Yade Sheet

Introduction

Yade Sheet (**Y**et **A**nother **D**ata **E**ditor) is a spreadsheets based data tool, it provides easy to use spreadsheet editor and data access runtime.

Forum | Email | Purchase from Asset Store

Getting Stared

Yade is easy to use. Let's following below steps to go through Yade features.

1. Create Sheet File

Right click in Project Window and select **Create -> Yade Sheet** Menu to create a yade sheet file. Let's name it `TestSheet`. Double click the sheet file to start editing.

2. Edit Sheet file

Double click the yadesheet file will open Spreadsheet Editor. Let's add some data to `A1` and `B1`.

A	B
Hello	Samples

3. Access Cell

Create a script named `Test.cs` in Unity Project Window and copy below content to the script.

```
using UnityEngine;
using Yade.Runtime;

public class Test : MonoBehaviour
{
    public YadeSheetData sheet;

    void Start()
    {
        Debug.Log(sheet.GetCell(0, 0).GetValue());
        Debug.Log(sheet.GetCell("B1").GetValue());
    }
}
```

After save script content and script reimported. Drag the script to any GameObject in Hierachy Window. And then drag the yadesheet file to link the script variable.

Click play button, we should see `"Hello Samples"` will ouput to Unity Editor Console Window.

4. Deserialize sheet to C# objects

We also can deserialize sheet data to c# objects. First, we need creat a class, say it's name is `Data`.

```
public class Data
{
    [DataField(0)] public string A;
    [DataField(1)] public string B;
}
```

We use `DataField` attribute to bind field to sheet columns. And then, we can deserialize the sheet data like this:

```
var list = sheet.AsList<Data>();
Debug.Log(list[0].A);
Debug.Log(list[0].B);
```

Output of above code is the same as section #3

We also can deserialize sheet data to dictionary like this:

```
sheet.AsDictionary<string, Data>(item => item.A);
```

Above code will create a dictionary with first column as keys.

5. Use YadeDatabase

Create `Resources` folder if there are not exists. We right click folder and execute the menu **Create** -> **YadeDatabase**, will create a YadeDatabase asset with default name `YadeDB`.

Let's drag the yadesheet to the inspector of YadeDatabase asset.

6. Use YadeDB

Now we can access the data using YadeDB Utilites like:

```
// The sheet name is TestSheet
YadeDB.Q("TestSheet", "A1").GetValue();
YadeDB.Q("TestSheet", 1, 0).GetValue();
```

7. Full Sample Code

We have use three ways to access sheet data, below is the full sample code:

```
using UnityEngine;
using Yade.Runtime;

public class Test : MonoBehaviour
{
    public YadeSheetData sheet;

    public class Data
    {
        [DataField(0)] public string A;
        [DataField(1)] public string B;
    }

    void Start()
    {
        Debug.Log(sheet.GetCell(0, 0).GetValue());
        Debug.Log(sheet.GetCell("B1").GetValue());

        var list = sheet.AsList<Data>();
        Debug.Log(list[0].A);
        Debug.Log(list[0].B);

        Debug.Log(YadeDB.Q("TestSheet", "A1").GetValue());
        Debug.Log(YadeDB.Q("TestSheet", 1, 0).GetValue());
    }
}
```

Spreadsheet Editor

General

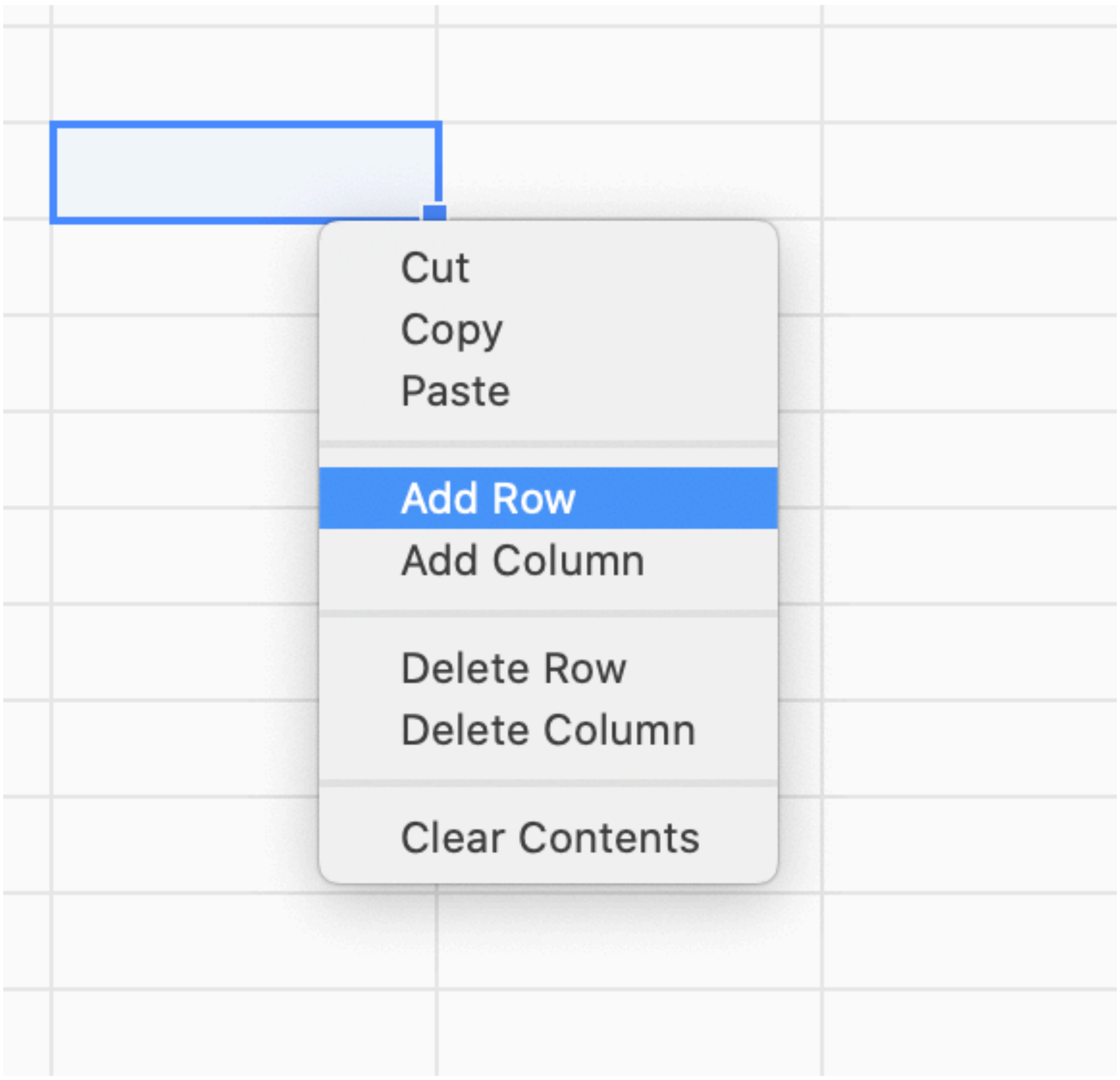
Auto Fill

Left mouse down on the auto fill handle in right bottom corner of selector and drag to auto fill cells.

