharepoint Source allows users to configure Collaborative Application Marchup Language. CAML is similar to XML and used to perform query operations on lists. **Note:** You must surround queries with tags.
Query Options CAML	Users can add options to their CAML queries. **Note:** You do not need to add the opening and closing query options tag in the query options editor.

Additional Information: For more information about using CAML, please see this Query schema MSDN article .

SharePoint Destination

Destination Icon	Destination Description
	The SharePoint destination is used to send data to a SharePoint object within an SSIS package. See the SharePoint Connection Manager to learn more about setting up the connection manager.



Option	Description
Connection	Select an existing connection manager or choose to create a new one.
Destination List	Choose the destination list from the drop down menu, this is where the data is sent.
Action	Once a list is chosen, you may select from three actions: • **Create** - This action creates rows in the selected list • **Update** - This action updates rows in the selected list. • **Delete** - This action deletes rows in the selected list.
Batch Size	Set the batch size, limited to 200.
Map SharePoint Destination Columns	Once your destination is selected you can map the **Input** and **Destination** columns.

Additional Information: See the Task Factory Error Row Handling article for more information about this functionality.

Task Factory Social Media

Last Modified on 31 March 2022

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Note: These Task Factory social media components require SQL Server version 2012 or higher.

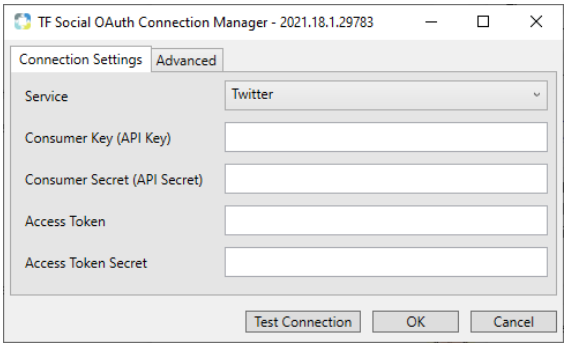
Social OAuth Connection Manager

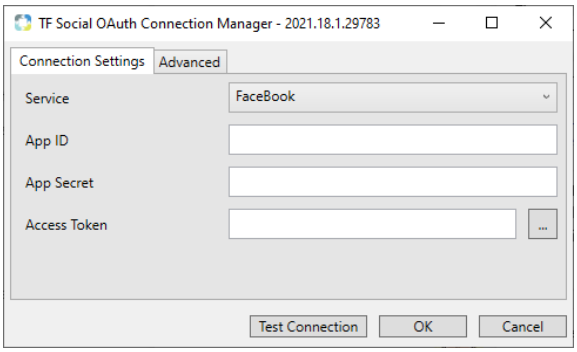
Important: To obtain Api Keys and Tokens, you need to first create an app developer account by visiting Facebook and Twitter's developer websites.

As of January 30, 2018, LinkedIn has been separated into its own connection manager due to their OAuth2 connection requirement. See the LinkedIn OAuth2 Connection Manager tab for more information.

The **Social OAuth Connection Manager** is used to set up a connection to social media sites. This connection manager is used within the Facebook Source and Twitter Source components.

Social OAuth Connection Settings

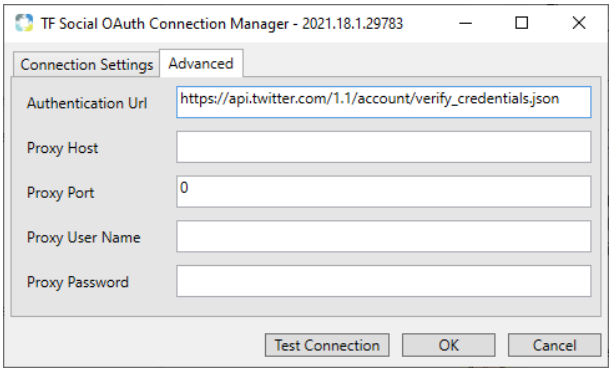


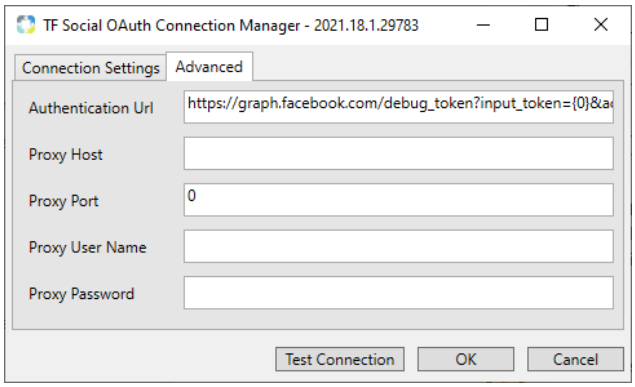


Option	Description
Service	Select this drop-down box to choose which social media to connect to.

Option	Description
(Twitter Options) **Service Consumer Key (API Key)** **Consumer Secret (API Secret)** **Access Token** **Access Secret Token**	Authentication information provided to users at Twitter's app developer site.
(Facebook Options) **App ID** **App Secret** **Access Token**	Authentication information provided to users at Facebook's app developer site.

Social OAuth Advanced



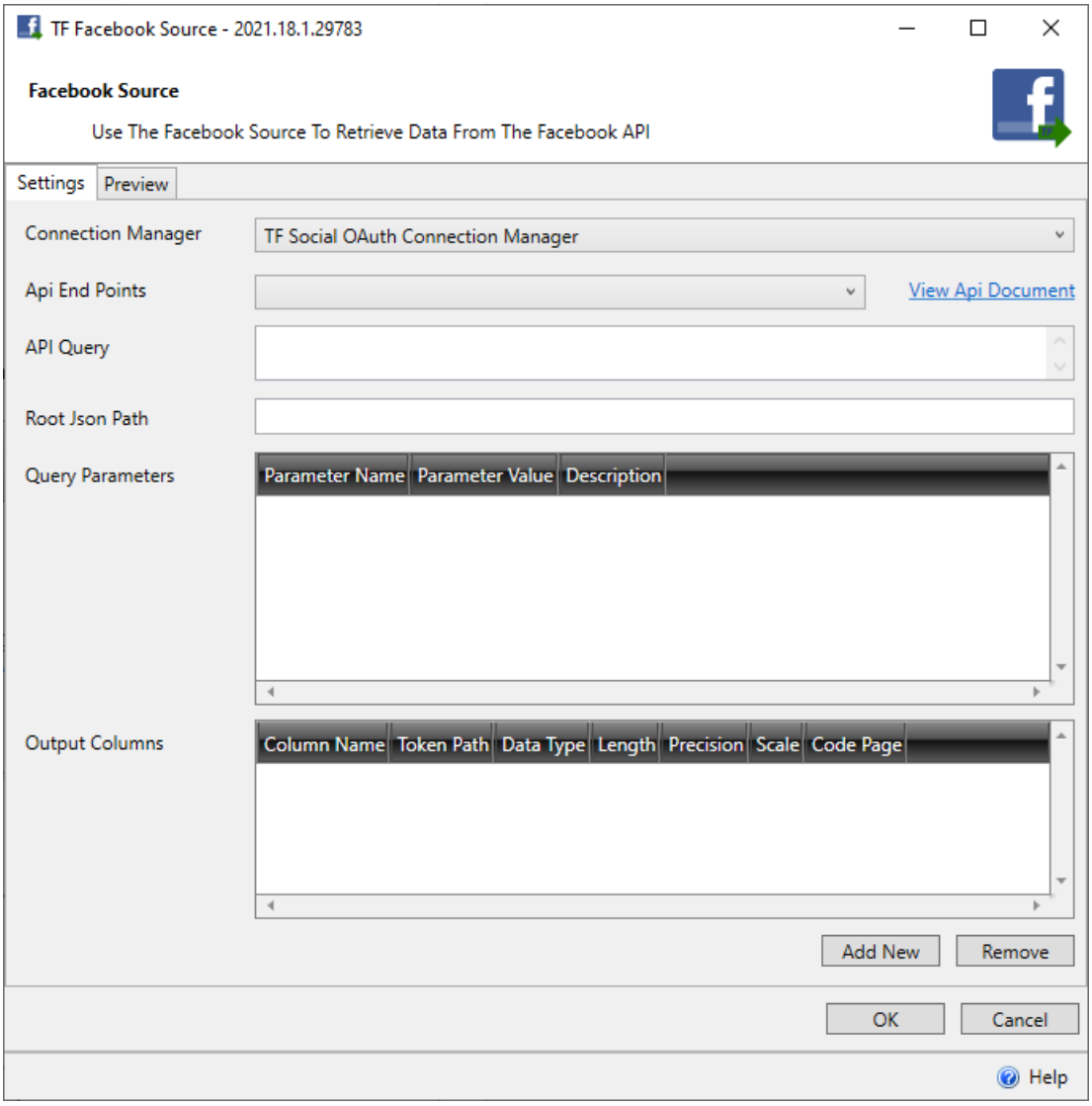


Option	Description
Authentication Url	The Url (with version) of the social media site the user wants to connect to.
Proxy Host	The ip address of the proxy the user wishes to connect through.
Proxy Port	The port of the proxy being used.
Proxy User Name	Users should enter their user name for the proxy if it is required.
Proxy Password	Users should enter their password for the proxy if it is required.

Facebook Source

Source Icon	Source Description
	The **Facebook Source** is used to extract data such as your user profile, status updates, friends, likes, and comments within an SSIS package. See the Social OAuth Connection Manager to learn more about setting up this component's connection manager.

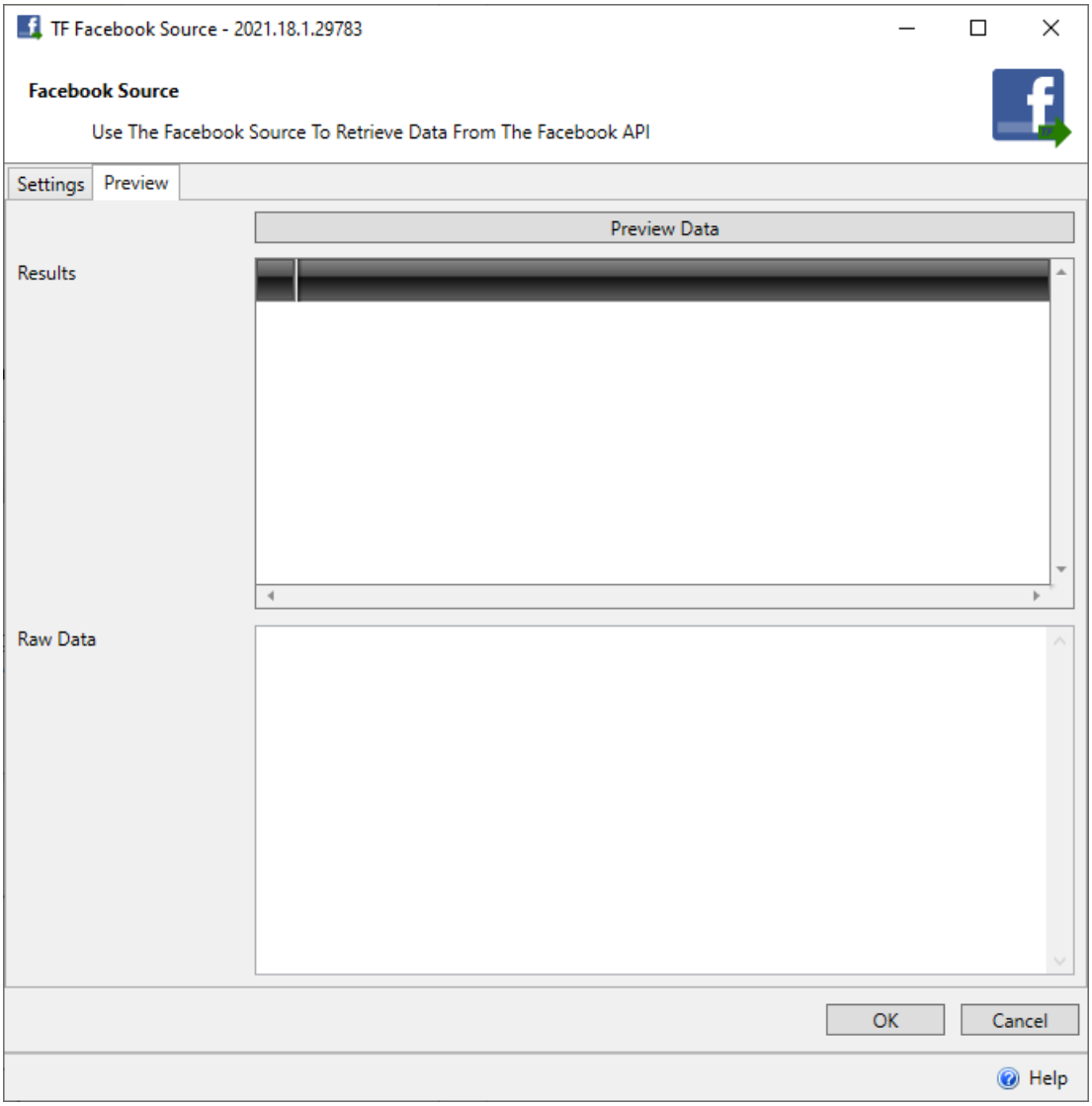
Facebook Source Settings Tab



Option	Description
Connection Manager	In this field, users can create a new connection manager or select a pre-existing one.
Api End Points	In this field, users select the endpoint they wish to connect to (such as User Profile or User Groups.) Selecting an endpoint auto-populates that field's parameters and commonly used Output Columns.
API Query	This field defines the query to be used in the API. Although several end points have been pre-configured, users can type in or paste custom calls that are not included with the component. These calls can be found by selecting the **View Api Document** link and adding the reference to the end of the API Query. Example: https://graph.facebook.com/{user-id}/groups
Root Json Path	Because Facebook returns the raw data in Json format, users need to define the root path when creating custom queries.

Option	Description
Query Parameters	User-defined parameters (such as id) based on the API query. Parameters can be set dynamically by entering the variable name. Example: <@User::ScreenName>
Output Columns	In this window, users can add and remove columns and define their name, token path (defined within the raw data), data type, length, precision, scale, and code page.

Facebook Source Preview Tab



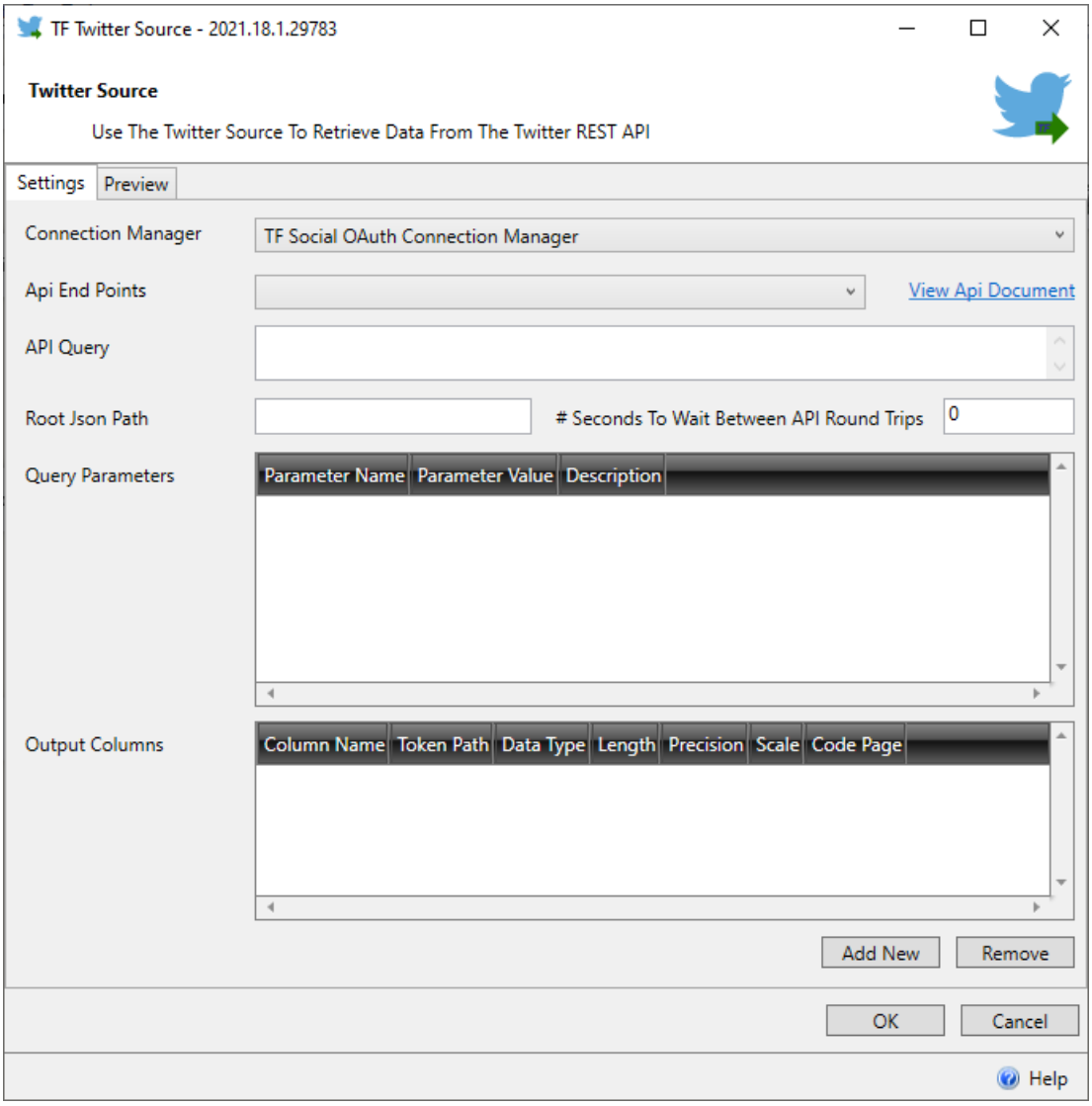
Option	Description
Preview Data	Selecting this button returns the results and raw data based on the user-defined settings.

Option	Description
Results	In this window, users can see a preview of the first few rows of data based on their configuration.
Raw Data	This window displays the Json data in raw form. Developers can use this to help determine root and token paths.

Twitter Source

Source Icon	Source Description
	The **Twitter Source** is used to extract data such as status updates (known as Tweets), followers, hashtags, and retweeted or favorited tweets within an SSIS package. See the Social OAuth Connection Manager to learn more about setting up this component's connection manager.

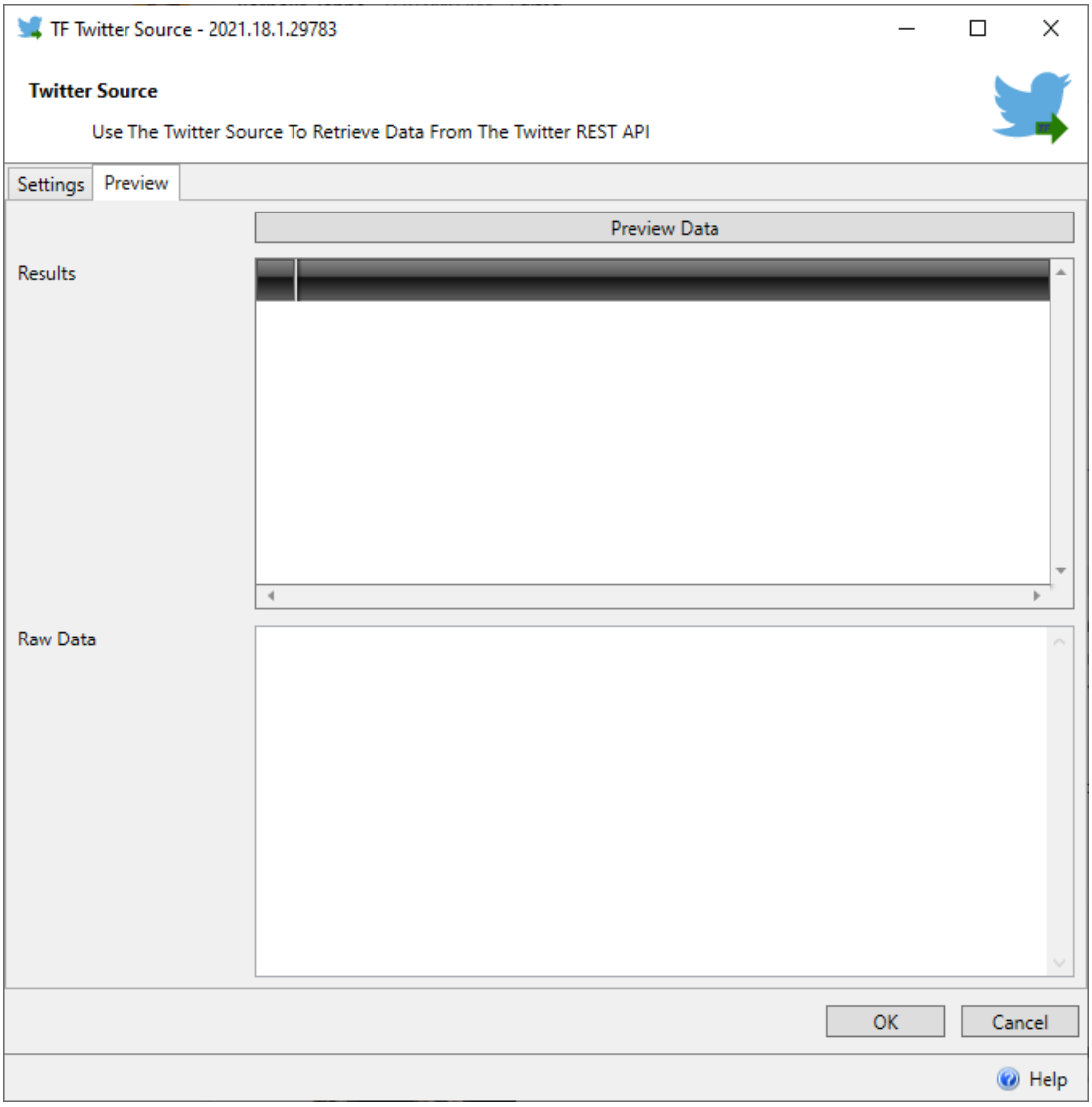
Twitter Source Settings Tab



Option	Description
Connection Manager	In this field, users can create a new connection manager or select a pre-existing one configured earlier.
Api End Points	In this field, users select the Api they wish to connect to (such as User Timeline, Followers/List, etc.) Selecting an endpoint auto-populates that field's parameters and commonly used Output Columns.
API Query	This field defines the query to be used in the API. Although several end points have been pre-configured, users can type in or paste custom calls that are not included with the component. These calls can be found by selecting the **View Api Document** link, and copy & pasting the Resource URL located on the page of the selected API.
Root Json Path	Because Twitter returns the raw data in Json format, users need to define the root path when creating custom queries. This can (usually) be found in the **View Api Document** link under **Example Result**.

Option	Description
# Seconds to wait between API round trips	For security reasons, Twitter places a 200 count limit per call at runtime and only 15 calls can be performed within 15 minutes. This limitation can cause a **too many requests** error when querying a large dataset such as a user's followers which has 10,000 users. To avoid this error, users can set a timer that pauses between calls.
Query Parameters	User-defined parameters (such as screenname or count number.) Parameters can be set dynamically by entering the variable name. Example: <@User::ScreenName>
Output Columns	In this window, users can add and remove columns and define their name, token path (defined within the raw data), data type, length, precision, scale, and code page.

Twitter Source Preview Tab



Option	Description

Option	Description
Preview Data	Selecting this button returns the results and raw data based on the user-defined settings.
Results	In this window, users can see a preview of the first few rows of data based on their configuration.
Raw Data	This window displays the Json data in raw form. Developers can use this to help determine a root and token path.

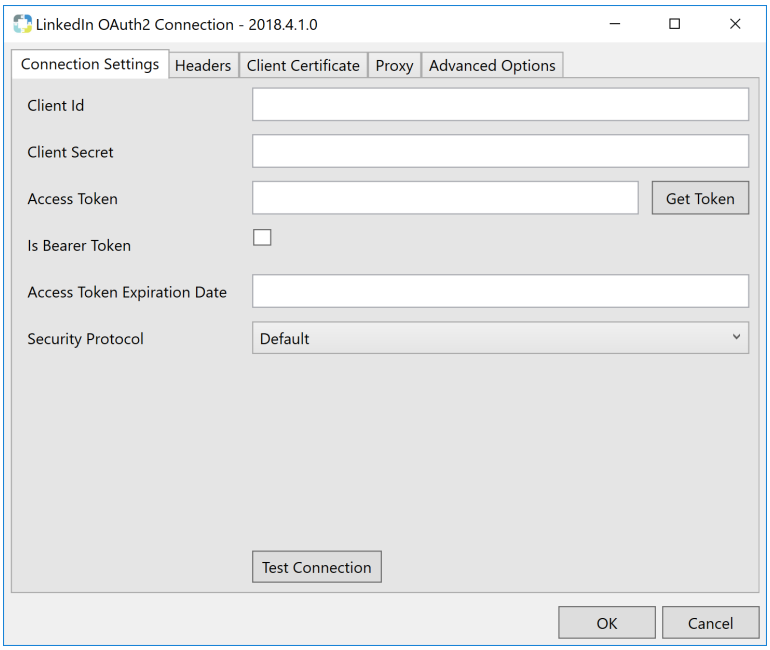
LinkedIn OAuth2 Connection Manager

Unsupported: The LinkedIn component was deprecated in version 2019.4.1 and is no longer supported.

The LinkedIn OAuth2 Connection manager is used with the LinkedIn Source data flow component.

Additional Information: Before using the LinkedIn Source, users must first create an app to obtain a Client ID and Client Secret. This can be done by visiting LinkedIn Developers.

LinkedIn OAuth2 Connection Settings

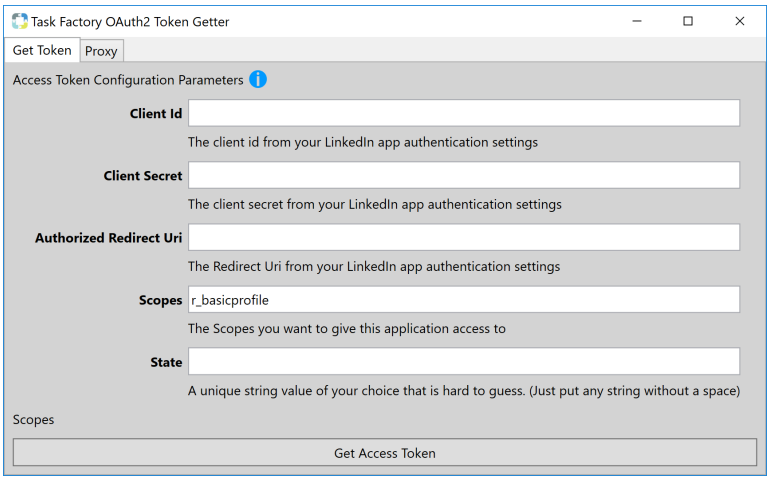


Option	Description
Client Id	Enter the Client ID provided by LinkedIn.

Option	Description
Client Secret	Enter the Client Secret provided by LinkedIn. **Note:** Client ID and Client Secret are located in the My Apps - Authentication Keys section at LinkedIn Developers.
Access Token	LinkedIn's OAuth2 connection requires an access token to make API requests. Select the **Get Token** button to open the **Token Getter** window.
Is Bearer Token	LinkedIn requires access tokens be designated as bearer tokens. This option is automatically selected after the user has created an access token in the previous step.
Access Token Expiration Date	Identifies the date and time a user's access token expires. This field automatically populates when the user completes the steps in the **Get Token** window.
Test Connection	Tests the connection after the connection manager has been configured.

OAuth2 Get Token

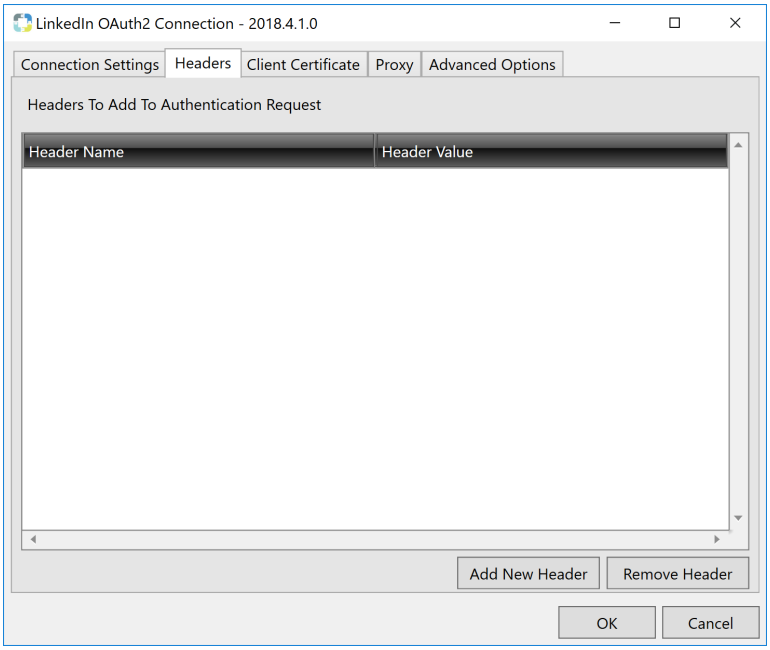
The **Token Getter** window displays when users select the **Get Access Token** button.



Option	Description
Client ID	The client ID entered in the connection settings.
Client Secret	The client secret entered in the connection settings.
Authorized Redirect Url	This is a required field. It must match a redirect URL configured within LinkedIn's developer website (located in the My Apps - OAuth 2.0 section).

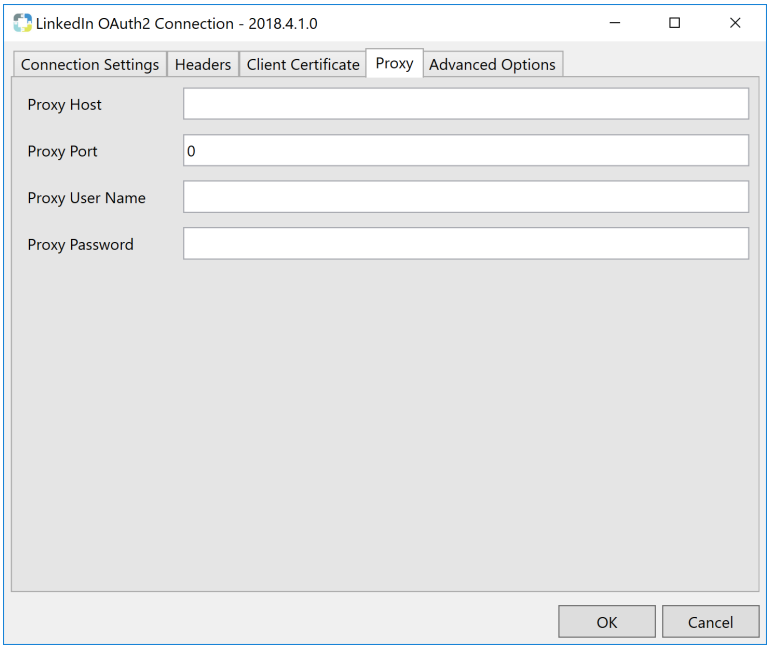
Option	Description
Scopes	Identifies the default application permissions. Users can add the following scopes: • r_basicprofile • r_emailaddress, • rw_company_admin • w_share Multiple scopes should be separated by a space.
State	A unique string value similar to a password. The state can be any word, number, or character combination that does not contain a space.
Get Access Token	After all fields have been completed, select the **Get Access Token** button to receive it. This opens a small browser within the window. The user must authenticate to LinkedIn and allow access to the API to complete the process.

LinkedIn OAuth2 Headers



Option	Description
Headers	LinkedIn requires a connection header called x-li-format with the value Json. This is automatically entered after the user completes the steps in the **Get Token** window.
Add New Header	Creates a new header.
Remove Header	Removes a selected header.

LinkedIn OAuth2 Proxy

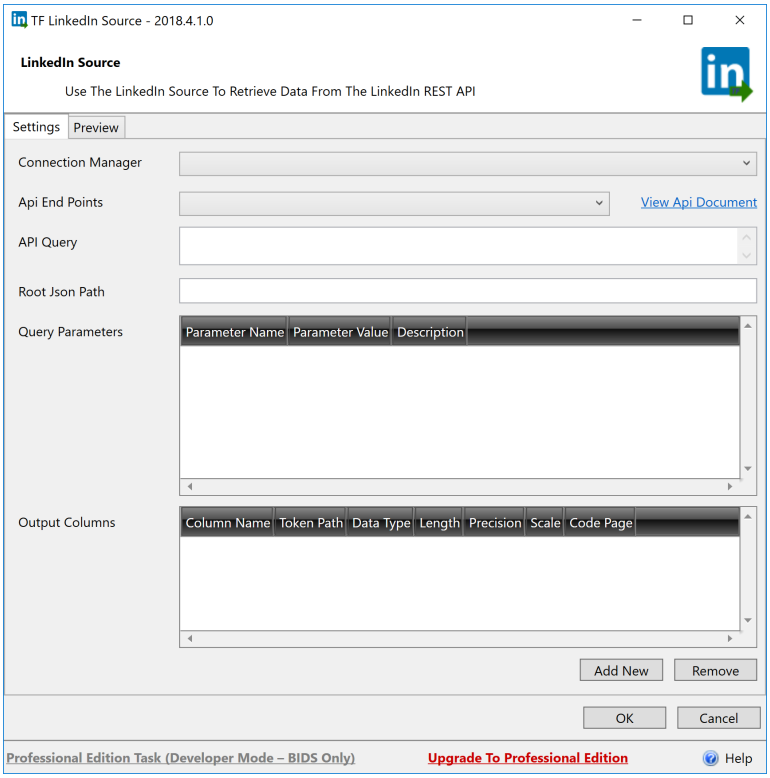


Option	Description
Proxy Host	Identifies the proxy address.
Proxy Port	Identifies the port used by the proxy.
Proxy User Name	Enter the user name for proxies that require user authentication.
Proxy Password	Enter the password for proxies that require user authentication.
Auth Scheme	Identifies which authentication scheme the proxy is using.

LinkedIn Source

Unsupported: The LinkedIn component was deprecated in version 2019.4.1 and is no longer supported.

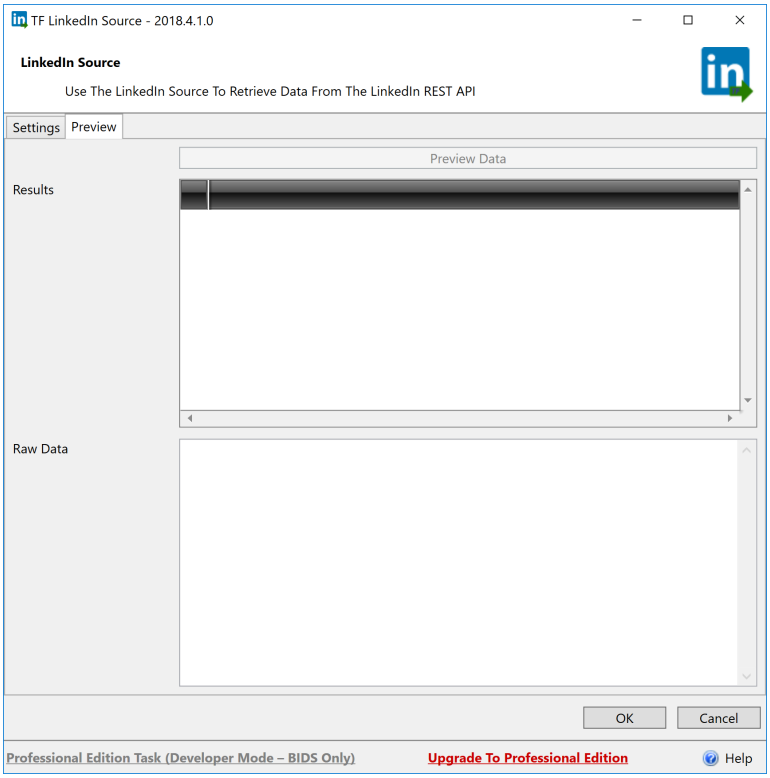
Source Icon	Source Description
	The **LinkedIn Source** is used to extract data such as name, status updates, connections, and headlines within an SSIS package. See the LinkedIn Social OAuth2 Connection Manager to learn more about setting up this component's connection manager.



Option	Description

Option	Description
Connection Manager	In this field, users can create a new connection manager or select a pre-existing one.
Api End Points	In this field, users select the Api they wish to connect to (**Get My Profile** and **Get Member Profile**.) Selecting an endpoint auto-populates that field's parameters and commonly used Output Columns.
API Query	This field defines the query to be used in the API. Although several end points have been pre-configured, users can type in or paste custom calls that are not included with the component. **Note:** Unfortunately, LinkedIn does not provide a comprehensive list of API calls so users have to search the internet if they wish to customize this field.
Root Json Path	Because LinkedIn returns the raw data in Json format, users need to define the root path when creating custom queries.
Query Parameters	User-defined parameters (such as id) based on the API query. Parameters can be set dynamically by entering the variable name. Example: <@User::ScreenName>
Output Columns	In this window, users can add and remove columns and define their name, token path (defined within the raw data), data type, length, precision, scale, and code page.

LinkedIn Source Preview



Option	Description
Preview Data	Selecting this button returns the results and raw data based on the user-defined settings.
Results	In this window, users can see a preview of the first few rows of data based on their configuration.
Raw Data	This window displays the Json data in raw form. Developers can use this to help determine a root and token path.

Task Factory XML

Last Modified on 31 March 2022

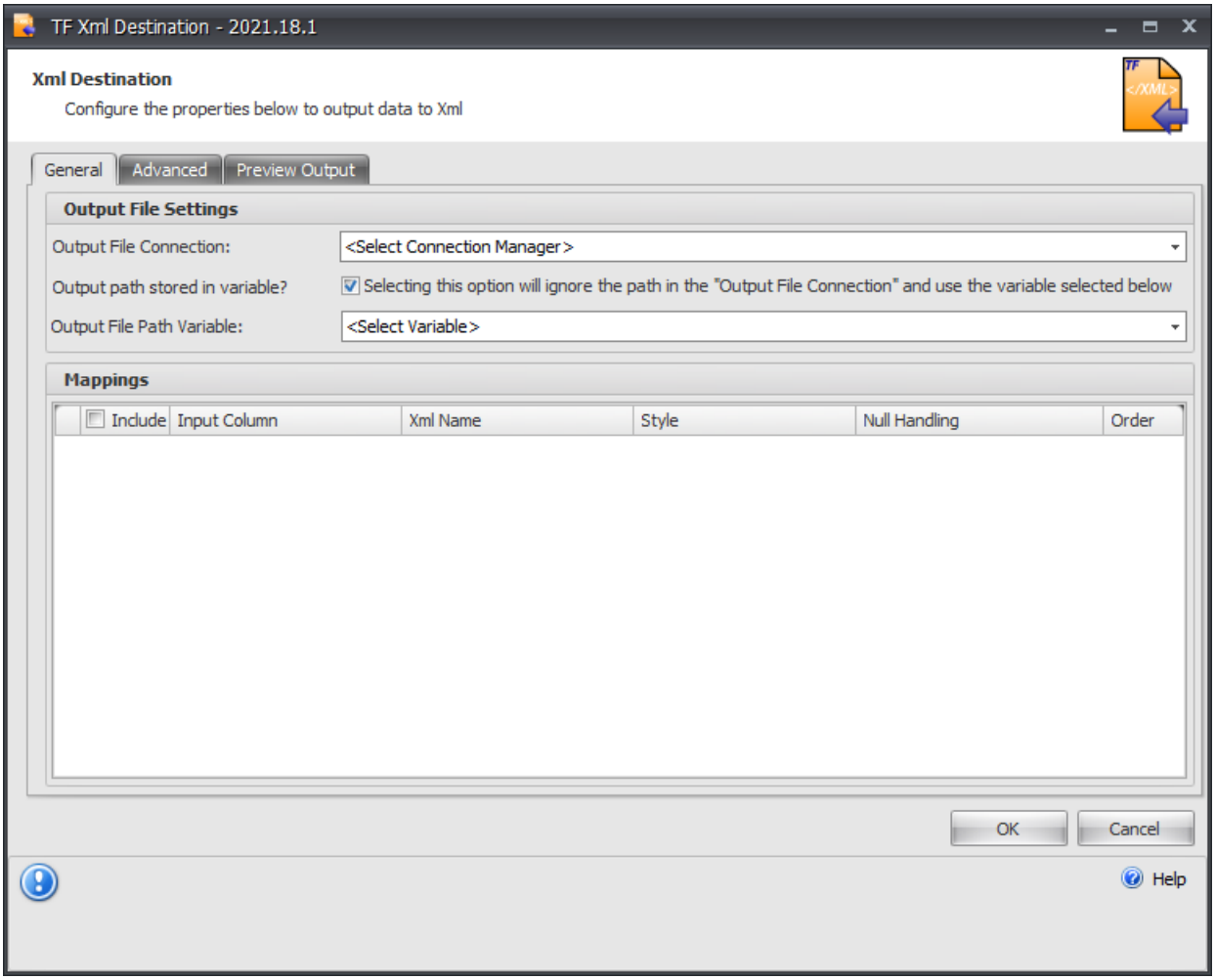
Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

XML Destination

Important: XML Destination is included in all releases for backwards compatibility, but has been discontinued for future development. It's recommended that users choose the XML Generator Transform or the XML Output Destination.

Destination Icon	Destination Description
	The **XML Destination** is used to export XML data within an SSIS package.

General



Output File Settings

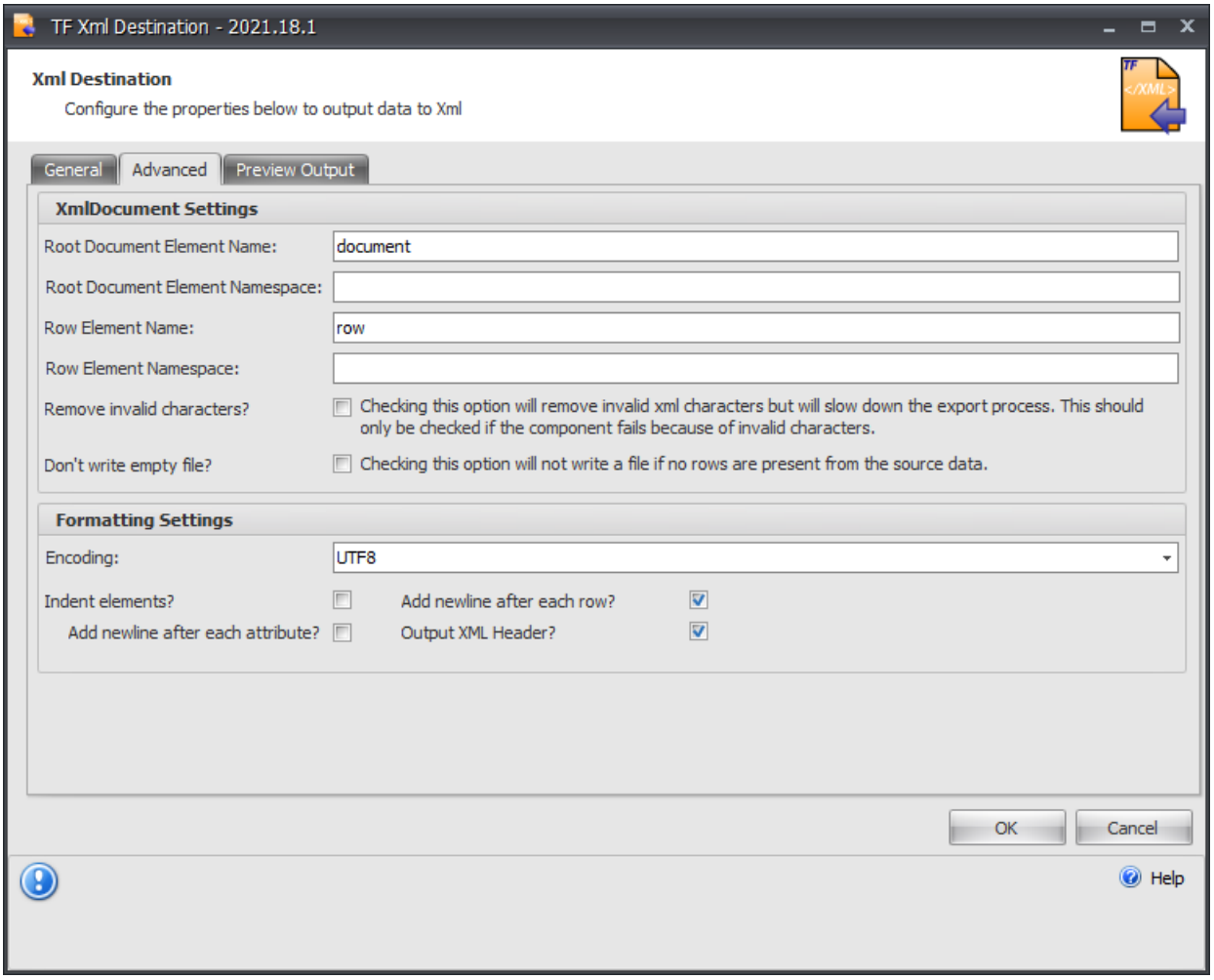
Option	Description
Output File Connection	The name of the file connection manager where the file is output. You can use either a flat file or file connection type.
Output Path Stored In Variable	Check this option to tell the XML Destination that the path of the outputted XML file is contained in the variable selected in **Output File Path Variable**.
Output File Path Variable	The name of the variable that contains the path where the XML file is output.

Mappings

Option	Description
Include	Check the box in this column if the data from the respective source column is included in the outputted XML file.
Input Column	Read only column that displays the source column that the row is for.

Option	Description
XML Name	The name of the element or attribute that is output to the XML file.
Style	• Attribute • Element • Element with cdata • Element with XML data
Null Handling	Define how to handle null source data: • **Output empty attribute / element** - Outputs empty element or attribute. • **Don't Output attribute / element** - The XML data is ignored and not output. • **Replace null data with user defined value** - Replaces null data with a user defined value.
Order	The order that the data is output to the XML file.

Advanced



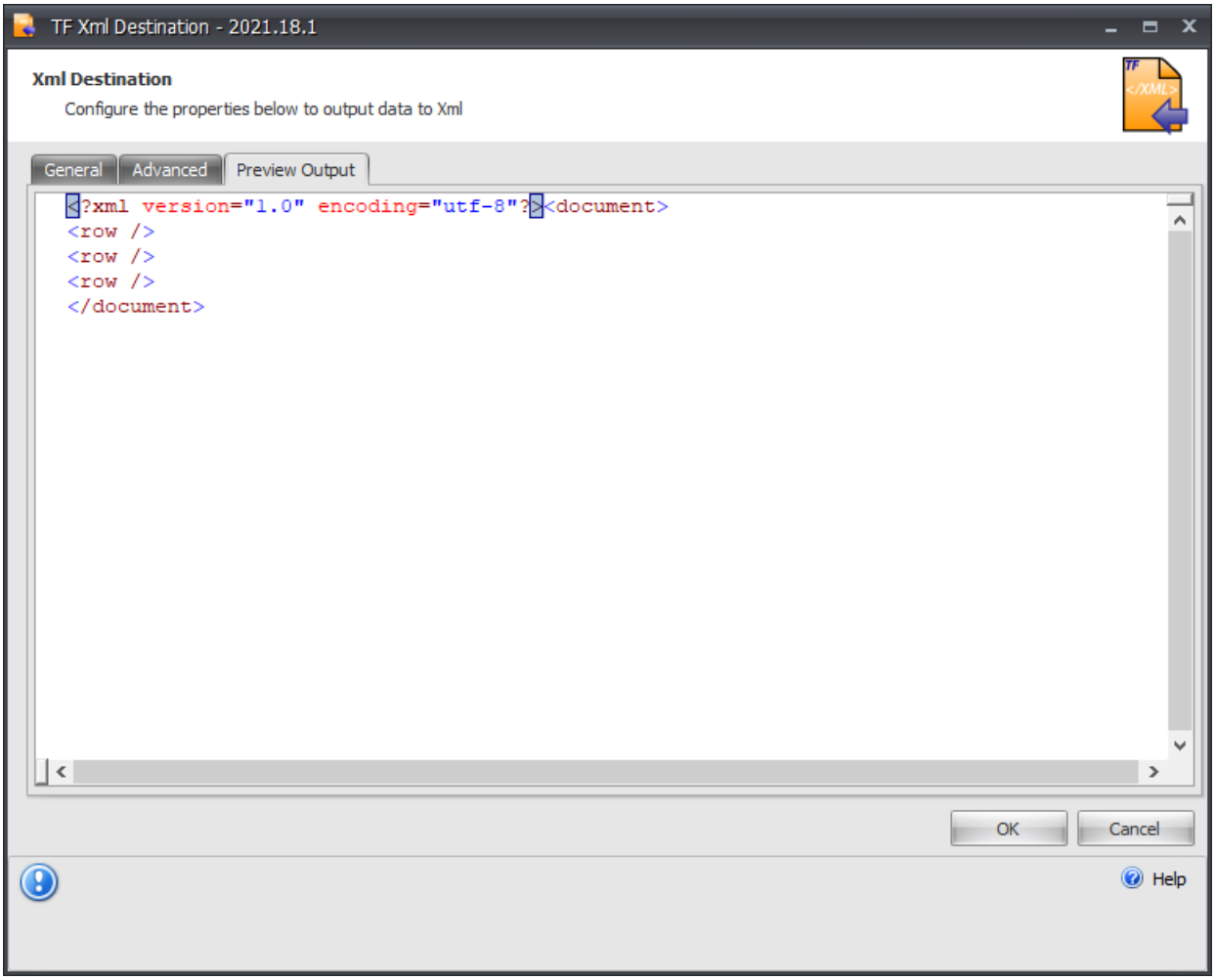
XMLDocument Settings

Option	Description
Root Document Element Name	The name of the root XML element.
Root Document Element Namespace	The name of the root XML element namespace.
Row Element Name	The name of the row XML element.
Row Document Element Namespace	The name of the row XML element namespace.
Remove Invalid Characters?	Selecting this option removes invalid characters from the source data before exporting it to XML.
Don't Write Empty File?	Selecting this option does not write anything when there is no output from the source.

Formatting Settings

Option	Description
Encoding	The encoding of the outputted XML file. • UTF-8 • Unicode • UTF-32 • ASCII
Indent Elements?	This option indents and adds a newline to each element of a row of data.
Add newline after each attribute	This option adds a newline to each attribute of a row of data. **Note:** This is only available with indent elements.
Add newline after each row	This option adds a newline after each row of data. **Note:** This is only available if indent elements is not selected.
Output XML Header	Determines whether or not the XML header is output with the XML file.

Preview Output



This tab previews the data that is output in the XML file.

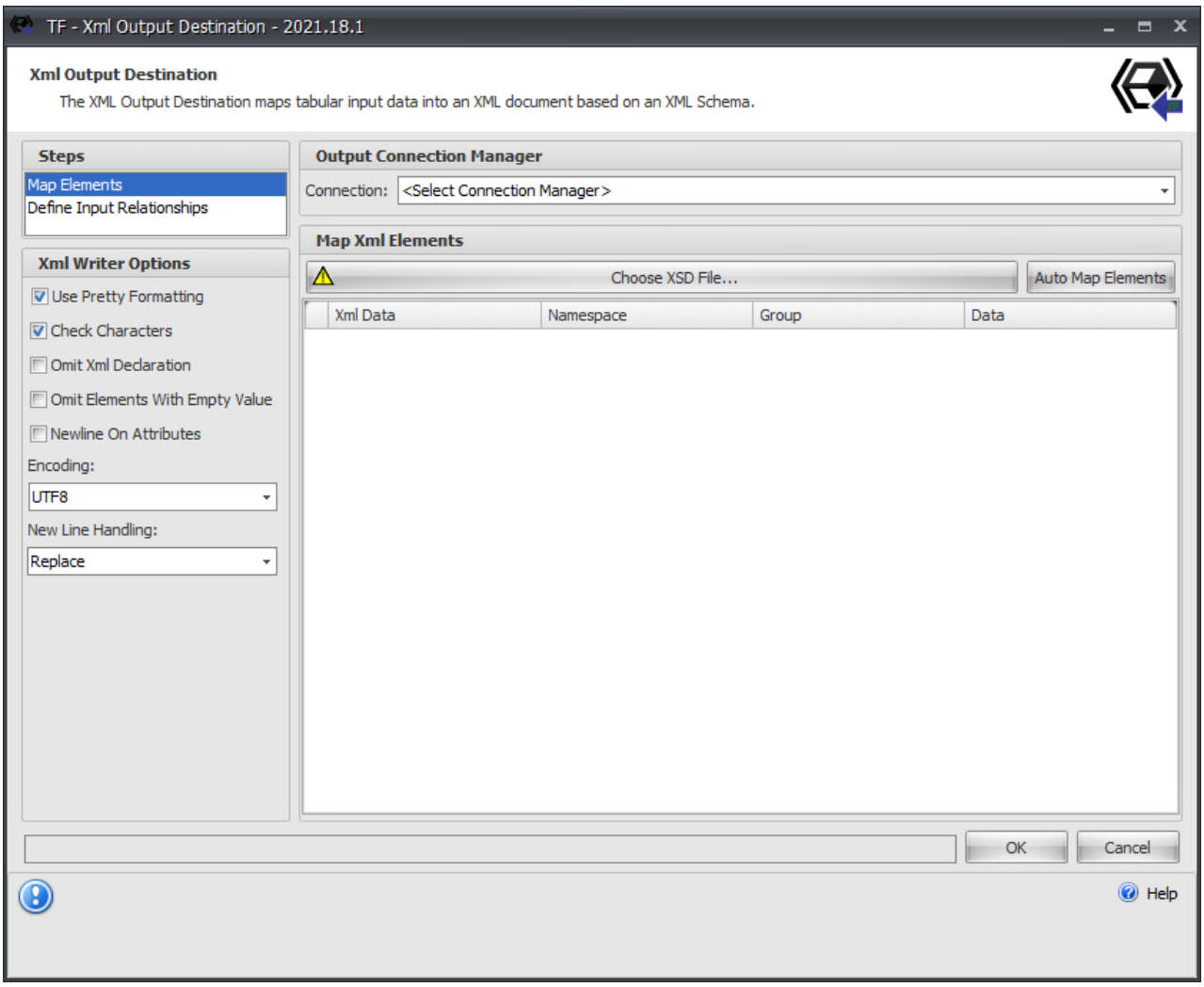
XML Output Destination

Destination Icon	Destination Description

Destination Icon	Destination Description
	The **XML Output Destination** is used to generate and write XML from one or more source inputs using an XSD mapped to input columns. Similar to the XML Generator Transform, users should opt for the XML Output Destination when writing XML files that use large datasets.

Requirements

- Valid XSD file
- When using more than one source, all inputs must be sorted



Steps

Option	Description
Map Elements	In this step users can map columns to elements in the right.

Option	Description
Define Input Relationships	In this step the right side section changes to **Map Input Key Columns**.

Output Connection Manager

Option	Description
Connection	Create or choose a connection to the output XML file.

Map XML Elements

After selecting the XSD file to use, the field populates.

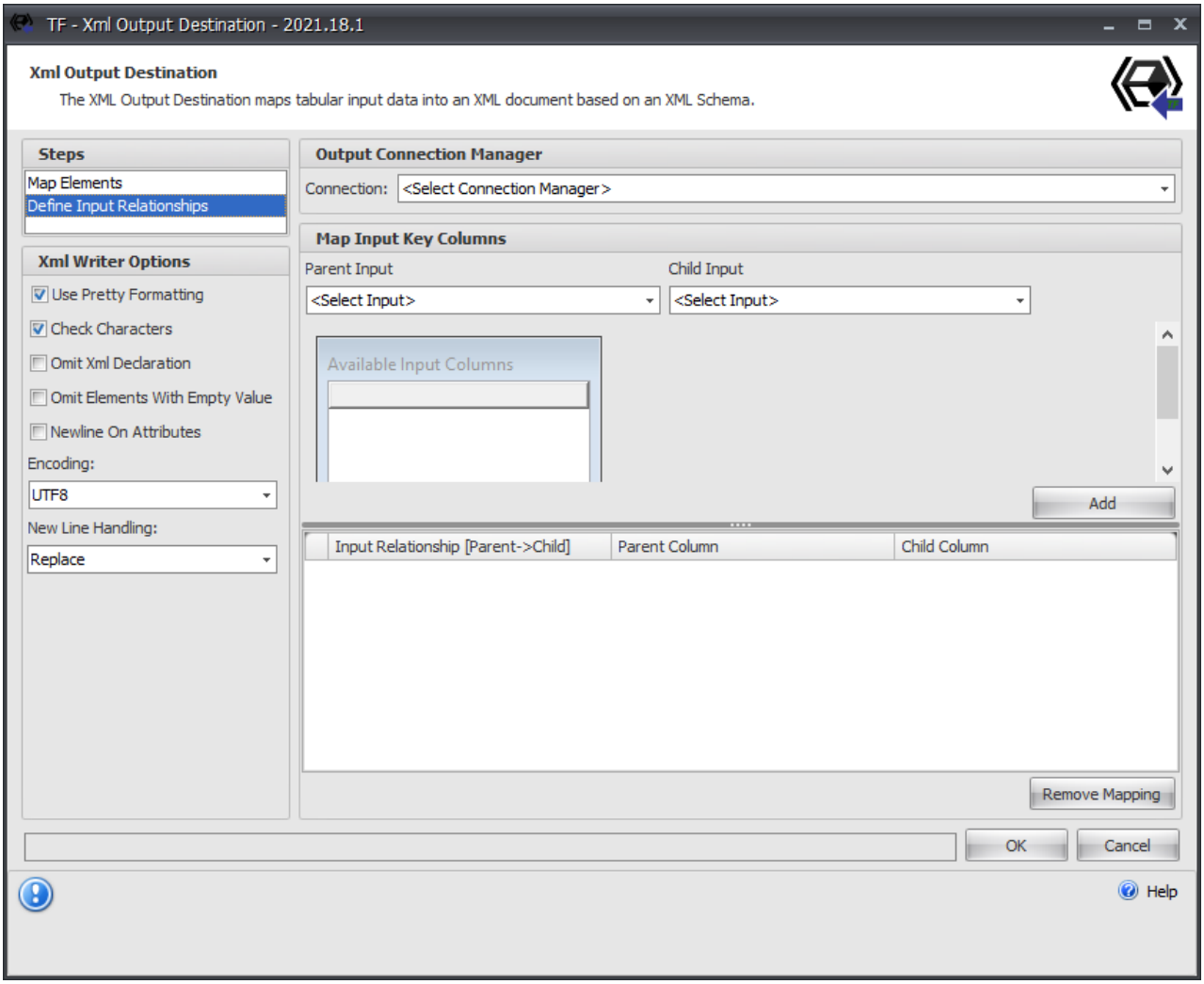
Option	Description
Choose XSD File	Allows users to select an XSD file.
Auto Map Elements	Selecting this auto maps elements and/or attributes to inputs with the same name.
XML Data	This is a read only view of the elements and attributes from the XSD.
Group	This is where you assign sequence elements an input to group the sequence on.
Data	The column for users to map XSD attributes and elements to input columns.

Option	Description

XML Writer Options	• **Use Pretty Formatting** - Toggle for output to use line breaks and indentations or a single line .
	
	Warning: : Pretty Formatting can increase file sizes.
	
	• **Check Characters** - Enabling this option fails the component when illegal characters are output.
	• **Omit XML Declaration** - The opening XML declaration. is removed.
	• **Newline On Attributes** - All attributes included in the output will be printed on a new line.
	• **Encoding** - Select from the following options the output encoding:
	  ◦ UTF8
	  ◦ Unicode
	  ◦ UTF without BOM (Byte Order Mark)
	• **New Line Handling**
	  ◦ **Entitize** - Replaces new line characters that would not be otherwise preserved with character entities.
	  ◦ **Replace** - Replaces new line characters with \r\n.
	  ◦ **None** - Leaves the input unchanged.

Additional Information: For more information to determine which handling should be used, see Microsoft's help file XmlWriterSettings.NewLineHandling Property.

Define Input Relationships



If multiple inputs are attached to the XML Output Destination and the selected XSD has parent-child sequences, users need to define these input relationships. This can be done by selecting the **Parent Input** and **Child Input**, dragging the relational key between the two, and then selecting the **Add** button.

Guid Key Sort Algorithm

Identifies the method in which the input data is sorted when using Guid as relationship keys.

> **Note:** This option only appears when Guid keys are used.

The following methods can be used:

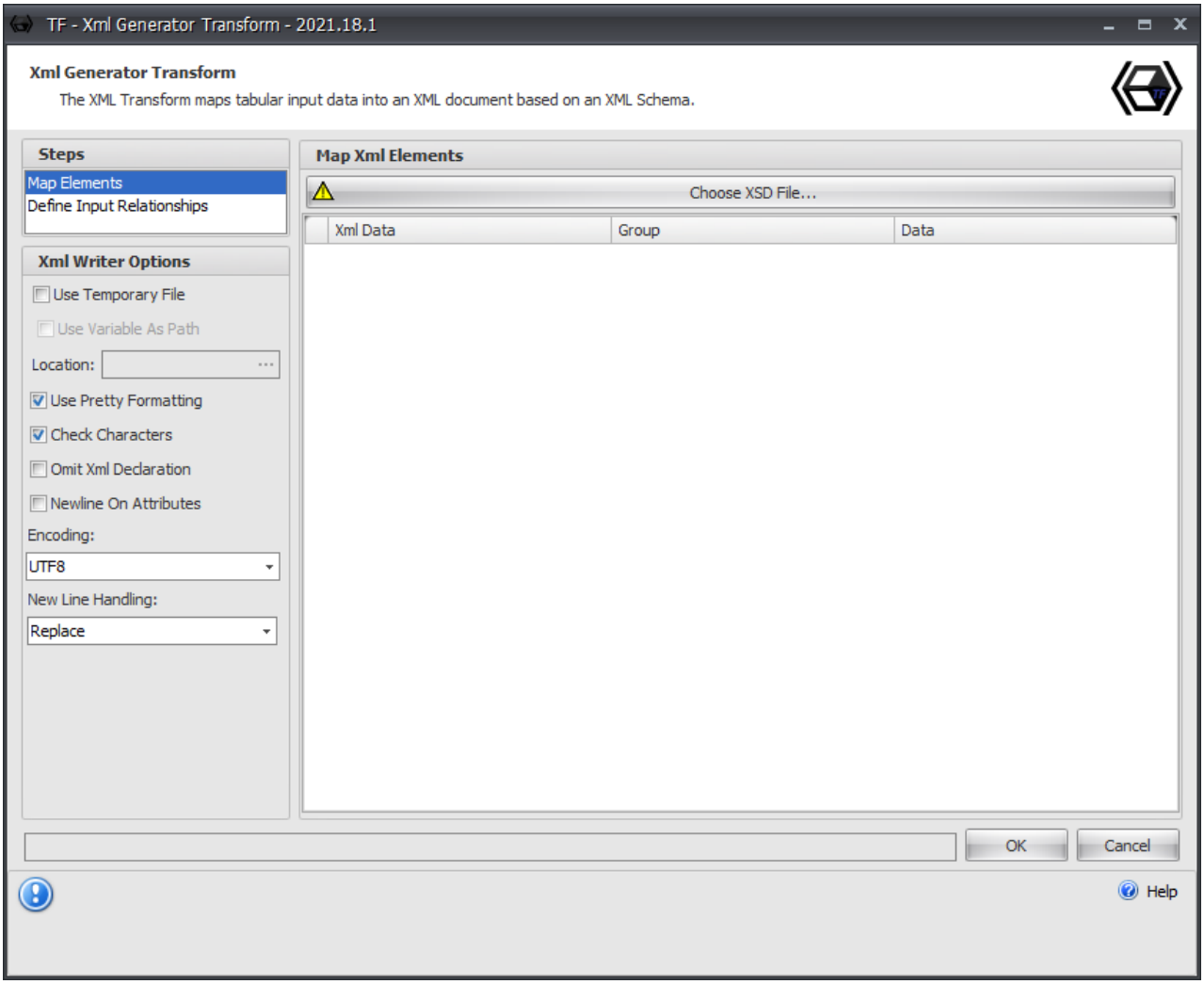
Method	Description
SQL Server Order By	Performed by using an **ORDER BY** clause within a query. Additionally, the **IsSorted** property need to be set to true and the key sort order set. These are found in the source(s) Advanced Editor (eg Ole DB Source and ADO.Net Source).

Method	Description
Sort Transform	Configured when using the SSIS native **Sort Transform** component between the source(s) and XML Output Destination.
.NET Guid	Configured when using .NET Guid sorting within an upstream Script Task.

XML Generator Transform

⚠ **Important:** The XML Generator processing occurs in memory, therefore, it's not recommended for large documents.

Transform Icon	Transform Description
	The **XML Generator Transform** is used to generate XML from one or more source inputs using an XSD mapped to input columns. It's used for producing hierarchical XML content that will not be stored in a file.



Option	Description
Steps	There are two steps: • **Map Elements** - In this step the right side section is **Map XML Elements**. • **Define Input Relationships** - In this step the right window changes to **Map Input Key Columns**.
Use Temporary File	This option is used to relieve memory pressure when using larger datasets. When this option is selected, each buffer writes to a temporary file, and then is sent to the XML file while reading the next buffer. **Note:** For extremely large datasets, it is recommended to use the XML Output Destination instead of the transform.
Use Variable As Path	Select this option to set the Temporary File's path within a variable.
Location	Defines the path to the Temporary File when not using a variable.

Option	Description
Use Pretty Formatting	Toggle for output to use line breaks and indentations or a single line. **Warning:** Pretty Formatting can increase file sizes.
Check Characters	Enabling this option fails the component when illegal characters are output.
Omit XML Declaration	The opening XML declaration. is removed.
Newline On Attributes	All attributes included in the output print on a new line.
Output Encoding	Selects the output encoding: • UTF8 • Unicode • UTF8 Without BOM (removes the byte order mark)
New Line Handling	• **Entitize** - Replaces new line characters that would not be otherwise preserved with character entities. • **Replace** - Replaces new line characters with \r\n. • **None** - Leaves the input unchanged.

Additional Information: For more information to determine which handling should be used, please see XmlWriterSettings.NewLineHandling Property.

