

SpreadsheetFlow Usage/Demo Guidelines

App: SpreadsheetFlow

This app displayed pre-configured components that allow you to test the SpreadsheetFlow's tools and capabilities.

Testing Scenarios

This documents guides you through the following scenarios:

- Scenario 1: Text Parsing: Single List of values
 - Use case 1.1: Parse to Collection Variable
 - Use case 1.2: Parse to DataRow without using a Template
 - Use case 1.3: Parse to DataRow using a Template
- Scenario 2: Text Parsing: Double list of Values
 - Use case 2.1 Spreadsheet Data
 - Use case 2.2 Double Delimited List
 - Use case 2.3 Lines of Values
 - Use case 2.4 Using a DataRow Template
- Scenario 3: Split Text Blocks
 - Use Case 3.1 By Group Size
 - Use Case 3.2 By Character Length
- Scenario 4: Configuring Toast Messages
- Scenario 5: Uploading and Processing a Text File
 - Use Case 5.1 CSV files
 - User Case 5.2 TSV files
- Scenario 6: Upload and Parse a File to a DataSheet
- Scenario 7: Select and Parse a File
- Scenario 8: Bulk Updating DataRows
- Scenario 9: Bulk Deleting DataRows

Test Data

A spreadsheet is provided with randomized test data. You can use it as needed for testing. Keep in mind that this dev org only has a data storage limit of **5MB** so errors may occur when processing too much data. You can clear previously created records to free up storage space.

The screenshot displays the 'SpreadsheetFlow Sample App' interface. At the top, there's a navigation bar with 'SpreadsheetFlow' and 'Home' tabs. Below this, the 'My DataSheets' section is visible, showing a list of data sheets with columns for 'DataSheet Number', 'DataSheet Name', 'Created By', 'Created Date', and 'Last Modified Date'. The list is currently empty, displaying 'No items to display.'.

Below the list, there's a 'Parse a List of Values' section with three tabs: 'Text Collection', 'DataRow (no Template)', and 'DataRow with Template'. The 'Text Collection' tab is active, showing a text input field with the value 'Apple,Carrot,Pear,Blueberry' and a 'Delimiter' field with a comma character. A 'Next' button is located at the bottom right of this section.

On the right side, a 'Toast' message titled 'File Upload' is displayed. It contains the text: 'You can use the contents of this **spreadsheet** to facilitate testing. Export it to a csv and tsv for File Upload tests and copy from it for Parsing tests.' The word 'spreadsheet' is highlighted with a red circle. Below this, it states: 'The spreadsheet contains formula fields that will randomize data values every time it is refreshed.' and 'This component is configured to allow .csv, .tsv, and .txt file extensions to be uploaded and the user specifies how to process the file contents.' At the bottom of the toast, there's a link: '> Upload File and Create DataSheet'.

Scenario 1: Text Parsing: Single List

This feature enables you to parse a list of delimited values into a text collection variable. There are Apex Actions to parse the list and calculate the number of values. The demo screenflow is pre-populated with sample data.

The screenshot displays the 'SpreadsheetFlow Sample App' interface. At the top, there's a navigation bar with 'SpreadsheetFlow' and 'Home' tabs, a search bar, and various utility icons. Below this, a section titled 'DataSheets My DataSheets' shows a list of data sheets with columns for 'DataSheet Number', 'DataSheet Name', 'Created By', 'Created Date', and 'Last Modified Date'. The list is currently empty, displaying 'No items to display.'

The main configuration area is titled 'Parse a List of Values' and is highlighted with a red border. It contains three tabs: 'Text Collection' (selected), 'DataRow (no Template)', and 'DataRow with Template'. Under the 'Text Collection' tab, there are two input fields: 'List of Delimited Values' containing the text 'Apple,Carrot,Pear,Blueberry' and 'Delimiter' containing a comma character ','. A 'Next' button is located at the bottom right of this section.

On the right side of the interface, there's a 'Toast' notification titled 'File Upload' with the following text: 'You can use the contents of this spreadsheet to facilitate testing. Export it to a csv and tsv for File Upload tests and copy from it for Parsing tests. The spreadsheet contains formula fields that will randomize data values every time it is refreshed. This component is configured to allow .csv, .tsv, and .txt file extensions to be uploaded and the user specifies how to process the file contents. > Upload File and Create DataSheet'.

Use Case 1.1 Parse to Collection Variable

This option parses a list of delimited values and returns a collection of text values and the number of values in the list.

The screenshot displays the 'SpreadsheetFlow Sample App' interface. At the top, there's a navigation bar with 'SpreadsheetFlow' and 'Home' tabs. Below this, a 'DataSheets' section titled 'My DataSheets' shows a list of 0 items, sorted by 'DataSheet Number'. The main content area is divided into two tabs: 'Parse a List of Values' (selected) and 'Parse Text Block'. Under 'Parse a List of Values', there are two sub-tabs: 'Text Collection' (highlighted with a red circle) and 'DataRow (no Template)'. The 'Text Collection' sub-tab contains a form with two input fields: 'List of Delimited Values' (containing 'Apple,Carrot,Pear,Blueberry') and 'Delimiter' (containing ','). A 'Next' button is located at the bottom right of the form. To the right of the main interface, a 'File Upload' panel is visible, containing text about using spreadsheet contents for testing and a link to 'Upload File and Create DataSheet'.

Use Case 1.2 Parse to DataRow without Template

This option parses a list of delimited values to a DataRow variable without using a DataRow template. By default, text fields are used in the parsing process. The screenflow uses the Apex Action SSF: Describe DataRow to display the assigned field values.