For now, AutoFill supports fill number series. If we don't want to fill number series, just pressing the `SHIFT` key on keyboard when dragging auto fill handle.

Add Row/Column or Delete Row/Column

Right click on selected cells and select items to add or delete row/columns



Delete Cell

Right click on selected cells and select `Clear Contents` item or Press `Delete` key on the keyboard will delete selected cells

Add Asset Cell

Drag assets from Project Window to yade sheet or using the formula function `ASSET`. For example:

> =ASSET("Assets/icons.json")

Above formula will create a asset cell which point to the asset icon.json

Add Enum Cell

Set raw value of cell following the format below:

> =ENUM("[Type]", "[MemberName]")

- `[Type]` threshold is the full name of the enum type
- `[EnumName]` is the name of a member of enum type

For example:

> =ENUM("UnityEngine.DeviceType", "Console")

Use Formula

Currently, Yade supports below functions and operators:

Functions:

- SUM
- MIN
- MAX
- CONCAT
- AVERAGE
- ASEET (return Unity Object)
- ASSETS (return array of Unity Objects)
- ENUM

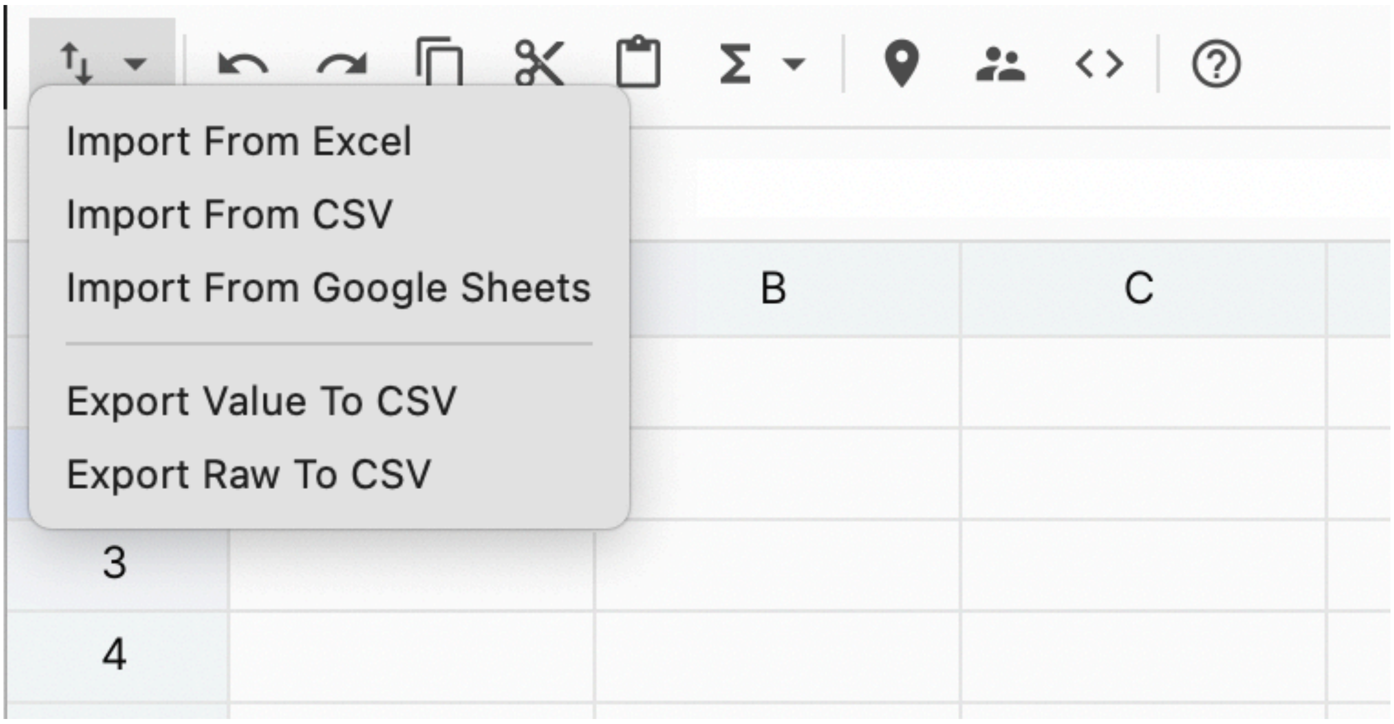
NOTE:

> 1. Drag assets to spreadsheet editor will create **ASSET** function automatically.
> 2. Drag assets to spreadsheet editor **with** `Ctrl` **(Windows) or** `Cmd` **(macOS) key pressing** will create **ASSETS** function automatically

Operators:

- Mins
- Add
- Multiply
- Divide
- Power

Import and Export



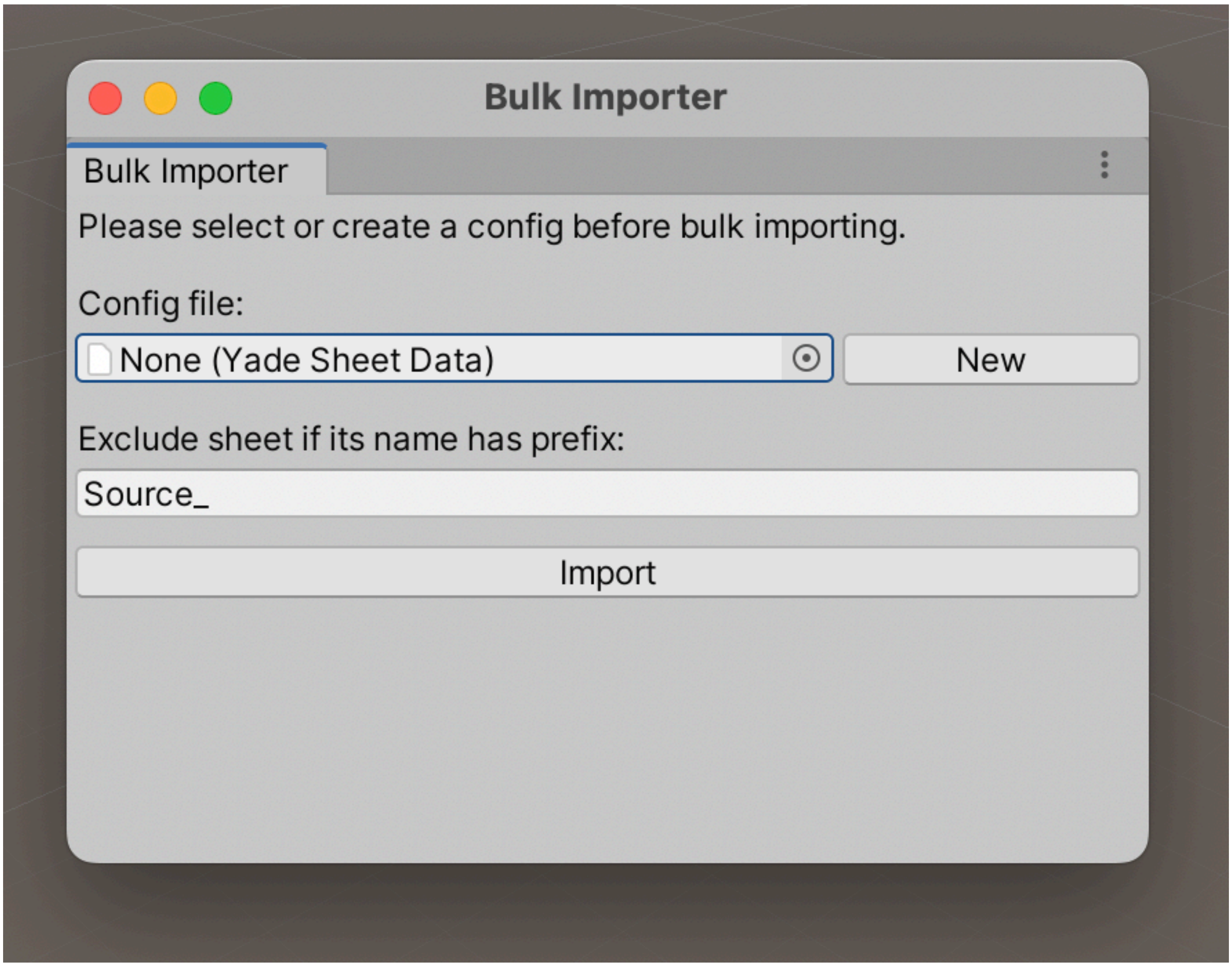
Import From Files

For now, YADE supports import data from:

- Excel files (.xlsx, xls)
- Google Sheets (public links)
- CSV file.

Bulk Import

Open the Bulk Importer window form **Tools -> Yade -> Bulk Importer** menu


Bulk Importer

Bulk Importer

Please select or create a config before bulk importing.

Config file:

None (Yade Sheet Data) | New

Exclude sheet if its name has prefix:

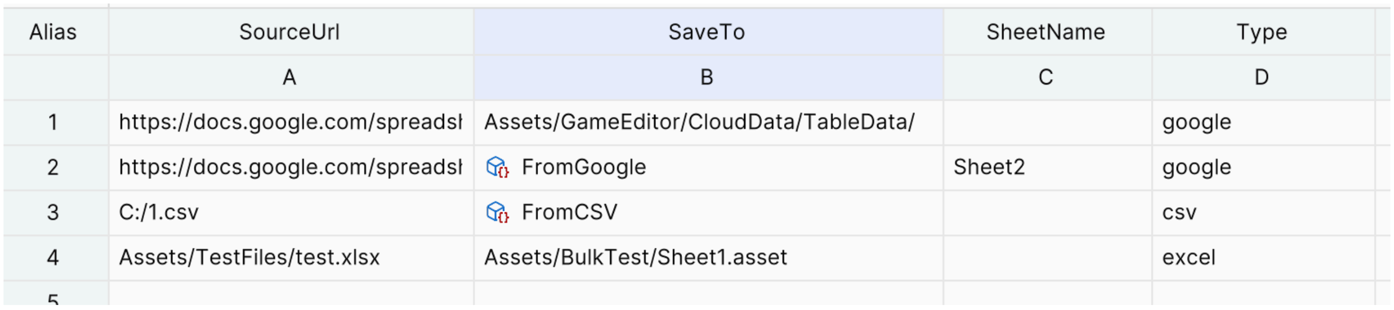
Source_

Import

Select config file or click the `New` button will create an config template file to config data sources.

Exclude sheet name prefix default is `Source_` which means the sheet has name starts with `Source_` will be ignored when importing.

The config file defines data sources show as below samples.



Alias	SourceUrl	SaveTo	SheetName	Type
	A	B	C	D
1	https://docs.google.com/spreadsl	Assets/GameEditor/CloudData/TableData/		google
2	https://docs.google.com/spreadsl	FromGoogle	Sheet2	google
3	C:/1.csv	FromCSV		csv
4	Assets/TestFiles/test.xlsx	Assets/BulkTest/Sheet1.asset		excel
5				

Format of data source:

- **SourceUrl:** the url of data we want to import
- **SaveTo:** the YadeSheet or path of YadeSheet or folder to save
- **SheetName:** the name of sheet we want to import. **Default it empty.** we can leave it as empty if csv file or just one sheets in excel/google sheets or just want to import first sheet
- **Type:** the type of the source. Built-in types are `csv`, `excel`, `google`. **Default is empty.** If type value is empty, Yade will check the file name and detect the type

Rules for the data sources:

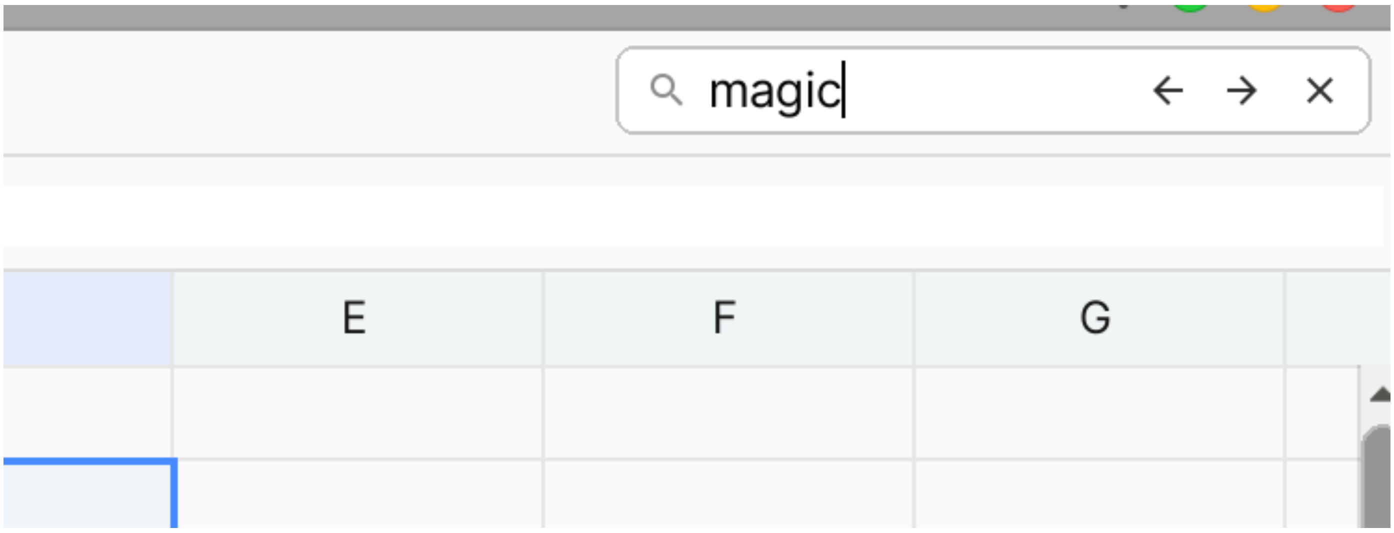
1. For any csv file, Yade will mapping it to one Yade sheet
2. For any Google Sheets or Excel, if the **SheetName** is empty and **SaveTo** is a folder, Yade will auto mapping each sheet to Yade sheet asset
3. For any Google Sheets or Excel, if the **SheetName** is not empty and **SaveTo** is a folder, Yade will mapping the sheet specific to a Yade sheet asset with same name
4. For any google sheets or excel, if **SaveTo** is a Yade sheet asset (Yade sheet or Yade sheet asset file path) and the **SheetName** is not empty, Yade will mapping the sheet specific to the Yade sheet asset
5. For any google sheets or excel, if **SaveTo** is a Yade sheet asset (Yade sheet or Yade sheet asset file path) and the **SheetName** is empty, Yade will mapping the first sheet to the Yade sheet asset

Export To Files

We can export data to CSV files by click the **Export** dropdown button. As below image show, we can export raw data (contains formula if exists) and data (don't contains formula) to CSV file.