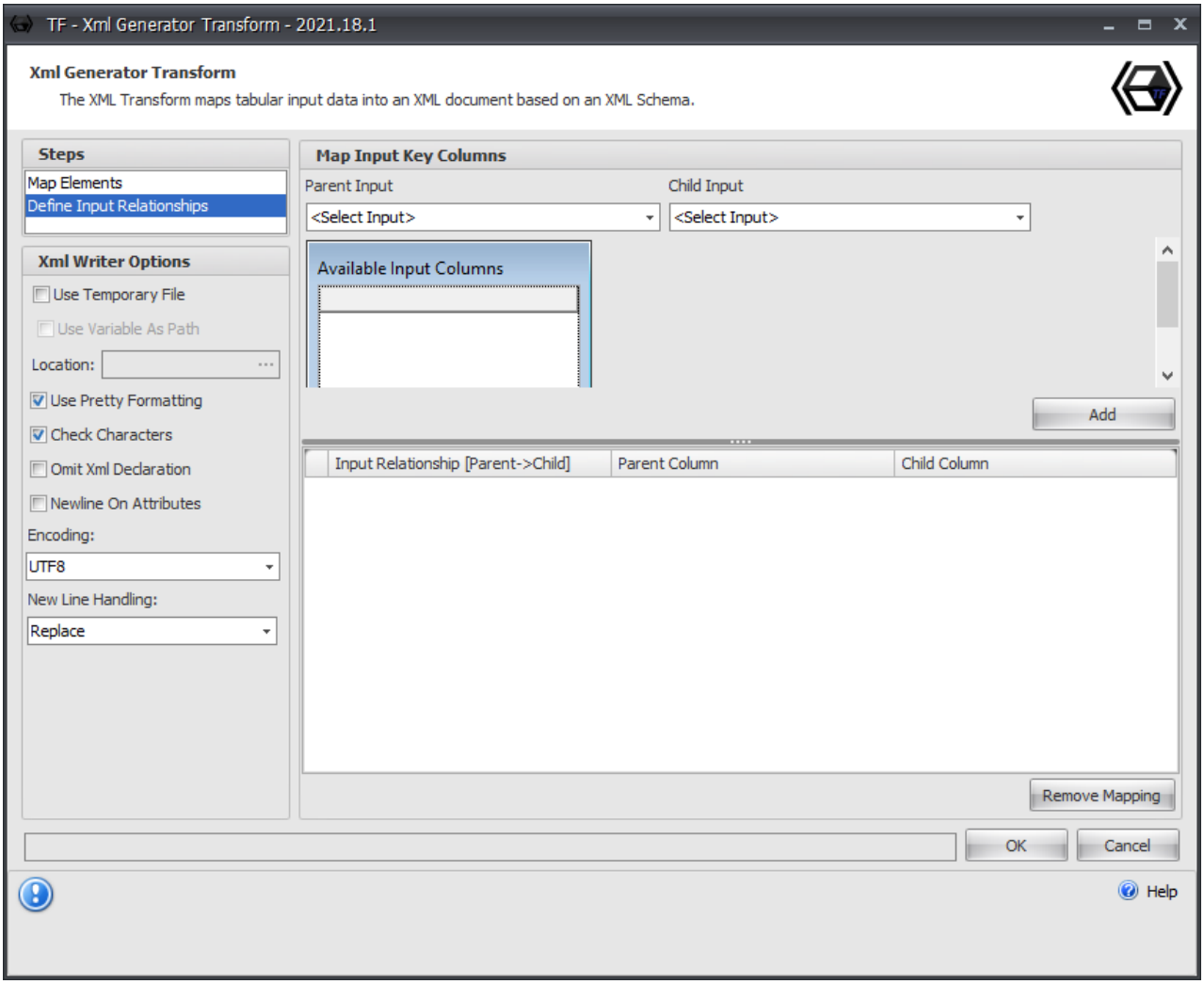
Map XML Elements

After selecting the XSD file to use, the field populates.

Option	Description
XML Data	This is a read only view of the elements and attributes from the XSD.
Group	This is where you assign sequence elements an input to group the sequence on.
Data	This is where you map XSD attributes and elements to input columns.

Define Input Relationships

If there are multiple inputs attached to the XML generator transform and your XSD has parent-child sequences, you need to define input relationships if you mapped inputs to those child sequences.



XML Source

XML Source is a native SSIS component.

Additional Information: See the XML Source article on Microsoft Docs for details.

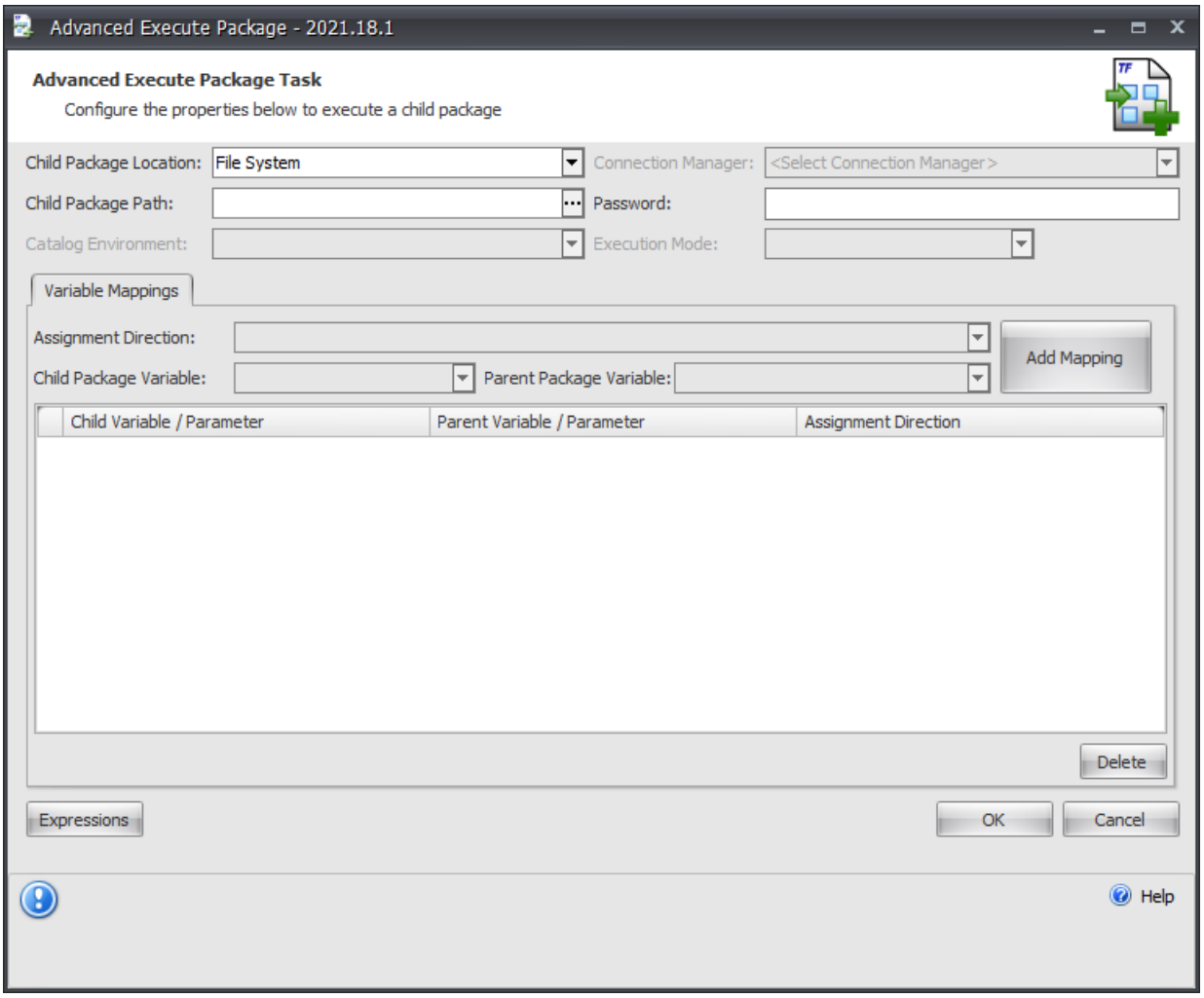
Task Factory Advanced Execute

Last Modified on 31 March 2022

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Advanced Execute Package Task

Task Icon	Task Description
	The **Advanced Execute Package Task** is used to execute child packages within an SSIS package from either a local file or a package stored on SQL Server. A unique feature of the **Advanced Execute Package Task** is the ability to map variables between the executing package and the child package.



Option	Description
Child Package Location	There are two choices for choosing a child package location: • **File System** - A local file on the executing machine • **SQL Server** - A package stored on a SQL Server
Connection Manager	Only available if SQL Server is selected in the Child Package Location. You may select either an ADO or OLEDB connection manager.
Child Package Path	Depending on the location choice, the package path either points to a local file or the path where it is stored on a SQL Server.
Password	If the child package is password protected you must enter the password here.
Catalog Environment	Users can select environment variables created within SQL Server.
Execution Mode (Catalog execution only)	Users define how the child package executes by selecting one of the following: • **In Process** - (Default) Downloads the package, creates a local ispac project, and executes it locally. • **Out of Process** - (Preferred) Executes the package on the server. This is the most common use because local resources are not used to run the package. You cannot read any parameters after execution. Parameters can only be written to. ◦ **Run In 32 Bit (Catalog execution with Out of Process mode only)** - Executes the child package in 32 bit mode.

Variable Mappings

You have three options for mapping variables from your package to the child package called **Assignment Directions**.

Variable Mapping Option	Description
Read Variable Form Child Package	This mapping direction reads the value of the variable selected in the **Child Package Variable** and assigns the value to the variable in the **Parent Package Variable**. The value of the variable is read after the execution of the child package occurs.
Write Variable to Child Package	This mapping direction writes the variable selected in **Parent Package Variable** and assigns the value to the variable selected in the **Child Package Variable**. The value of the variable is written before execution of the child package.
Read and Write Variable From Child Package	The mapping direction writes the variable value selected in **Parent Package Value** and assigns the value to the variable selected in the **Child Package Variable** before execution occurs and then reads the value of the variable selected in the **Child Package Variable** and assigns the value to the variable selected in the **Parent Package Variable** after execution occurs.

Data Type Matching of Variables

The data type of the mapped variables must be the same. For instance if you choose a string variable in the **Child Package Variable**, only string variables from your package display in the **Parent Package Variable** drop-down.

What is the difference between In Process and Out of Process modes in the Advanced Execute Package Task?

The following describes each Child Package Location and how variables/parameters are assigned:

File System

- Executes a child package by loading the package from the file system. It is then executed in the process of the parent package.
- The child package has no knowledge of the project (2012 and higher.)
- Child packages cannot use project level parameters or connection managers internally.
- Project level parameters can be assigned to child package variables.
- Parent package variables / parameters can be assigned to child package variables.
- Parent package variables can be assigned values from child package parameters and variables after execution.

SQL Server (MSDB, File System)

- Executes a child package by copying the package from SQL Server into the parent package and executing it in the process of the parent package.
- The child package has no knowledge of the project (2012 and higher.)
    - Therefore, child packages cannot use project level parameters or connection managers.
- Project level parameters can be assigned to child package variables.
- Parent package variables & parameters can be assigned to child package variables.
- Parent package variables can be assigned values from child package parameters and variables after execution.

Local Catalog Execution

- Copies the package from the Catalog Server into the parent package.
- The child package executes in the same process as the parent package.
- The child package has no knowledge of the project (2012 and higher.)
    - Therefore, child packages cannot use project level parameters or connection managers. Project level parameters can be assigned to child package variables.
- Parent package variables & parameters can be assigned to child package variables.
- Parent package variables can be assigned values from child package parameters and variables after execution.

Server Catalog Execution (New Feature)

- Will execute a package via server catalog execution (out of process) and wait on the package to finish

execution on the server.

- The child package has knowledge of project level variables and connection managers available within the project catalog of the child package being executed.
- The child package has the option of assigning a catalog project environment from the child package project to the execution.
- Child package variables can be assigned project level & package level parameters, package level variables as well as being able to use environment variables from the catalog environments.
- Parent level variables cannot be assigned values from child level parameters and variables after execution because the execution is performed out of process.

Overview of features

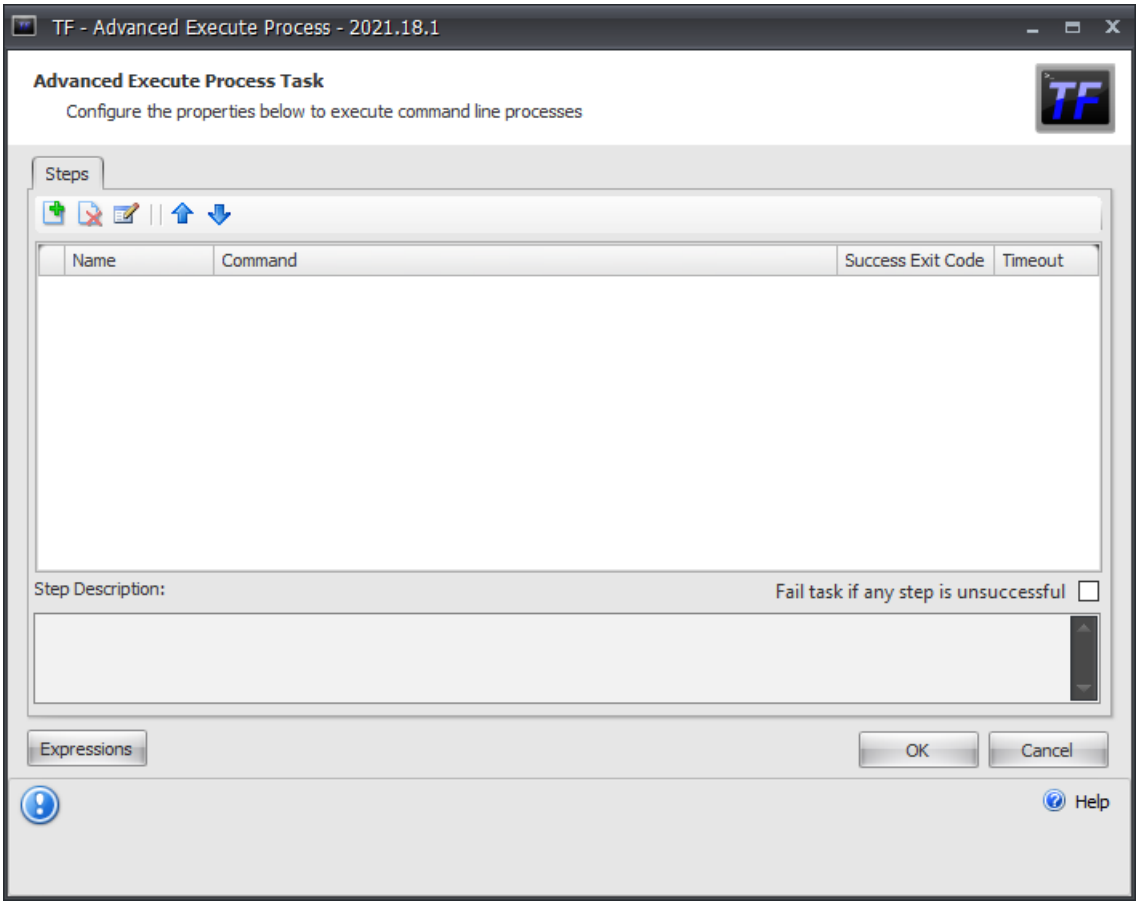
This chart is a quick summary of each location, variable assignment, and the execution mode that should be used:

Location	File System	SQL Server	Local Catalog/In Process	Server Catalog/Out Of Process
Parent to child variable access	Read & Write	Read & Write	Read & Write	Write
Child to parent variable access	Read & Write	Read & Write	Read & Write	None
Parent Parameter to child variable assignment	Read	Read	Read	Read
Child package project level parameter access	None	None	Yes via copy	Yes
Child package project level connection manager access	None	None	Yes via copy. No passwords available in connection strings	Yes
Can assign catalog environment variables to child variables	No	No	Yes	Yes
Can use catalog environments during execution	No	No	No	Yes

Location	File System	SQL Server	Local Catalog/In Process	Server Catalog/Out Of Process
Execution Process	In process	In process	In process	Out of process

Advanced Execute Process Task

Task Icon	Task Description
	The **Advanced Execute Process Task** is used to execute commands as if they were being executed from a command window. This allows for the use of basic commands like copy, ping, delete, etc. as well as using any executable you would like to be executed within SSIS. One of the best features is that it allows for multiple commands per instance of the task.



Steps

You are able to add multiple **steps** to the advanced execute process. A step is a single command that mimics what would be executed from the command line.

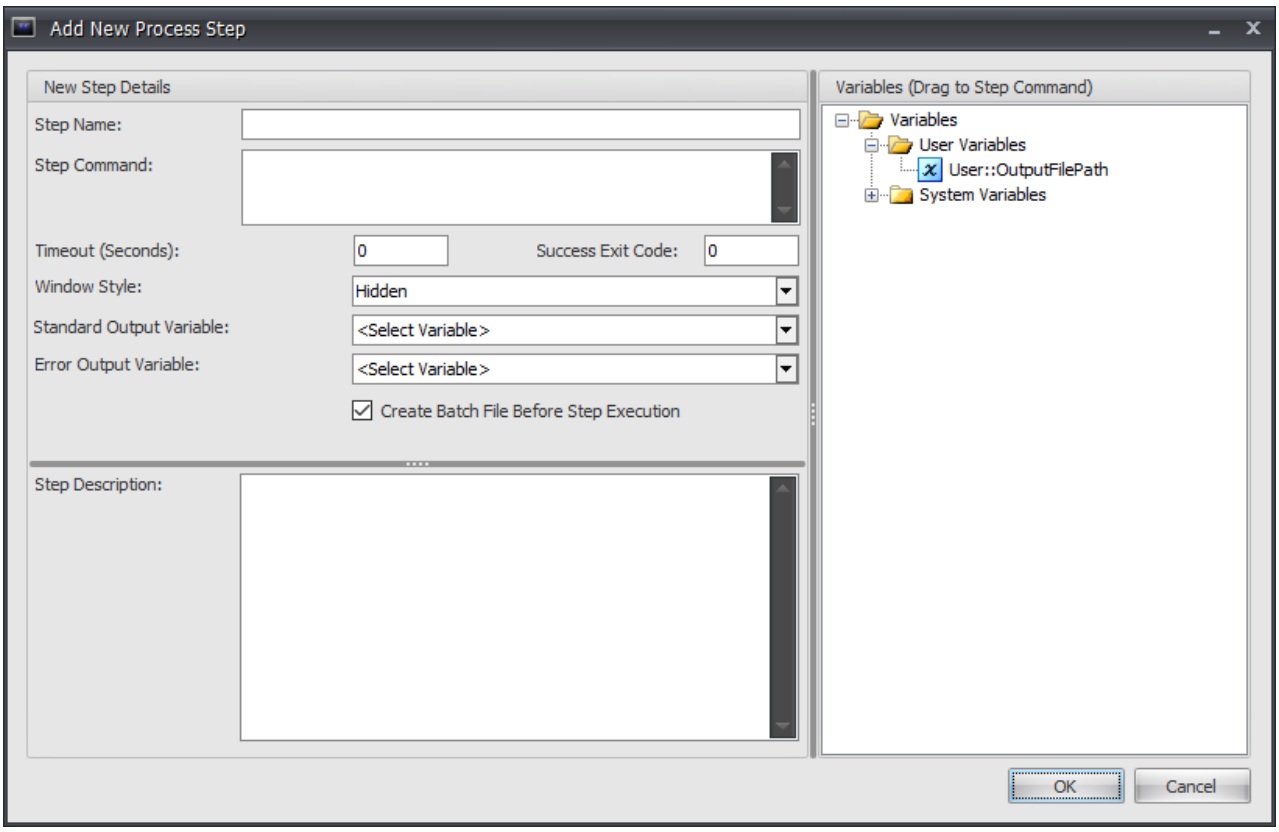
Toolbar

The toolbar is used to control the steps for the **Advanced Execute Process**.

Toolbar button	Description
	Select to add a new Step to the Task.
	Select to delete the currently selected Step in the Steps grid.
	Select to edit the currently selected Step in the Steps grid.
	Select to move the currently selected Step in the Steps grid up in the order of execution.
	Select to move the currently selected Step in the Steps grid down in the order of execution.

Add New Step

Selecting the **Add New Step** button opens the **Add New Process Step** window.



Option	Description
Step Name	The name of the current step.
Step Command	The command in which this step executes. The command can use any variables that are part of the package as replacement values in the command.
Timeout (Seconds)	The number of seconds you would like this step/command to run before the process is terminated. A value of 0 means there is no timeout.
Success Exit Code	The exit code of the command which indicates the step was successful. The default success code for the command line is 0.
Window Style	The style in which a command window is shown. The four options are: Hidden, Normal, Minimized, and Maximized.
Standard Output Variable	Choose to store the output to either a variable or command window.
Error Output Variable	Choose to store the error output to either a variable or command window.

Option	Description
Step Description	You can store notes about the step here. When you enter a description here, it shows up when the step is selected in the main UI as shown below.

Reordering the Steps

Use the blue arrows in the toolbar to change the order the steps are executed.

Task Factory HTTP Download

Last Modified on 31 March 2022

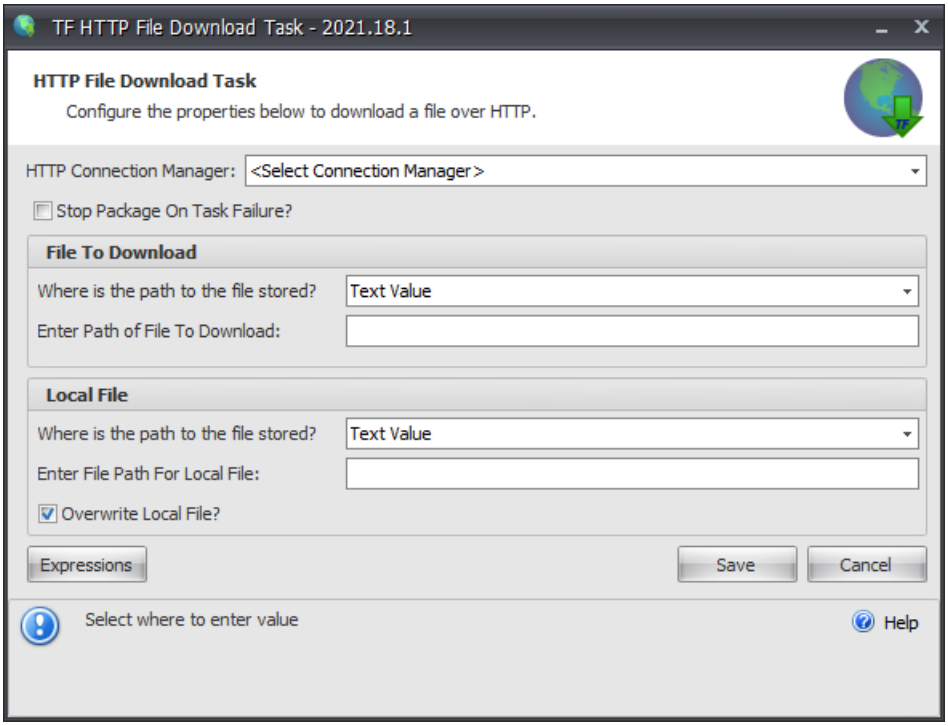
Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

HTTP File Download Task

Task Icon	Task Description
	The **HTTP File Download Task** is used to retrieve files using the **HTTP** or **HTTPS** protocols, and the **SSIS HTTP Connection Manager**.

Note: The **HTTP File Download Task** supports **HTTPS** protocol connections.

Important: The **HTTP File Download** ask uses the **Native HTTP Connection Manager** by Microsoft, and only supports TLS1.2. TLS1.0 and TLS 1.1 are not supported.



Option	Description
HTTP Connection Manager	The **HTTP Connection Manager** dropdown contains any **SSIS HTTP**, or **SSIS HTTPS** Connection Managers that are part of your package. You can create a new connection manager by selecting **Create New HTTP Connection** which displays the **HTTP Connection Manager Editor**.
Stop Package On Task Failure?	Selecting this option stops the execution of the package if a failure occurs.

File to Download

Option	Description
Where is the path to the file stored?	Select where the file to download is stored.
Enter Path to File to Download	This field's description changes depending on where the path to the file is stored.

Local File

This section is used to tell the HTTP File Download Task where to store the downloaded file locally.

Option	Description

Option	Description
Where is the path to the file stored?	There are four options for creating the path to the local file. • **Text** - Enter the URL to the file. • **Expression** - Use an expression to build the URL to the file. • **Variable** - Select a variable that contains the URL to the file. • **Connection Manager** - Configure a file connection manager to point to a local file.
Enter File Path For Local File	This property changes depending on where the path to the file is stored.
Overwrite Local File?	Set this property to **true** if the **HTTP File Download Task** should overwrite the local file with the file that is downloaded. Setting it to false causes the task to fail if the local file exists already.

Task Factory Compression

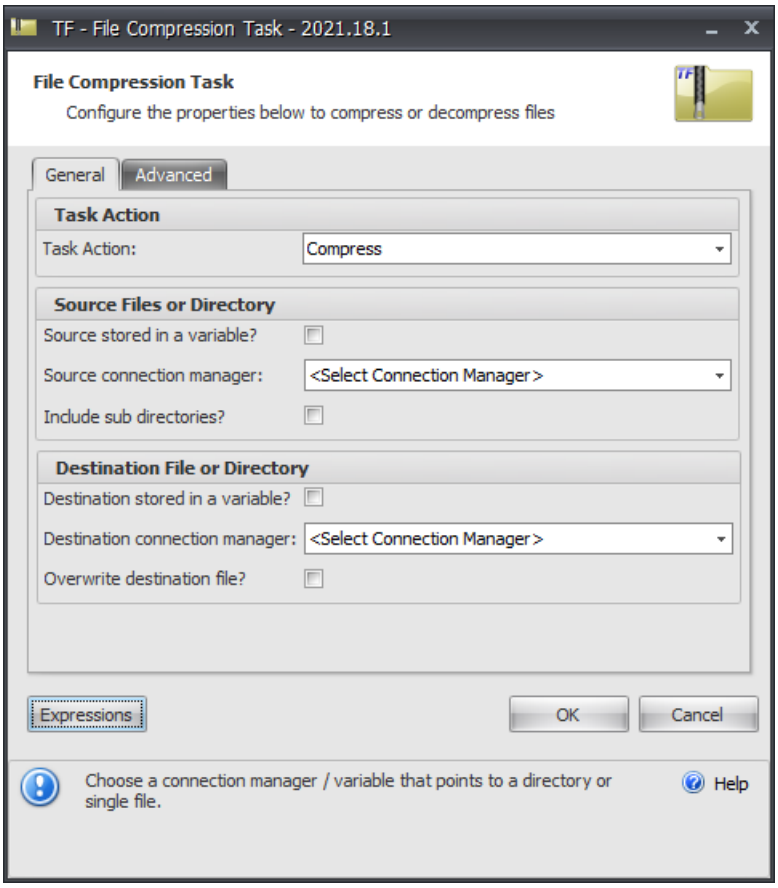
Last Modified on 05 May 2022

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Compression Task

Task Icon	Task Description
	The **Compression Task** is used to either compress or decompress a file.

General



Task Action

Option	Description
Task Action	There are two options for the Task Action: • Compress • Decompress

Source Files or Directory

Option	Description
Source stored in variable?	Selecting this checkbox replaces the **Source Connection Manager** with **Select Source Variable** and the drop down populates with all of the user created string variables within your package. The task expects the variable to contain the file or directory path to be compressed / decompressed.
Source connection manager	Available if **Source stored in a variable** is not selected. The source connection manager selected contains the path to the file or directory (zip compression only) that is used for compression / decompression. If you do not have an existing File or flat file connection you wish to use, you can create one here.
Include Sub directories?	Available only for Zip Compression. Selecting this option includes all subdirectories within the selected variable or connection manager path.

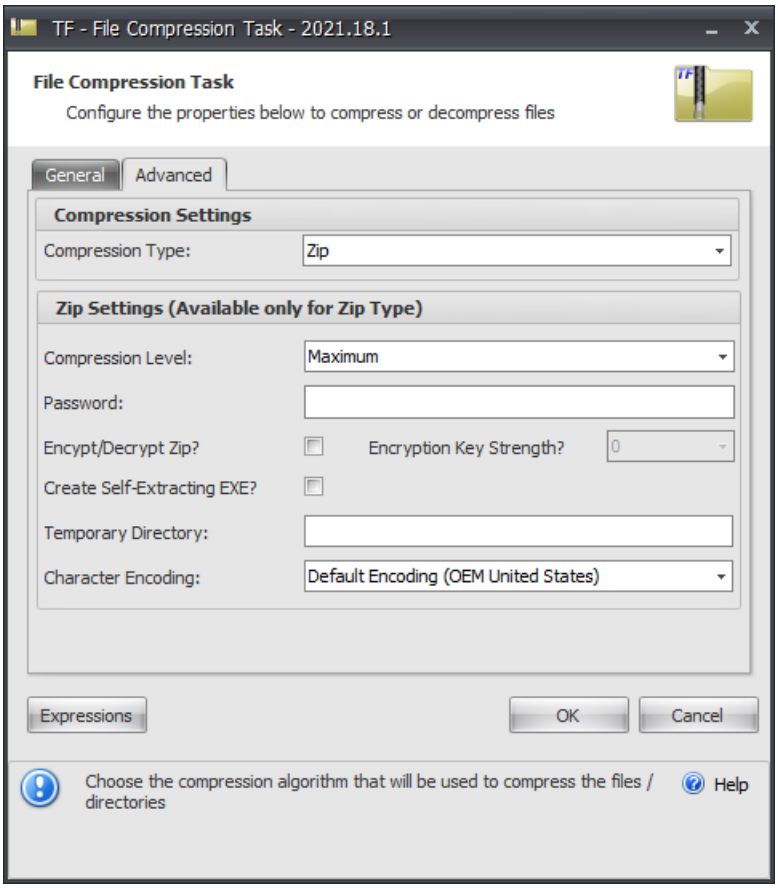
Destination File or Directory

Option	Description
Destination stored in a variable?	Selecting this checkbox replaces the **Destination Connection Manager** with **Select Destination Variable** and the drop down includes all of the user created string variables within your package. The task expects the file or directory to be compressed / decompress to be defined in the variable selected.
Destination connection manager	Available if **Destination Stored in Variable** is not selected. The destination connection manager should point to the connection manager that points to the file or directory (zip compression only) that is used for compression / decompression. If you do not have an existing file or flat file connection you wish to use, you can create one here. ⚠ **Important:** The only connection manager options that can be used for compressing are **Create File** and **Existing File**.)
Overwrite destination file	Select this if you want the compression task to overwrite the file if it already exists. If this option is not selected and the file already exists, an exception is thrown.

Expressions

Additional Information: See the Expressions article for more information on using the **Expressions** button functionality.

Advanced



Compression Settings

Option	Description
Compression Type	There are four compression algorithms that can be used: • Zip • GZip • Z • Bz2

Zip Settings (Available only for Zip Type)

Option	Description

Option	Description
Compression Level	Sets the level of compression to be used during the compression process. • **None** - No compression occurs. The files and directories are added to the zip file only. • **Fast** - A small amount of compression occurs but the compression time is fast. • **Normal** - The normal level of compression is used. • **Above Normal** - A slightly above normal level of compression is used. The compression time is slower. • **Maximum** - The maximum amount of compression occurs. The compression time is slower than all of the other levels.
Password	If used with a Task Action of Compress, the password specified is added to the zip file. This password is necessary to open the zip file by any end user. If used with a Task Action of Decompress, the password is used to decompress the file selected as the source file.
Create Self-Extracting EXE	Selecting this option creates a self-extracting exe instead of a zip file.
Temporary Directory	Enter here the directory where the temporary files are created during compression. If left blank it uses the default location.
Character Encoding	Specifies the character coding to be used in the archive.

Samples

See the taskfactory-samples repository on GitHub for the following Compression Task samples:

- **1_SetupCompressionTaskDirectory.dtsx** - Use to set up the compression samples
- **CompressionTask_Compress.dtsx** - Compression task configured to compression files into a zip file
- **CompressionTask_Decompress.dtsx** - Compression task configured to decompress files from a zip file

Task Factory File Gateway

Last Modified on 05 May 2022

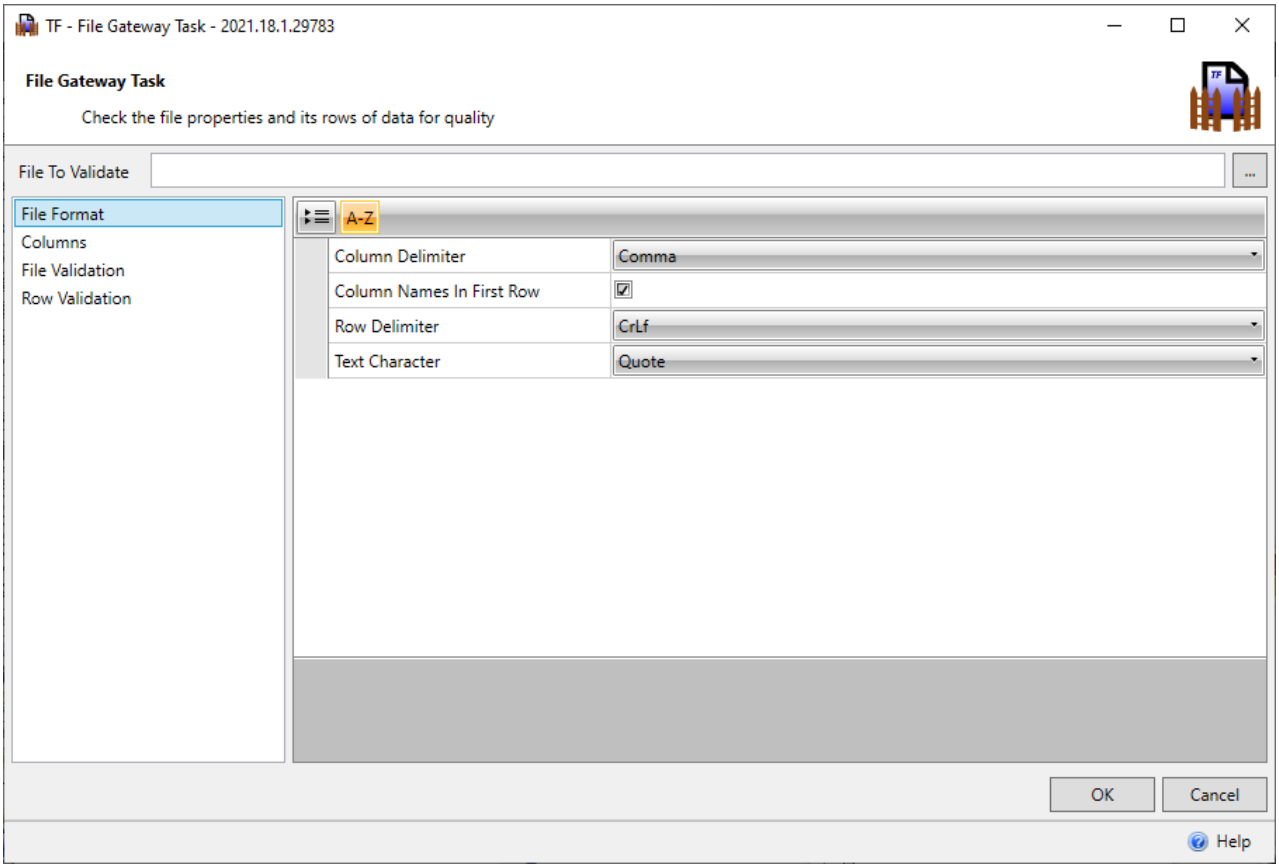
Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Note: File Gateway is available for SQL Server versions 2012 and higher.

File Gateway Task

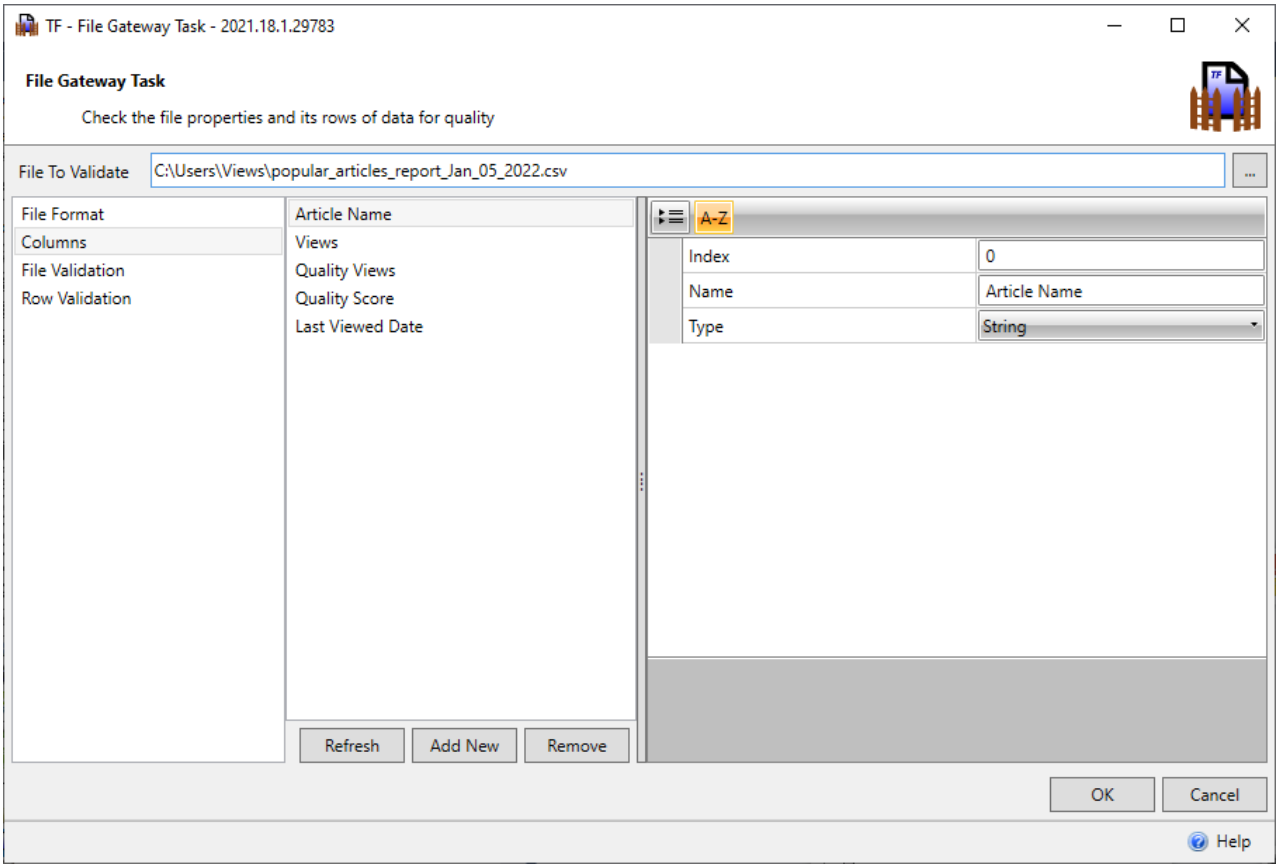
Task Icon	Task Description
	The **File Gateway Task** gives you the ability to validate delimited files before performing a downstream action on the selected file.

File Format



Option	Description
Column Delimiter	Identifies the character used to distinguish between columns. Users can select between comma, vertical bar (pipes), semicolon, colon, or tabs (\t).
Column Names In First Row	This option should be selected if the file contains column names in the first row.
Row Delimiter	This option identifies the performance that creates a new row. Users can select between carriage return line feed (CrLf), line feed only (Linefeed), and carriage return (CarriageReturn.)
Text Character	Identifies whether row data is wrapped in single quotes, double quotes, or no specific character.

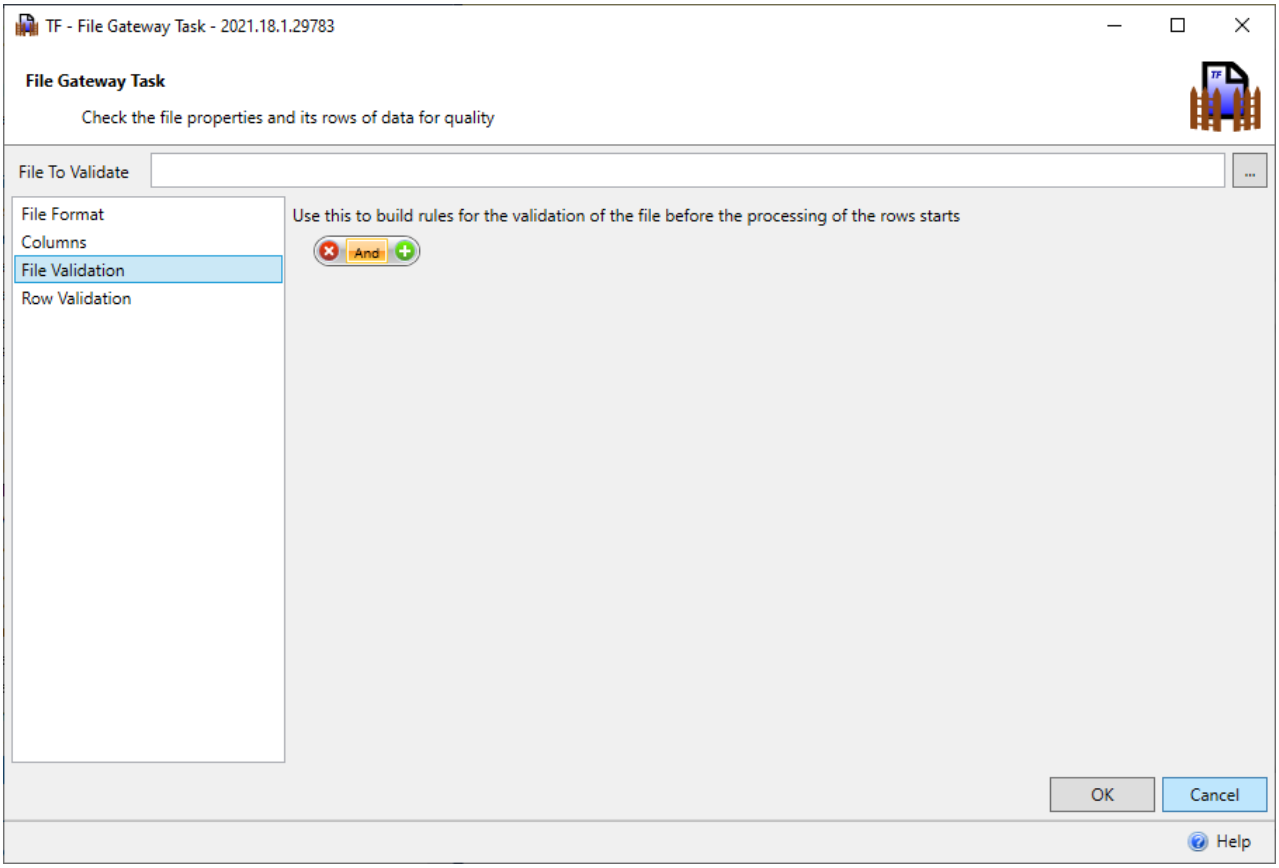
Columns



Option	Description
Index	Identifies the order (zero-based) of the columns.
Name	Users can set or rename columns.
Type	Users can set the column data type.

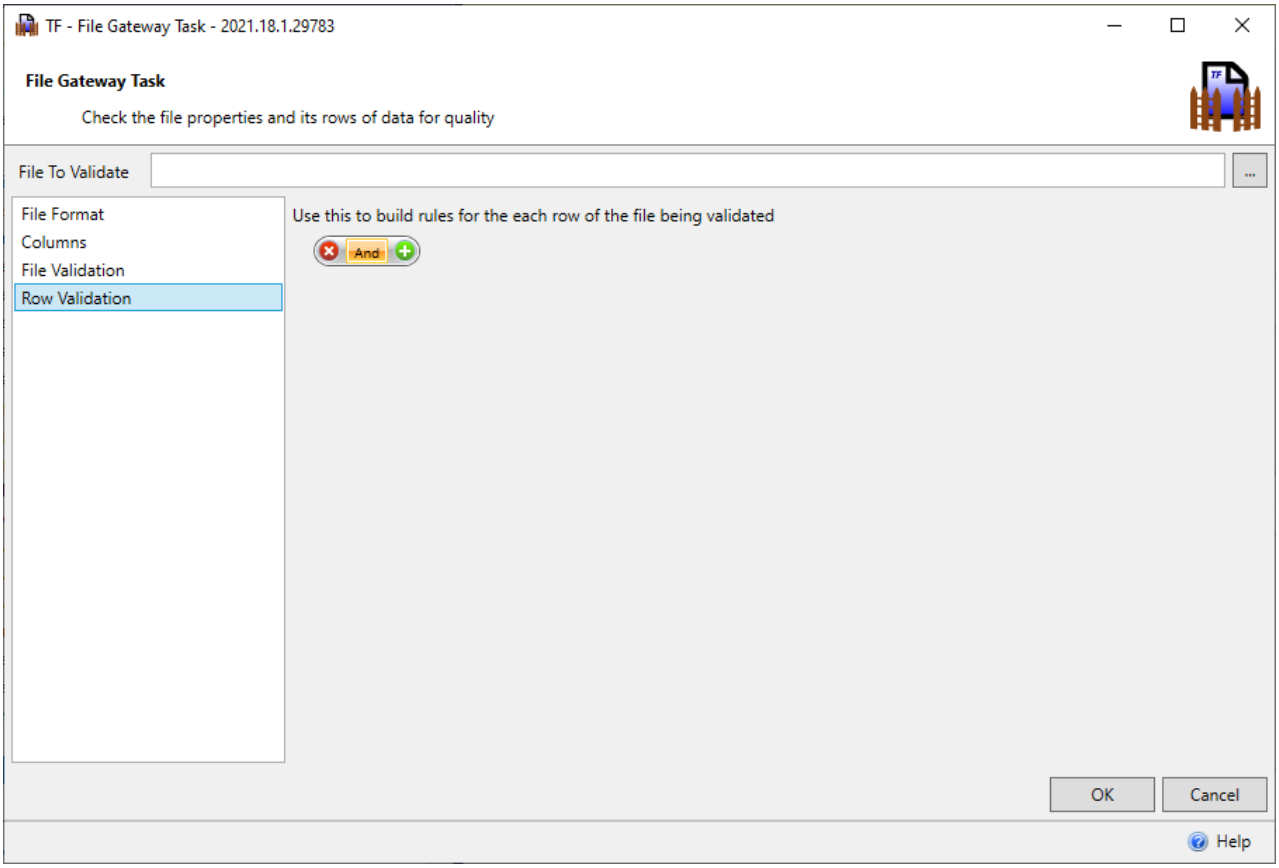
File Validation

Users can build rules for validation of the file before row processing begins. Users make such requirements as the file size should be equal to, greater than, or less than a specified size (in kilobytes), a specified number of column headers or rows, or the file's last modified date occurred on, before, or after a specific date. Users should note that multiple conditions can be created to validate the file.



Row Validation

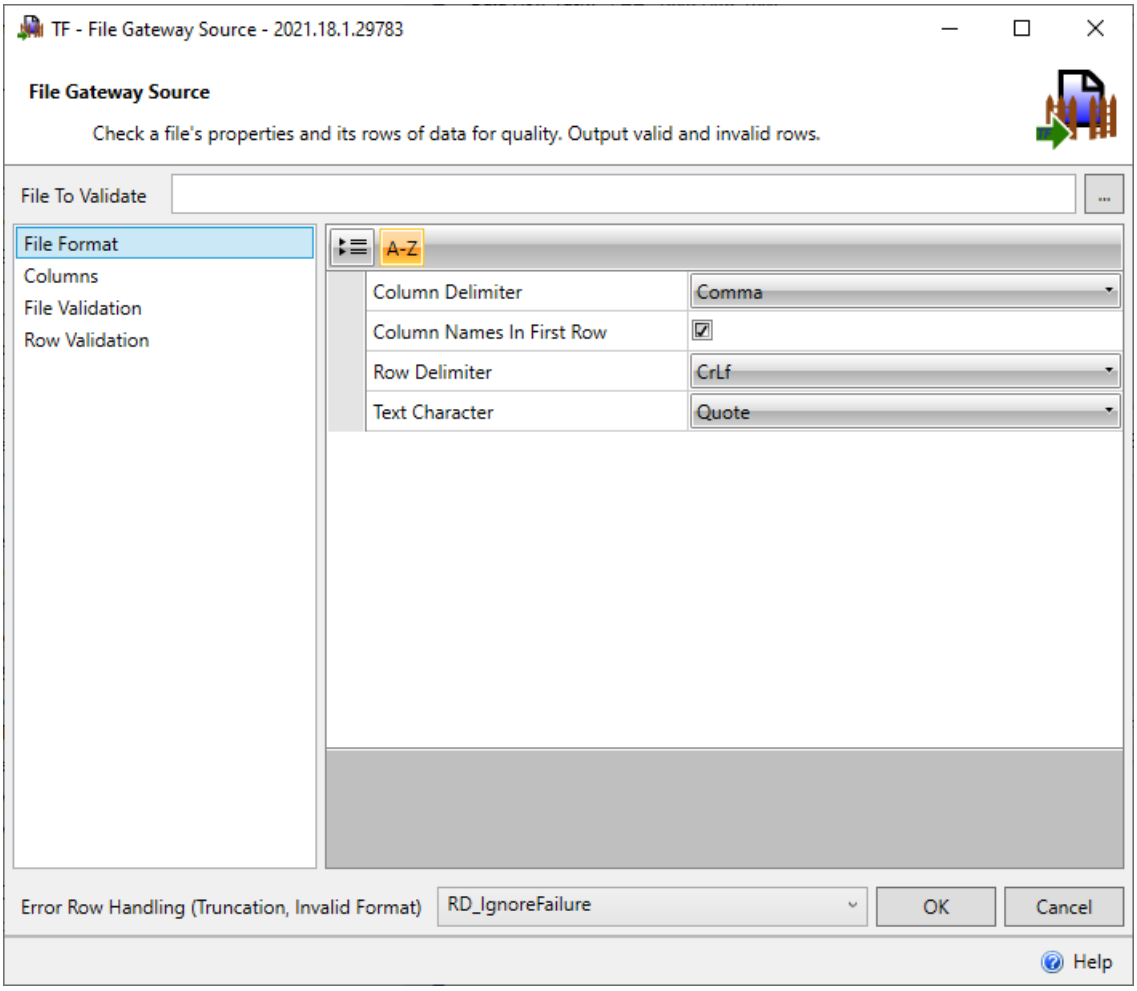
Users can build rules that each row must follow. For example, if a column should not contain null, users can add an **Is Not Null** condition for that column. If a NULL is found during the scan, the validation would fail causing the File Gateway Task to fail and, ultimately, the package to fail.



File Gateway Source

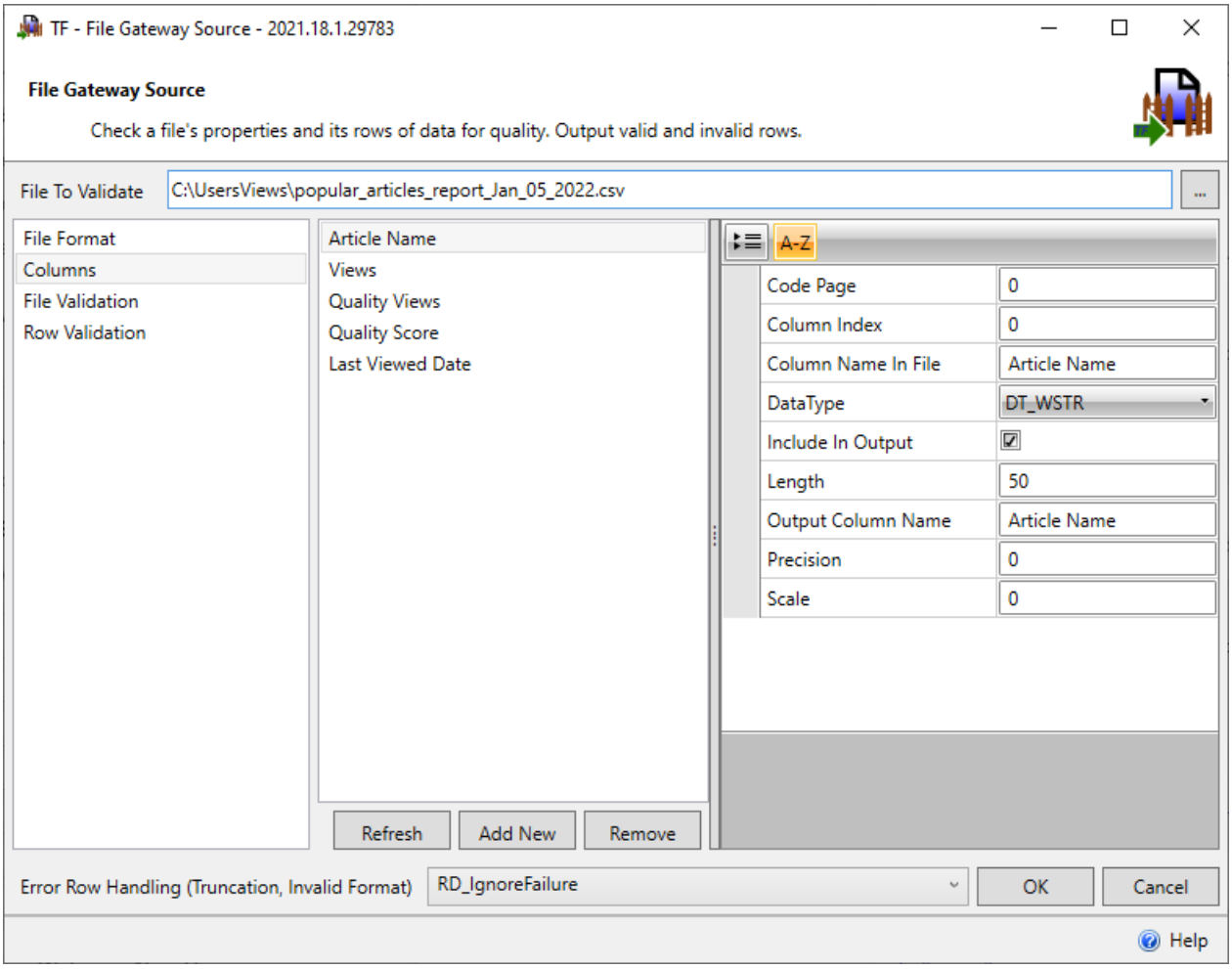
Source Icon	Source Description
	The **File Gateway Source** gives you the ability to validate delimited files or rows before performing a downstream action on the selected file.

File Format



Option	Description
Column Delimiter	Identifies the character used to distinguish between columns. Users can select between comma, vertical bar (aka pipes), semicolon, colon, or tabs (\t).
Column Names In First Row	This option should be selected if the file contains column names in the first row.
Row Delimiter	This option identifies the performance that creates a new row. Users can select between carriage return line feed (CrLf), line feed only (Linefeed), and carriage return (CarriageReturn.)
Text Character	Identifies whether row data is wrapped in single quotes, double quotes, or no specific character.

Columns

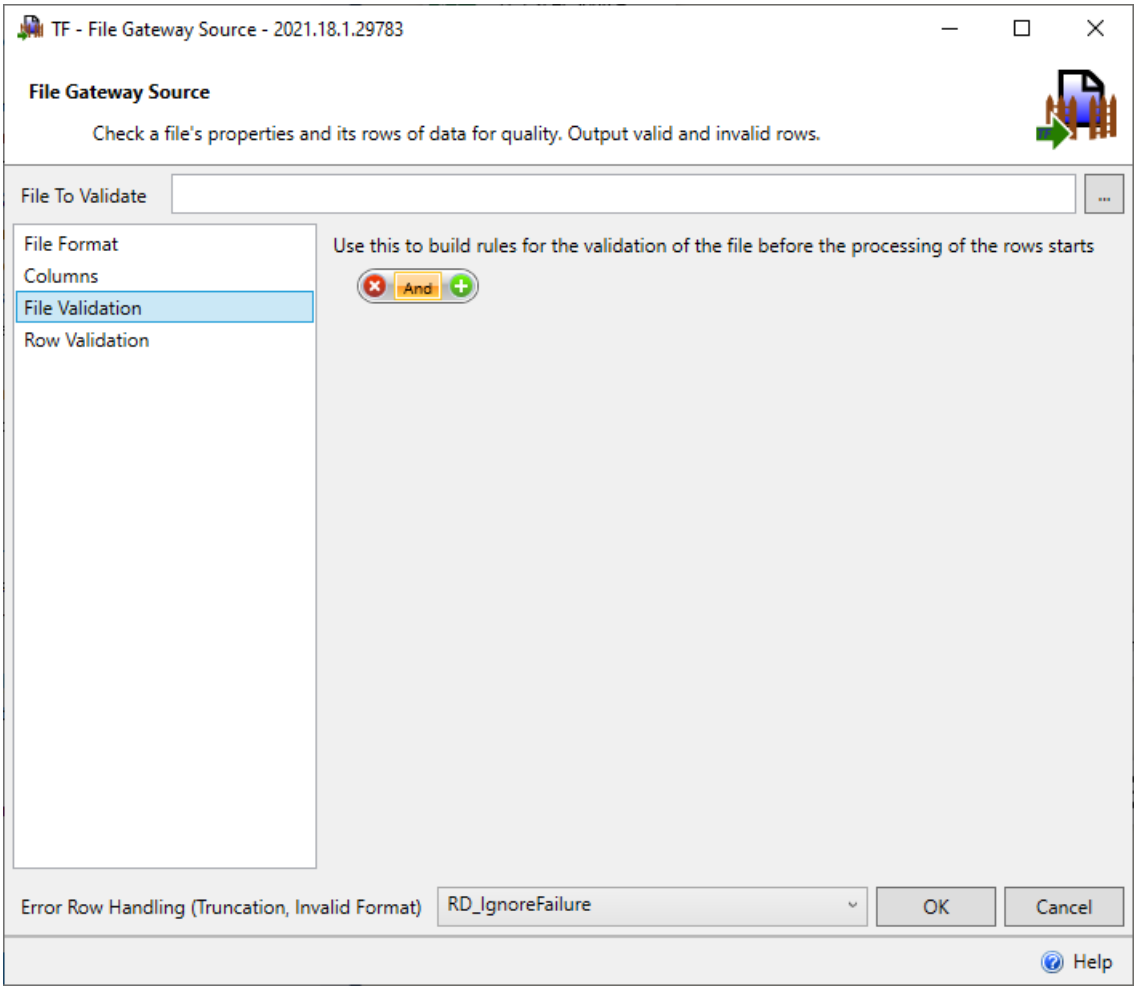


Option	Description
Index	Identifies the order (zero-based) of the columns.
Name	Users can set or rename columns.
Type	Users can set the column data type.

File Validation

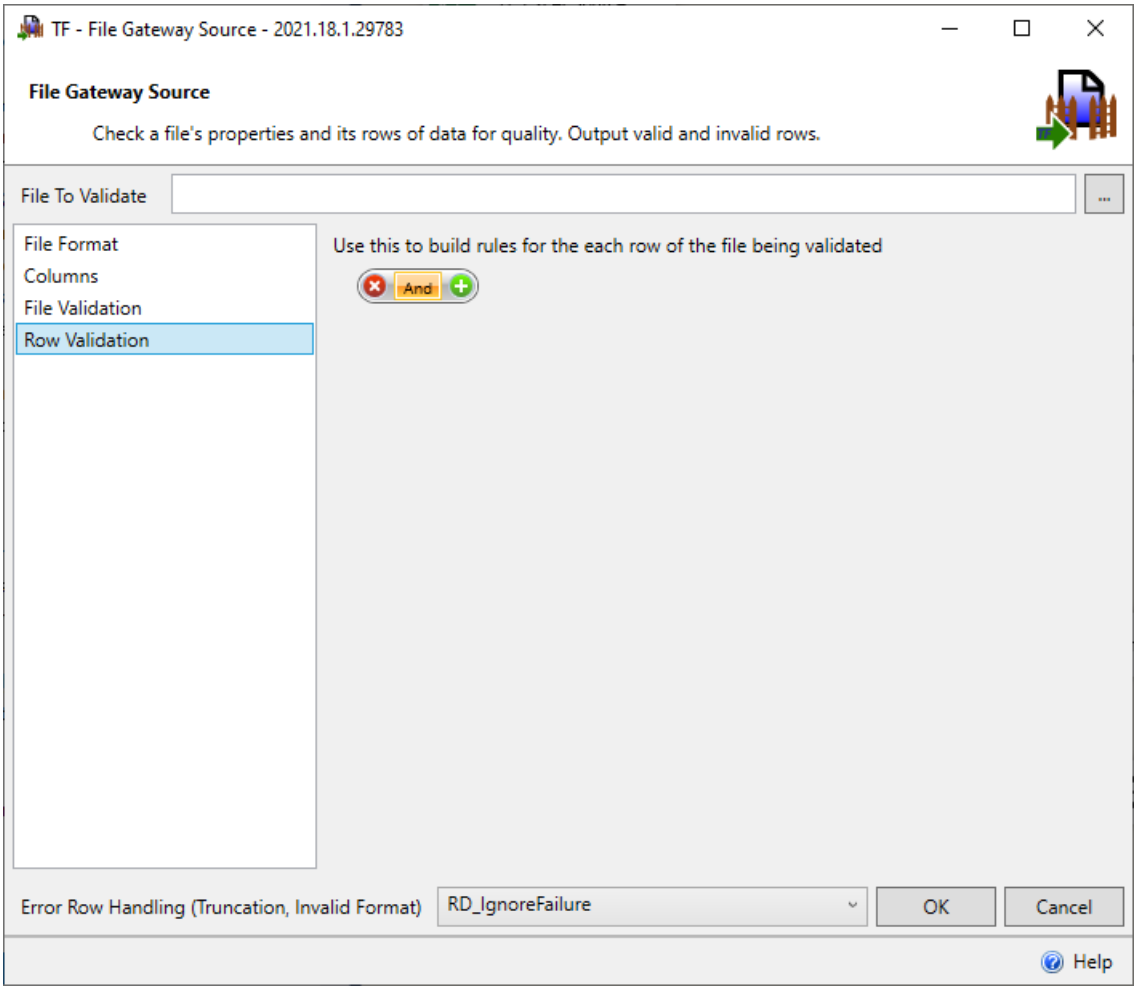
Users can build rules for validation of the file before row processing begins. Users make such requirements as the file size should be equal to, greater than, or less than a specified size (in kilobytes), a specified number of column headers or rows, or the file's last modified date occurred on, before, or after a specific date.

> **Note:** Multiple conditions can be created to validate the file.



Row Validation

Users can build rules that each row must follow. For example, if a column should not contain null, users can add an **Is Not Null** condition for that column. If a NULL is found during the scan, the validation would fail causing the File Gateway Task to fail and, ultimately, the package to fail.



The component has two successful outputs that can be used for Row Validation (valid and invalid output.)

Additional Information: See the Task Factory Error Row Handling article for more information about this functionality.

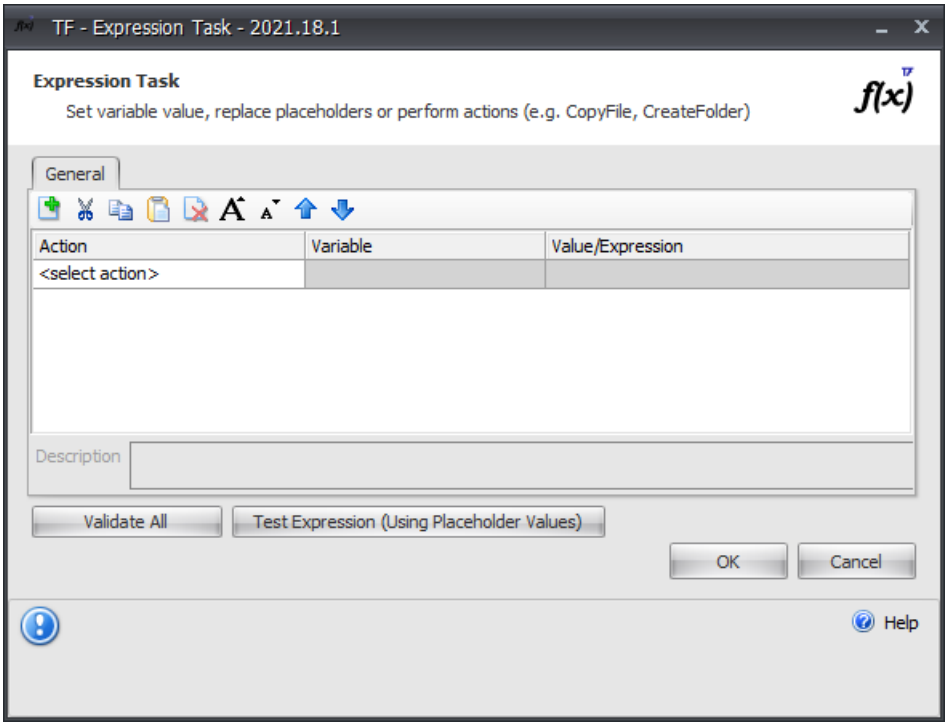
Task Factory Expression

Last Modified on 05 May 2022

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Expression Task

Task Icon	Task Description
	The **Expression** Task is used to set variables and call functions using our extensive expression engine. Each action is evaluated in the order they appear in the grid.