The screenshot displays the 'SpreadsheetFlow Sample App' interface. At the top, there's a navigation bar with 'SpreadsheetFlow' and 'Home' tabs. Below this, a section titled 'My DataSheets' shows a list of data sheets (currently empty) with columns for 'DataSheet Number', 'DataSheet Name', 'Created By', 'Created Date', and 'Last Modified Date'. The main configuration area is titled 'Parse a List of Values' and includes three tabs: 'Parse a List of Values', 'Parse Text Block', and 'Split Text into Blocks'. Under 'Parse a List of Values', there are three options: 'Text Collection', 'DataRow (no Template)' (which is highlighted with a red circle), and 'DataRow with Template'. The 'List of Delimited Values' field contains the text 'Apple,Carrot,Pear,Blueberry,Orange'. The 'Delimiter' field contains a comma ','. A 'Next' button is visible at the bottom right of the configuration area. On the right side of the screen, there's a 'File Upload' section with a 'Test Data' tab and a description of the spreadsheet's purpose for testing.

SpreadsheetFlow Sample App

My DataSheets

0 items • Sorted by DataSheet Number • Filtered by My datasheets • Updated 9 minutes ago

Search this list...

☐ DataSheet Number ↑ DataSheet Name Created By Created Date Last Modified Date

No items to display.

Parse a List of Values Parse Text Block Split Text into Blocks

Text Collection **DataRow (no Template)** DataRow with Template

* List of Delimited Values ⓘ

Apple,Carrot,Pear,Blueberry,Orange

* Delimiter ⓘ

,

Next

File Upload

Test Data

You can use the contents of this [spreadsheet](#) to facilitate testing. Export it to a csv and tsv for File Upload tests and copy from it for Parsing tests.

The spreadsheet contains formula fields that will randomize data values every time it is refreshed.

This component is configured to allow .csv, .tsv, and .txt file extensions to be uploaded and the user specifies how to process the file contents.

> Upload File and Create DataSheet

Use Case 1.3 Parse to DataRow using a Template

This option parses a list of delimited values to a DataRow variable using a DataRow template. The template provides field values required for creating the record. The ImportFieldType value determines if the parsed values are stored in the Text or Long Text fields. Header DataRow Types will always be stored in the Text fields. The screenflow uses the Apex Action SSF: Describe DataRow to display the assigned field values.

The screenshot displays the 'SpreadsheetFlow Sample App' interface. At the top, there's a navigation bar with 'SpreadsheetFlow' and 'Home' tabs. Below this, a section titled 'My DataSheets' shows a list of data sheets, currently empty. The main configuration area is titled 'Parse a List of Values' and has three tabs: 'Text Collection', 'DataRow (no Template)', and 'DataRow with Template' (which is selected and circled in red). Under the 'DataRow with Template' tab, there are two main sections: 'TextBlock Value' and 'List Value Delimiter'. The 'TextBlock Value' section contains a text input field with the value 'Apple,Carrot,Pear,Blueberry,Orange'. The 'List Value Delimiter' section has radio buttons for 'Comma' (selected), 'Semicolon', 'Tab', 'Spreadsheet Column', 'Spreadsheet Row', and 'Custom'. To the right of these is the 'DataRow Template Details' section, which includes three dropdown menus: 'DataRow RecordType' (set to '--None--'), 'ImportFieldType' (set to '--None--'), and 'DataRow Type' (set to '--None--'). A 'Next' button is located at the bottom right of the configuration area. On the right side of the screen, there's a 'Toast' notification titled 'File Upload' with the text: 'You can use the contents of this spreadsheet to facilitate testing. Export it to a csv and tsv for File Upload tests and copy from it for Parsing tests. The spreadsheet contains formula fields that will randomize data values every time it is refreshed. This component is configured to allow .csv, .tsv, and .txt file extensions to be uploaded and the user specifies how to process the file contents.' Below this, there's a link 'Upload File and Create DataSheet'.

Scenario 2: Text Parsing: Parse Text Block

This feature enables you to parse a double list of delimited values into a collection of DataRow variables. The screenflow is pre-populated with sample data and uses a data table component to display the parsed values.

The screenshot displays the 'SpreadsheetFlow Sample App' interface. At the top, there's a navigation bar with 'SpreadsheetFlow' and 'Home' tabs. Below this, a section titled 'DataSheets My DataSheets' shows a list of data sheets, currently empty. The main workflow area is divided into three steps: 'Parse a List of Values', 'Parse Text Block' (highlighted with a red circle), and 'Split Text into Blocks'. The 'Parse Text Block' step is active, showing a 'Spreadsheet Range' tab and a text input field labeled '* Spreadsheet Value'. To the right, a 'File Upload' panel provides instructions on using spreadsheets for testing and uploading files.

SpreadsheetFlow Sample App

DataSheets
My DataSheets

0 items • Sorted by DataSheet Number • Filtered by My datasheets • Updated 13 minutes ago

Search this list...

DataSheet Number ↑ DataSheet Name Created By Created Date Last Modified Date

No items to display.

Parse a List of Values **Parse Text Block** Split Text into Blocks

Spreadsheet Range Double Delimited List Lines of Values DataRows (with Template)

* Spreadsheet Value ⓘ

Next

File Upload

Test Data

You can use the contents of this [spreadsheet](#) to facilitate testing. Export it to a csv and tsv for File Upload tests and copy from it for Parsing tests.

The spreadsheet contains formula fields that will randomize data values every time it is refreshed.

This component is configured to allow .csv, .tsv, and .txt file extensions to be uploaded and the user specifies how to process the file contents.