Search Sheet

We can search cell content via search input in right top corner of main window.



For search input:

1. `Back Arrow` button will go to previous item of results
2. `Forward Arrow` button will go to next item of results
3. `Clear` button will clear search results
4. `Enter` key will go to next item of results when search input text filed is focusing
5. `SHIFT + ENTER` key will go to previous item of results when search input text is focusing

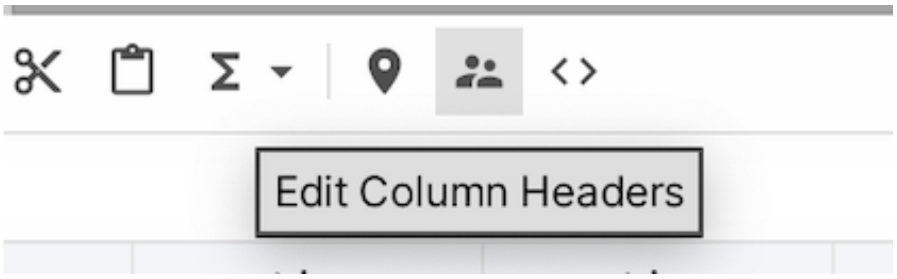
Ping In Unity

Click the **Map Location** icon button in toolbar will ping current sheet in **Project Window** of unity editor.

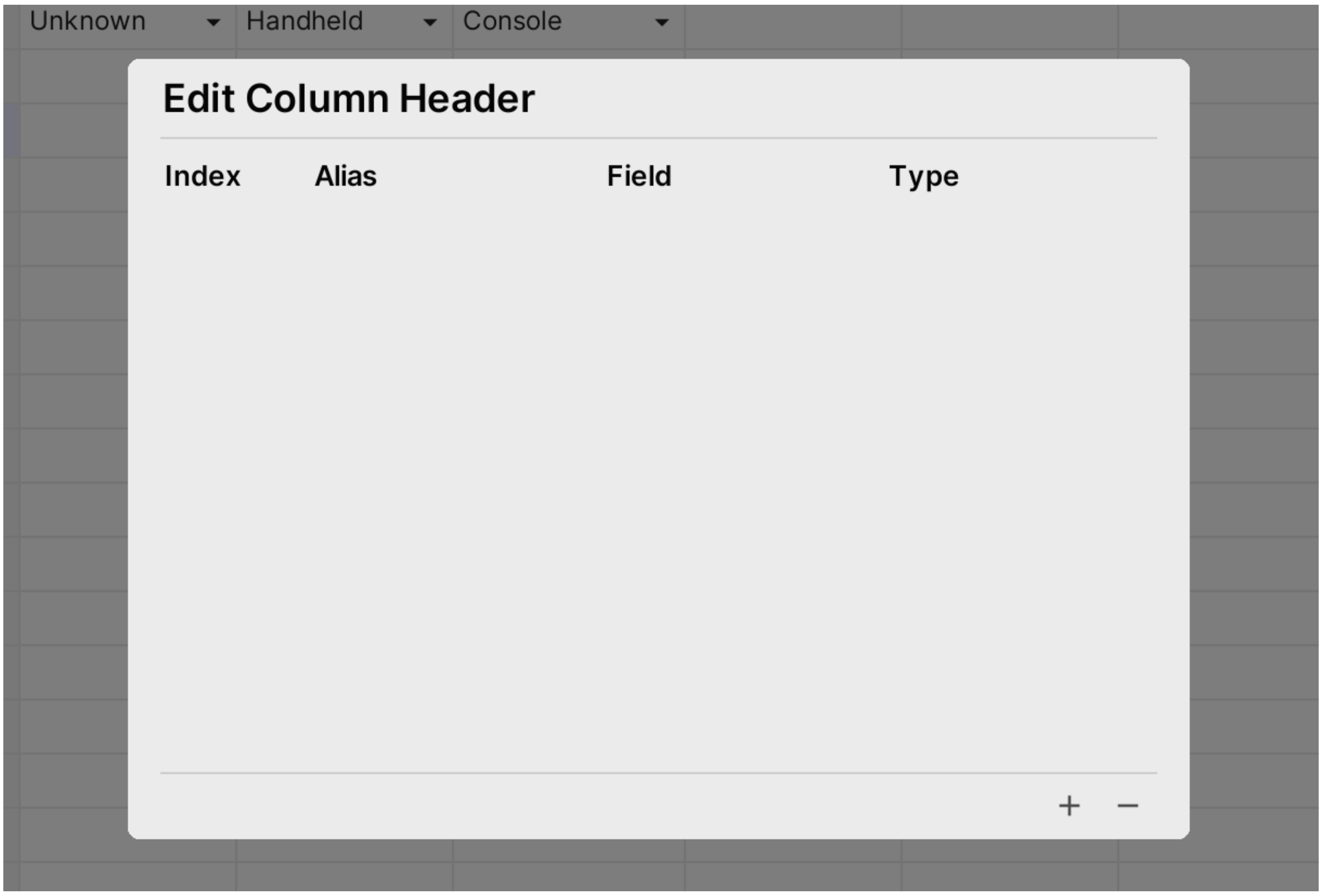
Column Headers

Change Column Headers Settings

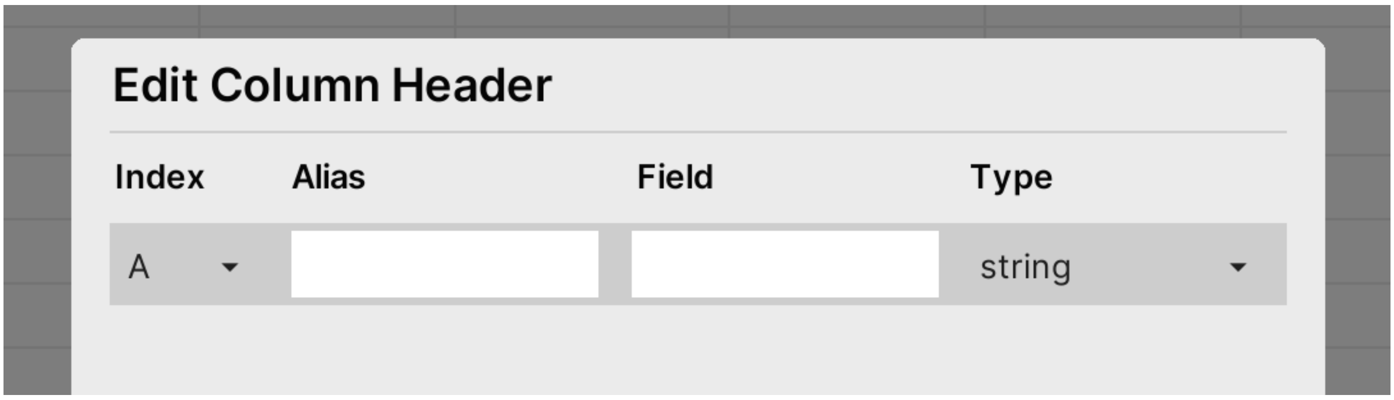
We can open the **Column Header Settings** window by clicking the button on the toolbar as below image show:



Column Header Settings window will display as below in new sheet without header setting before:



Click the + button at right bottom corner will create a column record.