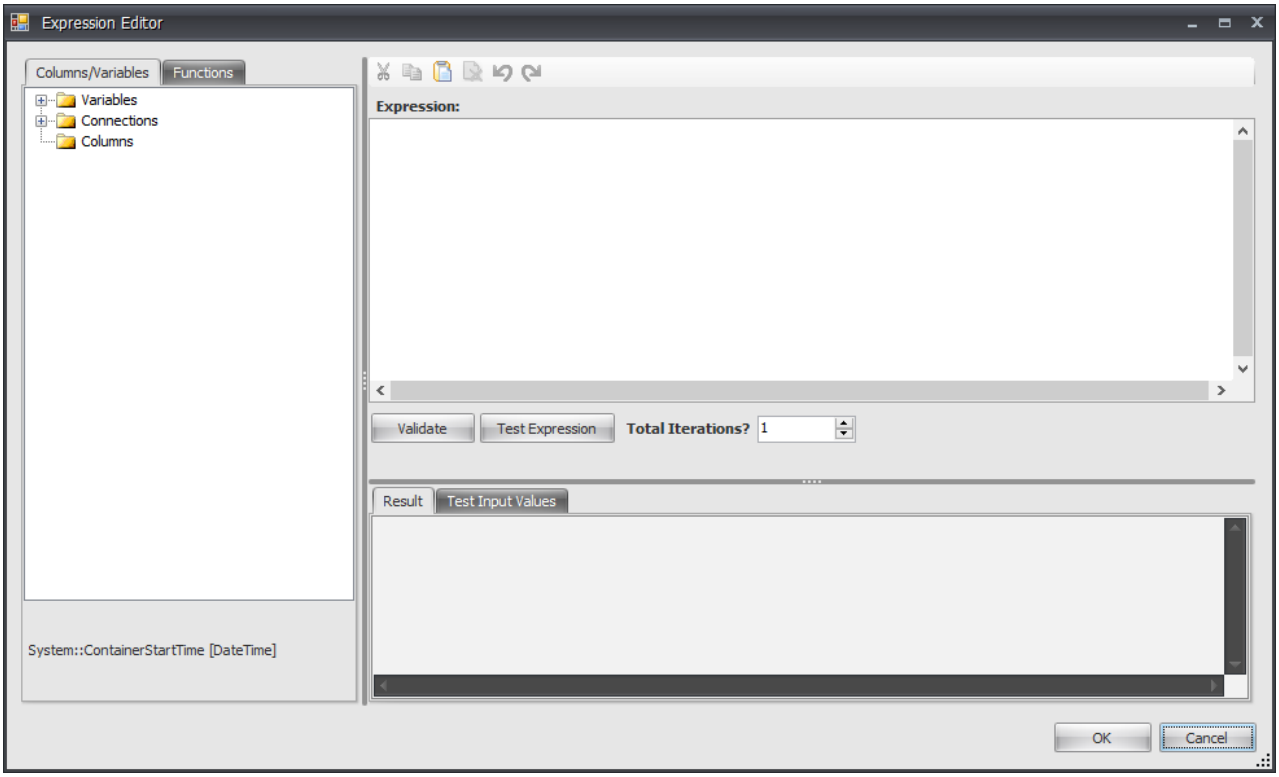


Option	Description
Action	The action this step in the expression task performs. There are five actions that can be performed: • **Set Variable** - This action sets the variable to the value entered in the **Value/Expression** column. • **Set Variable From Expression** - This action sets the variable to the result of the value returned from the expression in the **Value/Expression** column. • **Replace Placeholders - This action replaces any placeholders that are part of the variable selected in the Variable column.** ◦ **Note:** Placeholders need to be Identified with the **{{Namespace::Variable}}** syntax. For example, if you have a variable named **Str_Date**, inside of another variable you could write **The date is {{User::Str_Date}}**. • **Set Variable and Replace Placeholders** - This action sets the variable to the value entered in the **Value/Expression** column as well as replace any variable place holders in the text. • **Call Function** - This action executes the expression set in the **Value/Expression** column.
Variable	If the action requires you to select a variable, a drop down with all user defined variables displays.
Value/Expression	Depending on the action, this contains either a static value or an expression that executes at runtime.
Validate All	Selecting this button validates all the expression set in the **Value/Expression** column.
Test Expression	Selecting this button executes the selected expression using temporary values.

Expression Editor

The **Expression Editor** is used to create and edit expressions that are used in any expression engine in a Task Factory task.

UI Overview



Option	Description
Columns / Variables Tab	The **Columns/Variables** tab list all of the columns available for the current component being edited, the variables and parameters and the connection managers available for the package. The columns node also lists any variable expressions defined for the current component. Any icon in the tree view that has a green icon next to it means it is being used by an expression.
Functions Tab	The **Functions** tab lists all of the available functions that can be used in the advanced derived column. All functions are grouped into an appropriate group based on what the function either does or returns.
Expression Editor	The **Expression Editor** is used to define the expression for the column or variable expression. The syntax highlighting makes it easy to recognize the functions you are using in the expression. You can use multi-line expressions and add comments to your expressions by using the **/* */** syntax. Any column, connect, or function can be dragged and dropped from the tree view to the **Expression** window.
Validate / Test Expression Buttons	The expression editor allows you to validate and test your expressions during design time. The results appear in the **Result** tab.

Option	Description
Total Iterations?	The number of times the expression will execute when you select **Test Expression**.
Test Input Values Tab	Column/Variable, Data Type, Length, and Input Value information for the tested expression.

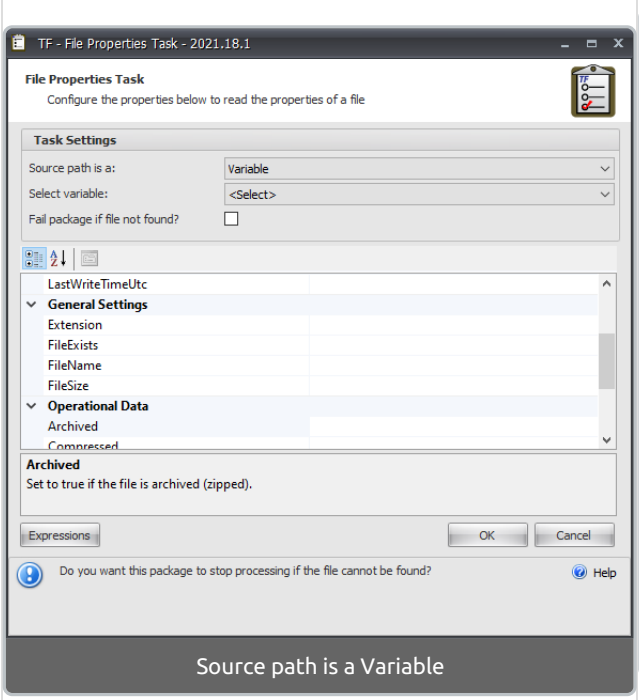
Task Factory File Properties

Last Modified on 05 May 2022

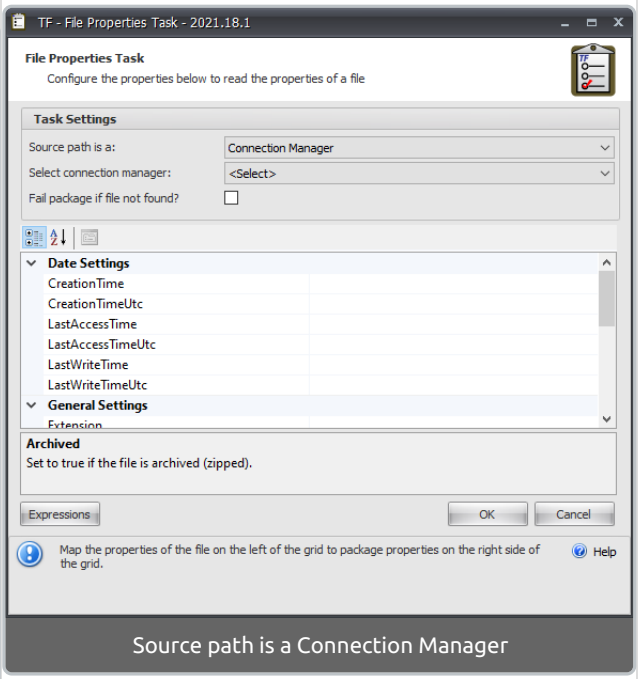
Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

File Properties Task

Task Icon	Task Description
	The **File Properties Task** is used to read the properties of a file and set package variable values based on the settings provided.



Source path is a Variable



Source path is a Connection Manager

Note: The **Compressed** property under **Operational Data** determines if a file is using NTFS compression.

Option	Description
Task Settings	• **Source path is a** - Select either Variable or Connection Manager. • **Select Variable** - Available if the source path is a variable. The drop down list populates with all the string variables in the package. • **Select Connection Manager** - Available if the source path is a connection manager. The drop down list populates with all of the File Connection Managers in the package.
Properties Grid	The properties grid is used to assign file properties that populate during execution. To set a value, select the right side of the grid to open the drop down menu with a list of variables or the option to create a new variable.

Task Factory PGP

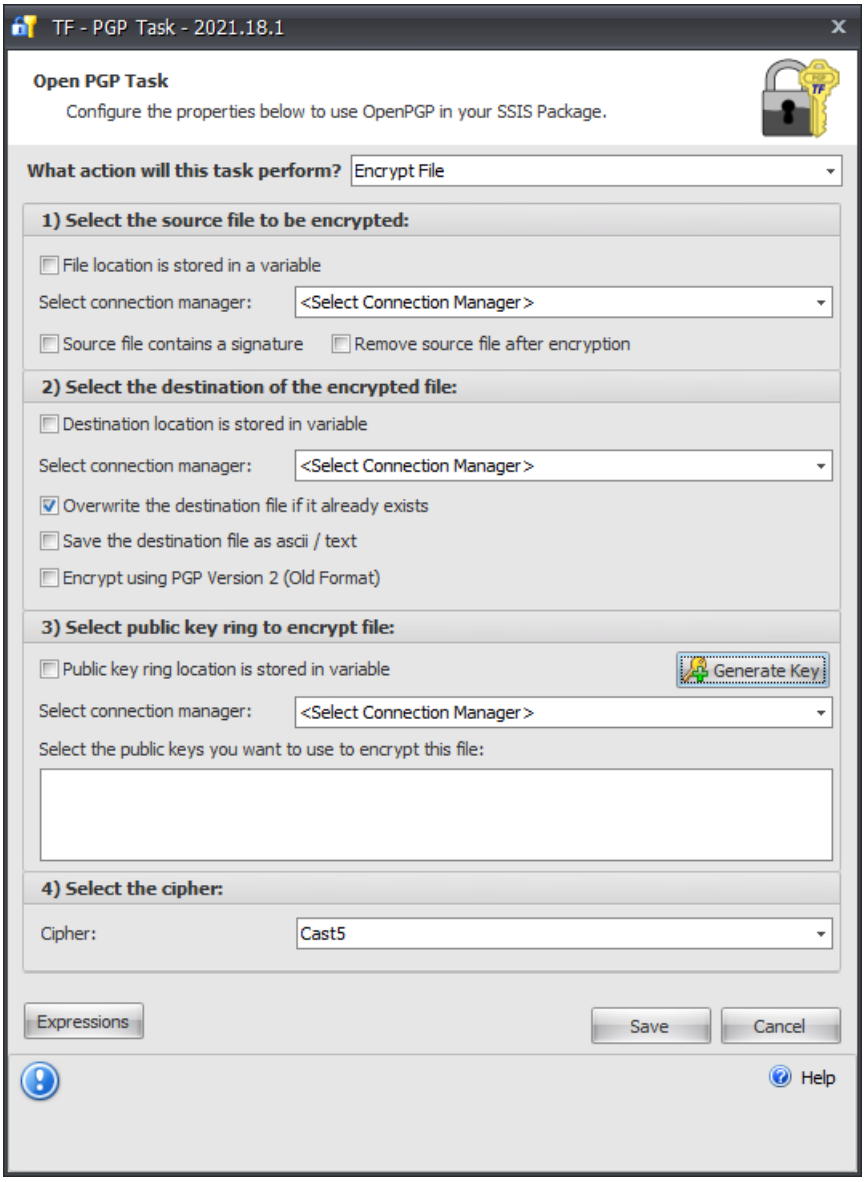
Last Modified on 09 August 2022

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

PGP Task

Task Icon	Task Description
	The **PGP Task** can be used to encrypt or decrypt a file using PGP encryption. It's also used to create and verify the signature of a file.

Encrypting a File



Select the source file to be encrypted

Option	Description
File location is stored in a variable	Selecting this option changes the **Select Connection Manager** drop-down menu to a drop-down containing the string variables in the SSIS package. The location of the source file must be stored in the variable selected.
Select Connection Manager	Available if **File location is stored in a variable** is not selected. Here the user selects or creates a connection manager that points to the source file.
Select Variable	Available if **File location is stored in a variable** is selected.
Source file contains a signature	With the PGP Task it's possible to create a file with a signature that is part of the file. Choose this option if the task encrypts a file with a signature.

Option	Description
Remove source file after encryption	This option removes the source file after the encryption process is finished.

Select the destination of the encrypted file

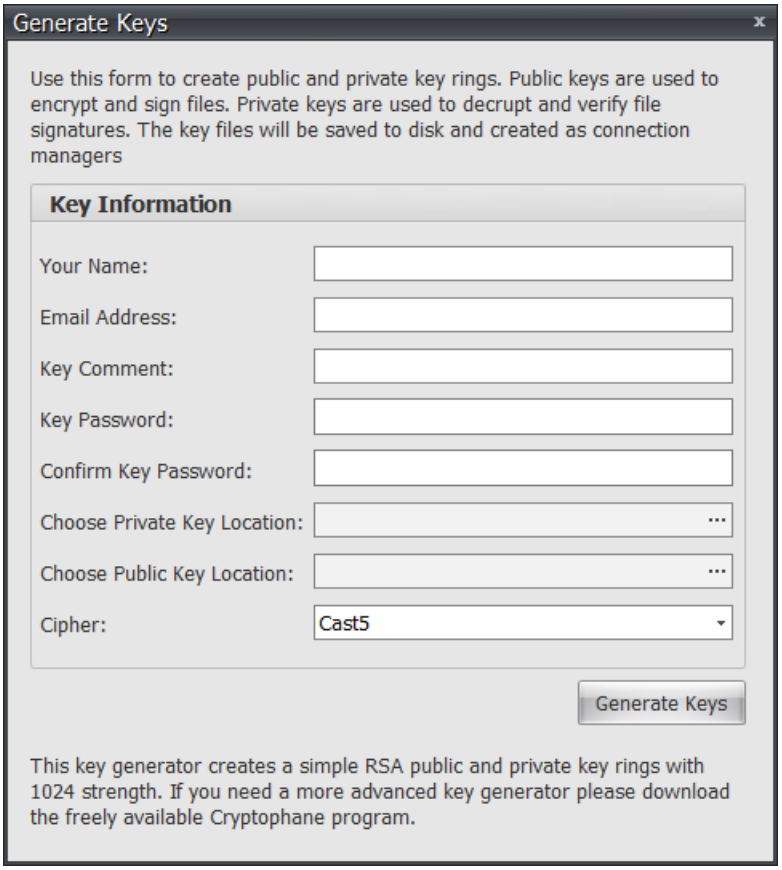
Option	Description
Destination location is stored in a variable	Selecting this option changes the **Select Connection Manager** drop-down menu to a drop-down containing the string variables in the SSIS package. The location of the destination file must be stored in the variable selected.
Select Connection Manager	Available if **Destination location is stored in a variable** is not selected. Here the user selects or creates a connection manager that points to the source file.
Select Variable	Available if **Destination location is stored in a variable** is selected.
Overwrite the file if it already exists	Selecting this option overwrites the destination file if it already exists. If the files exists, and this option is not selected, then the task fails.
Save the file as ASCII / text	Select this option if the task encrypts text and the text needs to be transmitted in an email or read by another text decryption program.
Encrypt using PGP Version 2 (Old Format)	This option is for legacy systems that need to decrypt the file using the old format of PGP.

Select public key ring to encrypt file

Option	Description
Public key ring location is store in variable	Selecting this option changes the **Select Connection Manager** drop-down menu to a drop-down containing the string variables in the SSIS package. The location of the public key ring file must be stored in the variable selected.
Select Connection Manager	Available if **Public key ring location is stored in variable** is not selected. Here the user selects or creates a connection manager that points to the source file.
Select Variable	Available if **Public key ring location is stored in variable** is selected.
Select the public keys you want to use to encrypt this file	This is where the available public keys form the public key ring file display. At least one public key must be selected to encrypt the file. Multiple public keys can be used to encrypt the file.

Generate Keys Button - The PGP Task (in encryption mode) can be used to generate key rings (public and private) to be used with the PGP task. Ideally these key rings are generated elsewhere but this is a good way

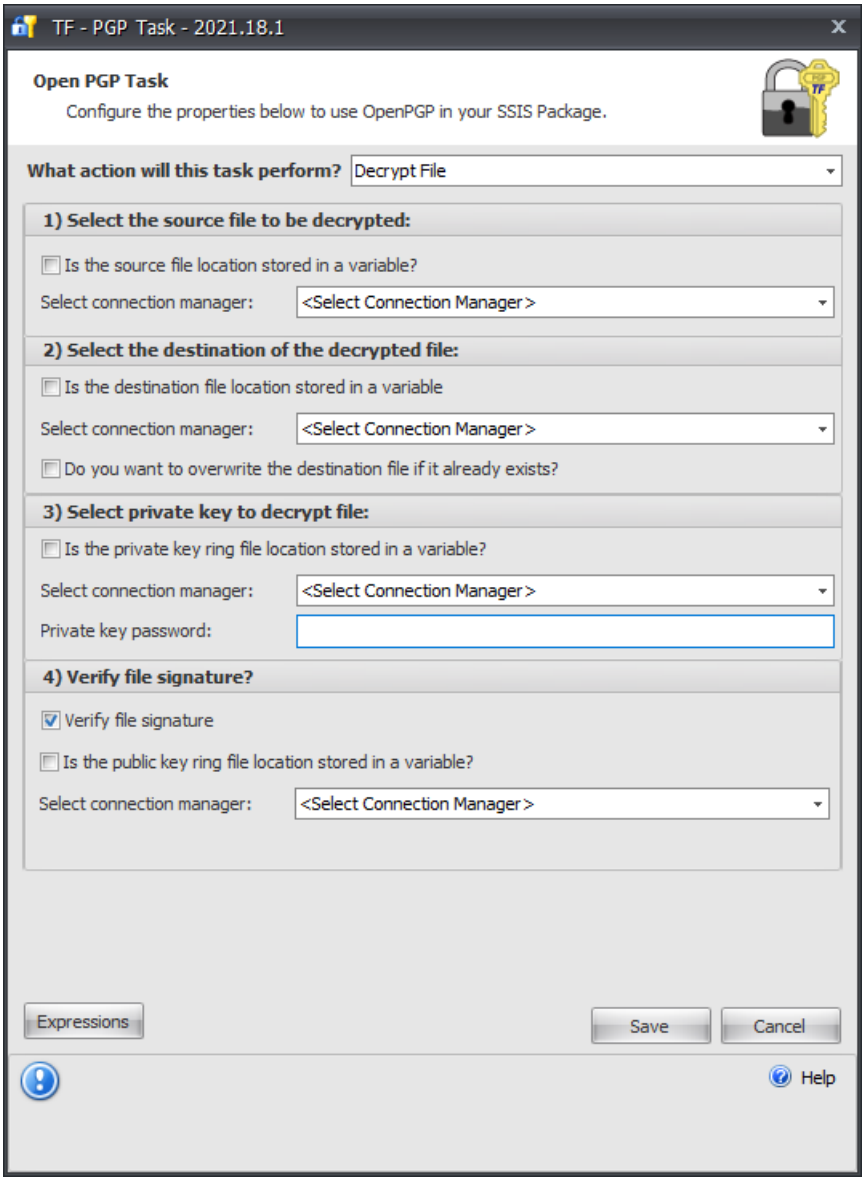
to test the PGP Task during development. To generate a key, simply fill out all of the data in the Key Information form and select **Generate Keys**.



Select the cipher

Option	Description
Cipher	Select the Cipher you want to use from the following options: • Aes128 • Aes192 • Aes256 • Blowfish • Camellia128 • Camellia192 • Camellia256 • Cast5 • Des • Idea • Safer • TripleDes • Twofish

Decrypting a File



Select the source file to be decrypted

Option	Description
Is the source file location stored in a variable	Selecting this option changes the **Select Connection Manager** drop down menu to a drop down containing the string variables in the SSIS package. The location of the source file must be stored in the variable selected.
Select Connection Manager	Available if **Is the source file location stored in a variable** is not selected. Here the user selects or creates a connection manager that points to the source file.
Select Variable	Available if **Is the source file location stored in a variable** is selected.

Select the destination of the decrypted file

Option	Description
Is the destination file location stored in a variable	Available if **Is the destination location is stored in a variable** is not selected. Here the user selects or creates a connection manager that points to the source file.
Select Connection Manager	Available if **Is the destination file location stored in a variable** is not selected. Here the user selects or creates a connection manager that points to the source file.
Select Variable	Available if **Is the destination file location stored in a variable** is selected.
Do you want to overwrite the destination file if already exists	Selecting this option overwrites the destination file if the file already exists. If the file exists and this option is not selected, then the task fails.

Select private key to decrypt the file

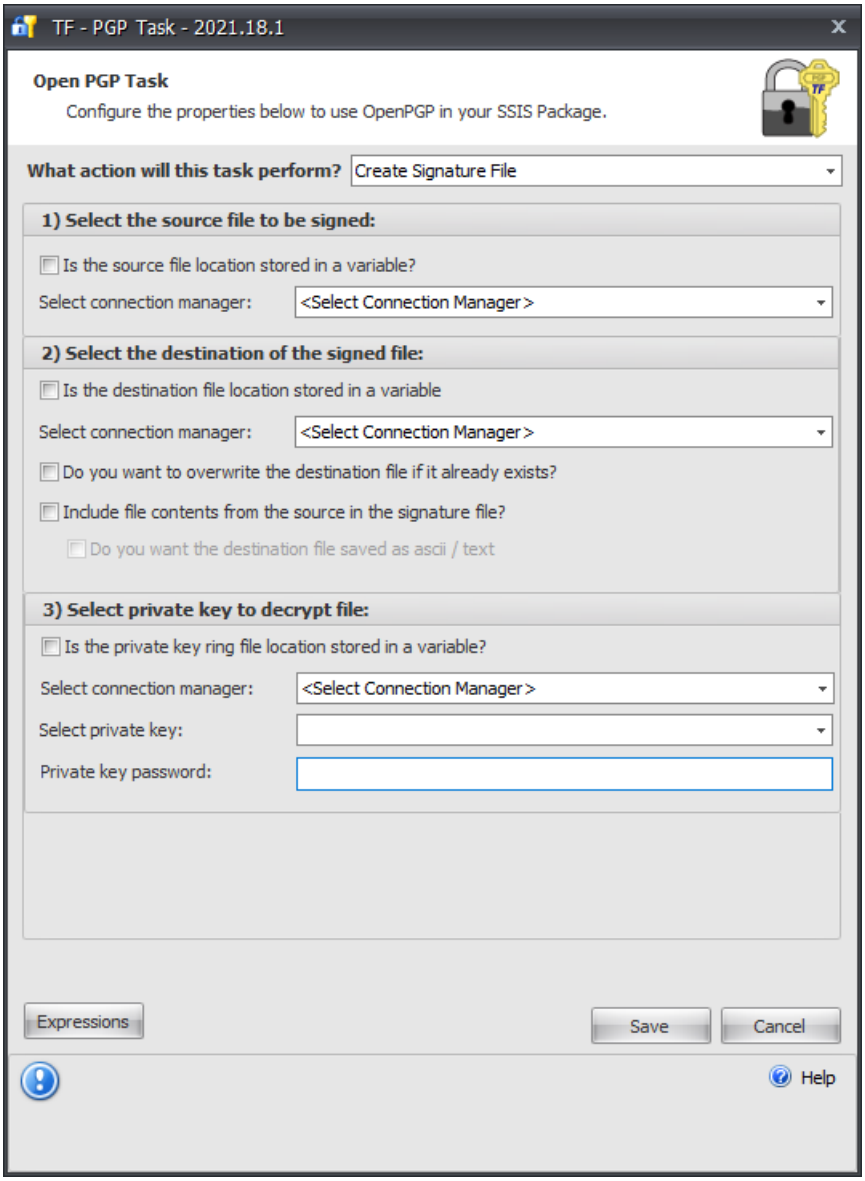
Option	Description
Is the private key ring file location stored in a variable	Selecting this option changes the **Select Connection Manager** drop down menu to a drop down containing the string variables in the SSI package. The location of the private key ring file must be stored in the variable selected.
Select Connection Manager	Available if **Is the private key ring location stored in variable** is not selected. Here the user selects or creates a connection manager that points to the source file
Select Variable	Available if **Is the private key ring location stored in variable** is selected.
Private key password	This is the password associated with the private key selected.

Verify file signature

Option	Description
Verify file signature	Selecting this option causes the task to look for an embedded file signature and verify it was signed using the public key ring selected.
Is public key ring file location stored in a variable	Selecting this option changes the **Select Connection Manager** drop down menu to a drop down containing the string variables in the SSI package. The location of the private key ring file must be stored in the variable selected.
Select Connection Manager	Available if **Is the public key ring location stored in variable** is not selected. Here the user selects or creates a connection manager that points to the source file.

Option	Description
Select Variable	Available if **Is the public key ring location stored in variable** is selected.

Create Signature File



Select the source file to be signed

Option	Description
Is the source file location stored in a variable	Selecting this option changes the **Select Connection Manager** drop down menu to a drop down containing the string variables in the SSIS package. The location of the source file must be stored in the variable selected.

Option	Description
Select Connection Manager	Available if **Is the source file location stored in a variable** is not selected. Here the user selects or creates a connection manager that points to the source file.
Select Variable	Available if **Is the source file location stored in a variable** is selected.

Select the destination of the signed file

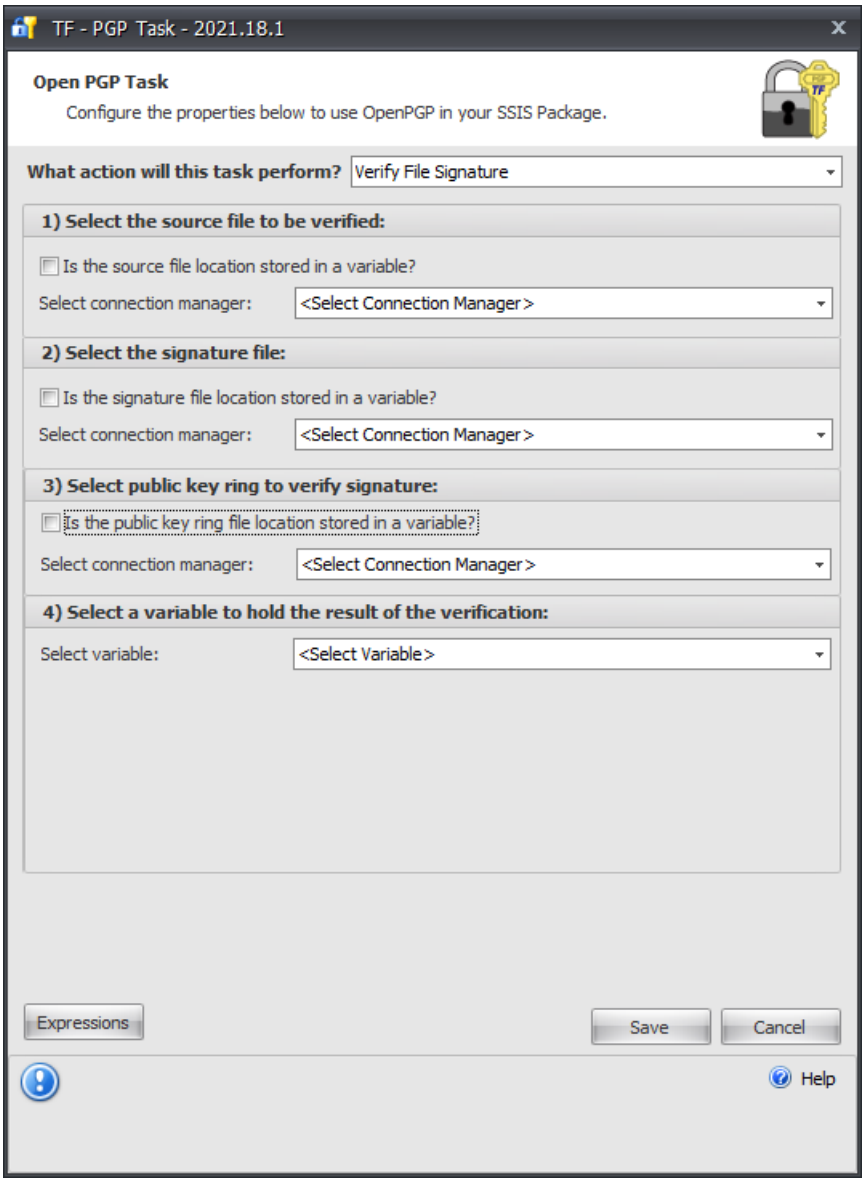
Option	Description
Is the destination file location stored in a variable	Selecting this option changes the **Select Connection Manager** drop down menu to a drop down containing the string variables in the SSIS package. The location of the destination file must be stored in the variable selected.
Select Connection Manager	Available if **Is the destination file location stored in a variable** is not selected. Here the user selects or creates a connection manager that points to the source file.
Select Variable	Available if **Is the destination file location stored in a variable** is selected.
Do you want to overwrite the destination file if already exists	Selecting this option overwrites the destination file if the file already exists. If the file exists and this option is not selected then the task fails.
Include file contents from the source in the signature file	Selecting this option include the contents of the file being signed as part of the signature file. This is part of a two step process to sign and encrypt a file: 1. Create the signature file, with the **Include file contents...** check box selected. 2. Encrypt the signed file in another PGP Task (Encryption Mode) with the **Source File Contains a Signature** option selected.
Do you want the destination file saved as ASCII / text	This option saves the file as a text file. Only select this if the source file being signed is a text document.

Select a private key to decrypt file

Option	Description
Is the private key ring file location stored in a variable	Selecting this option changes the **Select Connection Manager** drop down menu to a drop down containing the string variables in the SSI package. The location of the private key ring file must be stored in the variable selected.

Option	Description
Select Connection Manager	Available if **Is the private key ring location stored in variable** is not selected. Here the user selects or creates a connection manager that points to the source file.
Select Variable	Available if **Is the private key ring location stored in variable** is selected.
Select private key	Select a private key from the key ring.
Private key password	This is the password associated with the private key selected.

Verify Signature File



Select the source file to be verified

Option	Description
Is the source file location stored in a variable	Selecting this option changes the **Select Connection Manager** drop down menu to a drop down containing the string variables in the SSIS package. The location of the source file must be stored in the variable selected.
Select Connection Manager	Available if **Is the source file location stored in a variable** is not selected. Here the user selects or creates a connection manager that points to the source file.
Select Variable	Available if **Is the source file location stored in a variable** is selected.

Select the signature file

Option	Description
Is the signature file location stored in a variable	Selecting this option changes the **Select Connection Manager** drop down menu to a drop down containing the string variables in the SSIS package. The location of the signature file must be stored in the variable selected.
Select Connection Manager	Available if **Is the signature file location stored in a variable** is not selected. Here the user selects or creates a connection manager that points to the source file.
Select Variable	Available if **Is the signature file location stored in a variable** is selected.

Select public key ring to verify signature

Option	Description
Is the public key file location stored in a variable	Selecting this option changes the **Select Connection Manager** drop down menu to a drop down containing the string variables in the SSIS package. The location of the signature file must be stored in the variable selected.
Select Connection Manager	Available if **Is the public key file location stored in a variable** is not selected. Here the user selects or creates a connection manager that points to the source file.
Select Variable	Available if **Is the public key file location stored in a variable** is selected.

Select a variable to hold the result of the verification

Option	Description

Option	Description
Select variable	Select the variable that contains the result of the verification. The variable either contains true or false.

Task Factory Secure Config

Last Modified on 09 August 2022

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Important: The SecureConfig Task has no value within SSIS and doesn't perform any ETL processing. It can only be used with BIxPress.

Secure Config Task

Task Icon	Task Description
	The **SecureConfig Task** allows users to encrypt/decrypt configuration files within BIxPress. This component does not have a custom editor and is configured from the properties window. For more information, see the BIxPress Config Tools article.

Properties

TF SecureConfig Task Task

ConfigurationInformationVariable	
ContinueOnFailure	True
DelayValidation	**False**
Description	**TF SecureConfig Task**
Disable	**False**
DisableEventHandlers	**False**
ExecValueVariable	**<none>**
Expressions	
FailPackageOnFailure	**False**
FailParentOnFailure	**False**
ForcedExecutionValue	**0**
ForcedExecutionValueType	**Int32**
ForceExecutionResult	**None**
ForceExecutionValue	False
HasExpressions	False
ID	{12AF2E24-CBD6-4955-BC13-3F727A16F2DE}
IsolationLevel	**Serializable**
KeySource	DefaultKey
KeySourceValue	
LocaleID	**English (United States)**
LoggingMode	**UseParentSetting**
MaximumErrorCount	**1**
Name	**TF SecureConfig Task**
PackagePath	\Package\TF SecureConfig Task
TransactionOption	**Supported**
Version	0

Expressions

A collection of expressions. The evaluation result of each expression is assigned to a property and replaces the value of the property.

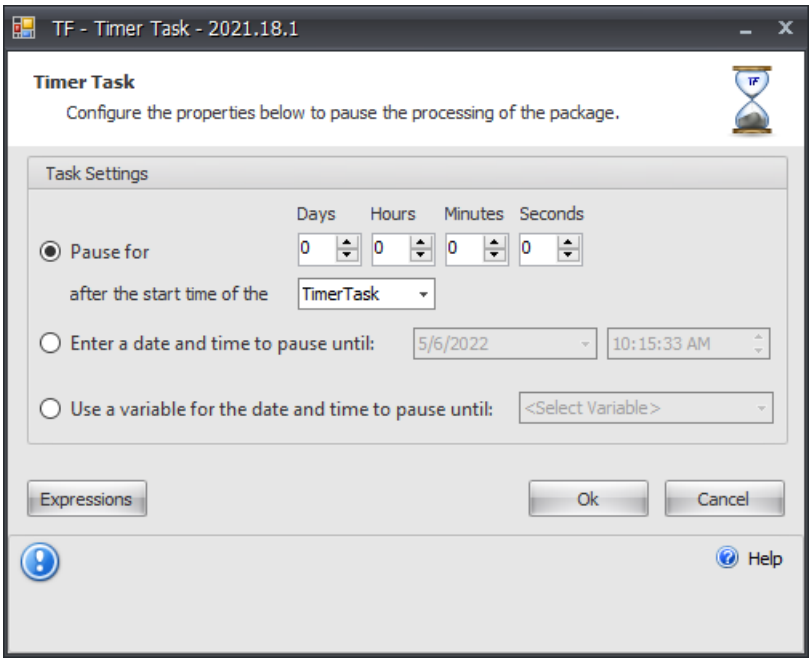
Task Factory Timer

Last Modified on 09 August 2022

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Timer Task

Task Icon	Task Description
	The **Timer Task** gives users the ability to configure a timer (delay) before beginning package execution. The component can also be used to add a delay between control flow tasks.



Option	Description
Pause for	The time to set the delay.

Option	Description
After the start time of the	Selection for Timer Task or Package. • Selecting **Package** delays execution of the package for the configured time. • Selecting **Timer Task** delays the control flow stream for the specified time when it reaches this component.
Enter a date and time to pause until:	Users can enter a specific date and time to delay a package or control flow.
Use a variable for the date and time to pause until	Users can dynamically set a delay by using a variable.

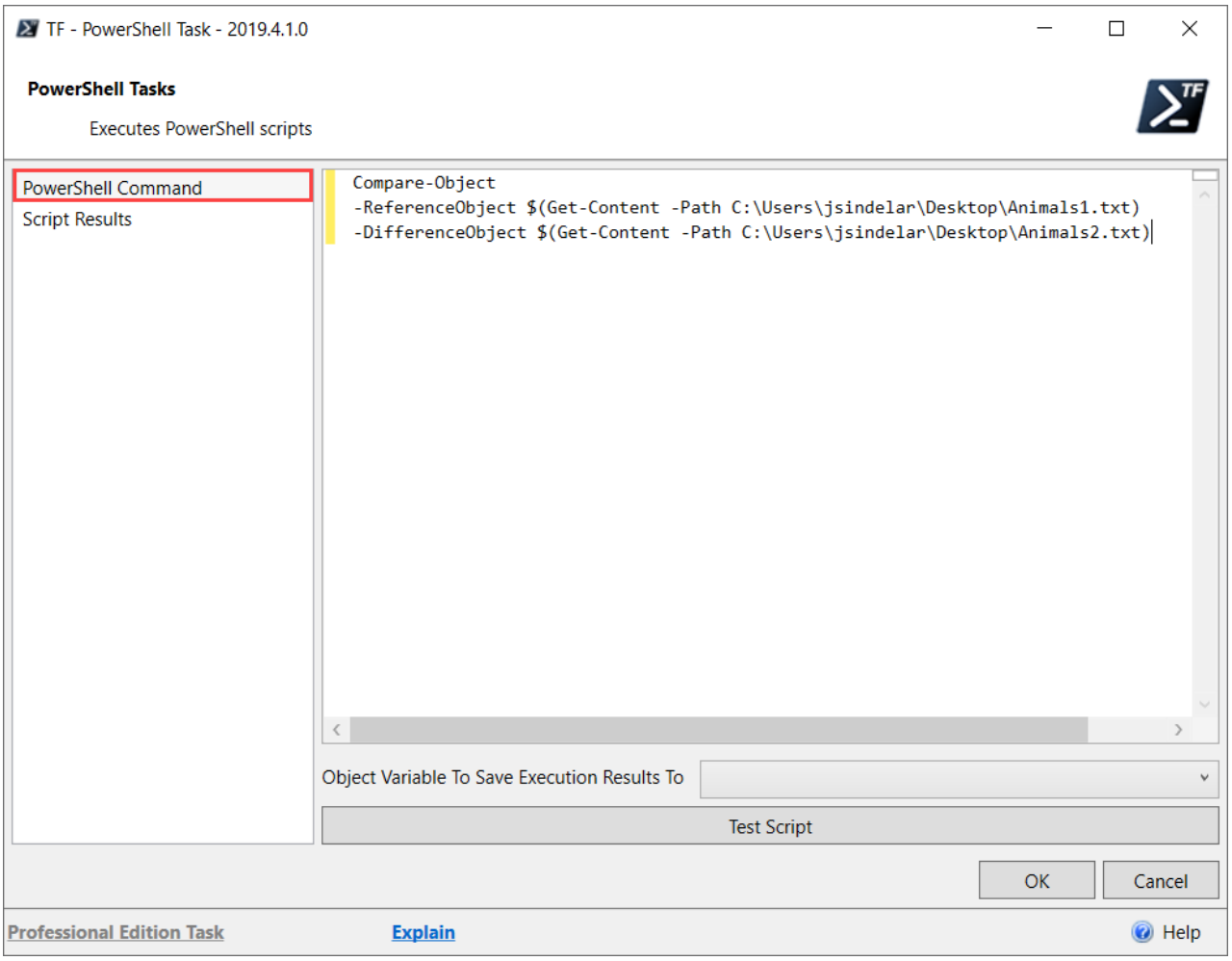
Task Factory PowerShell Task

Last Modified on 09 August 2022

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Task Icon	Task Description
	The **PowerShell Task** is a control flow component that executes a user-entered PowerShell script. Use the **PowerShell Task** to run PowerShell scripts directly in your SSIS package.

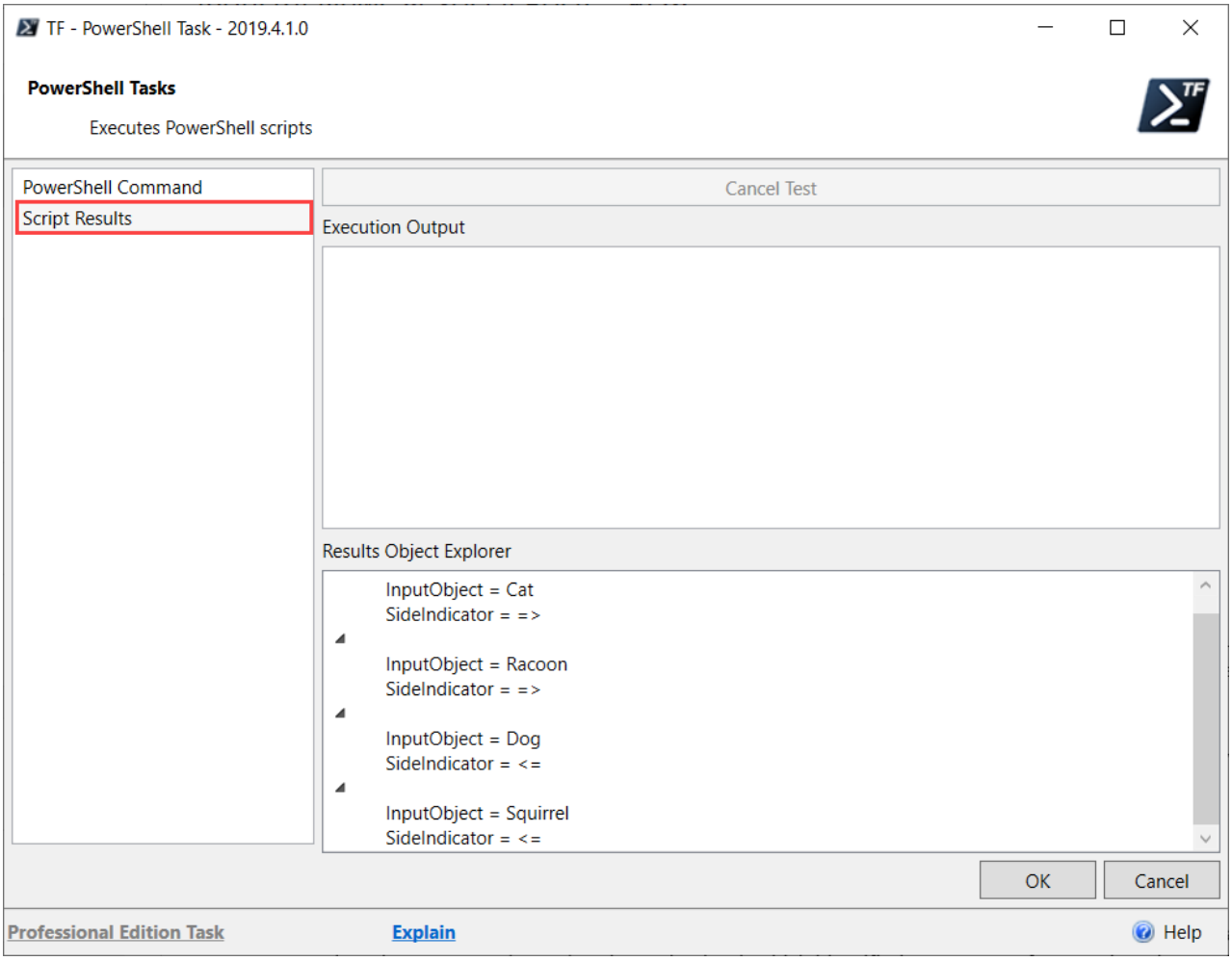
PowerShell Command



Component	Description

Component	Description
Command Window	Enter the PowerShell command you want to execute. **Note:** Variable replacements can be used with the following syntax: **<@User::VariableName>**
Object Variable To Save Execution Results To	Select the Object Variable used to save your task execution results. **Note:** You can only select this option if your object is a datatype.
Test Script	Select **Test Script** to display the Execution Output and Results Object Explorer data for the entered script.

Script Results



Component	Description
Execution Output	Displays any execution messages associated with your PowerShell script (error messages, warnings, etc.).

Component	Description
Results Object Explorer	Displays the sample results of your PowerShell script.

Additional Information: For information about using PowerShell Commands in SSIS, see the PowerShell MSDN documentation.

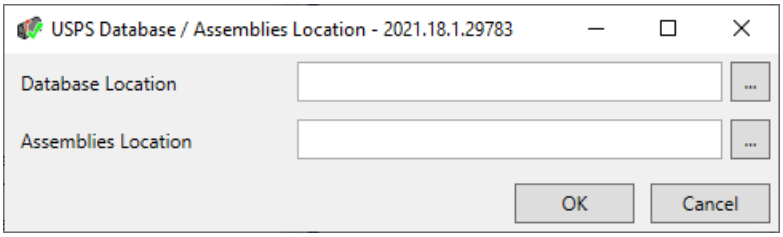
Task Factory Address Management

Last Modified on 10 August 2022

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Address Verification Connection Manger

Icon	Description
	The **Address Verification Connection Manager** is used to connect to your USPS database and Assemblies.

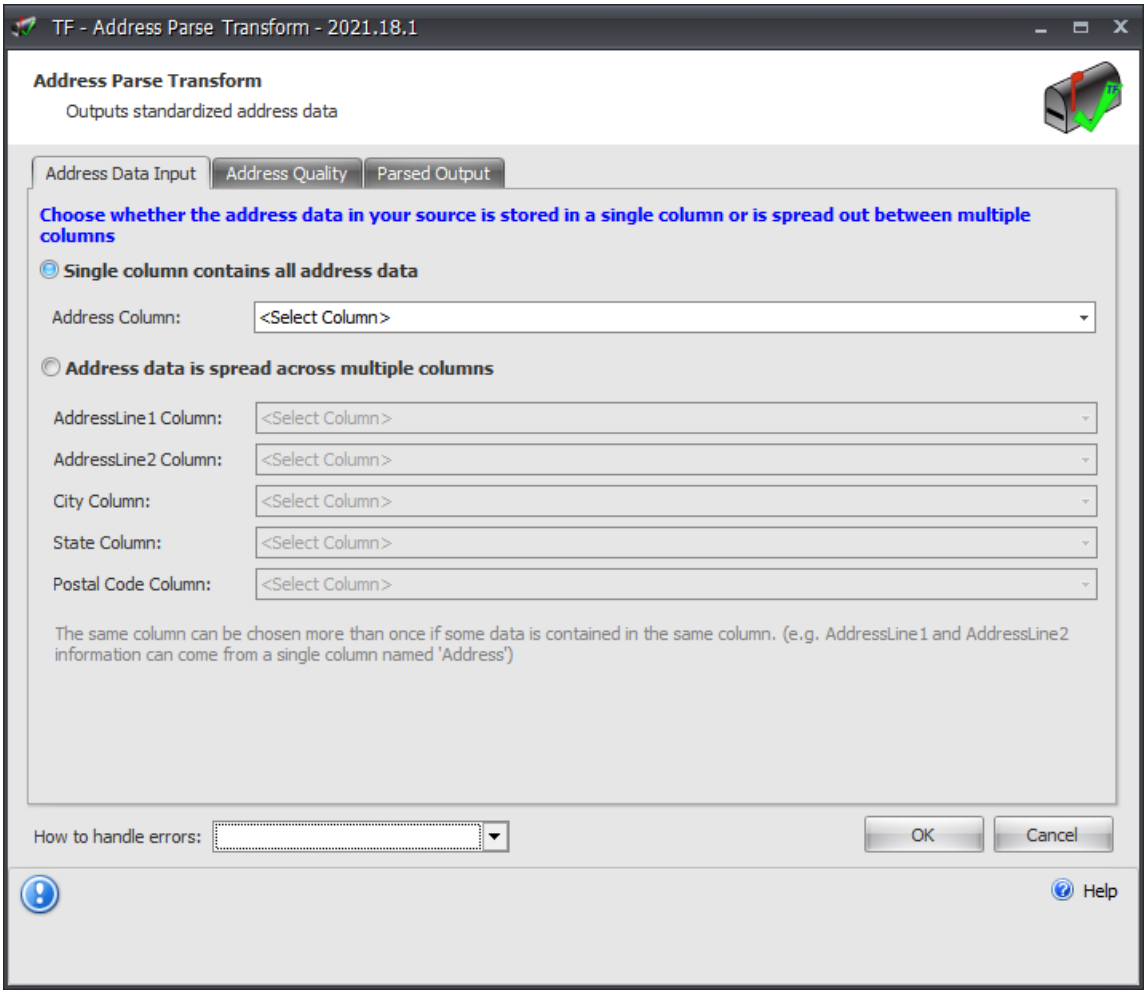


Option	Description
Database Location	The file path to the **dvd_comm** extracted folder. Select ··· to open the Browse For Folder window, and then select the appropriate file path. Select **OK** to save your connection.
Assemblies Location	The file path to the **ams_devkit** extracted folder. Select ··· to open the Browse For Folder window, and then select the appropriate file path. Select **OK** to save your connection.

Address Parse Transform

Transform Icon	Transform Description
	The **Address Parse Transform** is used to parse unformatted address data and transforms it into USPS standardized address data.

Address Data Input



Input Options

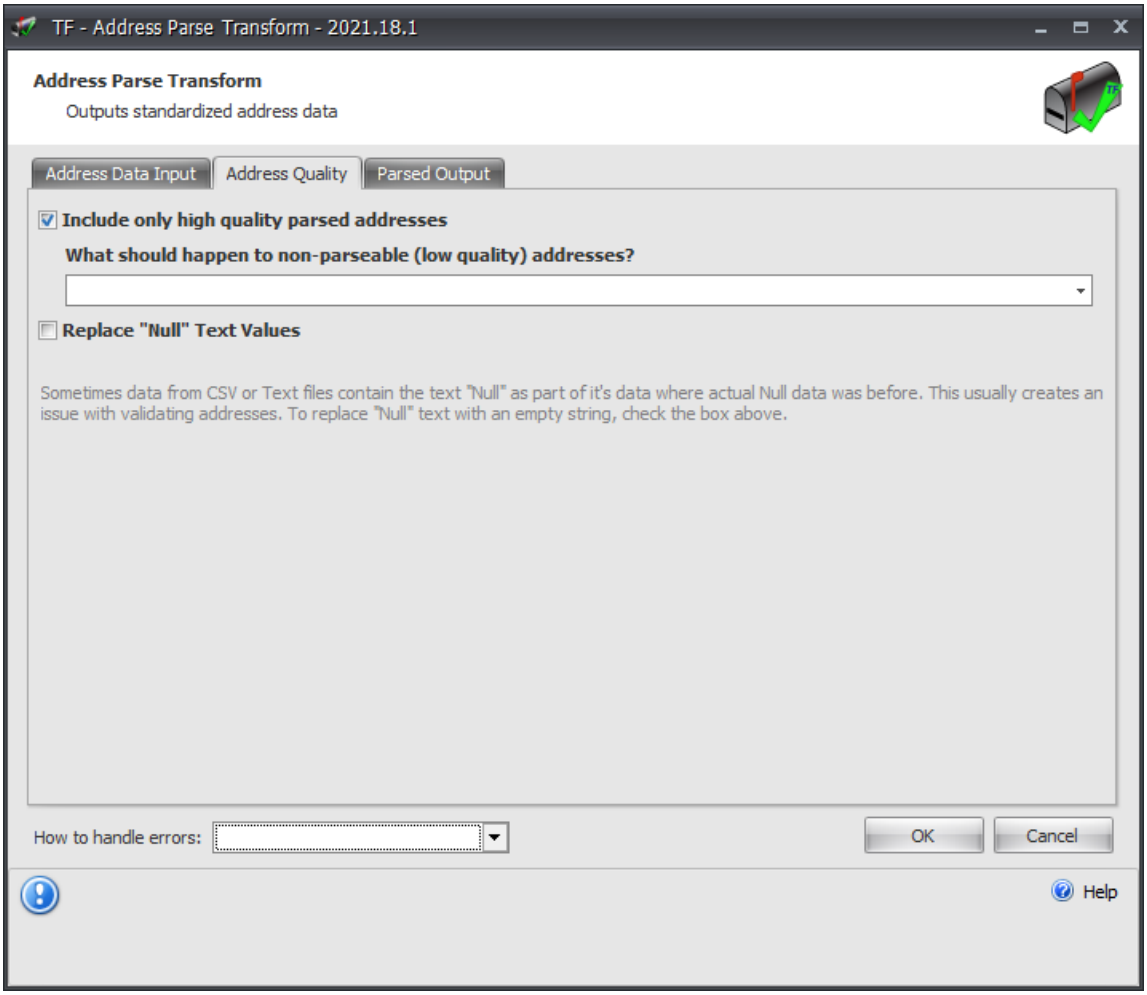
Option	Description
Single column contains all address data	Use this option when all of the address data from your source is contained in a single column.
Address data is spread across multiple columns	Use this option when the address data from your source is not contained in a single column. You choose which column from your source contains the address element needed by the transform.

Use the address data input tab to map the source input columns to the appropriate address elements in the address parse transform. Your source does not have to contain a one to one relationship to the address elements in the address parse. You can map a single column from your source to multiple address elements in the transform. For example:

AddressLine1 column:	Address
AddressLine2 column:	Address
City column:	CityStateZip

State column:	CityStateZip
PostalCode column:	CityStateZip

Address Quality



The address parse assigns a level of quality to two sections of the address:

1. Address Line 1 and Address Line 2
2. City, State, and Zip

There are three levels of quality that can be assigned to the two sections of an address:

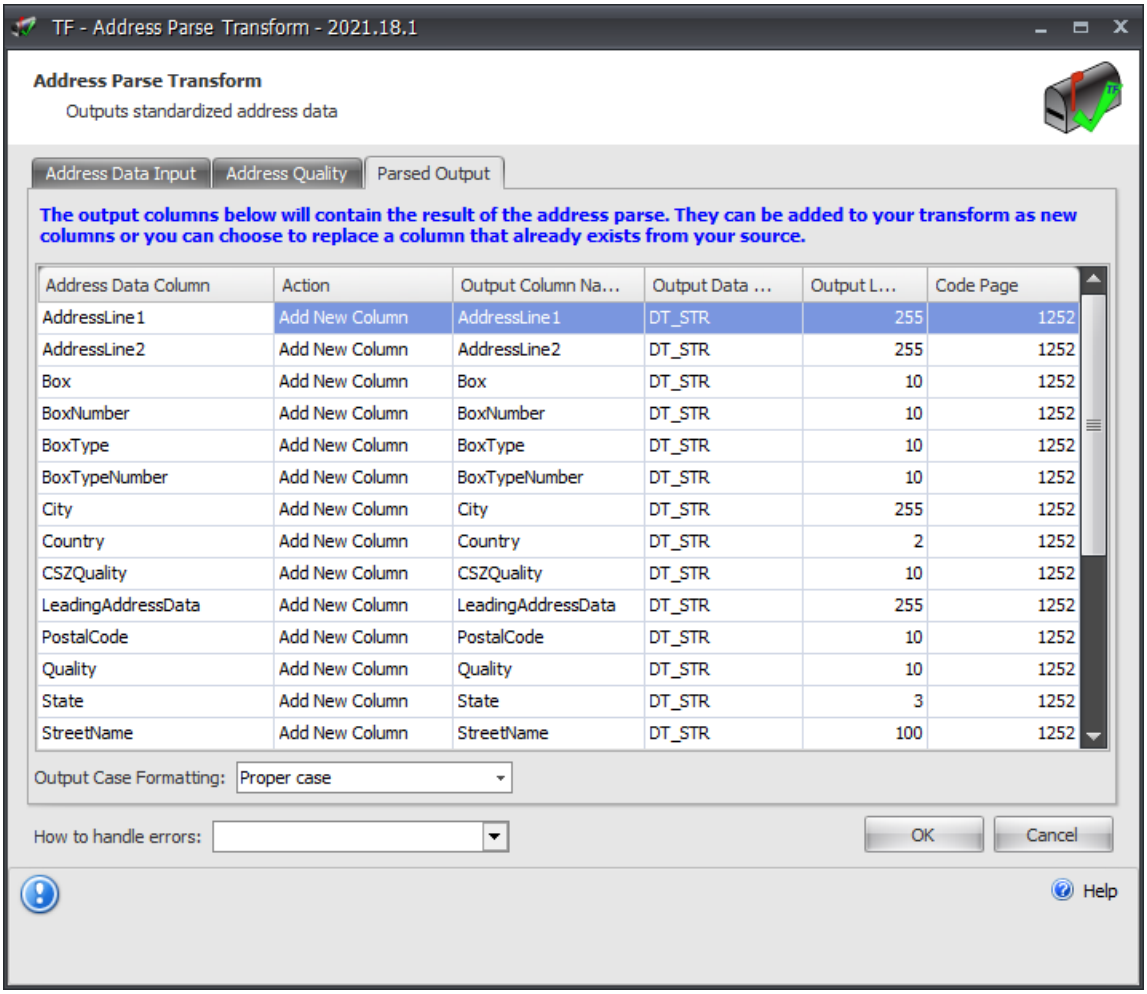
High	Medium	Low

High	Medium	Low
• Address Line 1 and Address Line 2 - The address is technically complete • City, State, Zip - These fields are technically correct	• Address Line 1 and Address Line 2 - The address is complete but missing a part of the address like apartment number or street suffix	• For Address Line 1 and Address Line 2 - The address is not verifiable • For City, State, Zip - Either the City, State, or Zip Code is missing

When the address parse runs it parses out the data and assigns the level of quality based on whether it can find a valid address and a valid city, state, and zip according to the rules above. You can either include all of the address as part of the output from the transform or you can select **Include only high quality parsed addresses**. Selecting this means that the output named **Parsed Address Output** contains addresses that are considered high quality. If you choose this option you must also make a selection from the **What should happen to non-parseable (low quality) addresses?** drop down menu.

Option	Description
Ignore Failures	This option skips the addresses marked as low quality.
Redirect Errors to Error Output	This option redirects the rows to the error output.
Fail Component	This option fails the component when the first invalid address is found.
Redirect Errors to Non-Parseable Output	This option redirects the rows to the **Non-Parsed Address Output**.

Parsed Output



The **Parsed Output** tab defines how you want the parsed address data outputted from the transform. By default there are new columns for each address elements and quality elements as part of the **Parsed Address Output**.

Output Columns

Column	Description
Address Data Column	The address data column defines the data that is output from the transform.
Action Column	The action column defines whether the output contains a new column or the data in an existing column. • **Add New Column** - This tells the transform to add a new column to the output. The columns to the right of the action column are used to define the properties of the output column. The data for the respective address data column is contained in the output column defined in the **Output Column Name** column. • **Replace Column** - This tells the transform to replace the data in a source column instead of using an output column. You need to select the name of the source column to replace in the **Output Column Name** column. If this option is selected, all of the columns to the right of **Output Column Name** are now read-only.

Column	Description
Output Column Name	Defines either the name of the output column if **Add New Column** is selected in the **Action** column or the name of the source column to replace if **Replace Column** is selected.
Output Data Type	Choose the output data type.
Output Length	Define the length of the column.
Code Page	Change the code page.
Output Case Formatting	Defines how the output is determined: • **Proper Case** - This transforms all of the data in the parse address to proper case (First letter of every word is upper case). • **Upper Case** - This transforms all of the data in the parse address to upper case. • **No Case Formatting** - This does not transform the case of the data.

Address Verification Transform

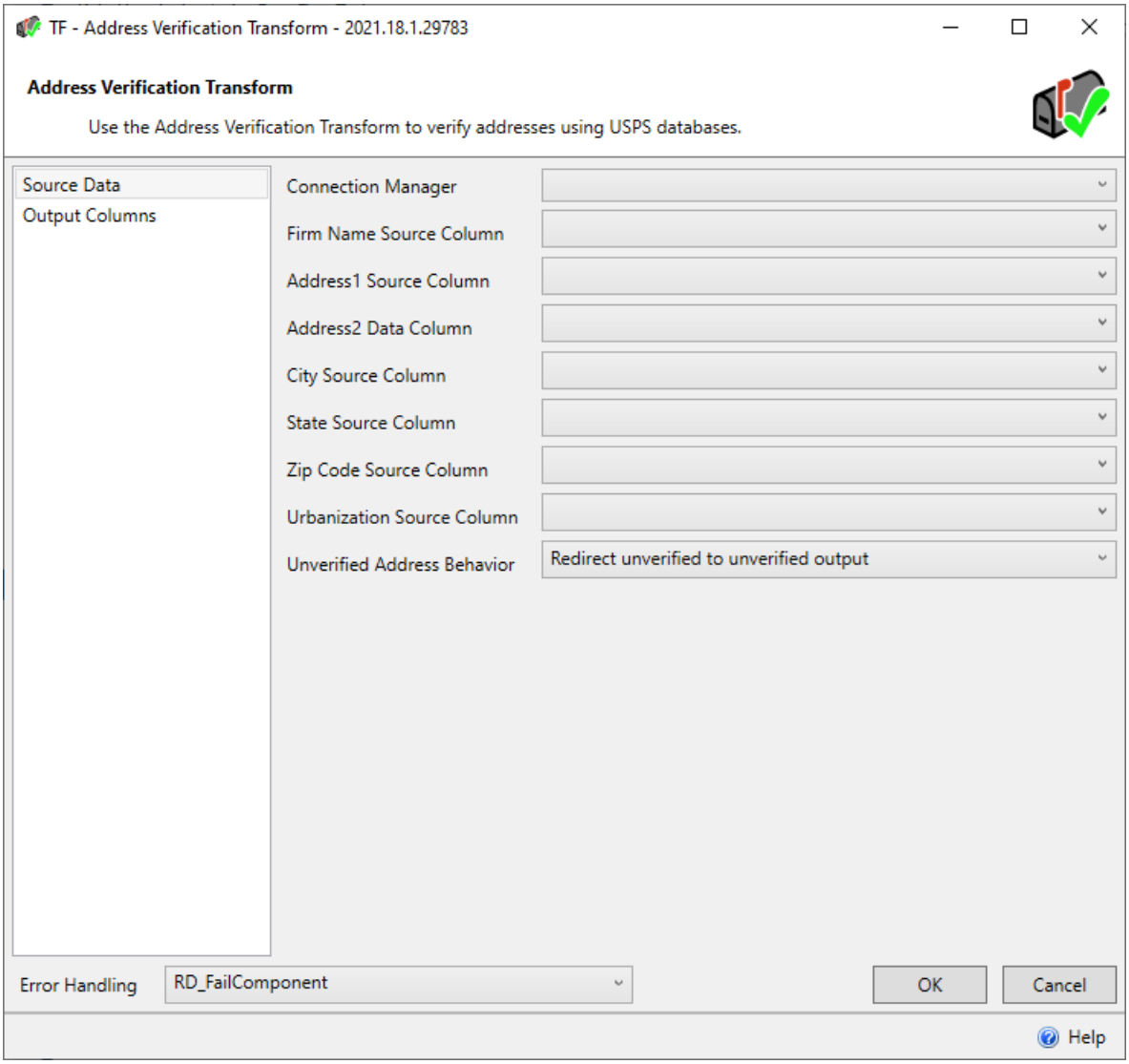
⚠ **Important:** Address Verification Transform is available for SQL versions 2012 and higher.

Address Verification Transform must be run in 32-bit mode. An internet connection is required to update the Address Verification database.

Transform Icon	Transform Description

Transform Icon	Transform Description
	The **Address Verification Transform** uses the USPS database to verify United States addresses.

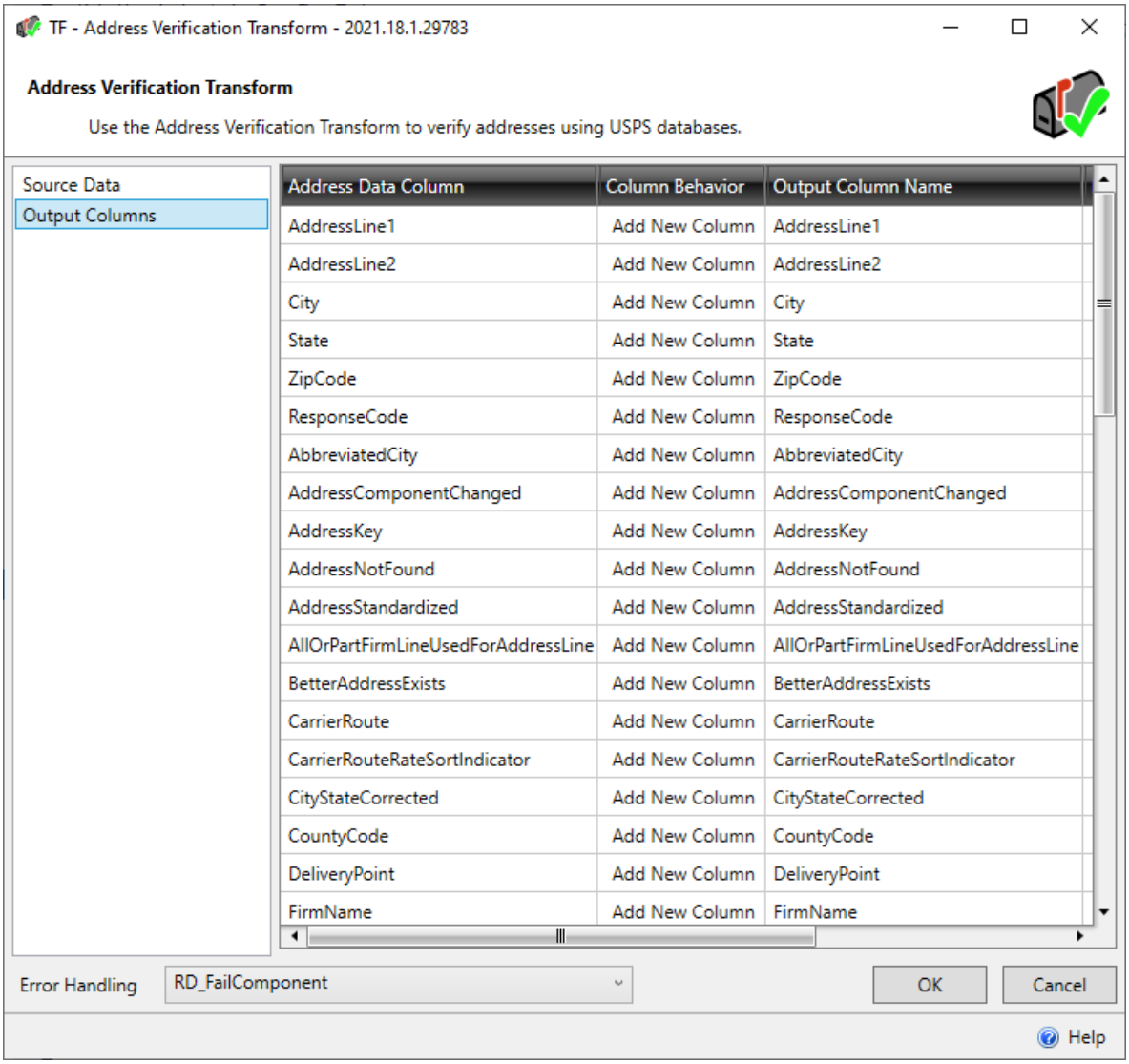
Source Data



Option	Description
Connection Manager	Select the Connection Manager for the Address Verification Transform.
Firm Name Source Column	(Optional) Maps the column that contains the person or firm's name.

Option	Description
Address1 Source Column	(Required) Maps the column that contains the first address line.
Address2 Data Column	(Optional) Maps the column that contains the second address line (e.g. suite or apartment numbers, building name, etc.)
City Source Column	(Required) Maps the column that contains the city.
State Source Column	(Required) Maps the column that contains the state.
Zip Code Source Column	(Required) Maps the column that contains the zip code.
Urbanization Source Column	(Optional) Maps the column that contains the urbanization code for Puerto Rico addresses.
Unverified Address Behavior	Options for handling unverified addresses: • **Ignore unverified (unverified addresses will be skipped)** - Skips the unverified address and continues execution. • **Redirect unverified to unverified output** - Redirects all unverified addresses to a designated unverified output. • **Fail upon first unverified address found** - Fails the component as soon as an unverified address is encountered.

Output Columns



Option	Description
Address Data Column	Identifies the column.
Column Behavior	Users can choose between the following: • Replace Column • Add New Column • Do Not Output Column
Output Column Name	Users can change the name of the output column.