> Upload File and Create DataSheet

Use Case 2.1 Spreadsheet Data

This option enables you to parse a range of spreadsheet cells into a collection of DataRow variables. You can use the sample spreadsheet provided for testing or use any spreadsheet you have. The delimiter parameters are pre-configured into the component. The screenflow uses a data table component to display the parsed values and the Apex Action SSF: Describe List of DataRow Records to display the assigned field values.

The screenshot displays the 'SpreadsheetFlow Sample App' interface. At the top, there's a navigation bar with 'SpreadsheetFlow' and 'Home' tabs. Below this, a section titled 'DataSheets My DataSheets' shows a list of data sheets, currently empty. The main configuration area is titled 'Parse a List of Values' and 'Parse Text Block'. Under 'Parse Text Block', the 'Spreadsheet Range' option is selected and highlighted with a red circle. Other options include 'Double Delimited List', 'Lines of Values', and 'DataRows (with Template)'. A text input field labeled '* Spreadsheet Value' is visible, followed by a 'Next' button. On the right side, a 'File Upload' panel provides instructions: 'You can use the contents of this spreadsheet to facilitate testing. Export it to a csv and tsv for File Upload tests and copy from it for Parsing tests.' It also mentions that the spreadsheet contains formula fields that randomize data values and that the component is configured to allow .csv, .tsv, and .txt file extensions.

Use Case 2.2 Double Delimited List

This option shows how a double delimited list of values using any delimiter value can be parsed into DataRow records. The screenflow uses a data table component to display the parsed values and the Apex Action SSF: Describe List of DataRow Records to display the assigned field values.

The screenshot displays the 'SpreadsheetFlow Sample App' interface. At the top, there's a navigation bar with 'SpreadsheetFlow' and 'Home' tabs. Below this, a section titled 'DataSheets My DataSheets' shows a list of data sheets, currently empty. The main configuration area is titled 'Parse a List of Values' and includes three tabs: 'Parse Text Block' (selected), 'Split Text into Blocks', and 'Lines of Values'. Under 'Parse Text Block', there are four options: 'Spreadsheet Range', 'Double Delimited List' (highlighted with a red circle), 'Lines of Values', and 'DataRows (with Template)'. The 'Double Delimited List' option is active, showing a text input field with the value 'Apple,Red,Fruit;Carrot,Orange,Vegetable;Pear,Green,Fruit;Blueberry,Blue,Fruit'. Below this, there are fields for 'First Delimiter' (set to ';') and 'Second Delimiter' (set to ','). A 'Next' button is at the bottom right of the configuration area. On the right side, a 'File Upload' panel provides instructions on using spreadsheet contents for testing and parsing, mentioning supported file extensions (.csv, .tsv, .txt) and the ability to upload files to create data sheets.

Use Case 2.3 Lines of Values

This option shows how lines containing a list of delimited values can be parsed into DataRow records. The screenflow uses a data table component to display the parsed values and the Apex Action SSF: Describe List of DataRow Records to display the assigned field values.

The screenshot displays the 'SpreadsheetFlow Sample App' interface. At the top, there's a navigation bar with 'SpreadsheetFlow' and 'Home' tabs. Below this, a section titled 'My DataSheets' shows a list of data sheets, currently empty. The main configuration area is titled 'Parse a List of Values' and includes tabs for 'Parse Text Block', 'Split Text into Blocks', and 'Lines of Values' (which is selected and circled in red). Under 'Lines of Values', there are two sections: 'TextBlock Value' and 'List Value Delimiter'. The 'TextBlock Value' section contains a text area with the following text: 'Apple;Red;Fruit', 'Carrot;Orange;Vegetable', 'Pear;Green;Fruit', and 'Blueberry;Blue;Fruit'. The 'List Value Delimiter' section has radio buttons for 'Comma', 'Semicolon' (selected), 'Tab', 'Spreadsheet Column', 'Spreadsheet Row', and 'Custom'. To the right of these sections are 'Row Delimiter' options: 'Comma', 'Semicolon', 'Tab', 'Spreadsheet Column', 'Spreadsheet Row' (selected), and 'Custom'. A 'Next' button is located at the bottom right of the configuration area. On the right side of the screen, a 'Toast' notification titled 'File Upload' is visible, containing text about using spreadsheet contents for testing and a link to 'Upload File and Create DataSheet'.

Use Case 2.4 Parsing to DataRows using a Template

This option shows how a DataRow template can be used to set how lines containing a list of delimited values can be parsed into DataRow records. The screenflow uses a data table component to display the parsed values and the Apex Action SSF: Describe List of DataRow Records to display the assigned field values. It also uses an Apex Action SSF: Get Assign Record Types which is used to display the running user's assigned record types in the picklist.

The screenshot displays the 'SpreadsheetFlow Sample App' interface. At the top, there's a navigation bar with 'SpreadsheetFlow' and 'Home' tabs. Below this, the 'My DataSheets' section shows a list of data sheets with columns for 'DataSheet Number', 'DataSheet Name', 'Created By', 'Created Date', and 'Last Modified Date'. A message indicates 'No items to display.'.

The main configuration area is titled 'Parse a List of Values' and includes tabs for 'Parse Text Block', 'Split Text into Blocks', and 'DataRows (with Template)'. The 'DataRows (with Template)' tab is selected and highlighted with a red circle. It contains a 'TextBlock Value' field with the text 'Name;Color;Category' and 'Apple;Red;Fruit'. Below this, there are two sections: 'List Value Delimiter' and 'Row Delimiter'. The 'List Value Delimiter' section has radio buttons for 'Comma', 'Semicolon' (selected), 'Tab', 'Spreadsheet Column', 'Spreadsheet Row', and 'Custom'. The 'Row Delimiter' section has radio buttons for 'Comma', 'Semicolon', 'Tab', 'Spreadsheet Column', 'Spreadsheet Row' (selected), and 'Custom'. To the right, the 'DataRow Template Details' section includes a 'DataRow RecordType' dropdown set to '--None--', an 'ImportFieldType' dropdown set to '--None--', and a checked checkbox for 'First Line of Values are Headers'. A 'Next' button is located at the bottom right of the configuration area.