- **Index Field**: Click the drop to select the column we want to set
- **Alias**: Name or short description
- **Filed**: Used for built-in code generator to generate Filed name of class, only can input words only combine with digit, alpha and `_`, and cannot start with digit
- **Type**: Used for built-in code generator to generate type of filed in class. If cannot find the type in dropdown menu, we can add new types to dropdown list by
    - **Method 1**: Call `DataTypeMapper.RegisterType` method in static constructor of any class. If custom class or enum we also can used Method 2.

```
  public class TypeRegister
  {
      static TypeRegister()
      {
          DataTypeMapper.RegisterType<DeviceType>(10001);
      }
  }
```

In the sample, we will register the enum `UnityEngine.DeviceType` to the dropdown list. **Note that, the type key should be unique that larger than 100**

- **Method 2**: If it's custom class, implement interface `ICellParser` and add attribute `TypeKey` to it. For custom enum we only need to add attribute `TypeKey`.

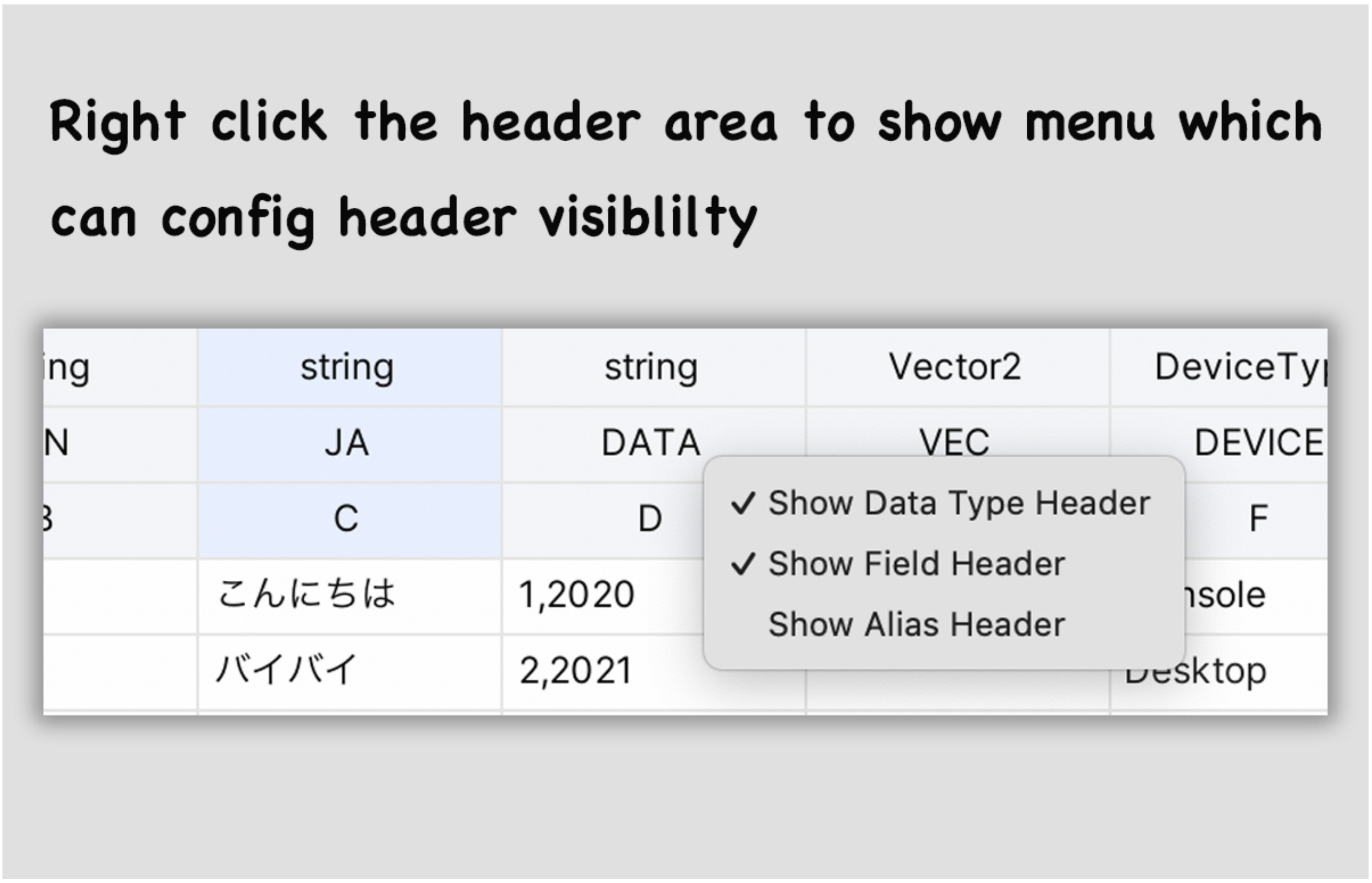
```
[TypeKey(10002)]
public class NumberData : ICellParser
{
    public int index;
    public int year;

    public void ParseFrom(string s)
    {
      var temp = s.Split(new char[] { ',' },
System.StringSplitOptions.RemoveEmptyEntries);
      if (temp.Length == 2)
      {
        index = int.Parse(temp[0]);
```

```
            year = int.Parse(temp[1]);
          }
        }
    }
```