The following list details the Address Verification Transform's output:

Column Name	Data Type	Length

AddressLine1	DT_WSTR	255
AddressLine2	DT_WSTR	255
City	DT_WSTR	255
State	DT_WSTR	3
ZipCode	DT_WSTR	10
ResponseCode	DT_I1	0
AbbreviatedCity	DT_WSTR	255
AddressComponentChanged	DT_BOOL	0
AddressKey	DT_WSTR	255
AddressNotFound	DT_BOOL	0
AddressStandardized	DT_BOOL	0
AllOrPartFirmLineUsedForAddressLine	DT_BOOL	0

BetterAddressExists	DT_BOOL	0
CarrierRoute	DT_WSTR	255
CarrierRouteRateSortIndicator	DT_WSTR	255
CityStateCorrected	DT_BOOL	0
CountyCode	DT_WSTR	255
DeliveryPoint	DT_WSTR	255
FirmName	DT_WSTR	255
FirstPostDirection	DT_WSTR	255
FirstPreDirection	DT_WSTR	255
FirstSuffix	DT_WSTR	255
IncorrectSecondaryNumber	DT_BOOL	0
InsufficientIncorrectData	DT_BOOL	0

InvalidCityStateZip	DT_BOOL	0
LookupStatus	DT_WSTR	255
MainPostOfficeOutputCity	DT_WSTR	255
MainPostOfficeOutputState	DT_WSTR	255
MatchedPrimaryNumber	DT_WSTR	255
MatchedSecondaryNumber	DT_WSTR	255
MilitaryMatch	DT_BOOL	0
MissingSecondaryNumber	DT_BOOL	0
MultipleResponseCanBeBrokenUsingTheLowestZip4	DT_BOOL	0
MultipleResponseDueToMagnetStreetSyndrome	DT_BOOL	0
NoMatchDueToEws	DT_BOOL	0
NonPoBoxDualAddress	DT_BOOL	0

NoZip4Assigned	DT_BOOL	0
PoBoxDualAddress	DT_BOOL	0
PrimaryName	DT_WSTR	255
PrimaryNumber	DT_WSTR	255
RuralRouteNumber	DT_WSTR	255
SecondaryNumber	DT_WSTR	255
SecondaryNumberUnit	DT_WSTR	255
SecondPostDirection	DT_WSTR	255
SecondPreDirection	DT_WSTR	255
SecondSuffix	DT_WSTR	255
SmallTownDefault	DT_BOOL	0
StandardizedOutputCityStateZip	DT_WSTR	255

StandardizedPrUrbanizationName	DT_WSTR	255
StreetNameChanged	DT_BOOL	0
UniqueZipCodeDefault	DT_BOOL	0
UniqueZipCodeMatch	DT_BOOL	0
UnofficialPostOfficeName	DT_BOOL	0
UnverifiableCityState	DT_BOOL	0
ZipCode4	DT_WSTR	255
ZipCodeAssignedWithMultipleResponse	DT_WSTR	255
ZipCodeCorrected	DT_BOOL	0
ZipMoveMatch	DT_BOOL	0

USPS Database Installation and Updates

⚠ **Important:** As of January 1, 2019, SentryOne no longer provides subscriptions or updates to the AVDB database. Customers need to subscribe and receive updates directly through the USPS's Electronic Product Fulfillment website. You must purchase the 32-bit version because the 64-bit version is not yet supported.

Previous customers that received updates through Pragmatic Works - the last update occurred in

November of 2018. Per the USPS, databases must be updated within 105 days of the previous update to maintain access. It's recommended customers update their database as soon as possible through the USPS's Electronic Product Fulfillment website to ensure they maintain access.

Additional Information: For more information about the pricing and fees for the USPS database subscription, see the following link.

Setting up the USPS Database for Task Factory

Complete the following steps to update your USPS database and assemblies when using the Task Factory Address Verification component:

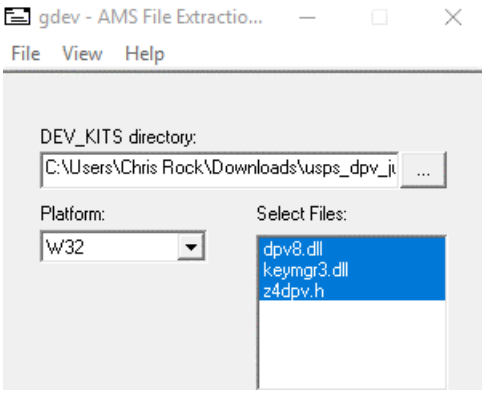
Step One : Download Assemblies and Database

1. Log in to the USPS's Electronic Product Fulfillment website, and then download the following files:
    - AMS Commercial DVD SHA-256
    - AMS Developer Kit SHA-256
    - DPV Developer Kit SHA-256

	Downloaded	AMS Commerical DVD SHA-256	2018-11-15	2018-10-19	dvd_comm.tar	3.6 GB
	Downloaded	AMS Developer Kit SHA-256	2018-11-15	2018-10-19	ams_devkit.tar	10.5 MB
	Downloaded	AMS Developer Kit Windows 32-bit (VS2017)	2018-11-15	2018-10-19	ams_vs2017_w32_devkit.tar	2.1 MB
	Downloaded	AMS Developer Kit Windows 64-bit (VS2017)	2018-11-15	2018-10-19	ams_vs2017_w64_devkit.tar	2.0 MB
	Downloaded	DPV® Developer Kit SHA-256	2018-11-15	2018-10-19	dpv_devkit.tar	1.1 MB

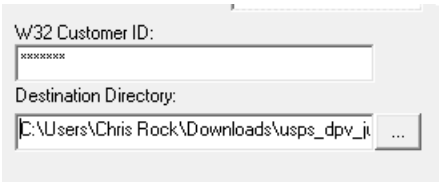
Step Two : DPV Extraction

1. Extract **dpv_devkit.tar** to the desired directory.
2. Run the **gdev.exe** application. **Note:** The **gdev.exe** application is located in the directory where **dpv_devkit.tar** was extracted.
3. Select the **W32 platform** and then select all the files in the **Select Files** section.

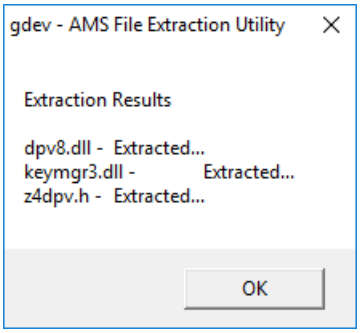


4. Enter your **USPS customer id** for the **W32 customer ID**. **Note:** You may need to email the USPS support team for this step.
5. Enter the destination directory. **Important:** Ensure you include the **** at the end of the path. For

example, **c:\usps\output** .

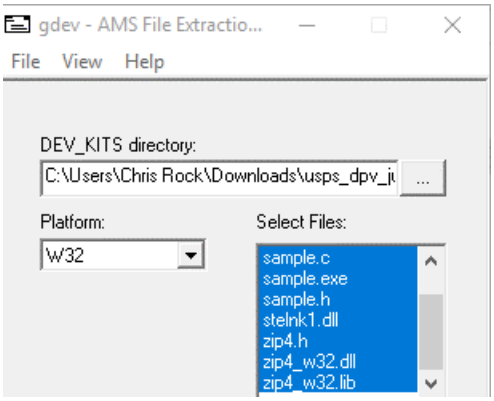


6. Select **Extract** to proceed with the extraction. ℹ **Note:** You should receive this message:

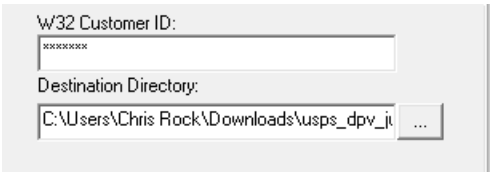


Step Three : AMS Extraction

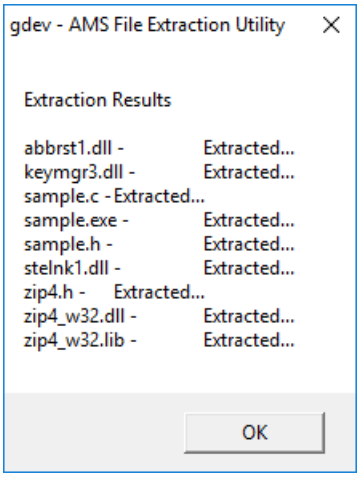
1. Extract the ams_devkit.tar file to the desired directory.
2. Run the **gdev.exe** application. ℹ **Note:** The **gdev.exe** application is located in the directory where **ams_devkit.tar** was extracted.
3. Select the **W32 platform** and then select all the files in the **Select Files** section.



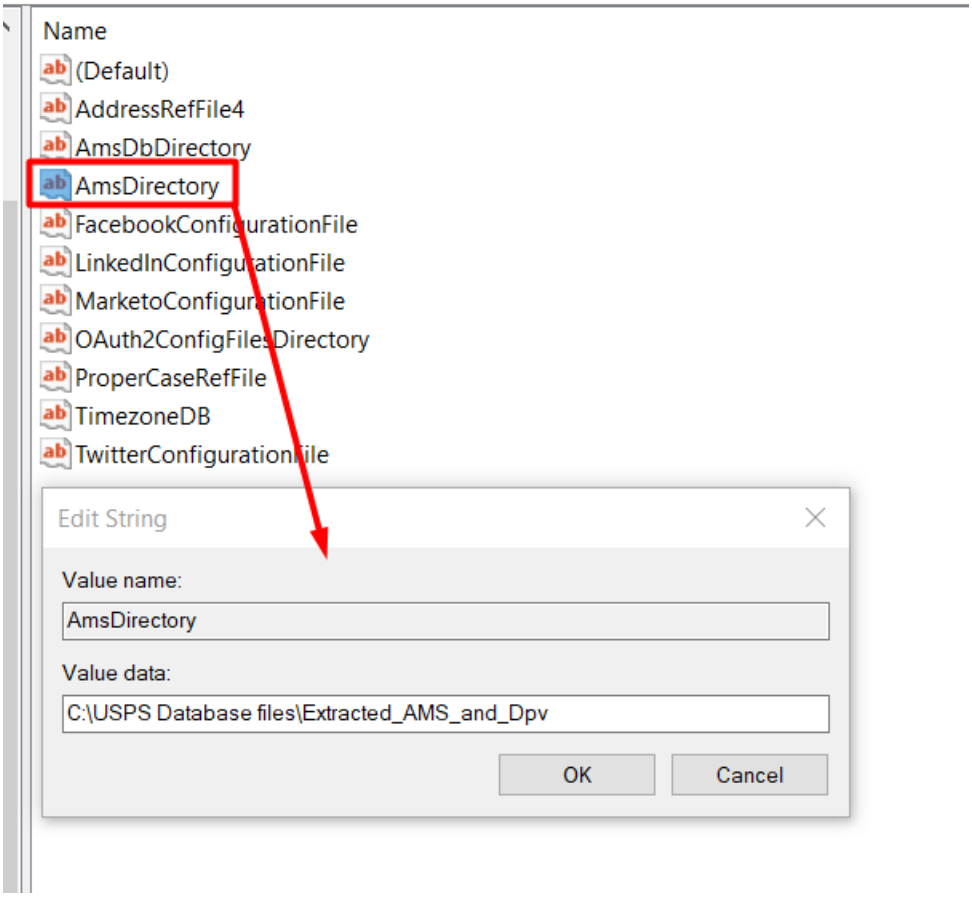
4. Enter your USPS customer id for the W32 customer ID. ℹ **Note:** You may need to email the USPS support team for this step.
5. Enter the destination directory. ⚠ **Important:** Ensure you include the **** at the end of the path. For example, **c:\usps\output** .



6. Select **Extract** to proceed with the extraction. ℹ **Note:** You should receive this message



7. Now copy the unencrypted files from both the AMS and DPV directories defined in the registry key **HKEY_LOCAL_MACHINE\SOFTWARE\Sentry One\Task Factory\AmsDirectory**. ℹ **Note:** You only need the .dll and .lib files.



ℹ **Note:** Now the directory should look like this :

Name	Date modified	Type	Size
abbrst1.dll	12/20/2018 5:25 PM	Application extens...	88 KB
dpv9.dll	12/20/2018 9:53 AM	Application extens...	84 KB
keymgr3.dll	12/20/2018 5:25 PM	Application extens...	24 KB
stelnk2.dll	12/20/2018 5:25 PM	Application extens...	128 KB
z4cxlog.dat	12/20/2018 12:56 ...	DAT File	1 KB
zip4_w32.dll	12/20/2018 5:25 PM	Application extens...	712 KB
zip4_w32.lib	12/20/2018 5:25 PM	Object File Library	33 KB

Step Four : USPS Database Extraction

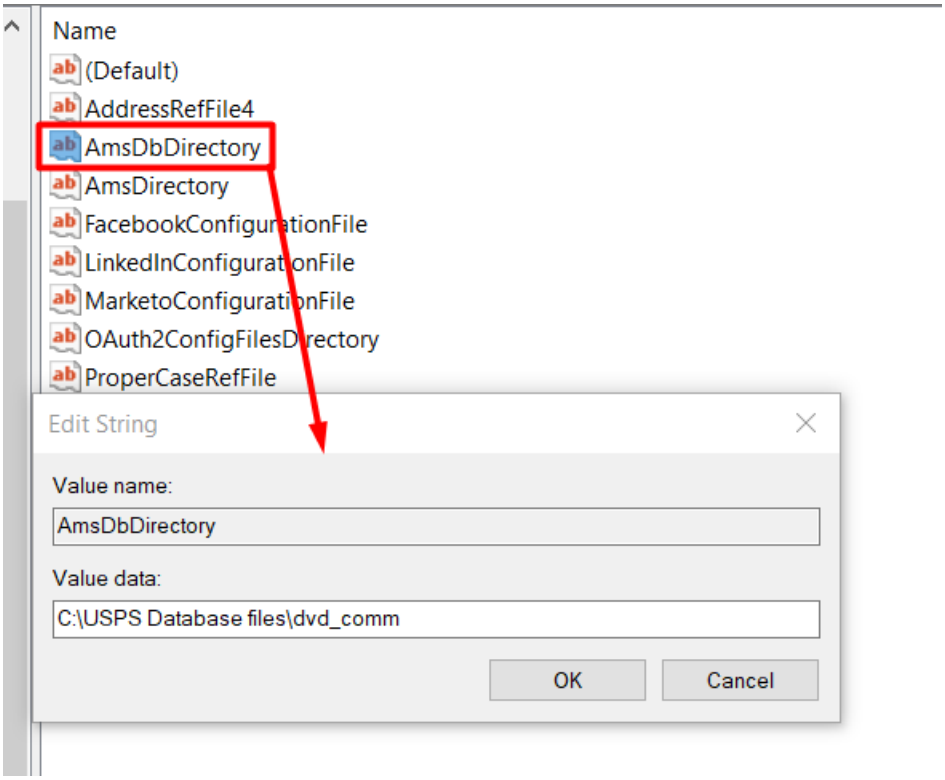
1. Extract the contents of **dvd_comm.tar** to a temp directory.
2. Rename the following:
    - **ams_comm** directory to **ams**
    - **ams_dpv** directory to **dpv**
    - **ams_elot** directory to **elot**

Note: Your directory should now look like this :

This PC › Local Disk (C:) › AMS

Name	Date modified	Type
ams	12/20/2018 12:46 ...	File folder
dpv	12/20/2018 12:45 ...	File folder
elot	12/20/2018 12:45 ...	File folder
lacslink	12/20/2018 12:46 ...	File folder
suitelink	12/20/2018 12:45 ...	File folder

3. Copy all of the directories from the temp directory to the directory defined in the registry key **HKEY_LOCAL_MACHINE\SOFTWARE\Sentry One\Task Factory\AmsDbDirectory**.

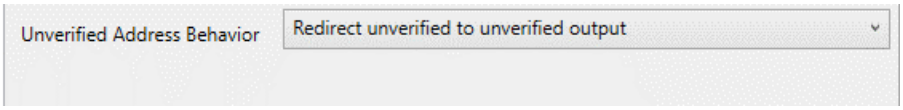


⚠ **Important:** The database expires after 105 days and users MUST update before this deadline to prevent the database from expiring. This is required per the USPS Database guidelines. Once the database has expired after the 105 day deadline, users have to contact the USPS vendor.

Understanding the USPS Address Verification Output

There are three outputs that are provided by the **Address Verification Transform**:

1. **Verified Address Output** – This output includes all addresses that are returned from the USPS database as either a ResponseCode of 31 (single address found) or 32 (address found, but more accurate address could be located). See response status table for full descriptions.
2. **Invalid Address Output** – This output includes all addresses that are returned from the USPS database with a response code of anything other than 31 or 32 .
3. **Multiple Address Output** – This output includes the other addresses found when an address is returned with a status of 22 (multiple addresses found). The address record with a status of 22 sends out of the Invalid Address Output if the **Unverified Address Behavior** is set to **Redirect unverified to unverified output** in the:

4. **Error Output** – This output contains any addresses where the data from the input column mapped to the **Address 1 Source Column** contains an empty or null value.

Output Columns

Verified Address Output Column Definitions

Column Name	Definition
AddressLine1	The standardized address line1 from the USPS database.
AddressLine2	The standardized address line2 from the USPS database.
City	The standardized city from the USPS database.
State	The standardized state from the USPS database.
ZipCode	The standardized zip code from the USPS database.
ResponseCode	The response code returned from the USPS database. See Response Code Table for the codes and definitions.
AbbreviatedCity	The abbreviated city.
AddressComponentChanged	An address component (i.e., directional or suffix only) was added, changed, or deleted in order to achieve a match.
AddressKey	Internal address database key (for indexing).

Column Name	Definition
AddressNotFound	Set to true if the address wasn't found.
AddressStandardized	Set to true if the address was standardized. For example, if STREET was in the delivery address, the system returns ST as its standard spelling.
AllOrPartFirmLineUsedForAddressLine	Set to true if information in the firm line was determined to be a part of the address. It was moved out of the firm line and incorporated into the address line.
BetterAddressExists	Set to true if the delivery address can be matched, but is known by another (preferred) name. For example, in New York, NY, AVENUE OF THE AMERICAS is also known as 6TH AVE. An inquiry using a delivery address of 55 AVE OF THE AMERICAS would set this value to true.
CarrierRoute	The USPS 4 digit carrier rate code.
CarrierRouteRateSortIndicator	Set to A, B, C, or D : • A = CR Sort Rates Apply Merge Allowed • B = CR Sort Rates Apply Merge Not Allowed • C = CR Sort Rates Do Not Apply Merge Allowed • D = CR Sort Rates Do Not Apply Merge Not Allowed

Column Name	Definition
CityStateCorrected	The spelling of the city name and/or state abbreviation in the submitted address was found to be different than the standard spelling. The standard spelling of the city name and state abbreviation are shown in the output address.
CountyCode	The USPS county code for the address.
DeliveryPoint	The USPS delivery point for the address.
FirmName	The standardized firm name.
FirstPostDirection	First or Left Post-direction.
FirstPreDirection	First or Left Pre-direction.
FirstSuffix	First or Left Suffix.
IncorrectSecondaryNumber	Set to true if the secondary number was incorrect.
InsufficientIncorrectData	Set to true if Insufficient / Incorrect data is found.
InvalidCityStateZip	Set to true if an invalid city, state and zip was found.

Column Name	Definition
LookupStatus	The return code value returned from the USPS database.
MainPostOfficeOutputCity	The city of the main post office for the standardized address.
MainPostOfficeOutputState	The state of the main post office for the standardized address.
MatchedPrimaryNumber	The primary address number.
MatchedSecondaryNumber	The primary secondary number.
MilitaryMatch	Set to true if a match was made with a military zip code.
MissingSecondaryNumber	Set to true if ZIP+4 information indicates this address is a building. The address as submitted does not contain an apartment/suite number. It is recommended that the requestor check the submitted address and add the missing apartment or suite number to ensure the correct Delivery Point Barcode (DPBC).
MultipleResponseCanBeBrokenUsingTheLowestZip4	More than one ZIP+4 Code was found to satisfy the address as submitted. The lowest ZIP +4 add-on may be used to break the tie between the records.

Column Name	Definition
MultipleResponseDueToMagnetStreetSyndrome	The search resulted in a single response; however, the record matched was flagged as having magnet street syndrome. “Whenever an input address has a single suffix word or a single directional word as the street name, or whenever the ZIP+4 File records being matched to have a single suffix word or a single directional word as the street name field, then an exact match between the street, suffix and/or post-directional and the same components on the ZIP+4 File must occur before a match can be made. Adding, changing or deleting a component from the input address to obtain a match to a ZIP+4 record will be considered incorrect.” Instead of returning a **no match** in this situation a multiple response is returned to allow access the candidate record.
NoMatchDueToEws	Set to true if the delivery address can be matched, but the EWS file indicates that an exact match will be available soon.
NonPoBoxDualAddress	CASS rule does not allow a match when the cardinal point of a directional changes more than 90%.
NoZip4Assigned	The ZIP Code in the submitted address could not be found because neither a valid city, state, nor valid 5-digit ZIP Code was present. It is also recommended that the requestor check the submitted address for accuracy.
PoBoxDualAddress	Set to true if the input contained two addresses. For example: 123 MAIN ST PO BOX 99.

Column Name	Definition
PrimaryName	The primary street name.
PrimaryNumber	The primary street number.
RuralRouteNumber	The rural route number.
SecondaryNumber	The secondary number of the street address.
SecondaryNumberUnit	The secondary number unit of the street address.
SecondPostDirection	Second or Right Post-direction.
SecondPreDirection	Second or Right Pre-direction.
SecondSuffix	Second or Right Suffix.
SmallTownDefault	Set to true because of an invalid delivery address. The input address record contains a delivery address other than a PO BOX, General Delivery, or Postmaster with a 5-digit ZIP Code that is identified as a “small town default.” The United States Postal Service does not provide street delivery for this ZIP Code. The United States Postal Service requires use of a PO BOX, General Delivery, or Postmaster for delivery within this ZIP Code.

Column Name	Definition
StandardizedOutputCityStateZip	The standardized city, state and zipcode.
StandardizedPrUrbanizationName	The standardized Puerto Rico urbanization.
StreetNameChanged	Set to true if the standardized address changed the street name.
UniqueZipCodeDefault	Set to true if a default match inside a unique ZIP Code.
UniqueZipCodeMatch	Set to true if an address with a unique ZIP Code was found.
UnofficialPostOfficeName	Set to true if the city or post office name in the submitted address is not recognized by the United States Postal Service as an official last line name (preferred city name), and is not acceptable as an alternate name. This does denote an error and the preferred city name will be provided as output.
UnverifiableCityState	Set to true if the city and state in the submitted address could not be verified as corresponding to the given 5-digit ZIP Code. This comment does not necessarily denote an error; however, it is recommended that the requestor check the city and state in the submitted address for accuracy.
ZipCode4	The last 4 digits of the standardized address zip code.

Column Name	Definition
ZipCodeAssignedWithMultipleResponse	Set to true if multiple records were returned, but each shares the same 5-digit ZIP Code.
ZipCodeCorrected	Set to true if the zip code was corrected when the address was standardized.
ZipMoveMatch	Set to true if the ZIPMOVE product shows which ZIP + 4 records have moved from one ZIP Code to another. If an input address matches to a ZIP + 4 record which the ZIPMOVE product indicates as having moved, the search is performed again in the new ZIP Code.

Unverified Address Output

This output contains all the input column data along with a column named **WhyInvalid** which tells you why the address was deemed invalid by the USPS database.

Error Output

This output contains all the input column data along with a column named **ErrorDescription** which tells you input data was deemed invalid by the address verification transform.

Multiple Address Output

Column Name	Description
ActionCode	A field indicating the action to be taken in updating the base master files.

Column Name	Description
AddOnHigh	The high-end ZIP add-on in a range of codes; the add-on of the last +4 code in a consecutive series of +4 codes assigned to the delivery address. Most delivery addresses are assigned a single +4 code; however, large companies may be given a range of +4 codes that can be used to route mail to specific departments. A range of +4 codes can also be assigned to a corresponding range of Post Office boxes, and each box is mapped, one-to-one, with a +4 code in the range. A single Post Office box can also be assigned a range of +4 codes. (See +4 Code.) Example: 12345-0001 through 12345-0009 => +4 High Number = 0009+4 codes associated with non-delivery areas are composed of valid ZIP™ sector numbers and “ND” for the ZIP segment number. Non-deliverable areas are areas to which the USPS does not deliver mail, such as vacant lots and land that borders railroad tracks. Mailers should not match an address identified as a non-delivery area. (See ZIP Sector Number or ZIP Segment Number.)

Column Name	Description
AddOnLow	The last four positions of a +4 code; the low-end +4 in a range of codes; the add-on of the first +4 code in a consecutive series of +4 codes assigned to the delivery address. For example, most delivery addresses are assigned a single +4 code. However, large companies may be given a range of +4 codes that can be used to route mail to specific departments. A range of +4 codes can also be assigned to a corresponding range of Post Office boxes, and each box is mapped, one-to-one, with a +4 code in the range. A single post office box can also be assigned a range of +4 codes. Example: 12345-0001 through 12345-0009 => +4 Low Number = 0001+4 Codes associated with non-delivery areas are composed of valid ZIP sector numbers and “ND” for the ZIP segment number. Non-deliverable areas are areas to which the USPS does not deliver mail, such as vacant lots and land that borders railroad tracks. Mailers should not match an address identified as a non-delivery area. (See ZIP Sector Number or ZIP Segment Number.)
BaseAlternateCode	Code that specifies whether a record is a base (preferred) or alternate record. Base records (represented as **B**) can represent a range of addresses or an individual address, such as a firm record, while alternate records (represented as **A**) are individual delivery points. Base Data Element Definitions Address Information System Products Technical Guide 62 records are generally preferred over alternate records. However, one exception is ZIP+4® product; in this file, government deliveries will only be listed on alternate records with the appropriate government building indicator (federal, state, or city) set.

Column Name	Description
CarrierRoute	A 4-byte code assigned to a mail delivery or collection route within a 5-digit ZIP™ Code. The first character of this identification is alphabetical, and the last three are numeric: • B### = PO box • H### = Contract • R### = Rural route • C### = City delivery • G### = General delivery
CongressionalDistrict	A standard value identifying a geographic area within the United States served by a member of the U.S. House of Representatives. If Army/Air Force (APO), Fleet Post Office (FPO), or Diplomatic/Defense Post Office (DPO), this field will be blank. If there is only one member of Congress within a state, the code will be **AL** (at large).
CopyrightDetailCode	Alphabetic code that identifies the type of record in a USPS® product file. • A = Alias record • C = Copyright record • D = Detail record • N = Seasonal record • S = 5-Digit ZIP scheme combinations record • Z = Zone split record
CountyNumber	The Federal Information Processing Standard (FIPS) code assigned to a given county or parish within a state. In Alaska, it identifies a region within the state. If APO/FPO/DPO, and the record type is **S**, **H**, or **F**, the county number will be blank.
FinanceCode	A code assigned to Postal Service™ facilities (primarily Post Offices) to collect cost and statistical data and compile revenue and expense data.

Column Name	Description
FirmOrBuildingName	The firm or building name.
LACSConvertedStatus	The Locatable Address Conversion Service (LACS) indicator describes records that have been converted to the LACS system (a product/system in a different USPS® product line that allows mailers to identify and convert a rural route address to a city-style address). Rural route and some city addresses are being modified to city-style addresses so that emergency services (e.g., ambulances, police) can find these addresses more efficiently. • L = LACS address: The old (usually rural-route) address that has been converted for the LACS system. • Blank = Not applicable
LastLine	A value that uniquely identifies a city state name in the City State product.
PostDirection	A geographic direction that follows the street name. Example: 123 MAIN ST N Street Post-Directional Abbreviation = N Spaces = No post-direction E = East N = North NE = Northeast NW = Northwest S = South SE = Southeast SW = Southwest W = West

Column Name	Description
PreDirection	A geographic direction that precedes the street name. Example: 123 N MAIN ST Street Pre-Directional Abbreviation = N Spaces = No pre-direction E = East N = North NE = Northeast NW = Northwest Data Element Definitions Address Information System Products Technical Guide 76 S = South SE = Southeast SW = Southwest W = West
PrimaryEvenOddCode	The address primary even/odd code. • O = Odd • E = Even • B = Both

Column Name	Description
PrimaryHighRange	A house, rural route, contract box, or Post Office box number. The numeric or alphanumeric component of an address preceding the street name. The high-end address in a range of addresses. Often referred to as house number. Examples: **Address Range** / **Primary High Number** 100 – 198 Main Ave / 198 101 – 199 Main Ave 199 / 199 A200 – A298 Main Ave A298 / A298 124-45 – 124-88 Main Ave / 124-88 101 – 120 PO Box / 120 22A – 22B RR 2 / 22B

Column Name	Description
PrimaryLowRange	A house, rural route, contract box, or Post Office box number. The numeric or alphanumeric component of an address preceding the street name. The low-end address in a range of addresses. Often referred to as house number Examples: **Address Range** / **Primary High Number** 100 – 198 Main Ave / 100 101 – 199 Main Ave 199 / 101 A200 – A298 Main Ave A298 / A200 124-45 – 124-88 Main Ave / 124-45 101 – 120 PO Box / 101 22A – 22B RR 2 / 22A

<table>
<tr><th>Column Name</th><th>Description</th></tr>
<tr><td>RecordType</td><td>The record type.

• G = General delivery
• M = Multi-carrier
• P = PO Box
• R = Rural Route
• S = Street</td></tr>
<tr><td>SecondaryEvenOddCode</td><td>The address secondary even/odd code.

• O = Odd
• E = Even
• B = Both</td></tr>
<tr><td>SecondaryHighRange</td><td>Code that describes an apartment, room, suite, space, floor, or other secondary addressing numeric or alphanumeric that follows a street address. The high-end secondary address component in a range of secondary addresses.

Examples:

<table>
<tr><th>Address Range</th><th>Address Secondary High Number</th></tr>
<tr><td>00000001 through 00000009</td><td>00000009</td></tr>
<tr><td>A through F</td><td>F</td></tr>
<tr><td>66A through 66E</td><td>66E</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Column Name</th><th>Description</th></tr>
<tr><td>SecondaryLowRange</td><td>Code that describes an apartment, room, suite, space, floor, or other secondary addressing numeric or alphanumeric that follows a street address. The low-end secondary address component in a range of secondary addresses.

Examples:

<table><tr><th>Address Range</th><th>Address Secondary High Number</th></tr><tr><td>00000001 through 00000009</td><td>0000000A</td></tr><tr><td>A through F</td><td>A</td></tr><tr><td>66A through 66E</td><td>66A</td></tr></table></td></tr>
<tr><td>SecondaryUnit</td><td>A descriptive code used to identify the type of address secondary range information in the Address Secondary Range field. This code may be useful in address matching, e.g., the secondary address numbers may indicate apartment, suite, or trailer numbers. (See Publication 28, Postal Addressing Standards, for values.)</td></tr>
</table>

Column Name	Description
StreetName	The official name of a street as assigned by a local governing authority. The Street Name field contains only the street name and does not include directionals (EAST, WEST, etc.) or suffixes (ST, DR, BLVD, etc.). This element may also contain literals, such as PO BOX, GENERAL DELIVERY, USS, PSC, or UNIT. Numeric street names that have numeric components that are four characters (or less) in length at are justified such that the low-order digit of the number is positioned in the fourth position of the street name field. This shift is made so that the numeric street names appear in numeric sequence.
Suffix	Code that is the standard USPS® abbreviation for the trailing designator in a street address. (See Publication 28, Postal Addressing Standards, for values.)
UpdateKeyNumber	Field that contains a number that uniquely identifies a record; used to identify the base record to which an add or delete transaction is being directed. The Update Key Number field is used only when applying transactions to the base file; it is not used in address matching and remains fixed for the life of the record. The field is alphanumeric and consists of the database segment code (V1, V2, W1, W2, X1, X2, Y1, Y2, Z1, or Z2) and eight characters containing an alphanumeric value ranging from 00000001 to AAAAAAAA.
Urbanization	An index to the City State file that provides the urbanization name for this delivery range.

Column Name	Description
ZipCode	A 5-digit code that identifies a specific geographic delivery area. ZIP Codes can represent an area within a state, or a single building or company that has a very high mail volume.

Response Status Table

Response Code	Definition
10	The address was sent to the verification database was invalid (i.e. contained a dual address)
11	The address supplied had an invalid zip code
12	The address supplied had an invalid state abbreviation code
13	The address supplied had an invalid city name

Task Factory Advanced Aggregator Transform

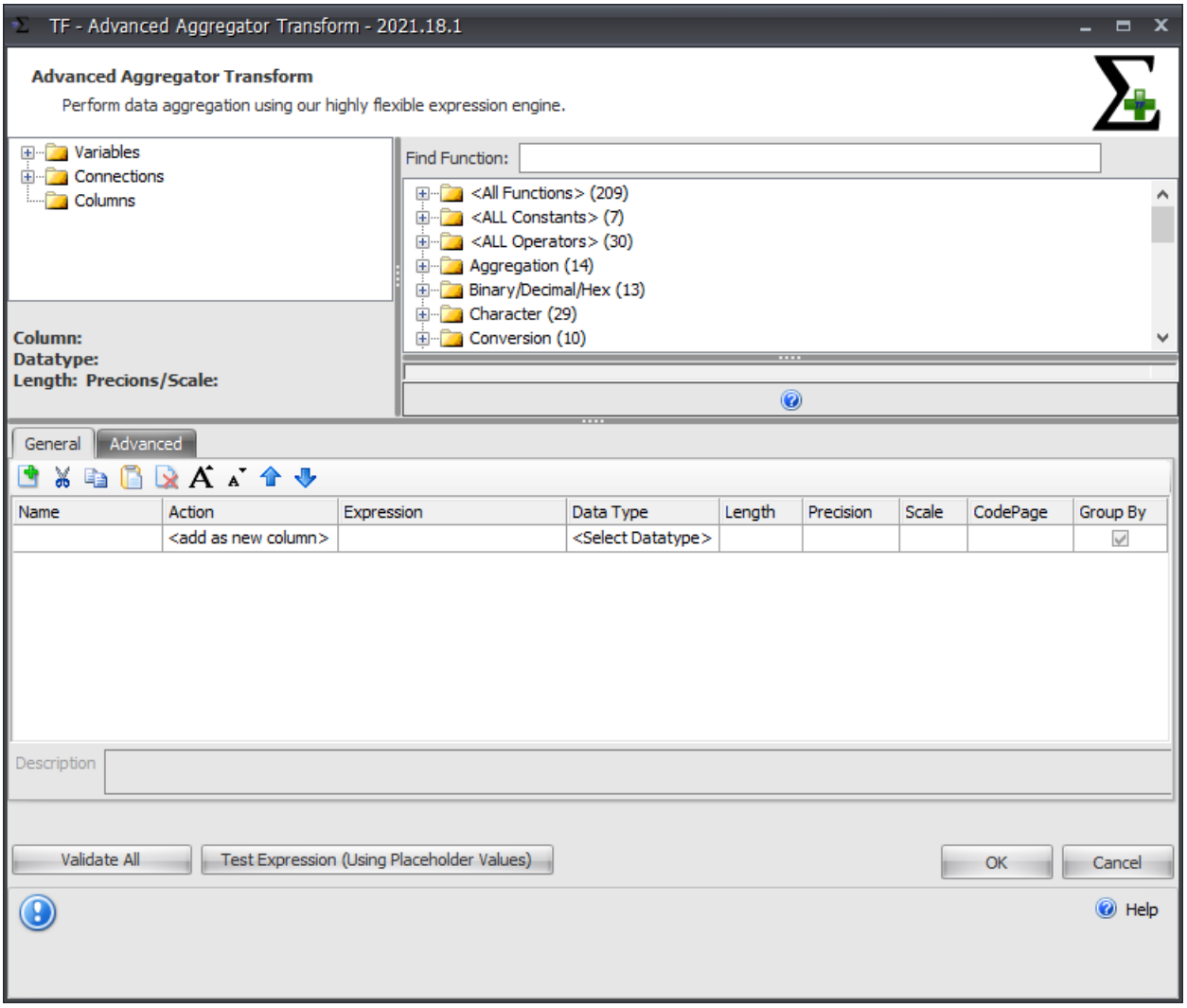
Last Modified on 11 August 2022

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

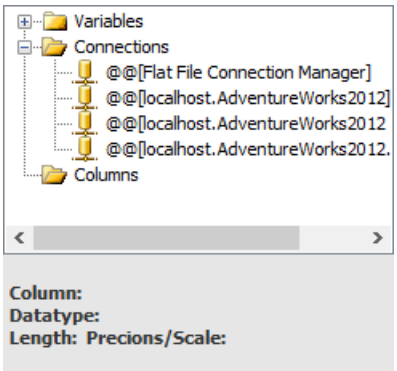
Advanced Aggregator Transform

Transform Icon	Transform Description
	The **Advanced Aggregator Transform** is used to aggregate rows of data from an upstream based on grouping.

General Tab

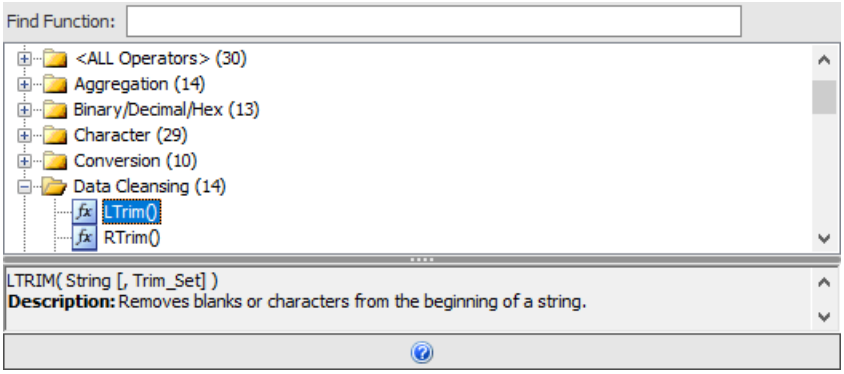


Column Browser



Located on the left hand side, the **Column Browser** window shows you all the available columns, variables and connections that can be used as part of a derived column expression. Selecting each plus sign next to a group in the tree view expands that group of items. Any item from the column browser can be dragged into an expression in the column grid below.

Functions Browser

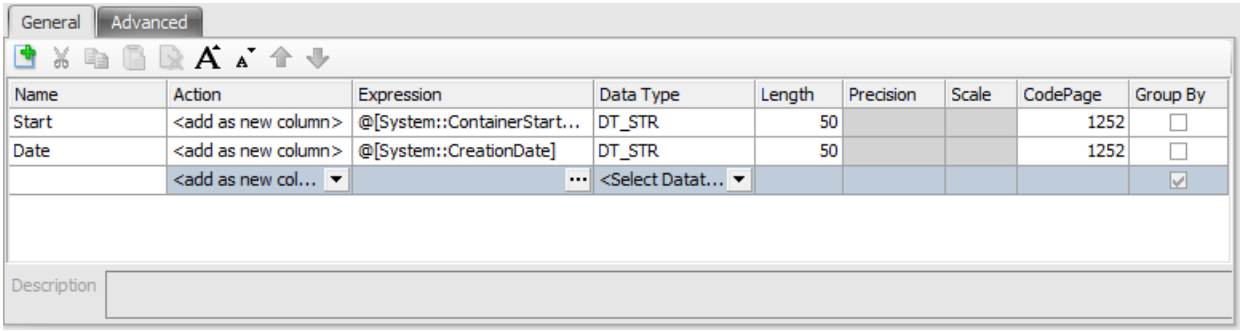


Located on the right hand side, the functions browser window lists all of the available functions that can be used in the **Advanced Derived** column (as well as any of the expressions based tasks). All functions are grouped into an appropriate group (e.g. Character, Data Cleansing) based on what the function either does or returns. For instance, the character function Left is listed under the **Character** group because it performs operations on a set of characters or a string. Selecting a function gives you a description, syntax overview, and examples on how to use the function.

Available Functions List

For a list of all available functions, along with syntax, descriptions, and examples, see the Functions List articles.

Columns Grid

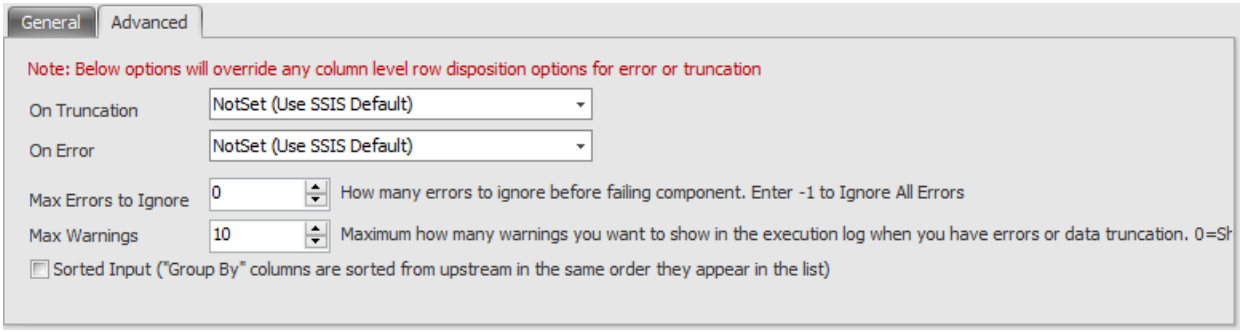


The **Columns Grid** (the bottom window) allows you to create expressions used for aggregation and variable expressions.

Column	Description
Name	The name of the column or expression.
Action	The action this column performs. • **Variable Expression** - This option creates a reusable expression that can be used in the expression column of the current column or other columns in the grid. • **Add As New Column** - This creates a new derived column based on the expression.
Expression	This is the expression used to perform the action from the action column.

Column	Description
Data Type	The data type for the current column.
Length	The length of the current column.
Precision	The precision of the current column.
Scale	The scale of the current column.
CodePage	The code page of the current column.
Group By	Determines whether the expression is used in the group by calculation.

Advanced Tab



Option	Description
On Truncation	Action to take when a truncation occurs.
On Error	Action to take when an error occurs.
Max Errors to Ignore	How any errors to ignore before failing component.
Max Warnings	Maximum number of warnings you want to show in the execution log when you have errors or data truncation.

Validate All Button

The **Validate All** button evaluates all of the expressions and checks if they have errors. An error or warning icon appears if there are any issues with the expressions.

Test Expression (Using Placeholder Values) Button

This button creates temporary values based on the data type and execute the expression that is currently selected in the grid.

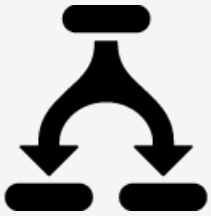
Note: To test the expression by entering your own values, use the expression editor.

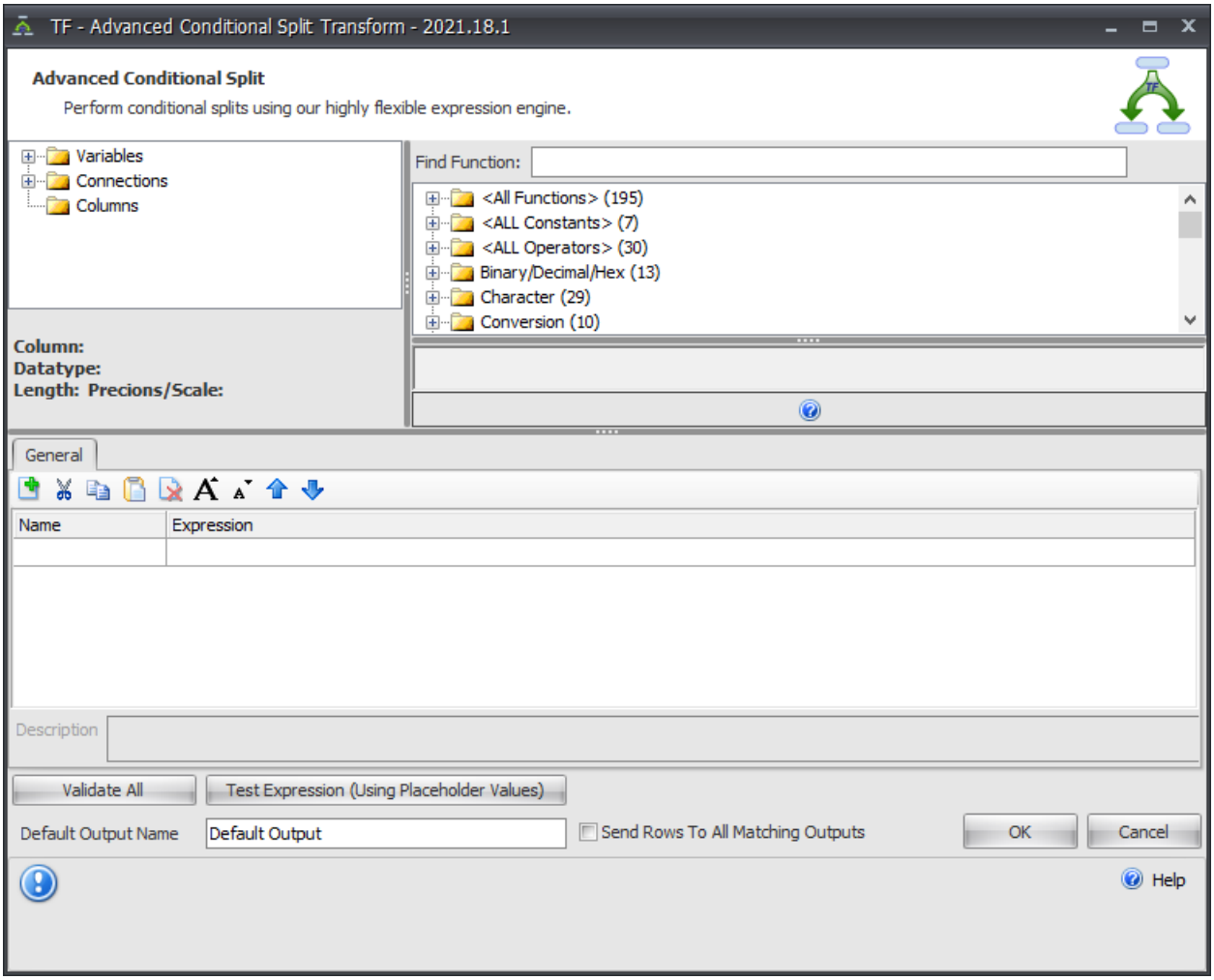
Task Factory Advanced Conditional Split Transform

Last Modified on 11 August 2022

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Advanced Conditional Split Transform

Transform Icon	Transform Description
	The **Advanced Conditional Split Transform** is used to conditionally route rows to different outputs depending on the expression defined by using columns, variables, and expressions.



Option	Description
Column Browser	Located on the left hand side, the column browser window shows you all the available columns, variables and connections that can be used as part of a derived column expression. Selecting a plus sign next to each group in the tree view expands that group of items. Any item from the column browser can be dragged into an expression in the column grid below.
Functions Browser	Located on the right hand side, the functions browser window lists all of the available functions that can be used in the advanced derived column (as well as any of the expressions based tasks). All functions are grouped into an appropriate group (e.g. Character, Data Cleansing) based on what the function either does or returns. For instance, the character function **Left** is listed under the **Character** group because it performs operations on a set of characters or a string. Selecting a function gives you a description, syntax overview, and examples on how to use the function.
Available Functions List	For a list of all available functions, along with syntax, descriptions, and examples, see the Functions List articles.
Output Grid	The bottom window which is the output grid allows you to define conditional split expressions to route rows matching the expression for that output.

Option	Description
General Tab	• **Name** - The name of the column or expression. • **Expression** - This is the expression used to conditionally split the rows of data.
Validate All Button	The **Validate All** button evaluates all of the expressions and check if they have errors. An error or warning icon displays if there are any issues with the expressions.
Test Expression (Using Placeholder Values) Button	This button creates temporary values based on the data type and executes the expression that is currently selected in the grid. To test the expression by entering your own values use the expression editor.
Default Output Name	This is the name of the output where any rows that do not match any of the expressions are routed.

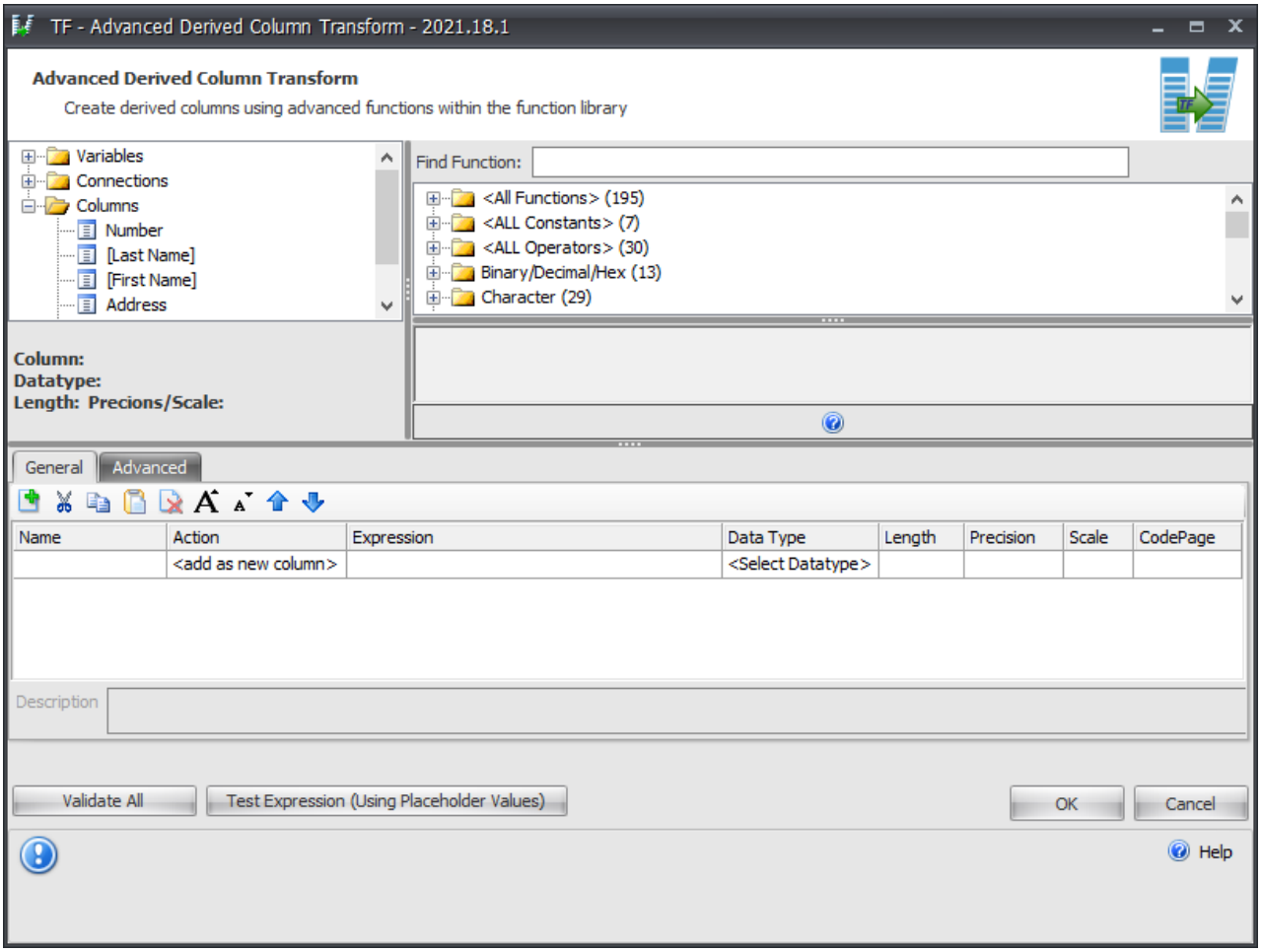
Task Factory Advanced Derived Column Transform

Last Modified on 15 August 2022

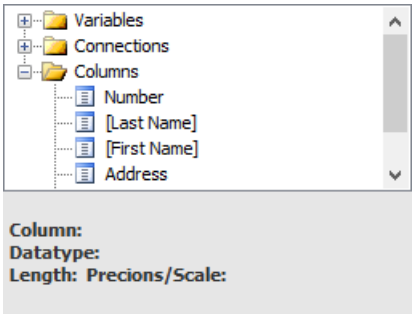
📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Advanced Derived Column Transform

Transform Icon	Transform Description
	The **Advanced Derive Column Transform** is used to create columns in an SSIS data flow using columns, variables, and expressions.

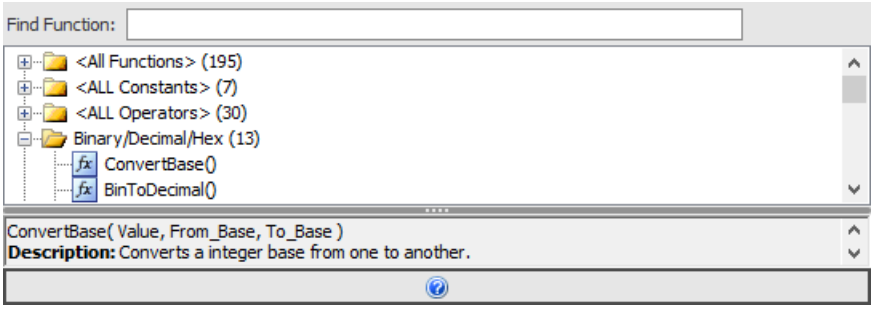


Column Browser



Located on the left hand side, the column browser window shows you all the available columns, variables and connections that can be used as part of a derived column expression. Selecting a plus sign next to a group in the tree view expands that group of items. Any item from the column browser can be dragged into an expression in the column grid below.

Functions Browser



Located on the right hand side, the functions browser window lists all of the available functions that can be used in the advanced derived column (as well as any of the expressions based tasks). All functions are grouped into an appropriate group (e.g. Character, Data Cleansing) based on what the function either does or returns. For instance, the character function Left is listed under the **Character** group because it performs operations on a set of characters or a string. Selecting a function gives you a description, syntax overview, and examples on how to use the function.

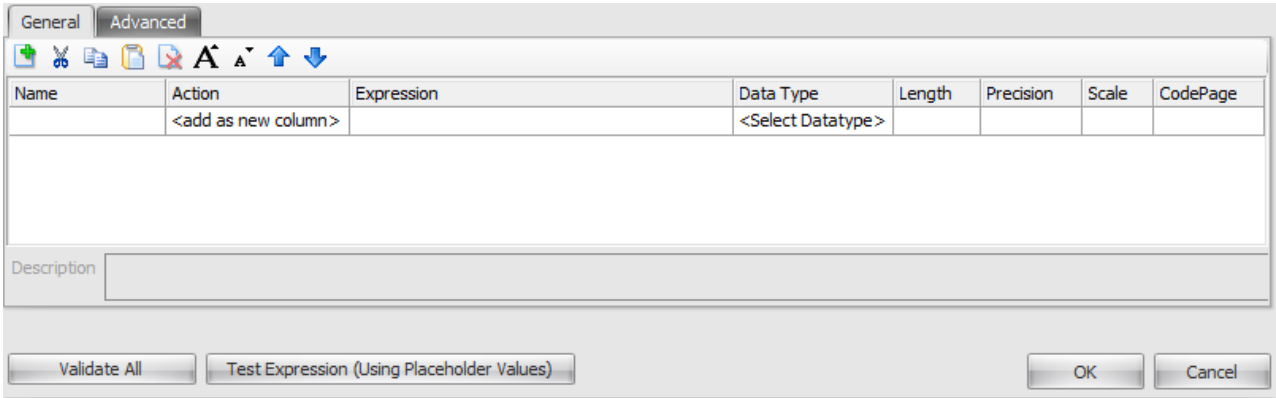
Available Functions List

For a list of all available functions, along with syntax, descriptions, and examples, see the Functions List page.

Columns Grid

The **Columns Grid** (the bottom window) allows you to define the new derived column, replace columns, and setup variable expression columns that can be used in other column expressions.

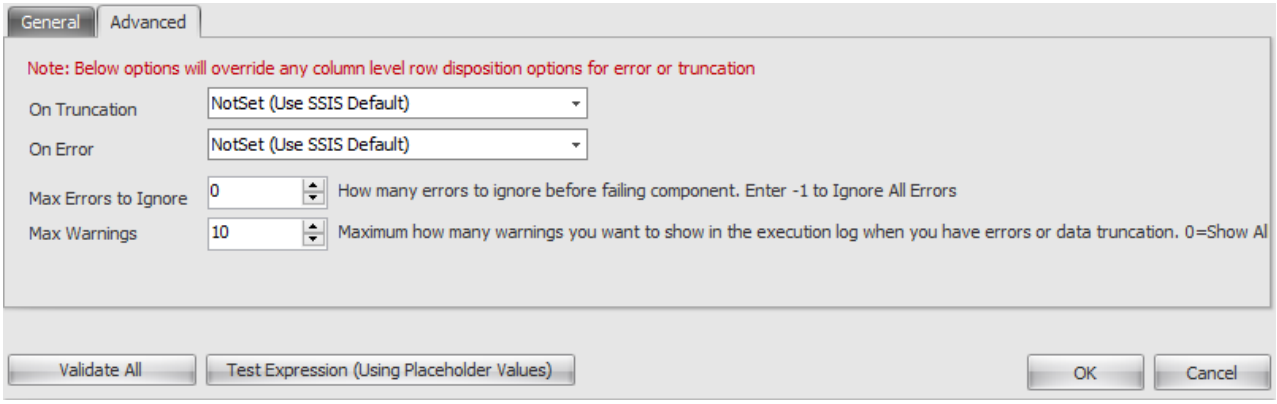
General Tab



Column	Description
Name	The name of the column or expression.
Action	The action this column performs. • **Variable Expression** - This option creates a reusable expression that can be used in the expression column of the current column or other columns in the grid. • **Add As New Column** - This creates a new derived column based on the expression. • **Replace Column** - This replaces an input column's data with the result of the expression.

Column	Description
Expression	This is the expression used to perform the action from the action column.
Data Type	The data type for the current column.
Length	The length of the current column.
Precision	The precision of the current column.
Scale	The scale of the current column.
CodePage	The code page of the current column.

Advanced Tab



Option	Description
On Truncation	Action to take when a truncation occurs.
On Error	Action to take when an error occurs.
Max Errors to Ignore	How any errors to ignore before failing component.
Max Warnings	Maximum number of warnings you want to show in the execution log when you have errors or data truncation.

Validate All Button

The **Validate All** button evaluates all of the expressions and checks if they have errors. An error or warning icon displays if there are any issues with the expressions.

Test Expression (Using Placeholder Values) Button

This button creates temporary values based on the data type and execute the expression that is currently

selected in the grid.

> **Note:** To test the expression by entering your own values use the expression editor.

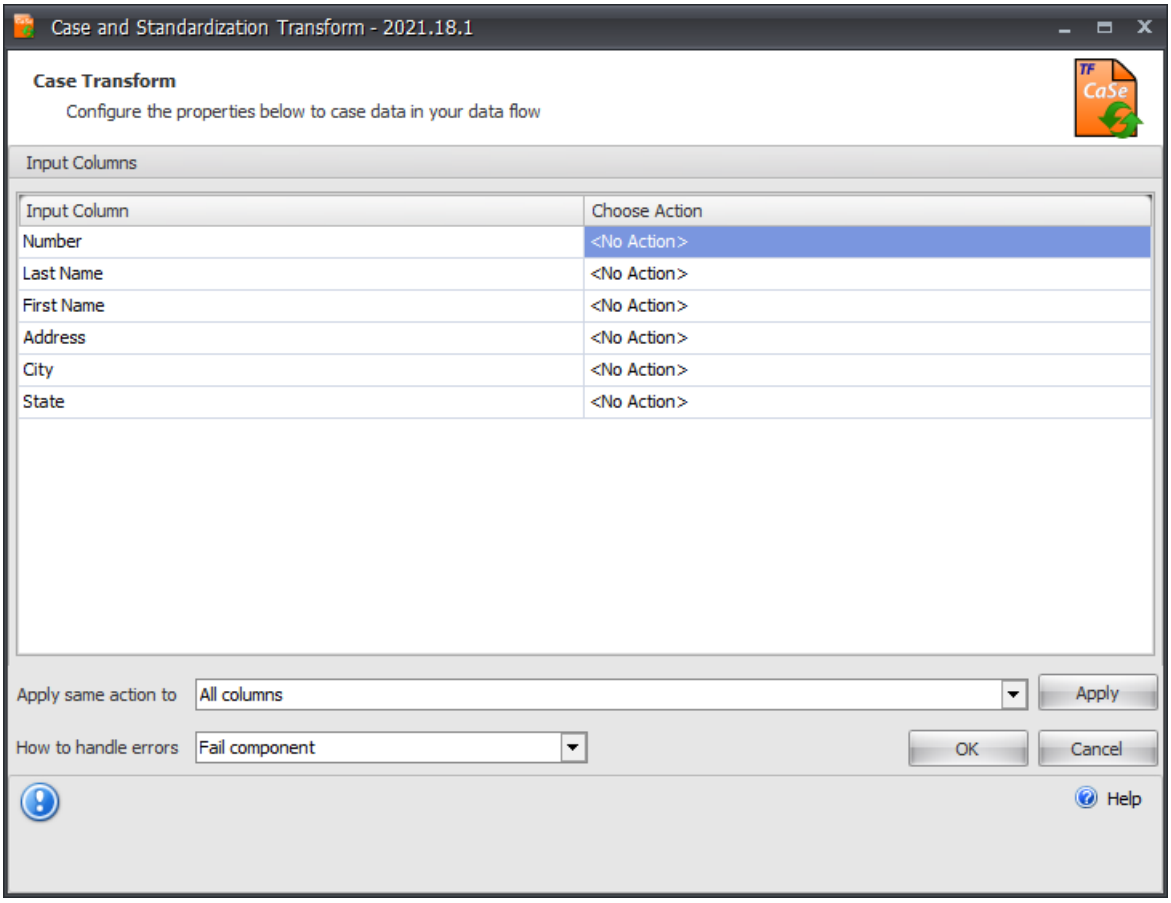
Task Factory Case Transform

Last Modified on 15 August 2022

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

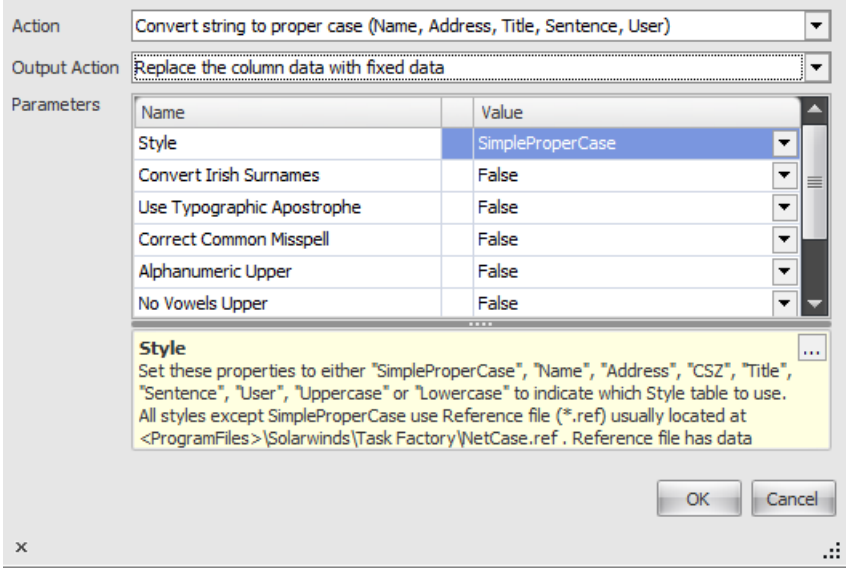
Case Transform

Transform Icon	Transform Description
	The **Case and Standardization Transform** is a transform that allows the user to format the case of source data using different styles of formatting.



Actions

In the **Choose Action** column, selecting on the tag displays the choose action drop down menu.



Action

> **Note:** By default no action is selected. Selecting an action opens the parameter options.

Option	Description
Convert string to proper case	This action is used to transform the case of source data.

Output Action

The **Case Transform** has the two output options:

Option	Description
Replace the column data with fixed data	
Output the fixed data to a new output column	If selected, the new output column name is i**nputcolumn_output**. For example, if the source column name is **FirstName**, the output column is **FirstName_output**.

Parameters

> ℹ **Note:** Parameters become available once an action is selected.

Option	Description
Style	Select the style table to use, when **SimpleProperCase** is selected then all other parameters are ignored.
Convert Irish Surnames	Set true or false to indicate whether or not to convert Irish surnames. ex. OHARA converts to O'Hara.
Use Typographic Apostrophe	Set this property true or false to indicate whether or not to use the typographic apostrophe in place of the standard apostrophe when converting Irish surnames.
Correct Common Misspell	Set this property true or false to indicate whether or not to correct common misspellings. ex. acn is corrected to can.
Alphanumeric Upper	Set this property true or false to indicate whether or not to convert alphanumeric data to all uppercase.
No Vowels Upper	Set this property true or false to indicate whether or not to convert alphabetic data containing no vowels to all uppercase. ex. mlk converts to MLK.

Option	Description
Ignore Mixed Case	Set this property true or false to indicate whether or not to convert mixed-case text. Use this setting to preserve existing mixed-case text. Text only converts when **Text-In string** is all uppercase or lowercase.

Additional Information: See the Task Factory Error Row Handling article for more information about this functionality.