On the right side of the interface, there's a 'Toast' notification titled 'File Upload' with a 'Test Data' section. It contains text explaining that the spreadsheet contents can be used for testing and that the spreadsheet contains formula fields that randomize data values. Below this, there's a link to 'Upload File and Create DataSheet' and a button labeled 'Upload File and Create DataSheet'.

Scenario 3: Splitting Text Blocks

This feature enables you to split large text blocks into smaller chunk sizes by either group size or character length.

The screenshot displays the 'SpreadsheetFlow Sample App' interface. At the top, there's a navigation bar with 'SpreadsheetFlow' and 'Home' tabs. Below this, a 'DataSheets' section titled 'My DataSheets' shows a list of 0 items. The main content area has three tabs: 'Parse a List of Values', 'Parse Text Block', and 'Split Text into Blocks' (which is highlighted with a red circle). Under the 'Split Text into Blocks' tab, there's a text area for 'TextBlock Value' containing a long string of text. Below this, there are two sections: 'Row Delimiter' with radio buttons for Comma, Semicolon, Tab, Spreadsheet Column, Spreadsheet Row (selected), and Custom; and 'Split Text Block By' with radio buttons for Group Size and Character Length. A 'Next' button is at the bottom right. On the right side, a 'File Upload' panel provides instructions on using the spreadsheet for testing and exporting to CSV/TSV.

SpreadsheetFlow Sample App

DataSheets
My DataSheets
0 items • Sorted by DataSheet Number • Filtered by My datasheets • Updated 24 minutes ago

Search this list...

☐ DataSheet Number ↑ ☐ DataSheet Name ☐ Created By ☐ Created Date ☐ Last Modified Date

No items to display.

Parse a List of Values Parse Text Block **Split Text into Blocks**

* TextBlock Value

TextValue_A01,TextCode_A01,TextId_A01,Reference_A01,Indicator_A01,Check_A01,Parameter_A01
TextValue_A02,TextCode_A02,TextId_A02,Reference_A02,Indicator_A02,Check_A02,Parameter_A02
TextValue_A03,TextCode_A03,TextId_A03,Reference_A03,Indicator_A03,Check_A03,Parameter_A03
TextValue_A04,TextCode_A04,TextId_A04,Reference_A04,Indicator_A04,Check_A04,Parameter_A04
TextValue_A05,TextCode_A05,TextId_A05,Reference_A05,Indicator_A05,Check_A05,Parameter_A05
TextValue_A06,TextCode_A06,TextId_A06,Reference_A06,Indicator_A06,Check_A06,Parameter_A06
TextValue_A07,TextCode_A07,TextId_A07,Reference_A07,Indicator_A07,Check_A07,Parameter_A07

* Row Delimiter

☐ Comma
☐ Semicolon
☐ Tab
☐ Spreadsheet Column
☒ Spreadsheet Row
☐ Custom

* Split Text Block By

☐ Group Size
☐ Character Length

Next

Toast **File Upload**

Test Data

You can use the contents of this [spreadsheet](#) to facilitate testing. Export it to a csv and tsv for File Upload tests and copy from it for Parsing tests.

The spreadsheet contains formula fields that will randomize data values every time it is refreshed.

This component is configured to allow .csv, .tsv, and .txt file extensions to be uploaded and the user specifies how to process the file contents.

> Upload File and Create DataSheet

Use Case 3.1 Split by Group Size

This option splits the text block into chunks limited to a max number of values. A character limit is also enforced, so whichever limit is reached first, prevails.

The screenshot displays the 'SpreadsheetFlow Sample App' interface. At the top, there's a navigation bar with 'SpreadsheetFlow' and 'Home' tabs. Below this, a 'DataSheets' section titled 'My DataSheets' shows 0 items, sorted by 'DataSheet Number'. A table with columns 'DataSheet Number', 'DataSheet Name', 'Created By', 'Created Date', and 'Last Modified Date' is shown, but it's empty with the message 'No items to display.'.

The main configuration area is titled 'Split Text into Blocks' and is highlighted with a red border. It contains the following settings:

- TextBlock Value:** A text area containing a list of 14 text blocks, each with a unique identifier (e.g., TextValue_A01, TextCode_A01, TextId_A01, Reference_A01, Indicator_A01, Check_A01, Parameter_A01).
- Row Delimiter:** A radio button group with options: Comma, Semicolon, Tab, Spreadsheet Column, **Spreadsheet Row** (selected), and Custom.
- Split Text Block By:** A radio button group with options: **Group Size** (selected) and Character Length.
- Group Size:** A text input field with the value '5'.
- Character Length:** A text input field with the value '500'.

A 'Next' button is located at the bottom right of the configuration area.

On the right side of the interface, there's a 'File Upload' section. It includes a 'Test Data' tab and a description: 'You can use the contents of this spreadsheet to facilitate testing. Export it to a csv and tsv for File Upload tests and copy from it for Parsing tests.' Below this, it states: 'The spreadsheet contains formula fields that will randomize data values every time it is refreshed.' and 'This component is configured to allow .csv, .tsv, and .txt file extensions to be uploaded and the user specifies how to process the file contents.' At the bottom of this section is a button labeled 'Upload File and Create DataSheet'.

Use Case 3.2 Split by Character Length

This option splits the text block into chunks limited to a max number of characters without breaking up a list of values.