Display Headers

Right click the header area

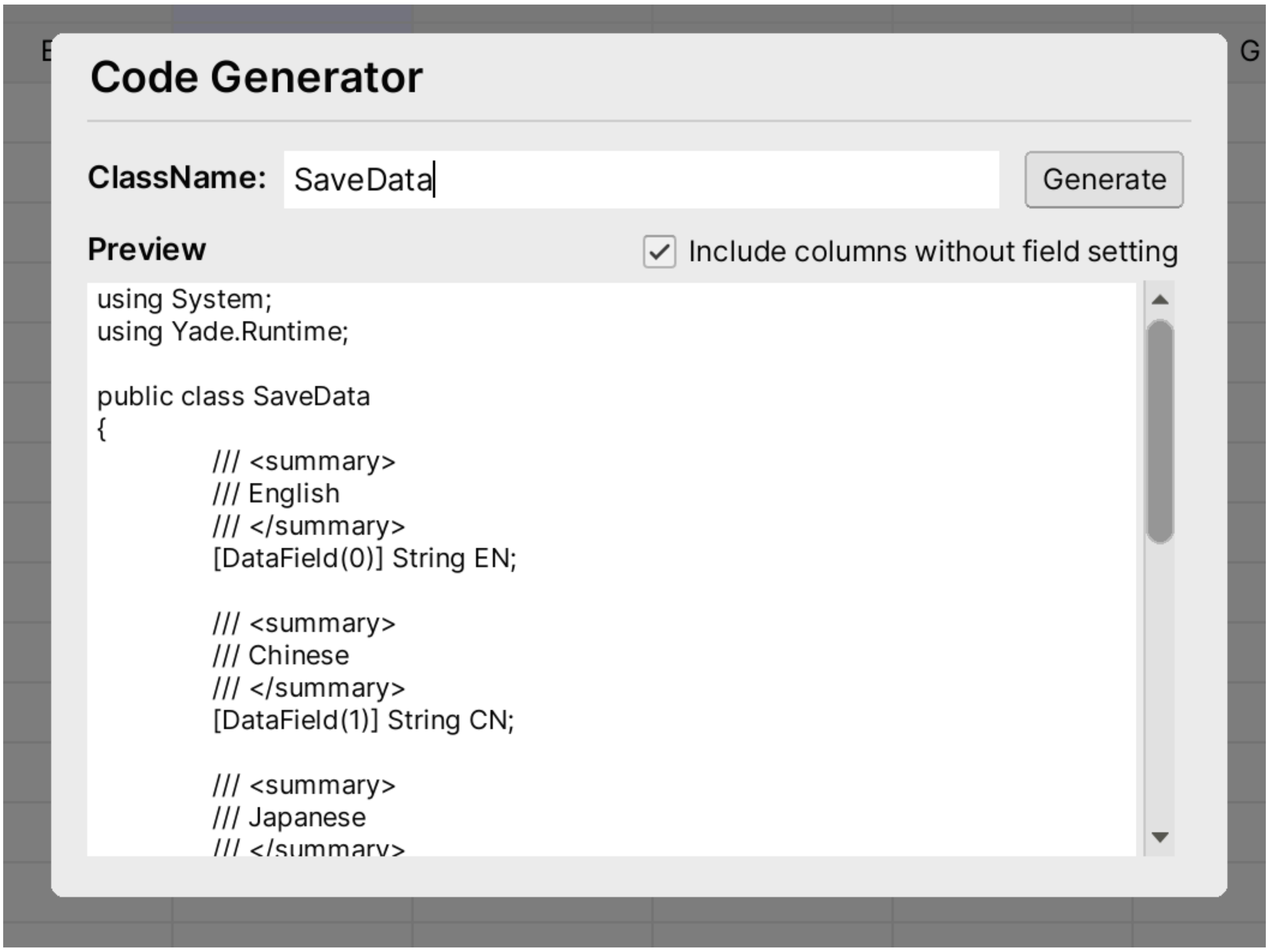


Right click the header area to show menu which can config header visiblilty

ing	string	string	Vector2	DeviceTy
N	JA	DATA	VEC	DEVICE
B	C	D		F
	こんにちは	1,2020		nsole
	バイバイ	2,2021		Desktop

Code Generator

Click the `<>` icon button will open the Code Generator window. Input the class name, preview area will update automatically. **NOTE: class name can only be words only combine with digit, alpha and `_`, and cannot start with digit**.



ClassName: SaveData

✓ Include columns without field setting

```
using System;
using Yade.Runtime;

public class SaveData
{
        /// <summary>
        /// English
        /// </summary>
        [DataField(0)] String EN;

        /// <summary>
        /// Chinese
        /// </summary>
        [DataField(1)] String CN;

        /// <summary>
        /// Japanese
        /// </summary>
```

Yade Runtime

Yade Runtime supports all platforms that support ScriptableObject.

Access Sheet

We can use YadeSheetData class to access the sheet data. Use the method `GetCell` to directly access data or deserialize the data to C# objects using `AsList<T>` or `AsDictionary<T>` extension methods.

Deserializer

Unity Objects and primitive types are supported by deseralizer. If you have custom class type, implements the `ICellParser` interface can supported by Yade deserializer.

Support Custom Data Type

Class that implements `ICellParser` interface can support by deserializer.

```
/// <summary>
/// Custom data type implement ICellParser which can support by AsList() method of
YadeSheetData.
/// If it also has TypeKey attribute, it will show in dropdown menu of type header in
Column Header
```

```
/// Settings Window of YadeEditor
/// </summary>
[TypeKey(10002)]
public class NumberData : ICellParser
{
    public int index;
    public int year;

    public void ParseFrom(string s)
    {
        var temp = s.Split(new char[] { ',' },
System.StringSplitOptions.RemoveEmptyEntries);
        if (temp.Length == 2)
        {
            index = int.Parse(temp[0]);
            year = int.Parse(temp[1]);
        }
    }
}
```

YadeDatabase

YadeDatabase is a group of Sheets, it provide more effective methods to query data of sheets.

YadeDB is the Utilites class for YadeDataBase.

Binary Serialization

Binary Serialization APIs can serialize Yade sheet or Yade Database to byte array and deserialize them from bytes. We can save the byte array to disk or upload to web services which depends on project requirements.

To enable the binary serialization feature, we need to set `BinarySerializerEnabled` property to `true`. For exampple:

```
// Turn on yade sheet instance binary serialization feature
YadeSheetInstance.BinarySerializerEnabled = true;

// Turn on yade database binary serialization feature
YadeDatabaseInstance.BinarySerializerEnabled = true;

// or Use YadeDB Utilities
YadeDB.SetBinarySerializerEnabled(true, dbName);
```

Binary Serialization Mode

Yade have two modes in binary serialization: **Full** and **Incremental**.

- **Full binary serialzation mode** will serialize all sheets data to bytes
- **Incremental binary serialization mode** only serialize **changes of data** to bytes. This is default behaviour of Yade

Serialize and Deserialize

All of YadeSheetData, YadeDatabase and YadeDB Utilites have method called `Serialize()` and `Deserialize()`:

- `Serialize()`: to serialize data to bytes
- `Deserialize()`: to deserialize types to data

Below is a code samples:

```
YadeSheetData sheet = ScriptableObject.CreateInstance<YadeSheetData>();
sheet.BinarySerializerEnabled = true;

sheet.SetRawValue("A1", "Hello");
sheet.SetRawValue(1, 0, "World!");

// Serialize data to binaries
byte[] bytes = sheet.Serialize();

// Deserialize binaries to data
sheet.Deserialize(bytes);
```

Get Online Sheet Data

Yade support two types of online sheet data: **CSV** and **Google Sheets** (public share link).

For the usage, below is a code sample

```
public IEnumerator GetSheetDataFromGoogleSheets()
{
    var url =
"https://docs.google.com/spreadsheets/d/1TOKg3YghXYLyxtaduC0r6q1bqUYoVLfPW8X7zR-
WwkY/edit?usp=sharing";

    // Download data from Google Sheets, 41631710 is the sheet id of the
    // Google Sheet. If no sheet id (gid) specific, the first sheet will
    // be downloaded
    var os = new OnlineGoogleSheet(url, "41631710");
    yield return os.DownloadSheetData();

    // Get YadeSheet
    var sheet = os.GetYadeSheetData();
}
```

After get YadeSheetData instance, we can use Binary Serialization to save data to local or update another sheet by `YadeSheetData.CopyTo` method. Below is sample for save to local by BinarySerialization:

```
public IEnumerator GetSheetDataFromGoogleSheetsAndSaveToLocal()
{
    var url =
"https://docs.google.com/spreadsheets/d/1TOKg3YghXYLyxtaduC0r6q1bqUYoVLfPW8X7zR-
WwkY/edit?usp=sharing";

    // Download data from Google Sheets, 41631710 is the sheet id of the
    // Google Sheet. If no sheet id (gid) specific, the first sheet will
    // be downloaded
    var os = new OnlineGoogleSheet(url, "41631710");
    yield return os.DownloadSheetData();

    // Get YadeSheet
    var sheet = os.GetYadeSheetData();

    // Save To local, We need to use full serialization mode here
    sheet.BinarySerializerEnabled = true;
    var settings = new BinarySerializationSettings();
    settings.Mode = BinarySerializationMode.Full;
    var data = sheet.Serialize(settings);
    File.WriteAllBytes(localFilePath, data);
}
```