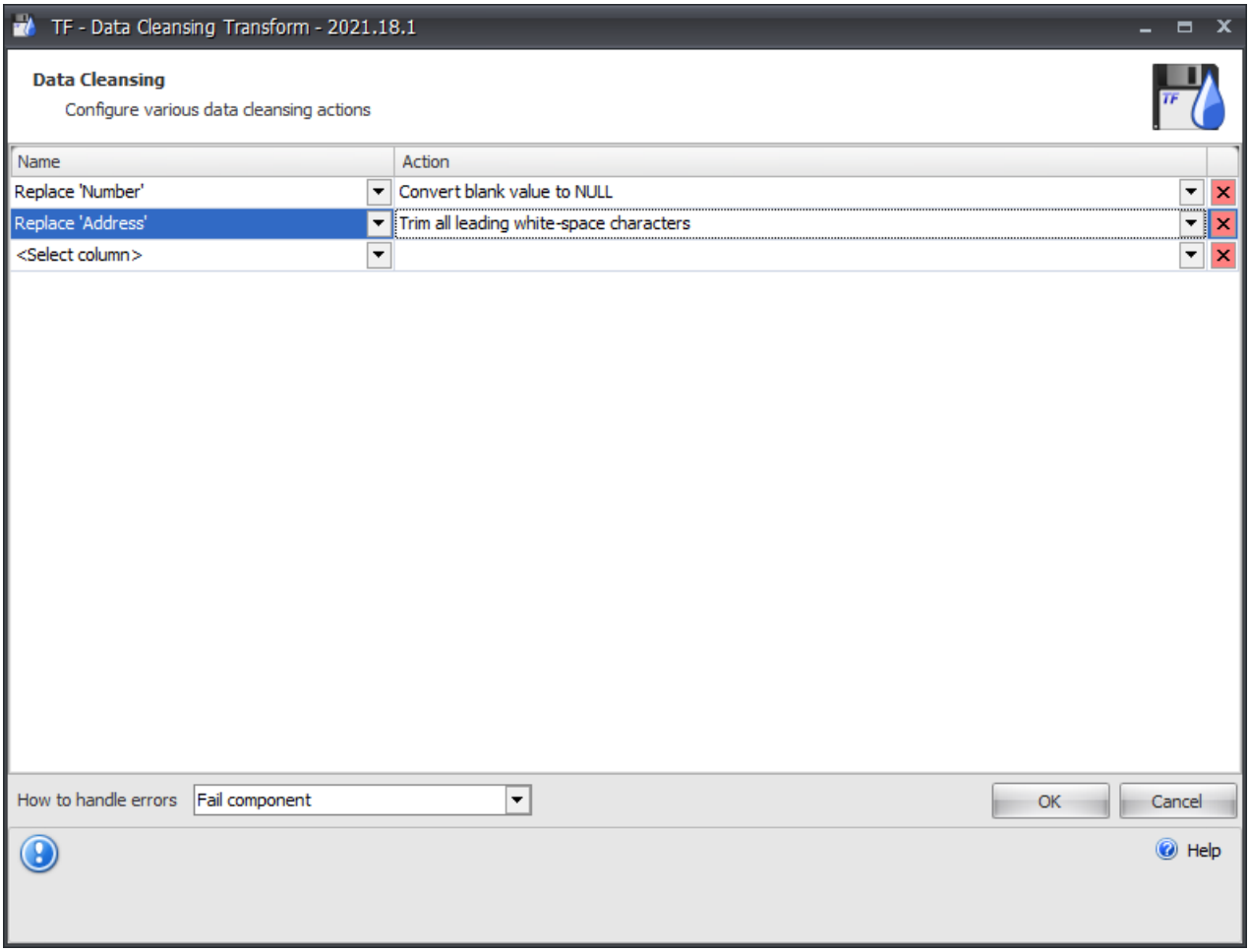
Task Factory Data Cleansing Transform

Last Modified on 15 August 2022

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Data Cleansing Transform

Transform Icon	Transform Description
	The **Data Cleansing Transform** is used to clean or alter data coming into the transform with rules defined in the UI.



UI Overview

Name

Select the name of the column to replace.

Action

Select the action to be taken on the data.

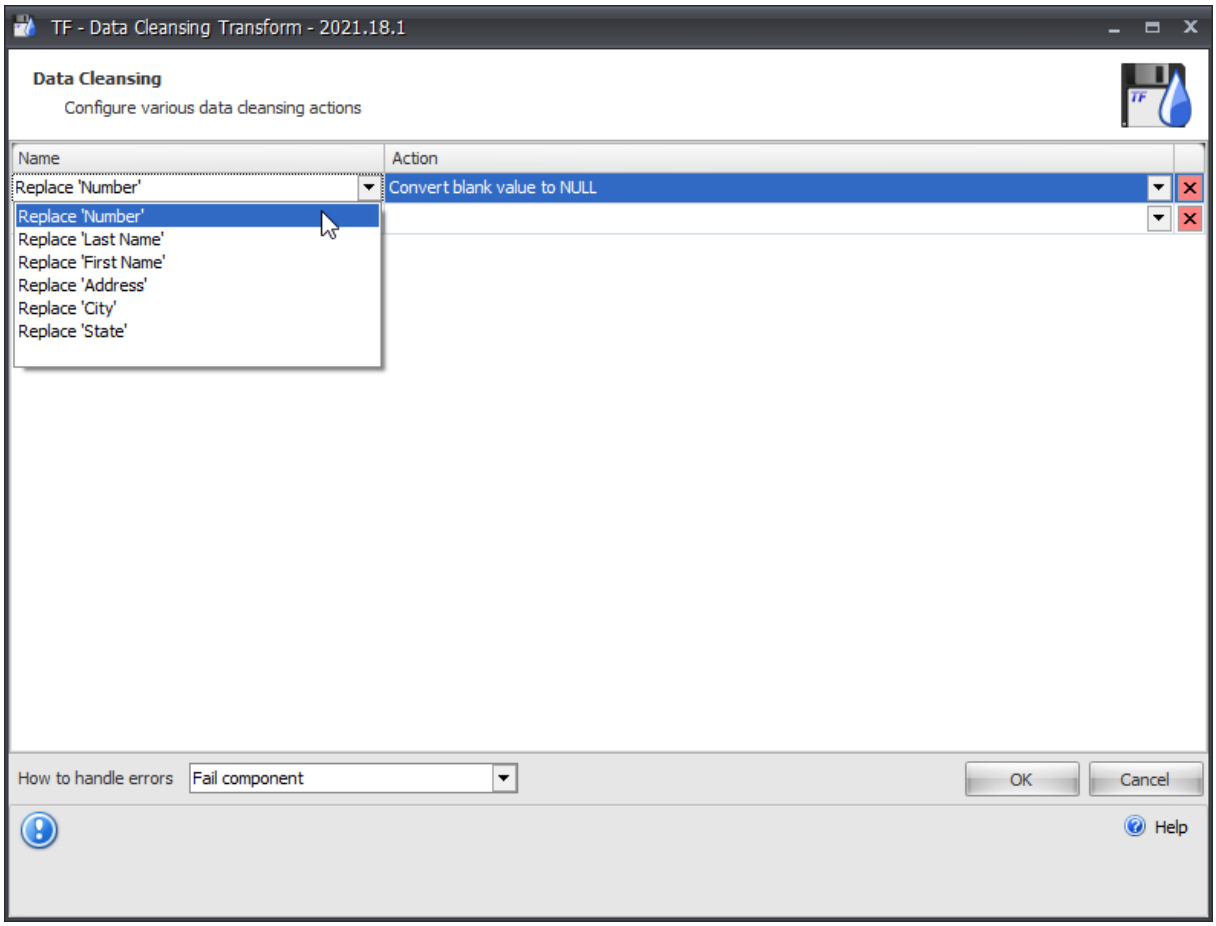
Action	Description
Convert string to proper case (Name, Address, Title, Sentence, User)	This allows you to replace the values of an input column with a properly cased output.

Action	Description
Trim Whitespace	• **Trim all leading and trailing whitespace characters** - Trims all whitespace from the beginning and end of the selected input column data. • **Trim all leading whitespace characters** - Trims all the whitespace from the beginning of the selected input column data. • **Trim all trailing whitespace characters** - Trims all the whitespace from the end of the selected input column data.
Trim specified characters	• **Trim specified characters from start and end** - Trims all the specified characters from the beginning and end of the selected input column data. • **Trim specified characters from start** - Trims all the specified characters from the beginning of the selected input column data. • **Trim specified characters from end** - Trims all the specified characters from the end of the selected input column data.
Change Date Format	Allows the conversion of a date from one format to another.
Convert NULL to user defined value	Allows the conversion of a NULL to a user defined value.
Convert blank value to NULL	Allows the conversion of a blank value to a NULL value.
Convert blank value to user defined value	Allows the conversion of a blank value to a user defined value.
Replace alpha/numeric/alphanumeric characters to user defined value	Replaces numeric characters with a user defined value.
Replace specified characters or words with user defined value	Replaces user specified words with a user defined value.
Replace bad date with user defined value	Replaces a bad date with a user defined value.
Replace matching regular expression pattern with user defined value	Replaces a regular expression pattern with a user defined value.
Extract data from input string using regular expression	Extracts data from an input string.
Replace invalid characters that cannot be part of an XML Document	Replaces invalid XML Document characters.

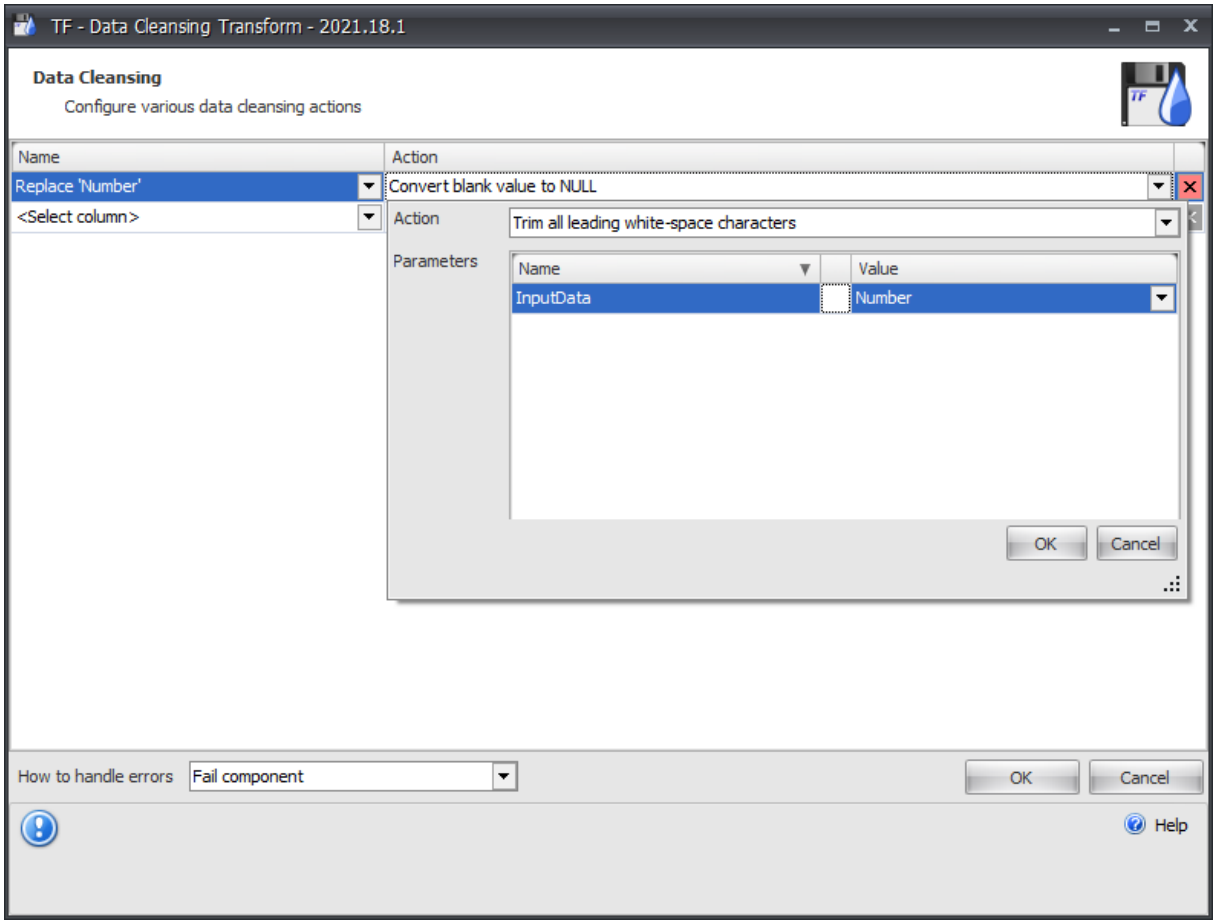
Action	Description
Replace non-printable characters	Replaces non-printable characters.

Defining a Data Cleansing Rule

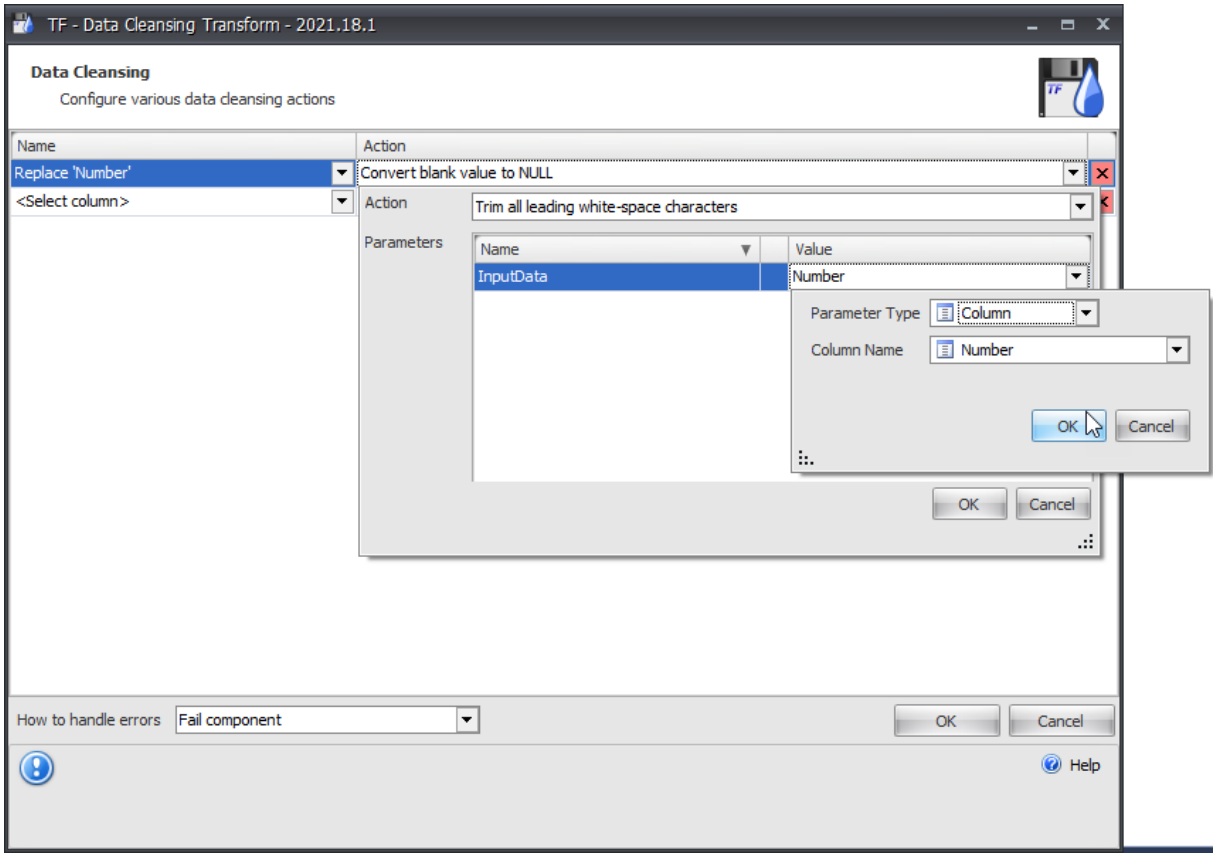
1. Choose a column to replace from the **Name** column of the editor window.



2. Next, select an action (rule) from the **Action** column of the editor window. This is the action taken on the data coming in.



3. Each parameter has a type associated with it. This allows you to choose how the transform should fill that parameter value during runtime execution.



4. Select **OK** to save the cleansing rule.

Additional Information: See the Task Factory Error Row Handling article for more information about this functionality.

Task Factory Data Flow Nugget Transform

Last Modified on 18 August 2022

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Data Flow Nugget Transform

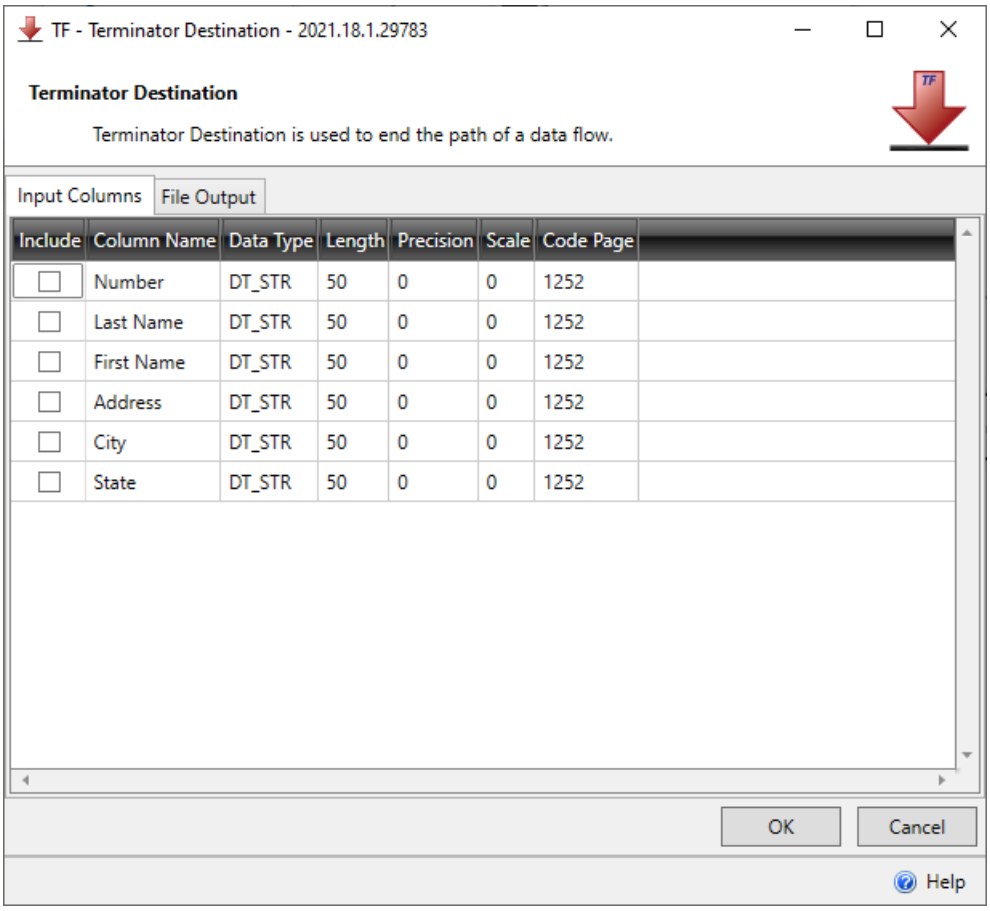
Transform Icon	Transform Description
	The **Data Flow Nugget Transform** allows you to configure a reusable data flow transform package.

Nugget Package Requirements

Your data flow nugget package must meet the following requirements to make it available for reuse in the **Data Flow Nugget Transform**:

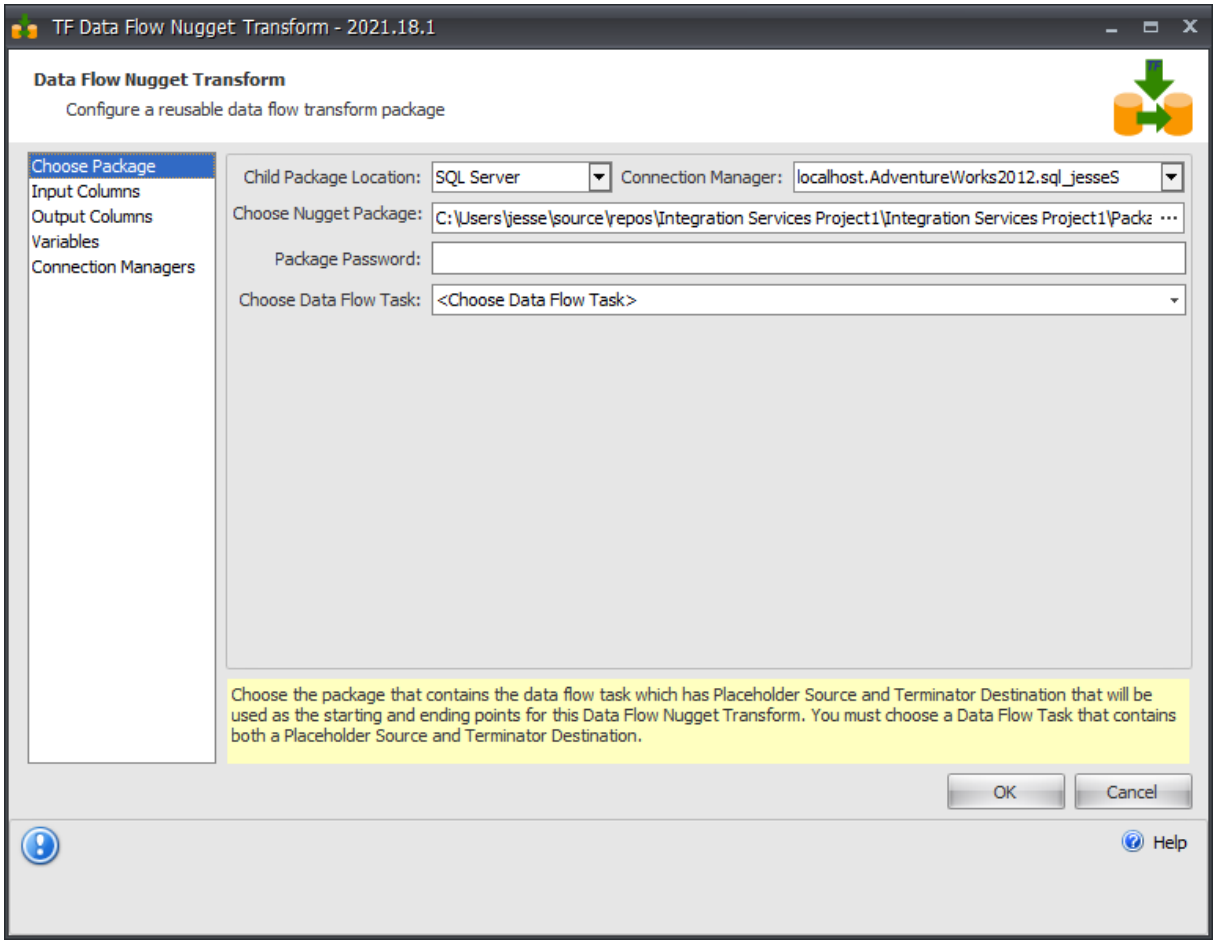
- Must have a data flow task - The **Data Flow Nugget Transform** reads a package looking for data flow tasks during setup. You must choose a package that contains a data flow task that follows the next two rules.
- Must have a placeholder source - The placeholder source is used to define the input columns for the data flow nugget.

- Must have a terminator destination - The terminator destination is used to define the columns that are returned from the data flow nugget package. Select the checkbox in the **include** column for the **Data Flow Nugget Transform** to detect that is an output column.

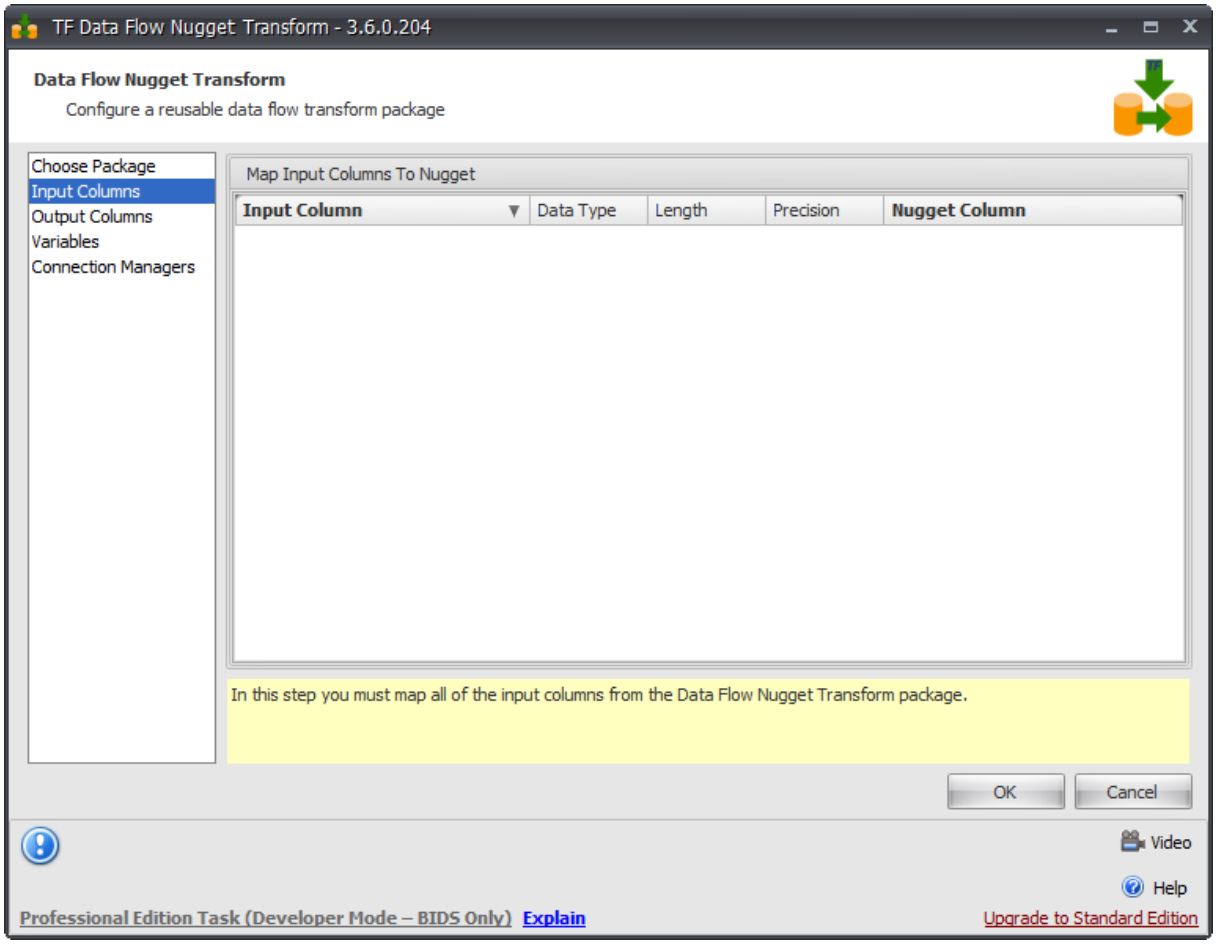


Setting Up a Data Flow Nugget

1. After you have setup a package to be used that meets the three requirements, choose the package in the transform editor's window. The child package can be located either on a file system, SQL server, or an SSIS Catalog. ℹ **Note:** If you choose SQL Server or SSIS Catalog, you must select a connection manager that connects to the server where the package is stored.



2. Next you map the input columns. Any of the columns that are defined in the placeholder source need to be mapped in the **Input Columns** step. The **Nugget Column** displays a drop down that contains all of the columns defined in the placeholder source. ℹ **Note:** If any columns defined in the placeholder source are not mapped in the **Data Flow Nugget Transform** you receive an error telling you to map the columns.



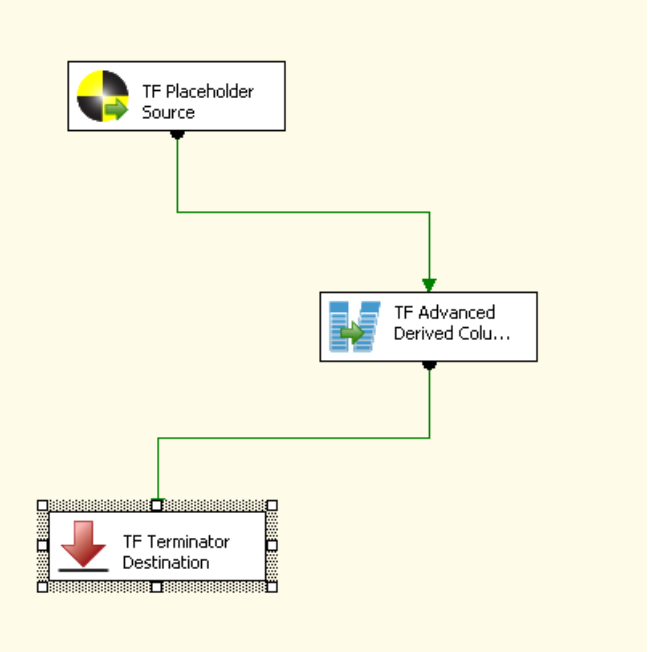
3. **Output columns** shows the columns defined in the Terminator Destination component of your data flow nugget package. What displays depends on whether you have an asynchronous component in your nugget package. With an asynchronous component, you see all of the columns marked as **Include** in the terminator destination in the **Output Columns from Nugget** grid.
4. On the **Variables** page, you are able to map the values of variables from the package that contains the **Data Flow Nugget Transform** to the variables that exist in the child nugget package.
5. The last step is to map the connection managers from the package that contains the **Data Flow Nugget Transform** to the connection managers that exist in the child nugget package.

Executing Data Flow Nugget Transforms

To execute a package that contains a **Data Flow Nugget** you need to use the **Advanced Execute Package Task**. The following shows the correlation between the three packages that are needed to execute **Data Flow Nuggets**.

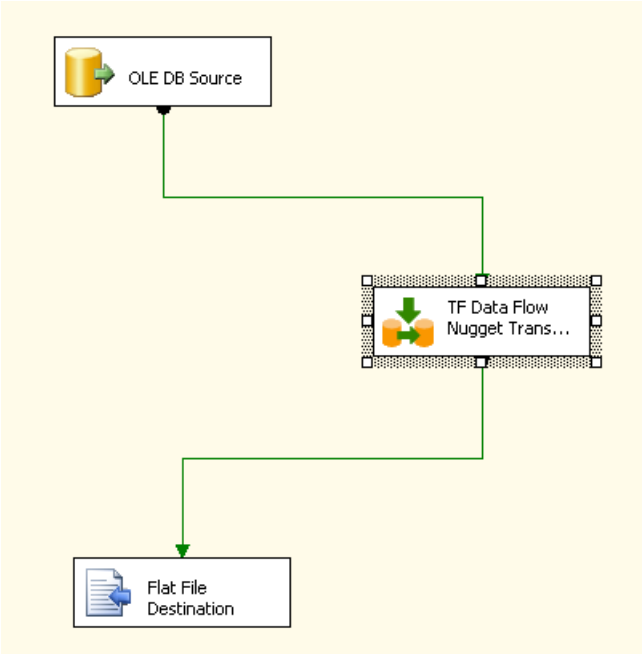
Package 1 - The Nugget Package

The nugget package is the reusable package that contains any of the reusable logic that executes within a **Data Flow Nugget Transform**.



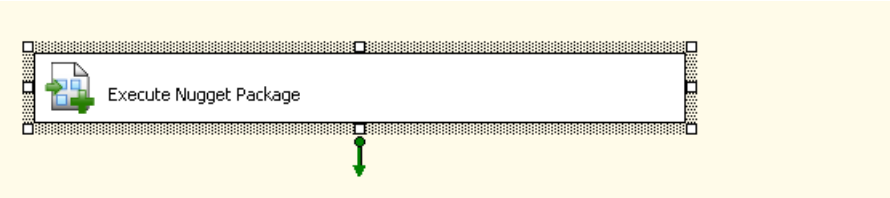
Package 2 - The Data Flow Nugget Transform Package

This package contains the **Data Flow Nugget Transform** that references Package 1.



Package 3 - The Advanced Execute Package Task Package

This package contains an **Advanced Execute Package Task** that references Package 2 as it's **Child Package**.



Task Factory Data Validation Transform

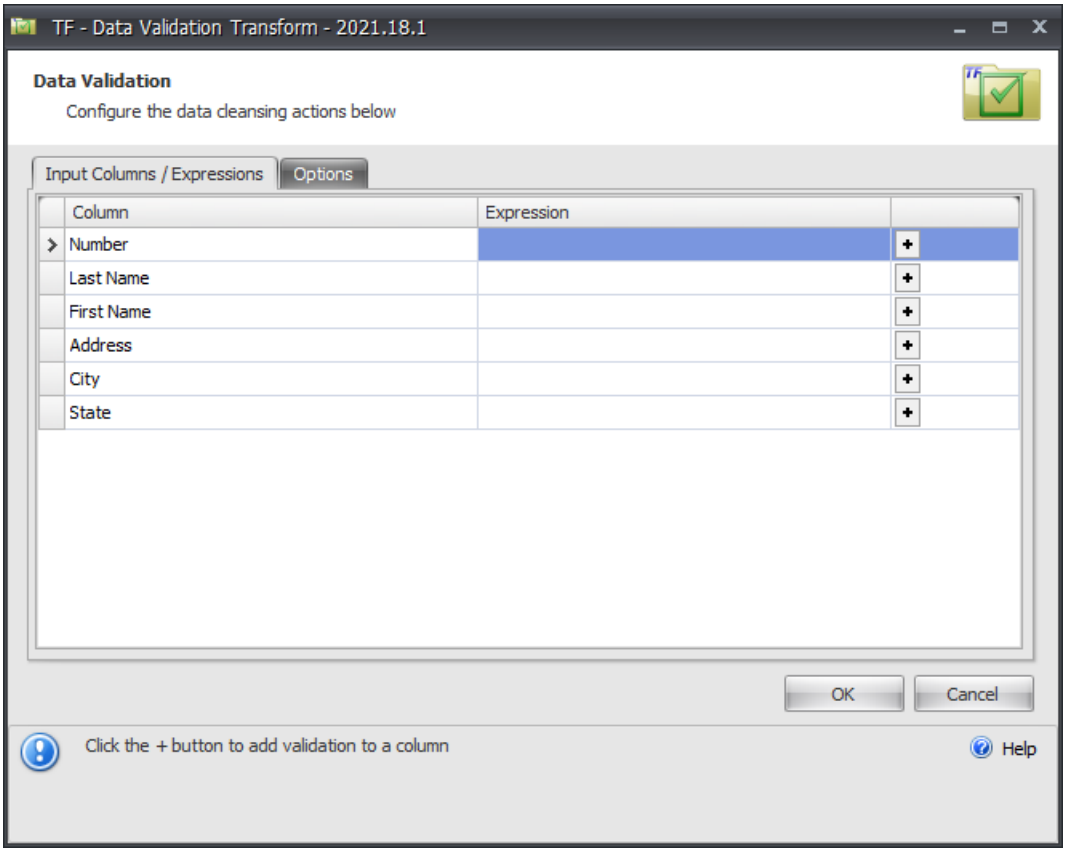
Last Modified on 18 August 2022

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Data Validation Transform

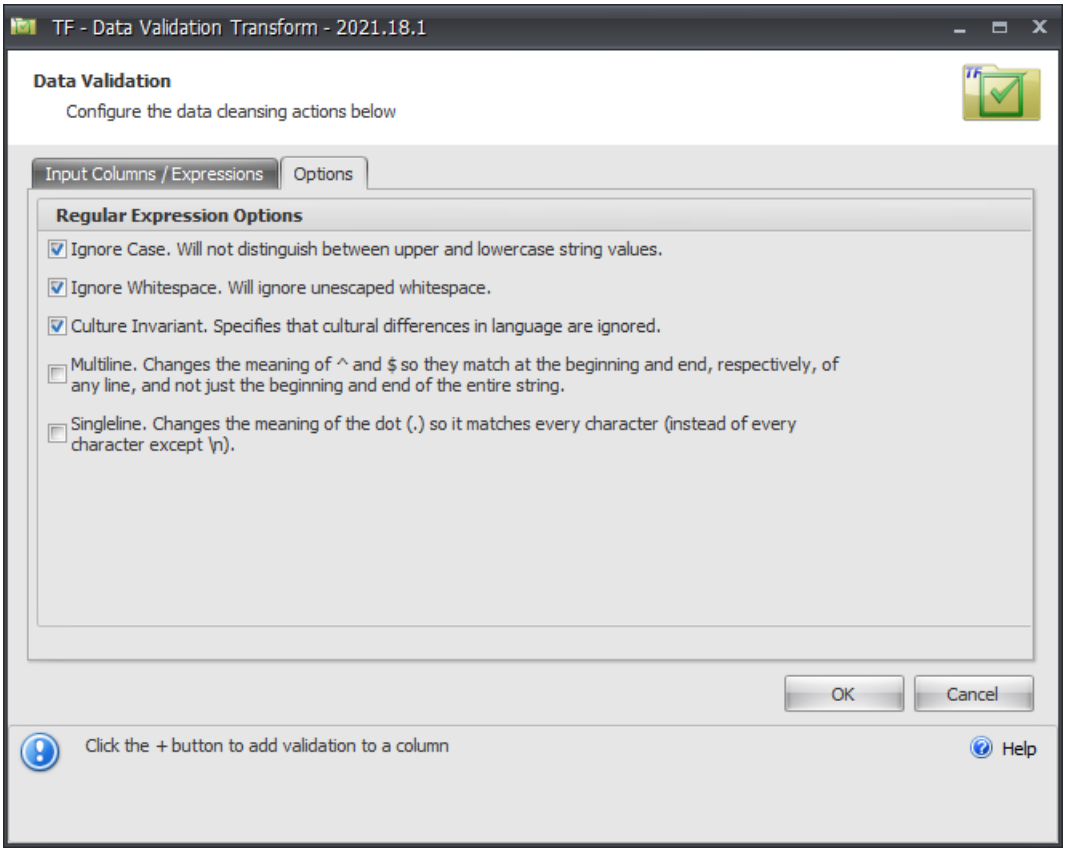
Transform Icon	Transform Description
	The **Data Validation Transform** validates data from an SSIS source by using regular expressions defined in the UI.

Options Tab



Option	Description
Column	Contains the column name of the input source column.
Expression	This contains the name of the expression used to validate the data in that source column if validation is added to it.
Buttons Column	This column contains the buttons used to **Add**, **Edit**, or **Delete** the expression. **Edit** and **Delete** are only available if validation has been added to the column.

Input Columns / Expressions



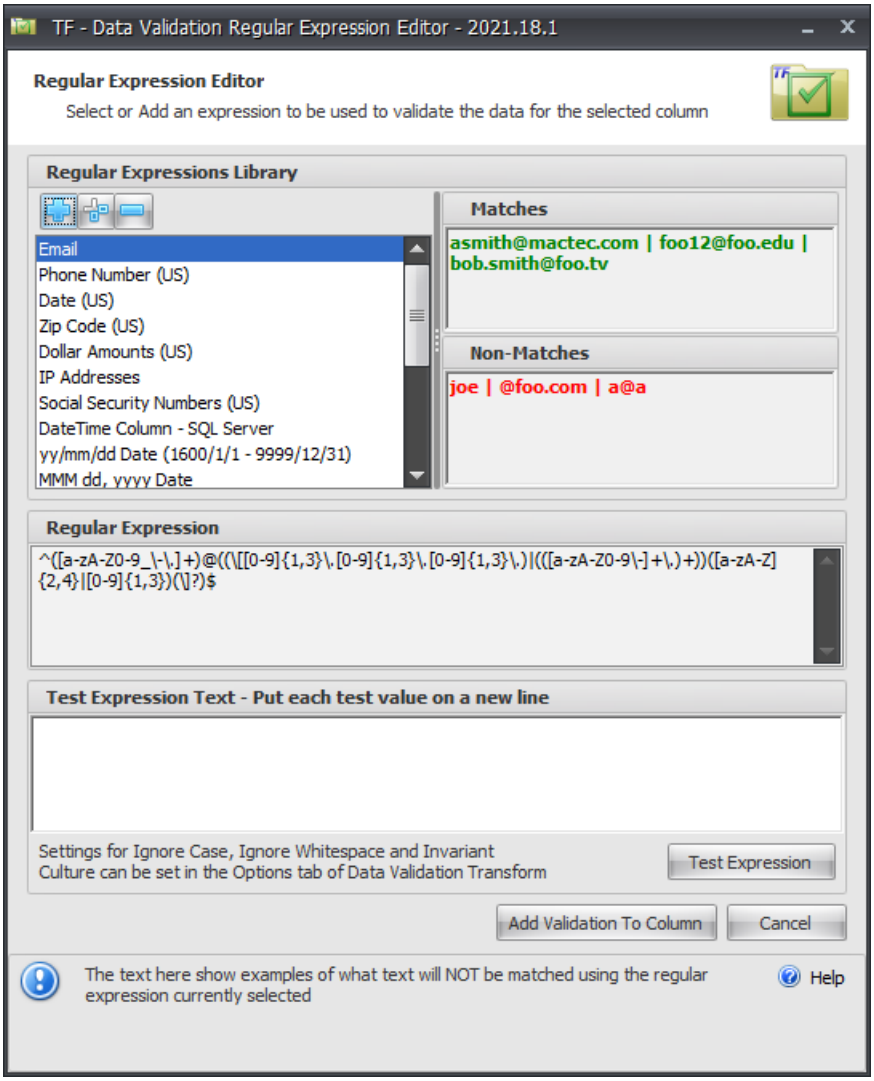
Regular Expression Options

Option	Description
Ignore Case	This prevents distinguishing between upper and lowercase string values.
Ignore Whitespace	This ignores unescaped whitespace.
Culture Invariant	Specifies that cultural differences in language are ignored.
Multiline	This changes the meaning of **^** and **$** so they match at the beginning and end of any line and not just the string.
Singleline	Changes the meaning of **. (period)** so it matches every character.

Adding / Editing / Deleting Validation

To add validation to a column, select the **+** button in the far right column of the **Input Columns / Expressions** tab. This opens the **Regular Expression** editor.

Regular Expression Editor UI Overview



The **Regular Expression Editor** is used to create, edit, and test regular expressions that can be used to validate your data.

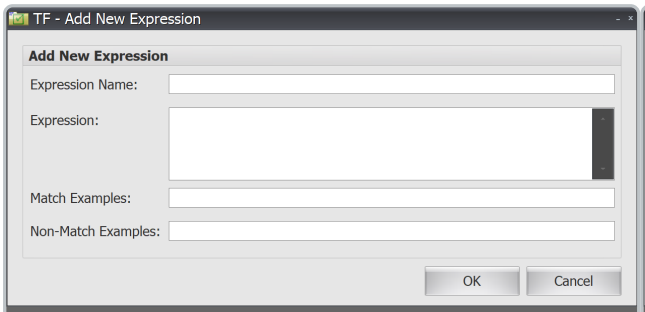
Option	Description
Expressions Library	• **Expressions List** - The expressions list contains all of the expressions that are part of your library. • **Matches / Non-Matches** - The text boxes show an example of what will or won't match the expression.
Regular Expression Textbox	This textbox shows the currently selected expression from the expressions list.
Test Expression Text	Here you can test the currently selected expression by adding text. Enter each value you want to test on a new line, and then select **Test Expression** to display the results.

Regular Expression Library Toolbar

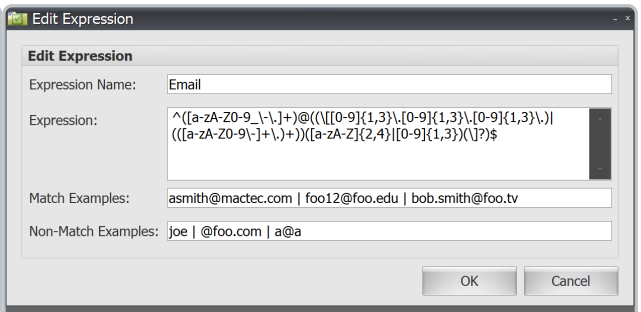
There are three options on the toolbar:

Toolbar Option	Description
	Add an expression.
	Edit an expression.
	Delete an expression.

Selecting **Add** or **Edit Expression** opens a similar window. Selecting **Add Expression** opens the **Add New Expression** window, which is blank, allowing you to input your own values. Selecting **Edit Expression** opens the **Edit Expression** window, which has the same fields, just populated with the current rules.



Add New Expression



Edit Expression

Task Factory Delete Batch Transform

Last Modified on 18 August 2022

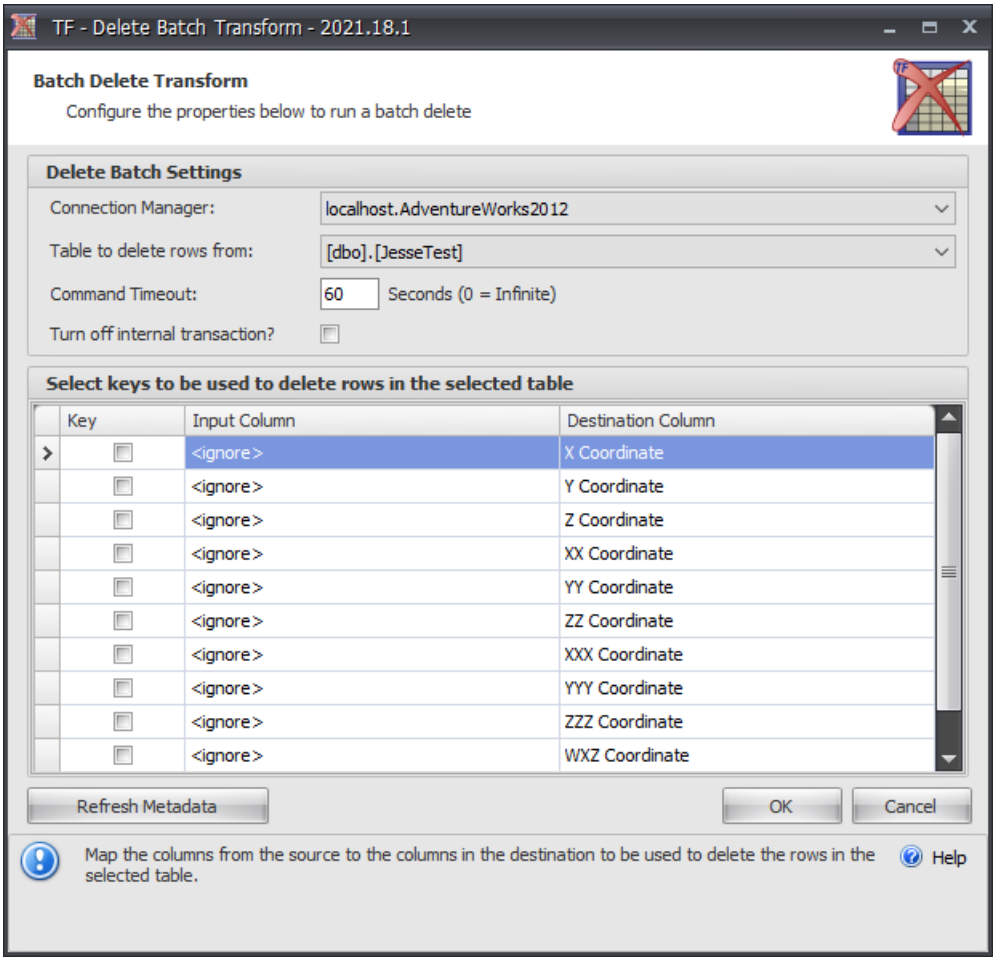
Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Additional Information: Please see the Azure Support page for more information before using this component with Azure databases.

Delete Batch Transform

Transform Icon	Transform Description
	The **Delete Batch Transform** allows you to quickly delete batches of data.

General Tab



Delete Batch Information

Option	Description
Connection Manager	Select an existing connection manager if available, or create a new one.
Table to delete rows from	Choose the table where you want to delete rows.
Command Timeout	Set the number of seconds before the command times out.
Turn off internal transaction?	Only use this option for performance reasons. It has the potential to send some data and not others if there is an error.

Select Keys to be used to delete rows in the selected table

Map the columns from the source to the columns in the destination to be used to delete the rows in the selected table.

Task Factory Dimension Merge SCD Transform

Last Modified on 16 August 2022

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Dimension Merge SCD Transform

Description

Note: Version 4.2.0.405 introduced the **memory optimized** property. This is a significant change, and we encourage you to read the memory optimized property section below before configuring the component.

Transform Icon	Transform Description
	The **Dimension Merge SCD Transform** is a custom data flow component used within SSIS to handle slowly changing dimension processing commonly used in data warehouses.

While SSDT/BIDS provides a standard SCD component, you may experience the following limitations:

- Slow performance
- Re-running the SCD Wizard can destroy parts of the data flow
- The component can't be used against all data providers
- Difficulty in troubleshooting the standard component's decision making abilities

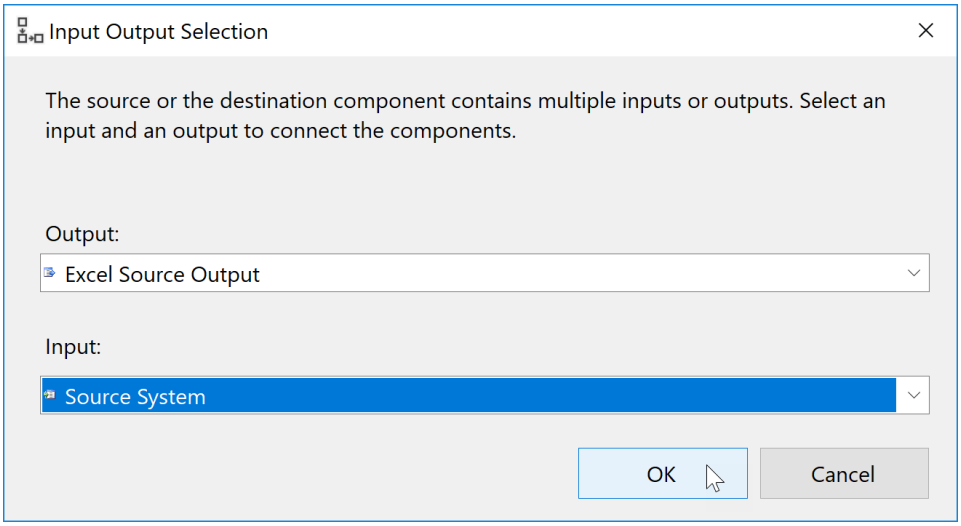
Task Factory's Dimension Merge SCD Transform maintains the functionality of the original SCD component while also providing:

- Better performance
- Non-destructive editing capabilities
- The ability to use a variety of data providers
- Improved UI that empowers you to customize the component's configuration and view/troubleshoot decision-making
- Input/Output auditing
- Improved logging capabilities

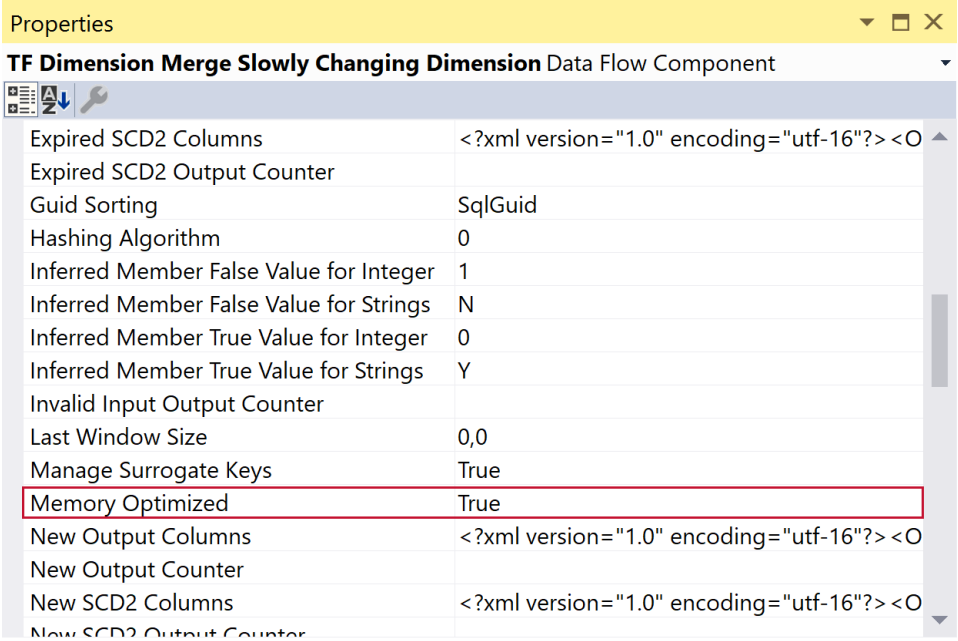
Setup

When using the Dimension Merge SCD Transform, you begin by connecting two of the following:

Connector	Description
Existing Dimension	This input provides data from a data warehouse's dimension table and must include the **housekeeping** columns (SCD2, effective, and expiry). **Note:** Although optional, it's recommended that you sort this input on the business key.
Source System	This input provides data from the source OLTP system needed to compare and insert/update to the data warehouse's dimension table. Source system and existing dimension data types must match. **Note:** Although optional, it's recommended that you sort this input on the business key.
Special Members	(Optional) In typical dimensions, special or unknown members may be used to link fact table rows to NULL or unknown dimension values. The special members input allows you to specify and maintain these special members in a data warehouse dimension table. (Special member rows should be treated as Type 1 changes to avoid duplicate versions of unknown values). Like the previous two inputs, it's recommended the Special Members input also be sorted on the business key.



Memory Optimized Property



Name
Specifies the name of the component.

As of version 4.2.0.402, the Dimension Merge Slowly Changing Dimensions component includes a new **Memory Optimized** property. This new property was introduced due to the retention of a large number of records that remained in memory in the original behavior. This caused memory usage of the package to grow significantly over the package execution time. By requiring the inputs to be sorted, this new model can process and remove rows from memory much earlier which reduces the overall memory footprint, and speeds up execution.

> **Note:** The following additional information about the Memory Optimized property:
>
> - When set to false, the component operates as it did in the past. Upgraded packages have this property set to **False** by default.

- New packages have this property set to **True** and have to meet certain restrictions (listed below). Otherwise you may encounter errors.
- If the inputs are not marked as sorted, the component uses the original behavior, even if **Memory Optimized** is set to **True**.
- If the property is set to **True**, all input rows must be sorted by the **Business Key** using a binary sort order. For example:

```
SELECT columns
FROM DimProductSCD
ORDER BY ProductAlternateKey
COLLATE Latin1_General_Bin2
```

> **Note:** The **ORDER BY** and **COLLATE Latin1_general_bin** clauses on this product dimension. This should be used when sorting within the source query and setting the Is Sorted/Sort Key properties in the Advanced Editor. When using a file as input, you need to add the SSIS sort component between the source and Dimension Merge component. This is a significant processing change, and it's recommended that you test your packages with both settings before deployment to ensure performance is optimized based on your dataset and configuration.

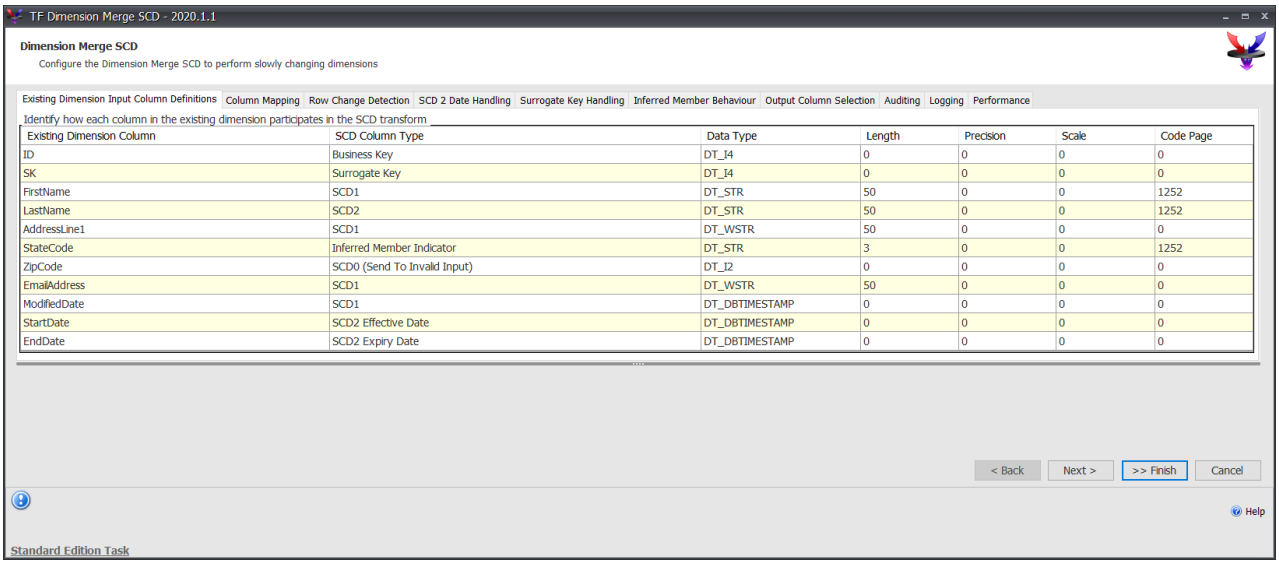
Existing Dimension Input Column Definitions Tab

From this tab, you can identify how each column in the existing dimension participates in the SCD transform. Columns can be defined as one of the following:

> **Note:** You may see an error at the bottom stating there is no Business Key selected, the Business Key can be selected from the **SCD Column Type** column.

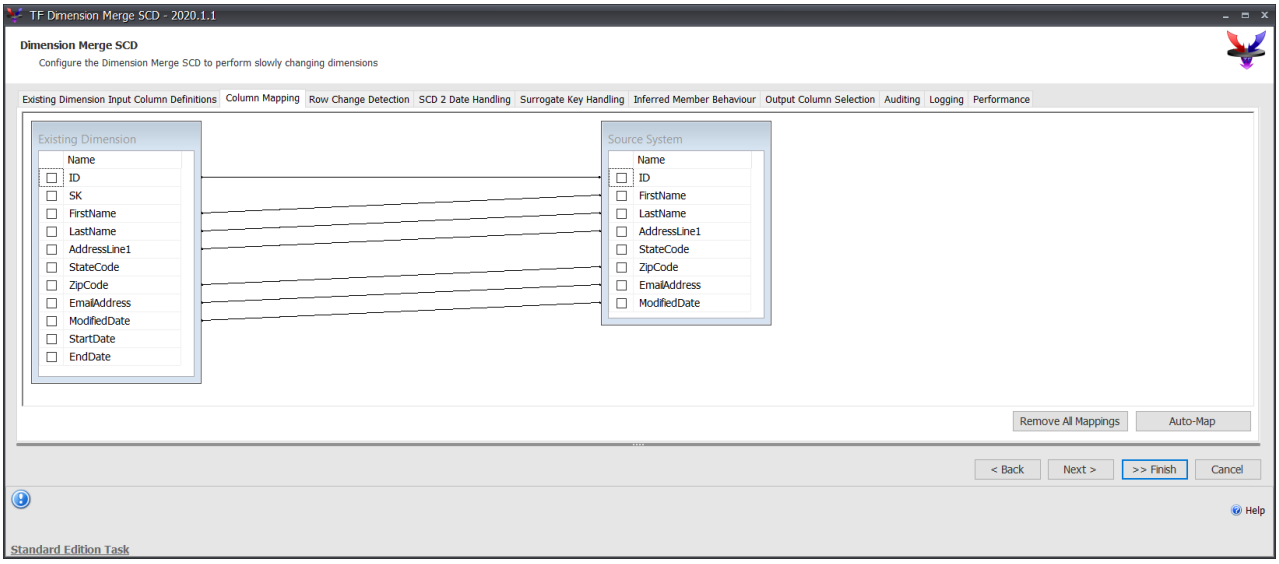
Column	Description
Not Used	Declares the column to not be used in the SCD processing.
Business Key	Identifies the business key column.
Surrogate Key	Identifies the surrogate key column.
SCD0 (Send To Invalid Input)	Identifies a column for Type 0 changes. Any change is sent to the Invalid Input's output.
SCD0 (Ignore Changes)	Identifies a column for Type 0 changes. Any change is ignored and sent to the Unchanged output.

Column	Description
SCD1	Identifies a column for Type 1 changes (dimension is overwritten with new values).
SCD2	Identifies a column for Type 2 changes (retains historical data).
SCD2 Current Record	A boolean column that can be used to identify the current record among related historical data.
Audit Column - Last Changed	A column that can be used to identify the moment a row was last changed.
Audit Column - Row Added	A column that can be used to identify a row that has been added.
Audit Column - SCD1 Update	A column that can be used to identify an SCD1 updated row.
Audit Column - SCD2 Change	A column that can be used to identify a row that has undergone an SCD2 change.
Inferred Member Indicator	A DT_WSTR column used to identify fact table references to a dimension that hasn't been loaded.



Column Mapping Tab

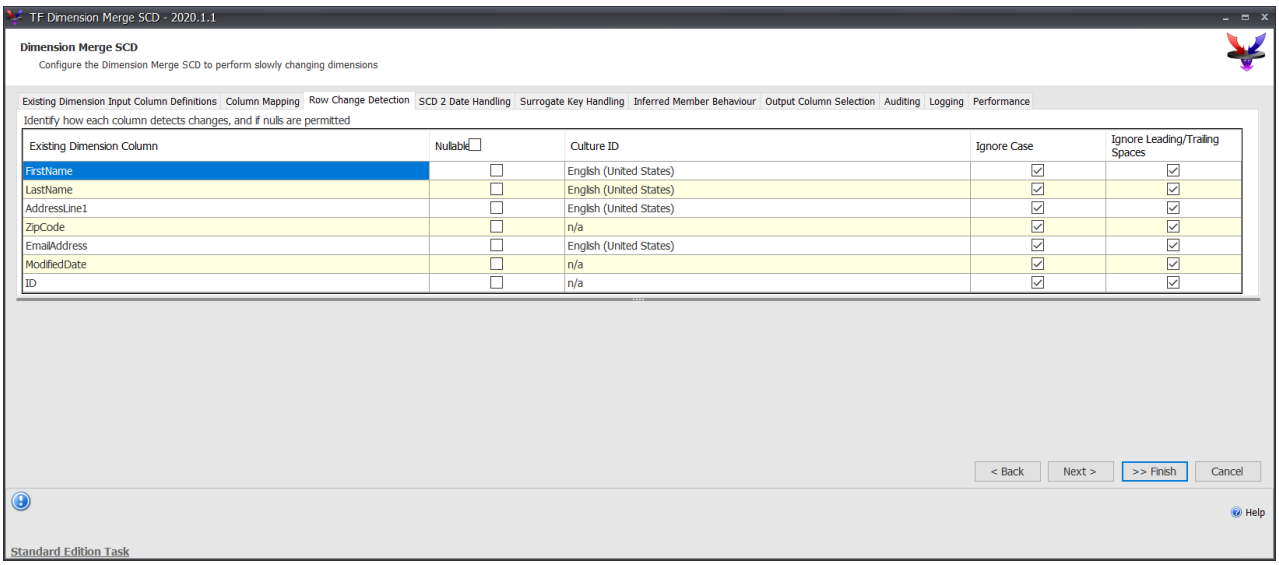
In the **Column Mapping** tab, you can specify the mapping between the existing dimension and source system.



Row Change Detection Tab

In the **Row Change Detection** tab, you can define how each column detects changes including case sensitivity, leading/trailing space, and columns with NULL values.

> **Note:** The Dimension Merge SCD redirects columns containing NULL values to the **Invalid Input** output unless that column is configured as **Nullable**.



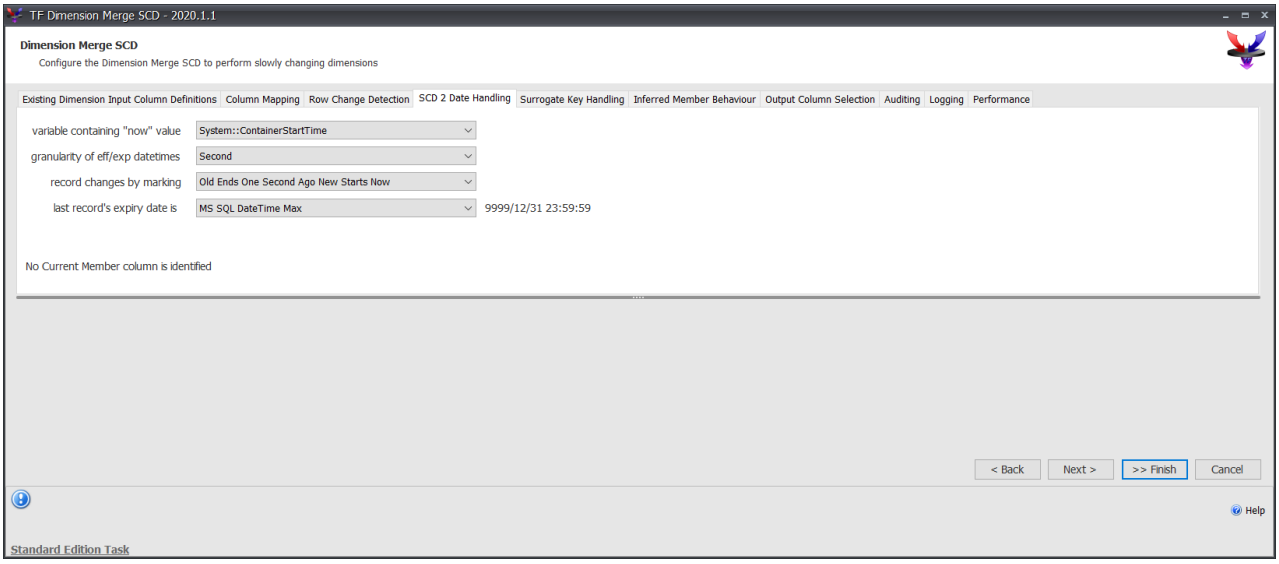
SCD2 Date Handling Tab

In the **SCD2 Date Handling tab**, you are given a variety of options for handling SCD2 dates including:

- Assigning **now** and **expiry** to system or user-created variables.
- The ability to select a granularity that ranges from day to millisecond (whose precision is also configurable).

- The ability to define a fixed-input expiry date.

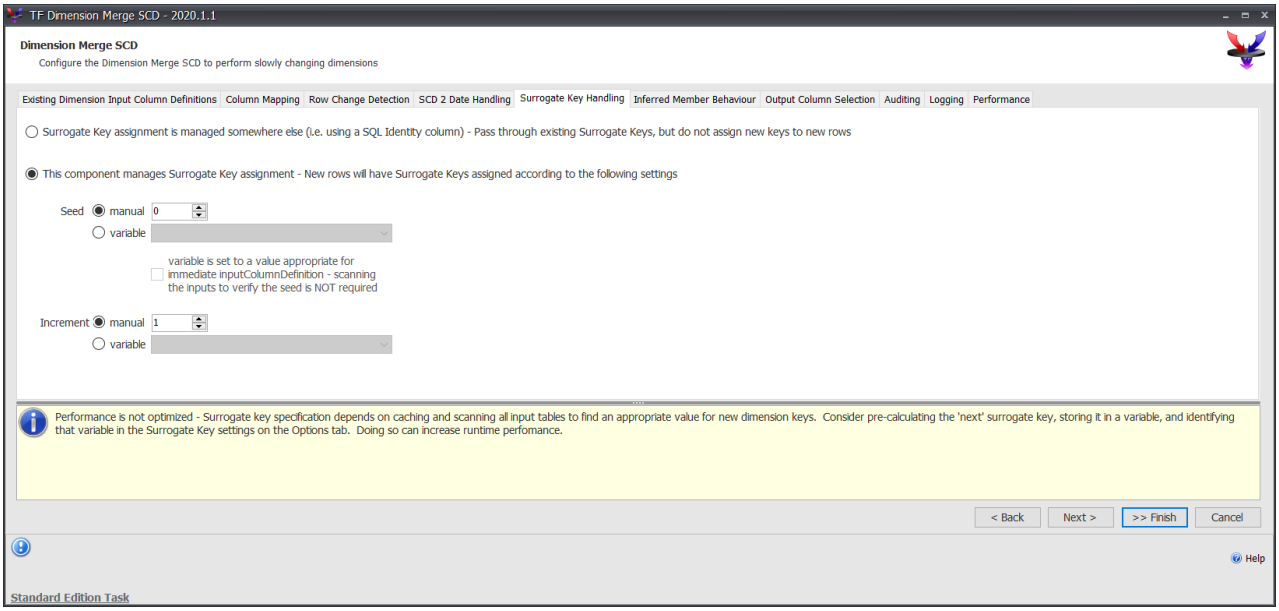
> ⚠ **Important:** You should pay careful attention to the example located to the right of the last record's expiry date. If the expiry column located in your dimension does not match the format of the example, this can cause unchanged rows to be sent to the **New** output due to the mismatch.
>
> For example: setting the granularity to millisecond with a last record's expiry date set to MS SQL DateTime Max requires a datetime entry of 9999/12/31 23:59:59.997. If a developer mistakenly uses .999, this causes a mismatch and sends unchanged records to the **New** output.