The screenshot displays the 'SpreadsheetFlow Sample App' interface. At the top, there's a navigation bar with 'SpreadsheetFlow' and 'Home' tabs. Below this, a 'DataSheets' section titled 'My DataSheets' shows a list of items (currently empty) with columns for 'DataSheet Number', 'DataSheet Name', 'Created By', 'Created Date', and 'Last Modified Date'. The main content area is divided into three tabs: 'Parse a List of Values', 'Parse Text Block', and 'Split Text into Blocks'. The 'Split Text into Blocks' tab is active and highlighted with a red border. It contains a text area for 'TextBlock Value' with a sample text string. Below this, there are two sections: 'Row Delimiter' with radio buttons for 'Comma', 'Semicolon', 'Tab', 'Spreadsheet Column', 'Spreadsheet Row' (selected), and 'Custom'; and 'Split Text Block By' with radio buttons for 'Group Size' and 'Character Length' (selected). A 'Character Length' input field is set to '500'. A 'Next' button is at the bottom right of the configuration panel. To the right of the main interface, a 'File Upload' panel provides instructions on using the spreadsheet for testing and uploading files.

SpreadsheetFlow Sample App

DataSheets
My DataSheets

0 items • Sorted by DataSheet Number • Filtered by My datasheets • Updated 26 minutes ago

Search this list...

No items to display.

Parse a List of Values Parse Text Block **Split Text into Blocks**

* TextBlock Value

TextValue_A01,TextCode_A01,TextId_A01,Reference_A01,Indicator_A01,Check_A01,Parameter_A01
TextValue_A02,TextCode_A02,TextId_A02,Reference_A02,Indicator_A02,Check_A02,Parameter_A02
TextValue_A03,TextCode_A03,TextId_A03,Reference_A03,Indicator_A03,Check_A03,Parameter_A03
TextValue_A04,TextCode_A04,TextId_A04,Reference_A04,Indicator_A04,Check_A04,Parameter_A04
TextValue_A05,TextCode_A05,TextId_A05,Reference_A05,Indicator_A05,Check_A05,Parameter_A05
TextValue_A06,TextCode_A06,TextId_A06,Reference_A06,Indicator_A06,Check_A06,Parameter_A06
TextValue_A07,TextCode_A07,TextId_A07,Reference_A07,Indicator_A07,Check_A07,Parameter_A07

* Row Delimiter

☐ Comma
☐ Semicolon
☐ Tab
☐ Spreadsheet Column
☒ Spreadsheet Row
☐ Custom

* Split Text Block By

☐ Group Size
☒ Character Length

* Character Length ⓘ

500

Next

Toast **File Upload**

▼ Test Data

You can use the contents of this [spreadsheet](#) to facilitate testing. Export it to a csv and tsv for File Upload tests and copy from it for Parsing tests.

The spreadsheet contains formula fields that will randomize data values every time it is refreshed.

This component is configured to allow .csv, .tsv, and .txt file extensions to be uploaded and the user specifies how to process the file contents.

> Upload File and Create DataSheet

Scenario 4: Configuring Toast Messages

Toast messages are used for bulk processing. This Screenflow tutorial provides a way to configure and send toast notifications. With simple customizations, you can override the toast message information sent by the application.

The screenshot displays the 'SpreadsheetFlow Sample App' interface. The main content area shows a 'DataSheets' section with a table of 'My DataSheets'. The table is currently empty, displaying 'No items to display.' Below this, there are tabs for 'Parse a List of Values', 'Parse Text Block', and 'Split Text into Blocks'. The 'Parse a List of Values' tab is active, showing a 'Text Collection' section with a 'List of Delimited Values' field containing 'Apple,Carrot,Pear,Blueberry' and a 'Delimiter' field containing ','. A 'Next' button is visible at the bottom right of this section.

On the right side, a 'Toast Message Configuration' panel is open, highlighted with a red border. This panel has two tabs: 'Toast' and 'File Upload'. The 'Toast' tab is selected. It contains the following configuration options:

- Type:** Radio buttons for success (selected), error, info, warning, and other.
- Mode:** Radio buttons for dismissible (selected), pester, and sticky.
- Duration (seconds):** A slider set to 5 seconds, with a range of 5-10.
- Title:** A text input field.
- Message:** A text input field.
- Next:** A blue button at the bottom right.

Scenario 5: Uploading and Processing Text File

This feature enables you to configure a screenflow element that will enable the user to upload a text file, define the type of file, display and/or set default parameters for the DataSheet and DataRow records, and options for batch processing and handling the header row.

The screenshot displays the 'SpreadsheetFlow Sample App' interface. The main content area is titled 'My DataSheets' and shows a list of data sheets (currently empty). Below this, there are three tabs: 'Parse a List of Values', 'Parse Text Block', and 'Split Text into Blocks'. The 'Parse a List of Values' tab is active, showing a 'Text Collection' section with a 'List of Delimited Values' (Apple,Carrot,Pear,Blueberry) and a 'Delimiter' (comma). A 'Next' button is visible at the bottom right of this section.

On the right side, the 'File Upload' configuration panel is open. It includes the following sections:

- File Upload** (highlighted with a red circle)
- Test Data**
- Upload File and Create DataSheet** (highlighted with a red circle)
- File Details**
 - Drag and Drop a CSV, TSV, or TXT file
 - Attach File to Process
 - Upload Files (button)
 - Or drop files (text)
 - File Type
 - CSV (Comma Separated Value) (selected)
 - TSV (Tab Separated Value)
 - Custom Delimiter
- DataSet Details**
- DataRow Details**
- File Processing Details**
 - Initial Rows to Skip (0)
 - Next Row Contains Headers (checked)
 - Save Header Row (unchecked)
 - Rows to Skip after Header (0)
 - Override Batch Size (unchecked)

A 'Next' button is located at the bottom right of the configuration panel.