Extend Yade

Add Fomula Function

Create a script file under **Editor** folder and create an class inhierted from class `FormulaFunction` . Below Sample code will create a function called `Hello` and it return fixed string `Supper man` .

```
using Yade.Runtime.Formula;

public class Hello : FormulaFunction
{
    public override object Evalute()
    {
        return "Supper man";
    }

    public override string GetName()
    {
        return "HELLO";
    }
}
```

Add A Data Exporter

Create a script file under **Editor** folder and create an class inhierted from class `Exporter` . Below sample create a exporter menu named `Hello Exporter` .

```
using Yade.Editor;

public class HelloExporter : Exporter
{
    public override bool Execute(AppState state)
    {
        // Do logic of exporter: read data from AppState.data and write to other files
        return true;
    }

    public override string GetMenuName()
    {
        return "Hello Exporter";
    }
}
```

Add A Data Importer

Create a script file under **Editor** folder and create an class inhierted from class `Exporter` . Below sample create a exporter menu named `Hello Exporter` .

```
using Yade.Editor;

public class HelloImporter : Importer
{
    public override bool Execute(AppState state)
    {
        // Do logic of importer: loading data from datasource and write them into
AppState.data

        // If return result is true, yade will refresh ui after importing completed
        return true;
    }

    public override string GetMenuName()
    {
        return "Hello Importer";
    }
}
```

Add Context Menu Item

To add an item to right click context menu of selected cells, create a script file under **Editor** folder and create an class inhierted from class `ContextMenuItem` . Below sample create a exporter menu named `AutoFillRandomNumber`.

```
public class AutoFillRandomNumber : ContextMenuItem
{
    public override void Execute(AppState state)
    {
        var selectedRange = state.selectedRange;
        if (selectedRange != null)
        {
            selectedRange.ForEach((row, column) =>
            {
                state.SetCellRawValue(row, column, Random.Range(int.MinValue,
int.MaxValue).ToString());
            });

            state.BindingSheet.UpdateUIBaseOnState();
        }
    }

    public override bool GetEnabledState(AppState state)
    {
        var selectedRange = state.selectedRange;
        if (selectedRange == null)
        {
            return false;
        }

        if (selectedRange.IsWholeColumn() || selectedRange.IsWholeRow())
        {
            return false;
        }

        return true;
    }

    public override string GetMenuKey()
    {
        return "FillRandomNumber";
    }

    public override string GetMenuName()
    {
        return "Auo Fill Random Number";
    }
}
```

Runtime API

IndexHelper

Public Static Methods

Method	Description
IntToAlphaIndex	Convert int to alhpa based index. For example, 0 to A, 1 to B
AlphaToIntIndex	Convert alpha based index to int index. For example, A to 0, B to 1
ToAlphaBasedCellIndex	Get alpha based cell index. For example, (0, 0) => A1, (1, 3) => D2
AlphaBasedToCellIndex	Convert alpha based index to cell index. For example, A1 => (0, 0)

Code Samples

```
// Convert int to alhpa based index. Below sample will return `A`
IndexHelper.IntToAlphaIndex(0);

// Convert alpha based index to int index. Below sample will return 0
IndexHelper.AlphaToIntIndex("A");

// Get alpha based cell index. Below sample will return A1
IndexHelper.ToAlphaBasedCellIndex(0, 0);

// Convert alpha based index to cell index. Below sample will
// return cell index class instance, that both of row index
// and column index are zero
IndexHelper.AlphaBasedToCellIndex("A1");
```

YadeSheetData

Public Properties

Name	Description
FormulaEngine	Formula engine inside sheet
BinarySerializerEnabled	Turn binary serialization on or off

Public Fields

Name	Description
data	Data of rows
columnHeaders	Data of column headers

Public Methods

Name	Description
GetColumnCount	Get columns count of sheet
GetRowCount	Get rows count of the sheet
GetCell	Get cell at specific position

Extension Methods

Name	Description
AsList<T>	Deserialize data to a List
AsDictionary<T>	Deserialize data as Dictionary
SetRawValue	Set raw value of cell
Serialize	Serialize data to byte array
Deserialize	Deserialize byte array to data
CopyTo	Copy sheet data to another YadeSheet

Code Samples

```
// ------Basic Operation--------------------//

// Get row count of yadesheet
int rowCount = yadesheet.GetRowCount();

// Get column count of yadesheet
int columCount = yadesheet.GetColumnCount();

// Get a cell of yade sheet
var cell = yadesheet.GetCell(0, 0)
var cellByAlpha = yadesheet.GetCell("B1");

// Set raw value of a cell
yadesheet.SetRawValue(0, 0, "Hello");
yadesheet.SetRawValue("C1", "=SUM(A1:B1)")


// ------Deserialize to C# Objects----------//

// Define sample data class
public class Data
{
    [DataField(0)] public int A;
    [DataField(1)] public string B;

    // ASSETS() function will mapping to array of Unity Objects,
    // We can use C# array or list to deserialize it.
    [DataField(2)] public Texture2D[] Textures;
}

// Deserialize to list
var list = yadesheet.AsList<Data>();

// Deserialize to dictionary with column A as key
```

```
var dict = yadesheet.AsDictionary<int, Data>(data => data.A);


// ------Binary Serialization--------------//

// Below is samples for binary serialization
// We have to set turn on the binary serialization
// before we modify sheet data
yadesheet.BinarySerializerEnabled = true;

// Then Serialize to sheet
var bytes = yadesheet.Serialize();

// And deserilize later from bytes
yadesheet.Deserialize(bytes);

// -------Data Copy---------------------//
YadeSheetData other = ScriptableObject.CreateInstance<YadeSheetData>();
yadesheet.CopyTo(other);
```

YadeDatabase

A database class that manage yadesheets as datatables.

Public Properties

Name	Description
Sheets	Sheets in the database
BinarySerializerEnabled	Turn binary serialization on or off

Public Methods

Name	Description
Query(sheetName, alphaIndex)	Query sheet to get a cell by alpha index
Query(sheetName, row, column)	Query sheet to get a cell by row and column index
Query<T>(sheetName, predicate)	Query sheet to get a collection of typed data
Query<T>(sheetNames, predicate)	Query sheets to get a collection of typed data
MapQuery<T>(sheetName, predicate)	Query sheet to get a dictionary with the first column of sheet as Keys of the dictionary. Rows with null or empty value of first column will be ignored.
QueryByKey<T>(sheetName, key)	Get a typed class row of sheet by key. **NOTE: The first column of a row is the key**
GetSheetByName(sheetName)	Get yade sheet instance by its name
SetRawValue(sheetName, alphaIndex, rawText)	Set raw value of a cell in sheet. **NOTE: ASSET formula does not support in build**
SetRawValue(sheetName, row, column, rawText)	Set raw value of a cell in sheet. **NOTE: ASSET formula does not support in build**
Serialize	Serialize data to byte array
Deserialize	Deserialize byte array to data