Surrogate Key Handling Tab

In this tab, you are given the option to manage surrogate key assignments to new rows or pass through the existing surrogate key and let the SQL identity column manage the assignment.

> ℹ **Note:** It's recommended to manage the surrogate key assignment somewhere else (first option) for optimal performance.

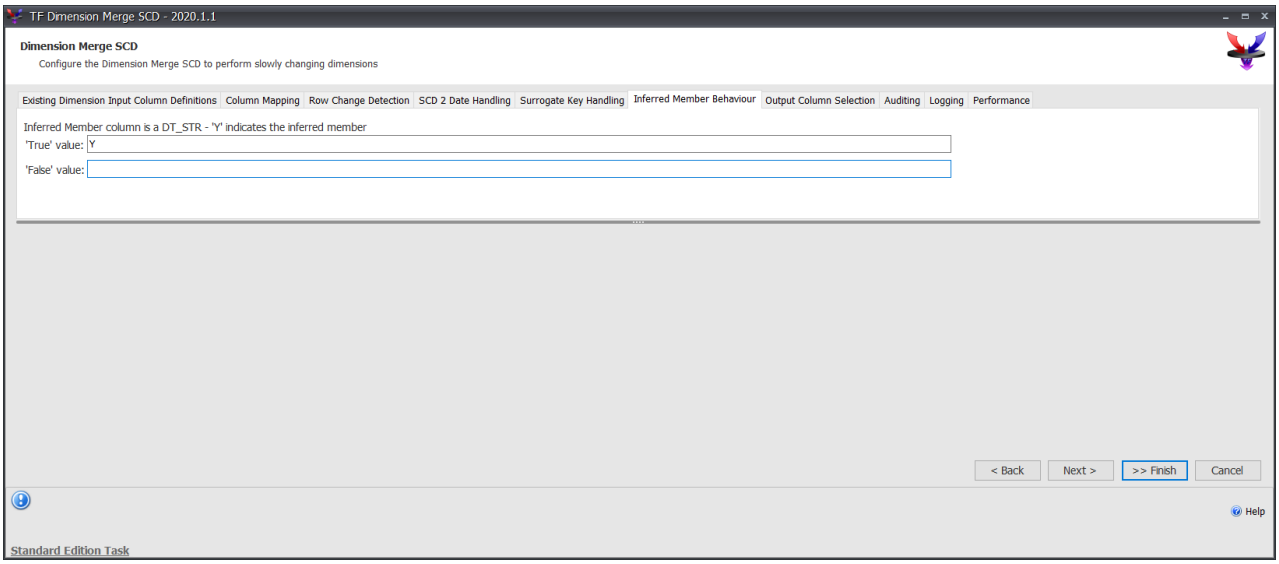


Inferred Member Behavior Tab

Use the **Inferred Member Behaviour** tab to confirm any inferred members that you have defined in the **Existing Dimension Input Column Definitions** tab. In SSIS, an inferred member acts as a placeholder value for currently unknown, or unloaded information. Inferred members are often used to replace these unknown or unloaded values that you don't want to treat as a NULL value. Inferred member values are replaced once the desired data loads into the package.

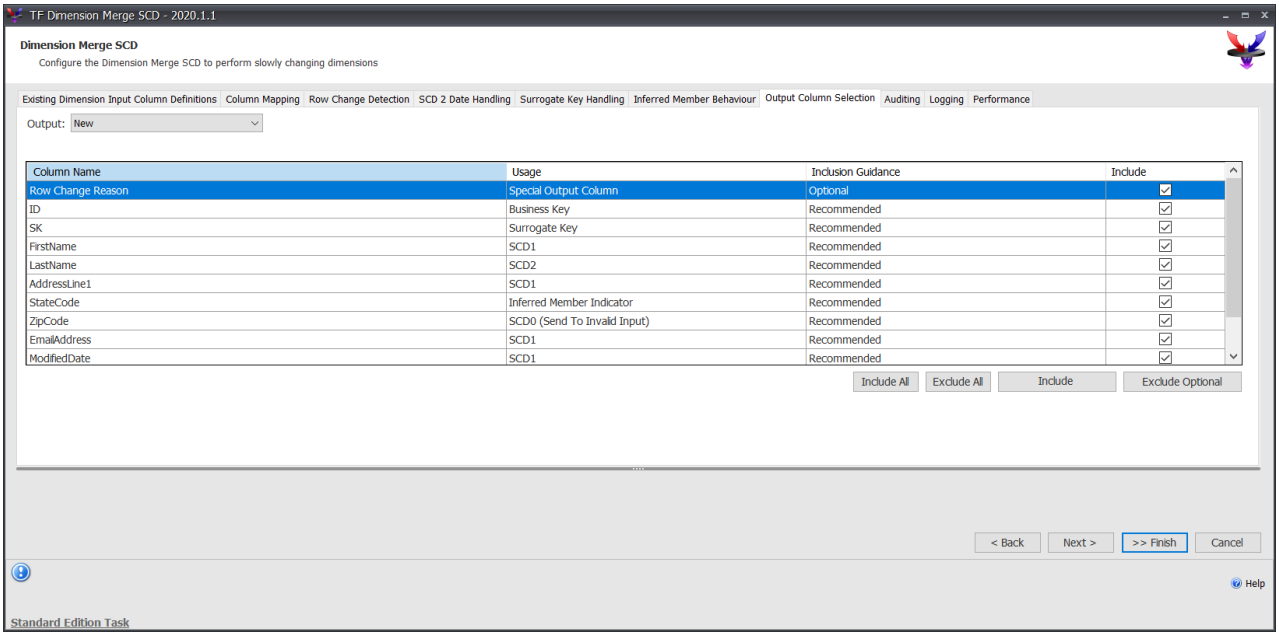
If you have tagged any column as an inferred member, those columns will display in the **Inferred Member Behavior** tab. Enter **Y** to indicate the value as **True** and confirm that the selected column is an inferred member, or select enter **N** to indicate that the value is **False** and that the column is not an inferred member.

Additional Information: For more information about Inferred Members in SSIS, see the Inferred Dimension Members MSDN article.



Output Column Selection Tab

In the **Output Column Selection** Tab, you can select which outputs to attach to the component and which columns are included/excluded in the output.

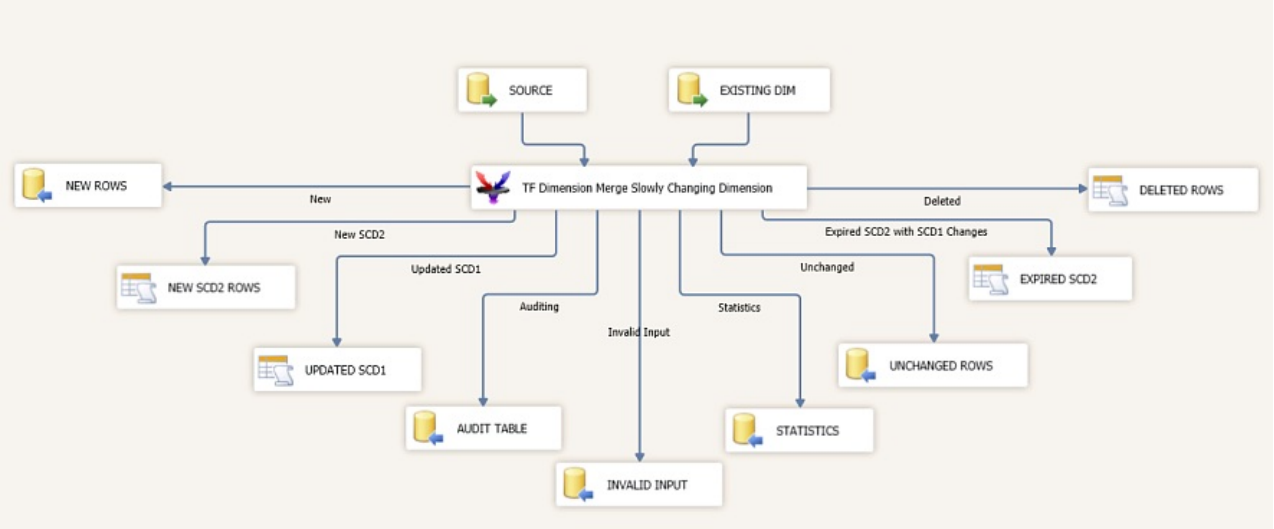


The final step in configuring the Dimension Merge SCD is to attach all of the outputs configured in the **Output Column Selection** tab. The following is a list of all available outputs as well as a brief description:

> ℹ **Note:** Once outputs are attached, re-open the Dimension Merge SCD Transform to dismiss warnings referencing unattached outputs. Keep in mind that warnings remain for any output that has been configured in the **Output Column Selection** tab that does not have the **Don't warn that this output isn't attached** button selected.

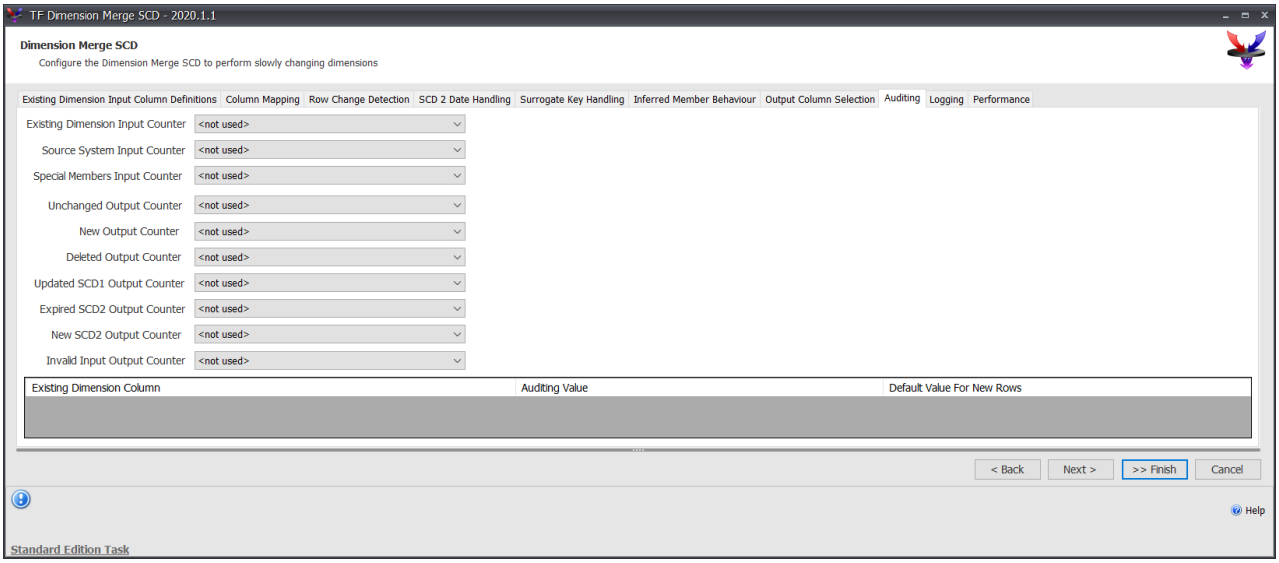
Output	Description
Auditing	The **Auditing** output processes Input/output counter information configured in the **Auditing** tab. If you don't want to include the **Auditing** output, you can dismiss the warning by selecting the **Auditing** output found on the Output tab, select the **Exclude All** button, and then select the **Don't warn that this output isn't attached** button.
Deleted	The **Deleted** output manages the records that are present and **active** within the Existing Dimension, but no longer appear within the Source System data stream. Records passed down the **Deleted** output can either be expired or deleted from the Existing Dimension. Expiration/deletion from the Existing Dimension are based on the Surrogate Key column(s). If you want your **Deleted** output to be expired, you can redirect your deleted rows to the **Expired SCD2 with SCD1 Changes** output. Additionally, if you don't want to include the **Deleted** output, you can dismiss the warning by selecting the **Deleted** output found on the **Output** tab, select the **Exclude All** button, and then select the **Don't warn that this output isn't attached** button.

Output	Description
Expired SCD2 with SCD1 Changes	The **Expired SCD2 with SCD1 Changes** output expires current SCD2 records that are being updated in the existing dimension. Records passed down the **Expired SCD2 with SCD1 Changes** are expired from the Existing Dimension based on the Surrogate Key column(s). If you don't want to include the **Expired SCD2 with SCD1 Changes** output, you can dismiss the warning by selecting the **Expired SCD2 with SCD1 Changes** output found on the **Output** tab, select the **Exclude All** button, and then select the **Don't warn that this output isn't attached** button.
Invalid Input	The **Invalid Input** output handles any invalid rows that match the configured rules found on the **Row Change Detection** tab (such as NULL values in columns not configured as Nullable). If you don't want to include the **Invalid Input** output, you can dismiss the warning by selecting the **Invalid Input** output found on the **Output** tab, select the **Exclude All** button, and then select the **Don't warn that this output isn't attached** button.
New	The **New** output handles records that are not present within the Existing Dimension, but found within the Source System. Any record passed down the **New** output needs to be inserted into the Existing Dimension. If you don't want to include the **New** output, you can dismiss the warning by selecting the **New** output found on the **Output** tab, select the **Exclude All** button, and then select the **Don't warn that this output isn't attached** button.
New SCD2	The **New SCD2** output works along side the **Expired SCD2 with SCD1 Changes** output to insert a new record for historical records. If you don't want to have an additional output, you can redirect the **New SCD2** output to the **New** output. If you don't want to include the **New SCD2** output, you can dismiss the warning by selecting the **New SCD2** output found on the **Output** tab, select the **Exclude All** button, and then select the **Don't warn that this output isn't attached** button.
Statistics	The **Statistics** output processes statistical information for the current package execution such as the number of Source System Input records, Existing Dimension Input records, and Interval Start/End times. If you don't want to include the **Statistics** output, you can dismiss the warning by selecting the **Statistics** output found on the **Output** tab, select the **Exclude All** button, and then select the **Don't warn that this output isn't attached** button.
Unchanged	The **Unchanged** output is an optional output that outputs all rows that have no changes between the Existing Dimension and the Source System data stream, or rows, that do not appear in the Source System data stream and are expired in the Existing Dimension. If you don't want to include the **Unchanged** output, you can dismiss the warning by selecting the **Unchanged** output found on the **Output** tab, select the **Exclude All** button, and then select the **Don't warn that this output isn't attached** button.
Updated SCD1	The **Updated SCD1** output records that contain updated values to non-historical columns. If you don't want to include the **Updated SCD1** output, you can dismiss the warning by selecting the **Updated SCD1** output found on the **Output** tab, select the **Exclude All** button, and then select the **Don't warn that this output isn't attached** button.



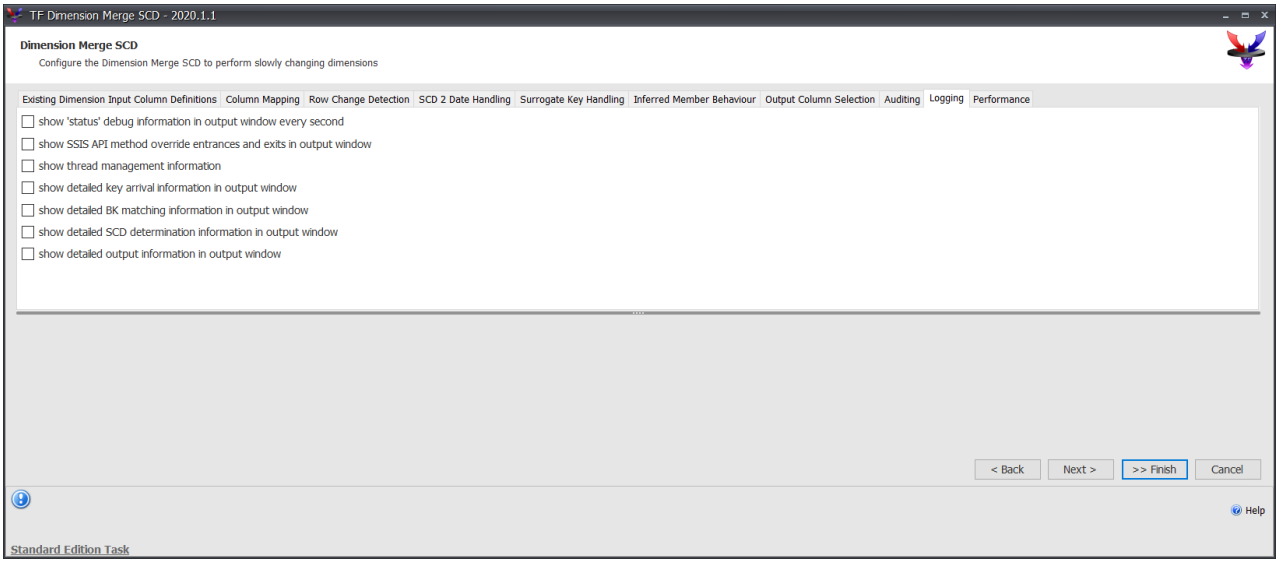
Auditing Tab

In the **Auditing** tab, you can set variables to display dimension, source system, and output counters that can be analyzed post-execution.



Logging Tab

Use the **Logging** tab to provide more descriptive information of the execution process viewed in the **Execution Results/Progress** tab within SSDT/BIDS.



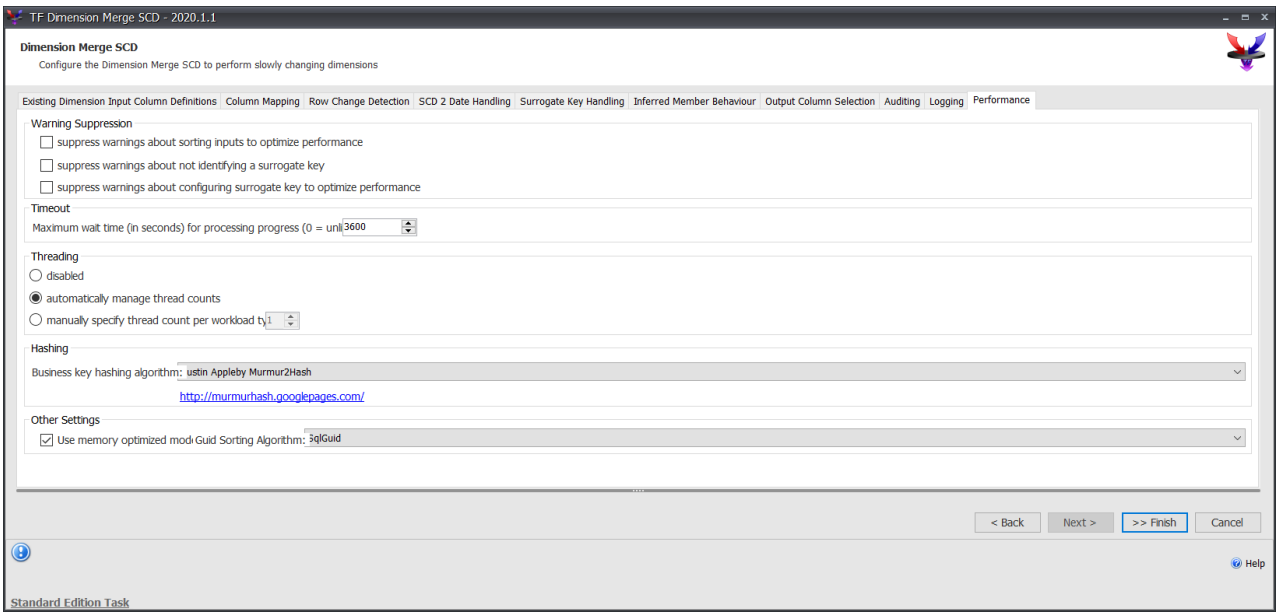
Performance Tab

In this tab, you can alter performance settings including:

Option	Description
Warning Suppression	Dismisses warnings when circumstances require a condition be ignored (such as not sorting input business keys).
Timeout	Developers can define a time (in seconds) to end execution for debugging.

<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>Threading</td><td>Sets the number of threads available in two pools - one for matching keys and the other for processing rows. Users can maximize processing speed by configuring the component as follows:
<table>
<tr><th>Setting</th><th>Description</th></tr>
<tr><td>Disabled</td><td>Sets the value to 1.</td></tr>
<tr><td>Automatically manage thread counts</td><td>Value is set to a number based on the processor cores of the machine it is being run on. ℹ Note: This option should be considered for smaller environments.</td></tr>
<tr><td>Manually specify thread count per workload type</td><td>Should be considered when running on an environment with multiple core processors. Additionally, the automatic option doesn't take into account other processes or packages that might be running, therefore, manually specifying the thread count might be a better option for larger or busy environments.</td></tr>
</table></td></tr>
<tr><td>Hashing</td><td>Security feature that allows you to determine which hash algorithm to use during processing.</td></tr>
<tr><td>Use Memory Optimized Mode</td><td>Sets the Memory Optimized property. For more information, see Memory Optimized Property under the Setup tab.</td></tr>
</table>

<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>Guid Key Sort Algorithm</td><td>Identifies the method in which the input data is sorted when using Guid as relationship keys. (This option only appears when Guid keys are used.) The following methods can be used:
<table>
<tr><th>Setting</th><th>Description</th></tr>
<tr><td>SQL Server Order By</td><td>Performed by using an ORDER BY clause within a query. Additionally, the IsSorted property needs to be set to true and the key sort order set. These are found in the sources Advanced Editors (eg Ole DB Source and ADO.Net Source.)</td></tr>
<tr><td>Sort Transform</td><td>Configured when using the SSIS native Sort Transform component between the source(s) and Dimension Merge SCD transform. (Sorting is done by adding an ORDER BY clause, however, the IsSorted and SortKeyPosition should not be set within the advanced editor.)</td></tr>
<tr><td>.NET Guid</td><td>Configured when using .NET Guid sorting within an upstream Script Task.</td></tr>
<tr><td>String</td><td>Sort Guids as strings. Note: Does not work with Memory Optimized mode set to true.</td></tr>
</table></td></tr>
</table>



Support Articles

Table Locking with SCD Merge

The above article provides a solution for slowly changing Dimension Merge components that appear to hang and not complete execution.

Additional Videos

Dimension Merge SCD - Configure Native SCD Example

Dimension Merge SCD - How to use it

Dimension Merge SCD - Intro to Sample Package

Dimension Merge SCD - Intro to test data

Dimension Merge SCD - T-SQL Merge Example

Task Factory Error Output Description Transform

Last Modified on 18 August 2022

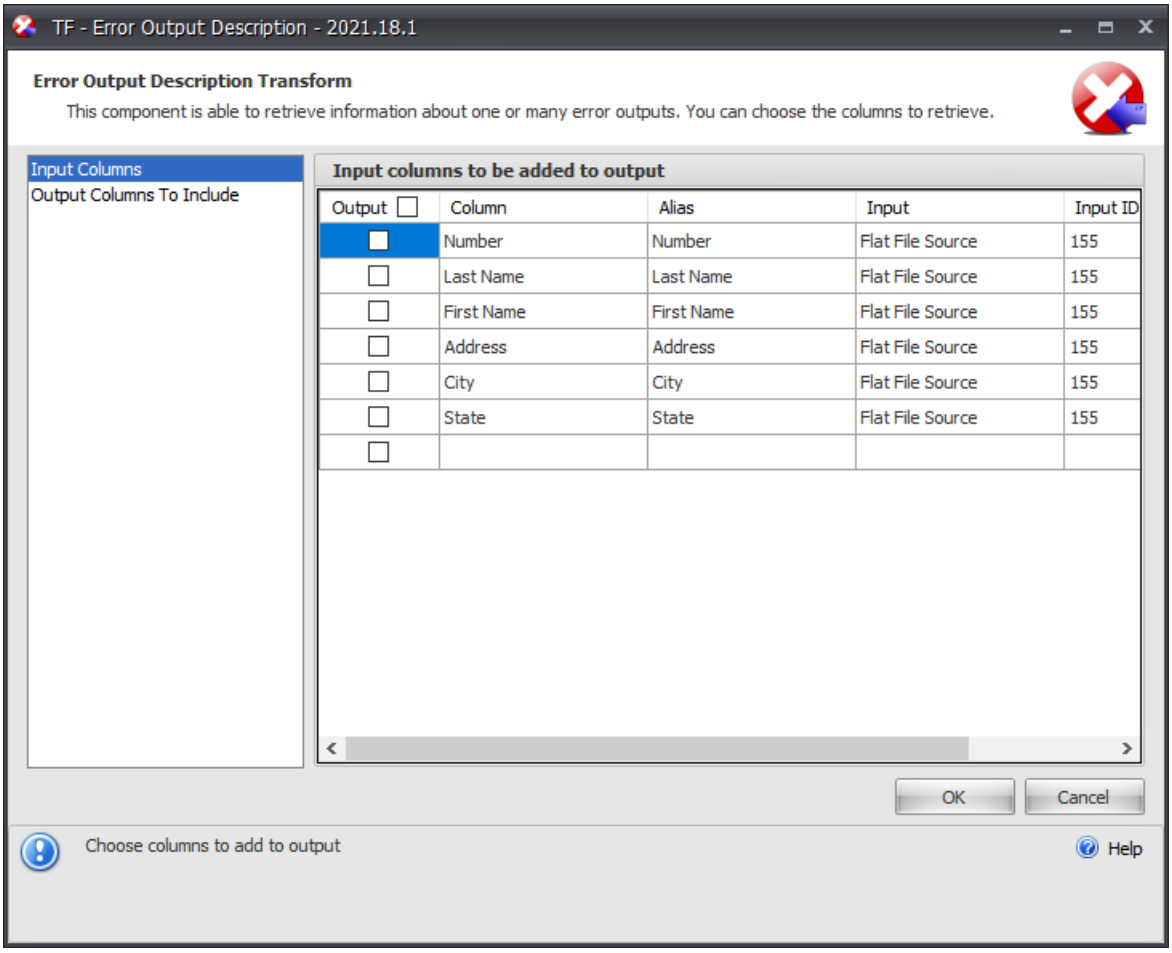
Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Note: The error output description allows for multiple error outputs from other components.

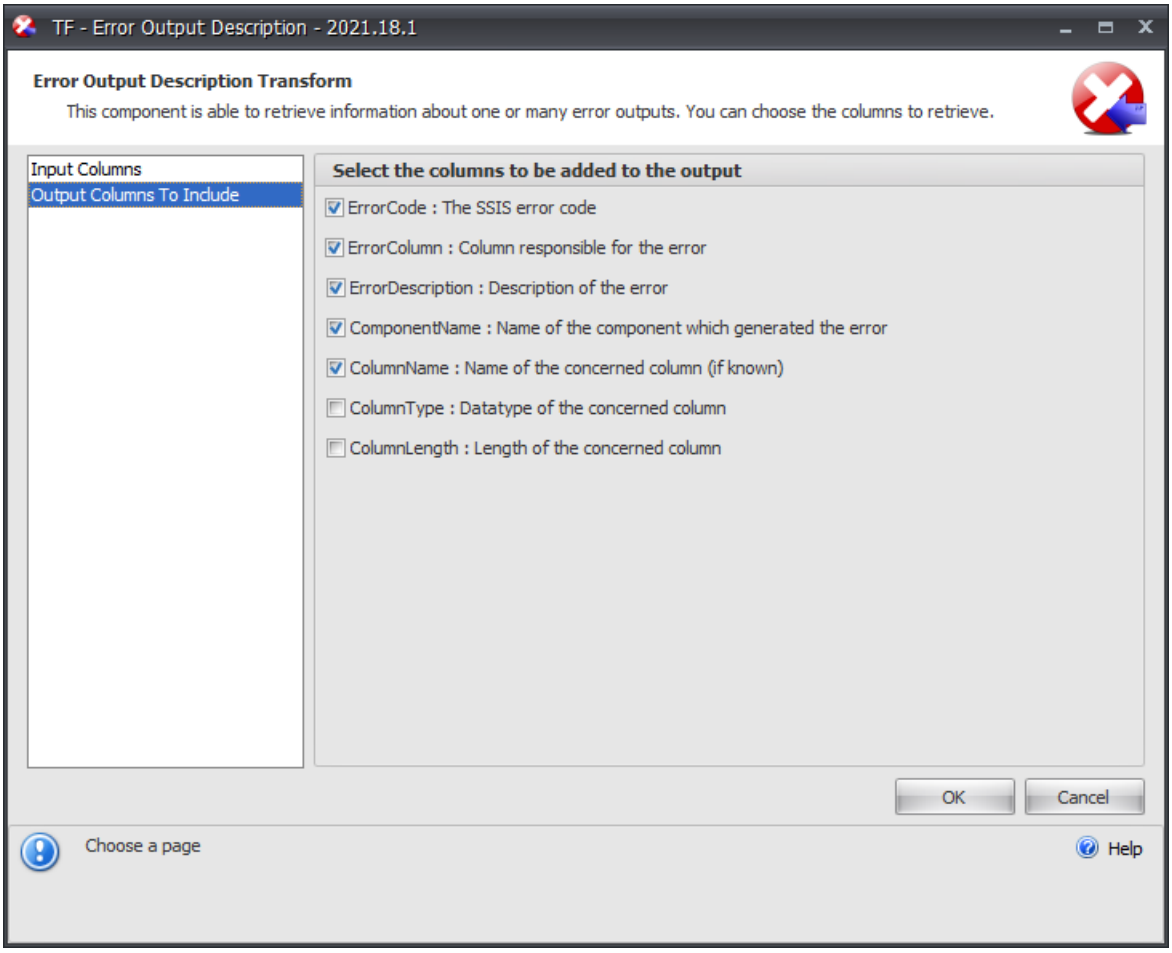
Error Output Description Transform

Transform Icon	Transform Description
	The **Error Output Description Transform** is used to add descriptive errors to your data flow components. To use the **Error Output Description Transform**, add the error output of a source or transform components to the error description and then output the more descriptive error data to a destination.

Input Columns



Option	Description
Output	Select this option to include original column in the output. (The box next to **Output** selects all columns.)
Alias	Select inside to rename the column.
Input	The input type of the information source.
Input ID	The input ID of the information source.
Lineage ID	The Lineage ID for the information source.

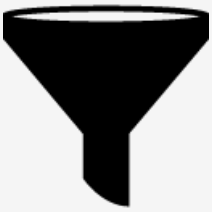

TF - Error Output Description - 2021.18.1
Error Output Description Transform
This component is able to retrieve information about one or many error outputs. You can choose the columns to retrieve.
Input Columns
Output Columns To Include
Select the columns to be added to the output
ErrorCode : The SSIS error code
ErrorColumn : Column responsible for the error
ErrorDescription : Description of the error
ComponentName : Name of the component which generated the error
ColumnName : Name of the concerned column (if known)
ColumnType : Datatype of the concerned column
ColumnLength : Length of the concerned column
OK
Cancel
Choose a page
Help

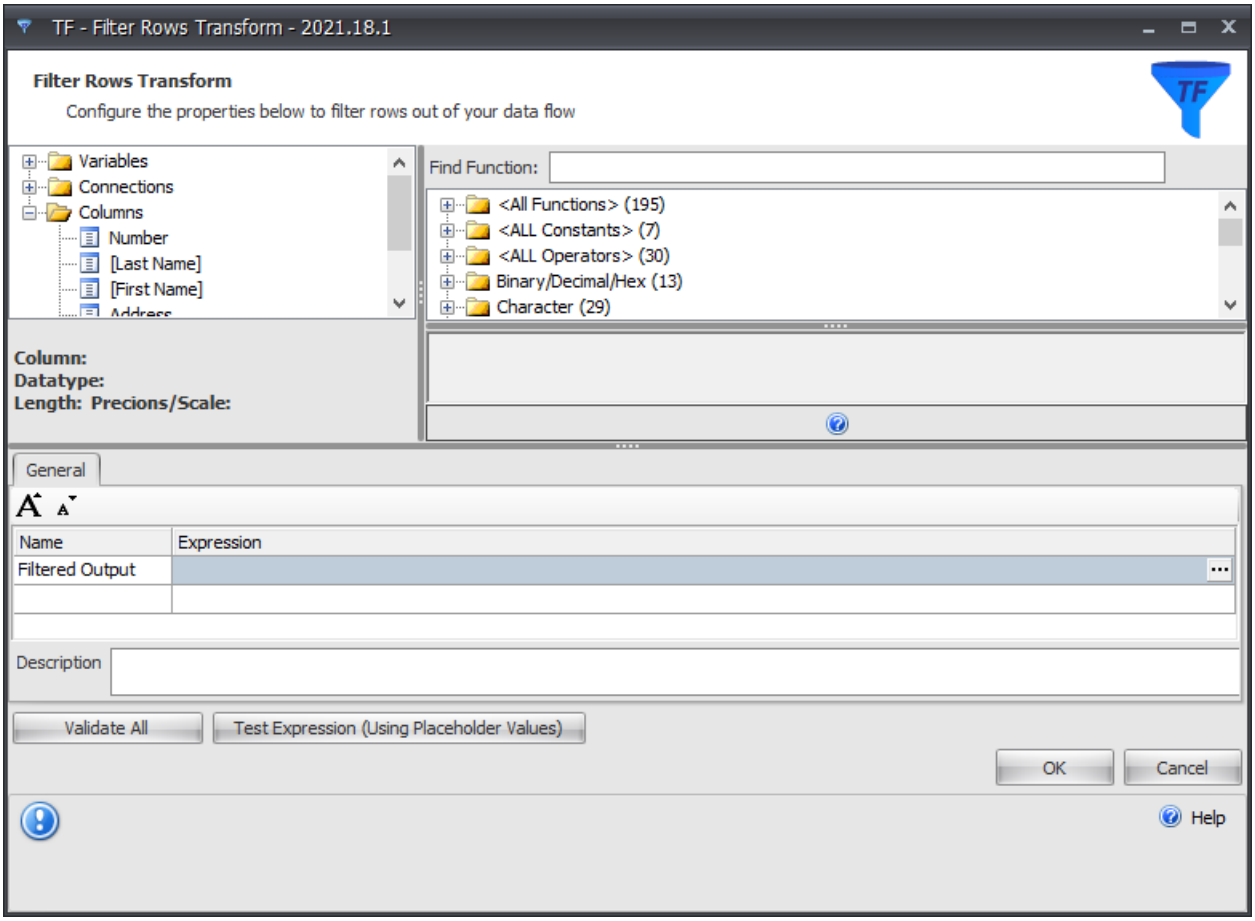
Task Factory Filter Rows Transform

Last Modified on 18 August 2022

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Filter Rows Transform

Transform Icon	Transform Description
	The **Filter Rows Transform** is used to filter rows from an upstream based on an expression.



Option	Description
Column Browser	Located in the top left, the column browser show all available columns, variable and connections that can be used as part of a derived column expression.
Functions Browser	Located in the top right, the functions browser lists all of the available functions that can be used in the advanced derived column.
Filter Grid	The filter grid allows you to define a single expression to filter rows matching the expression. • **Name** - This is the name of the output where any rows that do not match any of the expressions are routed. • **Expression** - This is the expression used to filter the rows of data.
Validate All	The **Validate All** button evaluates all of the expressions and checks if the expression has errors. An error or warning icon displays if there are any issues with the expressions.
Test Expression	This button creates temporary values based on the data type and executes the expression that is currently selected in the grid.

Available Functions List

For a list of all available functions, along with syntax, descriptions, and examples, see the Functions List

articles.

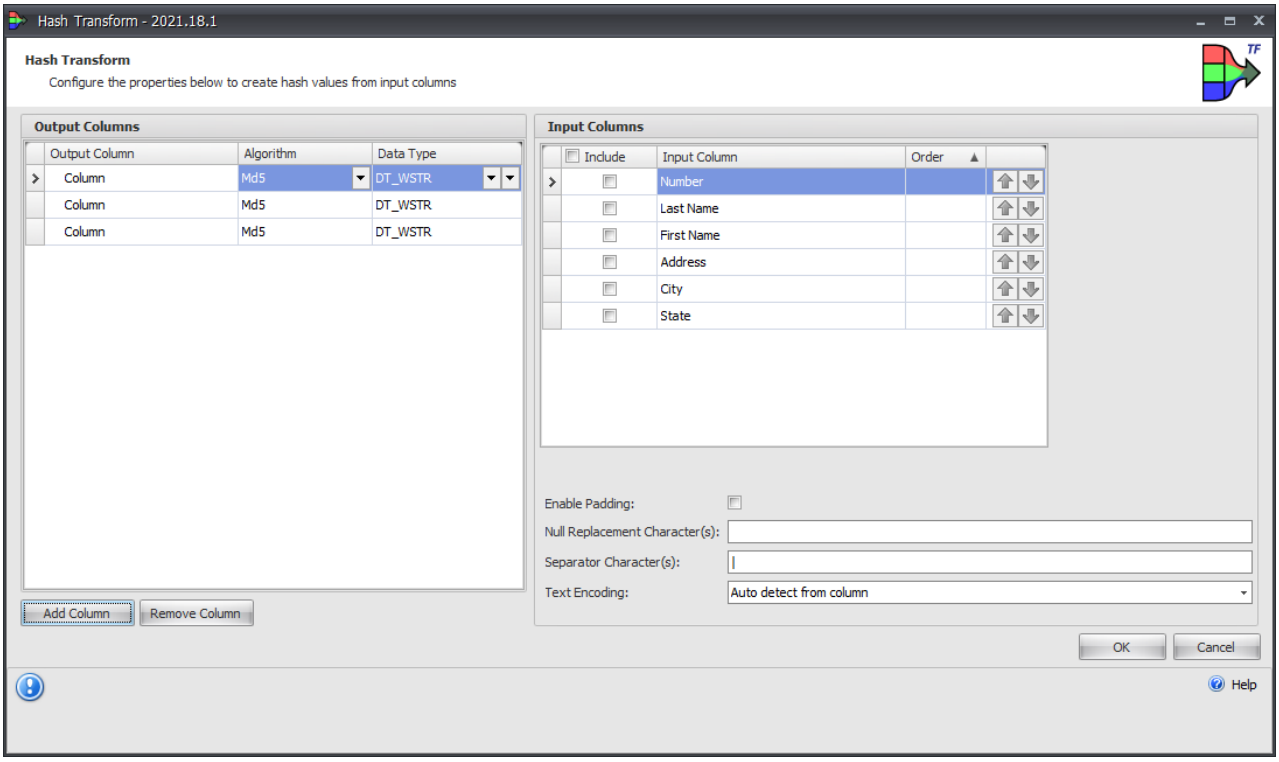
Task Factory Hash Transform

Last Modified on 18 August 2022

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Hash Transform

Transform Icon	Transform Description
	The **Hash Transform** is used for creating sets of hash values as new output columns from one or more input column.



Output Columns

Option	Description
Add Column	Creates a new output column output with the hashed value once configured.
Remove Column	Removes the currently selected column.
Output Column	The column name to be output.
Algorithm	The hashing algorithm to be used by the output column.

Input Columns

Option	Description
Include Column	Selects the input column to be hashed.
Input Column	The name of the input column.
Order	When combining columns, users can change the sequential order of the input columns by selecting the **up** and **down** buttons.

Option	Description
Enable Padding	Toggles the ability to include column padding in the output.
Null Replacement Character(s)	Users can replace Input Column NULL values with a designated character.
Separator Character(s)	When combining columns, this character is added between the selected columns. **Note:** The pipe character **|** is used by default when no value is entered.

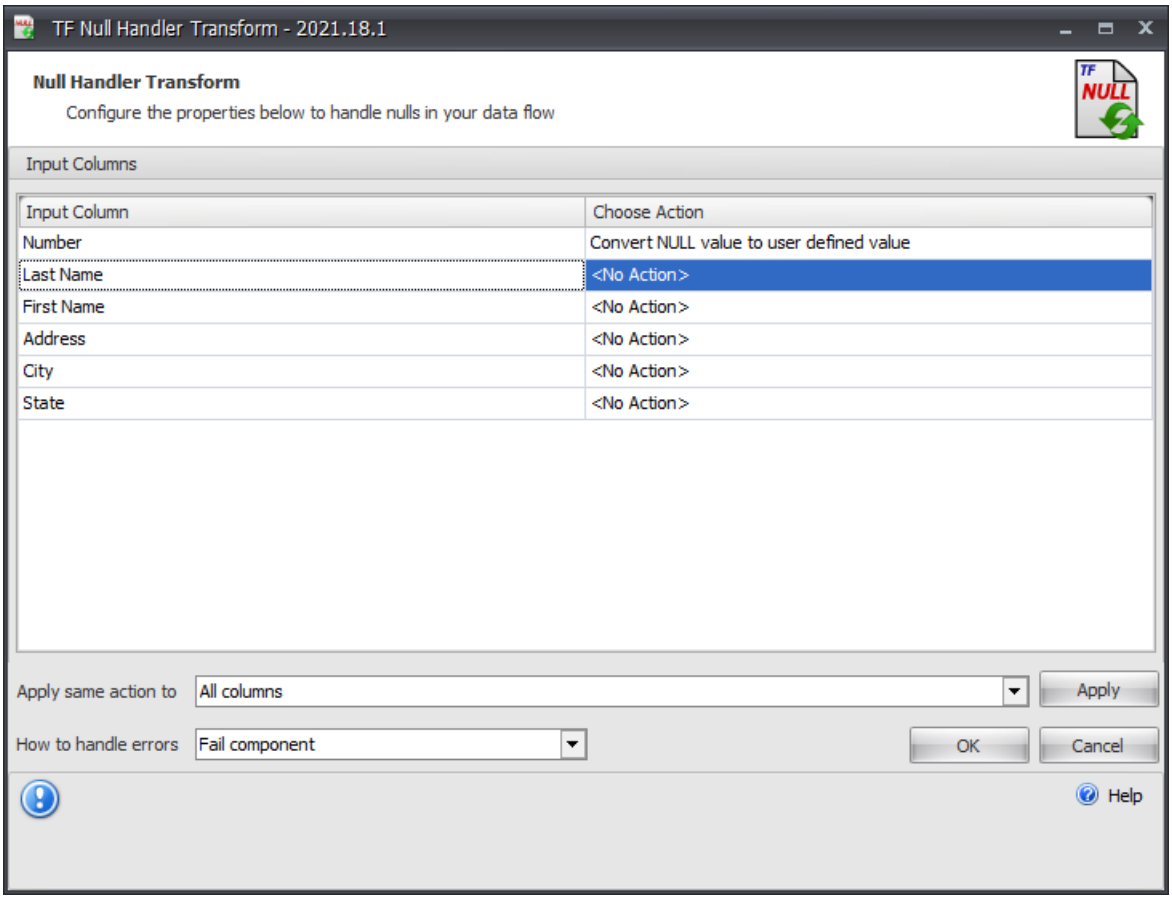
Task Factory Null Handler Transform

Last Modified on 16 August 2022

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Null Handler Transform

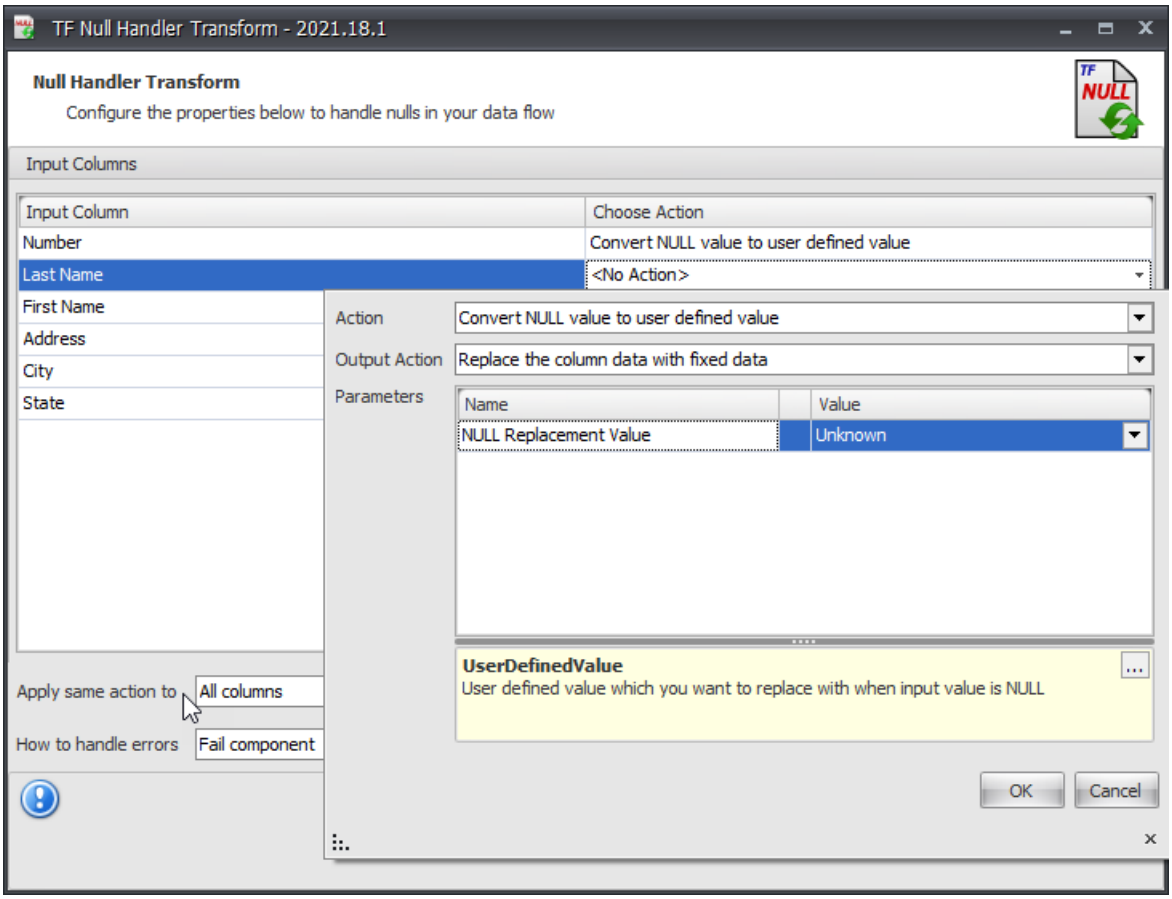
Icon	Description
	The **Null Handler Transform** allows the user to replace null or empty data with user specified data.



Input Columns

Option	Description

Option	Description
Input Column	The name of the input column.
Choose Action	There are three actions that can be defined to replace null or blank data: • **Replace Null Value with User Defined Value** - This action is used to replace null data with a user defined value. • **Replace Blank Value with Null** - This action is used to replace blank values with null value. This is important where systems depend on data being null in a column. • **Replace Blank Value with User Defined Value** - This action is used to replace empty strings with a user defined value.



Additional Information: See the Task Factory Error Row Handling article for more information about this functionality.

Task Factory Pack Data Transform

Last Modified on 17 August 2022

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

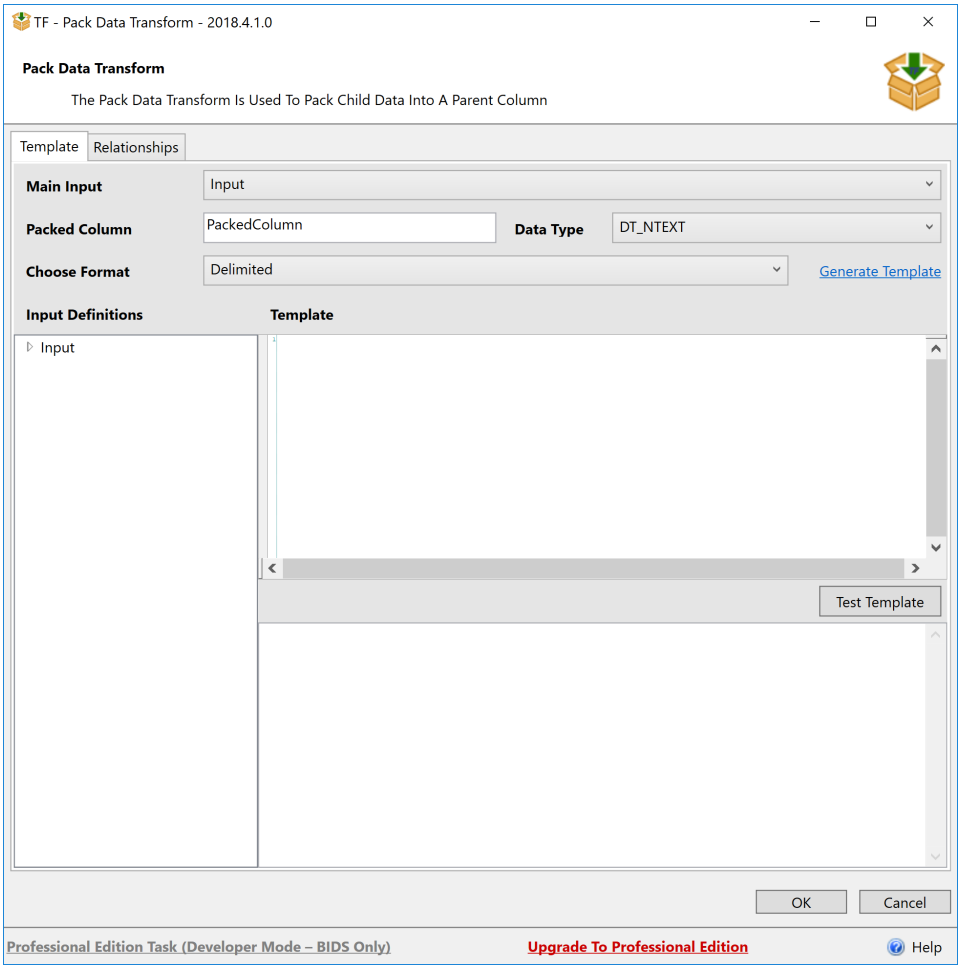
Pack Data Transform

Note:

- Pack Data is available for SQL Server versions 2012 and higher.
- When using multiple sources, all inputs must be sorted.

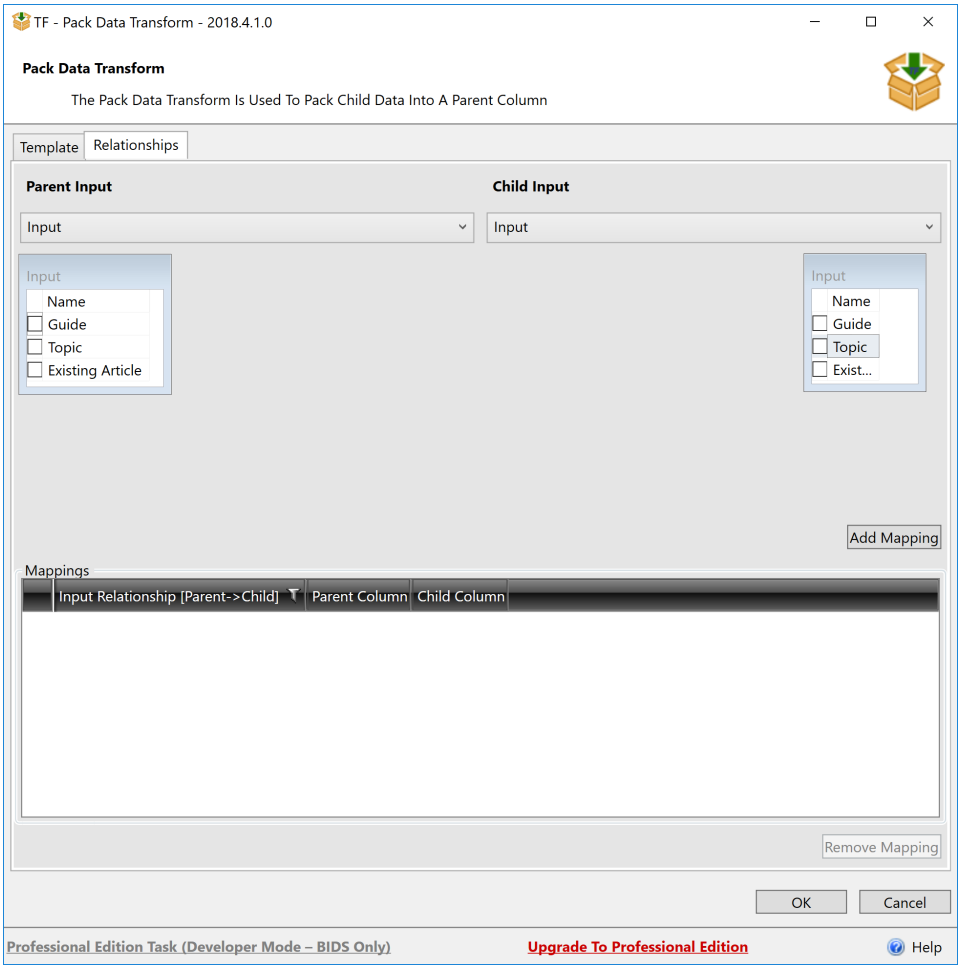
Transform Icon	Transform Description
	The **Pack Data Transform** is used to create a single column of delimited, XML, or Json data (determined by a user-defined template) from a single or multiple input columns.

Template Tab



Option	Description
Main Input	User selects the parent input.
Packed Column	User-defined name of the column to be output with packed data.
Data Type	User can select between two data types: DT_NTEXT or DT_TEXT.
Generate Template	Users can choose the packed data format (Delimited, XML, or Json) and the component automatically generates a template to be used for the packed output. These templates can be adjusted and customized by the user by deleting information not used or drag-and-dropping columns from the **Input Definitions** window.
Input Definitions	List of columns from the inputs.
Template	The design pane where users can view and customize the template used to generate the packed output.

Relationships Tab



Option	Description
Parent Input	Selects an input to identify as the parent.
Child Input	Selects an input to identify as the child.
Update Mapping	After connecting the two input keys, select this button to update the component with the defined relationship.
Remove Mapping	Selecting this button removes a selected relationship highlighted in the **Mappings** pane.

Pack Data Transform Generating Templates

The Pack Data Transform uses dotliquid as the templating engine which can be confusing for some users. The purpose of this page is to help users understand how the syntax is used within the component. This should also help users learn how to use the pre-configured templates as well as create their own within the component.

First, it's important to understand that you can create any delimited, Xml, or Json formatted template. To quickly generate one based on the input(s), select the **Generate Template** hyperlink. The Template window populates based on the selected format and input(s). Columns can also be added manually by dragging from

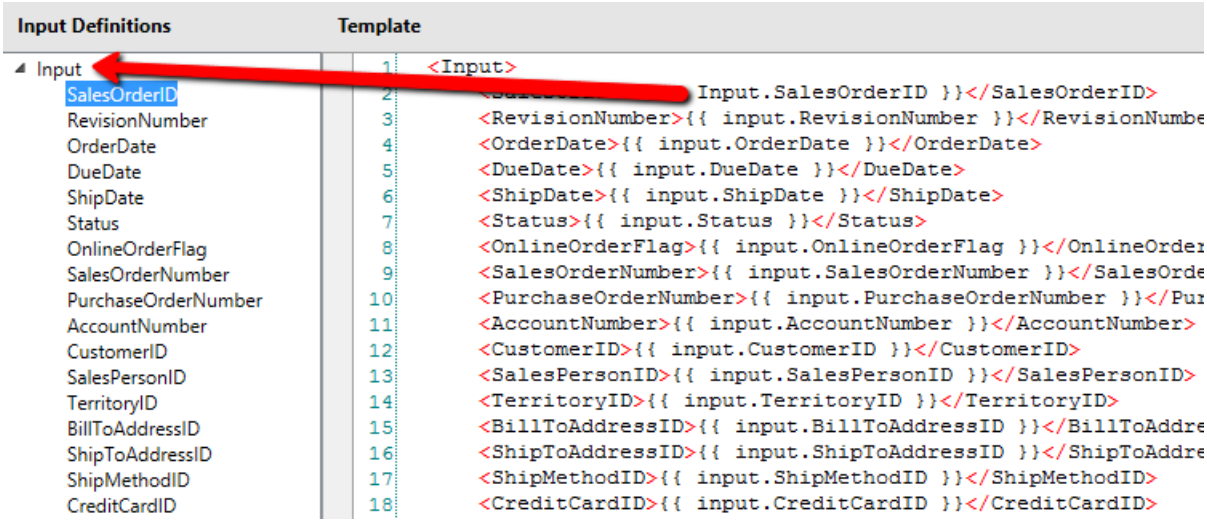
the **Input Definitions** window and dropping to the Template window.

Syntax

Data replacements are in the format of {{inputname.columnname}} Example: Everywhere the user sees {{Input.SalesOrderID}}, it will be replaced with the row data for SalesOrderID. It's further broken down as follows:

- Input = the name of the input on the left side of the UI (the top node of the Input Definitions window).
- SalesOrderID = column name from that input.

As you can see in the example below, the SalesOrderID input on the left corresponds to the {{Input.SalesOrderID}} in the XML on the right:



Adding IF statements

If conditional statements act the same as other languages (except uses the dotliquid syntax.) Users can add If statements by contributing the following:

```
{% if forloop.index > 1%},{% endif %}
```

IF Statements Example

```
{% if row1.CarID == 123 %} Write any text here {% endif %}
```

This reads **If CarID in row1 is equal to 123,** then **Write any text here** is added to the output. See the table below:

CarID	Info
123	Write any text here
456	Data

CarID	Info
789	Data

As you can see, the Info column added **Write any text here** because it matched the condition that **CarID = 123**.

> **Note:** Spaces must surround equality/inequality operators used in If statements.

Comments

Comments can be added between objects Simply use the following tags:

```
{% comment %} {% endcomment %}
```

Comments Example

```
{{input.SalesOrderID}}{% comment %} add comma between objects - this comment tag can be removed from temp
late {% endcomment %}
```

Loops

Loops can only be used with more than one input. In some cases, users may have to loop through more than one row that shares the same ID. To use loops in the Pack Transform, add the following:

```
{% for row1 in input1.Rows -%}
```

Row1 is the name of the loop iterator and is used to access any data from the rows in input1.

Loop Example

```
{{ row1.CarID }}
{{ row1.CarName }}
{% endfor -%}
```

Using more than one input with a parent – child relationship

The tables below are used to establish the parent-child relationship.

Parent Input has a single row with two fields. The parent is named **Input** (seen in the Input Definitions window).

OwnerID	Name
1	Jane Fields

Child input has two rows with two fields. The child is named **Input1** (seen in the Input Definitions window).

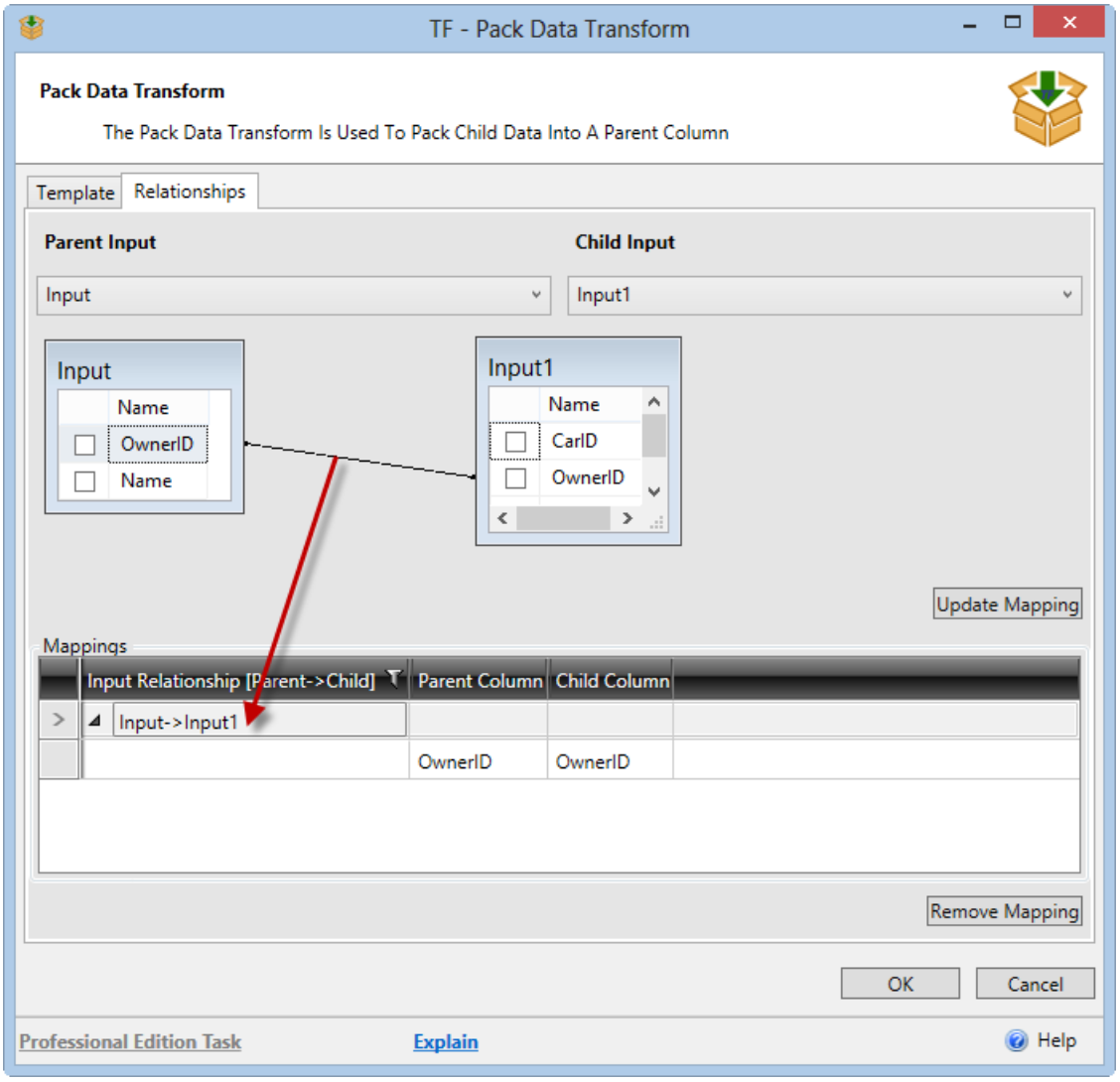
OwnerID	CarID	CarName
1	1	Maxima
1	2	Explorer

There is a relationship defined between the two inputs on OwnerID from Input and Input1.

Users can use the generate template once the relationship has been defined.

When using multiple inputs, a relationship between the two is needed in the relationship tab. To add this relationship, complete the following steps:

1. Select the Relationships tab.
2. Select the Parent Input in the first selection box.
3. Select the Child Input in the second selection box.
4. Drag the parent column key to the child column key that links the tables.
5. Select the **Update Mapping** button.
6. (Optional) Return to the Template tab and select the **Generate Template** hyperlink to auto-generate your template.



Multiple Input Example

We want to output an xml template defined as:

```
{{ input.Name }}
{{input.ID }}

{% for row1 in input1.Rows -%}
        {{ row1.CarID }}
        {{ row1.CarName }}
{% endfor -%}
```

In this example, row1 is the name of the loop iterator and is used to access any data from the rows in input1.

> **Note:** Templates with columns not contained upstream display a warning. Please ensure the column is available upstream or is not misspelled to dismiss the warning. This does not prevent the component from executing.

Additional Information: As noted previously, the Pack Data Transform uses dotliquid as the templating engine. Documentation on the syntax can be found here:

- Liquids for Designers
- Liquid Basics Introduction

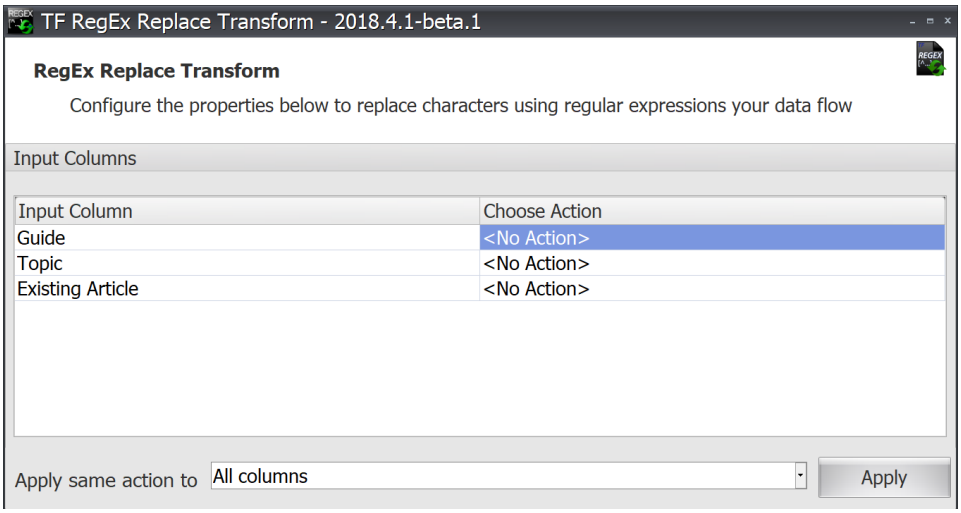
Task Factory Regex Replace Transform

Last Modified on 17 August 2022

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

RegEx Replace Transform

Transform Icon	Transform Description
	The **RegEx Replace Transform** allows the user to replace or extract data from a source column using regular expressions.



Input Columns

Option	Description
Input Columns	

Option	Description
Choose Action	The RegEx Replace Transform has two actions that can be used to either replace or extract data from a source column. • **Replace matching regular expression pattern with a user defined value** - This action allows the user to define a regular expressions to search column data and replace the data if the regular expression matches. • **Extract Data from input string using regular expression** - This action extracts data from a source column using regular expression.

Additional Information: See the Task Factory Error Row Handling article for more information about this functionality.