Use Case 5.1 CSV Files

▼ Upload File and Create DataSheet ▼ File Details Drag and Drop a CSV, TSV, or TXT file Attach File to Process  Upload Files Or drop files * File Type ☒ CSV (Comma Separated Value) ☐ TSV (Tab Separated Value) ☐ Custom Delimiter ▼ DataSheet Details * DataSheet Record Type Standard <td data-bbox="1050 302 1911 1068"> The File Details section provides options to upload a file and specify the type of file it is for processing. For this demo, csv, tsv, and txt file extensions are permitted. The File Type option indicates how the data is structured so it can be processed accordingly. For this use case, drag and drop a csv file and select the CSV file type. </td>	The File Details section provides options to upload a file and specify the type of file it is for processing. For this demo, csv, tsv, and txt file extensions are permitted. The File Type option indicates how the data is structured so it can be processed accordingly. For this use case, drag and drop a csv file and select the CSV file type.
--	--

Toast **File Upload**

> Test Data

✓ Upload File and Create DataSheet

> [File Details](#)

* File Type

- ☒ CSV (Comma Separated Value)
☐ TSV (Tab Separated Value)
☐ Custom Delimiter

✓ DataSheet Details

* DataSheet Record Type

Standard

DataSet Name

New DataSet

DataSet Category

--None--

DataSet Description

DataSet Date

DataSet Reference Text

DataSet Reference Number

The **DataSet Details** section provides fields specific to the DataSet record. Only the record type is required. All other fields are optional.

▼ DataRow Details

* DataRow Record Type

Standard

* Use DataRow Field Type

Text Fields

AutomationParameter

☐ Trigger Automation

▼ File Processing Details

* Initial Rows to Skip ⓘ

0

☒ Next Row Contains Headers

☒ Save Header Row

* Rows to Skip after Header ⓘ

0

☒ Override Batch Size

* Batch Size ⓘ

300

Process File

The **DataRow Details** section provides options for the DataRow records:

- **Use DataRow Field Type:** determines if the values are stored in the Text or Long Text Fields.
- **AutomationParameter:** set the AutomationParameter field which would be leveraged in any additional automations you decide to build.
- **Trigger Automation:** is used to initiate any automations that are configured. For the purpose of this application, the AutomationParameter is tagged as completed.

The **File Processing Details** section provides options for handling each “line of data”.

- **Initial Rows to Skip:** the number of rows to skip before processing data.
- **Next Row Containers Headers:** this indicates if the next row contains the column headers
- **Save Header Row:** this indicates if a DataRow Header record should be created
- **Rows to Skip after Header:** the number of rows to skip before processing data.
- **Override Batch Size:** this gives the option to change how many lines are processed and records created at a time. There are limits in place but it is up to the user to understand the limits of their use cases based on any additional automations they implement.

As the file is being processed synchronously, Toast notifications will be displayed on screen.

The screenshot shows the 'My DataSheets' section of the application. A green toast notification at the top states: 'File Processing Complete. File SpreadsheetFlow Security Review DataSet - DataSet (1).csv has been processed and DataSheet DS-00000007 CSV Test created.' Below this, a grey toast notification says: 'Processing File SpreadsheetFlow Security Review DataSet - DataSet (1).csv. Batch of 130 Records Created. Total DataRows Created: 130'. The main table lists data sheets with columns for 'DataSheet Number', 'DataSheet Name', 'File Type', 'Created By', 'Created At', and 'Updated At'. The first row shows 'DS-00000006' as a 'TSV Test' file created by 'Carlos Filip' on '2/4/2023, 8:41 PM'. At the bottom, there are tabs for 'Parse Single List', 'Parse Double List', 'Parse Spreadsheet Data', and 'Parse Custom Text Block'. A 'Close' button is visible in the bottom right corner of the interface.

DataSheet Nu...	DataSheet ...	File Type	Created By	Created At	Updated At
1	DS-00000006	TSV Test	Carlos Filip	2/4/2023, 8:41 PM	2/4/2023, 8:41 PM

Use Case 5.2 TSV Files

To process TSV files, follow the steps in Use Case 5.1 but select the TSV file type.

> File Details

• File Type

☐ CSV (Comma Separated Value)

☒ TSV (Tab Separated Value)

☐ Custom Delimiter

Scenario 6: Upload and Parse a File to a DataSheet

When you open a DataSheet record, you can add a preconfigured quick action or customize your own to upload and parse a new file to the DataSheet.

The screenshot displays the DataSheet interface for a record titled 'DS-00000006'. The top navigation bar includes a search bar and various utility icons. Below the navigation bar, the record title 'DataSheet DS-00000006' is shown alongside a series of action buttons: 'Edit', 'Upload & Parse', 'Select & Parse File', 'Bulk Update DataRows', 'Bulk Delete DataRows', 'Delete', and 'Change Owner'. The 'Upload & Parse' button is highlighted with a red rectangular box. The main content area is divided into two sections: 'Details' and 'DataRows'. The 'Details' section contains a table with the following information:

Details	
Record Type	Standard
DataSheet Name	TSV Test
DataSheet Category	
▼ Reference Values	
DataSheet Reference Text	
DataSheet Reference Number	
Created By	Carlos Hip, 2/4/2023, 8:41 PM
DataSheet Number	DS-00000006
Owner	Carlos Hip
DataRow Headers	0000006915
DataSheet Description	
DataSheet Date	
Last Modified By	Carlos Hip, 2/4/2023, 8:41 PM

The 'DataRows' section is currently empty. On the right side of the interface, there is a 'Files (1)' section showing a single file: 'SpreadsheetFlow Security Review DataSet - DataSet (1)' with a size of 27KB and a date of Feb 4, 2023. A 'View All' link is provided below the file list.