Code Samples

```
// Define sample data class
public class Data
{
    [DataField(0)] public string A;
    [DataField(1)] public string B;

    // ASSETS() function will mapping to array of Unity Objects,
    // We can use C# array or list to deserialize it.
    [DataField(2)] public Texture2D[] Textures;
}

var sheetName = "TestSheet";

// Query cell by alpha index in a sheet
database.Query(sheetName, "A1");
database.Query(sheetName, 0, 0);
```

```
// Query sheet by first column as Key. It will
database.QueryByKey<Data>(sheetName, "KeyOfDictionary");

// Query sheet as a collection with or without conditions
database.Query<Data>(sheetName);
database.Query<Data>(sheetName, item => item.A == "Hello");

// Query sheets as a collection with or without conditions
var sheetNames = new string[] { "TestSheet", "TestSheet1" };
database.Query<Data>(sheetNames);
database.Query<Data>(sheetNames, item => item.A == "Hello");

// Query sheet as dictionary with first column as Keys with
// or without conditions
database.MapQuery<Data>(sheetName);
database.MapQuery<Data>(sheetName, kvp => kvp.Value.B == "World");

// Set raw value of a cell
database.SetRawValue(sheetName, "A1", "Hello");
database.SetRawValue(sheetName, 0, 2, "=SUM(A1:B1)");

// Serialize and Deserialize
database.BinarySerializerEnabled = true;

var bytes = database.Serialize();
database.Deserialize(bytes);
```

YadeDB

YadeDatabase Utilites. The YadeDatabase asset should be binplaced under **Resources** folder. otherwise, this utilites will not working for the database.

Public Fields

Name	Description
DefaultDB	Deafult databse in Resources folder. Value is "YadeDB"

Public static methods

Name	Description
GetDatabase(database)	Get database by name
Q(sheetName, alphaIndex, database)	Query sheet to get a cell by alpha index
Q(sheetName, row, column, database)	Query sheet to get a cell by row and column index
Q<T>(sheetName, predicate, database)	Query sheet to get a collection of typed data
MapQ<T>(sheetName, predicate, database)	Query sheet to get a dictionary with the first column of sheet as Keys of the dictionary. Rows with null value of first column will be ignored.
QByKey<T>(sheetName, key, database)	Get a typed class row of sheet by key. NOTE: The first column of a row is the key
SetRawValue(sheetName, alphaIndex, rawText)	Set raw value of a cell in sheet. **NOTE: ASSET formula does not support in build**
SetRawValue(sheetName, row, column, rawText)	Set raw value of a cell in sheet. **NOTE: ASSET formula does not support in build**
Serialize	Serialize data to byte array
Deserialize	Deserialize byte array to data
SetBinarySerializerEnabled	Turn binary serialization on or off

Code Samples

```
// "YadeDB" is the default database name in yade.
// It's same value as YadeDB.DefaultDB const string
var dbName = "YadeDB";
var db = YadeDB.GetDatabase(dbName);

// ------------------------------------------------//
// ---BELOW SAMPLES WILL USE DEFAULT DATABASE -----//
// ------------------------------------------------//
var sheetName = "TestSheet";

// Define sample data class
public class Data
{
    [DataField(0)] public string A;
    [DataField(1)] public string B;
}

// Query a cell
```

```
var cell = YadeDB.Q(sheetName, "A1");
var cellByIndex = YadeDB.Q(sheetname, 0, 0);

// Query a cell by key, the key is the first column value
var data = YadeDB.Q<Data>(sheetName, "Hello");

// Query sheet as a collection with or without conditions
YadeDB.Q<Data>(sheetName);
YadeDB.Q<Data>(sheetName, item => item.A == "Hello");

// Query sheet as dictionary with first column as Keys with
// or without conditions
YadeDB.MapQ<Data>(sheetName)
YadeDB.MapQ<Data>(sheetName, kvp => kvp.Value.B == "World");

// Set raw value of a cell
YadeDB.SetRawValue(sheetName, "A1", "Hello");
YadeDB.SetRawValue(sheetName, 0, 2, "=SUM(A1:B1)");

// Serialize and Deserialize
YadeDB.SetBinarySerializerEnabled(true);
var bytes = YadeDB.Serialize();
YadeDB.Deserialize(bytes);
```

OnlineCSVSheet

Download data form online CSV content and parse to Yade Sheet.

Public Methods

Name	Description
DownloadSheetData	Download data form web
GetYadeSheetData	Parse data to Yade Sheet

Code Samples

```
var url = "https://www.amlovey.com/uas/YadeTest.csv";
var onlineSheet = new OnlineCSVSheet(url);
yield return onlineSheet.DownloadSheetData();
var sheet = onlineSheet.GetYadeSheetData();
```

OnlineGoogleSheet

Download data from Google Sheet (public share link) and parse to Yade Sheet.

Public Methods

Name	Description
DownloadSheetData	Download data form web
GetYadeSheetData	Parse data to Yade Sheet

Code Samples

```
var url =
"https://docs.google.com/spreadsheets/d/1TOKg3YghXYLyxtaduC0r6q1bqUYoVLfPW8X7zR-
WwkY/edit?usp=sharing";
var os = new OnlineGoogleSheet(url);
yield return os.DownloadSheetData();
var sheet = os.GetYadeSheetData();
```

Playmaker Support

After Playmaker installed in project, we can see the actions in Action Browser. Below is the actions of Yadesheet.

Action	Description
Get Cell Value	Get value of cell
Get Cell Raw Value	Get raw value of cell
Cet Cell Unity Object	Get unity object (texture, material, etc) of the cell
Get Cell By Index	Get cell by row and column index
Get Cell by Alpha Index	Get cell by alpha based index of row and column
Set Cell Raw Value by Alpha Index	set raw value of cell by alpha based index of row and column
Set Cell Raw Value	Set raw value of cell by row and column index
Yade DB Query Cell By Alpha Index	Query cell by alpha index using YadeDB
Yade DB Query Cell By Index	Query cell by index using YadeDB
Yade DB Set Cell Raw Value By Index	Set cell raw value by index using YadeDB
Yade DB Set Cell Raw Value By Alpha Index	Set cell raw value by alpha index using YadeDB

FlowCanvas Support

Download the FlowCanvas support Unity Package HERE

Action	Description
GetSheetCellByAlphaIndex	Get cell by alpha based index
GetSheetCellByIndex	Get cell by index
SetCellRawValueByIndex	Set raw value of cell by row and column index
SetCellRawValueByAlphaIndex	Set raw value of cell by row and column index
YadeDatabaseGetSheetCellByAlphaIndex	Get a cell from sheet by alpha index using Yade database
YadeDatabaseGetSheetCellByIndex	Get a cell from sheet by index using Yade database
GetCellValue	Get value of a cell
GetCellRawValue	Get raw value of a cell
GetCellUnityObject	Get unity object value of a cell

FAQ

Q: Odin Inspector show reference of Yadesheet type field of MonoBehaviour scripts is missing after double click or exiting play mode

A: Please let your MonoBehaviour script inheirt the `YadeSerializedMonoBehaviour` class, this will fix the issue.

Support

please visit https://www.amlovey.com/YadeDocs for more details. Or drop an email to amlovey@qq.com.