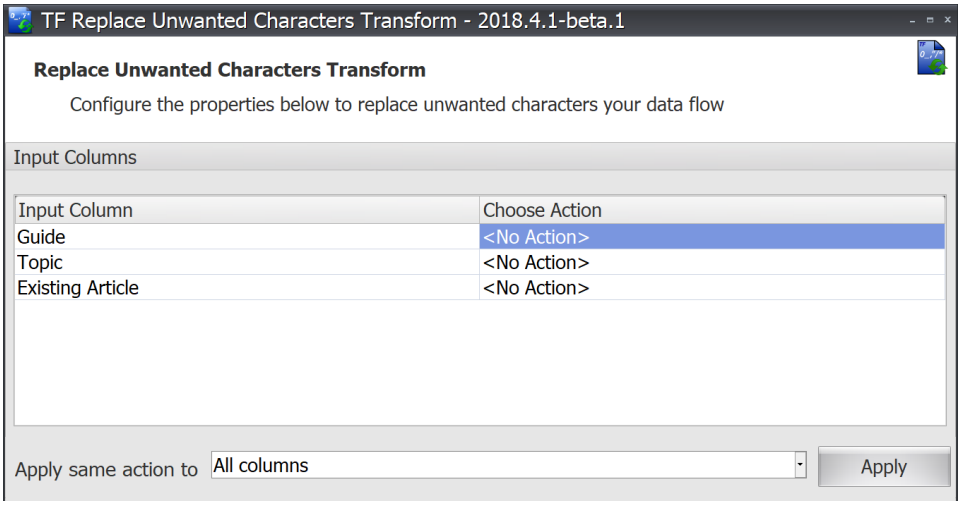
Task Factory Replace Unwanted Characters Transform

Last Modified on 18 August 2022

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Replace Unwanted Characters Transform

Transform Icon	Transform Description
0_,"/*	The **Replace Unwanted Characters Transform** is a flexible component that's used to remove and replace unwanted characters or words from your source data.



Input Columns

Option	Description
Input Column	Name of the input column.
Choose Action	There are two actions that can be used: • **Action** • **Output Action**

Action

Replace Alpha/Alphanumeric/Numeric characters and replace with user defined value - This option gives you six predefined options to replace characters. This option is useful if you need a simple way to remove unwanted characters without knowing the specific characters you want to remove, but know what you want to keep.

Option	Description
Parameters - Action	• **ReplaceAlpha** - Remove/replace letters in the alphabet from the input string. • **ReplaceNonAlpha** - Remove/replace anything other than letters in the alphabet from the input string. • **ReplaceNumeric** - Remove/replace numbers from the input string. • **ReplaceNonNumeric** - Remove/replace anything other than number from the input string. • **ReplaceAlphaNumeric** - Replace/remove any characters [A-Z], [a-z], and [0-9] from the input string. • **ReplaceNonAlphaNumeric** - Replace/remove any characters other than [A-Z], [a-z], and [0-9] from the input string.

Option	Description
Parameters - Replace Match Character With	The character or word to replace the characters defined in the action.

Replace Specified characters or words with user defined value - This options gives you the ability to replace any characters or words from the input source with a user defined value. This is useful when you need to be more selective with the characters that are replaced.

Parameters

Option	Description
Characters to be searched	Define what characters should be replaced. Certain characters need to be entered as escape sequences: • **Tab** - \t • **CarriageReturn** - \r • **LineFeed** - \n • **VerticalTab** - \v • **SingleQuote** - \' • **DoubleQuotes** - \" • **NullCharacter** - \0
Words to be searched	Define words or Unicode characters to be replaced. All words must be separated by a vertical bar (**|**). If you have some Unicode characters you want to replace then you must specify **\u**. For example to replace a Greek letter you would specify \u03B2.
Replace with	Define the characters or words to replace the unwanted characters or words defined in the **Characters to be searched** and **Words to be searched** parameters. Usually this value is left blank.
Ignore Case	Choose whether the words or characters being searched should use case matching. **Note:** The default is false.
Maximum Replacements	Specify how many occurrences you want to replace. Setting this value to -1 replaces all occurrences in the input string.

Replace invalid characters that cannot be part of an Xml Document - Replaces the **&**, **<** and **>** as well as **"** or **'** (double and single quote) within attributes.
Replace non-printable characters - Replaces all non-printable characters such as tab and carriage return.

Output Actions

Replace the column data with fixed data -

Output the fixed data to a new output column - The new output column name is **inputcolumn_output** for

example, if the source column is **FirstName** the output will be **FirstName_output**.

Additional Information: See the Task Factory Error Row Handling article for more information about this functionality.

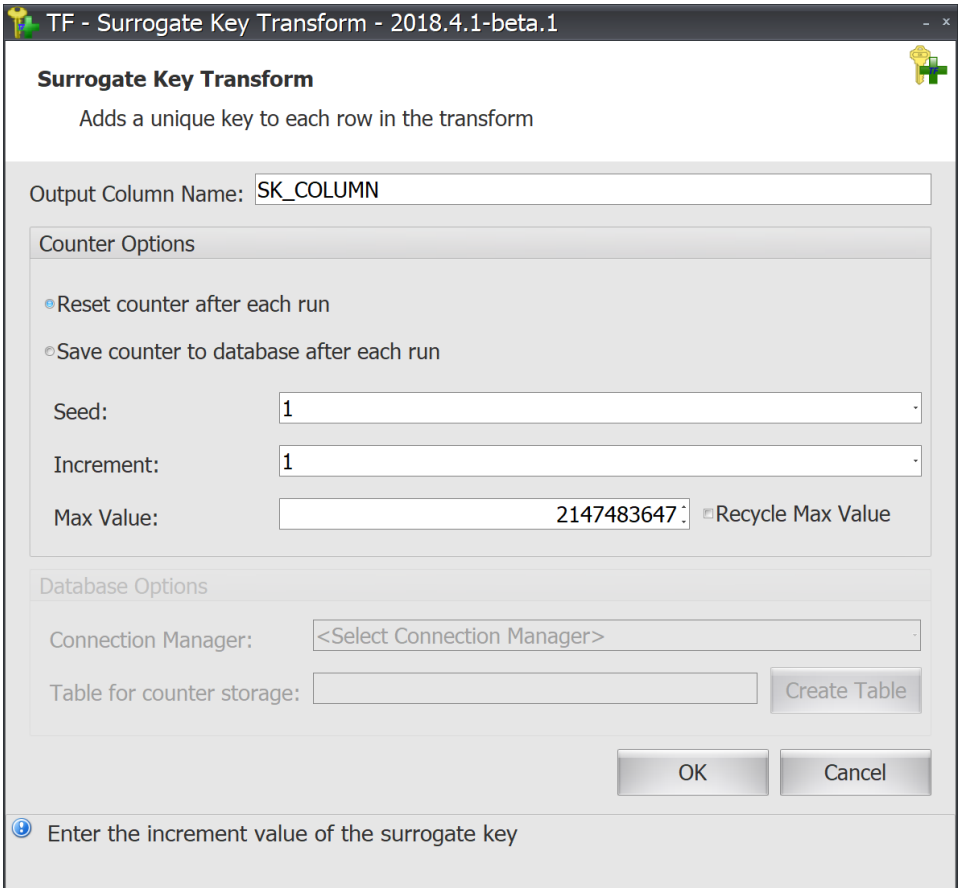
Task Factory Surrogate Key Transform

Last Modified on 19 August 2022

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Surrogate Key Transform

Transform Icon	Transform Description
	The **Surrogate Key Transform** adds a unique identifier column to source data. You can add the output column name, set the seed and increment of the output column generated by the surrogate key transform.



Output Column Name

The output column name is used to define the name of the surrogate key column that's included in the transform when executed.

Counter Options

Option	Description
Reset counter after each run	This option adds a surrogate key value to the data flow that starts at the seed value and increments based on the increment value.
Save Counter to database after each run	This option allows you to save the surrogate key value to a table in your database after each run. Choosing this option enables a few more options you must configure under the **Database Options** section.
Seed	The seed property is used to define the starting point for the surrogate key data. You can define the seed two ways: • **Text** - The surrogate key data starts with the number entered into the textbox. • **Variable** - The surrogate key data starts with the value from the selected variable.

Option	Description
Increment	The increment property is used to define the increment of the surrogate key data. You can define the increment two ways: • **Text** - The surrogate key data increments with the number entered into the textbox. • **Variable** - The surrogate key data increments with the value from the variable selected.
Max Value	Sets the maximum number the surrogate key can have. If you set it higher than the default, the output column that contains the surrogate key value is set to a DT_I8 value.
Recycle Max Value	If selected, this option rests the value of the surrogate key value to the Seed value when the max value is reached. If unselected, the transform fails with an error stating the max value has been reached.

Database Options

Option	Description
Connection Manager	Select the connection manager that contains the table which stores the surrogate key values.
Table for counter storage	If the table is already created, enter that table name here. If you are creating a new one, enter the desired name for the table here and then select **create table**.
Create Table button	Selecting this button attempts to make a table using the specified name. Selecting the button opens a message that displays the SQL statement used to create the table.

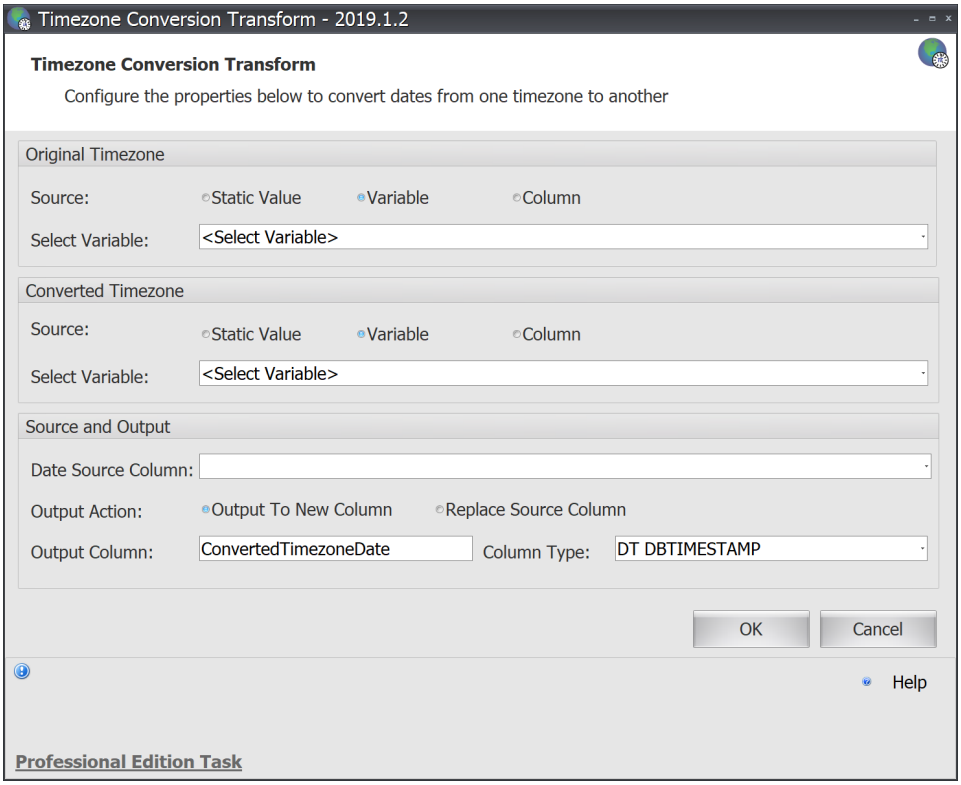
Task Factory Timezone Conversion Transform

Last Modified on 28 September 2020

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Timezone Conversion Transform

Transform Icon	Transform Description
	The **Timezone Conversion Transform** easily converts a date from one Timezone to another using the IANA TZ Database. ℹ **Note:** The **Timezone Conversion Transform** automatically accounts for daylight saving time (DST) adjustments.



Original Timezone / Converted Timezone

Option	Description
Source	Select where the original Timezone data is found. This can be one of three types : • Static Value • Variable • Column
Enter Static Value / Variable, Input Column	Depending on where the source is stored, you use this text box to point to the source.

Source and Output

Option	Description
Data Source Column	The column where the original date is retrieved. **Note:** This column must be a date, DBTimeStamp, or DBTimeStamp2 data type.

Option	Description
Output Action	Where the converted data is sent. There are two options. • **Output To A New Column** - In this mode, a new column name is given in the **Output Column** textbox, and then you select the output data type in the **Column Type** drop down. • **Replace Source Column** - In this mode, select a source data column to replace its value upon conversion.

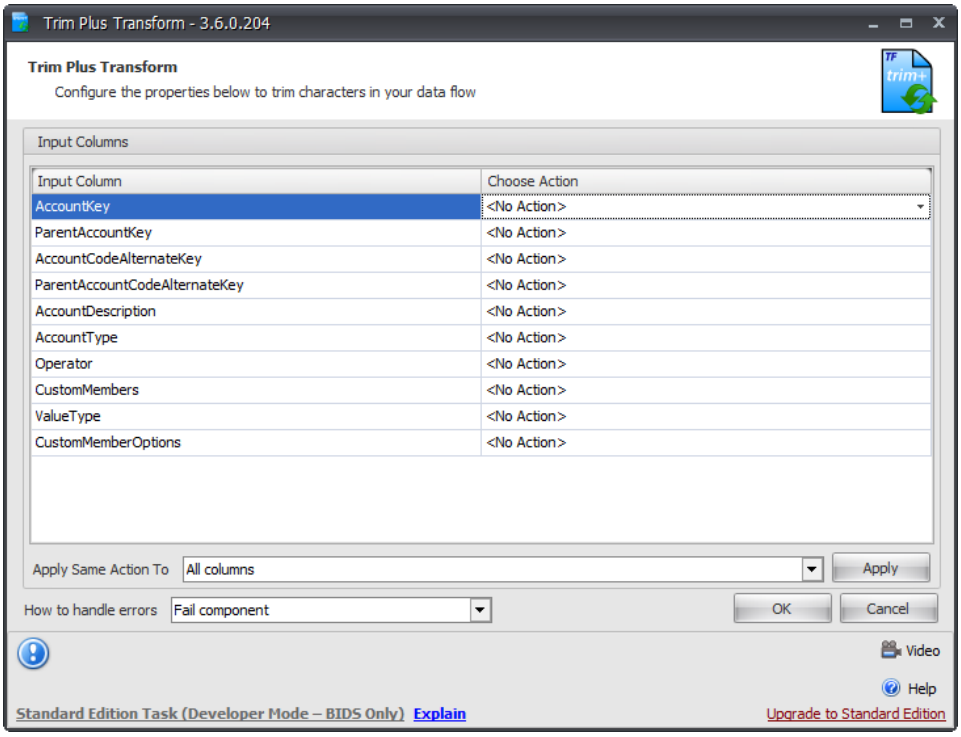
Task Factory Trim Plus Transform

Last Modified on 28 September 2020

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Trim Plus Transform

Transform Icon	Transform Description
trim+	The **Trim Plus Transform** allows the user to trim characters from the beginning and end of source data columns.



Option	Description
Input Column	The selected name of an input column where an action performs.

Option	Description

Choose Action	**Action** - The trim plus transform has six actions that can be used to trim characters from the source data.
	
	• **Trim all leading and trailing white-space characters** - This action trims all the spaces and tabs from the beginning and end of the selected columns data.
	• **Trim all leading white-space characters** - This action trims all the spaces and tabs from the beginning of the selected columns data.
	• **Trim all trailing white-space characters** - This action trims all the spaces and tabs from the end of the selected columns data.
	• **Trim specified characters from start and end** - This action trims all the specified characters in the **Characters to be trimmed** parameter from the beginning and end of the selected columns data.
	  ◦ Parameters
	  ▪ **Characters to be trimmed** - Specify characters that you want to trim form the start or end.
	• **Trim specified characters from start** - This action trims all the specified characters in the **Characters to be trimmed** parameter from the beginning of the selected columns data.
	  ◦ Parameters
	  ▪ **Characters to be trimmed** - Specify characters which you want to trim from the start or end.
	• **Trim specified characters from end** - This action trims all the specified characters in the **Characters to be trimmed** parameter from the end of the selected columns data.
	  ◦ Parameters
	  ▪ **Characters to be trimmed** - Specify characters which you want to trim from the start or end.
	  ▪ **Output Action** - There are two options for the output action.
	
	Output Action - There are two options for the output action:
	
	• **Replace the column data with fixed data** - This option replaces the existing data with fixed data.
	• **Output the fixed data to a new output column** - The new output column name is **inputcolumn_output**, if the source column is **FirstName** the output column is **FirstName_output**.

Additional Information: See the Task Factory Error Row Handling article for more information about this functionality.

Task Factory Unpack Data Transform

Last Modified on 30 September 2020

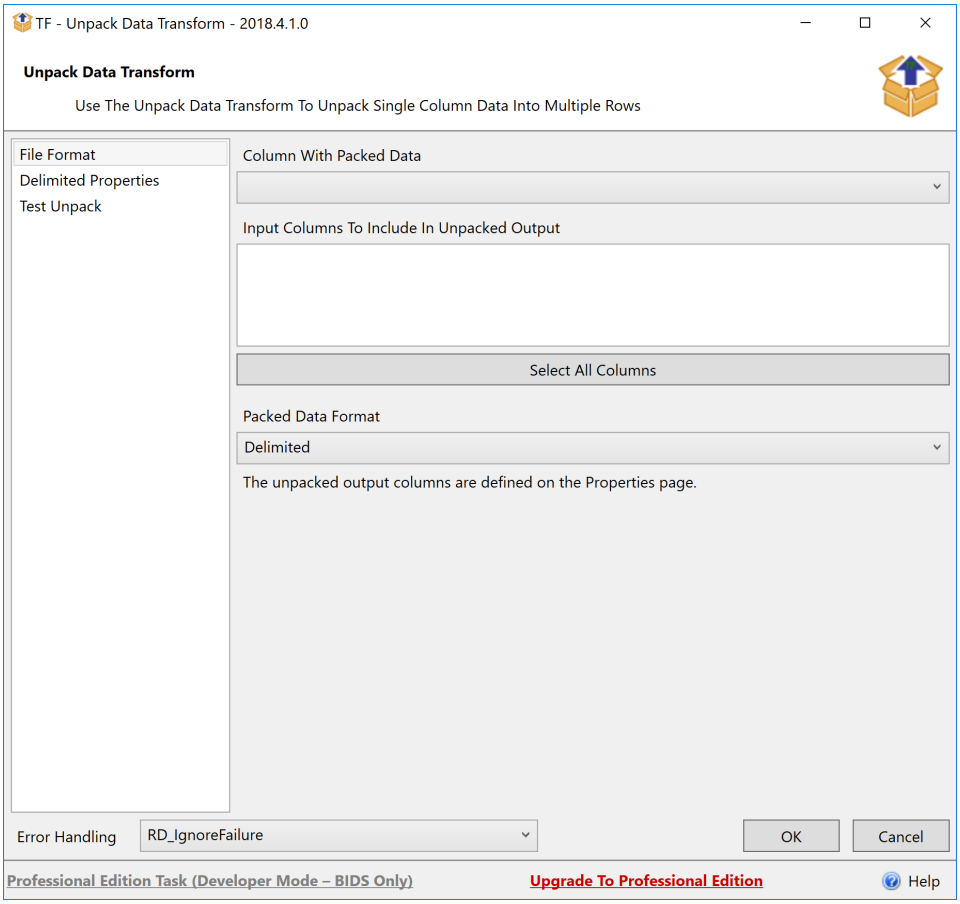
Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Unpack Data Transform

Important: Unpack Data is available for SQL versions 2012 and higher.

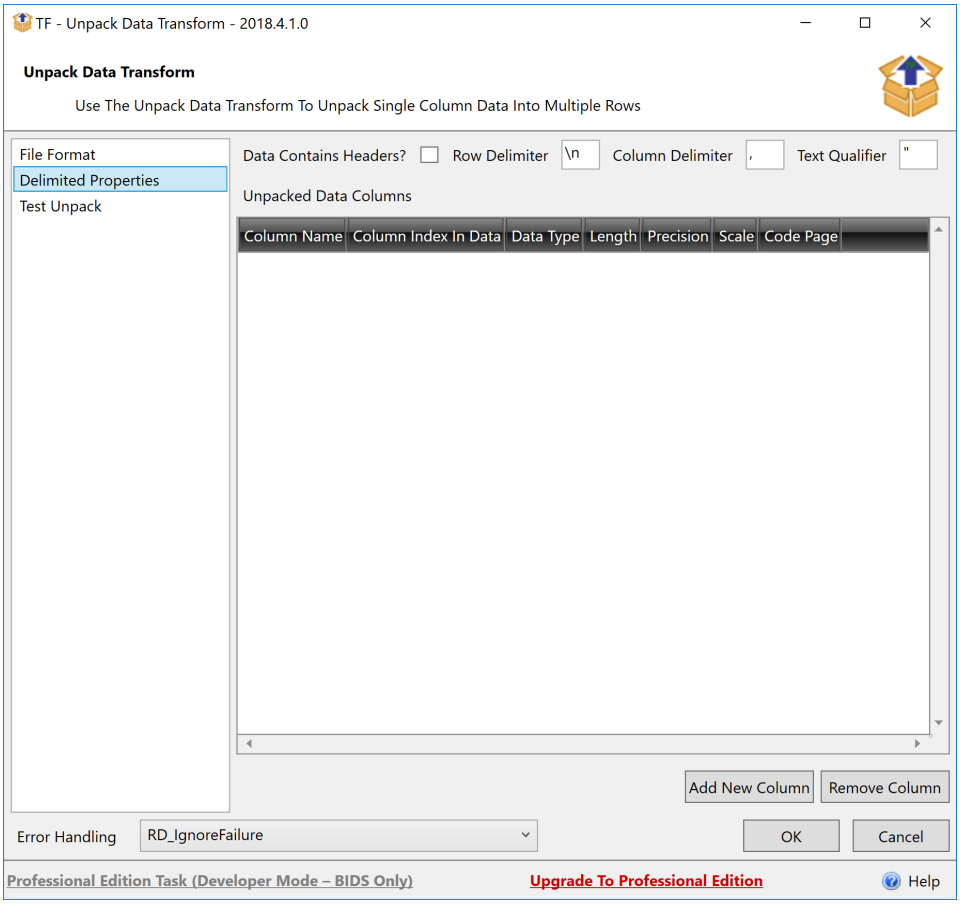
As of Feb 2, 2018, DT_EMPTY, DT_NULL and all datatypes that included BYREF have been removed. Any package configured to use these datatypes will error and have to be reconfigured.

Transform Icon	Transform Description
	The **Unpack Data Transform** is used to output delimited, XML, or Json data contained within a single column source.



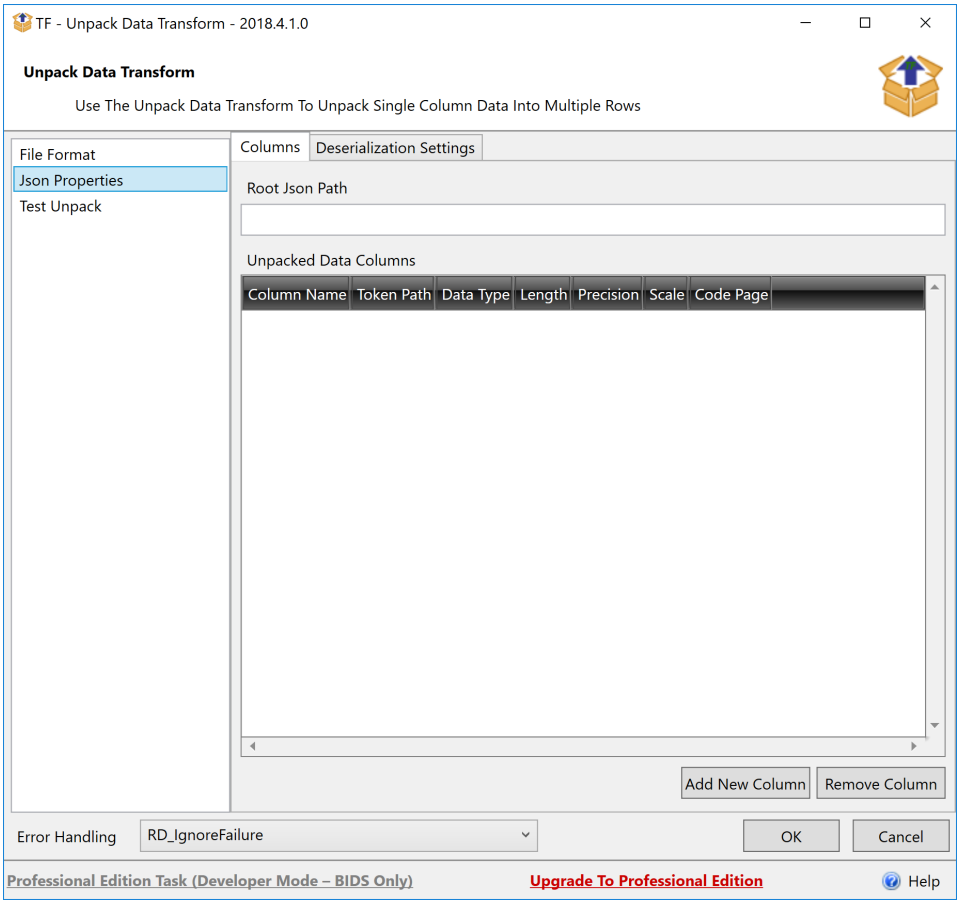
Option	Description
Column With Packed Data	Selects the column that stores the delimited or XML data to be unpacked.
Input Columns To Include In Unpacked Output	Selects other columns to be included in the Unpacked Output (non-selected columns continue down the standard output.)
Packed Data Format	Selects the format of the packed data . Data must be one of the following: • Delimited • JSON • XML

Delimited Configuration



Option	Description
Data Contains Headers?	Similar to the native Flat File Source, this selection identifies the first row as containing column headers.
Row Delimiter	Identifies a character or carriage return **(\n)** to signify a new row.
Column Delimiter	Identifies the character used to separate values for the different columns such as a comma.
Text Qualifier	Identifies the character used to wrap values such as quotation marks.
Unpacked Data Columns	Users can create, remove, and configure the name, index (zero-based), data type, length, precision, and scale of the columns being extracted or unpacked.

Json Configuration



Option	Description
Root Json Path	Json queries can return multiple levels, therefore, this field Identifies the root to be used.
Unpacked Data Columns	Users can create, remove, and configure the name, index (zero-based), data type, length, precision, and scale of the columns being extracted or unpacked.

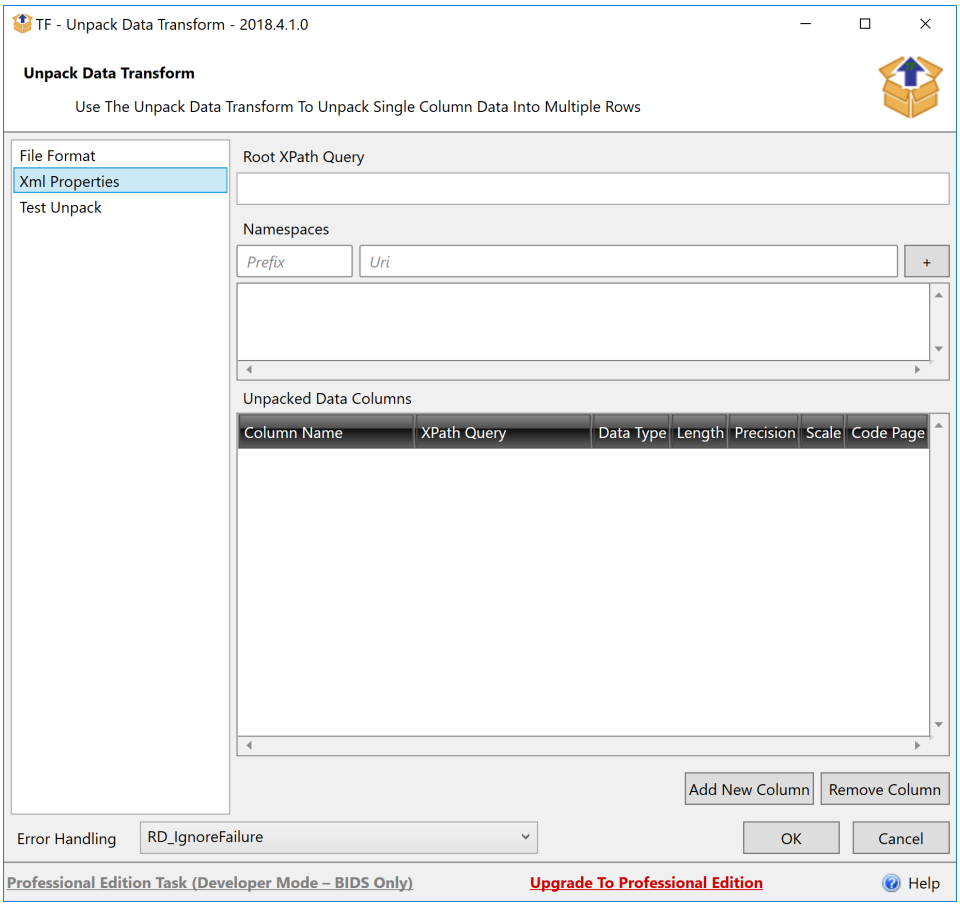
Using Wildcards (*)

The Unpack Data Transform allows for wildcards when arrays are returned.

Example:
Using the Json Results below, the Root Json Path would be **results[0].appInventory[*]**. This information tells the component to use the array found at position **0**. Because some child arrays do not have an object name, a **wildcard** or ***** can be used to return all child array objects.

```
{
  "results": [
    {
      "appInventory": [
        {
          "name": "Sample name 1",
          "identifier": "123",
        },
        {
          "name": "Sample name 2",
          "identifier": "456",
        },
        {
          "name": "Sample name 3",
          "identifier": "789",
        }
                ]
        }
    ]
}
```

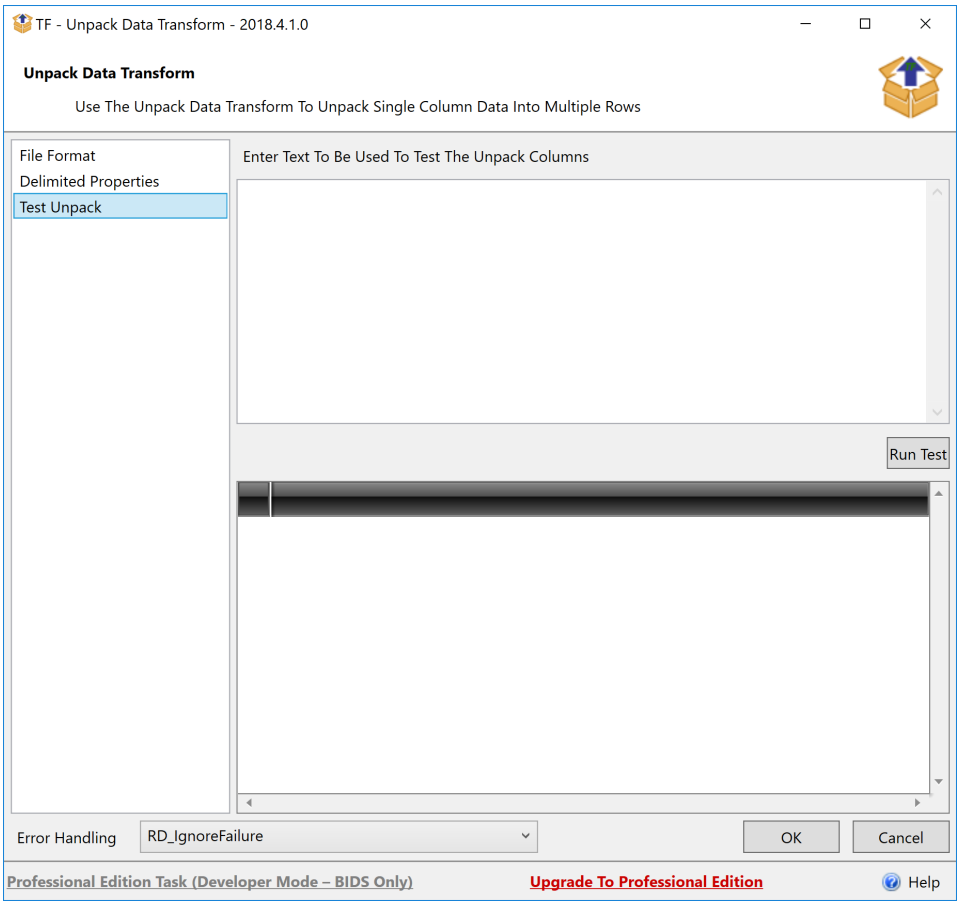
XML Configuration



Option	Description
Root XPath Query	Xml queries can return multiple levels, therefore, this field Identifies the root to be used.

Option	Description
Namespaces:	• **Prefix** - Identifies the prefix used in an xml namespace. ◦ Example: In the element the prefix is url. • **Uri** - The Url or address identified in the namespace. In the example above, http://www.site.com is the uri that should be configured.
Unpacked Data Columns	Users can create, remove, and configure the name, index (zero-based), data type, length, precision, and scale of the columns being extracted or unpacked.

Test Unpack Columns



Option	Description
Enter Text To Be Used To Test The Unpack Columns	Users can copy and paste a row of packed data into this field to test the component's configuration and output.
Run Test	Executes a test based on the component's configuration and data pasted into the window above. Results display in the window below.

Additional Information: See the Task Factory Error Row Handling article for more information about

this functionality.

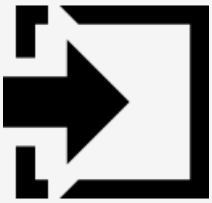
Task Factory Update Batch Transform

Last Modified on 25 October 2021

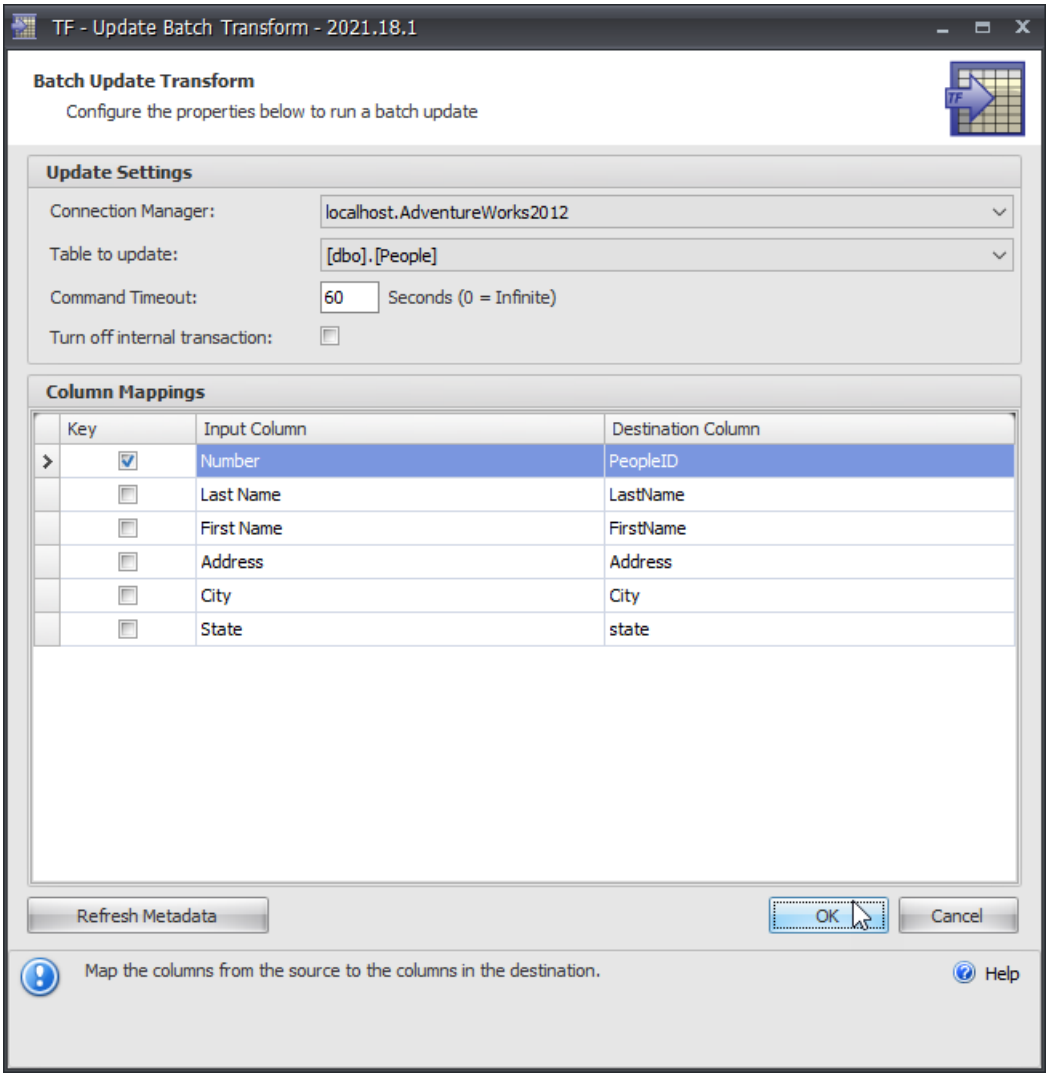
Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Note: Please see the Azure page for more information before using this component with Azure databases.

Update Batch Transform

Transform Icon	Transform Description
	The **Update Batch Transform** allows you to quickly update batches of data.

General



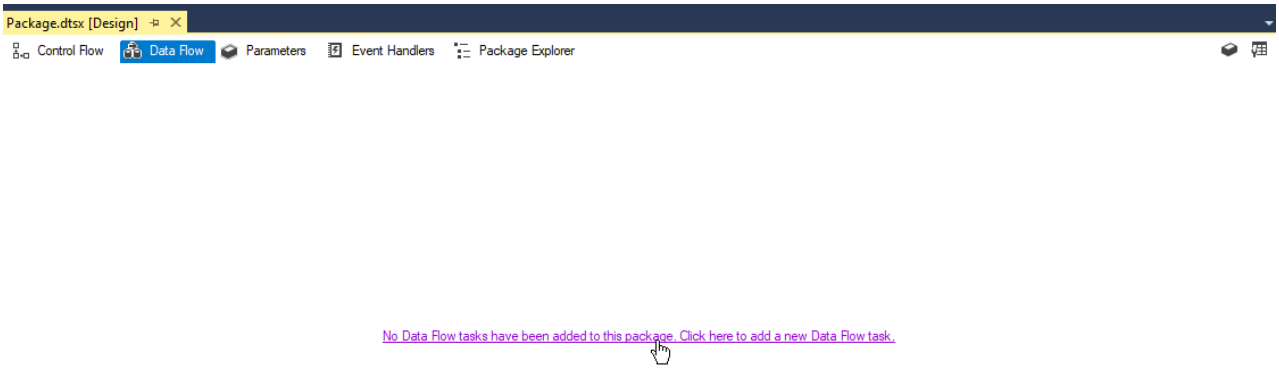
Option	Description
Batch Update Information	• **Update Connection Manager** - Select an existing connection manager or create a new one. • **Table to update** - Once the connection manager is set, select the table you wish to update. • **Command Timeout** - The number of seconds that the command waits before timing out during execution.
Column Mappings	Here you map the columns from the source to the columns in the selected destination table (Table to update). If the column names from the source are the same as the destination, the mappings occur automatically.
Turn off internal transaction?	This option turns off internal transaction. **Note:** Only use this for performance reasons.

Note: The Task Factory Update Batch Transform requires an input source and and ADO.NET Connection Manager destination.

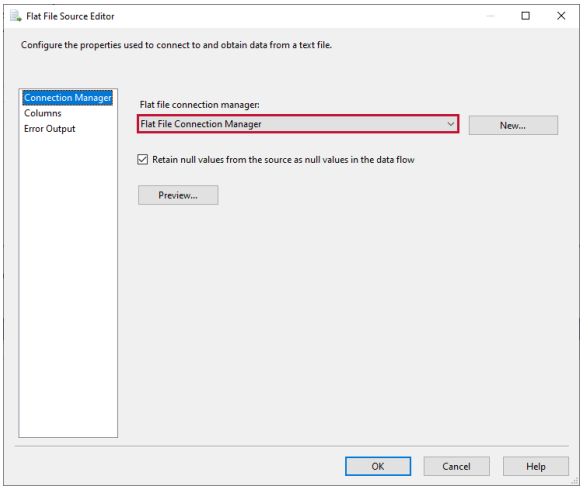
Example

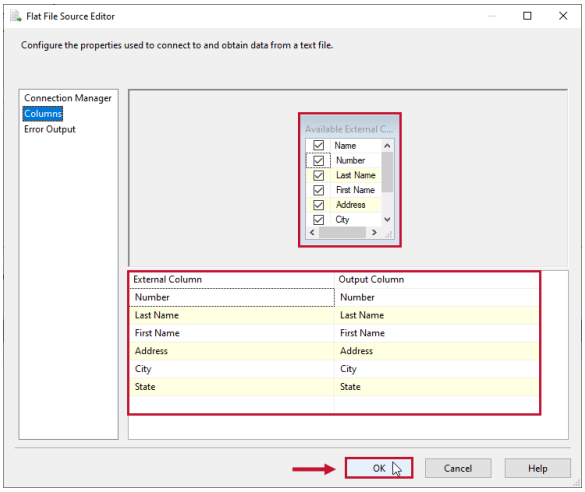
The following example displays using the Update Batch Transform in an existing package.

1. Create a new, or open the desired SSIS package.
2. Add a **Data Flow task** into your Control Flow, and then enter the Data Flow task workspace.



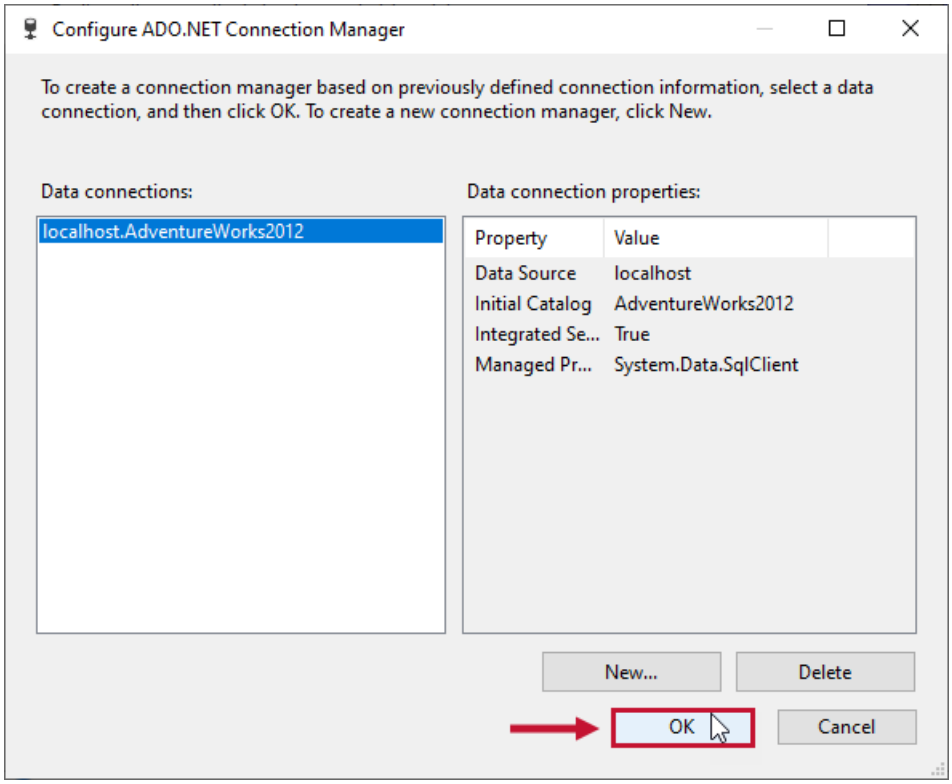
3. Add a source input file into your Data Flow. In this example, we have configured a Flat File Source, with a Flat File Connection Manager that connects to a local flat file with values separated by the "," delimiter.



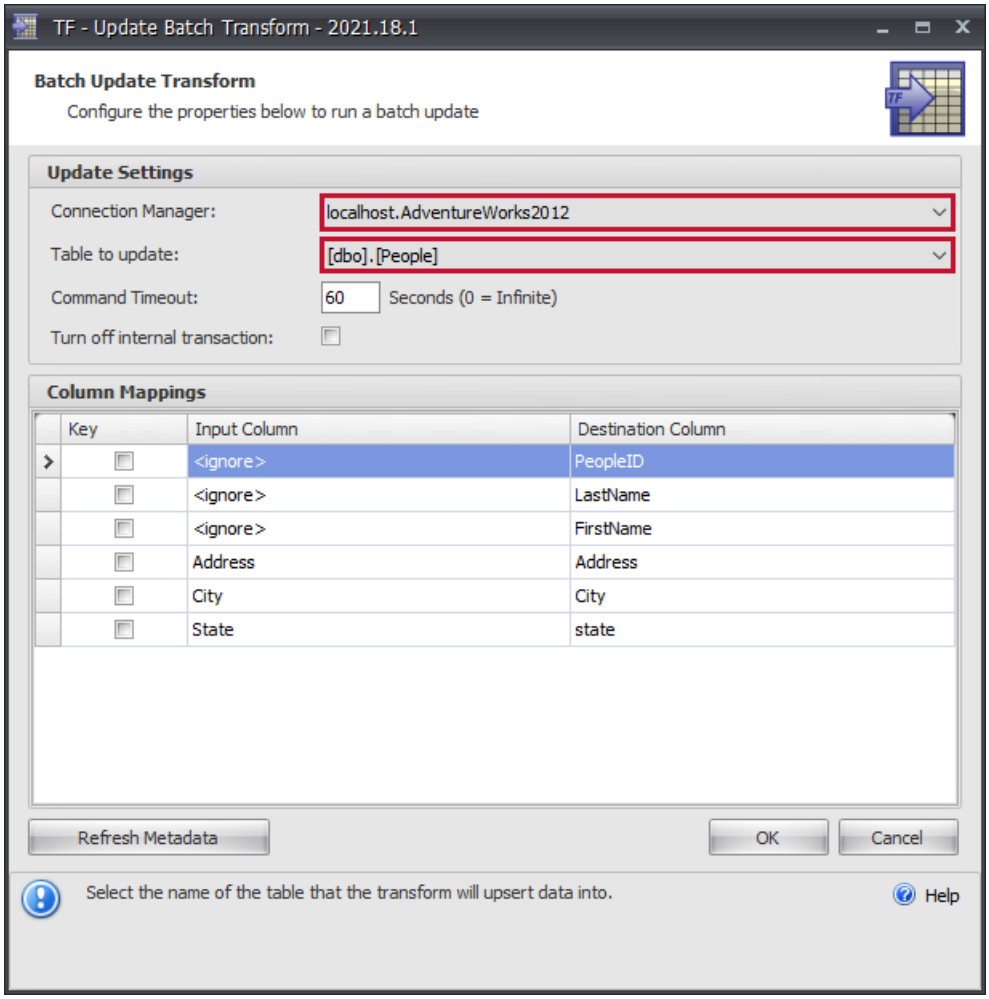


4. Add the Task Factory Update Batch Transform component to your Data Flow:

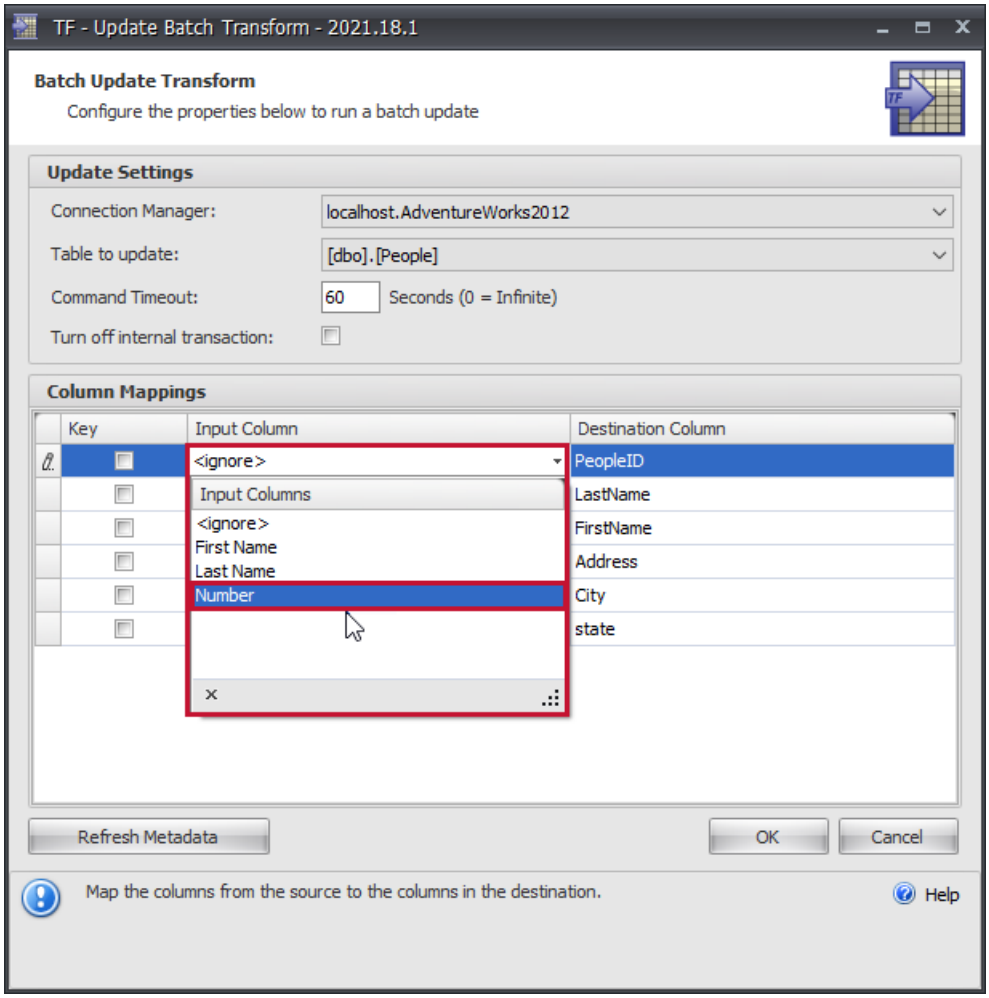
    1. Select an existing Connection Manager, or configure a new Connection Manager for the destination connection.



2. Select the table that the Update Batch Transform will upsert data into.

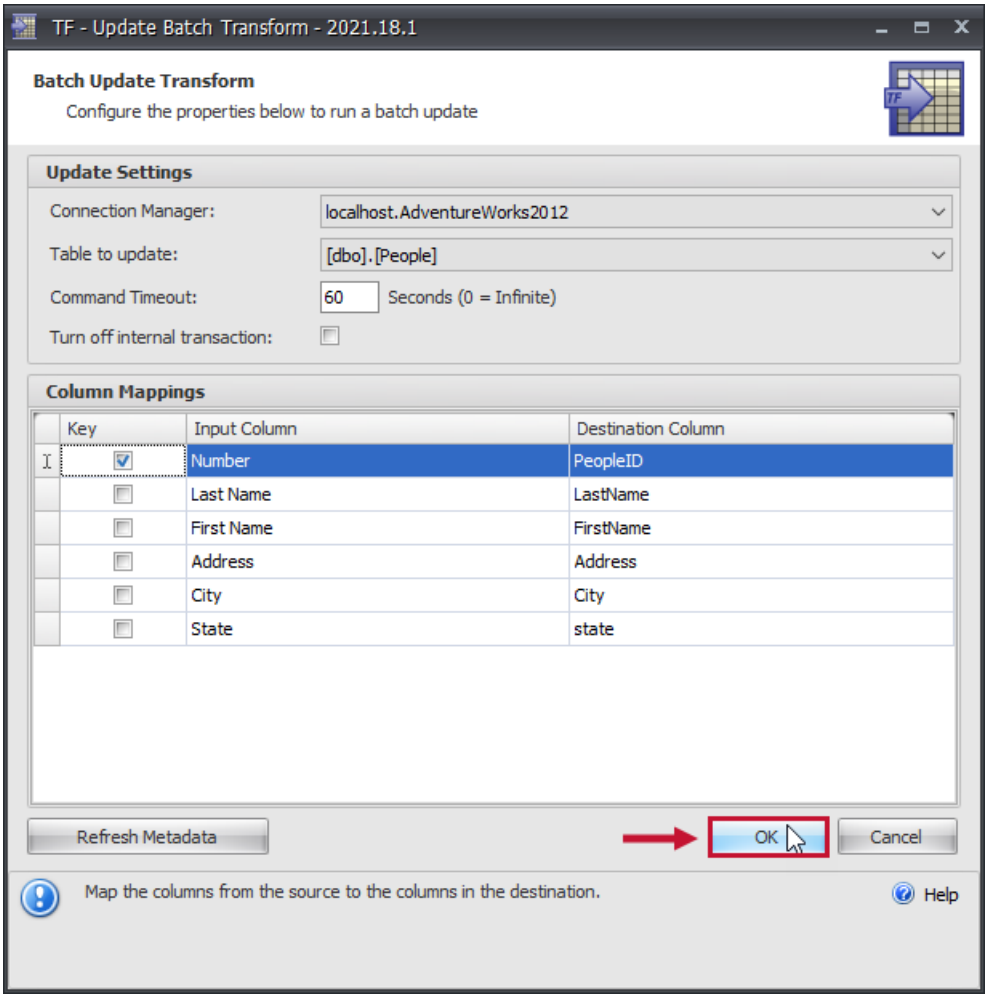


3. Match the input columns to the proper destination columns.

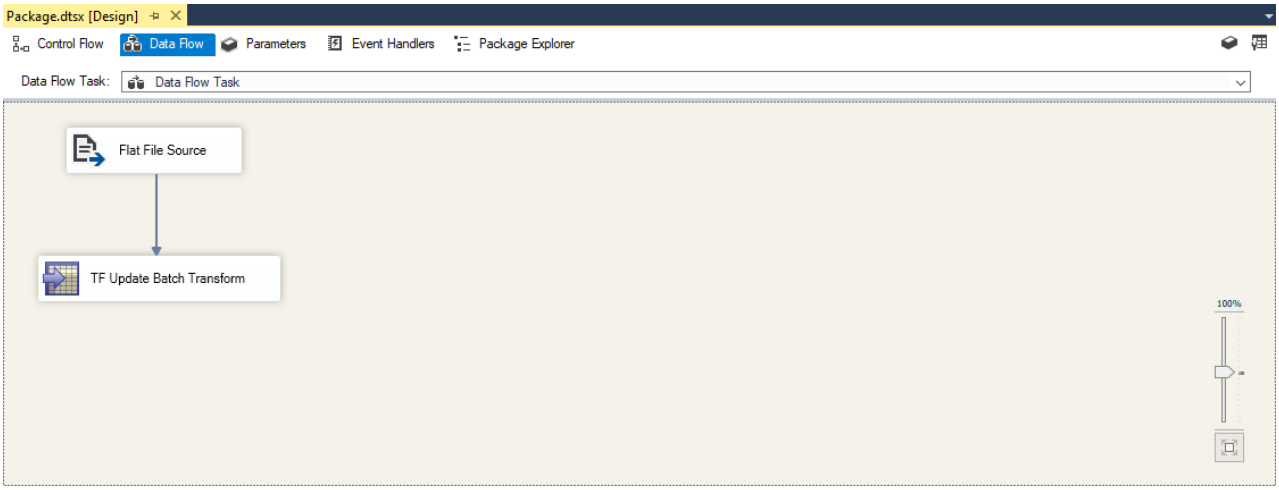


4. Set the **key** column.

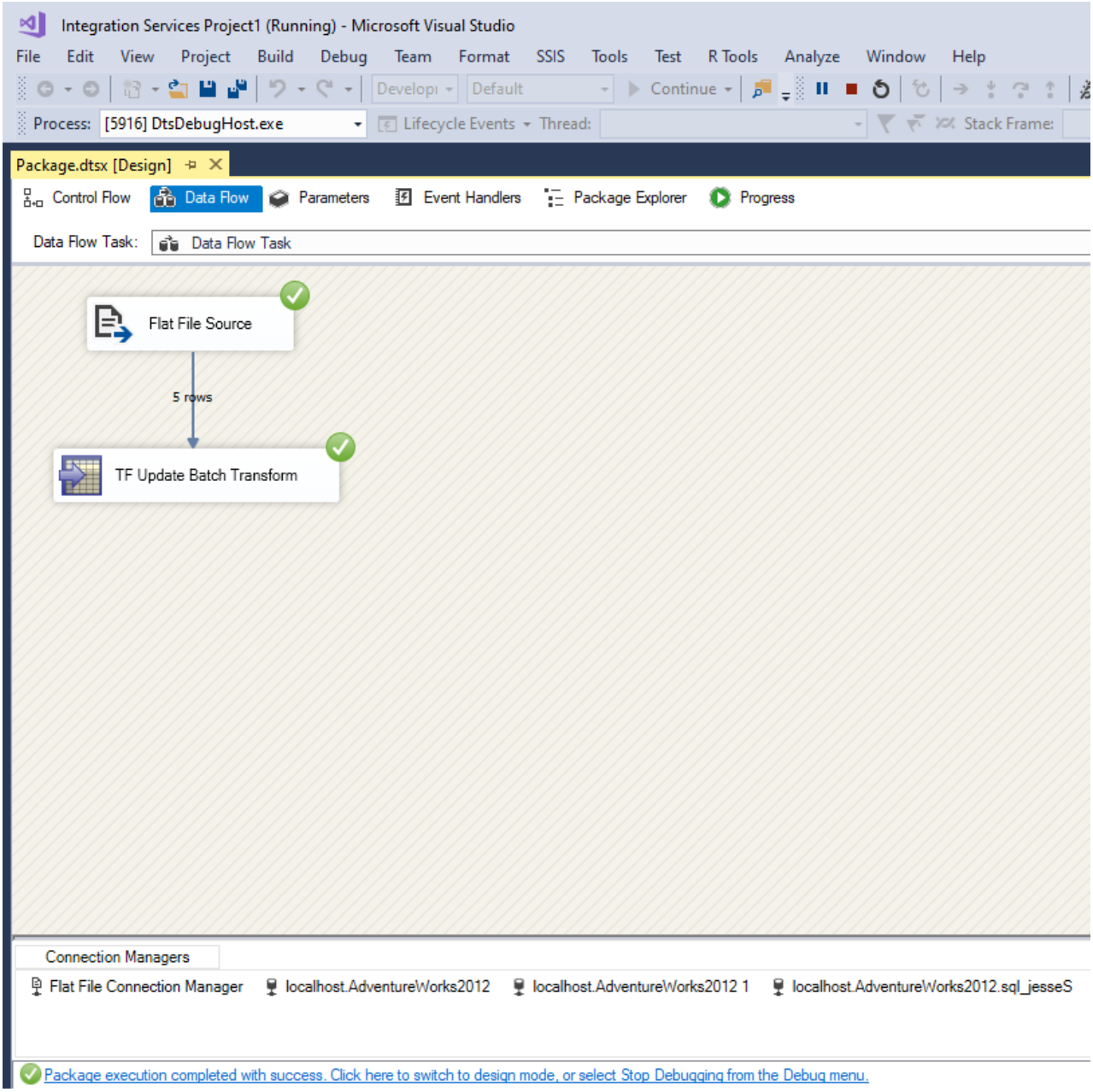
5. Select **Ok** to complete your configuration changes.



6. Map your **Flat File Source** to the **Task Factory Update Batch Transform**.



7. Execute the Package.



🏆 **Success:** The Update Batch Transform successfully executed! The following is sample output from the selected table used in this example:

100 %

Results | Messages

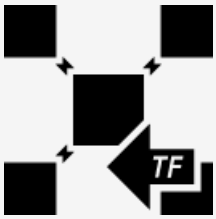
	PeopleID	LastName	FirstName	Address	City	state
1	1	Markel	Kyle	101 Fake Street	Los Angeles	California
2	2	Bob	Potts	102 Fake Stree	Los Angeles	California
3	3	Wilks	John	103 Fake Street	Los Angeles	California
4	4	Johnson	Bob	104 Fake Ave	Greenville	South Dakota
5	5	Watson	Timothy	101 Fake Ave	Charlotte	North Dakota

Task Factory Fact Table Destination

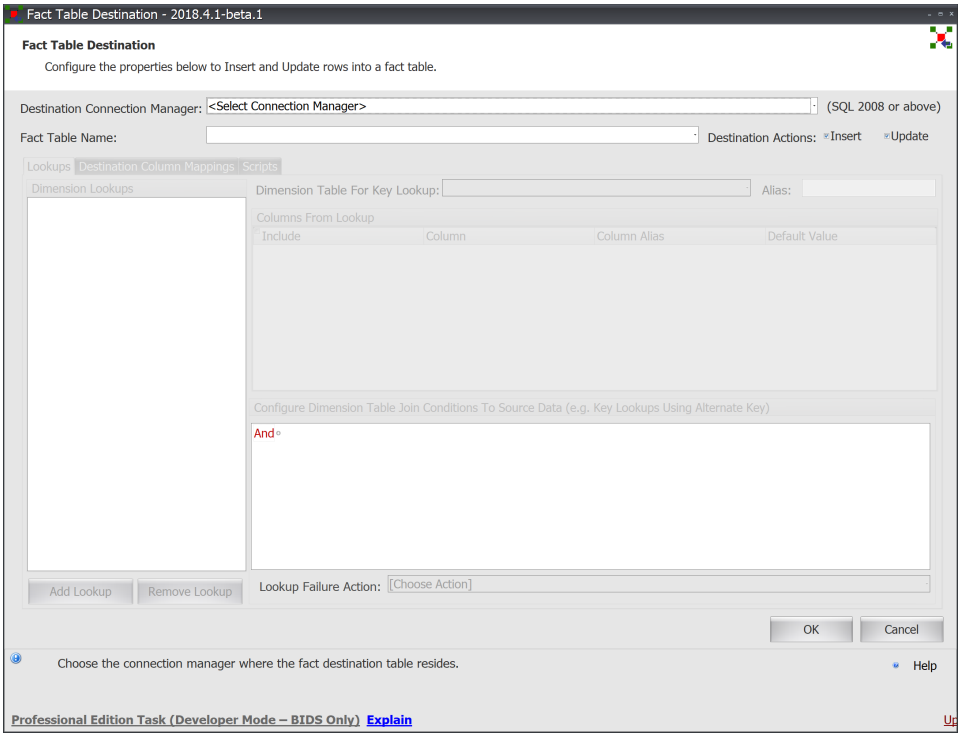
Last Modified on 30 September 2020

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Fact Table Destination

Destination Icon	Destination Description
	The **Fact Table Destination** gives developers the ability to perform all of the lookups required to insert or update a fact table.

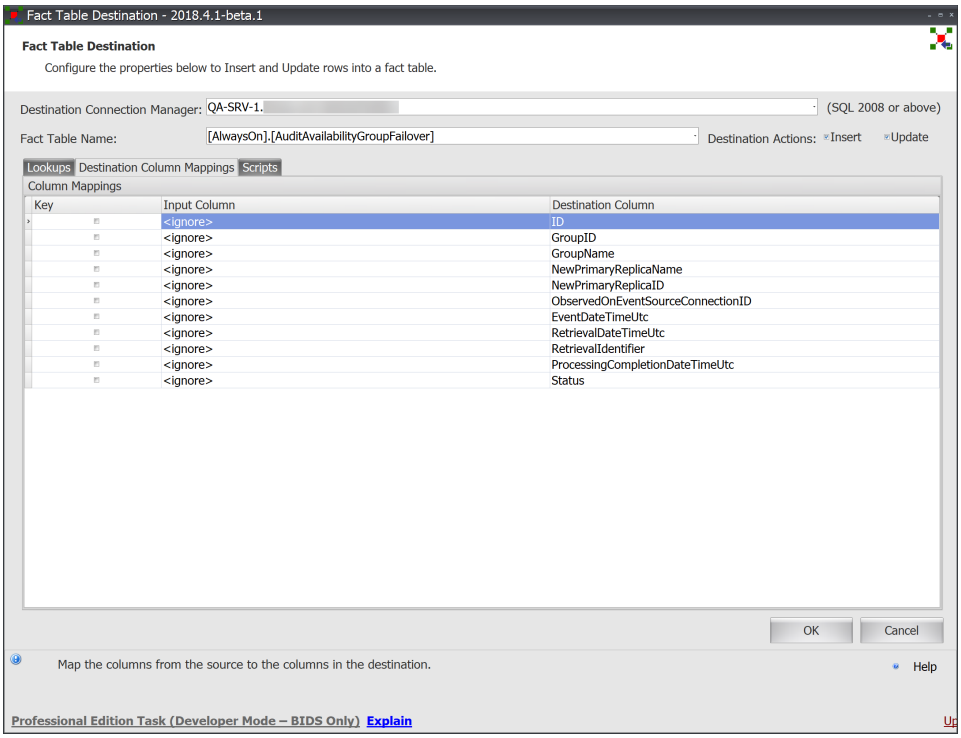
Options



Option	Description
Destination Connection Manager	Creates or selects a connection to the destination database (must be SQL Server 2008 or above).
Fact Table Name	Selects the destination fact table.
Destination Actions	• **Insert** - Users select this option to insert rows to the fact table. • **Update** - Users select this option to update rows in the fact table.
Dimension Lookups	• **Add Lookup** - Creates a new lookup ready for configuration. • **Remove Lookup** - Removes the selected lookup.
Dimension Table For Key Lookup	Selects the dimension table for each lookup.
Alias	Users can assign an alias for the selected dimension table. **Note:** This should be used for performing multiple lookups on the same dimension.
Columns From Lookup	• **Include** - Selects the column(s) to be used for destination column mapping. • **Column** - The name of the column. • **Column Alias** - Users can create a column alias. • **Default Value** - Users can assign a value to replace NULLs.

Option	Description
Configure Dimension Table Join Conditions To Source Data	• **And** - Select the expand button to create join condition(s). All lookups require a condition to be configured.
Lookup Failure Action (Set to each lookup)	• **Set To Default** - NULLs in the lookup are replaced with the default value defined in the Columns From Lookup. • **Fail** - NULLs in the lookup causes the component to fail execution.

Destination Column Mappings



Option	Description
Key	Selects the key(s) needed when performing updates.
Input Column	The user selected input column to be mapped to the destination.
Destination Column	Name of the destination column.

Scripts

Users can write custom TSQL queries to be performed before or after each buffer.

Fact Table Destination - 2018.4.1-beta.1

Fact Table Destination

Configure the properties below to Insert and Update rows into a fact table.