The options you can configure for this action are the same ones you can display or preset in Scenario 6 and 7.

The screenshot shows a web application interface with a modal dialog titled "Upload & Parse". The background shows a "DEMO Spreadsheet" with a "Details" tab selected, displaying metadata for a DataSheet named "DS-00000006". The modal dialog is divided into three sections:

- File Details:** Contains the instruction "Upload and Process a CSV file" and "drag and drop". It features an "Upload Files" button with a cloud icon and a dashed box labeled "Or drop files".
- DataRow Details:** Includes a dropdown for "DataRow Record Type" set to "--None--", a dropdown for "Use DataRow Field Type" set to "Text Fields", a text input for "AutomationParameter", and a checkbox for "Trigger Automation" which is currently unchecked.
- File Processing Details:** Includes a text input for "Initial Rows to Skip" with the value "0", a checked checkbox for "Next Row Contains Headers", an unchecked checkbox for "Save Header Row", and a text input for "Rows to Skip after Header".

A "Process File" button is located at the bottom right of the modal.

Scenario 7: Select and Parse a File

The Select & Parse File allows the user to select a file that is already uploaded to the DataSheet and parse it.

The screenshot displays the DataSheet interface for a specific DataSheet (DS-00000006). The interface includes a top navigation bar with a search bar and various icons. Below the navigation bar, there is a breadcrumb trail showing 'Home' and 'DS-00000006'. The main content area is divided into two sections: 'Details' and 'DataRows'. The 'Details' section contains a table with the following information:

Details	
Record Type	Standard
DataSheet Name	TSV Test
DataSheet Category	
Reference Values	
DataSheet Reference Text	
DataSheet Reference Number	
DataSheet Number	DS-00000006
Owner	Carlos Hip
DataRow Headers	0000006915
DataSheet Description	
DataSheet Date	

The 'DataRows' section is currently empty. On the right side of the interface, there is a 'Files (2)' section showing two files:

- SpreadsheetFlow Security Review DataSet - DataSet (1) (Feb 4, 2023 • 27KB • csv)
- SpreadsheetFlow Security Review DataSet - DataSet (1) (Feb 4, 2023 • 27KB • tsv)

A 'View All' link is present below the file list. The 'Select & Parse File' button in the top action bar is highlighted with a red box.

The options displayed are the same ones observed in Scenario 6.

Search...

Select & Parse File

File Details

Select a File

- ☐ SpreadsheetFlow Security Review DataSet - DataSet (1).unknown (2023-02-05 04:41:43Z)
- ☐ SpreadsheetFlow Security Review DataSet - DataSet (1).csv (2023-02-05 04:51:21Z)

File Type

- ☒ CSV (Comma Separated Value)
- ☐ TSV (Tab Separated Value)
- ☐ Custom Delimiter

DataRow Details

Use DataRow Field Type

Text Fields

AutomationParameter

☐ Trigger Automation

File Processing Details

Initial Rows to Skip ⓘ

0

☐ Next Row Contains Headers

Rows to Skip after Header ⓘ

0

Process File

Scenario 8: Bulk Updating DataRows

SpreadsheetFlow gives you the control to determine when you want to run automations, either automatically on record creation or manually.

The screenshot displays the SpreadsheetFlow web application interface. At the top, there is a search bar and a navigation bar with tabs for 'Home' and 'DS-00000006'. Below the navigation bar, the main content area is divided into two sections. The left section, titled 'DataRows (20+)', shows a table with columns: 'DataRow Number', 'DataRow ...', 'Text001', 'LongText...', and 'AutomationParameter'. The table contains six rows of data, each with a 'Detail' link. The right section, titled 'Files (2)', shows two files: 'SpreadsheetFlow Security Review DataSet - DataSet (1)' (csv) and 'SpreadsheetFlow Security Review DataSet - DataSet (1)' (tsv). A red box highlights the 'Bulk Update DataRows' button in the top right corner of the interface.

	DataRow Number ↑	DataRow ...	Text001	LongText...	AutomationParameter
1	0000006785	Detail	8108		[Completed:TSV1]
2	0000006786	Detail	8109		[Completed:TSV1]
3	0000006787	Detail	8110		[Completed:TSV1]
4	0000006788	Detail	8111		[Completed:TSV1]
5	0000006789	Detail	8112		[Completed:TSV1]
6	0000006790	Detail	8113		[Completed:TSV1]

This feature gives you several options:

- **Update DataRow Fields:** At least one option is needed. You can update the Automation Parameter and/or Trigger Automation.
- **Process Custom DataRow Ranges:** each DataRow has a number and by default, all DataRows are updated. This allows you to update only a specific range of Record Numbers.
- **Override Batch Size:** this enables you to override the batch size. The number of records, size of the records, and number of automations that will run help determine that batch size works for your needs.