Destination Connection Manager: QA-SRV-1. (SQL 2008 or above)

Fact Table Name: [AlwaysOn].[AuditAvailabilityGroupFailover] Destination Actions: Insert Update

Lookups | Destination Column Mappings | Scripts

Before Script

Use to define a SQL script to execute before each buffer of data is sent to the destination

After Script

Use to define a SQL script to execute after each buffer of data is sent to the destination

OK Cancel

Choose the connection manager where the fact destination table resides. Help

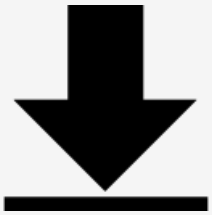
Professional Edition Task (Developer Mode – BIDS Only) Explain Up

Task Factory Terminator Destination

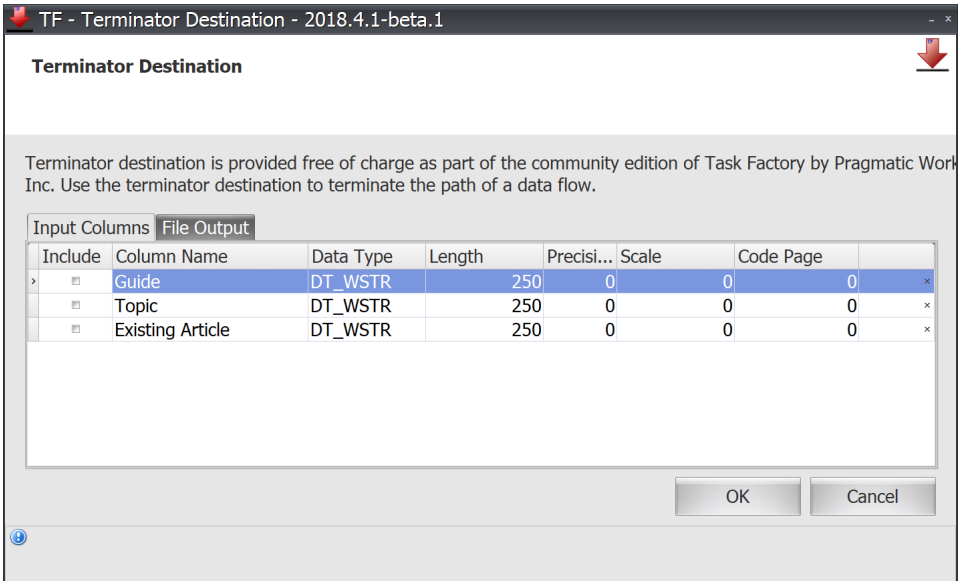
Last Modified on 30 September 2020

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Terminator Destination

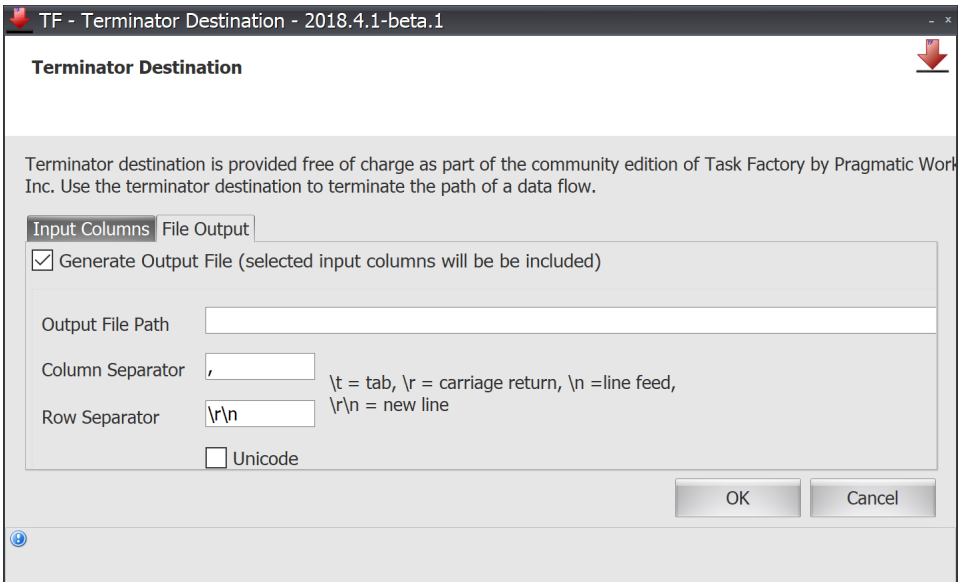
Destination Icon	Destination Description
	The **Terminator Destination** is used to quickly add an end to a data flow. It requires zero configuration. Simply add an output to the terminator destination. To use the advanced features to generate an output file, you can open the Terminator Destination and define how you want the output file to be defined.

Input Columns



The **Terminator Destination** is also used to define the end of a **Data Flow Nugget**. Select the include box to define an output column in the **Data Flow Nugget Transform**.

File Output



Option	Description
Generate Output File (selected input columns are included)	Select this to generate an output file.
Output File Path	Select the path to output the file to.
Column Separator	Choose the character to use to separate columns.
Row Separator	Choose the escaped character to separate rows.

Task Factory Upsert Destination

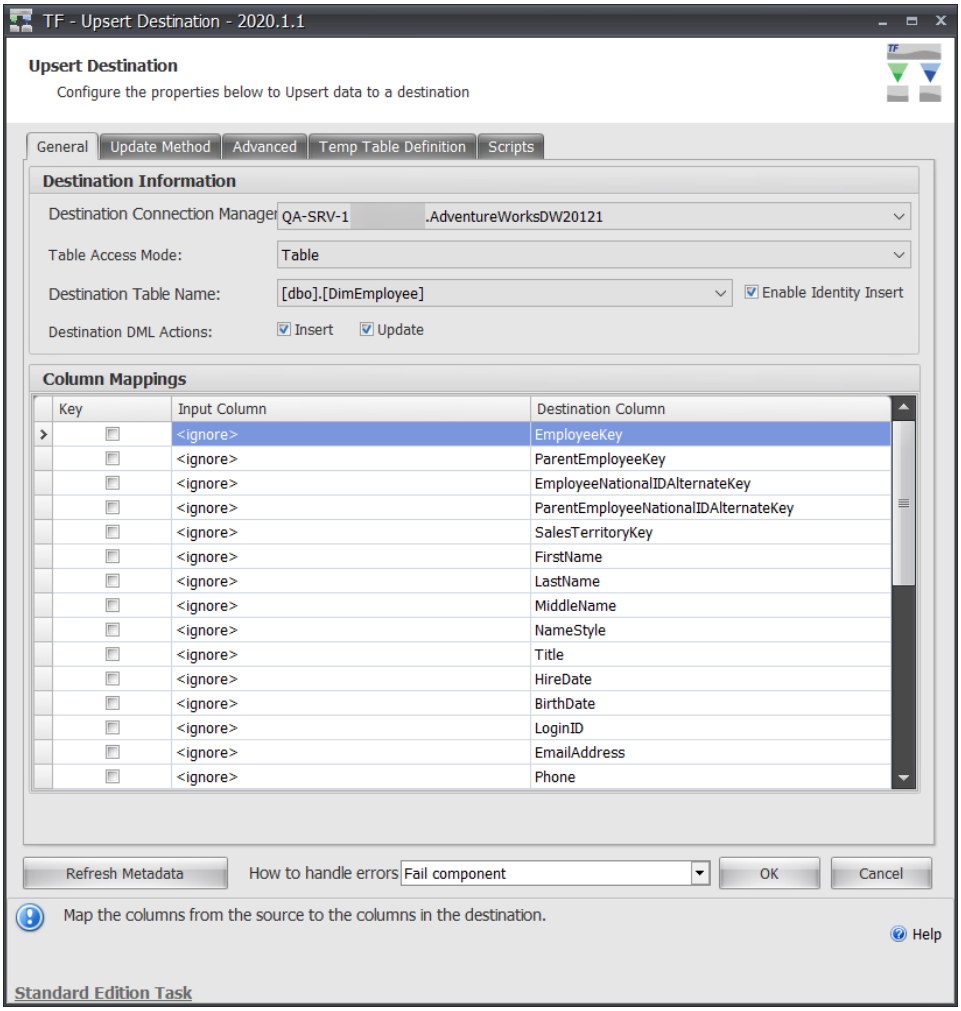
Last Modified on 28 September 2020

📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Upsert Destination

Destination Icon	Destination Description
	The **Upsert Destination** is used to update and insert data into a SQL database table. Use **Upsert Destination** to insert new rows and perform updates at the same time.

General

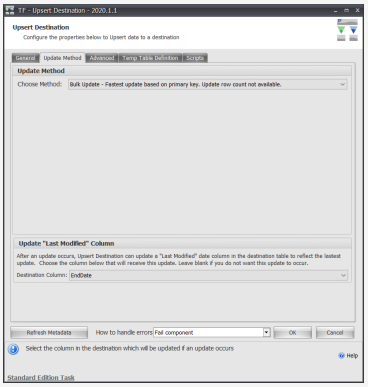


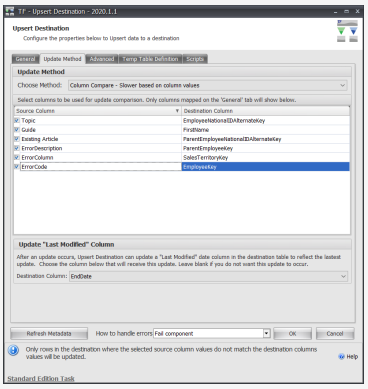
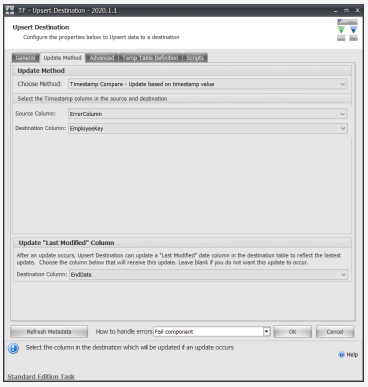
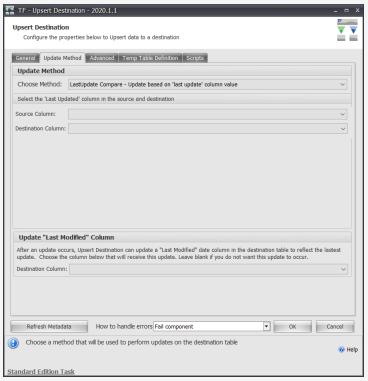
Option	Description
Destination Connection Manager	Select or create the connection manager that upserts data into the destination.
Table Access Mode	Table access mode allows you to define how the destination table name is selected. There are two modes: • **Table** - This mode allows you to choose the table in the **Destination Table Name** drop down. • **TableFromVariable** - This mode allows you to choose a variable that contains the destination table name in the **Choose Table Variable** drop down, only visible when **TableFromVariable** is selected.
Destination Table Name / Choose Table Variable	Choose either the destination table or the table variable.
Enable Identity Insert	This option is automatically selected when Upsert detects that the destination table contains an identity column. If the selected source for Upsert does not contain a column that inserts data into the identity column, deselect this option. If the source contains at least some rows that insert into the identity column, leave it selected.

Option	Description
Upsert Method	The upsert method defines how the upsert functions when executing the DML action. There are two modes: • **The standard upsert method without a flag column** - This mode inserts and updates records in the destination based on the Key(s) selected in the column mappings. If the row exists, it gets updated, if it does not exist, the row is inserted. • **Row Flag Compare** - This mode updates, inserts, deletes and rejects rows based on the settings in the **Row Flag Column** drop down in the **Row Flag Column Settings** tab. The tab is only visible if this upsert method is selected.
Destination DML Actions	• Insert • Update
Refresh Metadata	Refreshes the destination table's metadata.
How to handle errors	Select the **Redirect row to error output** option to redirect rows with errors to an error output. Additionally, you have the option to ignore the errors and continue execution or to fail the component once an error is encountered.

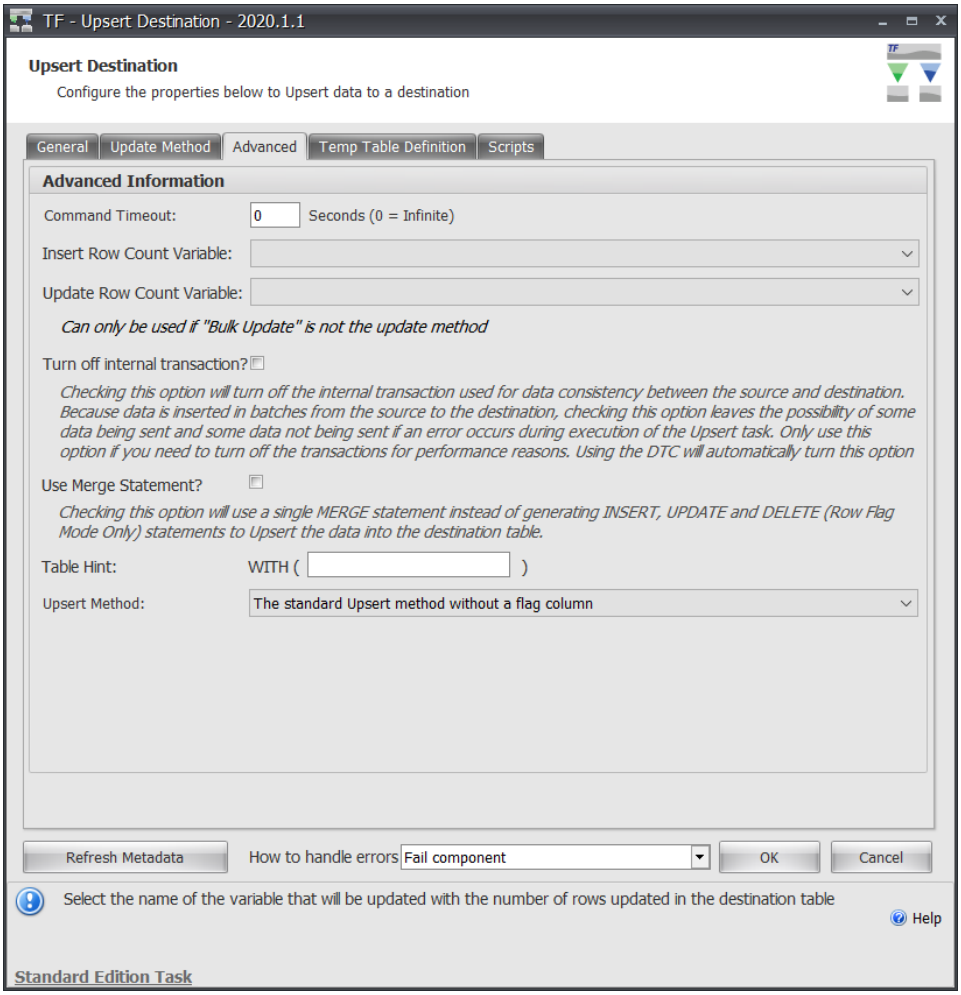
Update Method

The update method tab allows you to control if updates occur on the selected destination table by giving you four options:

Update Method	Description	Image
Bulk Update	**Bulk Update** is the default update method. In this method the destination is updated with whatever data is contained in the source as long as the keys selected in the columns mappings from the general tab match in the source and destination. It does not take into account whether data has changed or not.	

Update Method	Description	Image
Column Compare	**Column Compare** can be used to compare whether data in the source matches what is in the destination. If the data does not match in the selected columns, the update occurs. Otherwise the update is not performed. Keep in mind that this method slows down the upsert process because of the comparison process. **ℹ Note: Column Compare** uses **GetDate()** to generate the timestamp, and updates the rows per buffer/batch. For each buffer/batch, the row updates to the value of **GetDate()** at the time the insert/update/merge statement executes.	
Timestamp Compare	**Timestamp Compare** compares the value of a timestamp column from the source to a timestamp column in the destination.	
LastUpdate Compare	**LastUpdate Compare** compares a date column from the source to a date column in the destination. If the dates do not match, the row updates.	

Advanced Tab

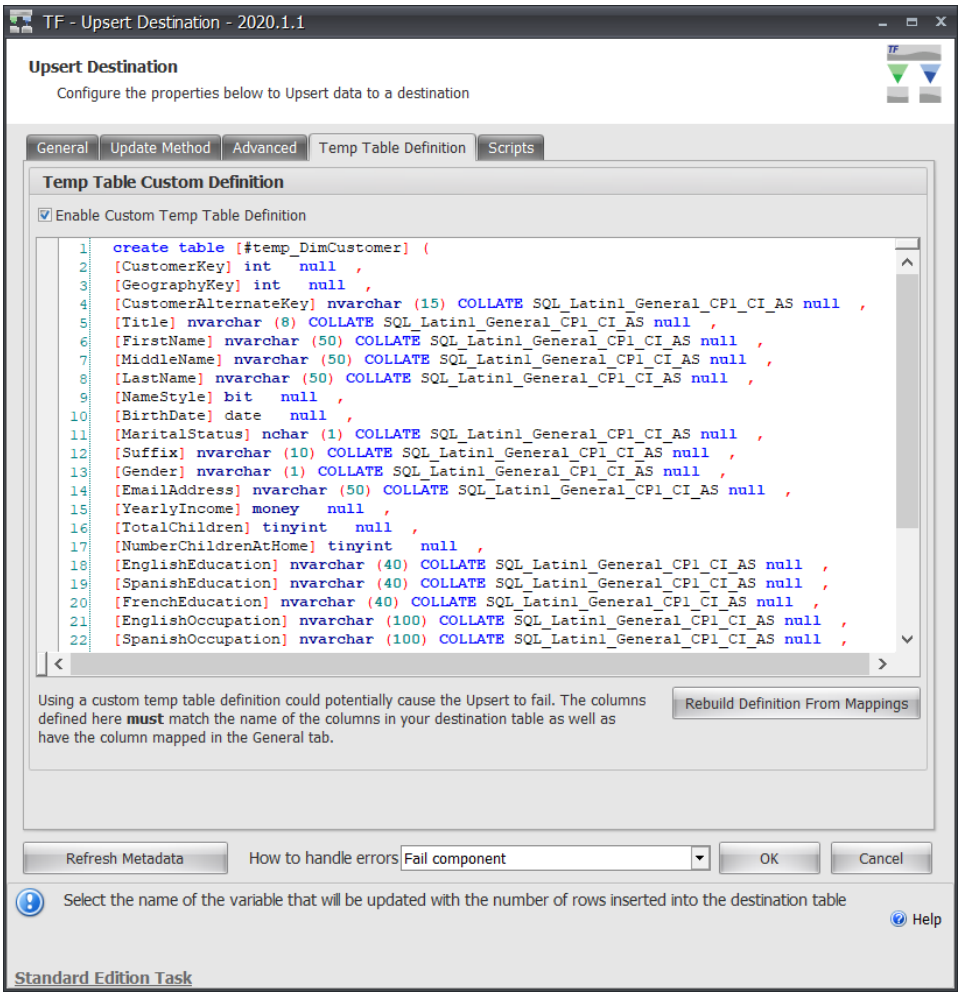


Option	Description
Command Timeout	Determines the time in seconds before a timeout occurs during the insert and updates to the destination table.
Insert / Update Row Count Variable	To assign the number of rows inserted / update into the destination table during the upsert process a variable can be assigned to hold those values by choosing a variable from the drop down box for each count.
Turn off internal transaction	Only use for performance reasons.
Use Merge Statement	Uses the **MERGE** statement instead of generating **INSERT**, **UPDATE** and **DELETE** statements to upsert the data into the destination table.
Table Hint	When the **Use Merge Statement** property is not in use, define table hints in this field. For example: memory optimized tables require SNAPSHOT as a table hint.

Note: When upserting into memory optimized tables:

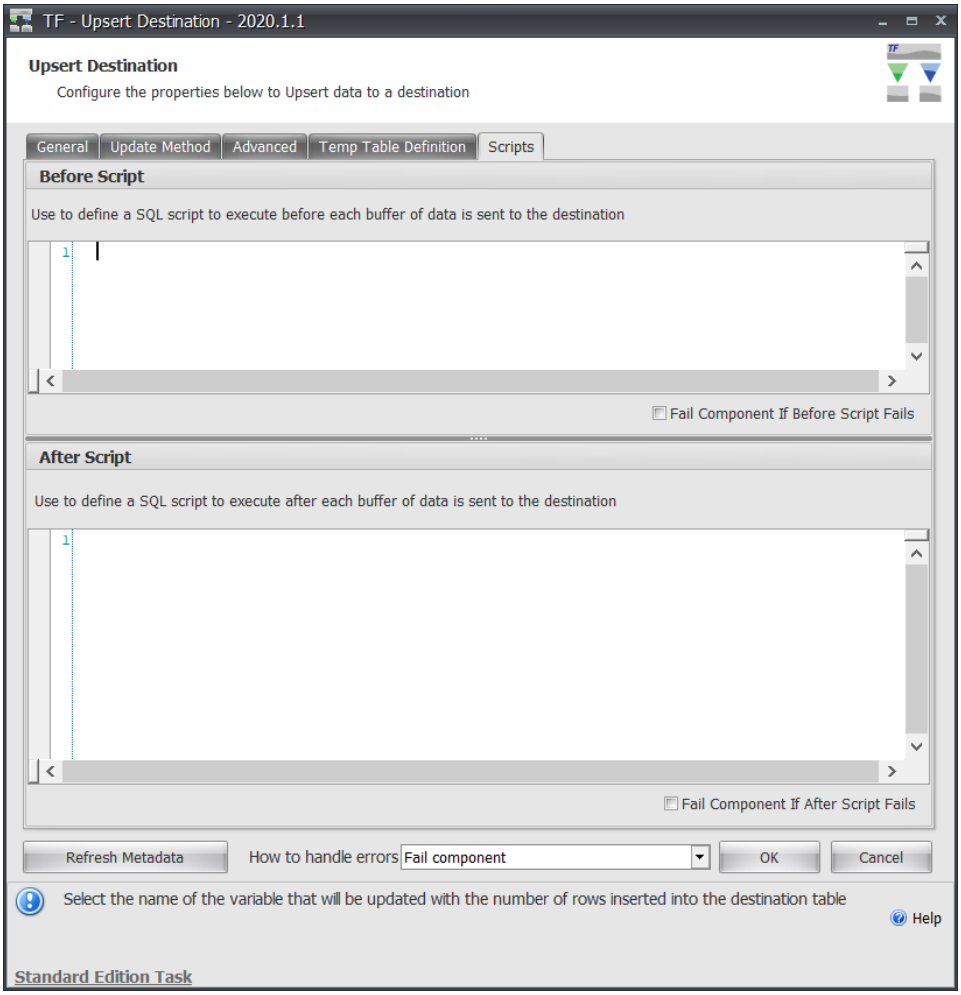
- Table hints cannot be in use when **Turn off internal transactions** property is set to true.
- Memory optimized tables do not allow use of the **merge** statement option.

Temp Table Definition



You can define your own custom temporary table.

Scripts



You can write custom T-SQL queries to be performed before or after each buffer.

> **Note:** When redirecting errors to the error output:
>
> - If a failure occurs during the regular upserting of rows through bulk copy, the component begins using a *row by row* mode. When this occurs, each row from the source is inserted/updated for the current buffer of rows; usually around 10,000. If any row fails to insert/update then that row is pushed to the error output.
> - If a failure occurs due to duplicate rows (based on the keys selected in the column mappings) in a single buffer, the first duplicated row is inserted, with all subsequent rows processed as updates to the first row. When this happens, you rarely see rows being sent to the error output unless a different error occurs on the individual insert/update. Non-sorted rows can present data quality issues.

Samples

See the taskfactory-samples repository on GitHub for the following Upsert Destination samples:

- **1_SetupUpsertDemoDBs.dtsx** - Use to set up the Upsert samples
- **UpsertDestination_BasicSetup.dtsx** - A simple upsert configuration
- **UpsertDestination_ColumnCompareWithAuditVariables.dtsx** - Upsert configured with comparison columns and auditing variables to show the number of records inserted / updates
- **UpsertDestination_ErrorOutput.dtsx** - Upsert configured with error output
- **UpsertDestination_InsertOnly.dtsx** - Upsert configured for insert only
- **UpsertDestination_UpdateOnly.dtsx** - Upsert configured for update only

Oracle Upsert Destination

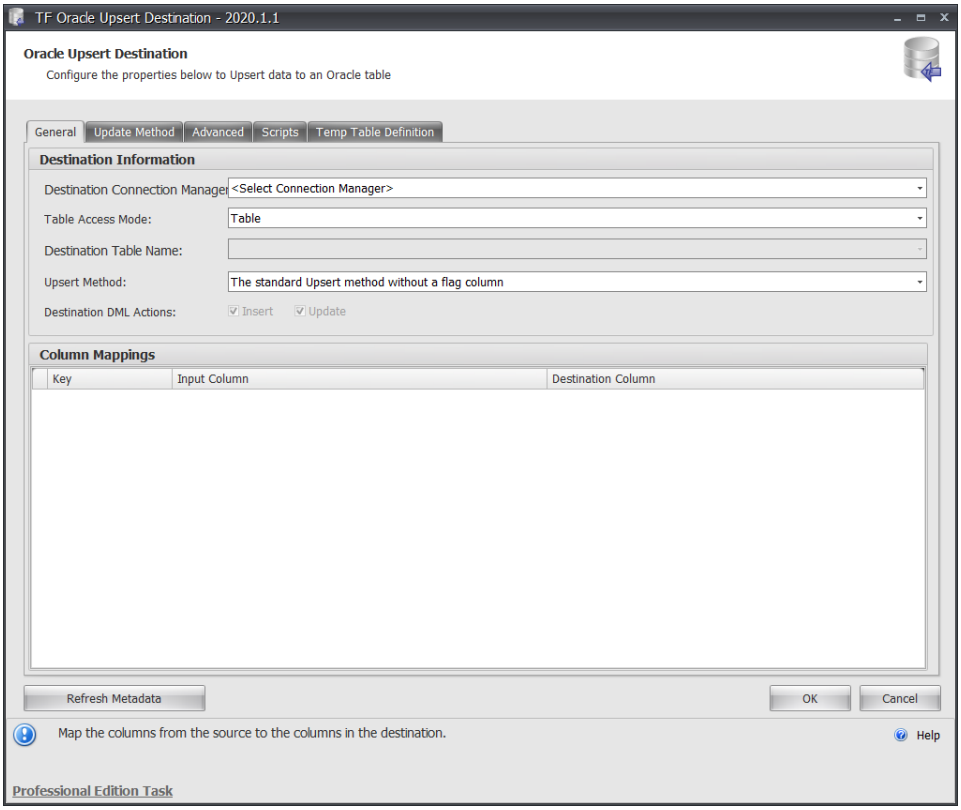
Destination Icon	Destination Description
	The **Oracle Upsert Destination** is used to conditionally insert and update data into an Oracle table within SSIS. Oracle Upsert Destination uses temporary tables. If during execution there if a failure, the removal of the temporary table could be effected and be left on the server.

Oracle Requirements

For 64-bit Task Factory:	For 32-bit Task Factory:

For 64-bit Task Factory:	For 32-bit Task Factory:
• 64-bit Oracle Data Access Components (ODAC) 11g with Oracle Developer Tools for Visual Studio • 64-bit Oracle 12c Managed Data Access Client	• 32-bit Oracle Data Access Components (ODAC) 11g with Oracle Developer Tools for Visual Studio • 32-bit Oracle 12c Managed Data Access Client

Oracle General



Oracle Destination Information

Option	Description
Destination Connection Manager	Choose an existing connection manager or create a new one. **Note:** The only supported connections are ADO.NET Oracle.DataAccess Client connections.
Table Access Mode	The table access mode allows you to define how the destination table name is selected. • **Table** - This mode allows you to choose the table in the **Destination Table Name** drop down. • **TableFromVariable** - This mode allows you to choose a variable that contains the destination table name in the **Choose Table Variable** drop down.

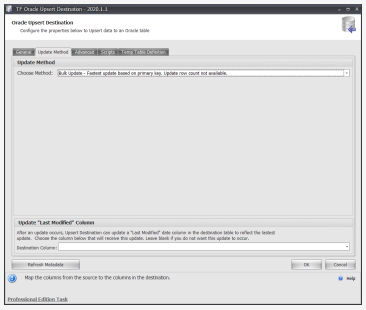
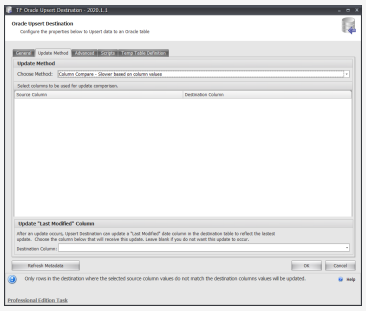
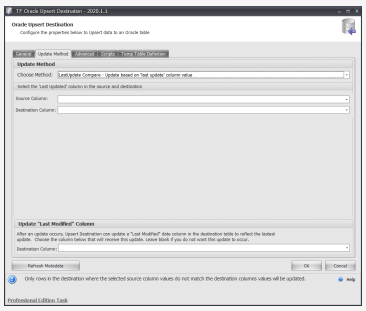
Option	Description
Destination Table Name	Available if **Table** was chosen as the access mode. Here you choose the table name from the populated drop down menu.
Choose Table Variable	Available if **TableFromVariable** was selected as the access mode. Here you select a variable.
Upsert Method	The upsert method defines how the upsert functions when executing the DML actions. There are two modes: • **The standard Upsert method without a flag column** - This mode inserts and updates records in the destination based on the key(s) selected in the column mappings. If the row exists, it gets updated, if it does not exist, the row gets inserted. • **Row Flag Compare** - This mode updates, inserts, deletes, and rejects rows based on the settings in the **Row Flag Column** drop down in the **Row Flag Column Settings** tab.
Destination DML Actions	Choose whether Upsert inserts or updates records in the destination table.
Refresh Metadata	Refreshes the destination table's metadata.

Oracle Column Mappings

Option	Description
Key	You must select at least one column to be used as the key.
Input Column	The column name from the input. To select the destination column, select the name of the input column in the grid to display a drop-down menu.
Destination Column	The column name from the destination that the input column is mapped to.

Oracle Update Method

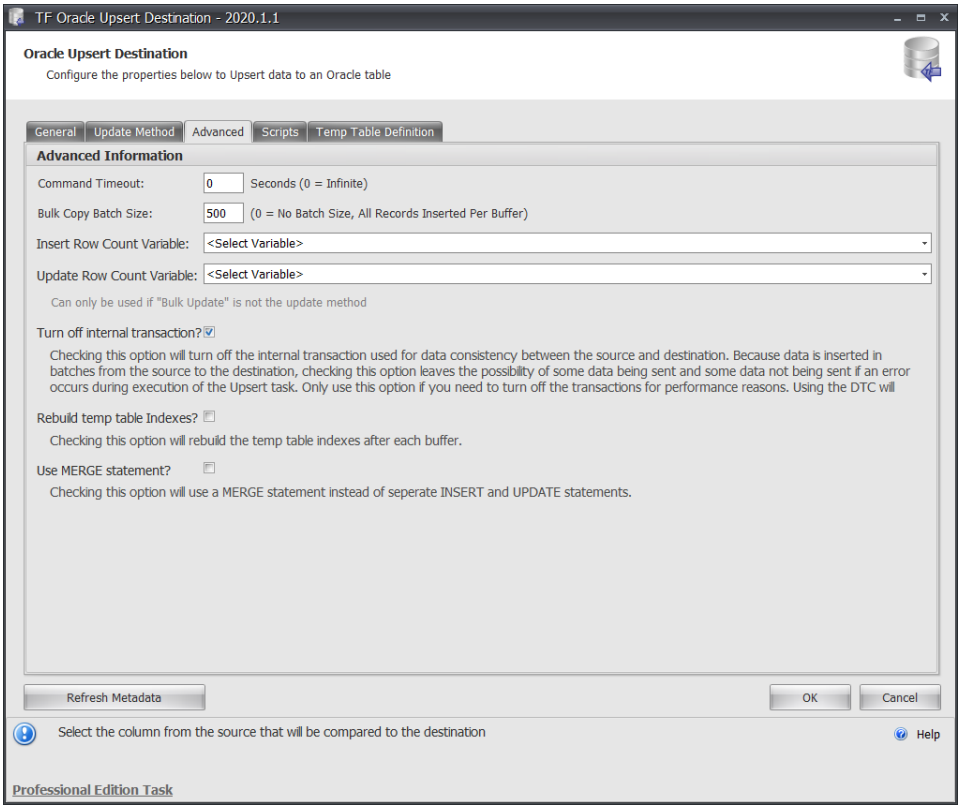
Option	Description	Image

Option	Description	Image
Bulk Update	**Bulk Update** is the default update method. In this method the destination is updated with whatever data is contained in the source as long as the keys selected in the columns mappings from the general tab match in the source and destination. It does not take into account whether data has changed or not.	
Column Compare	**Column Compare** can be used to compare whether data in the source matches what is in the destination. If the data does not match, the update occurs. Otherwise the update is not performed.	
Last Update Compare	**LastUpdate Compare** Compares a date column from the source to a date column in the destination. If the dates do not match, the row is updated.	

Oracle Update Last Modified Column

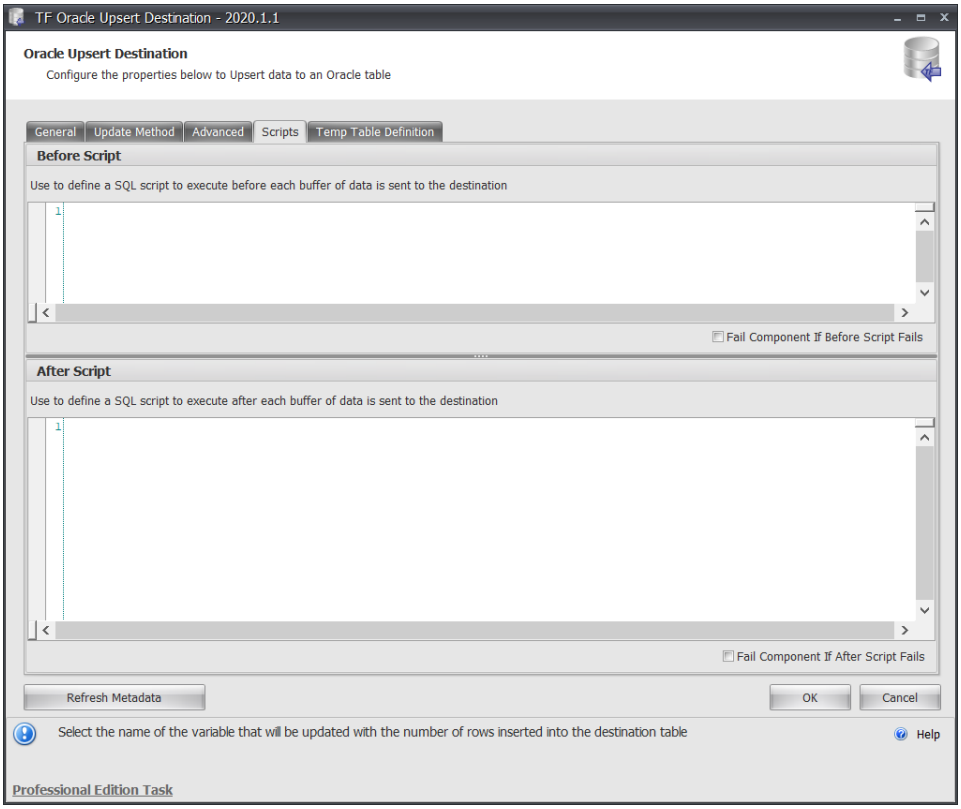
Allows you to select a column to update with the date and time when the row was updated.

Oracle Advanced Tab



Option	Description
Command Timeout	The **Command Timeout** determines the number of seconds before a timeout occurs during the inserts and updates to the destination table.
Insert Row Count Variable / Update Row Count Variable	To assign the number of rows inserted or updated into the destination table during the upsert process a variable can be assigned to hold those values by choosing a variable from the drop down menu for each count.
Turn off internal transactions	This option is used to turn off the internal transaction that upsert creates when executing the inserts and updates. This is useful if another component is going to retrieve data from the destination table at the same time as the upsert inserting and updating rows.
Rebuild temp table Indexes	Selecting this option rebuilds temp table indexes for each buffer.

Oracle Scripts



You can write custom TSQL queries to be performed before or after each buffer. To enable the **Scripts** tab, you must turn off internal transactions in the **Advanced** Tab.

Oracle Additional Videos

Support Articles

How to use the Row Flag Compare option in the Upsert Destination

See the above article for a step-by-step guide.

Upsert Destination Hangs

Article demonstrates the most common way to resolve the Upsert Destination hanging during execution.

Oracle Upsert Destination – Could not delete temporary table

If an error occurs in Oracle while the Oracle Upsert Destination is writing to the database, the component will attempt to rollback all changes and remove temporary tables that were in use. This can cause an Illegal ROLLBACK attempt to occur if Oracle believes the component is attempting to write to the temporary table while the rollback attempt is made. Typically, the error will present itself in SSIS as:

[TF Oracle Upset Destination [29]] Error: Oracle.DataAccess.Client.OracleException ORA-00600: internal error code, arguments: [4513], [130], [1]...

or

[TF Oracle Upsert Destination [29]] Error: Oracle.DataAccess.Client.OracleException ORA-24795: Illegal ROLLBACK attempt made at...

Oracle Upsert Destination Error

[TF Oracle Upsert Destination [26]] Error: System.Exception: Could not create temporary destination table: Could not add the unique index or primary key index to the destination table because no unique indexes or primary key indexes are defined for 'Tablename'

Creating a Task Factory Log File

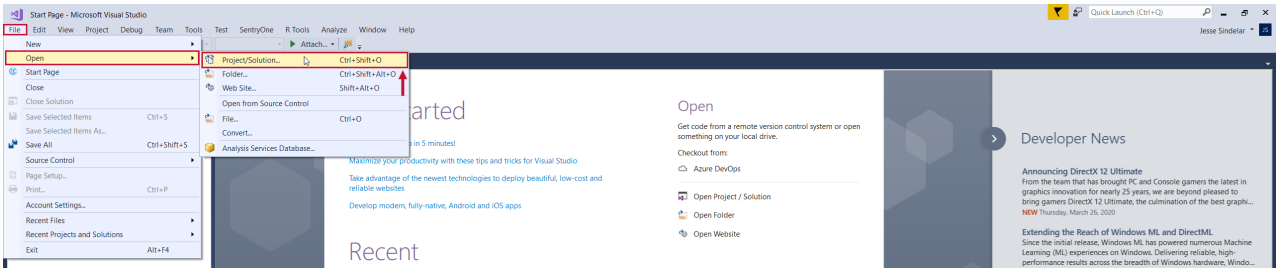
Last Modified on 28 September 2020

> 📢 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

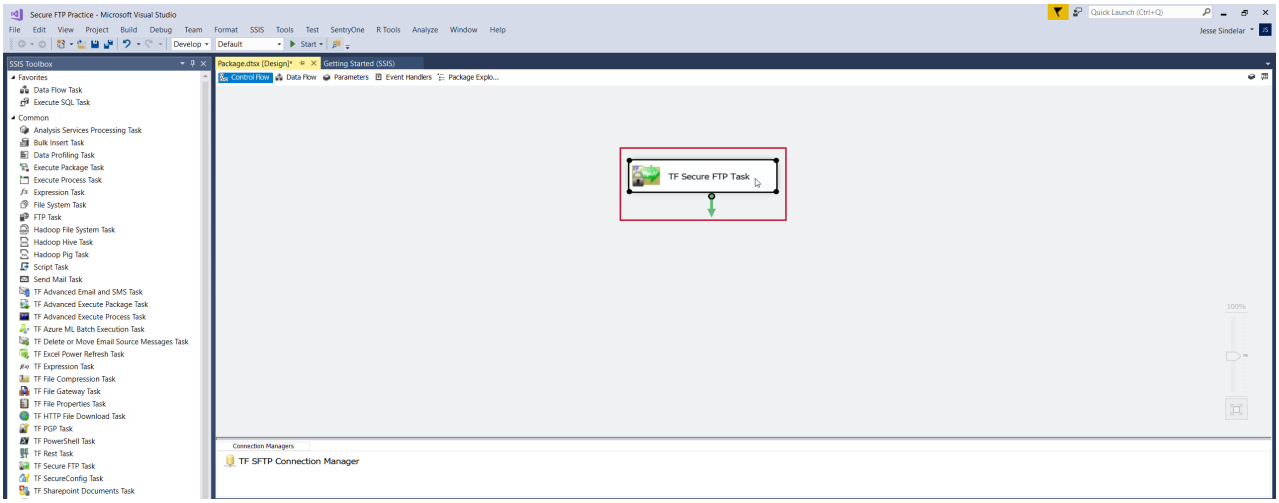
If you are running into any problems or errors with a Task Factory component, you can create a log file to record exactly what is happening. This log file helps you and support.sentryone.com diagnose any problems and work toward a solution.

To create a log file for your Task Factory component, complete the following steps:

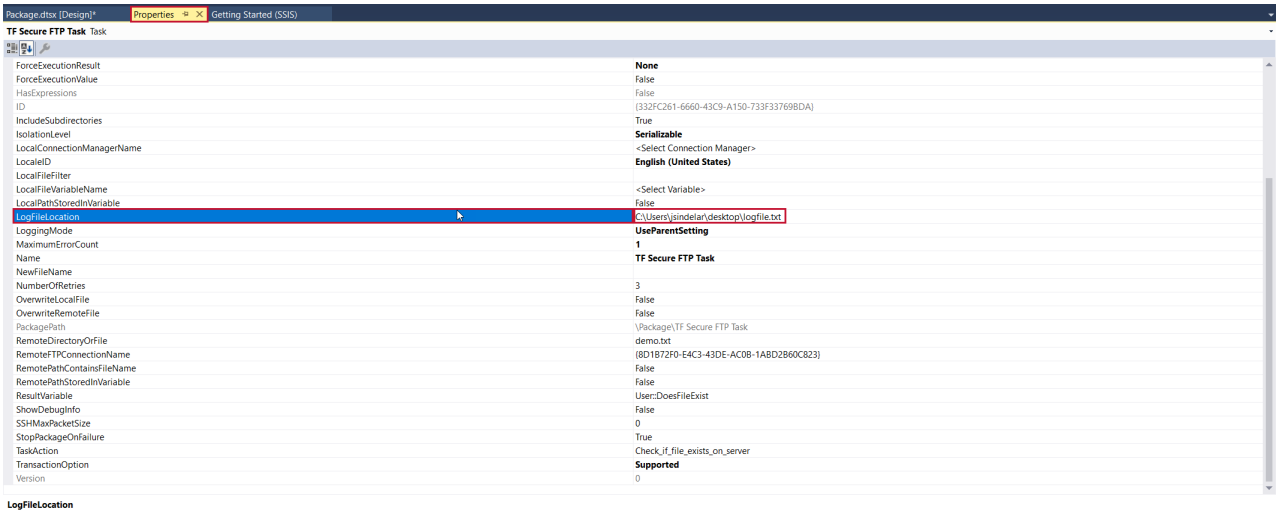
1. Open your Visual Studio package containing the Task Factory component.



2. Select the Task Factory component, then select the **F4** key to open the **properties** window.

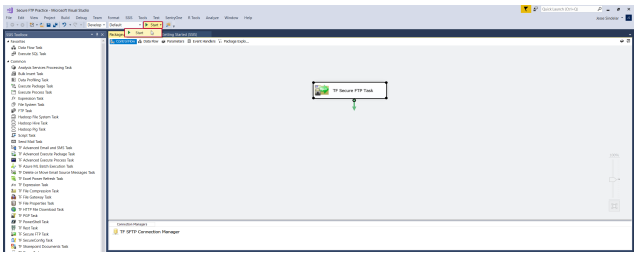


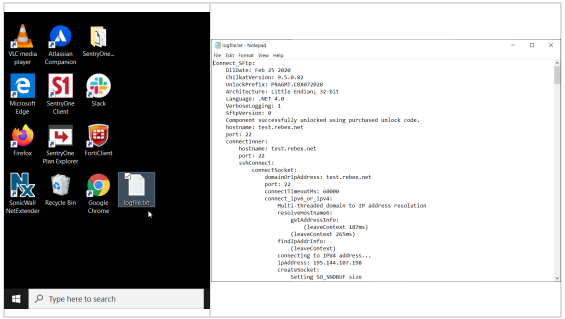
3. Go to the **Misc properties** section, then enter a fully qualified file path and name in the **LogFileLocation** property.



> **Note:** The file path and name should take the following example format: **C:\Users\username\Desktop\LogFile.txt**.

4. Save your package and **LogFileLocation** property settings. Execute the package to generate the file in the specified location.





> **Success:** Once your log file has been created, reach out to support.sentryone.com to find a solution!

Testing the Task Factory Upgrade

Last Modified on 28 September 2020

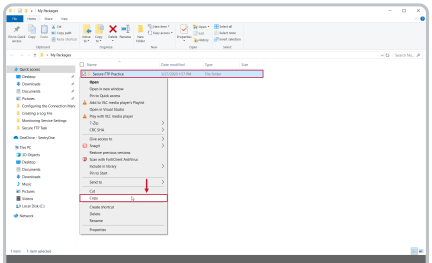
Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Warning: You must use the same version of Task Factory across any of your environments (dev, production, qa, etc.) to ensure optimal performance of Task Factory components.

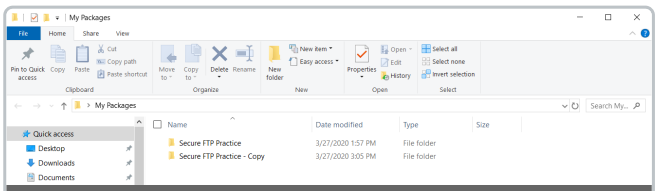
Important: SentryOne recommends having backups of your packages in a safely stored location.

Before committing to a full upgrade of Task Factory, you should test the upgrade on your packages to make sure that there won't be any unforeseen errors in your packages. To test your upgrade safely, we recommend that you take the following steps:

1. Make a copy of the package(s) from the older version of Task Factory that you want to test:



Copy your package

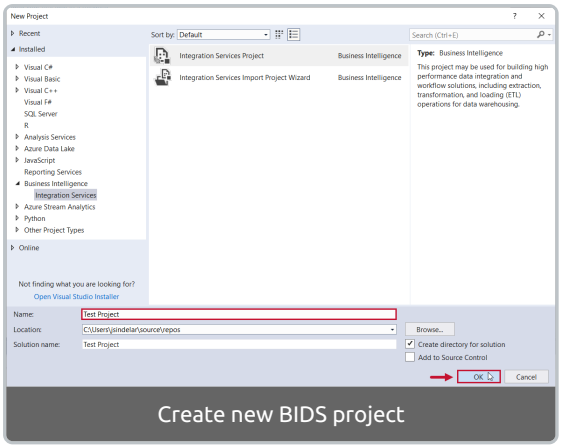


Copied Package

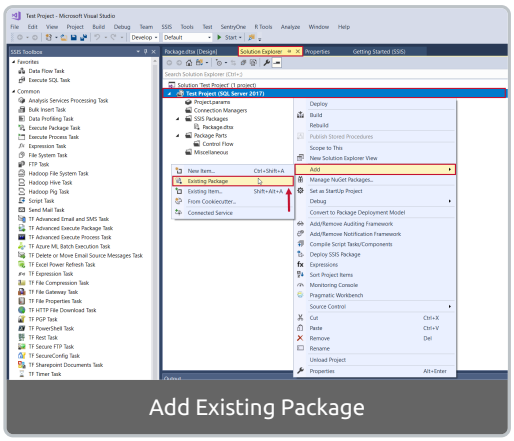
2. Uninstall the current version of Task Factory, then Install the new version of Task Factory:

Additional Information: For complete instructions on upgrading Task Factory, see the Upgrading Task Factory article.

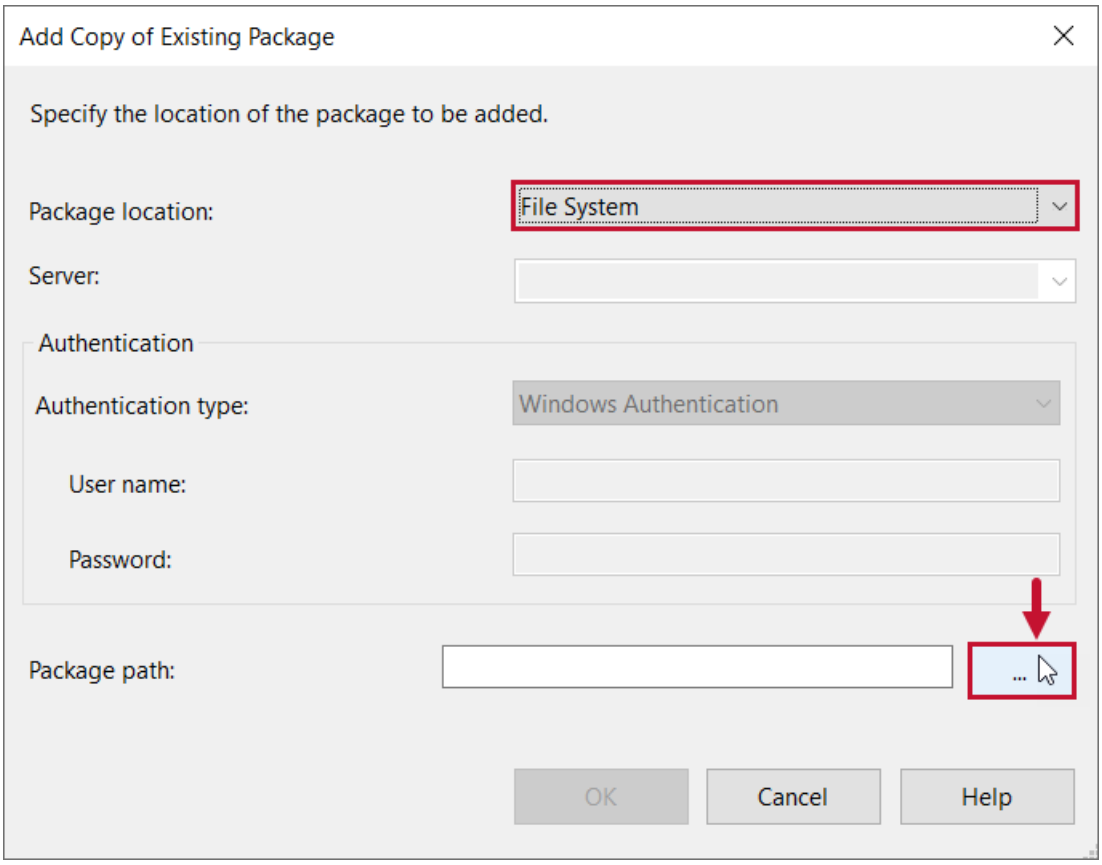
3. Create a new **Visual Studio BIDS** project. Open the **Solution Explorer** (View > Solution Explorer), right click your project, and then select **Add Existing Package** to open the **Add Copy of Existing Package** window.



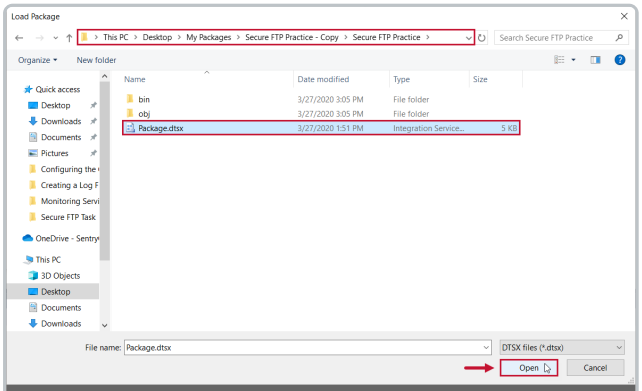
Create new BIDS project


Add Existing Package

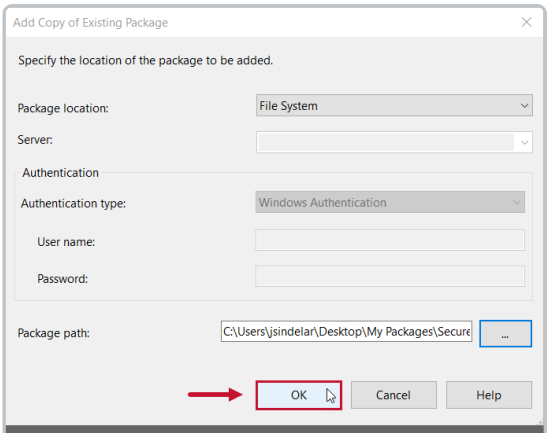
4. Select the **Package location** for the copied package, enter your authentication credentials (if applicable), and then select the ••• to select the file **Package path**:



5. Select the desired **file path** for your package, then select **OK** to add the copied package into your project:

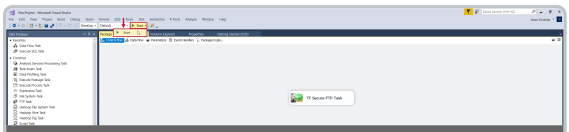


Copied Package File Path

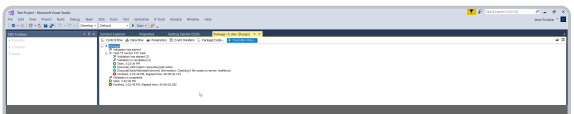


Add Copy of Existing Package

6. Execute the copied package on the new installation of Task Factory to ensure you don't run into any errors:



Execute your copied package



Execution results with no errors

Important: Testing your packages with this method ensures that you are not making any changes to your original packages.

Success: If your package executed without any errors, then you are ready to upgrade Task Factory across all your environments.

Warning: If you ran into any issues with executing your copied package, delete the copied package and project.

Task Factory Common Errors

Last Modified on 16 September 2021

Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit here for more details.

Through the course of installing or upgrading Task Factory, you can run into some common component errors in your packages. This article provides an overview for these common errors and ways to troubleshoot them.

Note: SolarWinds recommends testing your Task Factory upgrade before upgrading all of your environments.

SSIS.ReplacementTask Error

You may run into the **SSIS.ReplacementTask** error when you try to run a package after installing a new version of Task Factory.

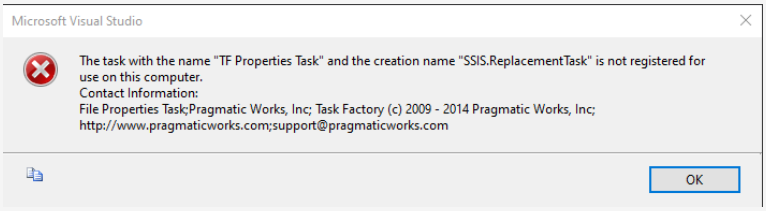
Note: The **SSIS.ReplacementTask** error may display as follows:

```
TITLE: Microsoft Visual Studio
------------------------------
The task with the name "TF Properties Task" and the creation name "SSIS.ReplacementTask" is not registered f
or use on this computer.
Contact Information:
File Properties Task;Pragmatic Works, Inc; Task Factory (c) 2009 - 2014 Pragmatic Works, Inc;
http://www.pragmaticworks.com;support@pragmaticworks.com
```

Symptoms

The **SSIS.RepalcementTask** error has the following symptoms:

Symptom	Image
Box icon on your Task Factory component or task	
Your package can't be executed or edited	

Symptom	Image
You can't see the data in your Task Factory components.	

Warning: If your package(s) exhibits any of these symptoms, **DO NOT** try to run the package, or open any of its components. It's recommended to delete this package if you have a saved copy or backup of the package.

Important: SolarWinds recommends having backups of your packages, and storing these backups in a safe and accessible location.

Cause

The **SSIS.ReplacementTask** error occurs when SSIS is looking for your Task Factory components and can't find them (your component(s) may look online for assemblies, which can take a while). When SSIS can't find your Task Factory components, it replaces the components with generic components, resulting in the **SSIS.ReplacementTask** error.

You may run into the **SSISReplacementTask** error if you've done the following:

- You've opened a package that was developed on an higher version of Task Factory in an older version. For example, you developed the package in version 2020.0, and opened it in version 2019.
- You're using the same version of Task Factory that was used to develop the package, but you are targeting a different SQL Server version. For example, you're package was created targeting SQL Server 2017, and when you've opened the package on your current machine, the package is targeting SQL Server 2019.

Solution

To fix the **SSISReplacementTask** error in your package(s), you can take one of the following actions:

- Run the package on the version of Task Factory where it was originally created and delete the corrupted package.
- Download the version of Task Factory where the package was developed and ensure that this version is installed across all of your machines.
- Use the version of Task Factory where the package is working across all of your environments.

- Ensure the package is targeting the version of SQL Server that it was targeting during its development.

Warning: You must use the same version of Task Factory across all your environments (dev, production, qa, etc.) to ensure optimal performance of Task Factory components.

Note: If you need any assistance solving this error, reach out to support.sentryone.com.

Task Factory Items are not showing up in the SSIS Toolbox

Symptoms

Recently downloaded Task Factory components are not displaying in the SSIS Toolbox.

Cause

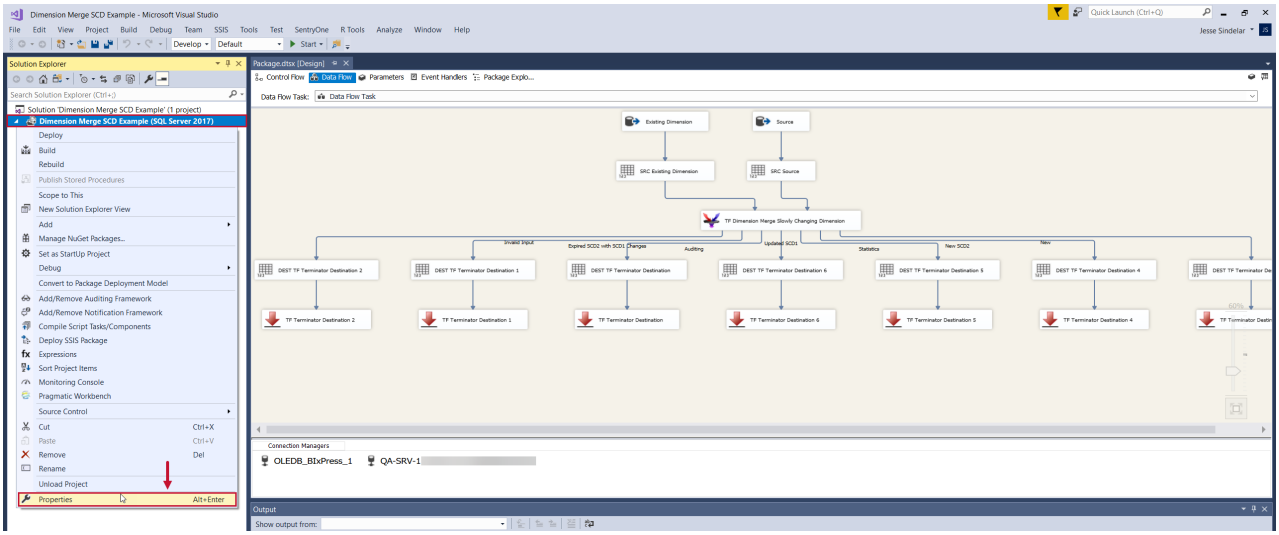
Beginning with SQL Server Data Tools 2016, backwards compatibility was added for building and editing packages for previous versions of SQL Server Integration Services. Opening your package in a newer version of Visual Studio may cause the package to target a version of SSIS that does not have Task Factory support. For example, if you custom installed Task Factory to support only SSIS 2014, and then you tried to work on a project that is targeting SSIS 2012, the Task Factory items won't display in your SSIS toolbox.

Solution

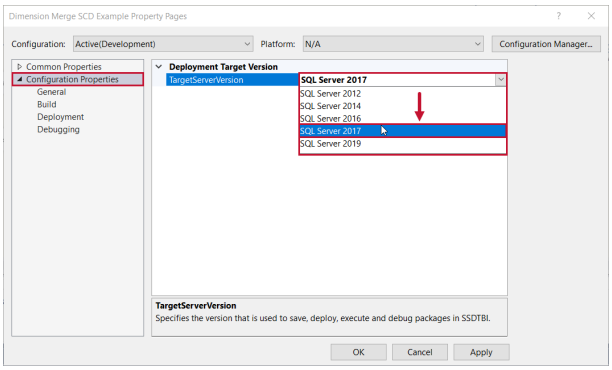
To ensure that the Task Factory components display in the SSIS toolbox, you can uninstall your current version of Task Factory, then reinstall Task Factory (making sure to select the versions of SSIS that you plan to work with).

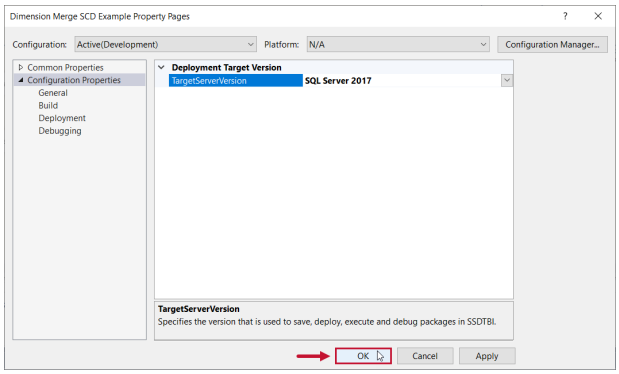
You can also change the **TargetServerVersion** of the SSIS project to match your installed version of Task Factory. To do this, complete the following steps:

1. Open your Visual Studio project. Right click your **Project** in the Solution Explorer window, then select **Properties** to open the **Property Page** window.

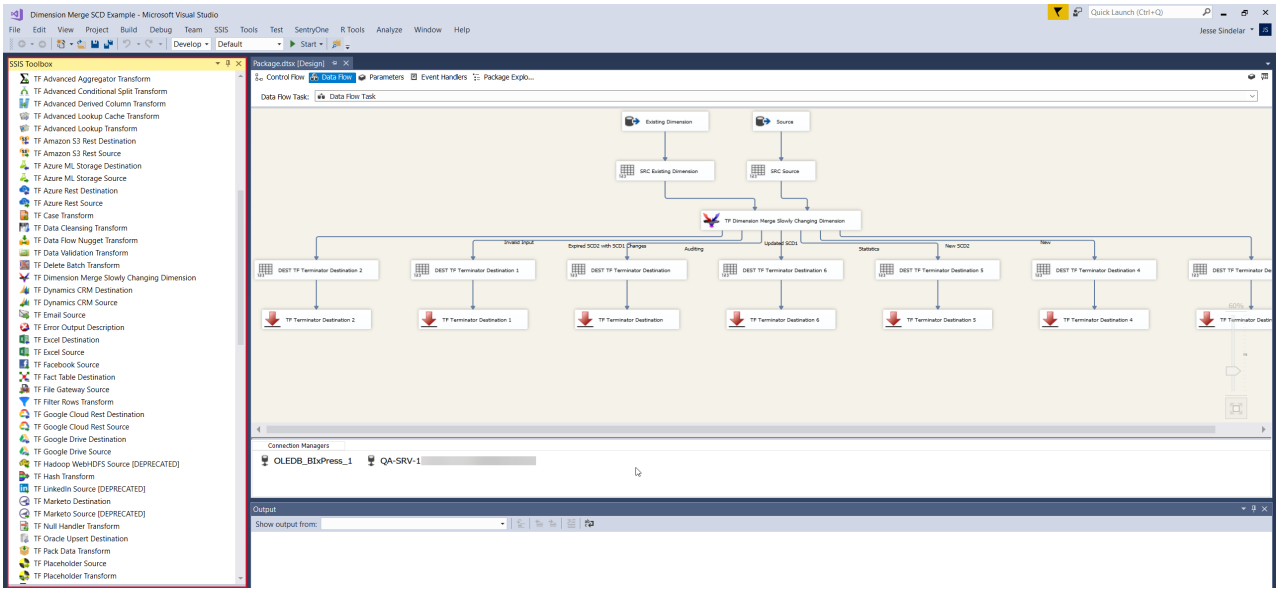


2. Select **Configuration Properties**, then select the desired **TargetServerVersion** from the drop-down list. Select **OK** to save your changes.





3. Save and close your package. Re-open the package to display the Task Factory components.



Support Articles

See the following articles on the Support website:

Advanced Execute Package Object Reference Error using In Process Execution

When you execute the package you receive an " *Object was not set to an instance of an object* " error.

Error with Temporary Directory

Error: file 'C:\Windows\TEMP\tb2igkdr.0.vb' could not be found

Getting a Login Failed Error After Deploying a Package to a Server

You built a package using a task that requires a login. You are able to run this package in Visual Studio without error however, when you deploy the package to the server you get a *Login Failed* or *Authentication Failure* error thrown.

Oracle Upsert Destination – Could not delete temporary table

If an error occurs in Oracle while the Oracle Upsert Destination is writing to the database, the component will attempt to rollback all changes and remove temporary tables that were in use. This can cause an Illegal ROLLBACK attempt to occur if Oracle believes the component is attempting to write to the temporary table while the rollback attempt is made. Typically, the error will present itself in SSIS as:

[TF Oracle Upset Destination [29]] Error: Oracle.DataAccess.Client.OracleException ORA-00600: internal error code, arguments: [4513], [130], [1]...

or

[TF Oracle Upsert Destination [29]] Error: Oracle.DataAccess.Client.OracleException ORA-24795: Illegal ROLLBACK attempt made at...

SalesForce Source Failed to Execute

Failed to execute query successfully, error message: *INAVLID_FIELD: (Name of custom column), ^ ERROR at Row: (row location number): Column (Column Location number) No such column '(Name of Column)' on entity '(name of object)'. If you are attempting to use a custom field, be sure to append the '__c' after the custom field name. Please reference your WSDL or the describe call for the appropriate names.*

This error message is associated with custom columns of the SalesForce Object being drawn from.

SalesForce Source Error

At execution of a package containing a SalesForce Source, the user sees the package fail and the following error detailed in the Execution Results/Progress tab:

[TF SalesForce Source [2]] Error: The "TF SalesForce Source" failed because truncation occurred, and the truncation row disposition on "TF SalesForce Source.Outputs[SalesForce Data].Columns[Name]" specifies failure on truncation. A truncation error occurred on the specified object of the specified component.

SFTP Task Could Not Authenticate

[ExecuteTask Failed:] Information: Could not authenticate user [Username].

SFTP Task Connection Attempt Failed

ExecuteTask Failed:] Error: A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond 129.130.13.3:49088

SFTP Task File Doesn't Exist on the Server

[ExecuteTask Failed:] Error: File /(SFTPFilePath) doesn't exist on the server.

SFTP How to Generate a Log File

When the Secure FTP Task does not work as expected, it may be necessary to have the task generate a log to help identify where the issue is occurring.

SFTP Task No Error

SFTP Task is not performing its designated function (i.e. Download File from Server, Upload File to server etc.), but it's showing as a success despite there being no sign the task actually performed the function requested.

Task Factory Deadlocking

On occasion when using some of the Task Factory destinations which are working in parallel to write to the same target table, deadlocking may occur. The error may look something like this, "*Your transaction (process ID #52) was deadlocked on {lock | communication buffer | thread} resources with another process and has been chosen as the deadlock victim. Rerun your transaction.*" This could affect the Upsert Destination, Delete Batch Transform, and the Update Batch transform.

Underlying Connection was Closed

When you're trying to connect to an API or other outside connection like SalesForce, SharePoint, Dynamics CRM, you may encounter this error: "*The underlying connection was closed: An unexpected error occurred on a send".* This error is typically due to a TLS version mismatch between the requesting party (SSIS) and receiving party (what you are connecting to).