06

Edit Upload & Parse Select & Parse File Bulk Update DataRows Bulk Delete DataRows Delet

ows

Files (2) X

Bulk Update DataRows

Automation Parameter

Trigger Automation

Process Custom Range of Records

From Record Number 6,785

To Record Number 6,915

Override Batch Size

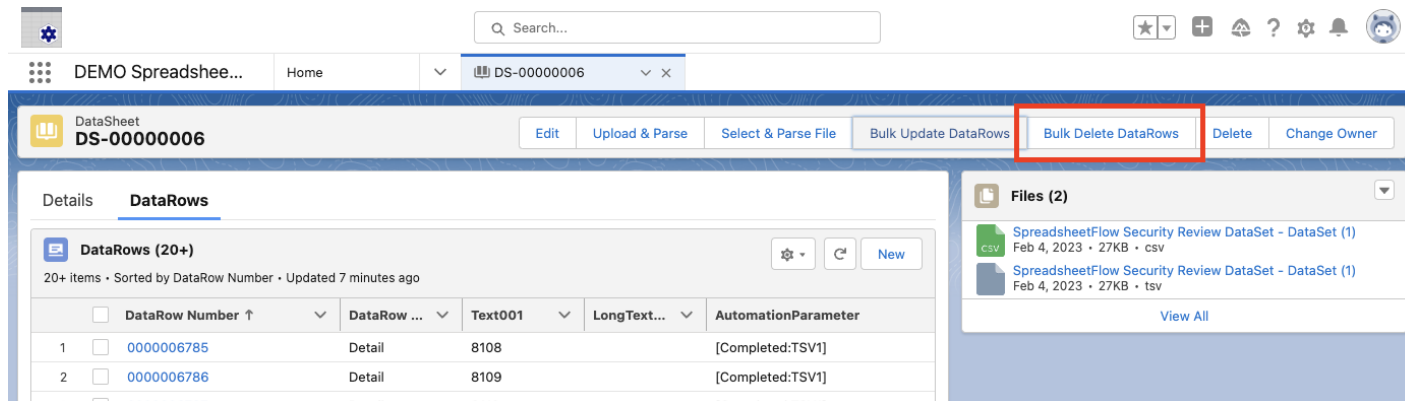
Batch Size 300

Proceed

5785		
5786		
5787		
5788		
5789		
5790		
5791		
5792		
5793		
5794		
5795		
5796	Detail	8119 [Completed:TSV1]
5797	Detail	8120 [Completed:TSV1]

Scenario 9: Bulk Deleting DataRows

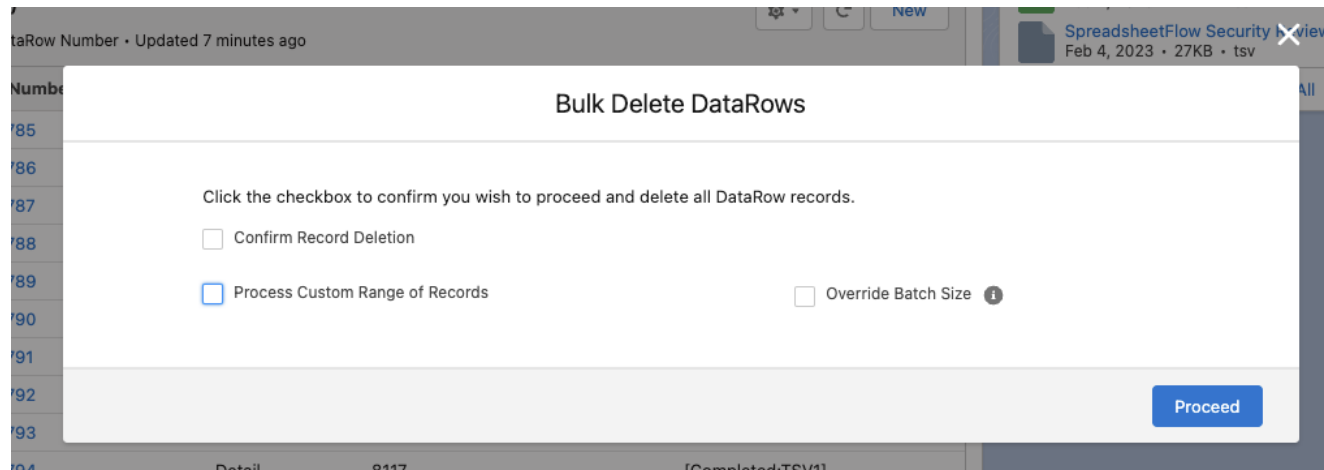
This feature is used to delete DataRows. All DataRows must be deleted before a DataSheet can be deleted.



The screenshot shows the DataSheet interface for DataSheet DS-00000006. The 'Bulk Delete DataRows' button is highlighted with a red box. The interface includes a search bar, a navigation bar with 'Home' and 'DS-00000006', and a toolbar with buttons: 'Edit', 'Upload & Parse', 'Select & Parse File', 'Bulk Update DataRows', 'Bulk Delete DataRows', 'Delete', and 'Change Owner'. The 'DataRows' tab is selected, showing a table with 20+ items. The table has columns: 'DataRow Number', 'DataRow ...', 'Text001', 'LongText...', and 'AutomationParameter'. The 'Files (2)' panel on the right shows two files: 'SpreadsheetFlow Security Review DataSet - DataSet (1)' and 'SpreadsheetFlow Security Review DataSet - DataSet (1)'. A 'View All' link is present below the files.

DataRow Number	DataRow ...	Text001	LongText...	AutomationParameter
1	0000006785	Detail	8108	[Completed:TSV1]
2	0000006786	Detail	8109	[Completed:TSV1]

Options to Bulk Delete DataRows are the same as Bulk Updating. Additionally, there is a confirmation checkbox that must be checked in order to proceed with DataRow deletion.



The screenshot shows the 'Bulk Delete DataRows' dialog box. It contains the following text and options:

Click the checkbox to confirm you wish to proceed and delete all DataRow records.

- ☐ Confirm Record Deletion
- ☐ Process Custom Range of Records
- ☐ Override Batch Size ?

A 'Proceed' button is located at the bottom right of the dialog box.